Helsinki University of Technology
Department of Computer Science and Engineering
Espoo 2008

PROCEEDINGS OF THE SEMINAR ON NETWORK SECURITY 12.12.2008

From End-to-End to Trust-to-Trust

Sasu Tarkoma, Jani Heikkinen (eds.)

Tutors: Ronja Addams-Moring, Tuomas Aura, Jani Heikkinen, Mikko Pitkänen, Teemu Rinta-Aho, Petri Savolainen, Xiang Su, Sasu Tarkoma, Jukka Valkonen

Helsinki University of Technology
Faculty of Information and Natural Sciences
Department of Computer Science and Engineering
Data Communications Software

Teknillinen korkeakoulu
Informaatio- ja luonnontieteiden tiedekunta
Tietotekniikan laitos
Tietoliikenneohjelmistot

Preface

The aim of the Fall 2008 Seminar of Network Security is to examine the implications of a recently proposed paradigm of Trust-to-Trust from different viewpoints including protocols, network design, applications, and services.

The original architectural principles for the Internet were the End-to-End and robustness principles. The former, in its original expression, placed the maintenance of state and overall intelligence at the edges, and assumed the Internet that connected the edges retained no state and concentrated on efficiency and simplicity. Today's real-world needs for firewalls, NATs, Web content caches have essentially modified this principle.

The End-to-End principle implies that application logic is executed by endpoints of communication and follows secondary principles such as minimality, generality, simplicity, and openness. In today's Internet, logic has been distributed between end hosts, middleboxes such as firewalls and NATs, and trusted 3rd parties, such as Web sites. It follows that for the end user, it is crucial that any application functionality related to the user's activities is executed in a trustworthy manner.

This observation has led to a reformulation of the original End-to-End principle called Trust-to-Trust (T2T). T2T gives an opportunity for principals to choose where application logic is executed by trusted points. The proposal for T2T has created a lot of discussion in the networking community and it remains to be seen how trust is reflected in the future Internet architecture.

The papers cover a lot of ground around the course theme including security protocols, ad hoc networking, peer-to-peer, and data-centric networking. The papers consider existing state of the art and in some cases offer new insights into distributed systems and their trustworthiness.

The students and their tutors have done a very good job in preparing the papers. I would like to thank Jani Heikkinen for help with planning and organizing this seminar.

Prof. Sasu Tarkoma

Department of Computer Science and Engineering

Helsinki, December 4th, 2008

Table of Contents

Towards an Architectural Design for the Future Internet

Parth Amin
Helsinki University of Technology
piamin@cc.hut.fi

Abstract

This paper extends the Nth stratum concept [3], which was proposed as a novel design for the future internet by applying Accountable Internet Protocol (AIP) [1] to the connection stratum of the Nth stratum architecture. The proposed design for the future internet has the potential to supersede the current internet and the current telecommunication network making it more scalable, interoperable, self-managed, secure, addressable, accountable and will also support seamless mobility. It also has a potential to increase the competitiveness of the networking industry and to improve the quality of life for citizens by creating a family of dependable and interoperable networks providing direct and ubiquitous access to information.

KEYWORDS: Future internet architecture, The Nth stratum concept, AIP, interoperability, security, mobility

1 Introduction

TCP/IP architecture forms the basis of the present internet cloud connecting various communicating system. By now it is clear that this architecture is not interoperable and not scalable, resulting to the internet patches such as NAT, IPSec, Mobile IPv6. Various architectures have differences in the support of basic features like QoS, mobility and security. Host connected to the internet cloud is not identified uniquely and also has a poor support for multi homing. Last but not the least, lack of accountability is one of the major pitfalls of the existing internet architectures as there is no fundamental ability to associate an action with the responsible entity. It has resulted to the security related issues such as source spoofing, denial of service, route hijacking and route forgery. All these limitations of the existing architectures, point to a need for a common architecture, which will form the basis for the future communicating system. Revolutionary research is being carried out in the areas of the designing the future internet by the industrial and academic players of EU, US and Asia in the form of various projects such as GENI (Global Enviornment for the Network Innovations), FIND (Future Internet Network Design), 4WARD, TRILOGY, ICT SHOK Future Internet, Asia Future Internet, etc. National Science Foundation (NSF), FP7 (7th Framework Programme) and National Institute of Information and Communication Technology (NICT) have major roles in managing their respective future internet projects in US, EU and Japan. Various designs have been proposed, of which one of them is The Nth stratum concept proposed as a part of EU funded project called 4WARD. This novel concept claims to support the continuous high pace of innovations in all aspects of different communication systems keeping the costs of deployment down and also supporting the interoperability between the present counterparts.

We hereby propose to apply the Accountable Internet Protocol (AIP) to the connection stratum of the Nth stratum architecture and thereby solve security issues such as spoofing, denial of service, route hijacking and route forgery. It will also make the architecture more scalable, interoperable, self managed, addressable, accountable and also support seamless mobility.

The rest of this paper is organized as follows. Section 2 briefly outlines the Nth stratum concept. It is followed by the brief explanation of AIP in Section 3. Section 4 proposes an architectural design towards the future internet by applying the AIP to the connection stratum of the Nth stratum architecture and solve the security and mobility issues. In Section 5, some of the limitations and open issues of the proposed architecture are discussed. Finally the paper concludes with the optimistic approach with the proposed design and its advantages for the future internet users.

2 The Nth stratum concept

The Nth stratum concept [3] is a *Clean slate approach* proposed in the EU Project 4WARD for the future communication system, which will revolutionize the existing communicating systems with the common interoperable architectural framework. It is designed keeping QOS, security and mobility as the key features for the future internet. A stratum is the fundamental entity of the Nth stratum concept. It is similar to the layer of the traditional OSI based communication system, as it provides services to the other stratum of its own communication system. Each stratum is formed of set of nodes/a node having the data processing functionality and a medium connecting the nodes. A stratum may need to get services from one or more strata to perform its own functionality. Each of the strata has Stratum Service Point (SSP), which defines the set of services offered by a stratum to other strata. It also offers information about specific properties and features of its own stratum. SSP provides the services to the other strata using the Stratum Transition Point (STP). STP translates the services offered by the other stratum to the form that is required by its own stratum. Moreover, there can also exist a peering relation between the two similar type of strata in the different communicating system. The peering relation is defined by the Stratum Gateway Point (SGP), which maps the parameters and other identifiers between the two different strata. Fig 1 depicts the principle strata for any communicating system consisting of horizontal and vertical strata. Horizontal strata include virtual machine stratum, connected endpoints stratum, flow stratum and information stratum providing the basic functionalities of the traditional OSI layered approach. Self management properties inherited by these horizontal strata along with the inheritance of the other abstract vertical classes (strata) like quality of service, security and mobility, differentiates the horizontal strata of the Nth Stratum Architecture from the traditional OSI layered model. Moreover the vertical strata also include the knowledge and governance stratum which are mainly responsible for managing the horizontal strata. Self-management point in the direction of being self-organized, self-configured, self-healed, self-protected, self-optimized, and self-tuned in order to meet performance and policy objectives. In the following subsections, further aspects of the stratum are described.

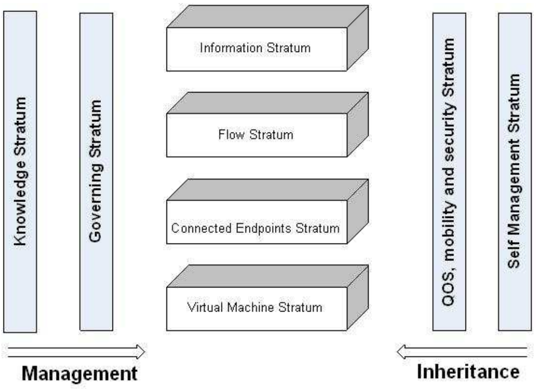

Figure 1: The Nth Stratum Architecture[3]

2.1 Inheritance

Principle of inheritance forms the basis of the Nth stratum architecture. It will help to define generic features and properties of a communication system and will also help to keep the framework consistent and coherent across the different communicating systems. QoS, security, mobility, self management parameters and policies are required to be generically defined at the top level and subsequently has to be inherited depending on the specific need of a stratum of a communicating system based on the principle of inheritance. Using a abstract stratum, one can define a stratum that is either partially defined or not defined at all. The only constraint for the abstract stratum is that one cannot execute the same in the run time environment.

2.2 Stratum instantiation

Stratum instantiation occurs within the nodes of a communicating system. As the communication between the two systems starts, different strata are instantiated on both the sides. Stratum also inherits and implements the properties of self-management, self-organization, self-configuration and self-healing. Information from the knowledge stratum also helps in the organization of nodes and SGP.

2.3 Stratum aggregation/concatenation

Stratum aggregation is applied to the two or more instantiated strata, which are concatenated via an SGP based on the domain border detection.

2.4 Horizontal and Vertical Strata

Horizontal strata deal with the connectivity/service/application aspects of a system, whereas vertical strata are responsible for the governance, performance, consistency and coherency of the system. Vertical strata impact the horizontal strata, as they are ones which are responsible for the overall governance and monitoring. Moreover vertical strata are implemented as a set of libraries, which will be inherited by the horizontal strata to build concrete and generic network architecture. Horizontal strata are managed by the vertical strata such as governing stratum and knowledge stratum.

2.5 Management Aspects

Self-adaptable and auto-piloted are the features seen as a part of the future internet design. It is expected that the network will dynamically update itself with the relevant protocols and algorithms on the basis of the user's need and network context in the real time. For example when there is a congestion detected in the network, the network adapts itself with QoS configuration or algorithms to adapt to the traffic conditions. Vertical strata such as governing stratum and knowledge stratum are responsible for the dynamically updating the horizontal strata on the real time basis. Governing stratum configures the required set of horizontal strata for a specific communicating system depending on the user's requirement. It uses the information obtained from the knowledge stratum, which is aware of the detailed view of the network state and also the individual stratum. The performance of a network depends on the factors such as efficiency of the governing stratum and knowledge stratum and also the set of algorithms and protocols supported at each individual horizontal stratum.

3 Accountable Internet Protocol (AIP)

Accountable Internet Protocol (AIP) [1] is a concept aimed at replacing the current IP based internet, with the two or more levels of flat addressing structure of the form AD:EID. AD is the identifier for the host's autonomous domain and EID is the globally unique host identifier. Both of these identifiers are derived from the public keys held by the domain and the host respectively, removing the need of the globally trusted authority. It thus has a globally unique host identifier, identifying all the hosts connected to the internet uniquely. Accountability is the first order property provided by AIP, which is based on the hierarchy of self certifying addresses.

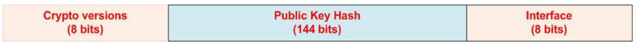

Figure 2: The structure of an AIP address[1]

It is claimed to be a solution to the IP layer security problems such as source spoofing, denial of service attack, route hijacking and route forgery.

IP layer of the internet has lot of vulnerabilities such as hijacked routes, DoS attacks and source spoofing. Various solutions are proposed for the same such as: complicated mechanisms that change the free access model of the internet, external sources of trust and operator vigilance. These proposed solutions seem to be the ongoing continuation of applying patches to the internet cloud. On the contrary, AIP is proposed as a solution written from scratch and addressing the key IP based architectural problem. AIP based internet will have a ability to associate an action with the responsible entity thereby assuring the missing property of the user accountability.

AIP has two or more levels of flat addressing structure, which is the much closer to the internet's original incarnation rather than today's Classless Inter-Domain Routing based aggregation. The uniqueness of the address lies in the self certification, without relying on any globally trusted authority. Both hosts and domains can prove their own addresses themselves. Although, such a flat addressing mechanism of AIP is not at all seen as a threat to the long term scalability of the internet, but one cannot deploy the same on the present router infrastructure. It makes pretty simple for routers to route the traffic, as it has to just look at the network portion of the address to route the data, till the data reaches the destination network.

AIP design assumes that internet is formed of separate administrative networks. Each administrative unit is formed of one or more Accountability Domains (ADs) having its own unique identifier. Each host is also assigned a globally unique end point identifier (EID). Thus AIP address of a host belonging to the domain AD would be AD:EID. As shown in the fig 2, last 8 bits of the EID are the interface bits, by which AIP host has a support of multi homing. It allows the host to attach multiple times to the same AD. Lastly, AIP also supports multiple levels of the hierarchy in the ADs, resulting to the address of AIP host as AD1: AD2: ... :EID.

3.1 Self Certification

AD is the hash of the public key of the domain and EID is the hash of the public key of the corresponding host. Usage of the self certifying address at the network layer, makes AIP first of its kind and it also has unique feature of providing accountability at the network layer. Source accountability mechanism is based on the unicast reverse path forwarding and EID verification. Last but not the least, it also supports gradual evolution of digital signature to cope with the weakening of the earlier schemes based on the crypto versions.

3.2 Forwarding and Routing

Packets have the destination address set to AD:EID. Routers use only the destination AD to route the packets, until they reach the destination AD. Once the packet is in the desired AD, routers use the EID to further route the packet. Similarly in a case, where there are more than one ADs present in the destination address, routers route the packet based on the first AD in the destination stack. Moreover, the interdomain routing will also happen in more or less as the same way as it happens today, that is based on the AD granularity using BGP. Lastly, the packet will be routed to the destination EID using the interior routing protocol such as OSPF within the AD.

3.3 DNS and Mobility

AIP supports multiple addresses for a single host. For example, if a host is connected to multiple ADs, then the host will have multiple AD:EID addresses in each domain. Similarly in case, where a host has multiple interfaces in the same domain, the last 8 interface bits in EID are used to distinguish the host address. Mobility support in AIP is based on the self certifying end point identifier (EID). Transport protocols on the top of the AIP layer work on the basis of the source and destination EIDs which remain unchanged while hosts roam from one AD to another. Moreover, mobility support is based on TCP Migrate [5] and HIP [4]. Mobile AIP host update the DNS with their current AD.

3.4 Source Accountability: Detecting and Preventing source spoofing

Source spoofing is an attack, in which the attacker uses the source address that has been assigned to another host. The attack becomes more difficult to detect, when the attacker is also able to receive packet while spoofing. AIP based hosts prevent such a spoofing attacks based on the self certifying addresses, in which routers drop the packets if the source address is spoofed. The technique is based on unicast reverse path forwarding (uRPF) [2]. Packets are accepted by the router, if the route to the packet's source address, points to the same interface on which the packet arrived. This concept works to prevent spoofing by the single homed clients, but it cannot cope with the multi homed clients. So a second mechanism is included along with uRPF to prevent spoofing against the multi homed host in the AIP. Public keys are used to verify the source address of a packet at couple of places. First, each first hop router (that is trusted by the network operator) verifies that its directly connected hosts are not spoofing. Moreover, each AD through which a packet passes verifies the specified source address.

EID verification:

If the first hop router R has not verified the source host, then it drops the packet and sends the verification packet V to the source. Verification packet V is made up of source and destination address of the packet, packet's hash and the interface on which packet arrived. R also attaches HMAC, with the V as a signature. The sender is also supposed to prove its identity EID by signing V with the private key associated with the EID. If the sender is able to produce the correct signature then R cashes this information and thereby also allow the subsequent packets from that sender as well. The sender is suppose to resend the packet that made the R to send the verifcation packet, as R drops all unverified packets. Hosts should not postively respond to the verfication request for the packet that it did not originate. To avoid the same, host must maintain the cache of the recently sent packets.

AD verification:

When a AD A receives a packet from the adjacent AD B, it must first verify that the source address is valid. If AD A trust the AD B, then it can forward the packet without verifying the same. If AD A does not trust B, then A performs uRPF checks to determine whether the packet arrived on the interface, which the route to the source would take. If the check succeeds, A forwards the packet, else A drops the packet and sends a verification packet to AD:EID as in the case of the EID verification. If the sender EID is able to reply positively, then the corresponding entry is added to the subsequent packets from AD:EID to pass, when they arrive on the verified interface.

Border routers must verify the incoming packets, whereas the interior routers can trust their border routers and need not do the same verification again. Peering router verification can be based on the bilateral contractual agreement between the two parties. To keep the number of cache entries lower, a router with more than threshold number of entries T for a single AD will replace all the individual AD:EID entries with an wildcard denoted as AD: *.

3.5 Shut-off protocol

DoS attack based on the source address spoofing are prevented by the AIP based verification. But, the flooding attack to a victim, with a traffic from the compromised host remains unaffected. To handle the same, the victim host sends the explicit *shut off* message to a host sending such a traffic.

3.6 Key Management

AIP needs to handle the issues related to the key management such as: key discovery, individual key compromise and cryptographic algorithm compromise. As the host's key is simply its address, the key is obtained once its address is known. Moreover, address can be known from the DNS server as it is known today. Individual key compromise includes protecting against compromise, detecting compromise and also dealing with it. To decrease the chance of compromise, hosts and domains should follow established key management practices, such as using time limited secondary keys for all online signings and keeping the primary keys offline, in a safe place. Moreover, if a host key is compromised, then the host merely adopts a new key and inserts it into the DNS record. Similarly if the domain key is compromised, then the domain has to revoke its key based on the inter domain routing protocol and via public registries. Public registries are maintained that has information of the peers for each AD and the ADs to which each EID is bound to detect the key compromise. Similarly to cope with the cryptographic algorithm compromise, each AIP address and every registry entry contains its own crypto version field. Versioning of the address supports the gradual phasing and applying the new algorithms.

4 Applying AIP to the Nth Stratum Architecture

Although IP has played a significant role for the existing internet, since more than thirty years, but due to various limitations as described, there lies a need to upgrade the current internet. The proposed architectural design for the future internet applies AIP to the connection stratum of the Nth stratum architecture as shown in the fig 3. It has a potential to supersede current internet and current telecommunication network, by the unique architectural framework supporting both of them. Goals of the 4ward project [6] that can be achieved are: scalability, extensibility, interoperability and self-management. Moreover, applying AIP to the Nth stratum architecture will also support seamless mobility for the users and also effectively address issues associated with the security, privacy and mobility at the network layer. The proposed architecture is anti spam efficient and also uniquely identifies the host connected to the internet.

The 4ward project is seen as an answer to the future internet challenge. The present internet lacks enough possibility to design, optimize and interoperate new networks forcing a convergence to an architecture that is suboptimal for many applications basically not supporting innovations within itself. So we can say that internet has reached to a critical point in the impressive development cycle requiring a major change. The Nth Stratum Architecture proposed as a part of the 4ward project is a design that is inter-operable, scalable and extensible. It provides these features as it is a generic framework which can be applied to the future communicating system. Moreover, the utility of networks is enhanced by making them self-managing using the vertical strata such as governance and knowledge stratum.

Applying AIP to the connection stratum of the Nth Stratum Architecture will solve the problems related with the security and mobility at the network layer. AIP will assure the user authentication based on the self certifying addresses and thereby protect from source spoofing, as routers will drop the packet if the source address is incorrect. Similarly it will also prevent denial of service attack, route hijacking and route forgery from the unauthenticated host. Moreover, AIP also provides the mobility support which is based on the globally unique host identifier (EID). Such a host that is based on the Nth Stratum Architecture and AIP can roam around the internet freely and has a unique public identifier. This is a unique feature not present in the present internet as well, which is based on NAT and public/private IP addresses.

The above proposed design aims at improving the quality of life for the world citizens by creating a family of dependable and interoperable networks providing direct and ubiquitous access to information. These future wireless and wireline networks will be designed to be readily adaptable to current and future needs, at an acceptable cost. Long term goal is to make the development of networks and networked applications faster and easier, leading to both more advanced

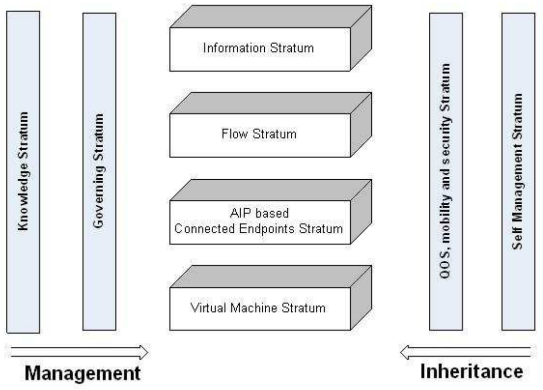

Figure 3: The Nth Stratum Architecture based on AIP

and more affordable communication services. Various innovative designs towards the future internet will allow new markets to appear, redefining business roles and creating new economic models. These goals can only be achieved by gathering a strong, industry-led consortium of the leading operators, vendors, SME, and research organisations, with the determination, skills and critical mass to create cross-industry consensus and to drive standardisation. Last but not the least, industry players will be motivated to move on further from the traditional TCP/IP approach to the innovative internet architectures, getting attracted by the high profit margins and reliable business models.

5 Limitations of the proposed Architectural Design

The proposed architectural design could not be deployed on the current router infrastructure, but surely is seen as a solution for the long term technology trends. Moreover, there also exist interoperability issues between the existing and the future communicating systems, based on the different architectures. The proposed design will also have several effects on routing such as forwarding and routing information bases (FIB and RIB) will increase in size. It will also result to the increase of diameter of the existing internet, due to the large domains being split into multiple ADs. Moreover, application of AIP will also include the CPU cost for the cryptographic operations, which is similar to those of adopting Secure BGP (S-BGP). Last but not the least, it can also be seen as: Building castles on the quicksand! as one needs to replace the existing TCP/IP based design with the proposed one.

6 Conclusion

This paper extends the idea of the Nth stratum concept by applying the AIP to the connection stratum. The proposed architectural framework for the future internet provides a holistic and systemic approach to development and design of network architecture for the future communication system. Moreover, the design is more scalable, interoperable, self-managed, secure, addressable, accountable and also support seamless mobility. The proposed design can be seen as a starting point for the further work of detailing the specification of this concept. It also has a potential to generate new economic opportunities for the industrial palyers with new classes of networked applications.

References

[1] D. G. Andersen, H. Balakrishnan, N. Feamster, T. Koponen, D. Moon, and S. Shenker. Accountable internet protocol (aip). *SIGCOMM Comput. Commun. Rev.*, 38(4):339–350, 2008.

[2] P. Ferguson and D. Senie. Network Ingress Filtering. RFC 2827, The Internet Engineering Task Force, May 2000. `http://www.ietf.org/rfc/rfc2827.txt`.

[3] M. Johnsson, J. Huusko, T. Frantti, F. Andersen, T. Nguyen, and M. Leon. Towards a New Architecture Framework - The Nth Stratum Concept Methods. Technical report, Mobi Media, July 2008.

[4] R. Moskowitz and P. Nikander. Host Identity Protocol Architecture. RFC 4423, The Internet Engineering Task Force, May 2006. `http://www.ietf.org/rfc/rfc4423.txt`.

[5] A. C. Snoeren and H. Balakrishnan. An end-to-end approach to host mobility. In *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 155–166, New York, NY, USA, 2000. ACM.

[6] M. Soellner. 4WARD Architecture and Design for the Future Internet. Technical requirements, Aug 2008. `http://www.4ward-project.eu/index.php?s=publications,Aug,08`.

Security in Friend-to-friend Networks

Nicolas Mahe
Helsinki University of Technology
nmahe@cc.hut.fi

Abstract

This paper offers a survey of the development of a new kind of network: Friend-to-friend network. This network can be defined as a P2P network, more precisely as a private P2P. Details of Freenet and security of F2F networks will be described.

KEYWORDS: Friend-to-friend, F2F, friends, nodes, security.

1 Introduction

Nowadays, peer-to-peer networks (P2P) are widely used for data sharing, VoIP and streaming. Thanks to a software dedicated to this kind of network, the connection to a P2P network is very easy. In 1999, Napster was the first widely used P2P-application. Napster was a centralized P2P because a central server was required to stock information where we could find data. The main problem with this structure was the same problem as a client-server network: there is a single point of failure. If the main server of Napster was out of service, all the network was out of service too. The centralized structure was not the best option for P2P networks. Other structures were created. The decentralized P2P structure is a network without any central server. In this case, each peer sends queries to other peers to find a data. A third structure of P2P consists of an hybrid P2P between centralized and decentralized P2P. The principle is that peers are connected to superpeers which are connected to each other.

Although P2P networks seem to be a wonderful world, this world presents some problems of security. The first problem with this type of network is that we can not be sure that the contents of data we download are good or bad. In fact, the data could have a valid name but is in reality a malware (computer viruses, worms, trojan horses, spywares) [8]. The second problem is that IP addresses are visible. Consequently, an attack to a peer is possible without too much difficulties. Finally, we can find an important quantity of private information because users do not use properly the software dedicated to P2P networks. We can easily imagine the consequences of sharing unexpected data. [4]

Since the beginning of this century, a new kind of network exists: Friend-to-friend network (F2F). According to Dan Bricklin [1], he introduced the term "friend-to-friend network" to describe this network. The principle of this network is that people share data only with friends, people whom they trust, and with friends of friends. This network is based on the proverb: "Friends of my friends are my friends".

The aim of this paper is to make an overview of the security of F2F networks. Section 2 will provide with backgrounds about P2P networks, security in P2P networks and finally F2F networks. In section 3, we will discuss about the security of F2F networks. Section 4 will be a presentation of a F2F network, Freenet. During this presentation of Freenet, we will try to understand why Freenet could be an interesting project for network security.

2 Backgrounds

First of all, we will provide some backgrounds in order to have a better understanding of this paper.

2.1 Security in networks

Security in networks is maybe as important as the functioning of the network. Thanks to the Web, we can collect information but also send emails, buy train or plane tickets and book a place in theatre. But all these activities via internet need to be securised. If the network is unsecured, we can have some problems when we pay for example. To secure transaction in the web, all information are generally encrypted in order to be unreadable if someone intercepts the information. With F2F networks, we will see that others solutions could be used to secure the network.

2.2 Concept of P2P networks

The main idea with P2P networks is that each peer is connected directly to other peers. But this idea is not typical to this kind of network because the idea of most protocols is that hosts need to be connected directly to each other to share data [4]. In the web, clients are connected directly to a server to download the contents. So, we can say that P2P network is only an extension of this idea. In F2F networks, we will see that this idea of connection between two hosts is still true but the sender and the requester will not be connected directly anymore. This concept will have an influence of the security of the network.

2.3 Security in P2P networks

Although P2P is an interesting network for data sharing, VoIP, streaming, it presents many security problems. The first problem we can detect with this network is related to the content of the data; it may be replaced by malware (computer viruses, worms, trojan horses and spyware). When we download a file, a common name of a file can hide a strong malware. We can find viruses and worms in email and some other programs but also in P2P networks. [4]

Another major inconvenience of P2P networks is that all IP addresses are visible. In fact, all peers can know from which peer a data comes from and to which peer a data is going to. Consequently, if we share a file and a user disagrees with our action, he could attack us very easily because he knows our identity. [3]

Finally, a third problem is concerning people who do not share their data correctly [4]. The reasons of this mistake are various because using a file-sharing system seems to be easy at first glance. But the system proposes many different options and it is easy to make confusion and to select a bad one or to select a wrong sharing folder. It is also frequent that people do not take time to tidy their data in the computer. Consequently, all data are in the same folder (for example 'My documents'). If they share the main folder, they share all their data. This is the same problem when a computer is used by many users of the family. If the parents are working on the computer during the week and the children use it during the week-end to download musics and videos, they will probably involuntary share their parents' data. Finally, some users think that sharing all their disk increases their popularity into the network and will provide them a better rate of downloadings. The consequence is that an important quantity of personal information is available in the network, like bank account numbers and passwords. According to a test realised by Eric Johnson, Dan McGuire, Nicholas D. Willey for their research [4], they spent some hours in a P2P network and found these results:

- Birth Certificate - 45 Results
- Passport - 42 Results
- Tax Return - 208 Results
- FAFSA (Free Application for Federal Student Aid) - 114 Results

With these information, we can imagine that people could take the identity of someone else.

2.4 Friend-to-friend networks

Friend-to-friend networks are one type of P2P networks. Figure 1 shows that F2F networks can be placed in the family tree of P2P networks. As we can see, we can divide network in two main parts: client-server and peer-to-peer. Client-server network is actually the main type of network. Generally, the Web is working with this type of network. In fact, when we are connected to a web site, we download and send some data but it is not a sharing of data. On the other hand, the idea of P2P network is the sharing of data. This type of network is divided into two parts: public and private P2P. The private part can also be divided into Group based and F2F. [7]

Public and private P2P have two different visions of P2P. The idea with public P2P is that we share our data with all the world. We only need to download a software dedicated for this application and then we can join the network and begin the sharing. On the other hand, private P2P corresponds to a group of a limited number of persons. The main advantage of private P2P is that people have a tendency to stay in the network after downloading what they wanted because it is easier to gain reputation of a small group than of a huge one. Moreover, if they want to keep the network alive, they know that they should stay in the network as long as possible [5].

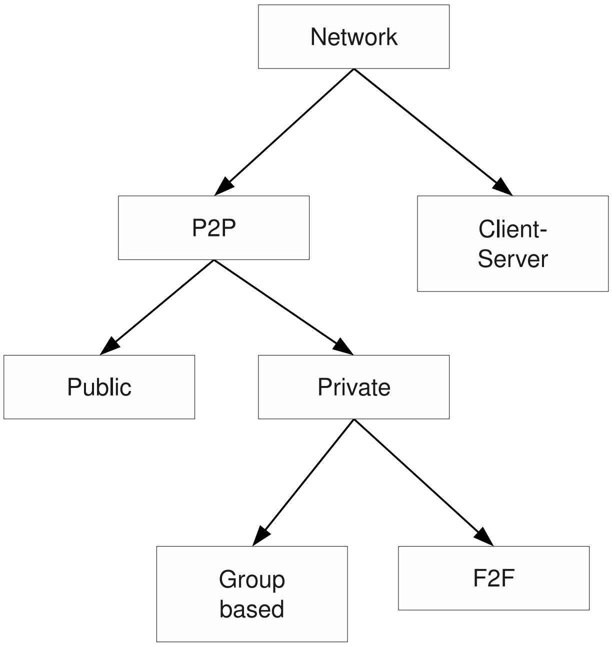

Figure 1: Friend-to-friend network in Peer-to-peer family tree [3]

We can now consider private P2P and distinguish Group based to F2F. Group based is made up by groups of users [7]. In each of these groups, users can share data together. One user can invite other people to join the group even when all users of the group do not know the new one. For this reason, we can say that Group based is not very private. A particularity of this network is that communication between groups is impossible. A user can belong to several groups but two users who belong to two different groups cannot communicate to each other, except if they both belong to a third group.

The structure of F2F is very different because we do not consider a group of users anymore but we now consider each user, also named a node. We are talking about nodes because the structure of F2F network is like a spider's web where all intersections correspond to a user, a node. In this network, each node is connected to many other nodes, his neighbours, which are the direct friends of the node [5]. Consequently, two nodes do not have the same friends. The data's sharing is quite simple. If a node searches a specific data, it sends a request to his direct friends. If one or several nodes have the requested data, the sharing can begin. Otherwise, the direct friends send the request to their direct friends. As we can see, a property of F2F is that we share data with our friends but also with the friends of our friends. With this system, a new parameter is added: HTL (Hops To Live). This parameter is important because it limits the number of nodes that the information can go through. Without this parameter, if a request is not resolved, the request can go through the network forever. Contrary to other types of P2P, there is no connection between the sender and the requester of the file (except if the two nodes are direct friends). The file will follow the same way as the request but in the other way. Consequently, all nodes which transmit the request could also receive the file and then send it to the next node. With some criterions, a node can decide to keep the file in order to provide the file quicker in the future. That is why a data is multiplied when many requests have been done on it. But after a delay, the file is deleted in order to let the place for another file. Finally, a particularity of F2F networks is that all nodes share data anonymously [6]. In fact, a node knows the IP addresses of its direct friends in order to be able to establish the communication but it does not know the IP addresses of its friends' friends. Each node transmits only files but no information about the identity of the other nodes.

Figure 2 shows us how communications work in a F2F network. [3]

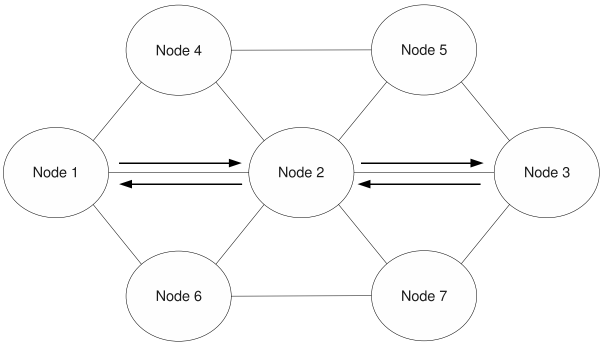

Figure 2: Communication in a F2F network

In we consider node 1, we can see that it has three different direct friends. When our node has a request, it sends it to his direct friends. If one or several friends have the information, they can begin the sharing. Otherwise, direct friends ask their direct friends if they have the requested information. For example, if node 3 has the information, it sends it to node 2, which then sends it to node 1. At the end, node 3 sends a file to node 1 through node 2. We can note that node 1 knows that file came from node 2 but it does not know that the file really came from node 3. Similarly, node 3 only knows that it sent the file to node 2. Finally, node 2 knows that the file came from node 3 and went to node 1 but does not know exactly where the file came from and where it went. With this system of data sharing, everyone can share files anonymously. [3]

Another characteristic of the F2F networks type is that data is encrypted during the transmission [3]. As we have seen before, data go through different computers before they reach the destination. So they are encrypted at the beginning of the transfer and decrypted at the end to protect intermediaries. In fact, some illegal data can circulate through our computer because of our neighbours. But because they are encrypted, we cannot read them and so we do not have any problem.

Finally, we can also add the fact that F2F networks are much more efficient than a traditional open network. The main reason is that nodes do not leave after downloading files; they stay in the network to allow it to work. In fact, in a classical P2P, nodes are generally leaving after downloading what they need because they do not want to waste bandwidth and be held liable for illegal downloadings. In F2F networks, nodes have real relationships to each other so they try to cooperate. [5]

3 Analysis about security of Friend-to-friend networks

Contrary to classical P2P networks, friend-to-friend networks present some security's advantages. These advantages concern the users but also the networks. If there is no security for the network, it could fall very quickly and be unusable.

3.1 Anonymity

As described previously, F2F networks respect the anonymity of the users. During the sharing, nobody is able to say where the information comes from (except the sender) and where it is going to (except the requester). So, they do not know the identity of the other nodes [2]. The only information they have is the IP addresses of their direct friends. This characteristic of F2F networks could be essential in some cases. In fact, because users are anonymous, the censorship is impossible. In most of developed country, people generally do not really need anonymously. However, in some countries like China, the inhabitants have no right of expression. Internet is controlled a lot and censured most of the time. If we say something against the regime, we could have a lot of problems and probably spend many years in jail. With a network where users are anonymous, the regime can not detect these users and can not punished them neither.

3.2 No malware

In a classical P2P network, some malwares can be introduced in the computer due to the network. In fact, a file could have a valid name but is in fact a walware (computer viruses, worms, trojan horses and spywares). In F2F networks, links between nodes are organised around real relationships. So everybody knows that file sharing is done between friends. In this case, each node tries to keep the network in good health and so avoid adding malware into the network. [4]

3.3 Inexperienced users

In the previous section, we have seen that inexperienced users can share unexpected data like bank account numbers, passwords or scan of identity card. The consequences for them could be very bad. In F2F networks, because the direct nodes are friends, if a node shares unexpected data, other nodes will tell it to delete the file. If the file is not download anymore, it will disappear of the network.

4 Case study: Freenet

Freenet can be classified as a F2F network. In this part, we will describe this network and explain some points about the security.

4.1 What is Freenet?

Mike Godwin, a Americain barrister said: "I worry about my child and the Internet all the time, even though she's too young to have logged on yet. Here's what I worry about. I worry that 10 or 15 years from now, she will come to me and say 'Daddy, where were you when they took freedom of the press away from the Internet?'" [3]

This sentence provided by mike Godwin explains the aim of Freenet in the network. The main idea of Freenet is to provide a freedom of speech through a peer-to-peer network with a strong protection of anonymity. But Freenet is not only a network for data sharing, like the other P2P networks, but also proposes some freesites, forums and wikis. Ian Clarke, the creator of Freenet and its coordinator, defines his network as "an Internet within an Internet" [3].

Freenet is a decentralized network where all users act anonymously. The decentralization of the network is very important in order to be resistant to any attack and acting anonymously is primordial if we really want a network of freedom. Ian clarke explains the reasons why anonymity is so important for this network. In the presentation of the philosophy of the network, he says that a major difference between animals and human beings is the ability of human beings to establish sophisticated communication and using abstract concepts. This communication is essential to improve our knowledges, which "improve our ability to survive and be successful" [3]. Under democratic governments, people are free of thinking and expression, which is not the case under non democratic governments. Figure 3 summarizes this idea of communication's freedom. The question we can ask now is to know whether dictatorship do not join the group of animals when we think of communication. In fact, communication under dictatorship is limited by the government as a basic communication.

The solution to avoid this communication's discrimination is to use some anonymous communication's tools. Freenet is one of them.

4.2 Freenet's working

In this part of the presentation of Freenet, we will see that the functioning of Freenet is original.

4.2.1 Darknet and opennet connections

There are two types of connections to the network of Freenet: darknet and opennet [3]. By default, darknet connection is used and is the most secured way. To use this type of connection, we have to indicate to the network our friends who will be our direct neighbours in the network. When connections between nodes have been created, these connections are fixed and there will be no change.

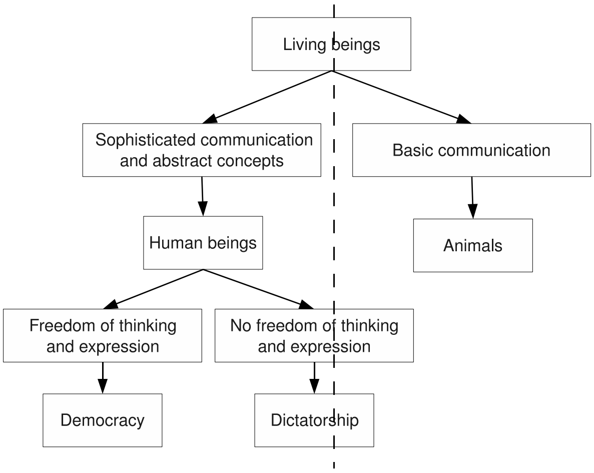
Living beings
Sophisticated communication and abstract concepts
Basic communication
Human beings
Animals
Freedom of thinking and expression
No freedom of thinking and expression
Democracy
Dictatorship

Figure 3: Communication, one difference between human beings and animals

Opennet connection is different to darknet connection for some reasons. First of all, we can use this kind of connection when we do not know anybody in the network but we all the same want to join it. In this configuration, we are placed in a random position into the network. We can note that the position does not depend on our location; our direct neighbours could be in America or Australia when we are in Europa. Because this place is probably not the most suitable one, we will switch our place with other nodes. In fact, Freenet is really efficient if nodes do not need a long time to get an information. If a node always makes requests to nodes in the other part of the network, we can easily understand that we can improve the communication. The best way to improve this communication is to switch the place of the nodes depending on their interests. According to the talk given by Ian Clarke and Oskar Sandberg in Berlin in 2005 [3], we can see that the nodes' switches improve considerably the efficiency of the network. In the part 'Sending a request', we will better understand how these switches could improve the efficiency of the network.

Finally, a mixed mode is also possible for users who do not know many people in Freenet. In this case, they have fixed neighbours but they also switch their places with other nodes. When these users have enough friends in Freenet, they can decide to disable the opennet connection to get more security.

4.2.2 Sending a request

Figure 4 shows us how a request searches a data through the network. [2]

Node A sends a request to his direct neighbours. In this case, we consider that he has only one neighbour, Node B (step 1). Because Node B does not have the requested file, he

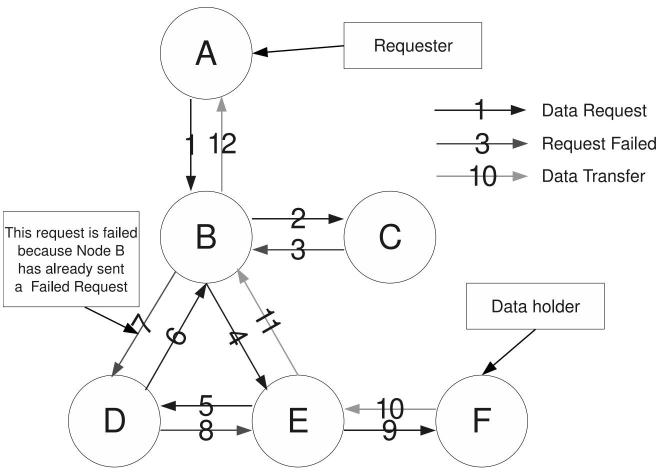

Figure 4: A typical request sequence

forwards the request to Node C (step 2) who answers that he does not have the requested file neither (Request Failed, step 3). So, Node B sends the request to Node E (step 4), which sends it to Node D (step 5). When Node D wants to send the request to Node B (step 6), Node B answers that he has already said that he has not the requested file. Consequently, the request goes back to Node D (step7) and Node E (step 8). Node E forwards the request to Node F (step 9) who replies with a positive answer; he sends the requested data (step 10). This data is transmited to Node B (step 11) before arriving to node A (step 12). As we can see, the request can go through the network during a long time before finding the requested node. The parameter HTL (Hops To Live) stops the request after a while if the request is not solved. [2]

As we can see, the request needed 9 steps before finding the right node. If we consider a real network with thousands nodes, we can imagine that a request needs more than 9 steps to find the right node. That is why nodes switch their places in order to resolve the requests quicker.

4.2.3 Data's encryption

During the transmissions, all data are encrypted. This encryption is very useful especially for intermediaries [3]. Because nobody is able to say if the contents of the data are legal or illegal, it is possible that illegal data move along the network. Moreover, intermediaries receive data before sending them. Sometimes, to improve the efficiency of the network, a node can decide to save the data in its disk because this data is often requested. To protect the intermediaries, this data should not be readable in order to avoid any legal problems.

4.3 Security point of view

The security in Freenet is a very important part of the project. In the presentation of the project in Berlin [3], Ian Clarke and Oskar Sandberg said that the project is not perfect in the sense that some secure problems could be find. But they also explain that it is important to propose a release of Freenet even if some parts of the project can be improved. In fact, if we do not propose any release, we will not have anything and the improvement would be impossible. Although this project is not perfect, it is certainly better that many tools we use every days.

4.3.1 Anonymity

The main goal of Freenet consists of a guarantee for all users to be anonymous in the network [2]. As we have seen above, anonymity can be primordial in some countries like China, where the rights of people are restricted. However, some people may want to know the identity of users. In theory, if a user uses a darknet connection, this identification is totally impossible. With an opennet connection, this identification is more or less possible but is very complicated. With some techniques, we can identify in which group of nodes a node belongs, but not really the identity of the node. At the present time, researches are done to improve the anonymity of people who use opennet connection.

4.3.2 Censorship

Censorship can also be a problem for the network but Freenet is resistant to this attack [3]. If we want to censor an information, we need to shut down all the nodes with this information. The best way to locate the nodes is to do a request and then to analyse all the answers. But, when we receive a data which comes through all the network, some nodes have probably saved it in their own disks. Consequently, not only we do not have censored the information, but we have also multiplied it in the network. Censorship is impossible in Freenet.

4.3.3 Harvesting

The aim of this attack is to destroy the network. But Freenet is a decentralized network, so without any central server. A solution is to get all IP addresses of nodes thanks to a modified version of a node. Then, we can get all IP addresses and we can attack all nodes easily. The solution proposed by Freenet is the darknet mode. With this mode, nodes are invisible in the network. [3]

4.3.4 Legal attacks

Finally, legal attacks can be considered [3]. First, we can imagine that the ports used by Freenet could be blocked. But Freenet use different ports for the communication. The solution could be to block all ports which means blocking all internet.

Another legal attack would be to block all well-known data as a Freenet data. However, since the last version of Freenet, all data are encrypted like other encrypted communication. Consequently, it is impossible for anyone to recognize a data which belongs to Freenet.

Finally, a government could decide to prohibit Freenet. In this case, only the darknet connection would be available.

5 Conclusion

For several years, peer-to-peer networks are used a lot but they often present some security's problems. This paper proposed a new kind of private peer-to-peer to solve this problem: friend-to-friend network. The idea with F2F networks is that people do not share with all the world anymore but only with a group of people, their friends. Then, if the request is not resolved, the request can be sent to the friends' friends. With this solution, we only share data with people we trust.

This paper presented also Freenet, which is a kind of friend-to-friend network. The main idea of Freenet is that all users of the network act anonymously. Thus, direct attacks again a node is not possible anymore.

We have seen that security in network is a very important part of network's science. Because we can do many different things in internet, including paying online, security is really primordial. To solve this problem of security, researchers create new techniques and also new networks. We can now wonder if all these techniques will be enough to protect private life of users and also their transaction thanks to the Web.

References

[1] D. Bricklin. Friend-to-friend networks. `http://www.bricklin.com/f2f.htm`.

[2] I. Clarke, O. S, O. Wiley, and T. W. Hong. Freenet: A distributed anonymous information storage and retrieval system. In *In Workshop on Design Issues in Anonymity and Unobservability*, 2000.

[3] F. community. The freenet project. `http://freenetproject.org/`.

[4] M. E. Johnson, D. Mcguire, and N. D. Willey. The security risks of peer-to-peer file sharing networks. `http://www.issa-la.org/`.

[5] J. Li and F. Dabek. F2f: reliable storage in open networks. In *In 5th International Workshop on Peer-to-Peer Systems*, February 2006.

[6] B. C. Popescu, B. Crispo, and A. S. Tanenbaum. Safe and private data sharing with turtle: Friends team-up and beat the system. In *In Proc. of the 12th Cambridge Intl. Workshop on Security Protocols*, 2004.

[7] M. Rogers and S. Bhatti. How to disappear completely: A survey of private peer-to-peer networks, 2007. `http://www.cs.ucl.ac.uk/`.

[8] M. Suvanto. Privacy in peer-to-peer networks. `www.tml.tkk.fi/Publications/C/18/suvanto.pdf`.

Device Compromise Detection and Revocation in Wireless Sensor Networks

Gurvinder Singh
Helsinki University of Technology
gsingh@cc.hut.fi

Abstract

This paper reviews the different techniques to detect compromised nodes in Wireless Sensor Networks (WSNs) and mechanisms to revoke compromised nodes keys. There are various approaches to detect compromised nodes in WSNs, one of the approach is based on attestation in which sensor node prove it's authenticity and integrity to base station or neighboring nodes. Attestation mechanism can be performed by either software attestation or hardware attestation. Each node in WSNs contains secret keys to secure the communication between base station and neighboring nodes. Key Revocation is a process in which keys related to compromised nodes is removed from trustworthy nodes, which causes compromised nodes to be removed from WSNs. Key Revocation methods are divided in two categories: Centralized Key Revocation scheme, in which base station is responsible for key revocation; Distributed Key Revocation, in which neighboring nodes decide the compromised nodes and revoke their keys.

KEYWORDS: Sensor Networks, Security, Compromise Detection, Key Revocation.

1 Introduction

Wireless sensor networks are assembled from a large number of interchangeable, low-cost, resource constrained devices and scattered into an area of interest to perform surveillance or monitoring tasks. Today wireless sensor networks are used in many security related applications like military operations, critical infrastructure protection e.g. nuclear power plants and burglary alarm systems. Wireless sensor devices are usually deployed to monitor physical or environmental conditions, such as temperature, sound, pressure or motion. Sensor devices are generally deployed in remote areas which provide easy access to an adversary. If an adversary is able to compromise sensor node in WSNs, he can reprogram sensor node to act on his/her behalf and able to attack on WSNs. For example, the adversary can cause the sensor node to send incorrect information to base station to hide some military activity or send false information about the location of certain troops or introduce some false warnings to raise alarms or may be denial of service.

To avoid such situations some mechanisms are needed, which can detect the target sensor node is still trustworthy or not. Attestation approach is one of the mechanism to detect compromised nodes, in which node proves it's trustworthiness to base station or neighboring nodes. Attestation technique is based on either Software attestation: *Software Code Update By Attestation (SCUBA)* [10] or Hardware attestation: *Period Broadcast Attestation Protocol (PBAP)* [3]. SCUBA is a software based attestation approach, which allow base station to detect and ,if possible, repair a compromised node through code updates. PBAP is a hardware based attestation approach, which uses Trusted Platform Module (TPM) [11] as trust anchor, which provide data sealing to specific platform configuration and cryptographic functionality. After compromised nodes are detected, we need to remove compromised nodes from wireless sensor network.

Key revocation is a mechanism to remove keys related to compromised nodes from WSNs. Key Revocation is done by using different procedures. The procedures are differentiated based on entity responsible to provide information regarding revocation process. *KeyRev* [13] is one of the scheme to revoke compromised nodes keys based on Centralized scheme. This scheme use key updating technique and make compromised nodes obsolete, as a result remove them from WSNs. *Reelection* [12] is another scheme to revoke compromised nodes keys, in which neighbour nodes decide that node is trustworthy or not using positive Voting. Sensor nodes form a club in which they broadcast 'Buddy List' which they trust. After having such lists from all club members, each node made the list of trusted nodes in club and vote according to that. As a result compromised node is not allowed to renew it's membership to group.

2 Background

Wireless sensor networks life cycle consist of four basic phases: *pre-deployment, initialization, operation* and *revocation*. In pre-deployment phase, WSN owner programs sensor nodes with secret information like keys, usually symmetric keys [2], and authentication signatures. This phase happens at owner premises and is considered to be as safe. Sensor nodes are then deployed and initialized. These sensor nodes try to establish keys with their neighbours, as a result set up a secure communication path between them.

In WSNs, nodes are mobile and process of setting up keys and path with neighbours is carry on throughout life cycle. An adversary can attack any node or set of nodes and try to compromise them during this duration. At any stage, one or more nodes can find another node/nodes misbehaving, as a result they may prompt a revocation process to remove all the credential related to compromised node/nodes from WSNs. While a compromised node can communicate with any other node in WSN and may have access to all other compromised nodes keys too, which make them possible to impersonate if they wish. Compromised nodes have not been running authorized software and thus do not necessarily follow protocols, to identify misbehavior, to revoke other compromised nodes, to vote honestly or delete keys shared with revoked nodes. To handle these problem in WSNs, we need reliable mechanisms which can detect compromised nodes out of WSNs and capable of removing credentials related to compromised nodes to make WSN trustworthy.

There are various types of threats in wireless sensor network. Some of the important threats are listed below.

- An adversary may capture nodes, analyze and replicate them, and surreptitiously insert these replicas at strategic locations within the network. Such attack is called as *Replication Attack* [7] and may have severe consequences. These compromised nodes may allow the adversary to corrupt network data or even disconnect significant parts of the network.

- *Denial-of-Message (DoM)* [5] attack deprives sensor nodes from receiving broadcasted massages. In Wireless sensor network, broadcast protocols assume a trustworthy environment. However, in safety and mission-critical sensor networks this assumption may not hold, as some sensor nodes might be adversarial.

- *Sybil Attack* [6] is particularly harmful attack against sensor and ad hoc networks, wherein a node illegitimately claims multiple identities. Such an attack can be exceedingly detrimental to many important functions of these networks, such as routing, resource allocation, misbehavior detection, etc.

3 Detection Mechanisms

In this section we will study mechanisms to detect compromised nodes in WSNs using different Attestation techniques.

3.1 Software Based Attestation

Software Code Update By Attestation (SCUBA) [10] protocol is a mechanism to attest sensor node based on software attestation, this enables the design of a sensor network to detect compromised nodes without false negatives, and either repair them with code updates or blacklist them.

3.1.1 Assumptions

Wireless sensor network consists of one or more base stations and several sensor nodes. Every sensor node and base station has a unique ID, referred as *node ID* and *base station ID*. The communication can be single-hop or multiple-hop between base station and sensor node. To authenticate messages between sensor node and base station we assume public key infrastructure is set up, in which sensor node knows authentic key of base station. But when adversary compromised sensor node, he can learn about key. In this case we need to set up new a key with base station for secure communication without relying upon pre-existing key. The *untampered code execution* mechanism, provided by Indisputable Code Execution (ICE) [9], is used for this work. Read Only Memory (ROM) is used to store *node ID* and *base station public key*, as attacker can't tamper the contents of ROM.

```
B->A:       (ICE Challenge)
B:          T1 = Current time
A:          Compute ICE checksum over memory region
            containing the ROM, the ICE verification
            function and the SCUBA executable
A->B:       (ICE checksum)
B:          T2 = Current time
            Verify (T2 - T1 ) <= Time allowed to
            compute ICE checksum
            Verify ICE checksum from node by
            recomputing it
A->B:       Hash of code memory
B:          Use hash from node to determine
            if node's code memory is modified
            Prepare code patches for sensor node
B->A:       Code patches
A:          Apply patches
```

Figure 1: SCUBA protocol between base station B and sensor node A.

3.1.2 SCUBA Protocol

The purpose of SCUBA protocol is to provide a method for base station to detect and ,if possible, repair a compromised node through code updates. The compromised node could contain malicious code, which can interfere with code update process. The protocol assumes that update can be performed in the presence of malicious code. For example, if base station sends a code patch, malicious code on node may try to fake the patch installation. The base station obtains a firm guarantee for code update procedure. It guarantees that either code update is successful or malicious code running on sensor node, which is preventing the application of code update. ICE is used to obtains this guarantee as it enables base station to get an *indisputable* guarantee that SCUBA protocol executable on sensor node (untrusted computing platform) will execute untampered.

Figure 1 shows a simplified version of SCUBA protocol. The base station invokes ICE verification function on the sensor node by sending a challenge. The ICE verification function computes checksum over memory region containing itself, SCUBA protocol executable and ROM containing base station's public key and sensor node's ID. If base station receives correct checksum from the sensor node within expected time, base station obtains guarantee that SCUBA protocol executable on sensor node will execute untampered. In this case base station repair node via code update. If received checksum is incorrect or it takes longer time than expected than base station presumes that malicious code on sensor node is interferring with code update process.

After computing and sending checksum to base station, ICE verification function in SCUBA protocol invokes hash calculation function, which sends hash of sensor node's memory contents to base station. Base station compares received hash with correct value of hash it has stored to determine weather node is compromised or not. If there is any difference between two hash values, base station can also ask for hash from specific memory location to pinpoint where changes are made by an adversary. Thus base station can send code updates only for those memory regions which are modified and thereby decrease the amount of data to be sent and increase overall performance.

Base station and sensor node need to authenticate the packets they receive. Authentication procedure uses hash chains to authenticate the packets. Base station generates hash chains and signs first element of hash chain with it's private key and sends this signed element to sensor nodes. Sensor nodes can verify it using public key of base station it has stored in ROM. To authenticate a sensor node, we use the fact that only node with correct memory layout will be able to generate correct ICE checksum within the expected time. As memory contents also contain *node ID*, so node being verified is able to generate checksum with in expected time. After computing checksum, sensor node sends first element of it's hash chain and Message Authentication Code (MAC) of this element to base station. MAC is computed using ICE checksum as a key. Base station independently generates ICE checksum and can verify MAC sent by sensor node. If the MAC of the hash chain member sent by the node verifies correctly at base station and are received within the expected time, the base station is guaranteed that hash chain element came from the correct node.

The SCUBA protocol described above deals with sensor node which are one hop away from base station. To attest node which are multi hop away, we need to calculate the network latency time with precision to be able to attest sensor node. In SCUBA protocol, *Expanded Ring* [10] method is used to calculate network latency. The method uses intuition that if the checksum computation time of a node is always measured by a neighboring node, then the network latency is always a single hop. To achieve this, base station first calculates ICE checksum computation time which are single hop away from it. Then base station ask these sensor nodes to calculate network latency of neighboring nodes which are single hop away from them. In this way base station calculate network latency with high precision for whole sensor network.

3.1.3 Security Analysis

- **Pre-computation and replay attack** An adversary can try to calculate checksum over memory region containing ICE verification function and target executable, before making any changes to memory contents and send this pre-computed checksum to verifier when asked for it. To prevent this attack SCUBA sends a random challenge to sensor node and node calculate checksum based on this challenge and thus preventing this attack.
- **Proxy Attack** An adversary may try to forward challenge to node, which is more resource rich and have complete memory contents of compromised nodes before making any modification. Therefore, remote node is able to compute checksum faster and send result back to base station. As resource rich node will do computation faster which provide time to forward challenge request to remote node and time to send reply back. But in SCUBA we assume adversary is not physically present and all nodes are having same resources and if adversary try to forward request to another network, then request must go through base station as wireless sensor network have small range communication capability. Base station block any request going to another network and thus SCUBA prevent proxy attack.

3.2 Hardware Based Attestation

Periodic Broadcast Attestation Protocol (PBAP) [3] is an hardware based attestation approach for hybrid WSNs using Trusted Platform Module (TPM) [11] as trust anchor for attestation protocol. TPM provides assurance of delievered attestation values. TPM also offers cryptographic functions which provide the foundation for attesting local platform using remote platform.

As WSNs are large scale and nodes are of low-cost and resource constrained, it is not feasible to have TPM in all the sensor nodes. Fortunately, many WSNs are organized in clusters and each cluster has it's own cluster head (CH) and cluster nodes (CN), such WSNs are known as hybrid WSNs. CH usually does some special task like data aggregation or key management for a number of CNs. Therefore, CHs are a valuable target for adversary, so it is logical to equip CHs with TPM. CNs should be able to verify weather CH is still trustworthy, even after multiple hops away. PBAP is an efficient protocol which allows CNs to validate the trustworthiness of a CH at regular intervals.

3.2.1 Assumptions

Cluster nodes are limited in their storage, computational, communication and energy resources. However, they have enough space to store key information and are able to perform basic operation like computing hash functions, symmetric encryption etc, but they are not able to do public key encryption. TPM is integrated in CH and is used to protect keys and other security related data. To subvert a CH, an adversary must reprogram and reboot sensor node to access security related data.

3.2.2 Period Broadcast Attestation Protocol (PBAP)

PBAP allows CNs to verify trustworthiness of CHs using TPM as trust anchor. TPM offers a concept called *sealing*, which allows a data block to be bound to a specific platform configuration. A sealed massage is created by selecting a range of platform configuration registers, a non-migration key and data block which should be sealed. TPM is then able to decrypt and transfer sealed data block, only if its current platform configuration matches the platform configuration, since when the sealing was executed. Sealing provides assurance that protected message will be recoverable only when system is in known state. PBAP enables only CNs to verify platform configuration of CHs. To verify trustworthiness of CNs, CH has to perform additional mechanism like redundancy checks or voting schemes.

PBAP uses Guy Fawkes-style [8] hash chains for authentication and extends it to enable attestation in hybrid WSNs. The protocol is divided into two phases. In the *initialization* phase CHs and CNs are preconfigured before deployment. In *attestation* phase CHs periodically broadcast an attestation message. This phase normally last for whole life cycle of CHs.

Initialization: Each CH$_i$ is preconfigured with non-migratable public key pair (e$_{CH_i}$,d$_{CH_i}$) and a hash chain C^{CH_i}. The seed value c$_0^{CH_i}$ of hash chain is generated on CH$_i$ using TPM's random number generator. CH$_i$ is assumed to have only one valid configuration, denoted as P$_{CH_i}$:= (PCR$_0$,...., PCR$_p$), where PCRs are registers of TPM. After booting, measurement regarding each component is performed and value is stored in registers. Each value of hash chain C^{CH_i} is sealed to this platform configuration P$_{CH_i}$:{c$_0^{CH_i}$}$^{e_{CH_i}}_{P_{CH_i}}$,......,{c$_n^{CH_i}$}$^{e_{CH_i}}_{P_{CH_i}}$ = Seal(P$_{CH_i}$,e$_{CH_i}$,c$_0^{CH_i}$),......,Seal(P$_{CH_i}$,e$_{CH_i}$,c$_n^{CH_i}$). Each CN$_j$ which interact with CH$_i$ is preconfigured with the last value c$_n^{CH_i}$ of its hash chain. After deployment, CN$_j$ can only keep values corresponding its CH and another certain number of CHs in nearby region to save memory.

Attestation: CH$_i$ and associated CNs (denoted as CN$_*$) are loosely time synchronized. Time is divided into intervals I$_\lambda$, λ= 1,...,n. At the beginning of each interval, CH$_i$ broadcasts attestation message consisting of values of hash chain in reverse order of generation with identifier I$_\lambda$ to CNs. If platform configuration has not been modified by adversary, CH$_i$ will be able to unseal hash values. In the first interval I$_1$, CH$_i$ unseals the hash value c$_{n-1}^{CH_i}$ and transmits it togeather with interval identifier. In the second interval c$_{n-2}^{CH_i}$ is unsealed and transmitted and so on. CN$_*$ check if the interval I$_1$ stated within received message with local I$_1'$ within certain error range. If they match, CN$_*$ verifies whether h(c$_{n-1}^{CH_i}$) = c$_n^{CH_i}$. If equation holds, CH$_i$ is considered trustworthy and value c$_n^{CH_i}$ is overwritten with c$_{n-1}^{CH_i}$. In the next interval CH$_i$ releases c$_{n-2}^{CH_i}$ and so on, which are similarly checked.

Due to unreliable communication, a CN could miss some messages. Thus, CNs should not immediately declare CH being untrustworthy but wait for a certain threshold of time. If a CN receive messages again, it can be resynchronized by applying hash function multiple times.

3.2.3 Security Analysis

- To compromise CH and forge trustworthy platform configuration, an adversary need to access hash chains. Therefore, he has to either access key used to seal hash chains or perform unseal command under compromised state. TPM acts as smartcard and offers high security for protected data against unauthorized access, which makes very difficult for adversary to access keys.Additionally, access to hash chain is only possible if platform configuration is not modified.

- An adversary may attack runtime caused by buffer overflow to access stored hash values. PBAP approach can not handle runtime attack caused by buffer overflow, because platform configuration is reported during initialization phase. But as the attack would result in modified system state, adversary can not be able to access hash chains.

- Replay attack, where an adversary first blocks the forwarding of legitimate hash values to collect them, then compromises a CH and finally releases these hash values. But this is not possible, because hash values are only valid for a specific interval, which is validated by each CN.

- PBAP is performed in cleartext and an adversary can distinguish between attestation and data messages. Therefore adversary can perform a selective forwarding attack by forwarding attestation messages, but blocking data messages. Such attacks are a general problem in WSNs and show that the PBAP is not resistant against all attacks in a multihop scenario.

4 Key Revocation

Key management mechanism is required to establish keys between sensor nodes or base station which exchange data between each other. It includes two aspects: Key distribution and Key revocation. Key distribution refers to the task of distributing secret key to provide communication secrecy and authenticity.Key distribution is often done in *pre-deployment* phase, which occur at WSNs owner premises and is considered to be safe. Key revocation refers to the task of securely removing keys which are known to be compromised.

In this section different key revocation mechanisms are discussed based on entity responsible for providing information regarding key revocation.

4.1 KeyRev: Centralized Key Revocation Mechanism

KeyRev is a centralized key revocation scheme, in which central authority (*base station*) is responsible to revoke compromised nodes from wireless sensor networks. Unlike most proposed key revocation schemes focusing on removing compromised keys, KeyRev scheme uses key updating techniques to obsolesce the key owned by compromised sensor nodes and thus remove them from network.

The basic key distribution scheme establishes two kinds of keys among sensor nodes: *pairwise keys* and *path keys*. Pairwise key is established among sensor nodes which shares the secret keys. Path key is assigned between sensor nodes which are within wireless communication range but do not share a key. Instead of using pairwise keys and path keys directly for confidentiality and authenticity, KeyRev uses *encryption key K_{encr}* and *message authentication code (MAC) key K_{mac}*. The encryption key and MAC key are generated by pseudo random generator, which is bound to pairwise key or path key and *session key* distributed regularly by base station. When the session key is updated, the encryption key and the MAC key are also changed. A sensor node always uses latest encryption key and MAC key to encrypt and sign the outgoing messages or decrypt and to verify the incoming messages. If there is a session key distribution scheme in which the revoked sensors cannot recover the new session key when they are revoked, these revoked sensors will be removed from the network because they cannot derive the new encryption keys and the MAC keys in the next session. Although an adversary may retain the pairwise keys and the path keys, the adversary cannot figure out the encryption keys and the MAC keys because of the pseudo-random function is used. Thus, key revocation problem is reduced to the session key update problem.

4.1.1 KeyRev scheme

The lifetime of WSN is divided into intervals called sessions. The duration of session can be dynamic or static depending on application. Base station broadcast session key K_j in the beginning of j-th session. Each sensor node has unique sensor ID *i*, where i $\in$ {1,.....,n} and n is largest ID. Each sensor node maintains a *node revocation list (NRL)*, which includes all sensor node IDs which have been revoked. NRL is checked for each incoming and outgoing messages to ensure only valid nodes are member of network.The encryption key and MAC key are bounded to session key K_j and path key K_{AB} between sensor node A and B as follows:

K_{encr} = F(MAC(K_{AB},K_j),1)
K_{mac} = F(MAC(K_{AB},K_j),2)

Where F(K,x) is pseudo random number function and x is an integer 1 or 2 for generating K_{encr} or K_{mac} respectively. Any message that A sends to B is encrypted by K_{encr} and signed by K_{mac}. Any message received by B from A, B always first verifies the message and then decrypt it. A sensor node always uses K_{encr} and K_{mac} corresponding to current session key K_j.

For any message transmitted in the network, encryption and authentication are done as:

$A \longrightarrow B : \{M|T_s\}_{K_{encr}}, MAC(K_{mac},\{M|T_s\}_{K_{encr}})$

Where M is message, T_s is the timestamp when sending message and MAC(K,R) denotes the computation of message authentication code of message R with key K.

To revoke compromised nodes from sensor network, a mechanism is needed to stop them from receiving session key and thus preventing them from driving K_{encr} and K_{mac} bounded to new session key. Thus compromised node will not be able to decrypt and authenticate themselves. To distribute session key in such a manner a session key distribution key is described in [13]. The session key distribution scheme must satisfy following criteria:

1. The compromised sensors should not be able to obtain new session keys.
2. The sensor node is time synchronized so that the current session keys can be identified.

Criterion 2 is easily satisfied with the help of timestamps. For criterion 1, session key scheme is derived in [13] based on personal key share distribution scheme in [4]. In session key scheme mentioned in [13], base station broadcasts list of compromised nodes in start of each time interval to all sensor nodes and nodes can update their node revocation list (NRL) by adding them in NRL.

4.1.2 Security Analysis

The KeyRev scheme satisfies following security properties:

- Session key distribution process is secure as to restore session key, some secret information is required, which is pre-distributed among sensor nodes. Adversary can not recover session key without this secret information. Session key distributed scheme [13] prevent compromised nodes from recovering new session keys.
- As KeyRev works on key update mechanism, this causes path key and pairwise key to be compromised by adversary. But adversary can not figure out K_{encr} and K_{mac}, if the session is updated. An adversary can attack with chosen plaintext to recover session key, but attack is time consuming and in meantime session key will be updated in next time interval.
- To prevent Revocation attack, in which adversary can impersonate as base station and start revocation of trustworthy nodes, KeyRev depends upon broadcast authentication schemes such as μTESLA [1]. μTESLA provides authentication to messages broadcasted by base station. Therefore, KeyRev is secure against revocation attack until base station is secured.

4.2 Reelection: Distributed key Revocation Mechanism

Reelection is a distributed key revocation mechanism, which requires majority of positive votes from its neighbors for approval at regular intervals. In this protocol, a node on joining the WSN and periodically must demonstarate that it is still trustworthy to be part of the network. Revocation becomes preventing a bad node from renewing its membership.

In Reelection, nodes form a club and each member broadcast *Buddy List* of neighbors it trusts. After receiving such lists from all neighbors, node cross-reference received lists and determine whether enough nodes have approved their buddies or not. If they approved then node continue to interact with buddies during next time period. Reelection mechanisms provide support for diverse trust strategies as compared to simple voting mechanism.

4.2.1 Lightweight Reelection with Buddy Lists

In some applications, diverse strategies are needed: risk-averse nodes might revoke neighbor as soon as one of its other neighbours had done so, while more relaxed nodes might continue to do business with any node that was still supported by two of its neighbors. In some application, one might want a diverse population of risk-averse and risk-loving nodes, so that network performed well in normal times but still performed acceptably under serious attack.

Reelection mechanism with buddy lists is in general enough to support diverse trust strategies. In this decision regarding revocation is done by neighboring nodes, after cross-referencing received buddy lists from neighbor buddies. If enough nodes approve their buddies then they continue to interact. Here definition of 'enough' is made independently of protocol mechanism.

Approved buddy lists are authenticated using Guy Fawkes-style [8] hash chains. After deployment, node A distributes a key authentication value $\mathrm{K}_{A,0}$ = $\mathrm{h}^{(T)}$(seed, A) to its neighbors. Buddy lists are signed with a session authentication key $\mathrm{K}_{A,i}$ = $\mathrm{h}^{(T-i)}$(seed, A) during time period i and $\mathrm{K}_{A,i}$ is not revealed until the start of time period i+1. Here is the protocol:

1. A $\longrightarrow \star$: k_{i-1} , $\mathrm{access}_{A,i}$(buddies)= $\langle$A, i, buddies, HMAC_{K_i} (A, i, buddies)$\rangle$
2. $\star$: Verify $\mathrm{access}_{A,i-1}$(buddies), delete offending neighbor's keys

Each node A broadcasts a list of approved neighbors $\mathrm{access}_{A,i}$(buddies), where buddies is a set of approved node identifiers. Notably, no pre-assigned storage or topological information is required, yet buddy lists work even under scenario where adversary present from start of deployment of sensor nodes.

5 Conclusion

In this survey paper, we try to inspect the security issues and different mechanisms to handle these issues in the wireless sensor networks, which may be a main disturbance to the operation of it. Due to the low resources and deployment in unattended and hostile areas, the wireless sensor networks are much more prone to all kind of security risks, such as information disclosure, intrusion, or even denial of service. As a result, the security needs in the wireless sensor networks are much higher than those in the traditional wired and adhoc networks.

Wireless sensor networks enable us to monitor environment conditions and surveillance remotely and efficiently. However, with the convenience that WSNs have brought to us, there are also increasing security threats for WSNs, which need to gain enough attention.

During this survey it is noted that adding security to resource constrained sensor devices in wireless sensor network with minimum overhead provide significant challenges and is an ongoing area of research.

References

[1] Adrian Perrig, Robert Szewczyk, J. D. Tygar, Victor Wen, David E. Culler . SPINS: security protocols for sensor networks. *Wireless Networks, vol. 8, no. 5*, pages 521-534, September 2002.

[2] Chan, H., Perrig, A., Song, D.X. Random key predistribution schemes for sensor networks. *IEEE Symposium on Security and Privacy (S & P)*, IEEE Computer Society Press:197 - 213, 2003.

[3] Christop Krauß , Frederic Stumpf, and Claudia Eckert. Detecting Node Compromise in Hybrid Wireless Sensor Networks Using Attestation Techniques. *Fourth European Workshop on Security and Privacy in Ad Hoc and Sensor Networks (ESAS)*, LNCS 4572/2007:203-217, 2007.

[4] Donggang Liu, Peng Ning, Kun Sun . Efficient Self-Healing Group Key Distribution with Revocation Capability. *Proceeding of the 10th ACM Conference on Computer and Communications Security (CCS '03)*, pages 231 - 240, 2003.

[5] McCune, Jonathan M., Elaine Shi, Adrian Perrig, Michael K. Reiter. . Detection of Denial-of-Message Attacks on Sensor Network Broadcasts. *IEEE Symposium on Security and Privacy*, pages 64 - 78, 2005.

[6] Newsome, J., Shi, E., Song, D., Perrig, A. . The Sybil attack in sensor networks: Analysis & Defenses. *IEEE Symposium on Information Processing in Sensor Networks*, pages 259 - 268, 2004.

[7] Parno, B. Perrig, A. Gligor, V. . Distributed detection of node replication attacks in sensor networks. *IEEE Symposium on Security and Privacy*, pages 49 - 63, 2005.

[8] Ross Anderson, Francesco Bergadano, Bruno Crispo, Jong-Hyeon Lee, Charalampos Manifavas, Roger Needham. A new family of authentication protocols. *SIGOPS Oper. Syst. Rev.*, pages 9 - 20, 1998.

[9] Seshadri, A., Luk, M., Perrig, A., van Doorn, L., Khosla, P. Using FIRE and ICE for detecting and recovering compromised nodes in sensor networks. *Technical Report CMU-CS-04-187, Carnegie Mellon University*, 2004.

[10] Seshadri, A., Luk, M., Perrig, A., Van Doorn, L., Khosla, P. SCUBA: Secure Code Update By Attestation in Sensor Networks. *WiSe'06 :Proceedings of 5th ACM workshop on Wireless security*, ACM Press, 2006.

[11] Trusted Computing Group:. Trusted Platform Module (TPM) specifications. Technical report, 2006. `https://www.trustedcomputinggroup.org/specs/TPM/`.

[12] Tyler Moore, Jolyon Clulow, Shishir Nagaraja, and Ross Anderson. New Strategies for Revocation in Ad-Hoc Networks. *Fourth European Workshop on Security and Privacy in Ad Hoc and Sensor Networks (ESAS)*, LNCS 4572/2007:232-246, 2007.

[13] Yong Wang, Ramamurthy, B., Xukai Zou. KeyRev: An Efficient Key Revocation Scheme for Wireless Sensor Networks. *ICC '07: IEEE International Communication Conference*, 2007.

OAuth and OpenID 2.0

Pauli Kaila
Helsinki University of Technology
Pauli.Kaila@iki.fi

Abstract

The internet is full of services and for most of these services a user has to have an account to be able to use them. This leads to the fact that a user usually has one account per service. The OpenID protocol promises to give a solution to this problem of users having multiple accounts to access the services they are using. A user can create an account for an OpenID provider service and then they can use that account on all sites that provide the support for authentication using OpenIDs.

Decision to begin using OpenID has to be made by the user, even though it is possible for services to only allow logging in using OpenIDs. The fact that the user has to know what an OpenID is and how to use it slows down the adoption of the OpenID protocol. Even now there are millions of potential OpenID users who have gotten their OpenIDs automatically from some service they are using, but they are not aware of their OpenIDs or the fact that what they could do with them. So in order for the OpenID to get traction the users need to be educated about the possibility of using OpenID for authentication and what services can be accessed using OpenID.

A user might also want to give a service access to data owned by the user and hosted on some other service without giving the service requiring the access their credentials to the service hosting their data. One possible solution for this problem is the OAuth protocol, which provides a way for the user to grant access to their data hosted on a service. Using OAuth is almost transparent to the users, i.e., the user does not need to know that they are using OAuth. Although to get the benefits from having OAuth support the users need to be educated to know what to trust.

KEYWORDS: OAuth, OpenID, authentication, authorization

1 Introduction

People using the internet today have access to and are using a wide array of different kinds of services. Some of those services are such, that a user might want to access them, but does not trust them enough to give them access to some of their private data like their e-mail address or even their credentials. On the other hand the number of services is so large that it would be useful to be able to use the same user identification for at least most of the services so that the user does not have to remember so many credentials. OpenID 2.0 [2] promises to solve these problems. It gives the user an OpenID identifier, which the user can use on every site that supports OpenID authentication (act as OpenID Relying Parties). OpenID 2.0 does all this in a way that does not give the site where the user is logging in access to the users credentials for their OpenID provider.

There are also services for which a user might like to give limited access to some data stored in another service they are using. For example a photo printing service for which a user wants to give temporary access to their photo storing service. OAuth [1] is a solution for this problem of services needing to access data stored on other services. The protocol describes how controlled access to a data hosted on a service can be given to another service without the user having to give their credentials to that another service needing the access.

Section 2 describes on a fairly high level how the OAuth protocol works. Section 2.1 discusses OAuth from a user point of view and in section 2.2 we discuss some security concerns of OAuth. Then in section 3 we briefly describe the OpenID 2.0 protocol. Followed by end user oriented discussion over OpenID in section 3.1 and with some security concerns in section 3.2. And finally in section 4 we conclude this study.

2 OAuth

OAuth [1] is a protocol which can be used by the users of some service (Service Provider) to allow another service (Consumer) to access the the Users data (Protected Resources) stored by the Service Provider without giving their credentials to the Consumer.

An example use case would be a photo printing service. The photo printing service would be the Consumer. The user has stored photos on a service in the internet. This photo storing service would be the Service Provider and the photos stored in that service would be the Protected Resources. Now if the Consumer and Service Provider know of each other and support OAuth the User can give the Consumer access to the Protected Resources without having to give her credentials for the Service Provider to the Consumer.

OAuth authentication is described in figure 1 and it has the following steps.

1. The Consumer requests a Request Token from the Service Provider.

2. The Service Provider grants the key to the Consumer.

3. The Consumer directs the User to the Service Provider.

4. The Service Provider obtains the Users authorization for the Request Token.

5. The Service Provider directs the User to the Consumer.

6. The Consumer requests to exchange the authorized Request Token to an Access Token.

7. The Service Provider grants the Consumer the Access Token.

8. The Consumer can now use the Access Token to access the Protected Resources.

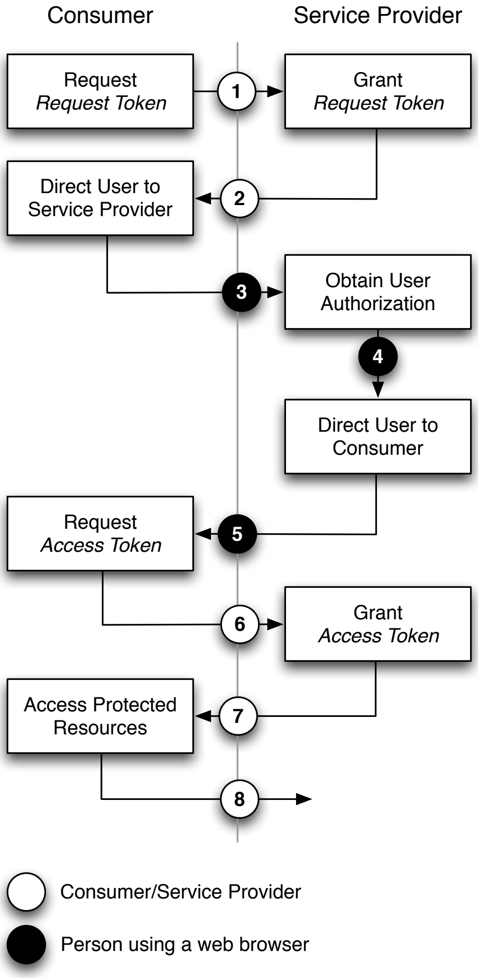

Figure 1: OAuth authentication [1]

There are some very well known services and companies that act as OAuth Service Providers like Google[1], YouTube[2] (owned by Google) and MySpace[3] to name a few. Google also provides an OAuth Playground[4] for trying out OAuth step by step.

2.1 OAuth from end user viewpoint

From the end user viewpoint OAuth is almost transparent. When using the service provided by the Consumer, the user will just be redirected to the Service Providers site and then redirected back after having successfully authenticated themselves for the Service Provider and agreeing to giving access for the Consumer. The user does not have to know anything special.

For an ordinary user who is not so privacy concerned the benefits of using OAuth might be left if the dark. If a user has not cared up to this point what service they give their credentials as long as that service seems trustworthy, why would they change their behavior now. Of course more educated users might demand implementation of OAuth from the services they are using in order to protect their credentials. But the ordinary users who are not so technically inclined or concerned about their privacy seem to trust almost anything in the internet. These ordinary users may have never heard of OAuth, they may have never thought that something like OAuth could be needed or that something like that could exist. These ordinary users happily give their Service Provider credentials to the Consumers without ever thinking that they could be able to use these Consumer services without giving them their credentials. So basically what OAuth needs in order to gain traction is for the Service Providers to start educating their users to use OAuth and demand OAuth implementations from the potential Consumers.

2.2 OAuth security considerations

For the users of OAuth as for the users of almost any other security or privacy measure, the biggest threat are the users themselves. If a user does not care where they give their credentials or if the user does not know what to watch for then they might be fooled to think that they are using OAuth even if they really are not. As OAuth heavily relies on the redirects between the sites, users might grow accustomed to the fact that they are redirected during a login process. A malicious party might use this to their advantage by pretending that they are using OAuth and redirect the user to a page that looks like a Service Providers page and in that way be able to snatch the users credentials. It has been suggested that OAuth could also be used so, that the user manually enters the address of the Service Provider in their browser and then also manually enter the Request Token on the Service Providers site. This is supposed to be a more secure way for the user, but it also makes the user experience a lot worse and needs that the users are educated to always do things like that (never ever believe where you are redirected to).

In a more elaborate scheme DNS spoofing, i.e., fooling the user to another site by sending fake DNS information for them, might be used to spoof the Service Providers address and then fool the users to give their credentials to the fake Service Provider when they are trying to use the Consumer. Using transport layer security on both the Consumer and Service Provider and also between them should help against this kind of attacks.

Even though the OAuth specification defines some ways to sign the OAuth request they are not the only ways to do it. So if for example the current signing methods for OAuth

[1] http://www.google.com
[2] http://www.youtube.com
[3] http://www.myspace.com
[4] http://googlecodesamples.com/oauth_playground/

(HMAC-SHA1 and RSA-SHA1) are proven to be not safe enough, then the signing methods can be changed without changing the actual protocol itself. There is also a possibility to use plaintext signing, but in that case the transport between the Service Provider and the Consumer definitely needs to be secured.

The OAuth specification has an appendix which addresses security concerns about the protocol and gives guidance on how to use the protocol securely. The specification for example suggest using transport layer security between the Consumer and Service Provider in order to protect the tokens and other data during the authentication. However since the answer to most of the concerns listed in the specification is employing transport layer security they could have made it a mandatory requirement for the protocol.

3 OpenID 2.0

OpenID [2] provides a way for users to identify them selves for services on the internet (Relying Parties) with their OpenID Identifier without the User having to give their user credentials or other private information for the Relying Party. The User can choose which OpenID Provider to choose and can switch the Provider if they so desire preserving their OpenID Identifier. OpenID is decentralized so there is no party that controls the whole scheme and there is no central authority who could or should approve Relying Parties or Providers.

OpenID 2.0 authentication is described in figure 2 and it has the following steps.

1. The End User initiates the authentication by presenting a User-Supplied Identifier

2. The Relying Party normalizes the User-Supplied Identifier and performs discovery to find out the OpenID Provider used by the End User. At this point the Relying Party could establish an association with the OpenID Provider by using Diffie-Hellman Key Exchange. The OpenID Provider could then use this association to sign all the subsequent messages.

3. The Relying Party directs the End User to the discovered OpenID Provider together with an OpenID authentication request.

4. The OpenID Provider authenticates the user.

5. The OpenID Provider directs the End User back to the Relying Party with the details on whether the authentication succeeded.

6. The Relying Party verifies the information received from the OpenID Provider. The information can either be verified based on the signature if the Relying Party and OpenID Provider have established an association, or the verification can be done by sending a direct request to the OpenID Provider.

7. The user is now authenticated and can continue using the services provided by the Relying Party.

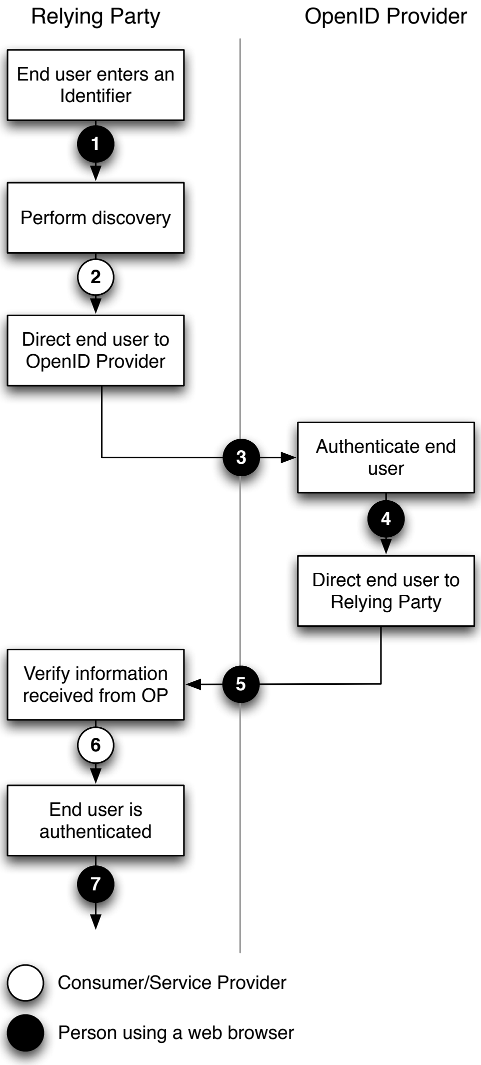
Relying Party
OpenID Provider
End user enters an Identifier
1
Perform discovery
2
Direct end user to OpenID Provider
3
Authenticate end user
4
Direct end user to Relying Party
5
Verify information received from OP
6
End user is authenticated
7
Consumer/Service Provider
Person using a web browser

Figure 2: OpenID 2.0 authentication

OpenID foundation has some very well known important corporate board members like Yahoo!, Microsoft, Google, Verisign and IBM[5]. Some of those board members like Yahoo! and AOL also provide OpenIDs for all of their users[6] making the potential user base for OpenID fairly large.

3.1 OpenID 2.0 from end user viewpoint

In contrast to OAuth's somewhat transparent behavior from the end users viewpoint, OpenID needs the user to know that they have an OpenID account, what it is, and how to use it. So even though OpenID has millions of potential users (after Yahoo! started as an OpenID Provider in January 2008, the number of OpenID accounts was 368 million[3]) most of those users probably do not even know that they have an OpenID account let alone having ever heard of OpenID. Some user confusion might also stem from the fact that not all OpenID Providers are OpenID Relying Parties. So for example even if Yahoo! is an OpenID Provider one can not

[5] http://openid.net/foundation/

[6] http://openid.net/get/

login to Yahoo! using some other OpenID Provider. Saying that Yahoo! supports OpenID does not tell the whole truth. OpenID can not really get traction until more big popular sites also act as Relying Parties.

For the user it might seem odd, at least in the beginning, that they need to type in addresses to websites or at least similar looking strings to be able to log in to sites acting as OpenID Relying Parties. But, as they get used to it, the odd feeling is at least supposed to go away and be replaced with a better user experience including a kind of semi single sign-on. This semi single sign-on comes from the fact that if the user has authenticated to the OpenID Provider they do not need to re-authenticate when they move to another Relying Party. They just need to tell that other Relying Party their OpenID Provider and they are supposedly good to go. From an ordinary users point of view the fact that they really only need one account is the most important selling point of OpenID. Only then comes the fact that the user only needs to trust their credentials to one site.

There is however one problem in trusting one OpenID Provider to control login to all the sites a user are using. If the users OpenID Provider is down, they can not access any of the services they have been using with their OpenID account from that Provider. This is one of the biggest problems from the end user viewpoint, although they might not even think about it. But if OpenID gains more use and one of the big OpenID Providers goes down, then users will realize how fragile the system really is. Even though the promoters of the protocol are saying that OpenID is decentralized and the user is not dependent on any one Provider and can change their Provider if they so choose, changing their Provider on the fly if the original Provider is down might just prove to be impossible.

Yahoo! and Google have both done user experience and usability research on OpenID [4][5]. Both of the research results share the same view that the concept of OpenID, while intriguing, is not something that is clear to the normal users. Users do not know the OpenID brand, they know the brand of their OpenID Provider. So in order for the users to understand that they can use their OpenID to login to a OpenID Relying Party they need to be able to see the brand of their OpenID Provider on the site of the Relying Party. Even though users are not limited to using the OpenID Providers which brands they see, the fact that users do not know what OpenID is makes OpenID basically much less flexible to the Relying Parties, since in order to properly use a new OpenID Provider a Relying Party needs to re-design their login page.

The research by Yahoo! and Google also point out the fact that when presented with the brand of their OpenID Providers, the users happily type in their Provider username and password in to the login screen of the Relying Party. This provides even more user confusion as the users are not logged in and are possibly shown an error telling about invalid login. This also shows how easily users are willing to give out their credentials to other sites without ever thinking twice. Both research articles are discussing and describing ways on how to improve the OpenID user experience but there still seems to be a long way to go, before users are educated away from their old habits of just giving their username and password everywhere.

3.2 OpenID 2.0 security considerations

Having one account to access every service sounds nice as long as the user stays in control of that account. If someone manages to steal their credentials they are using for the OpenID Provider, and do that without the user knowing it, then that someone can do almost anything in the users name, as long as the user does not get a whiff of it. Someone having stolen a users credentials might even act as them on sites they have never visited making noticing the identity theft even harder.

For OpenID it is also important that the user is educated on the potential dangers of misusing their credentials. The OpenID Providers need to invest in educating their users and stressing out that they should never give their credentials to any other party than their own OpenID Provider. The education also needs to have information on how the user can be sure that they are indeed accessing the service provided by their own Provider.

As is the case with OAuth also OpenID relies heavily on redirects between the sites of the OpenID Providers and the Relying Parties. This teaches the users to take it as a normal behavior in the login phase so a user might not be wary of the dangers when they are redirected to a untrustworthy site which looks to be trustworthy.

The OpenID protocol allows for anything or anyone to be an OpenID Provider. This leads to the fact that an OpenID Relying Party must actually choose whether or not they can trust on an OpenID Provider. Since OpenID protocol does not specify how a user is authenticated, creation of a Provider where no authentication is happening at all is possible. So this in essence makes OpenID less open than the specification says, as the OpenID Relying Parties will constrain the allowed set of OpenID Providers to some well know parties.

As is the case with OAuth, OpenID is also susceptible to DNS spoofing attacks. If someone is able to make a user think that they are giving their credentials to their own Provider then that someone may be able to steal the users OpenID. So using a OpenID Provider with transport layer security and proper certificates is a must in order to shield the users from the threats of the internet.

The OpenID 2.0 protocol specification discusses these same security concerns and also some others and does not provide any other answers to these concerns than user education and transport layer security between the Provider and the Relying Party. These two key facts with some other good guidance provided by the specification should make the protocol secure enough for most uses. However as is the case with OAuth, if transport layer security answers to almost every security concern, why not make it a mandatory part of the protocol.

4 Conclusion

Both OAuth and OpenID, while similar in action, promise to provide solutions to different problems in handling a users identity or identities in the internet.

The main selling point for OpenID for ordinary users seems to be that the user only need to remember a limited set of credentials to access the sites that do support OpenID authentication. There are additional selling points for more technically inclined or more privacy oriented people, like the ability to choose their authentication provider (or even be one themselves) and the fact that the user does not need to supply their credentials to any other party than the one they have decided to trust. For people who are more technically inclined the security of OpenID is fairly good, but if a user does not know what to watch for, they might be giving their credentials to spoofers and then the spoofers have even more sites to act on as the user. This means that users of OpenID need to be educated on how their OpenID should be used and how it should not be used. Also OpenID has the problem of having a single point of failure, which is, from the users point of view, the users OpenID Provider.

For OAuth however there does not seem to be such a strong selling point among the ordinary users. For some reason the ordinary user seems to trust the services they are using in the internet even though they should know not to do so. But with proper education to the users and how they should protect their credentials the adoption and approval of OAuth in the internet is probably going to rise. Again the people who are more interested in their privacy are more eager to adopt and request the services they use to use technologies like OAuth. Although here OAuth differs from OpenID since a user can be using OAuth even though they do not know anything about it, whereas OpenID needs that the user understands the concept.

Both of the protocols have currently some big players backing them, but only time will tell whether OAuth and OpenID will become mainstream protocols used by the majority of the internet or whether they will remain as niche protocols for some technically inclined users. It seems that the key issues that will make or brake these protocols are making them easy enough to use and educating the users about the benefits of using these protocols.

References

[1] *OAuth Core 1.0*, December 2007. http://oauth.net/core/1.0/.

[2] *OpenID Authentication 2.0 - Final*, December 2007. http://openid.net/specs/openid-authentication-2_0.html.

[3] Yahoo! announces support for openid; users able to access multiple internet sites with their yahoo! id. Yahoo! Press Release, January 2008. http://yhoo.client.shareholder.com/PRESS/releasedetail.cfm?ReleaseID=287698, quoted on November 18th 2008.

[4] E. Sachs. Usability research on federated login, October 2008. http://sites.google.com/site/oauthgoog/UXFedLogin, quoted on November 18th 2008.

[5] A. Tom. Yahoo! releases openid research. Yahoo! Developer Network Blog, October 2008. http://developer.yahoo.net/blog/archives/2008/10/open_id_research.html, quoted on November 18th 2008.

Lightweight IP: A mobility-aware secured L3 protocol

Sumanta Saha
Helsinki University of Technology
sumanta.saha@tkk.fi

Abstract

Despite of the elegance and success of current Internet, failure to scale and security vulnerabilities made re-engineering of the Internet one of the main focus of the research community today. Extensive research in the field of building trust and accountability in the future Internet is going on and resulted in various secured network layer protocols such as HIP (Host Identity Protocol), AIP (Accountable Internet Protocol) and PLA (Packet Level Authentication). However, most of the security aspects in these protocols are heavy in resource consumption. With the advent of more and more mobile devices capable of networking, its of utmost importance to keep in mind the interoperability of resource constraint devices while designing such protocols. Although future Internet has received much effort, enabling resource constraint devices in it did not receive a fair share. In the future era of all-IP networking, Internet tablets, and smart phones the use of a protocol specially designed for resource constraint devices would be inevitable. This paper will be concentrating on designing such a network layer protocol named Lightweight IP or LIP for resource constraint devices keeping the current trend of identity based protocols as a base.

KEYWORDS: LIP, lightweight Internet protocol, Resource constraint device, Trust, Accountability, Mobility

1 Introduction

As elegant and simple as it is, the Internet architecture has been a huge success and served the community with extreme efficiency. But from security point of view, the architecture is riddled with much vulnerability. The extremely simple core of the Internet does not have any security feature built into it, which allows a radical amount of attacks engineered each day with increasing amount of risk. And the attackers are confident on attacking the network due to the fact that with current architecture, it's never easy to locate an attacker based on their IP address. Originating untraceable spam is as easy as using tricks such as onion routing [23, 19] and route hijacking [18].

On the other hand, the Internet was not built for the mobile age of today either. When the Internet was engineered, most of the terminals were fixed to an infrastructure and it was highly unlikely that a host would be mobile. However, throughout the last decade, mobile computing and nomadic networking have been becoming more common. People are using handheld devices such as mobile phones, PDAs and laptops for connecting to the Internet using various types of access networks. In between the connected periods, these devices often experience disconnected state. These intermittently connected devices pose a new level of challenge to the traditional Internet architecture. The Internet as of today is not capable of supporting intermittently connected mobile devices without losing precious resource. Most of the protocols used in the Internet, such as IP [16] and TCP [17], do not perform well in presence of intermittently connected devices or moving hosts.

These problems are widely discussed in the Internet community and many solutions have been proposed [21, 5] to prevent the threats and to enable the current network to support mobile devices. But almost all of them are ad-hoc solutions trying to address only one kind of problem at a time. Another problem of these intermediate solutions is that they tend to destroy the basic argument of the Internet - the famous "end to end argument" - which dictates that the network should be simple at the core and the intelligence should be at the edges [4]. And to make it even worse, most of the new protocols today are being developed by not taking into consideration the resource constraint devices which are becoming more and more common these days.

To address the security vulnerabilities of the Internet and to make it capable of handling nomadic networks gracefully, what is needed is to re-engineer the underlying technology so that security features can be built upon it easily and mobile devices can roam around the network easily. This paper presents a novel network layer protocol named LIP (Lightweight Internet Protocol) based upon the current trend of research community such as HIP (Host Identity Protocol) [14], AIP (Accountable Internet Protocol) [2] and PLA (Packet Level Authentication) [13] while keeping in mind the requirement of supporting mobile devices and their low processing power. The proposal uses two experimental features of TCP namely TCP User Timeout Option [7] and TCP Retransmission Trigger [8] to achieve better performance in intermittently connected network [20]. For creating accountability, we use AIP-style cryptographically generated addresses. However, to make the scheme suitable for resource constraint devices, we propose a modified version of cryptographic operation based on ECC (Elliptic Curve Cryptography) which is lightweight and easy to compute in a constrained environment.

Section 2 analyzes the requirement of such a protocol and then discusses about the architectural decisions while section 3 describes in more detail the security feature built into LIP. Section 4 casts light on the mobility features of LIP. Our overall approach to the protocol architecture and its applicability is discussed in section 5. At last, section 6 contains the concluding remarks.

2 Lightweight Internet Protocol

To address the inherent security problems in the current Internet architecture, we propose a new network layer protocol which derives its characteristics from the current trend in protocol stack design - namely the "trust to trust" design. Furthermore, envisioning the ubiquitous pervasive computing in future world using small devices with weak processors, we also strive to make the protocol as lightweight as possible without sacrificing much of the security requirements.

2.1 Requirement Analysis

The requirement analysis of LIP is greatly influenced by the trust-to-trust paradigm of network architecture. Furthermore, in every step of the analysis, usability for battery-driven wireless devices were kept in mind.

Intrinsic Security: The protocol should provide intrinsic security features. The end points should be accountable for the packets they are generating and the data packets should be resistant to forgery. Any misbehaving host should be detected quickly. Any intermediate node should be able to check data packets for authenticity similar to PLA [13]. The security model should not be similar to IPSec [12] where only the end points can check the packets for integrity and authenticity. This requirement is necessary to get rid of malicious packets as early as possible saving the bandwidth of the network.

Lightweight Cryptography: The cryptographic operations required for the protocol should be lightweight but secure enough. The requirement of lightweight crypto comes from the fact that the protocol is designed by keeping in mind small-scale mobile devices. In addition to that, the crypto functions should be fast enough to keep up with the wire speed.

Mobility: Host mobility is becoming more and more inevitable in this age of handheld resource constraint devices. Users are no longer attached to a fixed terminal but they use many battery powered smart devices which can connect to network. It is desirable that any modern protocol supports user mobility while keeping the connection alive.

Performance in Intermittent Connectivity: Resource constraint devices tend to be mobile and tend to hop from one access network to another. The connectivity also tends to be disrupted sometimes as they move between access networks. Current protocol stack is unable to retain the connection alive while the devices change their network layer address and experience a disconnected phase intermittently. The new protocol should have the provision to support this kind of environment.

Payload Security: As an optional feature, the protocol should also support security of the payload using suitable encryption method available, depending on the processing power of the device. This feature is made optional because many lightweight devices do not have the resources to spend for encrypting the whole payload.

2.2 Architectural Overview

LIP is designed to support a network architecture depicted in Fig. 1. The hosts would be mobile or fixed, the connection to the network may be disrupted any time and the network should be secure and accountable. The host can be battery-powered, low processor device. To perform well under these extreme conditions the following design decisions has been made.

2.2.1 Addressing

LIP borrows the addressing style from IPv6 [6, 1] and AIP [2]. The address of hosts consists of two parts, namely, domain and host (Fig. 2). But unlike current IP addresses, the host part is always connected to a host rather than a location. So, when a host moves the host part of the address remains same. The host part of the address is constructed from two sources. First one is the public key of the host and the second part comes from the interface physical address as is done in IPv6. This allows the address to be self certifying as well as make it possible for each host to be multi-homed. The public key portion of the address is allocated to each host by a TTP (Trusted Third Party), which we will discuss more on a later section. A more extensive description of this approach of address generation, which is referred to as CGA (Cryptographically generated address), can be found at [3].

2.2.2 Position in a protocol stack

LIP positions itself under the transport layer protocols in the OSI protocol stack as it is a network layer protocol. It uses standard IP header fields along with some extra features to allow it provide the requirements necessary for a secured network. If used with an IPv6 stack, it is also possible to add the extra features of LIP as an extension of the IP header (Fig. 3). A simple change in IPv6 addressing structure would enable it to be used along with LIP features. Rather than extracting the host part of the IPv6 address from the physical address of the interface, cryptographically generated public key of the host assigned by a TTP is used with LIP. In that case, it would be possible to implement LIP over any functional IPv6 environment. But for other scenarios, such as IPv4 environment, the network layer of the protocol needs to be changed to LIP implementation.

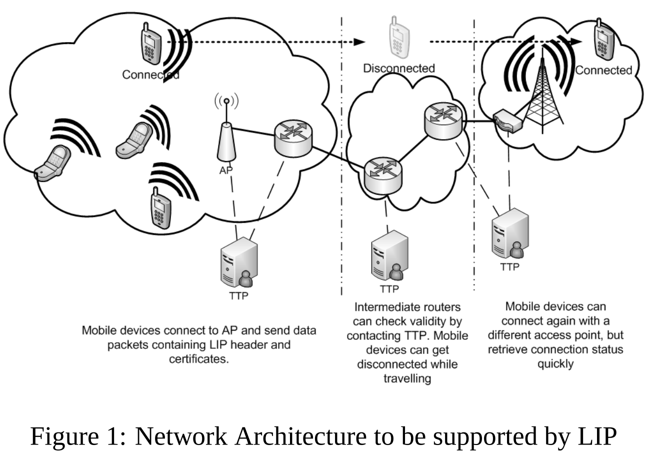
Connected
Disconnected
Connected
AP
TTP
TTP
TTP
Mobile devices connect to AP and send data packets containing LIP header and certificates.
Intermediate routers can check validity by contacting TTP. Mobile devices can get disconnected while travelling
Mobile devices can connect again with a different access point, but retrieve connection status quickly

Figure 1: Network Architecture to be supported by LIP

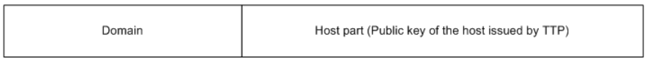

Figure 2: LIP Address structure

2.2.3 Header Fields

Almost all the fields of an IP header are retained as mandatory fields on LIP header. Along with those, some additional fields similar to PLA [13] is added. LIP uses a verification model similar to PLA to remove the per-packet verification process and various security vulnerabilities of an infrastructure-less protocol such as AIP. In this way, the protocol would require much less bandwidth than AIP while keeping the ability to verify the authenticity of packets at any point of time using the certificate inside the header fields. Additional fields in a LIP header are TTP certificate which is used for checking the authenticity of a packet by contacting the TTP, timestamp to ensure timeliness of the packets and restrict duplicity, sequence number to hinder replay attack, and a signature with the sender's private key for integrity. More detailed description of these fields can be found in [13].

2.2.4 Mobility

The provision for mobility is one of the basic requirements for any modern network layer protocols today. LIP incorporates this requirement by using a specially designed addressing structure which allows hosts to be mobile but keep the same host address. Other intricacies of mobility adopted in LIP is described in section 4.1.

2.2.5 Provision for intermittently connected device

As discussed earlier, current TCP/IP implementation is not suitable for intermittently connected devices due to the following reasons:

1. Changing domain causes network layer address to change
2. After certain period of disconnection, TCP decides a connected to be stale and aborts that [22]
3. TCP waits for retransmission timer to go off before starting to retransmit to a reconnected host [15] which sometimes causes loosing precious connected period

LIP uses two experimental features of TCP named TCP User Timeout [7] and TCP Retransmission Trigger [8] to get around these problems. More on the solution can be found in section 4.2.

3 Security Features of LIP

The main goal of trust to trust paradigm of LIP is to make the network secure and trustworthy, which leads to a whole section dedicated to LIP's security features. One of the prominent design criteria of LIP is also to make it lightweight. LIP tries to create a fine balance between highly secured and easily computable cryptographic methods. The cryptographic operations carried on in different phases of LIP operation are stated below along with the recommended methods for doing so.

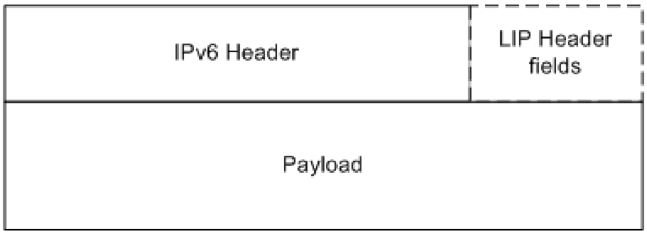

Figure 3: LIP as an extension of IPv6 Header

3.1 Cryptographic Operation and Address Generation

As stated before, LIP addresses are cryptographically generated addresses. When a new device is bootstrapped in an LIP environment, it first generates a public/private key pair for its own use. The public key is used as its own host identification and the domain part of the address comes from the nearest router or gateway router. As the length of the public key can be variable, so optionally LIP hosts can use a hash of the public key to make it a fixed width address. Alternatively, the smaller devices can just generate fixed length public key to avoid hashing. A bit in the header field can be used to indicate whether the address has been hashed or not. Rest of this section discusses about the cryptographical choices that had been made for LIP.

Public key cryptography is one of the most exquisite innovations of the last decade. It allows establishing a secure channel without sharing any key between the parties. But the downside of this kind of cryptography is heavy computation. They depend on operations which are not reversible or at least very hard to reverse. Operations used in public key cryptography includes discrete logarithm or factoring large primes. Among public key cryptographic algorithms, ECC (Elliptic Curve Cryptography) showed the most promise in the recent years due to the use of very short keys for high security level. It is also the most attractive cryptographic operation for lightweight devices because of its smaller operand length and relatively lower computational requirements. In LIP, we propose to use ECC as a cryptographic algorithm for signing and verification of the packets.

3.1.1 ECC Primer

We will give a very short primer to ECC which may be required to understand the following sections. For further clarification of ECC many extensive resources are available such as [11].

An elliptic curve over a finite field GF, i.e., a Galois field of order q, is composed of a finite group of points $(x_i,\ y_i)$ which satisfies the long Weierstrass form:

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \qquad (1)$$

where the coefficients a_i are elements in *GF(q)*. The equa-

Table 1: ECC scalar point multiplication performance [9]

Field Multiplier	Combinational Logic Blocks	Point Multiplier	Time(s)
Software Multiplier		Binary	6.039
		NonAdjacent Form	5.031
163 X 163 Multiplier	245	Binary	0.264
		NonAdjacent Form	0.224
163 X 163 4 digits	498	Binary	0.183
		NonAdjacent Form	0.115

tion 1 can be shortened to the following due to the fact that most of the time the q in $GF(q)$ is prime:

$$y^2 = x^3 + ax^2 + b \qquad (2)$$

where $a, b \,\epsilon\, GF(q)$.

Elliptic curve operations are closed under arithmetic operations. Details of operations can be found in Elliptic curve literature. The operation of ECC relies on the fact that given points P and Q in group, it is hard to find a number k such that $Q = kP$.

Elliptic curve operations are used to generate and verify digital signatures. The operations for generation and verification are given below.

If John wants to send a signed packet m to Jane, they must first agree upon the curve parameters with base point P. The field is GF(P) and the order of P is q. And John should have the ECC public and private key pair precomputed. Lets say the pair is (x, Q) where the public key $Q = xP$. The signature generation algorithm is as follows:

1. Select a random integer k from $[1, q-1]$
2. Calculate $r = x_1$(mod q), where $(x_1, y_1) = kP$. If $r = 0$, go back to step 1
3. Calculate $s = k^{-1}$(HASH(m) $+ xr$)(mod q), where Hash is an one way cryptographic hash function, such as SHA-1. If s is 0 then go back to step 1
4. The signature pair is (r, s)

To verify the packet m sent by John, Jane has to perform the following verification algorithm:

1. Verify that r and s are in the range $[1, q-1]$. If not, the signature is not valid
2. Calculate $w = s^{-1}$ mod q
3. Calculate $u_1 = ew$ mod q, where e = HASH(m) and $u_2 = rw$ mod q
4. Calculate $(x_2, y_2) = u_1P + u_2Q$
5. The signature is valid if $r = x_2$ (mod q), invalid otherwise

3.1.2 ECC implementation parameters for LIP

For lightweight devices, we assume that medium term security measures such as 163 bit ECC is enough because of the relative ease of the computation process. If for some reason, greater security level is required, application level security or IPSec is always available to use. As suggested in [9], we decided to use binary field arithmetic rather than a prime field because of the advantage of carry free arithmetic and also simplified squaring structure. Eisenbarth et al.[9] proposes a design of ECC for lightweight devices, which suggests three approaches to implement a lightweight ECC algorithm. First one is using a dedicated hardware coprocessor for doing cryptographic operations which would be too much expensive for mobile devices. Second option is to use a fully software implementation of ECC which is a really cheap solution and does not require any extra processor, but has the downside of being slower than the hardware version in order of magnitudes and being processor-hungry. The third option is a combination of software and hardware where the most expensive operations of ECC are done in a dedicated hardware module and other operations are done in software. This option seems to be the most reasonable one for LIP where a little amount of extra hardware can accelerate the whole cryptographic operation while keeping the most of it in software modules and thus minimizing the cost.

The core operation of ECC is multiplication $k \times P$, where k is an integer and P is a point on an elliptic curve. And according to the software realization of ECC done by the authors of [9], the software realization of GF(2^m) multiplication is the most expensive operation of all as depicted in table 1. Consequently, it is logical to adopt this approach of implementing the GF(2^{163}) multiplier in hardware while keeping all the other operation of ECC such as addition, inversion and reduction in software. For curve multiplication, Jacobian projective coordinates are used to avoid frequent use of inversion which is expensive in software. For the hardware implementation of multiplier, there are alternatives where we can use a bit serialized multiplier where only one bit of multiplicand is used in each iteration, or use digit-serial multiplier where multiple bits are multiplied at each iteration. We have adopted the second option as it produces much better performance (table 1) without increasing the area requirement much.

As shown in table 1, huge performance gain is possible using a small amount of hardware in the range of 250 to 500 extra combinational logic blocks, which also saves a huge amount of processing power and thus reducing battery power consumption. Even if we use the same cryptographic verification process used in PLA without any optimization, which uses three point multiplications per signature verification, it is possible to verify around three signatures per second as the hardware can perform around nine multiplications in a second.

3.2 TTP Certificate

After generating public key pair for using as address, next step would be to retrieve a TTP certificate which will allow the routers in the path of data packets to verify the authenticity of the packets. Use of fixed infrastructure such as the TTP can be sometimes unrealistic, especially when the network is highly mobile and the devices don't have any chance to contact any TTP. To avoid this scenario, it is also possible to use self certified addresses. Verification of certificates can be done at any intermediate routers by contacting the TTP and presenting the certificate. In case of unavailability of any TTP, the nodes can still continue to perform and wait for any time frame when a TTP is available for any of the nodes of the network, and then others can also verify their certificates using P2P refresh. The functionality of P2P refresh is as follows:

P2PRefresh: "At any point of time, in any place along the route of a packet, it can be verified by any router without contacting the source or destination" - to achieve this philosophy, each packet contains a certificate from a TTP which certifies the authenticity of the signature of the sender on the packet. For verifying the certificate the intermediate nodes must contact the TTP. But, in mobile environment it can be a problematic scenario because in many cases the network may be an ad-hoc network and there may not be any connection to any infrastructure node or the Internet. To overcome this limitation, we propose a form of P2P refresh where the peers are refreshed from one another if any of them has a connection to the infrastructure node.

Until the nodes have access to the TTP or the Internet, they function normally without verifying the certificate. If any security-critical packet arrives without verifiable certificate, the destination node can decide to drop it or use it by keeping track of the certificate for future reference. Whenever any of the nodes of the ad-hoc network gets in range of the Internet or an infrastructure node, it starts validating its own stored certificates and also sends signal to other nodes inside the ad-hoc network announcing its connectivity to the outside world. All the other nodes, as a response, start to refresh their collection of certificate verification through the announcer as shown in Fig. 4. Thus, any node storing certificates for security critical data can now verify the certificate and can optionally rollback the operations that it has done with the packets for which the certificate cannot be verified.

3.3 Accountability

Creating accountability in a network refers to the non-repudiation of hosts. Once a host sends a packet to the network, it is always possible to verify that the packet is actually sent by that very sender and none else. And from the sender's perspective it should be undeniable. In LIP, this feature of security is built into the protocol by the use of cryptographically generated addresses and TTP certificates as discussed earlier.

3.4 Payload Integrity Check

The integrity of the payload can be checked by verifying the signature on the packet. The sender signs the packets using their private key and the public key is the sender's address itself. So the verifiers just have to extract the public key from the address and then verify the signature on the packet to know whether the payload is changed or not.

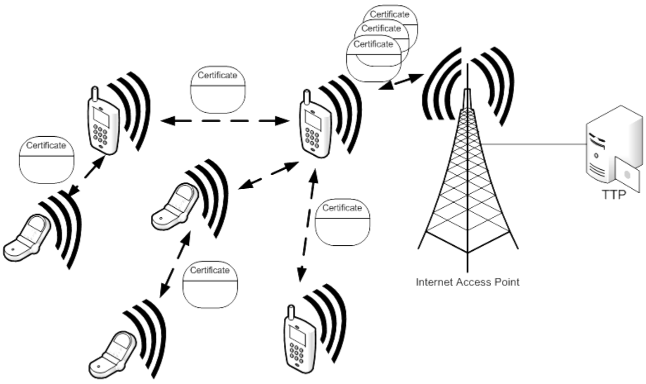

Figure 4: Peer to Peer refresh for certificate authentication

4 Mobility Features of LIP

One of the main reasons why resource constraint devices are resource constraint is mostly because they are mobile and it is not possible to attach then to a power source all the time, which leads to a battery-driven device. Consequently, it is inevitable that any network layer protocol designed for resource-constraint devices must have mobility management built into it. In this section of the paper, the mobility features of LIP will be discussed in more detail.

4.1 Mobility

Mobility in LIP is handled similar to HIP [14, 10]. Two types of mobility was taken into account while designing LIP, namely, macro and micro-mobility. The reason behind differentiating two types of mobility is the necessity for signalling and hand-off. In case of macro-mobility the mobile host travels away from one domain to another and so a drastic step needs to be taken to keep the connection alive while in micro-mobility the host merely changes its access point within same domain which can be handled much more easily.

4.1.1 Macro mobility

When a user moves from one administrative domain to another, even in LIP environment, its address changes due to the domain part of the address. The mobile host needs to initiate a process to let the connected peers know about its new address so that the ongoing connections can continue. It can be done using a handshaking process. The mobile host sends a update message to the corresponding host containing his new address, in return the corresponding host sends a echo request message to the new address to verify whether the host is actually available at that address. In return, the mobile host sends a third update message containing the echo reply. The update messages from the mobile host also con-

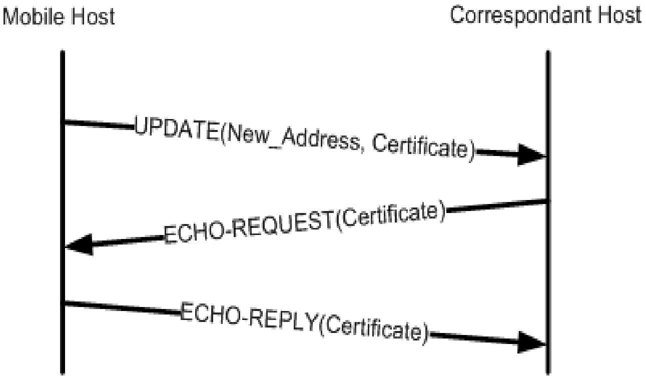

Figure 5: Three way handshake for mobile address change

tain the certificate of new TTP for the routers along the path to verify. Fig. 5 depicts the whole process of update.

4.1.2 Micro mobility

In case of micro mobility where mobile hosts are moving within a single domain, the signaling described above can be quite expensive. As we are dealing with lightweight devices which are expected to be mobile all the time, we cannot depend on the above process of mobility-handling always because it takes too much bandwidth and may not be always suitable for small devices. On the other hand, due to the architecture of LIP and its use of public key as the host part of the address, if a user moves within the domain the address of the device will not change at all. The only update that is needed is the change of Access Point. If the host moves from the range of one AP (Access Point) to another AP, then the AP needs to signal the router about the new terminal in its range so that the router can update its internal routing table. The corresponding hosts would send the data traffic to the domain gateway as before, just the data needs to be routed to a different AP within the domain.

4.2 Prevention against intermittent connection

A common feature of small mobile devices is that they are often intermittently connected, i.e., they get disconnected while people travel with them, or switch them off and on again. Traditional TCP/IP is not suitable for handling this type of scenario. The shortcomings of traditional protocol stack with intermittently connected devices are outlined below:

1. Whenever a user switches between access networks traditional IP address (IPv4) needs to be changed and consequently, the transport layer connection breaks.

2. TCP defines a system-wide Timeout which specifies the maximum time that data may remain unacknowledged by a peer. After the timeout expires, the connection is considered stale and system resources are reclaimed by aborting the connection. In this way, any temporarily disconnected peer can be considered lost and thus the transport layer connection would be aborted.

3. Another type of timer in TCP, namely Retransmission timer specifies that if a host stops receiving acknowledgements from a peer, it tries retransmitting the packet only after a predefined retransmission period. It does not matter whether the peer gets connected again long before the retransmission period ends; the first retransmission happens only after the timeout which results in idle connected time and underused resources.

LIP does not suffer from the IP address change scenario (problem 1) due to its address structuring. So the connection abort is not a factor in LIP environment. To address the problem caused by the TCP timeout option (problem 2), LIP uses the TCP User Timeout Option [7] in a way as proposed in [20]. This option allows hosts to exchange per-connection timeout requests which enable mobile hosts to maintain TCP connections between disconnected periods longer than the default timeout. When a host knows that it would be intermittently connected, then it negotiates a suitable timeout value with the server it is connecting to based on empirical basis. To alleviate the problem of retransmission delay (problem 3) TCP retransmission trigger [8] is used. It proposes to attempt speculative, additional retransmissions in case of a restored TCP connection to a previously connected device. This behavior tries to send retransmission packets whenever the TCP implementation detects a restored connection rather than waiting for the system wide retransmission timer to go off. According to the practical implementations done by [20], it has been shown that including these features raise the performance of intermittently disconnected devices up to 86%-96% of constant connectivity.

5 Discussion

The design of LIP has been engineered by keeping in mind the lightweight mobile devices with low processing power. Use of public key as the host address enables auto configuration of the devices as well as allows the secured trust enabled networking paradigm to be used. This type of addressing scheme also enables address to be used as an identifier of the device rather than as a locator. The paper proposes use of ECC as a key generating and verification process and elaborates an implementation of highly lightweight version of ECC. Along with cryptographically generated addresses, a certificate from a trusted third party is included in the header of LIP which enables any intermediate nodes in the path of the packet to verify and authenticate the packet. The use of trusted third party is necessary to reduce the amount of computation in the devices. This paper also proposes a way to carry on data communication in case of an ad-hoc network, where the nodes do not have access to infrastructure nodes, using P2P refresh process. The mobility of the hosts are handled much similar to the way as HIP does it, using a three way handshake for macro mobility and a local rendezvous point for micro mobility. This feature allows hosts to roam around the network without having to acquire new network layer address and establishing a new connection each time. The possibility of intermittent connection has also been taken into account. As small devices tend to be turned off by the user or due to power saving mode pretty quickly, so they experience disconnection in between their connected mode. To recover from the performance loss due to this kind of disconnected period, LIP uses TCP user timeout and TCP retransmission trigger.

6 Conclusion

In the coming age of pervasive computing and all-IP networks, the need for a network layer protocol which is inherently secure and lightweight is unavoidable. This paper outlines such a protocol with the relevant features such as security, accountability, mobility and energy efficiency for mobile devices. Although not very extensively, this paper casts light on the implementation of the most important features of the proposed protocol which makes LIP suitable for lightweight devices. The research area is not ripe enough and more work needs to be done in the future to further optimize the different techniques specified here to support energy efficiency without sacrificing any of the properties.

References

[1] J. Abley, P. Savola, and G. Neville-Neil. Deprecation of Type 0 Routing Headers in IPv6. RFC 5095 (Proposed Standard), Dec. 2007.

[2] D. G. Andersen, H. Balakrishnan, N. Feamster, T. Koponen, D. Moon, and S. Shenker. Accountable Internet protocol (AIP). In *Proc. ACM SIGCOMM*, Seattle, WA, Aug. 2008.

[3] T. Aura. Cryptographically Generated Addresses (CGA). RFC 3972 (Proposed Standard), Mar. 2005. Updated by RFCs 4581, 4982.

[4] R. Bush and D. Meyer. Some Internet Architectural Guidelines and Philosophy. RFC 3439 (Informational), Dec. 2002.

[5] W. R. Cheswick, S. M. Bellovin, and A. D. Rubin. *Firewalls and Internet Security: Repelling the Wily Hacker*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2003.

[6] S. Deering and R. Hinden. Internet Protocol, Version 6 (IPv6) Specification. RFC 2460 (Draft Standard), Dec. 1998. Updated by RFC 5095.

[7] L. Eggert and F. Gont. Tcp user timeout option. work in progress(draft-ietf-tcpm-tcp-uto-09), 2008.

[8] L. Eggert, S. Schuetz, and S. Schmid. Tcp extensions for immediate retransmissions. work in progress(draft-eggert-tcpm-tcp-retransmit-now-02), 2005.

[9] T. Eisenbarth, S. Kumar, C. Paar, A. Poschmann, and L. Uhsadel. A survey of lightweight-cryptography implementations. *IEEE Design and Test of Computers*, 24(6):522–533, 2007.

[10] A. Gurtov. *Host Identity Protocol: Towards the Secure Mobile Internet*. John Wiley and Sons, 2008.

[11] T. Hasegawa, J. Nakajima, and M. Matsui. A practical implementation of elliptic curve cryptosystems over gf(p) on a 16-bit microcomputer. In *PKC '98: Proceedings of the First International Workshop on Practice and Theory in Public Key Cryptography*, pages 182–194, London, UK, 1998. Springer-Verlag.

[12] S. Kent and K. Seo. Security Architecture for the Internet Protocol. RFC 4301 (Proposed Standard), Dec. 2005.

[13] D. Lagutin. *Redesigning Internet - The Packet Level Authentication architecture*. Licentiate's thesis in computer science, Helsinki University of Technology, Espoo, Finland, 2008.

[14] R. Moskowitz and P. Nikander. Host Identity Protocol (HIP) Architecture. RFC 4423 (Informational), May 2006.

[15] V. Paxson and M. Allman. Computing TCP's Retransmission Timer. RFC 2988 (Proposed Standard), Nov. 2000.

[16] J. Postel. Internet Protocol. RFC 791 (Standard), Sept. 1981. Updated by RFC 1349.

[17] J. Postel. Transmission Control Protocol. RFC 793 (Standard), Sept. 1981. Updated by RFC 3168.

[18] A. Ramachandran and N. Feamster. Understanding the network-level behavior of spammers. *SIGCOMM Comput. Commun. Rev.*, 36(4):291–302, 2006.

[19] M. G. Reed, P. F. Syverson, and D. M. Goldschlag. Anonymous connections and onion routing. *IEEE Journal on Selected Areas in Communications*, 16:482–494, 1998.

[20] S. Schütz, L. Eggert, S. Schmid, and M. Brunner. Protocol enhancements for intermittently connected hosts. *SIGCOMM Comput. Commun. Rev.*, 35(3):5–18, 2005.

[21] A. Snoeren and H. Balakrishnan. An End-to-End Approach to Host Mobility. In *6th ACM MOBICOM*, Boston, MA, August 2000.

[22] W. Stevens. *TCP/IP Illustrated, Volume 1: The Protocols*. Addison-Wesley, 1994.

[23] P. F. Syverson, D. M. Goldschlag, and M. G. Reed. Anonymous connections and onion routing. *Security and Privacy, IEEE Symposium on*, 0:44, 1997.

Tor and Onion Routing: Protecting your privacy

Ravishankar Borgaonkar
Helsinki University of Technology
rborgaon@cc.hut.fi

Abstract

The growth of information and services on the Internet is accompanied by concerns over privacy. Traffic analysis is a key threat to the right to privacy. Traffic analysis and eavesdropping techniques can be used to attack secured systems, extract secret information of the user and be used to abuse or spam systems. Onion Routing can support anonymous communications over public networks by providing near real-time and bi-directional anonymous TCP connections that are resistant to both eavesdropping and traffic analysis attacks. This paper presents a technical description of Tor protocol which is based the design of Onion Routing protocol, that not only protect the privacy of the sender and recipient of a message but also the message content as it traverses through the networks. Thus, it provides untraceability; unobservability to the user on the move. In addition, it allows user to run and access hidden services and gives censorship resistance.

KEYWORDS: Privacy, Onion Routing, Tor, Anonymous, Traffic Analysis

1 Introduction

The most popular uses of the Internet over the last few years have been email and web browsing. Therefore, to ensure communications privacy, there is a need of an anonymity infrastructure to enable users to perform these activities free from intrusion by various attackers. Internet communications can be encrypted, so that an attacker can not extracts the content of these commuunication. HOwever, this still reveals the fact that two parties are communicating. Moreover encryption solves only part of the anonymity issue: It hides what is being said, but not who is communicating. Traffic analysis and eavesdropping techniques can be used to attack secured systems, extract secret information and be used to abuse or spam systems [5]. By using Tor and Onion Routing concept, Anonymous services take that next step to protect the parties in an online communication.

Onion Routing (OR) is a traffic protection system developed and prototyped by the Naval Research Laboratory. It uses a forwarding-and-mixing approach first proposed for e-mail by David Chaum in [4]. Onion routing accomplishes privacy according to the principle of David : messages travel from source to destination via a sequence of proxies (onion routers), which re-route messages in an unpredictable path. To prevent an adversary from eavesdropping on message content, messages are encrypted between routers. The advantage of onion routing (and mix cascades in general) is that it is not necessary to trust each cooperating router; if one or more routers are compromised, anonymous communication can still be achieved. This is because each router in an OR network accepts messages, re-encrypts them, and transmits to another onion router. An attacker with the ability to monitor every onion router in a network might be able to trace the path of a message through the network, but an attacker with more limited capabilities will have difficulty even if he or she controls one or more onion routers on the message's path. The current state of the art in anonymous networks is Tor, the second generation onion router project based on the onion routing concept. Routing in Tor is done at the transport level in the protocol stack and only supports TCP. Applications using Tor protocol access the network through a web proxy interface called Privoxy. It means that all applications can use Tor for anonymous communication with the support of these web proxies. Also these application does not need to be modifiled in order use Tor services. Tor gives privacy to the mobile user by adding perfect forward secrecy, congestion control, directory servers, integrity checking, configurable exit policies and a practical design for location hidden services via rendezvous points [6].

The rest of the paper is organised as follows. Section 2 describes Onion Routing concept. In addition, we also discuss how this protocol can provide anonymity and privacy to the mobile user in Section 3. In section 4, we present Tor protocol which is based on onion routing technology. Section 5 describes a step-by-step working procedure of the Tor software. Benefits of using Tor architecture to the user are discussed in section 6. We address the limitations of Tor in Section 7. Conclusions are presented in Section 8.

2 Onion Routing

Onion routing was conceived in 1996 by David M. Goldschlag, Michael G. Reed, and Paul F. Syverson for the NRL's research group in high assurance systems. It is an architecture in which a proxy uses special routers to create anonymous connection to other proxies. It operates by dynamically building anonymous connections within a real-time network of David Mixes, also known as onion routers, which are connected through permanent Transmission Control Protocol:TCP connections.

Onion Routing consists of two parts: the network infrastructure that accommodates the anonymous connections, and a proxy interface that links these connections to unmodified applications. The public network contains a set of Onion Routers. An Onion Router is a store and forward device that accepts a number of fixed-length messages, performs some cryptographic transformations on them and then forwards the encrypted messages to the next destination in a random order. An anonymous connection is routed through a sequence of neighboring Onion Routers.Onion proxies act as interfaces between the applications and the network infrastructure. In Onion Routing, the functions of a proxy can be split into two: one part links the initiator to the anonymous connection and the other part links the anonymous connection to the responder. To build onions and define routes the onion proxies must know the topology and link state of the network, the public certificates of nodes in the network, and the exit policies of nodes in the network [13].

Onion routing, like mixed-key cryptography, uses a bulky but carefully secured initialization procedure to establish fast, lightweight communication via symmetric keys that only the legitimate participants in the communication know [7]. These participants include the sender, the receiver, and any subset of a network of proxy servers that forms a path from the sender to the receiver. Public-key cryptography keeps the global routing information secret from all participants except the sender, and thereby protects the path of communication from discovery by an attacker at any intermediate node [8].

Routers communicate with each other over TCP. Some routers also can serve as entry funnels, they can accept connections from the clients of the network. Some routers can serve as exit funnels, they can create TCP connections leaving the network to the actual Internet services that are being accessed through the Onion Routing network. Such services can be world wide web, e-mail, peer-to-peer applications, etc. An onion is a multilayred data structure. It encapsulates the route of the anonymous connection within the onion routing network. Each layer in it contains-backward crypto function (DES-OFB<output feedback mode>; RC4) forward crypto function (DES-OFB, RC4) [14]; IP address and port number of the next onion router ; expiration time; key seed material (used to generate the keys for the backward and forward crypto functions).

3 Anonymity by Onion Routing

Suppose, Alice and Bob wants to communicate anonymously using onion routing concept. On the Alice's machine, client application establishes an anonymous connection to a server by connecting to an application proxy. Role of these application proxies, like a SOCKS proxy [9] is to accept protocol specific connections from application, and converts them into a generic protocol. Packets of these protocols are then forwarded to an onion proxy.

As shown in the figure 1, the onion proxy (on Alice's machine) constructs a random sequence of routers on the network it knows about. In this sequence, router A is a an entry funnel and router C is as an exit funnel. It also constructs an onion which is defined in section 2. Then, these onions are encapsulated in a packet and sent to the entry funnel. Structure of these onions would be like this: A[B [C [Bob [M]]]]. In this onion, M is the message which Alice wants to send Bob. Message M is encrypted by using Bob's public key and this encrypted packet is then reencrypted with the public keys of the router C, B , and A respectively. Router A, an entry funnel, is able to decrypt the onion with its private key. This decryption process reveals IP address of the router B and chunk of encrypted data. Then it forwards this chunk to the router B, and the same procedure repeats at the exit router C. Bob receives the message sent by Alice from the router C. In other words, Bob is getting all the messages from the IP address of the router C, not from the Alice's IP address.

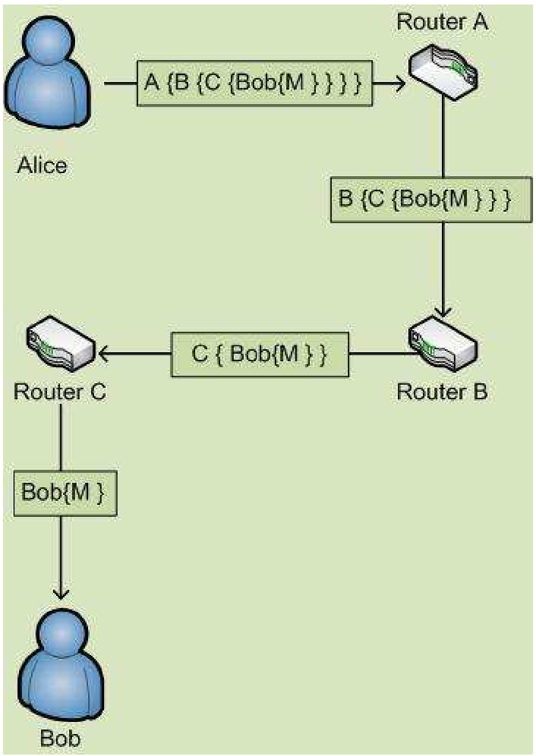

Figure 1: Anonymous communication between Alice and Bob

In this entire process, router first has to decrypt the onion with its own private key in order to retrieve the next hop in the route.Therefor, it is impossible for the attacker or sniffer who intercepts the onion to extract the IP address for the next hop, because only the router knows his private key. In addition, the process of encrypting the entire onion for each hop provides a big advantage by making the onions completely different at each router. Thus it is very hard to correlate it between routers.

Onion connection operates in three phases: Connection Set-up, Data Movement and Connection termination. Following sub-sections 3.1, 3.2 and 3.3 describes them respectively. These phases are important in order to understand Tor architecutre which is the advanced version of Onion Routing.

3.1 Connection Setup

Setup begins with the initiator creates a layered data structure (Onion) that specifies properties of the connection at each point along the route,e.g., cryptographic control information like the different symmetric cryptographic algorithms and keys used during the data movement phase. Each Onion Router along the route decrypts onion packet it receives using his private key. This process gives the cryptographic control information for this onion router, the identity of the next onion router in the path for this connection,and the embedded onion. The onion router pads the embedded onion to maintain a fixed size and sends it onward. The final onion router in the path connects to a responder proxy, which will forward data to the remote application. In short connection steps will be:

- The application is configured to connect to the real destination
- Upon a new request , the application proxy decides whether to accept the request
- Opens a socket connection to the onion proxy
- Passes a standard structure to the onion proxy
- Standard structure contains: An application type (e.g., HTTP, FTP, SMTP.); Retry count (number of times the exit funnel should retry connecting to the destination); Format of address that follows (e.g.,NULL terminated ASCII string) [14]; Address of the destination (IP addr and Port number)
- Waits for the response from the exit funnel before sending application data.
- upon reception of the standard structure, the onion proxy: decides whether to accept the request ; establishes an anonymous connection through some randomly selected onion routers by constructing and passing along an onion; sends the standard structure to the exit funnel of the connection; after that, it relays data back and forth between the application proxy and the connection
- Upon reception of the standard structure, the exit funnel: tries to open a socket connection to the destination; it sends back a one byte status message to the application proxy through the anonymous connection (in backword direction); if the connection to the destination cannot be opened, then the anonymous connection is closed; otherwise, the application proxy starts sending application data through the onion proxy, entry funnel, anonymous connection, and exit funnel to the destination

3.2 Data Movement

After the anonymous connection is established in first phase, data can be sent in both direction. Onion proxy (the initiator's) receives data from an application,breaks it into fixed size cells ,and encrypts each cell multiple times using the algorithms and keys specified in onion data structure. Before sending data over an anonymous connection, the initiator's onion router adds a layer of encryption for each onion router in the route. Each onion router in the connection path removes one layer of encryption, so data arrives at the receiver as plaintext. This layering process occurs in the reverse for data moving backword and successive onion routers encrypting data using different algorithms and keys.

3.3 Connection Termination

Anonymous connections are terminated by the initiator, the responder, or one of the onion router in the middle of the connection path. For this purpose, a special DESTROY message is propagated by the onion routers [14]:

-If an onion router receives a DESTROY message , it passes it along the route (forward or backward).

-The router sends acknowledgement to the onion router from which it received the DESTROY message.

- If an onion router receives an acknowledgement for a DESTROY messages it frees up the corresponding ACI.

ACI- A connection identifier and it is unique on a link.

4 Tor Architecture

Tor [3] is a widely distributed overlay network for anonymizing network traffic. The Tor network is based on Onion Routing protocol design [13] described above in Section 2 and 3. However, it differs in some implementation details. These important differences are *perfect forward secrecy*, *congestion control*,*directory servers*,*integrity checking*, *configurable exit policies* and *a practical design for location hidden services via rendezvous points* [6]. In section 3, Onion Routing protocol allows only the last router in a route to act as the exit funnel. But, by adding configurable exit policies, Tor changes this concept slightly by allowing any router along the route act as an exit funnel. Thus, Tor makes it more harder for the attacker (who is observing the end of a circuit) to figure out where the traffic goes. Moreover, it reduces the latency of all nodes and also this added feature makes it possible for an individual user to increase his or her anonymity. Third difference, directory servers is descried in the following Section 4.1.

Tor relays arbitrary TCP streams over a network of relays, and is particularly well tuned to work for web traffic, with the help of the Privoxy [2] content sanitizer. Tor uses a traditional network architecture: a list of volunteer servers is downloaded from a directory service. Then, clients can create paths by choosing three random nodes, over which their communication is relayed. Instead of an onion being sent to distribute the cryptographic material, Tor uses an iterative mechanism. The client connects to the first node, then it request this node to connect to the next one. The bi-directional channel is used at each stage to perform an authenticated Diffie-Hellman key exchange.

Tor architecture consits of following terms: Directory servers, cells , circuit and streams. These are discussed in the following sections.

4.1 Directory Servers

Tor routers are registered with a directory service and each router reports its IP address, public key, policies about what traffic it will accept, and a bandwidth value that is determined by monitoring the peak bandwidth achieved by the router over a period of time [12]. These servers maintain list of which onion routers are up and exit policies. Directory server keys ship with the code. They control which nodes can join the Tor network. These directories are cached and served by other servers, to reduce bottlenecks.

4.2 Cells

Cells are the unit of communication in Tor. After the anonymous connection set-up, onion router communicate with one another. This communication traffic passes along anonymous connections in fixed-size cells. Each cell packet consits of a header and a payload, and is 512 bytes. A header contains a circuit identifier (circID) that specifies which circuit the cell refers to and a command to take actions on the cell's payload. Cell could be a control cell or a relay cell depending on their command. Control commands are: padding (used for keepalive); create or created(used to build new circuit); and destroy (used to terminate a circuit connection) There is an additional header at the front of the payload in Relay Cell, which contains a streamID; an end-to-end checksum for integrity checking; the length of the replay payload; and a relay command. Relay commands are: relay data, relay begin, relay end, relay teardown, relay connected, relay extend and relay extended, relay truncate and relay truncated, relay sendme, and relay drop.

4.3 Circuit and Streams

Tor builds one circuit for each TCP stream and that can be shared by many TCP streams. It avoids the delay caused in building a circuit due to public-key cryptography and network latency) Users' onion proxies build a new circuit periodically if the previous ones have been used, and expire old used circuits that no longer have any open streams. Onion proxies keeps rotating to a new circuit once a minute. A user's onion proxy construts circuits incrementally by negotiating a symmetric key with each onion router on the circuit, one hope at a time. This process of building a circuit creates session keys by using Deffie-Hellman key exchange protocol (to provide perfect forward secrecy). Once user has established the circuit, he can send Realy cells. After receiving it, an Onion router looks up the corresponding circuit, and decrypts the realy header and payload with the session key for that circuit.

A Tor architecture design is described in the paper [6] by Roger Dingledine, Nick Mathewson and Paul Syverson .

5 How Tor Works

We have installed Tor software and Privoxy on our machine to check how it works. We used a Vidalia, a cross-platform controller GUI for the Tor software. Vidalia [1] helps to set up and manage a Tor server. We try to load BBC.com home page through Tor and it looks like this: (This is a simplified process.)

1. User A's Tor client obtains a list of Tor nodes from a directory server.
2. Tor client incrementally builds a circuit of encrypted connection through relays on the network.
3. Client contacts two TOR nodes, which we call A and B, and requests their public keys.
4. Our Machine forms the web request to BBC.Com.
5. It encrypts the web request in key B, then encrypts the result (along with the address of B) in key A.
6. The packet is sent to node A.
7. Node A, which has the private portion of key A, decrypts the packet. Inside is another address (that of B) and an encrypted packet. Thus, Node A knows you you are, but it does not know what we are transmitting, or who we are sending it to. It forwards the packet to Node B.
8. Node B, which has the private portion of key B, decrypts the packet. Inside is the transmission to BBC.com, which by its nature says where it should be sent. Thus, Node B knows what we are transmitting, and who we are transmitting it to, but has no idea who we are or where the packet came from. Node B sends the packet to BBC.com
9. BBC.com gets the packet and replies, sending the reply to Node B. Note that BBC.com, like Node B, has no idea who we are or where the packet came from.
10. Node B gets the reply and forwards it to Node A.
11. Node A gets the reply and forwards it to our machine.(Node A and B perform additional encryption so the nodes can not read the reply as it make back to our machine and there can be more than two nodes in the chain.)
12. We get the BBC.com homepage.

6 Protecting Privacy Using Tor

6.1 Anonymity in Common Social Interactions

There are many factors why anonymity is important. First there are repressive governments that forbid access to certain sites, censor the Internet, and then track clients who show interest in particular topics. There are people who want to tell the truth without fear of repercussion, such as corporate whistleblowers and bloggers. Aonymity protocols, such as anonymous cash and voting protocols,that allow the client to remain anonymous. However, their problem is that the client could still be tracked by its IP address ,either by the server or by an attacker spying on the network traffic near the server. Tor prevents these problems and gives freedom to clients to post anonymous messages on blogs, discussion forums and also to send anonymous emails, by hiding users IP address.

6.2 Untraceability and Unobservability to the Mobile User

Tor helps to hide the mobile user's location from servers to which it connects. Tor Provides location privacy for the mobile user towards internet servers by giving the user the control over their personal identifiable information(PII) on the internet while on the move. Mobile user wants to hide personal identifiable information while on the move. VPN is a method to achieve this, however, a VPN reveals the mobile user's location to the VPN gateway, which is not always desirable. Here also, Tor achieves this. The onion layered architectural encryption hides nature and characteristics of the traffic between node-pairs. The mobile user will leak less PII to the access network on the move because all communication is encrypted and goes through Tor only. However, end-to-end encryption needs to achieve untracebility and unobservability in this scenario.

6.3 Censorship Resistance

There is a growing trend of censorship on the Internet at country-wide level. China is the country most famous for having a firewall which both prevents users inside the country from having access to outside content and also logs who is accessing what. Tor will automatically try to access a Internet service through many Tor servers until it succeeds in establishing a connection. Users will be able to access websites blocked by their ISP or their government through Tor as long as the tor exit node has access to the website.

6.4 Hiding Server Location

Tor can be configured in two ways; as a client and as a server. Tor servers can run hidden services. This means that if Tor server or service does not reveal information that gives away any information which gives a clue to it's location then the services location really is hidden from the world. The obvious advantages are:

- Government can not find out who is running the service.
- Government can not shut down these services.

6.5 Bypassing the Firewall

Tor could be used as a tool to bypass the firewall. In the firewall, network administrator sets inbound and outbound rules to control network traffic. It gives protection against the *IP spoofing attack*. An IP spoofing attack occurs when an attacker outside the network pretends to be a trusted computer either by using an IP address that is within the range of IP addresses for users network or by using an authorized external IP address that network trust and to which network administrator wish to provide access to specified resources on the network.

Firwall protects the network against this attack by checking whether the packet's source IP address is valid. In Tor, we can set the exit router for our circiut connection path using leaky-pipe circuit topology studied in Section 4. Suppose, an attacker wants to access different prohibited services inside the network of "ABC" university. One trusted user, Alice, inside the "ABC" network is running Tor client and acting as a relay server. Attacker can set the address of Alice's machine as an exit router for this anonymous connection. When attacker tries to access serives inside the "ABC" network , all his access queries would go through Alice's machine. This means server assumes that trusted client Alice is trying to access these services and gives access to them. In this way , Attcker could get access to such prohibited services. However, to bypass the firewall, at least an user inside the network should be running Tor client application and acts as a relay server. In addition, firewall rules defined by the administrator should allow inbound and outbound Tor traffic, in other words, the client inside that network must be able to run Tor application. This method of bypassing the firewall could be a weakness of Tor in some sense.

7 Tor Limitations

7.1 Abuse

Running an exit node could create a risk for abuse. There has been a lot of misuse of Tor for spam, threats, hacking, abusive IM and illegal file sharing. Moreover, hackers are also using it for hiding and running their Botnets. After the introduction of more strict policies for exiting traffic this has been reduced. The default exit policy now rejects private IP subnets, email (SMTP), Usenet News (NNTP), Windows file sharing and also a number of popular file sharing applications (eDonkey, Gnutella, Bittorrent). However, an user running an exit node could be contacted by the Police for abusing through Tor. Also, there might be issues with ISP aggreements forbidding traffic relaying.

7.2 Attacks and Weakness

Web traffic is stateless; each web request is not tied to any other in any persistent way. When user load a web page, browser nearly-simultaneously requests the page and all the images, media, embedded frames, ads, etc. on the page. The web server sets a session cookie (a cookie that is deleted when user closes the browser) when user load the first page, and uses that to track your movement through the site. It is not a tracking process, it is just linking all users' page loads together to provide a sense of state or flow and the web does not work without it. Thus, Privoxy has to let these session cookies through. This can leak a little bit of information about the user. Larger number of Internet sites are collecting personal information from users through forms, cookies, online registrations, or surveys than ever before. If the user fills in web forms, registrations or surveys, or the web browser accepts cookies, Tor will not prevent tracking of the user. Tor only hides the client IP address from the server. If the client user or software voluntarily gives identifiers or other information to the server, Tor cannot prevent that because it works only at the IP layer. It does not prevent application-layer protocols, such as user input or HTTP cookies, from leaking the user identity to the server.

Tor anonymizes the origin of user's traffic, and it makes sure to encrypt everything inside the Tor network. But Tor does not encrypt all traffic thoughout the internet, more specifically traffic between the exit router and the server. Because the exit router has to spit out unencrypted data, otherwise the final destination would have no idea as to what it was receiving. There is a need of end-to-end encryption at the application layer for that(e.g. SSL). Tor is slightly vulnerable to traffic analysis between the client and the entry router. It does some padding of packets but not much, and it does not not attempt to change the timing of packets. Therefore, traffic analysis may reveal the type of traffic (e.g. web browsing,VoIP or BitTorrent).

Tor protects users against traffic analysis [8]. But if the attacker has the control of first and last link of a Tor Connection, attacker could observe the traffic. Correlating timing of traffic at the end points is the most effective means of attacking Tor. If the traffic timing and packet size match up, it is possible to determine that a user may be communicating with someone. This requires an attacker to be either extremely lucky to find this or have a global view of the network [11].

The user can hide his online activity using Tor. If someone hacks into the users' computer (or seizes it, in the case of legal authorities), they do not need to have logs from the other end to know what the user have been doing; users' computer probably has records of their activity. To delete these logs could help little in the legal case but modern data recovery can get deleted data easily. To avoid this, it is necessary to have a computer that does not keep any records. This is hard to do in a normal Windows or Linux system (they both arbitrarily swap portions of memory to disk during normal use.) For better better anonymity and privacy, it is necessary to boot a LiveCD environment (i.e. an Operating System with no hard drive or writeable media), where all evidence is destroyed when the computer is powered down.

Anyone can be act as a exit Tor node. In the example stated in the section 5 , Node B gets to see users' traffic in both direction. It does not know users' IP address. Users are trusting on these exit nodes on the internet and giving them ability to carry out man-in-the-middle (MITM) attacks on them. These exit node may not forward internet traffic to BBC.com at all, but rather could to a fake site. Exit node could edit the traffic to add a virus or Trojan. So Tor may risk users' privacy.

Since DNS is a UDP protocol, the client machines will often attempt to query DNS servers outside of the Tor network (which is a TCP stream). By making these requests anyone monitoring the clients connection can determine the servers or the client is attempting to access. These protocols need to be Torized so that they can connect over a TCP socket.

The new implementation in Tor which allows individual users to run a Tor node is a negative aspect of Tor in comparison to classical Onion Routing concept. Using this option, a malefic third party with enough resources could compromise the anonymity of all users by adding a significant number of nodes to the network and perform analysis of all traffic through these compromised nodes. Thus, it risks privacy and anonymity of the user. This paper [11] proves that traffic analysis is theoretically possible by adding such corrupt servers.

In systems such as Tor, the Sybil attack is generally capable of revealing the initiator of a connection [15]and there is no defense against this attack [6], [10].

The attacker can flood Tor routers or nodes with requests. This may cause a denial-of-service (DOS), as the mass of encrypted packets requires significant computing resources to process or simply exhaust the available bandwidth. Tor protocol does not provide protection against these DOS attacks.

8 Conclusion

We have presented a protocol Onion Routing and an overview of how this protocol could help users to protect their privacy and anonymity on the Internet. The purpose of Onion Routing is to protect the anonymity of a user who wants to communicate over a network. In particular, it will hide the destinations of all communications initiated by the user. Any outside observers will not be able to tell whom the user is communicating with and for how long. Tor,which is based on Onion Routing concept, is an anonymizing Internet proxy service designed to circumvent traffic analysis by proxying TCP traffic within chained, encrypted tunnels. It offers untraceability; unobservability to the user on the move; provides censorship resistance; and allows to run and access hidden services without fear of the government. However, it has some problems that need to be addressed by the research community. Overall, it is a valuable tool, but if someone wants to track you down badly, and they have resources or authority, they could still do so.

References

[1] A cross-platform controller gui the tor software, http://www.vidalia-project.net/, Last visited Nov. '08.

[2] Privoxy, http://www.privoxy.org/, Last visited Nov. '08.

[3] Torproject, http://www.torproject.org/, Last visited Nov. '08.

[4] D. Chaum . Untraceable Electronic Mail, Return Address, and Digital Pseudonyms. In *Communication of the ACM*, volume 24, pages 84–88, Feb 1981.

[5] G. Danezis. Introducing traffic analysis,"http://homes.esat.kuleuven.be/ gdanezis/taintrobook.pdf", Last visited Nov. '08.

[6] R. Dingledine, N. Mathewson, and P. Syverson. Tor: the second-generation onion router. In *SSYM'04: Proceedings of the 13th conference on USENIX Security Symposium*, pages 21–21, Berkeley, CA, USA, 2004. USENIX Association.

[7] D. M. Goldschlag, M. G. Reed, and P. F. Syverson. Hiding Routing Information. In R. Anderson, editor, *Proceedings of Information Hiding: First International Workshop*, pages 137–150. Springer-Verlag, LNCS 1174, May 1996.

[8] M. Hooks, J. Miles, F. Css, P. Reynolds, and O. Astrachan. Onion routing and online anonymity, 2006.

[9] M. Leech, M. Ganis, Y. Lee, R. Kuris, D. Koblas, and L. Jones. RFC 1928: SOCKS protocol version 5, Apr. 1996. Status: PROPOSED STANDARD.

[10] B. N. Levine, C. Shields, and N. B. Margolin. A Survey of Solutions to the Sybil Attack. Tech report 2006-052, University of Massachusetts Amherst, Amherst, MA, October 2006.

[11] S. J. Murdoch and G. Danezis. Low-cost traffic analysis of tor. In *SP '05: Proceedings of the 2005 IEEE Symposium on Security and Privacy*, pages 183–195, Washington, DC, USA, 2005. IEEE Computer Society.

[12] R. Snader and N. Borisov. A tune-up for Tor: Improving security and performance in the Tor network. In *Proceedings of the Network and Distributed Security Symposium - NDSS '08*. Internet Society, February 2008.

[13] P. Syverson, G. Tsudik, M. Reed, and C. Landwehr. Towards an Analysis of Onion Routing Security. In H. Federrath, editor, *Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability*, pages 96–114. Springer-Verlag, LNCS 2009, July 2000.

[14] P. F. Syverson, D. M. Goldschlag, and M. G. Reed. Anonymous connections and onion routing. In *SP '97: Proceedings of the 1997 IEEE Symposium on Security and Privacy*, page 44, Washington, DC, USA, 1997. IEEE Computer Society.

[15] M. K. Wright, M. Adler, B. N. Levine, and C. Shields. The predecessor attack: An analysis of a threat to anonymous communications systems. *ACM Trans. Inf. Syst. Secur.*, 7(4):489–522, 2004.

Security and Privacy Issues of Biometric Identification

Timo Päkkilä
Helsinki University of Technology
timo.pakkila@tkk.fi

Abstract

Biometric identification is a method for identifying people based on what they are rather than using traditional knowledge- and token-based identification methods. Some identification methods include fingerprints matching, retinal and iris scanning, and DNA matching. Fingerprints have been widely used in certain applications but not yet in wider commercial use.

There are several issues relating to use of biometric technology. These include performance and accuracy issues, and the security of biometric data storage and transfer. Another types of problems consist of social and privacy issues. Performance and accuracy are crucial in order to be able to use biometric methods in any application. General audiences may, in addition, have difficulties understanding the advantages of giving up their physical characteristics to security usage. They might feel uncertain and wary about how the biometric data will be used, and whether they are stored in a secure manner.

This paper introduces the reader to biometrics and discusses related problems that hinder vast usage of biometric technology.

KEYWORDS: Biometrics, Identification, Verification, Fingerprints, Iris Scan, Privacy, Security

1 Introduction

Traditional authentication is performed by a token-based or knowledge-based system. In these authentication schemes someone usually remembers a password or has an access card with which he is granted access somewhere. A third authentication scheme is based on granting access based on something what a person is, and biometrics belongs to this category. Biometrics is, therefore based on what a person is rather than what he possesses or knows. This type of identification can be considered more secure than other types of identification as it is more difficult to steal person's physical properties or to fabricate a characteristic than apprehend a access card or a password.

Biometric identification and verification are identifying people based on what they are, for example, fingerprint, facial image or iris pattern. These qualities are unique for every human being and, thus, can be used for identification purposes. Some problems in biometrics are related to privacy issues and certain security problems such as biometric data storage and accuracy of the methods. Privacy issues rise from people being uncomfortable of giving up personal information such as retinal images or fingerprints to a database. That can be perceived as a threat or invasion to one's privacy.

Despite the obstacles, biometric identification is becoming more reliable and accurate for human identification and it has been researched widely in recent years. For example, fingerprints and facial images have been in use for quite a long time already, especially in criminal investigation usage. Nevertheless, biometrics has not yet been used widely, because of some privacy and feasibility issues. New methods have also been investigated as there has been research on light biometric methods with unobtrusive identification for low-security applications [1]. They have suggested using such features as body weight and height, possibly in combination, in order to achieve more accurate results.

This paper presents an overview of biometrics and its problems. Introduction to biometric technology is given in section 2, and sections 3 and 4 discuss some security and privacy issues.

2 Background on Biometrics

This section introduces the reader to biometrics in general and presents some biometric methods. Some biometric methods are fingerprint recognition, facial thermogram and iris scanning. None of the methods have all the properties and fulfill every criteria mentioned in the following section but can still be used in real-world applications. The decision which method to use depends on the environment, users of the application and how accurate and fast the identification needs to be.

2.1 Properties of Biometrics

Biometric technology can be used in three general applications [8]: identification, verification and screening. They all use biometrics for identifying but verification is different as it uses additional data, such as smart-cards or passwords, in verifying the identity of a person. Screening is identifying without interaction of the person being identified. The following list explains shortly all three applications:

- *Identification*: identification means basically identifying a person solely based on a biometric characteristic, such as a fingerprint. For instance, a person's fingerprint is scanned and then the system searches a database for a match. If a close enough match is found in the database, the person is then identified.

- *Verification*: verification differs from identification in that a person presents an additional credential, such as

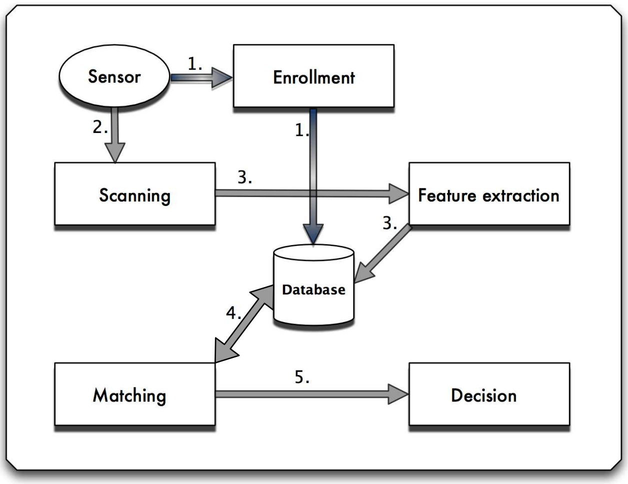

Figure 1: Biometric verification/identification system.

a smart-card holding the person's data, besides the biometric characteristic. The system then scans the biometric characteristic and compares it to the data on the smart-card. Verification is successful if the smart-card data and scanned data are close enough.

- *Screening*: screening is not very much different from identification; it is used to determine whether a person belongs to a certain watch-list of identities. The difference lies in that screening must occur without person knowing about the scanning which makes the matching harder because the scanning conditions are not ideal as they can be in identification processes. Screening is often used at airports or sports events to search for known criminals.

Biometric verification and identification consists of a few steps which are depicted in figure 1 and explained in the following list. Screening does not have exactly the same process, because it happens without interaction of the person being identified, but essentially it works in similar fashion.

1. *Enrollment*: in the enrollment phase the characteristic of a person is gathered and loaded to a database. This phase is separate from the others in the sense that it needs to be done before the biometric system is actually used for verification, identification or screening. In case of screening the enrollment usually occurs when someone has taken into custody for a felony.

2. *Scanning*: scanning is done using a sensor which reads the characteristic of a person. Scanning naturally happens also when enrollment is done but here enrollment is considered as the whole process of obtaining the biometrics and saving it to a database in a correct form.

3. *Feature extraction*: after scanning is done, the data needs to be analyzed and certain features extracted from the data in order to be able to compare the data in a database. After features are extracted the data is saved to a database.

4. *Matching*: matching is the process of searching the database for a match for the scanned biometric data.

5. *Decision*: after the matching process is done, system returns the best possible match and the decision, whether this match is close enough for identifying the person, is made based on the accuracy demands from the system.

Ideal biometrics has the following properties [7]:

- *Universal*: every human being possesses it.

- *Unique*: no two person share the same characteristic.

- *Permanent*: characteristic does not change over time nor can be altered.

- *Collectable*: technology exists for collecting the characteristic.

Universality can be fulfilled very closely but always there exist people who, for instance, have lost their fingers, or their eyes are damaged in a way that they cannot be used for identification. Uniqueness is achieved with many of the identification methods but the problem lies in technical inaccuracies. Most human characteristics are permanent but some might change when getting older or due to an accident. Collectability is fulfilled with the methods mentioned here, since otherwise they would not be discussed. However, some methods demands more from technological perspective that other. DNA matching, for instance, needs quite a lot of work and human interaction, while with some methods the entire matching can be fully computerized.

In addition to the preceding properties, the following criteria need to be taken into account when implementing a biometric system [7]:

- *Performance*: accuracy, speed and robustness of the method.

- *Acceptability*: public resistance of giving the characteristic to identification use.

- *Circumvention*: can the system be fooled easily.

Performance is a very important property because if matching takes too long it is not practical to use such methods in real-time applications. Acceptability is a difficult property to fulfill and there are no perfect solution for achieving total acceptance for any biometric method. Potential users need to be convinced that biometrics are a better way to identify them, it gives additional security and that it has not any major disadvantages or security flaws. Circumvention must be made as difficult as it can be done. Achieving this makes it easier to achieve also acceptability as users will be more relaxed if they know that no-one can fake their biometric data. In other words, no-one should be able to easily fool a biometric system either by using fake identification data or stealing another person's data.

2.2 Biometric Identification Methods

Two types of biometric characteristic exist: behavioral and physiological. Behavioral characteristics include such methods as signature, voice and keystroke. It has been researched that consumers do not consider behavioral characteristics as reliable as physiological biometrics and, thus, these methods will not be discussed in this paper [10].

In the following list are some existing physiological identification methods introduced briefly. Some of them will be discussed in more detail in section 3 when discussing the security of the methods from different aspects and in section 4 when discussing privacy and social issues.

- *Fingerprints*: fingerprints are the most known method of biometric identification and is considered to be quite accurate but at the same time it has a disadvantage of being traditionally used for criminal identification.

- *Iris scanning*: iris is the part of eye that has color. Iris scanning is a well researched area, and according to researchers it is very prominent method when accuracy and performance is concerned [14, 3].

- *Retinal scanning*: retina is also a part of the eye. Retinal scanning seems a viable method along with iris scanning [7].

- *Hand geometry*: hand geometry has been studied as an identification method for a few years and it has been considered to be most suitable for low-security applications [5, 4].

- *Facial image*: facial image is a picture of human face which distinguishes distinctive characteristics from it in order to identify a person. Facial images do not require physical interaction when identifying and they have been used in variety of applications. [9]

- *Facial thermogram*: facial thermogram is a pattern of heat radiated from human face captured by an infrared camera. This method does not require interaction with the person being identified which makes is possible to use it in certain applications where for example fingerprints could not be utilized .[9].

- *Finger vein recognition*: veins in the fingers of an individual are unique just like fingerprints and can, therefore, be used to person's identification [12].

- *DNA identification*: DNA is a well known identification method but currently there are no equipment to perform DNA matching in field circumstances. It is very accurate but needs quite a lot calculation power to be useful.

Besides these methods there are several others, but these are mentioned here as they represent the most researched methods and are, thus, considered to be the most viable methods in real-world applications, some of which are already in use in a number of locations.

2.3 Current State of Biometrics

Fingerprints have been used for criminal investigation for decades and a few years back every tourist traveling to USA should have fingerprints or an iris image embedded on her bio-passport or visa in digital form [18]. This policy is partly due to the terrorist attack in New York in September 2001. Passports are going to have some piece of biometric data included on a smart-card embedded on the passport in several counties. In the EU, biometric passports will include digital images of the owner and in the future also fingerprints.

Iris scanning has been taken into use at the Amsterdam airport in the Netherlands to speed up passport and visa control [8]. Similar technique is utilized in restricting access of employees to high-security areas at the Schiphol airport. It is obvious that there is a trend that implies that we will see a lot more biometric technology utilized in several applications all over the world in the near future.

3 Security of Biometric Identification

This section introduces some security related issues in biometric identification methods. There are several issues hindering vast usage of such methods in identification. Even though there have been significant progress in biometric technology and computers' calculation power has increased rapidly, some challenges still remain, and these challenges are discussed in this section.

3.1 Identification Data Storage

Storing identification data in a way that it is accessible but at the same time extremely secure and tamper proof is very difficult. Storage type will be different for different application types. A facility where a retinal image scan is required to grant access requires different storage solutions compared to a laptop having fingerprint-based identification for logging in. However, in every application the data need to be stored in an encrypted form because, for instance, if a perpetrator gets access to the database of biometric templates, as a piece of biometric data is called, they must be in such a format that they are not useful outside the system nor can be used in any harmful way.

Another issue in biometrics is acquiring the biometric data from users. With a user-name and password registering to system can be done solely online with a computer and an Internet connection. Biometric data capture require special equipment and knowledge how to use the equipment. Therefore, acquiring biometric data is much more complicated and perhaps due to this fact it may not be the best choice for every low-security application. An advantage in this more complicated acquiring process is that it makes the system more resistant to perpetrators.

3.2 Identification Data Transfer

Identification data is transfered from the scanning device to some processing unit which does the matching by comparing the acquired data to previously stored identification data.

This process needs to be secured in a way that it is not possible or at least extremely difficult to tamper with any data involved in the matching process.

Different approaches are researched and most of them suggest that every piece of data is sealed with cryptographic tools such as hash functions [19]. Waldmann et al. [19] suggest an on-card matching scheme for biometric verification. Their approach consists of a tamper resistant casing where the matching is performed and a cryptographic authentication method that consists of an RSA public key cryptography based encryption and authentication. Devices, in this context the user-card and the Security Module Card (SMC), authenticate themselves and agree upon symmetric keys which are used to encrypt all the data needed to be transfered.

Their solution is meant to be secure against such attacks as man-in-the-middle and replay attacks. Man-in-the-middle attack can happen when a perpetrator presents itself as another entity which someone is going to trust, and after that the perpetrator can monitor all the data he receives and possible acquire some confidential information. Replay attack is another type of attempt to acquire some piece of confidential data, it consists of reusing already authorized credentials, from a legitimate user, to access confidential data.

A certificate authority (CA) is used to grant certificates, which are needed for RSA public cryptography encryption, for user-cards and the security module card, latter of which is part of the actual smart-card interaction module which handles the matching of biometric data. Data transmission need to be secured against interception and eavesdropping at the fingerprint sensor and the smart-card interface. Securing is done through encryption based on the keys agreed upon device authentication.

3.3 Accuracy Issues

Accuracy is a measure with biometric identification and means how often identification is performed correctly. Two metrics are defined in the field of biometrics. False rejection rate (FRR) is a percentage which indicates the portion of identification attempts that are rejected when the attempts were legitimate. False acceptance rate (FAR) is a percentage which indicates the portion of identification attempts that are accepted even though they should have failed. In a security critical application it could be claimed that FAR is more important and must be very close to zero because, if someone can get access just by luck or trying several times, the system is not secure enough. A high FRR value would indicate that even legitimate users have difficulties in getting identified correctly, and this situation would lead to unaccessible applications which is neither desirable.

Iris scans are considered to be very accurate with both low FRR and FAR values. Jain et al. [8] have collected a table of recent studies relating to accuracy of some biometric methods. In 2005 a study of iris-based system were conducted which revealed that the rate values were a little under 1 for both FAR and FRR. Studies from 2003 and 2004 showed that fingerprints achieved a reasonably low FRR of 0.1-2% and FAR of 1-2%. Values this low are good enough for medium security applications, higher-security applications would require rates very close to zero or, a more realistic approach, the use of multi-biometrics, which means using more than one method in combination in order to achieve better accuracy.

There are several aspects that can diminish the accuracy of biometrics. One aspect is that does identification data ever get outdated as it is known that when people get older their physical appearance changes. Jain et al. [8] discuss some reasons for variation in accuracy of biometrics. They present that because biometric data never gets exactly the same form when reproducing the data, which leads to that mistakes can occur in matching previously acquired data and current data. Some reasons for these include accidents that can make a characteristic irreproducible, jewelry and facial hair which can have an impact on hand and facial recognition. In addition dirt, sweat or dry skin can make some biometric methods less accurate.

3.4 Performance Issues

DNA identification would be the most accurate way of identification but, at least for now, no such equipment exist that is powerful enough for real-time identification of DNA [13]. Most of other methods are less power consuming and, thus, are more usable in everyday applications.

Nakanishi and Western [13] argue in their paper that a single biometric identification method may not be sufficient in high security applications. They suggest that multi-biometrics are utilized in that kind of applications. Using multiple methods may hinder the performance of biometric systems but may be required at least as long as any single method is not reliable enough. Multi-biometrics has multiple meanings in this context: several biometric methods; same method with several algorithms, multiple units of the same biometric method; multiple sensors, i.e., using different sensors in sampling the same biometric data. By using multi-biometrics we get a more accurate, but at the same time, more power-consuming identification process.

4 Privacy and Social Problems in Biometrics

This section discusses some problems with biometrics related to human resistance for using individual data for the use of potentially more secure identification and verification. Privacy issues that is present when biometrics are considered and some general ethical and social concerns in using human characteristics for security purposes.

4.1 Awareness and Interest

There have been research about how biometrics has an impact on the behavior of humans [2, 6, 16]. Pons and Polak [16] have conducted a study on computer science students' familiarity on biometrics. Their study consisted of two phases. The first one was a questionnaire of how well the students understand the use of biometrics, how familiar they are with biometric identification methods and how interested they are in using biometrics in identification. The first phase revealed that students knew very little about the subject and were not especially interested in them.

The second part comprised of an educational session where the students were told about the basic aspects of biometrics, and another questionnaire about whether they see any security benefits, privacy concerns or would they be willing to provide biometric information about their selves if a student identification card would be replaced at the university library with biometric identification. The results showed that even computer science students were not familiar with biometrics and not really even interested in using them.

The students' answers concerning privacy, security and willingness to provide biometric data were very close to neutral values, which tells that they did not have any significant opinion either for or against biometrics. Thus, the study clearly reveals that biometrics and related security issues are not known by large audiences, or even a small sample of technologically-aware students.

4.2 Privacy Concerns

In an article from Pons [15] an online marketing approach utilizing biometric technology is evaluated. The main point in the article is that using biometrics as consumer identification instead of an IP-address is more accurate and, therefore, marketing towards individuals could be more precise and useful to customers. They present an idea that users could be identified using a fingerprint scan that is embedded in a mouse that is used as a pointer device with a computer. An obvious disadvantage in this is that consumers may not want to be monitored more accurately because they would feel that the monitoring company would know too much about their preferences and consuming habits. Consumers can also be wary about how the collected information is actually used, they might not fully trust the companies that are responsible for gathering information about their preferences.

In an article from Pons and Polak [16], is suggested that biometrics could be used in universities for person identification instead of student ID cards. This could help making biometrics more comfortable and familiar for students who, hopefully, would become more willing to use biometrics also in other applications in the future. Through this, privacy concerns related to using biometrics is hoped to be mitigated.

According to Liberatore [11] a number of paradoxes exist when discussing privacy issues of biometric technology based surveillance. One of these paradoxes is that increasing more accurate surveillance can mitigate as well as reinforce fears. Those who are part of the surveillance infrastructure may feel that they are more in control of people and, therefore, feel more secure, but at the same time other people may feel that the world is so dangerous that every move people make has to monitored, and that may have a negative impact on the perceived feeling of security. More surveillance can in addition cause the feeling that why everyone needs to be monitored as most people are not criminals, and how the surveillance data is actually used.

4.3 Social Acceptance

Heckle et al. [6] have studied the perception and acceptance of fingerprint-based identification when paying purchases with a credit card in the Internet. The study consisted of certain task needed to be performed, all of them included ordering books from an online book store. There was no actual fingerprint matching happening but the fingerprint reader was authentic. The participants were told that they should play along and consider using the fingerprint reader as a part of a fully functional authentication system. The study revealed that users perceive fingerprint scanning as a viable and beneficial method for identification. However, users did not have very clear idea of how secure fingerprints actually were when compared to conventional user-names and passwords. The participants in the study also indicated their concerns related to privacy issues, some of them felt that biometrics is an invasive method and their usage reaches too personal levels.

Riley et al. [17] have studied using fingerprints and hand veins among the elderly for identification means. Their goal was to investigate how well fingerprints perform among older people, because it has been suggested that fingerprints' accuracy deteriorates when users gets old enough. Another goals was to study do older people see biometric technology as an acceptable method for identifying individuals. The results showed that vein-based technology was preferred to fingerprints overall. There was no significant difference between the performance or average verification time of the methods but enrollment was faster and more successful with the hand vein method. This implies that previous research may have correct results as the enrollment of fingerprints failed with a third of the participants. Vein-based method was additionally preferred by the elderly in every aspect, for instance it was considered easier, faster and less stressful to use. Overall there was no actual resistance towards biometrics but some of the participants would have liked to have more information about the subject.

5 Conclusion

Biometric identification methods have traditionally been in government, military and criminal investigation use but they are becoming more easily deployable, more accurate and more affordable as the technology has improved at fast pace. All of these have made it possible to start their usage with general audiences in wider scale.

Several different methods exist and some of them are suitable for certain applications while other are better for other applications. Most common method is fingerprint scanning and facial images. The latter of these can be used without interaction with the user being identified. This makes it possible to use it in improving security, for instance, in a public event where authorities can monitor the audience and try to find known criminals.

Ideal biometrics has certain properties and also certain restrictions, both of which are important when designing real-world applications. The application determines which method is most suitable as all the methods have pros and cons of their own. Some are more suitable in low-security applications, while others can also be utilized in security critical areas. Iris and retinal scans are the most accurate methods and also have good performance levels, which make them ideal for higher-security applications. Fingerprints are also quite accurate but demand less expensive equipment which makes them very cost efficient solution, for instance in laptops.

Several privacy and social acceptance issues still exist related to the use of biometrics. These issues are very hard to overcome as biometrics intrinsically is an invasive identification method. People should be educated that the invasion brings more advantages than disadvantages. Security is increased as identification is more accurate, and it is harder to gain access if you are not authorized are some advantages while disadvantages would include being potentially more easily monitored and the mistrust towards the parties that control biometric data, that is, whether they use the data only for agreed purposes.

References

[1] H. Ailisto, M. Lindholm, S.-M. Mäkelä, and E. Vildjiounaite. Unobtrusive user identification with light biometrics. In *NordiCHI '04: Proceedings of the third Nordic conference on Human-computer interaction*, pages 327–330, New York, NY, USA, 2004. ACM.

[2] A. Alterman. "A Piece Of Yourself": Ethical Issues In Biometric Identification. *Ethics and Inf. Technol.*, 5(3):139–150, 2003.

[3] R. P. Broussard, L. R. Kennell, R. W. Ives, and R. N. Rakvic. An artificial neural network based matching metric for iris identification. volume 6812. SPIE, 2008.

[4] Y. Bulatov, S. Jambawalikary, P. Kumarz, and S. Sethiay. Hand recognition using geometric classifiers. In *DIMACS Workshop on Computational Geometry, Rutgers University, Piscataway, NJ*, pages 14–15.

[5] M. Ferrer, A. Morales, C. Travieso, and J. Alonso. Low cost multimodal biometric identification system based on hand geometry, palm and finger print texture. *41st Annual IEEE International Carnahan Conference on Security Technology, 2007*, pages 52–58, Oct. 2007.

[6] R. R. Heckle, A. S. Patrick, and A. Ozok. Perception and acceptance of fingerprint biometric technology. In *SOUPS '07: Proceedings of the 3rd symposium on Usable privacy and security*, pages 153–154, New York, NY, USA, 2007. ACM.

[7] A. Jain, L. Hong, and S. Pankanti. Biometric identification. *Commun. ACM*, 43(2):90–98, 2000.

[8] A. Jain, A. Ross, and S. Pankanti. Biometrics: A tool for information security. *IEEE Transactions on Information Forensics and Security*, 1(2):125–143, June 2006.

[9] A. Jain, A. Ross, and S. Prabhakar. An introduction to biometric recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(1):4–20, Jan. 2004.

[10] L. A. Jones, A. I. Antón, and J. B. Earp. Towards understanding user perceptions of authentication technologies. In *WPES '07: Proceedings of the 2007 ACM workshop on Privacy in electronic society*, pages 91–98, New York, NY, USA, 2007. ACM.

[11] A. Liberatore. Balancing security and democracy: The politics of biometric identification in the european union. EUI-RSCAS Working Papers 30, European University Institute (EUI), Robert Schuman Centre of Advanced Studies (RSCAS), Oct. 2005.

[12] D. Mulyono and H. S. Jinn. A study of finger vein biometric for personal identification. *International Symposium on Biometrics and Security Technologies, 2008. ISBAST 2008.*, pages 1–8, April 2008.

[13] Y. Nakanishi and J. Western. Advancing the state-of-the-art in transportation security identification and verification technologies: Biometric and multibiometric systems. *Intelligent Transportation Systems Conference, 2007. ITSC 2007. IEEE*, pages 1004–1009, 30 2007-Oct. 3 2007.

[14] P. Padma Polash and M. Maruf Monwar. Human iris recognition for biometric identification. *10th international conference on Computer and information technology, 2008. ICCIT 2007.*, pages 1–5, Dec. 2007.

[15] A. P. Pons. Biometric marketing: targeting the online consumer. *Commun. ACM*, 49(8):60–66, 2006.

[16] A. P. Pons and P. Polak. Understanding user perspectives on biometric technology. *Commun. ACM*, 51(9):115–118, 2008.

[17] C. Riley, H. McCracken, and K. Buckner. Fingers, veins and the grey pound: accessibility of biometric technology. In *ECCE '07: Proceedings of the 14th European conference on Cognitive ergonomics*, pages 149–152, New York, NY, USA, 2007. ACM.

[18] D. C. Shin, J. Kim, and B. Noh. Trends of biometric and test techniques of k-nbtc. *The 9th International Conference on Advanced Communication Technology*, 1:141–146, Feb. 2007.

[19] U. Waldmann, D. Scheuermann, and C. Eckert. Protected transmission of biometric user authentication data for oncard-matching. In *SAC '04: Proceedings of the 2004 ACM symposium on Applied computing*, pages 425–430, New York, NY, USA, 2004. ACM.

Security Analysis of DTN Mail Application

Shengye Lu
Helsinki University of Technology
slu@cc.hut.fi

Abstract

Delay-Tolerant Network (DTN) allows mail application work in challenged network environment. DTN network is a message-based overlay network, which uses a store-carry-and-forward approach. However, the novel transmission approach poses new challenges to security of communicating applications. This paper analyzes the security issues of a DTN mail application. We perform on-wire eavesdropping at different locations of DTN Mail application architecture, and then analyze the security of DTN Mail application on the basis of testing result.

KEYWORDS: DTN Mail application, privacy, traffic eavesdropping

1 Introduction

In the situation where no infrastructure is available, or mobile networking is carried out over unstable or opportunistic contacts, many traditional applications based on TCP/IP protocol stack do not work, because they rely on the assumption of stable underlying networking.

To make communication in this situation possible, the Delay Tolerant Network Research Group (DTNRG) of Internet Research Task Force (IRTF) has been developing a message-based overlay network [1], called Delay Tolerant Network (DTN). It uses a store-carry-and-forward method to provide reliable transmission. DTN network offers a solution to communicate in a challenged network environment. Instead of using TCP/IP, application transfer data by using Bundle Protocol(BP) [7]. Using Bundle Protocol, an application can encapsulate self-contained application data unit into separate DTN bundles, which does not need end-to-end path between communicating parties [5]. In DTNRG, these applications are called DTN applications.

The new transmission mechanism introduces new challenges for DTN application security. The newly emerged security problems are very different from those of traditional applications, which use TCP/IP for transmission. In a challenged network environment, a connection or an opportunistic contact is a precious resource. In order to make full use of the connection resources, the DTN applications need to encapsulate application data into self-contained DTN bundles rather than fragment them into separate packets. Therefore, if a DTN bundle is eavesdropped, a considerable amount of privacy in the application data will be exposed to an attacker.

On the other hand, security of the lower layer networking protocols which DTN application uses also have large effect on DTN application security. As we know, many DTN applications are designed to work in unstable and mobile networking environments, for example, Bluetooth or Mobile Ad-hoc Networks (MANET). In mobile environments, computers easily leak privacy information to the local access link when they attempt to get network service outside their home domain [10]. The compromised privacy are often leaked in many layers of the protocol stack.

All this makes the security of DTN application a complex problem. This paper analyzes the security issues of DTN Mail application. We use an existing implementation of DTN Mail application by Hyyryläinen et al. [9] and perform on-wire eavesdropping tests at different locations. Based on traffic analysis, we look into the security issues of DTN mail application and discuss possible solutions. Section 2 introduces the DTN Mail application. Section 3 explains its architecture and our testing environment. Eavesdropping test results and DTN Mail application security are analyzed in section 4, followed by conclusion in section 5.

2 Background

In a classical email application architecture, a sender uses Simple Mail Transfer Protocol (SMTP) to send out email. Mail destined to a receiver is delivered to the mailbox by SMTP protocol and picked up by an email client via Post Office Protocol (POP) or Internet Message Access Protocol (IMAP). Usually SMTP and POP/IMAP are operated over TCP/IP.

Obviously this solution does not work in challenged network, where the end-to-end path between communicating parties might not exist, or suffer from long delays, or only have opportunistic connections with frequent disruption. DTN overlay network proposed by IRTF DTNRG provides one solution for communication in the challenged networks.

Based on DTN overlay network, Hyyryläiinen et al. [9] developed a DTN Mail Application, which allows a mobile end-user to receive and send email while roaming in a challenged network. The general idea of the DTN Mail application is: sender's DTN enabled device encapsulates RFC 2822 [6] compliant messages into DTN bundles. Then, DTN Bundle protocol and convergence layers transmit the bundles across a challenged network. Receiver's DTN enabled device receives the bundle from the DTN overlay network, recovers it into the original email message, and delivers it further.

3 Security of DTN Mail Application

In order to explore the security problems of DTN application, especially DTN Mail application, which has more requirements for security and privacy, we perform eavesdropping tests based on DTN Mail application. We use the DTN Mail implementation provided by our DTN research group [9] and DTN router (i.e., DTNd) from DTN reference implementation [3].

3.1 Architecture of DTN Mail Application

The architecture of the DTN Mail application is illustrated in Figure 1. The components in our DTN Mail application tests include:

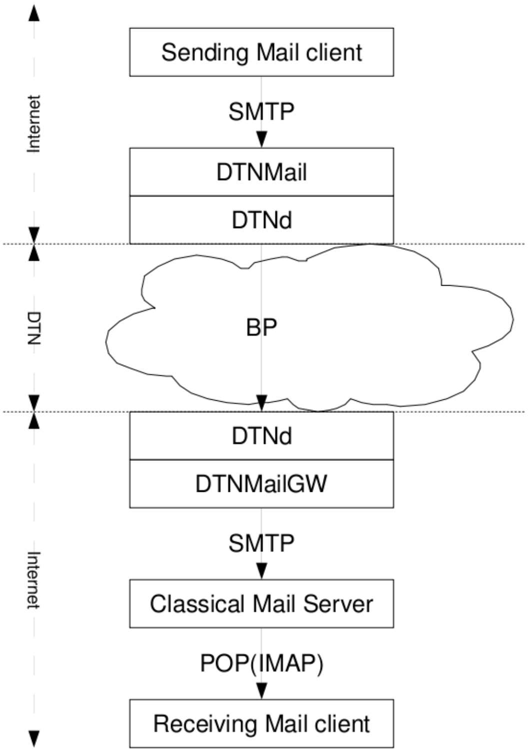

Figure 1: Testing environment.

- Sending mail client and Receiving mail client are just normal email clients, for example, Firefox Thunderbird.
- DTNMail node works as an interface between the normal Internet and a challenged network. It contains DTNMail component and a DTN router component.

  DTNMail exchanges SMTP packets with Sending mail client. DTNMail uses Open Network Computing Remote Procedure Call (ONC RPC) to cooperate with the DTN router. DTN router accepts and transfers DTN bundles with other DTN routers, and all traffic going in or out from a DTN network must pass through it.
- DTNMailGW node consists of DTNMail gateway (i.e., DTNMailGW) and one DTN router.

  Just as the DTNMail node, DTNMailGW node is an interface between the normal Internet and a challenged network. DTN router component exchanges bundles with other DTN routers. DTNMailGW component exchanges SMTP packets with traditional mail servers.
- Classical Mail server is just a traditional mail server located in the normal Internet. In this testing, we use the mail server of our department.

In this architecture, sending mail client and receiving mail client are in the normal Internet environment. The network between the DTNMailGW node and the DTNMail node is a challenged network. In this network, DTN routers form a DTN overlay network, using the bundle protocol to transmit data.

The transactions in the DTN Mail application follow these steps:

1. Sending mail client sends email to DTNMail node by using SMTP.
2. DTNMail node encapsulates received SMTP messages into the DTN bundles (BP). Then, bundles enter DTN overlay network via a DTN router and travel in the challenged network.
3. The bundles will finally be delivered to the receiver when the receiver's DTN-enabled devices (i.e., DTNMailGW in Figure 1) makes contact with some DTN routers, which are carrying the mail bundles.
4. The receiver's DTN-enabled device recovers the mail bundles to RFC 2822 compliant messages and forwards them further via SMTP to receiver's traditional email server.
5. Finally, the receiver picks up the email over POP or IMAP using an email client connected to a traditional email server.

3.2 Security Mechanisms for DTN Mail

In order to provide confidentiality and integrity for email, encryption and digital signature can be used in DTN Mail application. In Figure 1, when DTNMail receives email messages in Multipurpose Internet Mail Extensions (MIME) format from an email client, it will encrypt it using public key, and optionally signs it before encapsulating it into DTN bundle. DTNMail's public key is in the form of an X.509 certificate, which is signed by the DTNMailGW. DTNMailGW acts as a certificate authority in this architecture. When a mail bundle finally arrives at DTNMailGW, DTNMailGW decrypts the message and verifies the user certificate whether it is signed by DTNMailGW or not. Only after a successful verification, the DTNMailGW will relay the original message over the Internet to a traditional mail server.

3.3 Traffic Eavesdropping Test

Be design, DTN Mail application encapsulates entire email headers, body, and all its attachments in one DTN bundle [9]. An attacker can eavesdrop the DTN bundle either on wire or over air, then use a packet analyzer to interpret application data from the DTN bundle. As a result, privacy of application data is heavily violated.

For the experimental part of this paper, we use a traffic analyzer tool, called Wireshark [1], to capture traffic in the DTN Mail application architecture, and then interpret messages from the captured traffic. Our objective is to explore the vulnerability of DTN Mail application security in the case of an eavesdropping attack. Figure 2 shows the eavesdropping locations, where we use Wireshark to capture traffic in our tests. In order to compare an email bundle with the original email messages, we capture SMTP packets between sending email client and DTNMail. From the viewpoint of an email client, DTNMail serves as a normal SMTP server. Besides of that, We capture DTN bundles from the link between the DTN router in DTNMail node and the DTN router in DTNMailGW node, trying to interpret original email messages from the DTN email bundle.

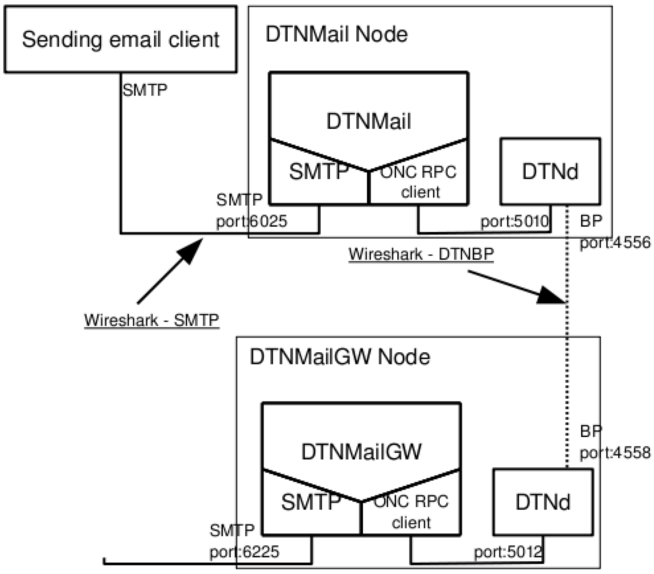

Figure 2: Eavesdropping points in the architecture.

In our testing, DTN routers use TCP convergence layer [2] to transfer DTN bundles with flooding based routing protocol in the DTN bundle layer. We capture traffic in a situation where the encryption mechanism in section 3.2 is not in use, as well as in a situation where the encryption mechanism is used.

4 Analysis of the Result

In this section, we use Wireshark traffic analyzer to interpret messages from the eavesdropped traffic. Based on the results, we look into the security issue of DTN Mail application.

4.1 Eavesdropping Tests Result

Figure 3 shows a TCP stream analysis result based on SMTP packets captured from the link between DTNMail and sending email client. As shown in Figure 3, Wireshark successfully interprets SMTP messages and email messages. Many details are revealed, including the email, sender's address and receiver's address.

```
220 Welcome to DTN Mail Daemon's SMTP Server
EHLO [127.0.0.1]
500 Command Unrecognized:
HELO [127.0.0.1]
250 OK
MAIL FROM:<lushengyehut@gmail.com>
250 OK
RCPT TO:<slu@netlab.tkk.fi>
250 OK
DATA
354 Start mail input; end with <CRLF>.<CRLF>
Message-ID: <491DB092.2020701@gmail.com>
Date: Fri, 14 Nov 2008 19:08:34 +0200
From: slu <lushengyehut@gmail.com>
User-Agent: Thunderbird 2.0.0.17 (X11/20080925)
MIME-Version: 1.0
To: slu@netlab.tkk.fi
Subject: Hello
Content-Type: text/plain; charset=ISO-8859-1; format=flowed
Content-Transfer-Encoding: 7bit

hi,

It was nice to meet you! BTW, do you like Italian food?
.
250 OK
QUIT
221 Server closing transmission channel
250 OK
```

Figure 3: TCP stream analysis of the SMTP transactions.

Figure 4 depicts the TCP stream analysis of the DTN bundle protocol transactions eavesdropped between the DTN router in the DTNMail node and the DTN router in the DTNMailGW node. Wireshark is used to decode the application layer data as ASCII characters.

```
......R.......;........X@dtn.//slu@netlab.tkk.fi/mailto
.//pc20.netlab.tkk.fi/mailto.none....>Received: from [1
27.0.0.1] by [127.0.0.1]; Fri, 14 Nov 2008 19:08:34 +02
00
Message-ID: <491DB092.2020701@gmail.com>
Date: Fri, 14 Nov 2008 19:08:34 +0200
From: slu <lushengyehut@gmail.com>
User-Agent: Thunderbird 2.0.0.17 (X11/20080925)
MIME-Version: 1.0
To: slu@netlab.tkk.fi
Subject: Hello
Content-Type: text/plain; charset=ISO-8859-1; format=flowed
Content-Transfer-Encoding: 7bit

hi,

It was nice to meet you! BTW, do you like Italian food?
 ..@@@@
```

Figure 4: TCP stream analysis of the DTN BP transactions.

As demonstrated in Figure 4, by eavesdropping DTN bundle transactions, an attacker can discover email headers, email body and Endpoint IDs (i.e., EID) of the DTN routers.

We perform the same testing when the encryption mechanism of DTN Mail is enabled, as described in section 3.2. Figure 5 shows the result of the TCP stream analysis of DTN bundle transactions.

From Figure 5, we can find that DTNMail protects the privacy of email by encrypting and signing the email body (i.e., using S/MIME). Attackers can use packet analyzer to interpret email headers and EIDs of DTN routers from DTN

[1] http://www.wireshark.org/

```
@@......V.......?....P./.XDdtn.//slu@netlab.tkk.fi/mail
to.//slu@pc20.netlab.tkk.fi/mailto.none.....Message-ID
: <3686501.1226686288398.JavaMail.slu@lenovo-jo>
Mime-Version: 1.0
Content-Type: application/pkcs7-mime; name="smime.p7m";
smime-type=enveloped-data
Content-Transfer-Encoding: base64
Received: from [127.0.0.1] by [127.0.0.1]; Fri, 14 Nov
 2008 20:11:27 +0200
Date: Fri, 14 Nov 2008 20:11:26 +0200
From: slu <lushengyehut@gmail.com>
User-Agent: Thunderbird 2.0.0.17 (X11/20080925)
To: slu@netlab.tkk.fi
Subject: hello

MIAGCSqGSIb3DQEHA6CAMIACAQAxgeEwgd4CAQAwRzA6MQswCQYDVQQ
GEwJBVTEMMAoGA1UECBMDdGtrMQwwCgYDVQQKEwN0a2sxDzANBgNVBA
MTBm5ldGxhYgIJAKB+IfFuSNykMA0GCSqGSIb3DQEBAQUABIGABRd0d
llSEMlfMpiOXwomDpqXkQUAK3Rkja/lOIDQY7zwn7LVjCoCm2BxvIGA
Y0F/mtf57JE1JnIM6Y/E60tQl5xQY4HO+yFU+qIOO29T0vEW/9RTC4t
v2DIZs5g9OHPZpaIfkmJ2McFAYdvvpTnw0FVW9GPtxCf57Fll5oUedQ
0wgAYJKoZIhvcNAQcBMBkGCCqGSIb3DQMCMA0CAToECC/Ka95A

...
```

Figure 5: TCP stream analysis of the DTN BP transactions (with encryption).

bundles, but they can not discover the content of the email body.

4.2 Confidentiality and Integrity

From Figure 4, we can find that DTN bundle protocol does not provide any confidentiality or integrity protection for application data. Application data are encapsulated as payload blocks into DTN bundles without any modification. In the case of DTN Mail Application, there are three options to improve the confidentiality and integrity security for application data.

Firstly, as the security mechanism adopted in section 3.2, before application data are encapsulated into a DTN bundle and enter DTN overlay network, they are partly or wholly encrypted by a DTN application proxy. In our DTN Mail application, the DTN application proxy is DTNMail or DTNMailGW. DTNMail intercepts SMTP messages from an email client, and performs S/MIME encryption and signing operation on the email body. When the DTN bundle finally leave DTN overlay network, they are decrypted by the DTN application proxy. In our case, DTNMailGW performs S/MIME decryption and verification signature operation. Using this method, application data obtain confidentiality and integrity security services while they are traveling across DTN overlay network. Attackers can discover some DTN bundle layer routing information from eavesdropped traffic, for example, in Figure 5, sender endpoint ID and receiver endpoint ID are plaintext.

Secondly, DTN Mail Application can adopt end to end security. In other words, sending email client and receiver email client provide S/MIME services before email starts traveling via SMTP and after it arrives to the destination. In this way, email body are safe during the whole path. This option has the same problem as the previous one: eavesdropper can interpret the sender endpoint ID and the receiver endpoint ID information. One thing worth of mentioning is that email headers can not be encrypted in this case, because they are needed by SMTP to do application layer routing.

Thirdly, Bundle Security Protocol [8] can provide hop by hop security when the DTN bundles are transmitted within the DTN overlay network. Unlike the first option, this option provides a comprehensive security solution at the bundle layer, hence, it is not subject to the specific DTN application proxy. Bundle Security Protocol defines some bundle blocks to provide confidentiality and integrity service for payload by the use of encryption mechanism. However, because the DTN bundle layer uses endpoint addresses to perform routing, the confidentiality for the source or destination endpoint addresses, or any other endpoints in bundle blocks can not be protected [4]. One solution to this issue is using bundle-in-bundle encapsulation (BiB). Unfortunately, currently there is no state-of-art implementation of Bundle Security Protocol yet.

4.3 Denial of Service

The certificate verification mechanism of section 3.2 protects classical Mail Server from Denial of Service (DoS) attacks at the application layer. The reason is that DTNMailGW verifies user certificate before it relays SMTP messages to classical Mail Server. If the verification fails, it will drop the packets.

DoS attacks may happen at any layer. At the bundle layer, DTN nodes use limited storage resource to store the carried bundles. Their transactions with each other usually involve long latencies. These facts make DoS attack more effective. Bundle Security Protocol tries to solve this problem by using hop by hop verification mechanisms. But the mechanisms themselves also consume many resources in DTN overlay network.

4.4 Leaking Privacy from Lower Layer

DTNs utilize convergence layer for transmission, e.g. TCP convergence layer, Bluetooth convergence layer, or ethernet convergence layer. Similarly, a convergence layer utilizes the underlying layer where it resides on to implement exact transmissions. So the underlying layers might leak some user privacy as well.

For example, in WLAN Ad-hoc environment: since no server provides networking configuration services, DTN nodes have to use zeroconf or service discovery to form an opportunistic link dynamically. In other words, node should announce itself and discover peers in order to dynamically establish contact with other DTN-enabled WLAN Ad-hoc nodes. During the process of announcement and discovery, they will leak many private details, for example, IP addresses, or application related information.

In our testing, Wireshark did not detect privacy leak related with service discovery, because we are not using a dynamically formed opportunistic contact, instead we are using a pre-configured link. However, this is not often the case in real world.

5 Conclusions

The seminar paper looks into the security problems of an DTN application, by using DTN Mail application as a real world example. The testing results indicate that the bundle protocol does not protect the privacy of application data. There are several possible solutions to provide DTN Mail application security.

On application layer, DTN application proxy uses S/MIME to protect email body. The paper shows the eavesdropping test result for this scenario. S/MIME can also be used in the email client, and this provides an end to end solution for communication security.

On bundle layer, we can extend the bundle protocol to support confidentiality, integrity and DoS-resistance features within the DTN overlay network. Bundle layer security is currently a hot research topic, because many issues about it remain open. Because of the unstable connection characteristic of a challenged network, many existing security solutions and protocols can not be adopted in a DTN overlay network. For example, TLS needs several transactions for exchanging encryption parameters; in DTN environment however, several transactions might result in unacceptable latency. Besides, most security mechanism are based on encryption, for which a key management mechanism is needed. However, currently there is no delay-tolerant method for key management yet.

References

[1] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss. Delay-Tolerant Networking architecture. RFC 4838, DTN Research Group, April 2007.

[2] M. Demmer and J. Ott. TCP Convergence Layer. Draft, DTN Research Group, 2006.

[3] M. D. et al. DTN reference implementation. Technical report, DTN Research Group, 2006.

[4] S. Farrell, S. Symington, H. Weiss, and P. Lovell. Delay-Tolerant Networking Security Overview. Draft, DTN Research Group, 2007.

[5] J. Ott, T. Kärkkäinen, and M. Pitkänen. Application Conventions for Bundle-based Communications. Draft, DTN Research Group, 2007.

[6] P.Resnick. Internet message format. RFC 2822, DTN Research Group, August 1982.

[7] K. Scott and S. Burleigh. Bundle Protocol Specification. Draft, DTN Research Group, 2007.

[8] S. Symington, S. Farrell, H. Weiss, and P. Lovell. Bundle Security Protocol Specification. Draft, DTN Research Group, 2007.

[9] T. Hyyryläinen and T. Kärkkäinen and C. Luo and V. Jaspertas and J. Karvo and J. Ott. Opportunistic email distribution and access in challenged heterogeneous environments. In *Proceedings of the second ACM workshop on Challenged networks*, pages 97 – 100, 2007.

[10] Tuomas Aura and Janne Lindqvist and Michael Roe and Anish Mohammed. Chattering Laptops. In *proceedings of the 8th Privacy Enhancing Technologies Symposium (PETS)*.

Trust in Operations Security of Computational Grids

Urpo Kaila
Helsinki University of Technology
ukk@cc.hut.fi

Abstract

Grid infrastructures for scientific computing differs from traditional computing services because of an architecture based on middleware and administration by the use of virtual organisations. Moreover,some concerns have been expressed on the security of computing grids. Grid infrastructures might cause new information security risks in addition to existing risks related to all IT systems.

Measures for operations security such as acceptable use policies and user security guidelines are one fundamental prerequisite for creating trust in computing environments - in addition to secure system architecture and system administration.

This paper surveys operations security documentation for end users in three computational grid infrastructures in the Nordic Countries. The survey is limited to such security guidelines for end-users, which are publicly available on the Grid provider web site. The survey shows that few security guidelines or policies for acceptable use were directly available and the policies were inconsistent and quite different when compared with each other. Our contribution is to show that it would be in the interest of the grid principals to synchronise security policies and define a lowest common level for security.

KEYWORDS: Computational Grid, Trust, Operations Security, M-grid, SweGrid, Norgrid

1 Introduction

Advanced research in science, engineering and, also in some formalized and quantitative variants of social sciences requires robust and reliable ITC services. Computational research has replaced and complemented tedious laboratory tests because of the efficiency and repeatability of simulations. Moreover, researchers can often do their experiments much faster and more extensively *in silico* (with the help of computer based modelling and computational science) instead of *in vitro* (through physical experiments in laboratories or by field testing).

As governments see research and development as a mean for increasing economic growth and the competitive edge of their nations, significant investments for computational services in national research and education networks has been made worldwide. Instead of dedicated and self-managed computing servers, researchers have increasingly favoured grid computing as a more flexible and powerful research infrastructure, often managed and operated by themselves. The basic operational principles of computational grids will be presented in the next section.

The security of the computational services can be threatened by, among others, errors, misuse, or malicious intruders [18] . Some dramatic intrusions and root compromises in some grid infrastructures raised the question whether all reasonable measures are performed to ensure system integrity and service availability.

Proper technical and administrative security measures to cope with identified risks should be taken to avoid service breaks due to the reasons described above. According to the basic principles of information security, risks should be mitigated by administrative, logical or physical security controls, such as policies, access control and monitoring [14]. Security controls should, as commonly suggested,define acceptable use, inform users and other parties how to comply with guidelines, and also technically prevent or mitigate unwanted behaviour. Due care and diligence requires proactive and systematic measures to mitigate information security related risks.

The purpose of this paper is to show how trust is managed in grid computing and who is responsible for security. What are the measures to ensure the security of the systems providing computational services to the researchers? How can adequate trust in the services provided be assured ? Despite the many improvements in technical security for grid infrastructures compared with traditional computing servers we will focus on the management side of security in this paper.

The current paper will limit itself to the **operations security** of grid computing and exclude questions related for example to architecture and protocol design. There are several overlapping information security domains, such as access control, cryptography, and operations security. Operations security is used to identify and define security controls. One could say that operations security is the most obvious and visible set of security measures. Operations security involves the `administrative management` of information processing operations, the concepts of security for operations controls, resource protection, auditing, monitoring, and intrusion detection and prevention [14].

Moreover, this study compares publicly available security guidelines for end users of a grid service. Security guidelines for system administrators, as well as technical implementations of security architecture and security monitoring, will be out of the scope of this paper despite their importance for maintaining trust and security. We will survey and compare how operations security is implemented on some infrastructures for grid computing in the Nordic Countries.

The following grid infrastructures will be surveyed for user security guidelines:

- The Finnish M-grid
- The Swedish SweGrid
- The Norwegian NorGrid

To limit the scope of the paper, additional computational grids have not been included.

In Section 2, we introduce an outline of the architecture and functions of grid infrastructures. Section 3 describes some security threats related to grid infrastructures. After defining methods and material for the study in section 4, a case study of three computational grids in the Nordic countries is presented in section 5. The paper will present results in section 6 and in some discussions in section 7 on how the operations security of computational grids should be improved, based on the findings in the comparisons. Also, possibilities for further research will be discussed. Finally, a short conclusion in section 9 will summarise the paper.

2 Computational Grid Infrastructures

The benefits and potentials of grid computing has already been presented in 1998 in a classical paper by Foster et al. [13] :

> "A computational grid is a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities."

Compared with traditional services for scientific computing grid computing adds the concepts of middleware for enabling the use of distributed systems and virtual organisations for distributed administration.

Foster et al. [12] have compared principal differences between grid computing compared with other systems. According to the comparison, computing resources are not administered centrally in grid computing, open standards are used, and a non-trivial quality of service is achieved.

From an individual researchers point of view, grid computing is an available network service enabling computational tasks which are not possible or feasible, due to reasons related to performance to run on local workstations or servers.

Grid computing has evolved as a more flexible middle range alternative to centralized scientific computing on supercomputers. Scientific computing has long traditions in hard sciences as physics and chemistry but other branches of research have also noted the benefits of simulation by computing instead of tedious testing and experiments.

Several technologies and technical implementations for computation grids coexist. In the Nordic Countries, researchers have cooperated with National Research and Education Networks through the NorduGrid Collaboration to create and support the Advanced Resource Connector (ARC), also refered to as the NorduGrid middleware. ARC is based [11] on several Open Source solutions like Globus Toolkit®for (R) (GT) libraries, Grid Security Infrastructure (GSI), OpenLDAP, OpenSSL, and Simple Authentication and Security Layer (SASL) [16] Grid jobs are submitted to computing clusters via the gridftp protocol.

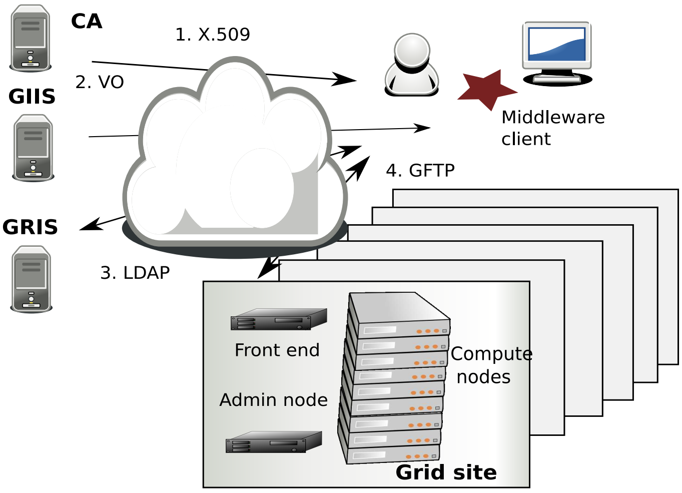

Figure 1: Basic topology in ARC based Grids

Fig. 1 illustrates, from the user point of view, the basic use case in an ARC based grid environment consists of roughly of the following stages:

- User receives a X.509 certificate from a trusted Certificate Authority
- The user is joined to a Virtual Organisation
- When submitting a grid job a LDAP query on available resources is made to the Grid Index Information Service (GIIS) and Grid Resource Information Service (GRIS)
- The actual grid job and related data is sent to the computing nodes over gridftp (GFTP) through authentication and authorisation in the front end node.

3 Security Threats related to Grid Infrastructures

Grid computing seems to include not only all the classical information security risks, but also new risks related to grid technologies and the innovative organisational structures of grid services[8].In addition, there have been some security concerns related to grid services due to it's nature of new technology operated by persons outside the sphere of the traditional systems administrators of scientific computing. However, compared with traditional scientific computing, Grid technology and protocols include many extraordinary proactive security controls to ensure safe operations in open environments [23] .

The greatest security challenges for computational grids are perhaps after all not based on improper technical security controls but on issues related to operations security, that is, how to motivate, educate and oblige all users and administrators to ensure security. Despite a generic well known principle to automate security through technical security controls in protocols and services, few technical controls can totally ensure trust if users and administrators threats the system from inside.

For computational grid infrastructures the operations security challenges are accentuated due the cross-organisational management by virtual organisations. Hence, computational grids must in addition to implement best practices in technical security controls, also implement operations security measures to ensure the lowest common security nominator.

4 Methods and materials

The researcher as the typical user of a computational grid is, or at least should be, concerned about how the computing environment implements the core objectives of information security, confidentiality, integrity, and availability [14]. Errors, malfunctions, service breaks, deviations, and incidents can endanger the schedule or even the results of the research. Also, research data and some of the results might be confidential and for this reasont o be protected against unauthorised access. The requirement of confidentiality might result from the competitive nature of the research or reasons related to compliance, for example privacy laws which apply for the research data.

From the researchers point of view it is a question of trust. As pointed out in [10],the interest of the end-user is not only related to the technical end-to-end reliability of the services but also the trust and trustworthiness that the services are operated in accordance with the interest of the end-user. We will show how providing trust is implemented in grid computing by means of operations security measures as manifested in user related security policies and guidelines.

In grid computing, as pointed out in [8], there might be even more stakeholders than in commercial internet services. The roles providing and supporting the service can include local grid site system administration, grid federation, local (university) IT Computing Centres providing Data Centre services, local or national Computer Security Incident Response Team (CSIRT) and the National Research and Education Network (NREN) providing network carrier services.

We studied the publicly available documentation of operations security guidelines in some computational grid infrastructures to see how trust is defined and described in the security policies and in the security guidelines.

In this initial and exploratory study, we aim at clarifying whether the security related rights and responsibilities for different stakeholders are defined in the security guidelines. The research questions for each grid infrastructure are:

Q1: Does the service require the end-user to follow an acceptable use policy (AUP)?

Q2: Does the AUP include obligations for the end-user?

Q3: Is the grid service committed to provide a minimum service or availability level?

Q4: Does the security policy include sanctions such as closing of an account at risk for not complying with the AUP?

Q5: Is the AUP supported by security guidelines for the end-user on how to practically implement the security measures?

In this paper we will only perform a short survey based on publicly and trivially available information on the grid web site. No interview of helpdesk, customer service or administration will be done, although much more knowledge and understanding could be achieved using such more time-consuming methods.

5 Case Study

5.1 The Finnish M-grid

Material Sciences National Grid Infrastructure (M-grid) is a joint project between CSC, seven universities and The Helsinki Institute of Physics (HIP). The project consists of building a computer network in which computers are situated at different locations but are in shared use through grid technology. The systems are connected to the Nordic NorduGrid network, but access is currently limited to M-grid partners and CSC customers [4].

Q1: Yes, there is an AUP [1], but it is not announced on the web pages of the service, The AUP can be found from https://extras.csc.fi/mgrid/sec/, but also there the link is broken and requires some extra searching.

Q2: Yes, The AUP include several obligations for the end-user, among other:

- The M-grid services and systems are intended for professional, academic research or education. - Your account is personal and may not be shared with other people. - You must protect your account and private keys with good passwords. - You must respect privacy and confidentiality of other users' files and data.

Q3: Yes and no. In the security policy if M-grid [2] it is stated that the site will ensure high availability of the system, and delegating responsibilities to administrators. Also, it is said, 'management has approved, supports and will enforce this policy'. Guaranteed availability levels are although not announced.

Q4: Both. Current AUP and Security Policy is mostly a recommendation although it is stated that Accounts of users who are not affiliated with the site any more should be removed or disabled within two weeks.

Q5: Yes, there is an additional M-grid User Security Guideline [3]

5.2 The Swedish SweGrid

Swegrid is a Swedish national computational resource, consisting of over 3000 cores spread out over 6 clusters at 6 different sites across Sweden. The sites are connected through the high-performance GigaSunet network [7].

Q1: Yes and no. According to the site web pages Researchers linked to any Swedish academic institution may apply for (computing) time on SweGrid resources but there is no single Acceptable us policy available on the Swegrid web pages. According to the web pages applications should be directed to SNAC (Swedish National Allocations Committee) [6], which in turn directs researchers linked to any Swedish academic institution to apply for time on HPC resources at the following six national centres in Sweden: HPC2N, PDC, NSC, UPPMAX, C3SE, or LUNARC.

There is no single common acceptable use policy for all Swegrid users but the centres have somewhat compatible but heterogeneous policies of their own small scale projects:

- HPC2N, Rules for using HPC2N resources and services [17],
- PDC, Rules for computer, network and system facilities [15],
- NSC, Rules for NSC user account holders [5]
- UPPMAX, Rules for NSC user account holders [22]
- C3SE, No online acceptable use poliy was found
- LUNARC, Lunarc Rules Concerning Accounts [9]

The procedures for applying for computing time differs depending on the size of the project, other procedures applies for medium and large scale projects

Q2: Yes and no.

The different local policies do include some obligations for the end-users, for example:

ˋExploitation of defective configurations, program errors or any other method to secure a higher level of privilege then authorised is prohibitedˊ ref17

There are great variations between the the policies in scope and in levels of obligations.

Q3: No. Statements of guaranteed availability was not found

Q4: Yes and no.

For example the NSC rules state that breach of the rules may result in account termination and/or legal prosecution.

Q5: Yes. Several security guidelines for users were found, although they varied much in scope and depth.

5.3 The Norwegian NorGrid

NorGrid [19] develop and maintains national grid infrastructure that provides easy and secure access to distributed resources, provides large aggregate capacities for computation, storage and data transfer, optimizes the utilization of the overall resource capacity, and make Norway an attractive partner in international grid collaborations.

Grid security procedures for users are described in the NOTUR[20] guide for users [21]..

Q1: Yes. All Nordgrid partners have some AUPs.

Q2: Yes. The partner AUPs include some strictly worded obligations.

Q3: No. Documents describing guaranteed or achieved service availability was not found.

Q4: Yes. Sanctions were described.

Q5: Yes. There were some advice in the guidelines.

Question	Mgrid	Swegrid	Norgrid
Q1	1	X	1
Q2	1	X	1
Q3	X	0	0
Q4	X	X	1
Q5	1	X	1

Table 1: Operations Security User documentation survey on Nordic Computational Grids

6 Results

Table 1 presents the results in tabular form. Number 1 stands for yes, number 0 for no and X for both or indefinite. To facilitate interpretation of the table, we present the questions asked again:

Q1: Does the service require the end-userto follow an acceptable use policy (AUP)?

Q2: Does the AUP include obligations for the end-user?

Q3: Is grid service committed to provide a minimum service or availabilty level?

Q4: Does the security policy include sanctions such as closing of an account for not complying with AUP?

Q5: Is the AUP supported by security guidelines for the end-user on how to practically implement the security measures?

The study and results shows that, operations security as manifested in end user related security policies and security guidelines, were implemented in heterogeneous ways in the grid infrastructures studied. That is, by definition, not a negative finding, as by the very definition grid infrastructures are based on distributed administration. Common or similar policies could just point to a lack of distributed administration.

A somewhat disturbing fact found out by study, was that in some grid infrastructures, no operations security measures were at least publicly documented on the internet. We also found some inconsistent, outdated and quite short guidelines and policies.

Some security guidelines presented handling of certificates primarly as a technical challenge necessary to solve to be able to send grid jobs. We would also like to see that the users simultaneously trained about the security implications and proper storage of the certificates.

This paper has made a very limited scratch on the surface and the depth of the survey could be questioned. More reliable surveys would need to also include both technical audits, interviews and assessment of grid system administration procedures as well.

To manually search for security documentation on the internet is by no means an exhaustive method. Documents might be available on print, guidelines communicated only manually, web pages can be temporary unavailable or the researcher might by human error just miss the link were the policies and guidelines can be found. Regardless of the shortcomings of this simple method, a total lack of security documentation is an indication of poorly implemented operations security. Documentation only doesn't enable and empower operations security, In addition security training, continuous awarness building and motivation is needed.

The research questions could be scrutinised, were they the optimal ones to ask. Now, when some results are already known, better questions could be formulated for future research. More extensive research could formulate and test implementations of well known taxonomies on operations security

The methods used in this paper can also be criticised for a kind of ethnocentrism. The questions asked had perhaps a bias to practices implemented in the home country of the researcher. It is often too easy to think that familiar practices are more secure and reliable than those practised by others.

7 Discussion

Does security really matter in Grid computing? One of the main reasons to invest in security is to maintain trust in system and services. The concept of trust for network based services has been clarified by a classical paper by Clark and Blumenthal [10], where it is stated that

> " trust or trustworthiness thus includes some of the issues associated with security".

A trend of moving the end-to-end or trust-to-trust logic from lower layers to more application-specific layers of communication do make sense. Regulation or security controls between the ends or the trusting parties is a controversial question also for other stakeholders. The user or policy has sometimes as anecdote been described as the eighth (OSI) layer. We think, that that there should specially in Grid infrastructures and NRENs also be a reciprocal trusts between security policies of different interconnected sites. One site and one user can make large infrastructures on peril by just missing simple rules of operations security. All threats cannot be mitigated by good protocols and technical security controls if the user doesn't behave.

Whether or not the end-user trusts their systems, convenience drives end- users to use them. In computational grids the challenge is that often a single end-user does not have staked more than perhaps a computational job which can be generally easily repeated in case of system failure. Ensuring long term service availability and system integrity should be in the interest of governments and principals who want to secure the prerequisites for computational research.

Also for grid users it can be difficult to trust the complex system. Trust architecture cannot be convincingly implemented by technical security controls only, but needs operations controls to complement the technical ones.

Can the users in an environment with multiple stakeholders, such as computational Grids depend solely on the technical security controls built in to the system architecture? Our answer is no. To obtain true end-to-end trust, people also need to be obliged, educated, and motivated to and for operations security.

Without making any allegations of insufficient operation security controls at those computational grid sites which have lately suffered from serious hostile intrusions, many grid administrators probably do try to assess the risks also related to operations security.

International bodies, such as EGEE, NDGF and E-IRG, who are developing technologies and services for grid infrastructures should encourage and support projects for harmonising and synchronising operations security controls for computational grids.

Also, the actual effects and implementations of operations security should be researched in much further details.

8 Conclusion

In this paper we have surveyed the implementation of operations security in three computational grid infrastructures in the Nordic Countries based on security documentation for end users. The survey has been limited to publicly available material on the web site of the grid providers.

Operations security has been assessed, because in addition to secure architecture and technology, both users and administrators need to comply with security practices to ensure availability and integrity of the grid infrastructures.

Despite the limited scope and economical methods of the survey, our results show that the implementations of operations security vary greatly from site to site. Nevertheless, most sites although communicate the most fundamental security behaviour to existing and future users.

In the future, more effort should be made to harmonise and synchronise operations security controls in an between computational grids to avoid unnecessary security threats in the form of account and system compromises. Also, operations security for computational science related services should be an object for further comparative research.

9 Acknowledgement

The writer of this paper wish to thank Dr. Vera Hansper, Mr Arto Teräs and Mr. Michael Gindonis of CSC IT Center for Science for advice and interesting discussions and his tutor, Mr. Jani Heikkinen from Helsinki University of Technology for patience and many improvements.

References

[1] M-grid acceptable use policy, 2008. `http://wiki.hip.fi/twiki/bin/view/Extranet/MGrid/SecurityWG/AcceptableUse, accessed2008-10-20`.

[2] M-grid security policy, 2008. `http://wiki.hip.fi/twiki/bin/view/Extranet/MGrid/SecurityWG/MgridSecurityPolicy, accessed2008-10-20`.

[3] M-grid user security guide, 2008.

[4] M-grid web site, 2008. `https://extras.csc.fi/mgrid/,accessed2008-10-20`.

[5] Rules for nsc user account holders, 2008. `http://www.nsc.liu.se/start/apply/rules.html,accessed2008-11-10`.

[6] Swedish national allocations committee web site, 2008. `http://www.snac.vr.se/,accessed2008-11-10`.

[7] Swegrid web site, 2008. `http://www.swegrid.se/,accessed2008-11-10`.

[8] I. A. Cormack, U. Kaila. *CSIRTs and Grids - discussion paper*. TF-CISRT, 2005.

[9] Center for scientific and technical computing LUNARC Lund University. Lunarc Rules Concerning Accounts, 2008. `http://www.lunarc.lu.se/Start/LunarcRules,accessed2008-11-10`.

[10] D. D. Clark and M. S. Blumenthal. The end-to-end argument and application design: the role of trust. Telecommunications Policy Research Conference, 2007.

[11] P. Eerola, T. Ekelof, M. Ellert, J. R. Hansen, A. Konstantinov, B. Konya, J. L. Nielsen, F. Ould-Saada, O. Smirnova, and A. Waananen. The nordugrid architecture and tools, 2003.

[12] I. Foster. What is the grid? - a three point checklist, July 2002.

[13] I. Foster and C. Kesselman. *The Grid: Blueprint for a Future Computing Infrastructure a*. Morgan Kaufmann Publishers, 1998.

[14] M. H. Tipton, F.Tipton. *Information Security Management Handbook: Vol.3*. CRC Press, 2006.

[15] KTH (Royal Institute of Technology). Rules for computer, network and system facilities, 2008. `http://www.pdc.kth.se/systems_support/accounts/test/rules/,accessed2008-11-10`.

[16] J. Myers. SASL: Simple authentication and security layer. RFC 2222, IETF, Oct 1997.

[17] National center for Scientific and Parallel Computing. Rules for using hpc2n resources and services, 2008. `http://www.hpc2n.umu.se/account/rules.html,accessed2008-11-10`.

[18] L. Nixon. The stakkato intrusions: What happened and what have we learned? *Cluster Computing and the Grid Workshops, 2006. Sixth IEEE International Symposium on*, 2:27–27, May 2006.

[19] NorGrid - Norwegian GRID Initiative. Norgrid web site, 2008. `http://www.norgrid.no/,accessed2008-11-10`.

[20] NOTUR - The Norwegian Metacenter For Computational Science. A guide for using the notur facilities, 2008. `http://www.notur.no/quotas/rules/,accessed2008-11-10`.

[21] NOTUR - The Norwegian Metacenter For Computational Science. The regulations of use by the notur partners, 2008. `http://www.notur.no/quotas/rules/eachsite.html/,accessed2008-11-10`.

[22] Uppsala Universitet. Computer usage policies at the it department, 2008. `http://www.it.uu.se/datordrift/datorregler,accessed2008-11-10`.

[23] V. Welch, F. Siebenlist, I. Foster, J. Bresnahan, K. Czajkowski, J. Gawor, C. Kesselman, S. Meder, L. Pearlman, and S. Tuecke. Security for grid services. *High Performance Distributed Computing, 2003. Proceedings. 12th IEEE International Symposium on*, pages 48–57, June 2003.

Trust Management mechanisms

Ming Li
Helsinki University of Technology
ming.li@tkk.fi

Abstract

Internet applications is prevalent in recent years, such as e-commerce, web-based access to services and inter-personal interactions by e-mail. The trustworthiness of these services become a major concern. This paper reviews the various definition of trust and properties of trust management. Some typical examples of trust management solutions and their application in practice are described. A SNS-based trust management concept is proposed.

KEYWORDS: Trust Management, public key credentials, reputation system

1 Introduction

Along with the rapid development of network and communication technologies and increment of the population using computers in the last decade, more and more interactions concerning sensitive and private information are carried out on the Internet, such as online business, net-bank service, financial reports transmission and son on. This trend, resulting from the emerging forms of distributed systems, leads to the migration from centralized access control systems to distributed network-based trustworthiness access. Trust becomes a critical issue in distributed systems. A flexible, effective, distributed and independent trust management mechanism is needed. To address this challenge, trust management solutions have been proposed to protect users' confidential information. They differ from the traditional mechanisms which are usually used for database or operating system's access control. Trust management is more suitable for peer to peer web-based Internet services. For example, in a e-commerce scenario, buyers must trust the sellers who provide their advertised commodities before paying them. Thus, a trust relationship should be established previous to the transaction. In this paper, we provide an overview of the trust management research and review the state of the art of trust management solutions. Finally, a SNS based trust management mechanism is proposed.

The rest of this paper is organized as follows. Section 2 provides an overview of the trust management concepts and model. In section 3, some typical trust management solutions are introduced and analyzed. And a SNS based trust management mechanism is proposed in this section also. Some applications related to trust management are discussed in section 4. And finally conclusion is made in section 5.

2 Trust management overview

Generally, trust is a mixture subject relative to trust persons' or services' properties, including competence, credibility, reliability etc. There is no consensus on the definition of trust in the literature. Researchers usually define the trust in a specific way in a certain context environment. In an electronic commerce context, We define trust as "the steady reliance on the credibility, competence and security of a e-transaction".

The term trust management[5] was first created by Matt Blaze and his colleagues in 1996. A trust management system consists of numerous roles and properties. The main roles of a trust management system is the trustor and trustee. During a e-business transaction, the trustor is a service provider providing some resources, such as a software execution permission or an application service. The trustee is an agent (representing a transaction partner or an individual customer) requiring access to the trustor's advertised services. A trust decision is binary decision on whether to trust or not to trust the trustee considering its trustworthiness.

Trust management research stems from authorization and authentication. Digital certificates[pubic key certificates] can be used to authenticate one's identity or membership in a community. Credentials are enough, when the trustor is convinced of the trustee's identity and knows it to be a member of some sufficiently trusted community. However, the digital certificates cannot automatically or dynamically handle whether permit an identity to access trustor's resources or not. In order to make it possible to automatically control whether certain credentials have sufficient right to perform certain actions, Policy languages[10, 8] are proposed.

In real e-commerce world, the trustworthiness of an entity changes dynamically. The history behavior of the trustee should be considered during a trust decision. In 2000, though intrusion detection systems are proposed to monitor users' behavior, the collected information was not taken into consideration. None of the existing systems then yet covered monitoring and re-evaluation of trust[18]. Since then, numerous reputation systems have been proposed. Two representative reputation systems in practice are eBay and Amazon. Nevertheless, a reputation system usually gleans and aggregates the history behavior of a group of entities within a specific community. It would be effective and efficient if different reputation communities could share the reputation information in a friendly way. To solve this issue, [16] proposed "A reputation and Trust management Broker Framework for web applications". In the following section, some representative trust management solutions would be discussed in detail.

3 Trust management Solutions

3.1 Public key credentials

A digital certificate is issued by a certification authority(CA) and verifies that a public key is owned by a particular entity[1]. It usually incorporates a digital signature to bind a public key with an identity which describes the owner's information like name, organization, address etc.

In a typical public key infrastructure(PKI) scheme, the signature will be of a certificate authority (CA). The certification authority is not responsible for verifying the trustworthiness of the key owner, but merely authenticates the owner's identity. Due to the implicitly reduction of the trustor's risk in dealing with the trustee, it's necessary to establish a service access or service provisioning trust relationship. However, governing what resources or services the trustee is allowed to access is not administrated by the certificate infrastructure, but is left up to the application level. Two of the main certificate systems dealing with authentication are PGP and X.509. They are also basis for the subsequent trust management systems.

Pretty Good Privacy is a computer program that provides cryptographic privacy and authentication[1]. It is often used for personal communication especially for e-mail type of applications by signing, encrypting and decrypting e-mails to enhance the security. PGP as well as other similar products follow the OpenPGP standard[13] for encrypting and decrypting data.

PGP supports message authentication and integrity checking. Integrity checking can detect whether a message has been changed since it was accomplished by the sender, and authentication can be used to determine whether it was actually sent by the sender who claimed to be. The sender leverages PGP to create a digital signature for the message by signature algorithms. Firstly, PGP calculates a hash from the plaintext, and then produces the digital signature from that hash using the sender's private key. And then, once receiving the message from sender, the recipient utilizes the sender's public key and the digital signature to recover the original hash. The recipient compares this hash with the one which is calculated by himself from the recovered plaintext. If these two hashes match, it can be assumed with high possibility that the received message has not been modified during the transformation and the message is sent by the claimed sender.

PGP was created by Philip Zimmenrmann in 1991. While originally used mainly for encrypting/decrypting the contents of e-mail messages and attachments from a desk client, PGP products have varied since 2002 into a set of encryption/decryption applications which can be determined by an optional central policy server. PGP encryption applications include e-mail and attachments, digital signatures, laptop full disk encryption, file and folder security, protection for IM session, batch file transfer encryption and protection for files and folders stored on network servers and , more recently, encrypted and/or signed HTTP request/response by means of a client side and a server side plug in.[11]

In contrast with PGP, the X.509 trust model is a strictly hierarchical trust model for authentication[7]. Every entity having a certificate signed by CA is compulsory. In X.509, all the certificate authorities are integrated to a CA tree. When a user generates a public/private key pair, it register the public one to a CA which can make certification on it. If two users register with the same CA, it would be very convenient to communicate with each other, just by exchanging their certificates directly. But if two users register with diverse CAs, they have to seek help from the high level CAs, until a common CA is reached. Thus, the certification authority tree is mapped to a trust tree.

It is important to note that neither of these models can be used to model trust in all domains. Two steps are needed by PGP and X.509. 1) bind a public key to its owner; 2) bind the access rights to the public key. However, the step2 is out of the certificate framework. PGP and X.509 only support partial trust management because they just certify the identity but not bind it to access rights or authorized actions to sources. To guarantee the entities' trustworthiness, we need to bind the access rights to a public key in one system. In the following section, some classical trust management systems are described.

3.2 Trust Negotiation Systems

Due to the limitation of Public key credentials based approaches, like PGP and X.509, which do not bind access rights to the owner of the public key, some researchers proposed trust negotiation(TN)[15] systems.

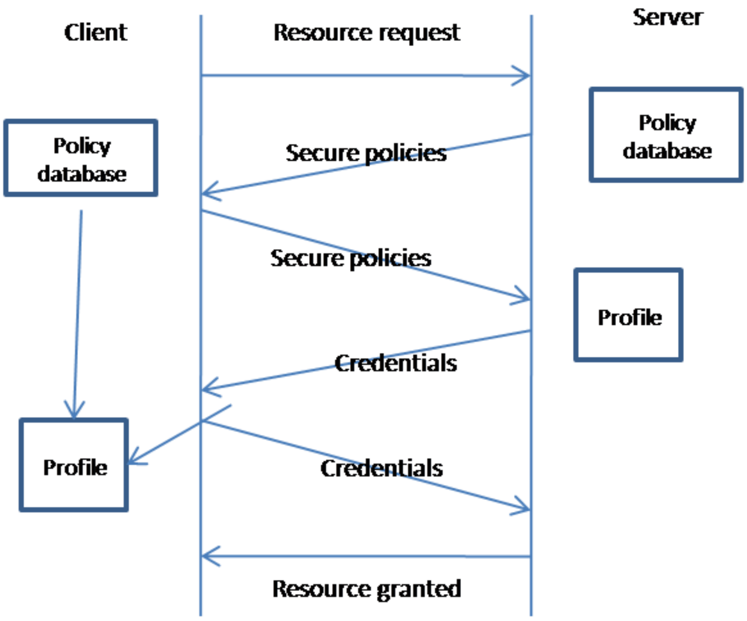

Figure 1: A trust negotiation(TN) process

As illustrated in Figure 1[2], a trust negotiation system consists of a client(trustee) and a server(trustor). The client asks for some resources possessed by the server, like some sensitive information and service. During the negotiation, the trust relationship between them increase by verifying the properties of each other. Every participator's properties are described by digital credentials which are gleaned by each party and reserved in credential repositories. In order to bind access rights to digital credentials, disclosure policies are used to govern access control to resources. Disclosure policies specify the credential constraints that the correspondent party must provide to gain the access right.

PolicyMaker[10] is a trust management application, developed at AT&T Research Laboratories. It authorizes the access rights to public keys. The PolicyMaker system is actually a query engine[18]. It assesses whether a required action is consistent with the disclosure policies. The inputs to the PolicyMaker interpreter include disclosure policies, the received digital credentials and requiring description. The output of it is tri-response, namely yes, nor or a list of restraints which can make the require to resources acceptable.

Other trust negotiation systems such as Keynote[9] and Rule-controlled Environment For Evaluation of Rules and Everything Else(REFEREE)[8], either make some improvement or extension based on PolicyMaker. Although they can are good approaches in some context for trust management, they address only authorization based on public keys, which still does not comprehensively solve the entire trust management problem. They concentrate on establishing resource access control over trust, and possibly service provision trust in a static way. Due to the dynamic properties of human beings(represented by trustee or trustor), the trustworthiness of an entity is dynamic also. So as to make trust model more dynamic, the history behavior of entities should be taken into consideration as well. Reputation system is[3] what we need and it will be discussed in the next subsection.

3.3 Reputation systems

Due to the growth and popularity of online transactions and electronic business activities, previous distributed authorization and authentication mechanisms can only provide trust management in a static way. They can not guarantee that an entity will always behave in a good way in every aspect.Reputation system is one of the solutions to address this issue by collecting feedback ratings.

In the real world, when people go shopping, reputation of the company is a key reference to make a decision on whom you could rely upon and whom you could not. In the Internet, reputation systems act as the same way as that in the real world. In a reputation system, feedback rating from entities are gleaned over a period of time to reflect the dynamic trustworthiness of an entity. The entities here could either be trustees or trustors.

The reputation systems that Amazon and eBay have adopted are two representative examples. In practice, there are lots of diverse ways to compute trust. Amazon calculates an average of product ratings based on customer reviews; eBay lays out the feedback score and the percentage of positive feedbacks among all feedbacks. A successfull reputation system should make it hard to build up good reputation so that a user is less likely to abuse its hard earned reputation[16]. But in order to prevent that some customers to give malicious feedback, some measurements can be adopted to improve the current reputation system, such as giving high weights to trustors from higher reputation customers.

As the centralized reputation systems, Amazon and eBay also have the problems of vulnerability and scalability. All the trust ratings for some customers in Amazon can not be shared to eBay. To address this issue, a good trust recommendation mechanism is needed, which will be discussed in the following subsection.

3.4 SNS-based Trustworthiness recommendation system

Sharing reputation or trust ratings between different communities is a great challenge. One user's reputation rating in one community cannot be mapped to that in another community directly, because different communities have diverse trust perception of the same rating. Even if there may be some solutions to solve the consistency among different reputation system communities by giving different weights to communities according to their overall reputation rating. But this can lead to confution or mass because of the dynamic of reputation rating of different communities and the difficulty in setting up different weights to distinct reputation communities due to the competition among them.

In this paper, a SNS(Social Network Service)[17] based Trustworthiness Recommendation Mechanism(STRM) is proposed to address the share of reputation ratings among distinct communities. SNS is online communities where people can share their interest, activities, multimedia content and so on. In recent years, incremental amount of people start to use SNS related applications, especially social network websites, like Facebook, MySpace, Xiaonei and so on. Different SNS applications focus on diverse categories, such as former classmates, relatives, colleagues. The key feature of the SNS originates from the Six Degree of Separation theory[12]. Its general concept is that anyone in the world can know every other person by its friend's introduction iteratively within six times. The closest friend is regarded the direct friend whose degree is one. The friends of one degree friends are the two degree friends. Therefore, go as this way, the most unfamiliar friends are the six degree friends.

In the proposed STRM, we give diverse weights to friends according to their degree. By this way, we link SNS degree weights to trust recommendation system to share SNS-based reputation ratings among different communities. Due to the consistent perception of degree of friends, different communities can share every other community' recommendation rating value directly.

As illustrated in Figure 2, the previous recommendation systems can not unify the rating values among different reputation communities. Even if a central agent is deployed as a arbitration, there would be still no consensus from different perspectives about the weights of rating value from other communities.

While, if the proposed STRM is adopted as described in Figure 3, then the raging value by different users in different SNS-based communities would be mapped to the degree of friendship. In this way, all reputation rating value is connected together compatibly

Some simple mapping policy between rating value and degree of friendship is formulated as follows.

In Table 1, the basic Value is depended on the degree of the friendship, and the recommendValue is determined by the friends. High degree friends could give high basicValue

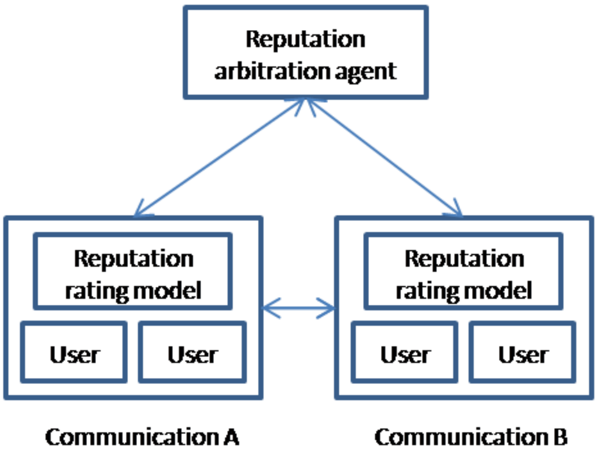

Figure 2: reputation rating system

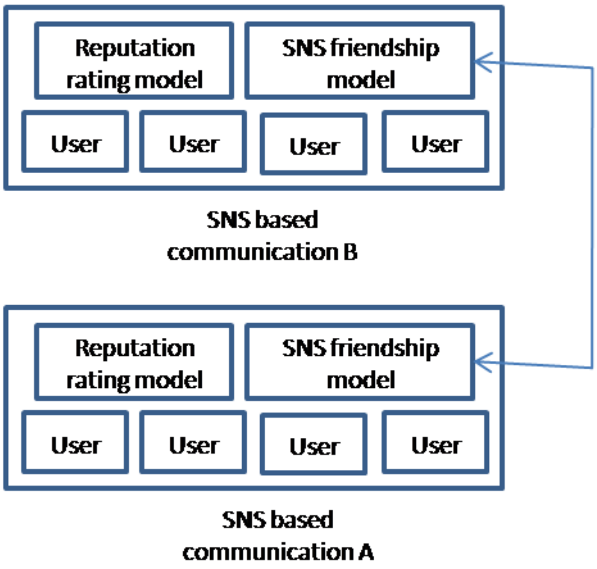

Figure 3: SNS BASED reputation rating system

and large recommendValue scope. For example, a two degree friend can give the reputation value combined with basicValue 5 plus recommendValue 1. So the final reputation value is 6.

4 Applications of Trust Management

Trust management is used widely in the practice. A few application domains that highlight some specific trust requirements are described in this section.

Along with the widespread utility of advanced technologies in clinical care, patient care becomes more and more complex. Clinical Information System(CIS)[14] is a solution designed to manage the patient data into the computerized information system. The development of large scale of ICS networks facilitate the use of cryptography to guarantee privacy, authentication and integrity of patient medical care records. This in turn necessitates the trust management in health networks. The trust management model of CIS must provide a mechanism to authorize physicians' cryptographic keys to undertake certain types of medical records. In [4], the authors demonstrate the advantage of the PolicyMaker approach to trust management. The PolicyMaker uses the policy management and certification dissertations that it receives as inputs to permit or refuse a request.

Degree	basicValue	recommendValue
One degree	6	1-6
Two degree	5	1-5
Three degree	4	1-4
Four degree	3	1-3
Five degree	2	1-2
Six degree	1	1

Table 1: degree of friendship mapping to basic reputation value

Another trust management used domain is information retrieval systems. The major questions on trust are "Does this piece of information meet my viewing needs?", "Will I get the information I requested?", and "Will the information have any effects on my system?"[18]. The first question can be solved for the internet web pages content by PICS[6] label. The principle idea of PICS is that there should be a filter in the middle of web pages viewers and web contents. PICS defines specifications for the format and distribution of labels describing web pages content.by means of adding meta-document.

5 Conclusion

With the expansion of broadband Internet and the emerging communication technologies, the growth of e-commerce, military or other access to sensitive e-resources continues. In such an open networked world, conventional systems adopting identity-based access control mechanisms, such as databases and operating systems no longer hold. How to establish the trust relationship between entities with no prior knowledge of each other becomes an issue. Researchers have proposed trust management as a novel mechanism to protect users' privacy and resources, and to enable people to deal with the uncertainty. This paper concentrates on providing a general overview of the state of the art on trust management. A few application domains that leverage some specific trust management systems are discussed.

References

[1] An Introduction to Cryptography. In *PGP 6.5.1 User's Guide, Network Associates Inc*, pages 11 – 36.

[2] Bertino, E.; Ferrari, E.; Squicciarini, A. Trust negotiations: concepts, systems, and languages. In *Computing in Science & Engineering*, volume 06, pages 27 – 34, July-Aug 2004.

[3] Bin Yu, Munindar P. Singh. A social mechanism of reputation management in electronic communities. In

Proceedings of Fourth International Workshop on Cooperative Information Agents, 2000.

[4] M. Blaze, J. Feigenbaum, and J. Lacy. Managing trust in medical information systems. Technical report, AT&T Technical Report, 1996.

[5] Blaze M., Feigenbaum J. and Lacy J. Decentralized Trust Management. In *IEEE Conference on Security and Privacy, Oakland, California*, 1996.

[6] BLAZE M.,FEIGENBAUM J.,RESNICK P.,STRAUSS M. Managing trust in an information-labeling system. In *Conference Security in Communication Networks, Amalfi , ITALIE*, pages 491–501, 1997.

[7] A. C. and F. S. Internet X.509 Public Key Infrastructure Certificate Management Protocols. RFC 2510, The Internet Engineering Task Force, 1999. `http://www.cis.ohio-state.edu/htbin/rfc/rfc2510.html`.

[8] Chu Y.-H., Feigenbaum J., LaMacchia B., Resnick P. and Strauss M. REFEREE: Trust Management for Web. In *Applications, 1997, AT&T Research Labs*, 1997. `http://www.research.att.com/~jf/pubs/www6-97.html`.

[9] M. B. et al. The KeyNote Trust-Management System. RFC 2704, The Internet Engineering Task Force, September 1999. `http://ietf.org/rfc/rfc2704.txt`.

[10] Feigenbaum J. Overview of the AT&T Labs Trust Management Project: Position Paper. In *1998 Cambridge University Workshop on Trust and Delegation*, 1998.

[11] S. Garfinkel. *PGP: Pretty Good Privacy*. O'Reilly, 1995.

[12] J. Guare. *Six Degrees of Separation*. Dramatists Play Service Inc, 1992.

[13] H. F. D. S. R. T. J. Callas, L. Donnerhacke. OpenPGP Message Format. RFC 4880, The Internet Engineering Task Force, November 2007. `http://ietf.org/rfc/rfc4880.txt`.

[14] Jao, C.S. Helgason, C.M. Zych, D. The role of clinical information systems in improving physician productivity: Challenges facing the adoption of an electronic charge capture system. In *Systems, Man and Cybernetics, 2007. ISIC. IEEE International Conference*, pages 3464–3468, October 2007.

[15] K. Seamons et al. Protecting Privacy During On Line Trust Negotiation. In *Proc.2nd Workshop on Privacy Enhancing Technologies*, pages 129–143, 2002.

[16] Kwei-Jay Lin; Haiyin Lu; Tao Yu; Chia-en Tai. A reputation and trust management broker framework for Web applications. In *e-Technology, e-Commerce and e-Service, 2005. IEEE '05. International Conference*, pages 262–269, 29 March-1 April 2005.

[17] K. F. Stanley Wasserman. *Social network analysis*. Cambridge University Press, 1994.

[18] Tyrone Grandison and Morris Sloman. A Survey of Trust in Internet Applications. In *4th Quarter 2000 issue of IEEE Communications Surveys & Tutorials.*, 2000.

Reputation-based Trust Models in Peer-to-Peer Networks

Flutra Osmani
Helsinki University of Technology
fosmani@cc.hut.fi

Abstract

Peer-to-Peer file sharing networks are widely used today as a means of sharing and distributing digital content. However, the anonymous and open nature of P2P networks makes them suitable platforms for sharing inauthentic files and free-riding. One way to deal with such inauthentic resource attacks and resource abuse is to construct a reputation-based trust model which assesses the trustworthiness of the participating peers based on their past behavior. The trust mechanism should scale well for large number of peers and should not require any central authority in order to compute their trust values. Various simulations have shown that the deployment of a trust scheme in peer-to-peer networks significantly decreases the number of inauthentic file downloads under various threat circumstances, as well as reduces the number of free-riders. This paper evaluates few of the existing reputation-based trust models in terms of their efficiency and performance, and it further presents practical examples of reputation systems being used in P2P applications.

KEYWORDS: Reputation, Trust Management, P2P Networks, Algorithms, Performance, Evaluation

1 Introduction

Peer-to-Peer (P2P) networks have set a new model for content distribution in the Internet. The scalable and distributed nature have made P2P a very attractive solution for file-sharing among peers, each of whom plays the role of both the provider and the customer of the service. The lack of a central authority for coordination, central database and reliable connections; the open and anonymous nature of P2P - leave an open door for anonymous peers to inject, among other malicious content, inauthentic files in the network. Moreover, the fact that any entity can join, and no entity is trusted or monitored, further expose the P2P system to various attacks and allow for malicious users to use the network without contributing back to it. Therefore, trusting other peers and assessing the quality of the content they provide is crucial for the performance and the future of the P2P file-sharing systems; isolating free-riders and increasing the sharing-ratio among peers improves the performance of the system. Such circumstances necessitate for new methods which address the problem of inauthentic resource attacks, but most importantly, help identify and isolate malicious peers who can pollute the network with unlimited amount of malicious content. Such methods should reduce the number of free-riders and effectively increase the number of the sharers within the system. As a result, a few reputation-based trust models have been proposed to tackle such attacks and provide with a feasible solution to trust management, including Eigentrust [14], PeerTrust [10], Simitrust [6], CuboidTrust [3] and trust management scheme with P-Grid [8, 7]. With their main goal being the trust formation among peers, these trust models try to define the reputation of a peer *i* by computing a global trust value, which is a collection of assessments of all other peers in the network that have interacted with this peer *i*. In this respect, the information about transactions among peers is highly distributed throughout the P2P network, therefore every peer can only compute an approximate trust global value of a peer *i*, as well as of the whole network. Depending on the design details of each trust method, the trust metric which evaluates peer's reputation constitutes of important dimensions: direct and indirect transactions, context, or quality of resource [3]. In the next sections, this paper elaborates on few of the trust schemes; it further evaluates their effectiveness in reducing the inauthentic resource attacks, but primarily, in identifying and isolating the malicious users from the P2P network. In Section 2, trust concepts and design principles of a trust scheme are discussed. In Section 3, few of the existing reputation-based trust methods are explained and evaluated in terms of their effectiveness and efficiency; their underlying algorithms are assessed in terms of computation overhead, performance and security. In Section 4, two practical examples of such trust methods being implemented in actual P2P applications are given. In this regard, the trust schemes are evaluated for their effectiveness in eliminating the problem of resource abuse, caused by free-riders.

2 Managing Trust

In electronic market, reputation has been used as a fostering mechanism for cooperation among buyers and sellers. In a setting such as eBay, traders make use of a feedback mechanism to gain their reputation values [9]. This feedback mechanism enables each trader to publicly rate any other trader, by submitting his feedback to a dedicated server. Contrary to this scenario, the decentralized nature of P2P requires no central servers but puts the burden of computing and maintaining of global trust values on the peers themselves. Under the latter infrastructure, trust management obtains a different dimension; it becomes a mechanism that enables mutual trust establishment, based on the reputation of peers. The reputation is derived from direct or indirect knowledge of previous transactions among peers, and it is used as a means to assess the level of trust one peer puts into another peer [8].

Aberer et al [8] emphasize the importance of *context* when considering trust, but context considerations are out of scope in this paper.

Aberer et al further present a model which helps build a trust assessment method as follows: one can derive the trustworthiness of peer p from **B**, where p is part of the community **P** and **B** constitutes of all the transactions that have occurred among the peers of community **P**. However, the question of data management immediately arises in this situation: how to reliably obtain and efficiently maintain this set of *behavioral data* **B**, as well as, how can peers obtain the necessary data from **B** in order to compute trust of other peers. Moreover, it is a flawed approach to determine the trust of a peer based on data sources - whose reliability cannot fully be determined, therefore it is crucial that each peer providing data about their previous transactions should as well be assessed for its trustworthiness. P-Grid [7] has been proposed to address this issue - which will further be elaborated in the next section of this paper.

Even under the scenario where there is a lack of a central data source [8], where peer p relies partially on the data obtained from direct interactions and on the data provided indirectly through *witnesses* in order to obtain reputation values, issues arise to whether the data received from witnesses is reliable at all, since they can be malicious.

Thus, in order to properly define design details of a reputation-based trust scheme for a decentralized P2P system, all of the above questions, among others, should be addressed. For the reputation system to become fully scalable and effective in a distributed environment, according to [14], the following issues should be addressed:

- The shared ethics is defined and enforced by the peers and not a central authority. In other words, each peer stores the transactions it has had with other peers and is therefore able to rate them

- Peer's anonymity should be enforced; each peer's reputation should be associated with an identifier and not an IP address

- Newcomers to the system should not benefit initially; the reputation of peers should be build by constant good behavior. Moreover, malicious peers with bad reputation should not be able to change their identifiers in order to obtain the status of a *newcomer*

- The underlying algorithm should have minimal overhead in terms of computation, storage, and message complexity

- The system should be resilient to collectives of malicious peers who try to subvert the system

3 Trust-Based Models

Selection of honest peers and reliable information remain the two most critical components that ensure a successful performance for a P2P file-sharing system; tasks that are particularly hard to achieve under a fully decentralized architecture. The new approaches suggest for trust-based mechanisms that would address a wider range of issues: inauthentic resource attacks, resource abuse, or increase of incentives for peers to share. Moreover, trust-based schemes seem to be the preferred method even for data validation - where simple hashing techniques or cryptographic signature based solutions such as [2] would have offered efficient ways to confirm content authenticity. Wang et al [15] suggest that a trust evolution model - to build trust relationships among peers but also predict the quality of the resources provided by the peers, is the new differentiated approach from the traditional stance. In this regard, Wang et al argue that some prior knowledge about the peers is not sufficient to solve the problem in a decentralized P2P architecture, where peers inevitably interact with anonymous and strange peers.

The new *trust evolution model*, according to Wang et al, should accommodate decentralized trust formation, evolution, and propagation. In the following sections, few of trust-oriented schemes will be discussed.

3.1 EigenTrust

Kamvar et al [14] propose a secure distributed trust algorithm, based on *transitive trust*: a peer i will have a high opinion of those peers who have provided it with authentic files. Additionally, peer i is more inclined to trust those peers, since peers who provide with authentic files naturally are assumed to be honest in reporting their trust values. In EigenTrust [14], the global reputation of a peer i is a collection of the local trust values reported by other peers, whose opinions are weighted against their own global reputations. In the algorithm, the notion of *pre-trusted peers* is introduced - a set of peers **P** that are known to be trusted, in order to guarantee faster convergence and break up malicious collectives, who report false trust values and strive to inject inauthentic files in the system. Kamvar et al further address this security issue by proposing a secure score management scheme where the trust value of one peer is computed by several score managers **M**, who are responsible for that particular peer. To assign these score managers to peer i, EigenTrust makes use of multi-dimensional hash functions and distributed hash tables (DHT) [13].

When requesting content, peers can make use of the computed global trust values in order to download from honest peers, thus avoid downloading from malicious ones. Moreover, obtaining high global trust values implies that peers share a high amount of authentic files, thus providing an incentive for users to share and get rewarded with higher trust values. The simulations show however that peers with the highest global trust values are always chosen as downloading sources, thus leading to an imbalance of the network and lack of scalability. The situation gets worse when peers with high trust values accumulate even more trust, eventually becoming the download sources for virtually all the content requests in the network.

However, in terms of algorithm's efficiency in reducing the number of inauthentic file downloads, the above scheme successfully manages to eliminate them, since malicious peers would need to gain high trust values in order to be selected as download sources. During simulations [14] under various threat models, the algorithm proved to be efficient even in cases where the malicious peers teamed up in collectives, with the goal of raising their trust values. As previously mentioned, the presence of pre-trusted peers facilitates the breaking of such collectives, thus minimizing the amount of inauthentic downloads.

Another scenario to be considered is when malicious peers try to raise their trust values by occasionally injecting some authentic files. Though their impact obviously results in a higher number of inauthentic downloads, it comes with a higher price for malicious peers: they have to share more authentic files.

Therefore, Eigentrust successfully manages to reduce the number of inauthentic files, and in turn, isolate the malicious peers. Moreover, the algorithm used converges as fast as in less than 10 iterations, which means that in less than 10 exchanges of updated values among peers, the computed global trust values do not change significantly any more. Also, the algorithm puts as low overhead as possible on peers, by limiting the number of local trust values that each peer reports [14].

3.2 CuboidTrust

In EigenTrust [14], the basic assumption that - peers who provide with authentic resources are likely to report correct trust values, is a risky assumption and therefore the reporting peers should be assessed for their trustworthiness as well. While EigenTrust considers peer's *contribution* with authentic files as the main parameter in forming trust, CuboidTrust [3] adds to its model two additional trust factors: *trustworthiness* and *quality of resource*.

In CuboidTrust, the *contribution* parameter implies two things: a peer having a high contribution parameter score implies that the content stored at this peer is authentic with a high probability; a peer with a low contribution store indicates that the content stored on him is most probably inauthentic. It is the *trustworthiness* parameter which defines peer's trustworthiness in reporting honest values on other peers. In typical models, a peer *i* rates any other peer *j* based on their previous interactions, whereas in CuboidTrust model, a peer is given the opportunity to rate the quality of each resource it has received from any other peer.

Chen et al [3] construct a cuboid with coordinates (*x, y, z*) in order to represent the quality of resource *z*, stored at peer *y*, rated by peer *x*. Chen et al use the Cuboid to define: the contribution score vector (C), the trustworthiness score vector (T), and the quality of resource vector (Q) of a peer *i*, which will be used to mathematically derive four *relations* among the three trust factors. According to [3], the derived relations reflect the following:

- *The relation from (T) to (C)* - C reflects the contribution of peer *i* by considering the experiences of all other peers in the network. The relation could be illustrated as follows: contribution of peer *i* would increase if he was rated by a peer *j*, whose trustworthiness score was high; otherwise, the contribution score of peer *i* would decrease if high ratings were given to him from peer *j*, who has negative trustworthiness score.

- *The relation from (Q) to (T)* - Under the assumption that resource *j* is authentic and has a positive (high) quality score: if peer *i* rated resource *j* with a positive score, this would have increased the trustworthiness of peer *i* itself; otherwise, the trustworthiness of peer *i* would decline if he gave negative quality score to resource *j*.

- *The relation from (T) to (Q)* - The quality of resource *i* would increase if positively rated from a peer *j* who has positive trustworthiness score; otherwise, resource *i* could be labeled as inauthentic if rated negatively by peer *j* who has negative trustworthiness score.

- *The relation from (C) to (T)* - Assuming that the contribution score of peer *j* is high: if peer *i* gave a positive score to peer *j*, this would increase the trustworthiness of peer *i*; otherwise, if peer *i* were to give a negative score to peer *j* - this would suggest that peer *i* is malicious, therefore his trustworthiness would be reduced.

The above derived relations between the three trust factors reveal that that the global contribution parameter determines whether a peer is trustworthy or not; the shared resources of peers with high global contribution are generally authentic, and the resources are inauthentic if peers have low global contribution scores.

The performed simulations on CuboidTrust [3] show that its efficiency is better than that of EigenTrust [14] or PeerTrust [10], as it takes only six cycles for the CuboidTrust algorithm to make the fraction of inauthentic file downloads not change dramatically any further. Further, they indicate that CuboidTrust outperforms the other two trust models, as only 7 percent of all the downloaded resources result in inauthentic file downloads - a smaller value compared to the ones of EigenTrust and PeerTrust.

3.3 Trust Management with P-Grid

While the above trust models address the issue of peer trustworthiness, they don't offer a clear solution on how to store and maintain the *behavioral data* in a decentralized manner. P-Grid[7] was proposed as part of a more complete solution to trust management in P2P[8], where each peer *i* can file a complaint about peer *j* by associating the complaint value with a **k** - a key which corresponds to the identifier of peer *j*. These complaints are stored under assigned peers, though the same complaints data can be stored in several peers, to ensure redundancy. When asking for the trustworthiness of peer *j*, peer *i* formulates a query - containing the key **k** and sends it to the peers in the network. The query is forwarded until the peers responsible for the complaint values are found, who in turn reply with their stored feedback on peer *j*. As more than one peer is responsible for storing complaints about peer *j*, if peer *i* receives the same copies of complaints from multiple peers for peer *j*, then it is safe to assume the reputation of the peer *j*. Consequently, a higher number of complaint replicas and the complex decision criterion that the algorithm implements significantly increase the quality of assessment.

4 Applications of Trust Models

4.1 GNUnet

GNUnet [4] is a framework for secure and anonymous peer-to-peer networking that makes no use of centralized or trusted services. However, the distributed and anonymous nature of GNUnet file-sharing system, where any entity can join the network and no entity is trusted, opens this P2P network to various attacks and abuse of resources. Since host behavior is not monitored in GNUnet, malicious participants can make use of the P2P network without contributing back to it (*free-riding*), further opening the network to potential denial-of-service attacks. Such circumstances necessitate for a trust scheme which would identify the malicious peers - the ones that don't contribute to the network, and isolate them, in order to fairly allocate resources to honest peers.

Grothoff [4] presents an excess-based economic model based on *trust* as a currency - to ensure fair resource allocation in peer-to-peer networks based upon prior behavior of peers, as well as defend against malicious participants who aim to abuse such resources. Any other currency parameter, different from *trust*, would require the installation of a central trusted authority, to which a currency such as *money* would belong, thus contradicting with the design principles for a decentralized, anonymous P2P network. Moreover, even under the existence of a trusted authority, there would be no guarantee that all the participating peers would be delivering the money they promised, thus certain peers may not be able to regain all of their money. The implemented trust-based economic model in GNUnet is simple: no peer owns trust, but its earned trust is stored at other peers, based on actual interactions among them, request and reply exchanges respectively. This design consideration implies a basic assumption: peers that consistently contribute to the file-sharing network earn the trust of their peers, otherwise peers that have never communicated have no opinion about each other. The latter fact further indicates that trust is not *transitive*, such that in a scenario where node A trusts node B, and node C trusts node A, this does not suggest that node C will trust node B.

Based on the above principle, peers request for content by assigning a *priority* value to their requests; this value represents the amount of the requesting peer's trust which is to be reduced at the recipient peers. On the other hand, peers that reply to the request will earn trust at the requesting peer by the amount that was initially put on the corresponding request. Depending on the existing load on the recipient peers, they may however: reply to the requests without reducing trust on the sending peers; or drop the incoming requests with the lowest priority first, in order to serve the peers who were willing to risk the highest amount of trust for their requests. As a result, each node constructs a local view of trust for each peer, and in the future, it will decide whether to serve that peer or not based on: the trust information it has about the requesting peer, and its own existing load. Moreover, the latter design principle reveals an additional positive property of this process: the P2P network gets infused with *trust* when the network or peer has excess resources, by not charging for its services. Finally, considering that n bytes of excess resources are used to infuse trust in the network, an attacking peer can not decrease network's performance so long as the bytes it uses are smaller than n.

The above model, however, arises an important dilemma: what if peers increase or decrease trust of other peers arbitrarily? Grothoff [4] argues that modifying trust values arbitrarily would have a negative impact on the performance of the peers themselves, since they may be accidentally increasing trust of a malicious peer or of a peer that may never be beneficial to them. An additional unacceptable behavior could occur under this model: a group of malicious nodes collaborating in order to trick a set of intermediary nodes into providing resources for free. Such a scenario could be avoided if the intermediary nodes would charge for the services - such that the sum of the priorities of the forwarded requests is less than the priority of the received request.

Assuming the above attacks are eliminated, the trust-based model [4] is assumed to deny resources to peers that abuse the network and therefore optimally allocate them to honest peers. On the other hand, to address the issue of inauthentic file-sharing in GNUnet, more specifically, the content verification, an encoding scheme [2] has been proposed. Under this content encoding scheme, the P2P content is encrypted with a key **k**, where **k** = H(content). While, under this implementation, the end-nodes can verify the authenticity of the files they download, the scheme as well enables the intermediaries to verify that a particular reply matches its relevant query, without having the intermediaries decrypt the responses.

4.2 BitTorrent

The functionality and the performance of P2P file-sharing systems, including that of BitTorrent, depends on the sharing of resources among peers. The higher the *sharing-ratio* enforced in the system, the higher the performance for all the peers in the network. BitTorrent implements a *Tit-for-Tat* policy [1] which seeks for pareto efficiency, in other words, gives the users within a swarm incentives to share in order to increase their performance. The BitTorrent protocol creates a tit-for-tat exchange of data between peers, based on their short-term bandwidth capabilities. In addition, *Tit-for-Tat*'s optimistic unchoking property offers the new-coming peers a chance to get their first pieces of data and this way bootstrap in the process. Unfortunately, under the Tit-for-Tat policy, a continued upload cannot be used to apply a tit-for-tat data exchange in other downloads, therefore clearly discouraging peers from sharing after their download has completed [11].

A P2P file-sharing system performs well when all the participating peers are willing to share content which, most importantly, is not malicious. Almost all of the P2P file-sharing systems deploy some form of a *reputation* mechanism, with the goal of enforcing an appropriate sharing-ratio in the system. In the following section, a secure and distributed reputation mechanism deployed in a BitTorrent-based P2P system - Tribler [5], aiming to enforce a balanced sharing-ratio, is presented.

4.2.1 BitTorrent-based Tribler

BitTorrent-based Tribler [5] makes use of BarterCast[11] protocol - a distributed mechanism for reputation management. In Bartercast, the history of direct uploads and downloads among peers is spread in the network using a gossiping protocol; peers store statistics locally - to create a local view of the data transfer in the network. Using these statistics, each peer can calculate the direct data it has exchanged with other peers; consequently, each peer computes the reputation of other peers. However, the information reported by other peers on their actual direct uploads and downloads may not be reliable, since not all peers in the network are honest. To eliminate the security issue where peers report false feedback, *maxflow-algorithm* [12] is integrated within the reputation scheme; the algorithm computes the maximum flow between two peers over all possible ways, thus giving an approximation for the contribution of each peer in the system. Maxflow-algorithm identifies lying peers, colluders, and hitchhikers; as a result, a peer can gain high reputation if and only if it has uploaded to the direct peer or to peers with high reputation values. None of these malicious peers will be successful in reporting false upload statistics, which are assumed to derive from direct upload and downloads, as the maximal flow between peers will always be computed and checked against the reported values for consistency.

According to Meulpolder et al [11], the performed experiments, in a simulation environment of a hundred active peers in ten different swarms during one week, have shown a decrease in reputation for free-riders and eventually a decrease in their download speed; though the probability that a peer with a high reputation value at one peer might have a low reputation at another peer is not excluded. Moreover, during the first days of the experiment, free-riders had a higher speed than the sharers, possibly due to their initial higher upload bandwidth capability, which in Tit-for-Tat translates into a higher download speed.

5 Conclusion

This paper presented few of the existing reputation-based trust models which are used to address the issue of inauthentic resource attacks in P2P networks, but most importantly, identify and isolate malicious peers from the network. Though different in design details, such trust mechanisms help identify the malicious peers and isolate them from the network, by assigning them low trust values. The reputation systems were further evaluated for their performance and efficiency in reducing the number of inauthentic file downloads. Finally, using practical examples, this paper has presented the effectiveness of such trust-based schemes in reducing the number of free-riders, therefore in increasing the fair resource allocation to the honest peers.

References

[1] B. Cohen. Incentives build robustness in bittorrent. In *In Workshop on Economics of Peer-to-Peer Systems*, May 2003.

[2] K. Bennett, C. Grotho, T. Horozov, and J. T. Lindgren. An Encoding for Censorship-Resistant Sharing, 2003.

[3] R. Chen, X. Zhao, L. Tang, J. Hu, and Z. Chen. *CuboidTrust: A Global Reputation-Based Trust Model in Peer-to-Peer Networks*. Springer Berlin / Heidelberg, 4th edition, 2007.

[4] C. Grothoff. An Excess-Based Economic Model for Resource Allocation in Peer-to-Peer Networks. Wirtschaftsinformatik, 2003.

[5] J. A. Pouwelse and P. Garbacki and J. Wang and A. Bakker and J. Yang and A. Iosup and D. H. J. Epema and M. Reinders and M. Van Steen and H. J. Sips. Tribler: A social-based peer-to-peer system. In *Concurrency and Computation: Practice and Experience*, 2007.

[6] Jingtao Li and Xueping Wang and Bing Liu and Qian Wang and Gendu Zhang. A reputation management scheme based on global trust model for peer-to-peer virtual communities. In *Advances in Web-Age Information Management*, volume 4016, 2006.

[7] Karl Aberer. P-grid: A self-organizing access structure for p2p information systems. In *In CoopIS*, pages 179–194, 2001.

[8] Karl Aberer and Zoran Despotovic. Managing trust in a peer-2-peer information system. In *CIKM '01: Proceedings of the tenth international conference on Information and knowledge management*, pages 310–317. ACM, 2001.

[9] Krit Wongrujira and Aruna Seneviratne. Monetary incentive with reputation for virtual market-place based P2P. In *CoNEXT '05: Proceedings of the 2005 ACM conference on Emerging network experiment and technology*, pages 135–145. ACM, 2005.

[10] Li Xiong and Ling Liu and Ieee Computer Society. Peertrust: supporting reputation-based trust for peer-to-peer electronic communities. In *IEEE Transactions on Knowledge and Data Engineering*, pages 843–857, 2004.

[11] M. Meulpolder, J. Pouwelse, D. Epema, and H. Sips. BarterCast: Fully Distributed Sharing-Ratio Enforcement in BitTorrent. Technical Report PDS-2008-002, Delft University of Technology, 2008. `http://pds.twi.tudelft.nl/`.

[12] Michal Feldman and Kevin Lai and Ion Stoica and John Chuang. Robust incentive techniques for peer-to-peer networks. In *EC '04: Proceedings of the 5th ACM conference on Electronic commerce*, pages 102–111. ACM, 2004.

[13] Robert Morris and David Karger and M. Frans Kaashoek and Hari Balakrishnan. Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications. pages 149–160, 2001.

[14] Sepandar D. Kamvar and Mario T. Schlosser and Hector Garcia-molina. The EigenTrust Algorithm for Reputation Management in P2P Networks. In *In Proceedings of the Twelfth International World Wide Web Conference*, pages 640–651. ACM Press, 2003.

[15] Yuan Wang and Ye Tao and Ping Yu and Feng Xu and Jian L. A trust evolution model for p2p networks. In *ATC*, pages 216–225, 2007.

RFID Systems - The Security Viewpoint

Tommi Häkkinen
Helsinki University of Technology
Tommi.Hakkinen@hut.fi

Abstract

Radio frequency identification (RFID) systems have gained an important role in automated wireless identification of objects and people. RFID tags are cheap to produce and it is estimated that in the future their usage will increase to trillions and eventually almost all objects from cars to milk bottles have an identification tag. It is undisputable that RFID technology enables a great deal of new innovations and improvements to the daily life, but at the same time they have significant security implications. This paper surveys the current security challenges of RFID systems, and suggested cryptographic and non cryptographics solutions.

KEYWORDS: RFID Security, RFID Privacy, RFID and Cryptographic primitives.

1 Introduction

Radio Frequency identification is a technology based on the need to remotely and automatically identify objects. RFID systems have been used already for many years, but there still exists some major unresolved security issues.

The prevailing problem is that the privacy and confidentiality of an RFID user can not be guaranteed with a generally agreed standard. The main challenge in implementing a secure RFID system that could meet all the security objectives, has been to embedd classical cryptographic primitives in low cost tags.

The section 2 of this survey gives an overview of the RFID technology and related standards. In section 3, the most critical security threats and risks are summarised. Finally, in sections 4 and 5 the state of the art cryptographic solutions are introduced and analysed.

2 Overview of Radio Frequency Identification

2.1 Fundamentals

Basic RFID systems consist of tags, which behave as transponders, and readers. RFID tag stores a simple identification number, which usually is either 96 or 128 bits long. Every tag contains a microchip, capacitor and an antenna coil [14]. With these elements, tags can communicate wirelessly with readers and identify themselves. This is a very important feature, since now items like grocery can be identified without a line of sight (compare barcode). Speciality of the RFID compared to other wireless communication systems is the cost. Because of the simplicity, RFID tags can easily be mass produced, which drives the unit price down. In 2005, American retailer WalMart started to utilize RFID massively in their products. This created an immediate demand for 1 billion tags and the unit costs dropped to 5-10 US cents [16]. Computing power and communication range of low cost RFID tags are limited, which sets challenges for suitable security solutions.

2.2 Transponders and Readers

Both the RFID tag and reader are two-way radio interfaces. They transfer information to both directions on radio frequency band that is either low frequency (LF), high frequence(HF) or ultrahigh frequency (UHF). LF tags normally operate at 125 kHz or 134 kHz. Normally HF tags uses frequency of 13,56 MHz and UHF tags frequencies between 866 MHz and 960 MHz [3].

An RFID reader is controlling the communication of the system and acts as a gateway between the tag and the backend system, if there is one. The Reader is connected to high efficiency antenna that broadcasts data and power to tag. Normally, RFID systems have a backend host that can be, for example, a database of a spare part catalog. The antenna of the reader generates an electromagnetic field with a relatively small radius that usually is some meters. When an RFID tag enters the range of the reader, the tag is activated and it can transfer information to reader.

The energy transfer can be based on two different couplings, inductive coupling or backscatter coupling. The former technique is used with frequencies below 30 Mhz. The magnetic field generated by the reader, induces a voltage in the coil of the tag, which transfers data back to the reader by using load modulation. Latter technique, backscatter coupling is used for frequencies above 100 Mhz. Tag reflects back the energy that reader produced. Reflected, or echoed signal is modulated with the transferred data [14].

RFID tags can be divided in three main categories: passive, semi-passive and active tags [3]. Passive tags are the most common ones and they are cheapest to produce. They do not have their own power sources and they communicate only when in close presence of the reader. Passive tags draw their energy from the electric field produced by the reader. Because passive tags do not have own energy source the operations they can perform are somewhat limited compared to active tags. Although, there are these evident limitations that passive tags have, they can do more than merely identify themselves to reader. The microchip can contain writable memory for storing and processing data.

An active RFID tag has own battery, which is used to power the microchip and broadcast the signal to reader. Thanks to the power source, more powerful tags can communicate up to 1 kilometer [3] and their battery lifes can be as long as 10 years. More powerful back channel signal level lowers the error rate of the communication, which in some cases is a problem with passive tags. Internal power source of active tags enables that they can contain substantially more resources, like memory and logical gates, and thus allow richer applications. In this paper we concentrate mainly on cryptographic applications and their requirements. Currently the downside with active tags is their higher producing cost and bigger size, which restricts their adoption in large scale.

Third tag type, semi-passive tag, is categorized between the passive and active tag. They have their own power sources, which are used only for powering the microchip but not broadcasting the signal. Signal is transferred with the same technique as with passive tags but the power needed to activate semi-passive tag is much lover. This increases the sensitivity level and thus lowers the bit error rate. Another advantage compared to the active tag is the lower power consumption. This is because the active tag does not use any extra power for broadcasting the signal to reader [3].

Depending on the type of the tag, they can have different alternatives to activate their microcontrollers. Since the passive tag can not operate without external energy it is activated only inside the range of the electric field that reader produces. Two other tag types can activate themselves also outside the range of the reader and thus operate independently.

2.3 Standards

There exists a wide variety of different standards covering RFID systems. Typically those standards define aspects such as the air inteface (how tags and readers communicate), communication protocol, data content and security protocols. Many commercial and nonprofit organizations describe RFID standards for different needs and purposes. This section covers a set of noteworthy standards from the International Organization for Standardization (ISO) and EPCglobal.

2.3.1 Identification cards

Contactless integrated circuit cards are categorized according to their communication range to three different types. ISO 10536 standard defines close coupled cards. These cards operates below one centimeter distance of the reader. Proximity cards (ISO 14443) have an operation range up to one meter and Vicinity cards (ISO 15693) greater than meter [20].

2.3.2 Item management

ISO 15961 RFID for Item Management standard addresses the common functional commands and syntax features for data protocol. ISO 15962 defines data encoding rules and logical memory functions that is used to exchange information for item management. ISO 15963 is a standard of unique identification of RFID tag. It defines the numbering system, registration procedure and the traceablity of tags during their usable lifetimes. ISO 18000 standard aims to ensure that the air interface protocol is universal. It describes how tags and readers communicate in different frequency bands [1].

2.3.3 Electronic Product Code

EPCglobal is a joint venture of EAN international and the Uniform Code Council [20]. It leads the development of industry-driven standards for the electronic product code. EPC is a 96-bit number that consist of: EPC version number, domains, object classes and individual instances. The purpose of the the so called EPCglobal framework is to provide the ability to trace the products through the production and supply chain [2].

3 Security Risks and Threats

Automated identification is gaining more and more attention, and it is estimated that the adoption of RFID technology will increase drastically in the next coming years. This fact emphasizes the importance of premium security analysis and awareness of the potential security issues. Even though, RFID security has been quite active research area in last few years, there are still some open fundamental issues to be addressed. This section introduces the most significant security risks and threats that must be considered with every RFID implementation.

3.1 Privacy

Privacy is the most fundamental concern with RFID systems. There are two main categories of privacy issues: clandenstine tracking and inventorying [12]. In the following, we will discuss about these two elemental problems and why RFID systems are so vulnerable for them.

The basic operation principle of RFID tags is that they are always on. This means that whenever there is a reader within the reading range, the tag will respond to reader interrogation without acknowledging the user. And because most RFID tags emit their unique identifier number during the reading process, it is possible for clandenstine reader to scan all the tags inside their coverage area. Even though, the reading range of the tag is in most cases very short, it is possible that reader can emit high amount of power and thus extend the reading distance.

A person who is carrying a tag, is constantly exposed to external tracking since the tag usually can not be switched off. The privacy is even more threatened when the serial number of the tag is combined with some personal information [12]. One example of this kind of tag is electronic passport. Even the data with cryptographic protection is vulnerable for tracking. When a tag exposes personal information about the carrier, it is possible to track where and when the person moves. This kind of information can be very valuable for marketing purposes.

Certain tag types contain also information about the items they are attached to. Typically these EPC tags provide information about the manufacturer and the product code of the object. Persons who are carrying these EPC tags are subject to inventorying without their knowledge. A reader can scan the items person has, and log that information to different registries. In theory, it is possible to collect information about where person shops, what medications she is using, what are her cloth sizes and so forth. For the moment, the preceding scenario is not that relevant because there does not yet exist an extensive RFID network that would enable large scale tracking and inventorying. But while the RFID infrastructure expands, it is inevitable that these security deficiencys will be exploited [3].

3.2 Counterfeiting (cloning)

Counterfeiting of products is a severe problem for many industries. Companies' that have invested extensively on their product development and research, are increasingly intrested to protect their intangible assets. RFID technology is considered as a potential anti-counterfeiting measure against the illicit manufacturers [6].

The main principle with RFID tags is that they provide an unique identifier for items or persons. This principle is compromised if the uniqueness can not be guaranteed. Basic RFID tags, which do not contain cryptographic functionalities, are vulnerable to counterfeiting attacks. The system memory of the tag can be scanned and replicated easily. With the cloned tag, it is possible to impersonate other person and, for example, gain access to restricted areas. In [22], Jonathan Westhues describes how he created a low cost device that read and simulate commercial proximity cards. The capabilities of low cost RFID tags are rather limited compared to illicit actors, who are expected to be able to perform extensive experiments with the elements of the system [6].

Even though, it is recognized that the resistance to tag cloning is limited, unique identification of objects can significantly reduce counterfeiting. Unique identifiers helps to monitor the flow of the good and thus to detect if duplicate id occurs [12].

3.3 Eavesdropping

Transmission between a reader and tag takes place over an insecure channel, which may be eavesdropped by any third party reader. Low cost RFID tags provide little resistance against that an illicit party can monitor the conversation and obtain security sensitive information.

Eavesdropping can be either active or passive. Passive eavesdropping may be performed inside the operating range of the tag and reader. And as the name suggests, the illicit third party observes and records the communication, but does not initiate communication. In active eavesdropping, a third party scans tags within the malicious scanning range and attempts to read the contents without authorisation [6].

RFID tags are constantly endangered to man-in-the-middle or relay attacks. Third party can monitor the conversation between a tag and reader, and retransmit an altered message. Mutual authentication of a low cost tag and reader is an active research area. So far, a common de facto standard has not been recognized, even though many different authentication protocols are proposed [6, 12, 13].

3.4 Denial of Service

Denial of service (DoS) is an attack against the system availablity. An attack can easily be carried out by placing a large number of virtual labels for a reader to be identified. Malicious device can overload the reader by simulating the operation of tags, and prevent the legimate communcation. It is also possible to perform DoS against tags. Attacker may, for example, repeatedly ask identification of label, thus making them unaccessible for authorized readers.

One form of denial of service attack is to jamm the air interface by creating noise in the frequency band in use. In the worst case, the blocked channel may have devastating effects in critical systems [6].

3.5 Tag Killing

Tag Killing can be considered as one sort of denial of service attack. It aims to wipe out the functionality of a tag and make it useless. The idea of killing a tag is simple. An adversary sends a lot of queries to the tag, which starts new authentication session for every request. If the target tag uses Ohkubo type protocol [19], it can not send responses to reader after the maximum n answers have been accomplished. For challenge-response type protocols [17], tag killing attack aims to exhaust the memory of the tag. For example, if an adversary reader sends 2^{20} queries to tag, the the tag should store 2^{20} random numbers, thus requiring memory about 10MB [10].

In this context, it is worthwhile to mention that tag killing is not only a hostile method, but also a potential answer to some privacy issues. Tags can be designed to destroy themselves intentionally, so they can not be tracked or inventoried clandestinely.

4 Security Mechanisms

4.1 Security Objectives

As presented in previous section, RFID systems are vulnerable to several security threats. To address these issues, mechanisms must be implemented to achieve the security objectives. Table 1 lists the security objectives that must be taken into consideration when designing an RFID system. An effective security mechanism can fulfil all objectives but in practise, it rarely is sensible to do so. The potential security threats should be identified for each RFID solution separately. Normally, the cost is the main factor in automated identification and that states the level of security [6].

Security Property
Confidentiality
Integrity
Availability
Authenticity
Anonymity

Table 1: List of security properties

In RFID techology, the confidentiality objective describes the mechanism to secure sensitive information between a reader and tag. Confidentiality can be achieved by encrypting the air interface channel, and thus preventing an illicit party to eavesdropp the communication [14, 6].

Securing the data integrity involves guaranteeing that the communicated information is not tampered by an unauthorized party. When the data integrity is ensured with message authentication codes, man-in-the middle attack can be prevented. Although, the computation power of present low cost RFID tags is so limited that there are no means of providing content security [6].

Availability of RFID systems is very important issue, since readers and tags must always be ready to detect each other when entering inside the communication range. The availability can easily be disturbed with a frequency jamming as described in section 4.4. RFID systems meeting the availability objective will ensure that service is always in place.

The authenticity of a tag is the key objective in secure RFID systems. In general, the unique identifier of a tag is not tamper resistant. Without a proper crytographic mechanism, an illicit party can steal the identity of a tag and gain access to sensitive information.

The last security objective discussed in this paper is anonymity. The privacy of a user is considered as the biggest security threat in RFID technology. When anonymity objective is met, consumer tracing and inventorying should not be possible. Anonymity can be achieved, for example, with physical shielding or blocking tags.

4.2 Cryptographic Primitives

Providing security and privacy for RFID systems requires some cryptographic primitives implemented in suitable protocols. Primitives used in RFID cryptosysems can be divided in three categories: hash functions, symmetric and asymmetric encryption [6].

Hash functions are the most widely used proposals to solve the security related problems in RFID systems. They are based on symmetric keys that are shared between the verifier and tag. In the following, several general RFID authentication protocols are introduced.

4.2.1 Hash Lock

In this simple security scheme [21], each tag stores the hash of a random key K as the meta-ID (meta-ID = hash(K)) of tag. When a reader queries a tag, the tag responds with its meta-ID. Backend system uses the meta-ID to find a proper key from database and sends the key back to tag. The tag hashes the key and compares it to the meta-ID. Since the key is sent in open channel, it can be captured and used to spoof the reader [15].

4.2.2 Randomized Hash Lock

Randomized hash lock scheme is an extension to the previous solution (hash lock) [21]. Each tags has unique ID and random number generator. The tag gets a random number r for every session and calculates a hash number c (c=hash(ID,r)). Random number r, and hash number c are transmitted to reader. Backend uses the r to calculate hash numbers for all IDs stored in database. When a match is found, the reader authenticates itself by sending the matched ID to the tag [20].

4.2.3 Hash Chain

The hash chain method [18] is dynamic authentication protocol that uses two different hash functions to create authentication information. Every tag has an initial unique value $S_{t,i}$ that is used to calculate the return hash code $a_{t,i} = G(S_{t,i})$ to a reader. When the return hash code is calculated and transmitted to reader, the tag uses second hash function H to update a new confidential value $S_{t,i+1} = H(S_{t,i})$. RFID system that uses this authentication has a certain maximum length of hash chain that represents the maximum number of times to read a tag. Hash chain protocol satisfies confidentiality, integrity and anonymity properties, but its downside is the high computing power that is needed both at the backend and tag.

4.2.4 Light Weight Authentication

Light Weight security model [11] is not a hash based scheme, but since it is a simple authentication protocol, it can be discussed in this context. This protocol introduces a challenge-response mechanism, which uses only simple XOR operations and no gryptographic primitives [15]. The model relies on carefully synchronized pseudonum rotation. A tag authenticates to verifier only after the verifier has first authenticated to the tag. The verifier transmits a pseudonym key β_i that is unique to a pseudonym key α_i received from the tag.

Asymmetric authentication protocol utilizes a public and private keys to authenticate the communicating parties. Authentication process can be initiated by both a reader and tag, For example, the reader encrypts an authentication message with the public key of the tag. The tag decrypts the message with its own private key and transmits the same message back to the reader encrypted with reader's pubcic key. The reader then decrypts the received message and verifies it was the initial message. General asymmetric encryption solutions require relatively high computing power, and thus are inefficient for RFID systems [3]. Cui et al. [23] proposed a lightweight asymmetric authentication protocols for RFID systems.

Authentication protocol provides a mechanism to identify the communicating parties, but it does not solve the security problem related to eavesdropping. As the wireless communication is vulnerable for adversary to steal sensitive information, the information channel must be secured by encrypting the traffic. A modern 128-bit symmetric-key encryption module requires more than 100,000 logical gates [3], which is considerably more than appropriate for RFID tags . Thus, the encryption implementations for RFID must trade off encryption strength against the microchip complexity. Several studies have suggested different ecryption techniques suitable for low cost RFID systems. Feldhofer et al. proposed a lighweight symmetric-key implementation [8] that is based on 128-bit AES. Their solution consumes 4.5 μW of power and requires approximately 4400 logical gates, which is too much for low cost tags but suitable for mid cost RFID tags.

Asymmetric key encryption is a significant challenge for RFID systems. It requires more complex algorithms compared to symmetric encryption, but also provides many advantages. An elliptic curve cryptography (ECC) proposed in [4], is a public key processor for low cost RFID systems. The processor has arithmetic unit and and memory. It requires 6300-7800 gates, depending on the length of the key. Asymmetric encryption techniques for RFID systems are currently under active research and it is justified to expect that within next few years, strong enryption methods will be utilised in large scale. Moore's law predicts that the number of transistors on microchips douples every two years. Thus, the unit costs of public key tags are expected to decline accordingly.

4.3 Shielding and Blocking

In addition to cryptographic implementations, the privacy of RFID users can also be protected with a quite different approach. One very practical solution is the Faraday Cage. The simplest way of implementing a Faraday Cage is to wrap an RFID tag to an aluminum foil. This isolates tags from any kind of electromagnetic waves. However, there are limits how many tags can be protected in this way. For example, tags in clothing is almost impossible the protect in this manner. Shielding of RFID tags is a quick fix for yet unsolved privacy issues, but it is not a realistic long term solution.

One suggestion to address the privacy issue is the kill command. In this scheme, the tag could be deactivated permanently with an unique password [5]. It is argued that kill command is not an effective solution to the privacy problem, while, for example, it does not give an answer to in-store tracking.

Killing the tag deactivates the tag forever, while there might be a need only to block the identification for some period of time. RSA Laboratories developed an alternative method to protect the privacy: the blocker tag [7]. The blocker tag is able to block the functioning of a reader by broadcasting the whole scale of identification codes. It creates a secure zone of couple of meters around the device. However, the blocker device prevents the functioning of all tags, including those that are not wanted to be blocked. Blocker tag is considered more like a niche solution for those who want actively protect their privacy.

4.4 Bill of Rights

An RFID expert, Simon Garfinkle has proposed so-called Bill of Rights, a set of regulations that would serve as a framework that companies should voluntary follow [9, 16]. It consist of five rights that are:

1. The right to know whether products contains RFID tags.

2. The right to have RFID tags removed or deactivated when products are purchased.

3. The right to use RFID enabled services without RFID tags.

4. The right to access a data stored on RFID tag.

5. The right to know when, where and why tags are read.

These five abovementiond rights are not likely to turn into law, but they are more like guidelines for companies wishing to deploy RFID technology. Consumers could then boycott companies that violate these rights. Without a commonly accepted principles, there is a significant risk that manufacturers might try to misuse the RFID technology, and thus compromise the consumer privacy.

5 Conclusion

RFID systems have already been used for many years in large scale of applications. It is expected that their utlization will increase drastically during the coming years. There still remain a number of security related issues that should be resolved and standards that should be harmonized.

A lot of effort has been put to a research of RFID cryptosystems and many cryptographic protocols are proposed. So far, the manufacturing costs have been too high to include strong cryptosystems in low cost tags. The constanst increase in the computing power of microchips and the decrease in their unit costs will undoubtly enable that strong cryptosystems can be integrated in low cost tags.

After the biggest technical and legislative issues concerning privacy and confidentiality are resolved, it is evident that both consumers and manufacturers will benefit significantly from the potentiality of RFID technology.

References

[1] International organization for standardization. `www.iso.org`.

[2] Epcglobal, 2005. `www.epcglobalinc.org`.

[3] S. Ahson and M. Ilyas. *RFID Handbook: Applications, Technology, Security and Privacy*. CRC Press, 2008.

[4] L. Batina, J. Guajardo, T. Kerins, N. M. P. Tuyls, and I. Verbauwhede. An elliptic curve processor suitable for rfid-tags. cryptology eprint archive, report 2006/227, 2006.

[5] A.-I. Center. 900 mhz class 0 readio frequency (rf) identification tag specification. draft march 2003.

[6] P. H. Cole and D. C. Ranasinghe. *Networked RFID systems and Lightweight Cryptography*, chapter Anti-counterfeiting and RFID, pages 32–43. Springer, 2008.

[7] A. J. et al. The blocker tag: Selective blocking of rfid tags for consumer privacy. in v. atluri e. 8th acm conference on computer and communications. acm press.

[8] M. Feldhofer, S. Dominikus, and J. Wolkerstorfer. Strong authentication for rfid systems using the aes algorithm, proceedings of ches 2004, lecture notes on computer science, vol. 3156, pp. 357-370, 2004.

[9] S. Garfinkel. Application, security and privacy. bill of rights, center for research on computation and society harvard university, 2005.

[10] D.-G. Han, T. Takagi, H. Kim, and K. Chung. *Computational Science and Its Applications - ICCSA 2006*, chapter New Security Problem in RFID Systems ag Killing, pages 375–384. Springer Berlin / Heidelberg, 2006.

[11] A. Juels. Minimalist cryptography for low-cost rfid tags. in: Blundo, c. cimato, s. (eds.) scn 2004. lncs, vol. 3352 pp. 149-164. springer, heidelberg (2005).

[12] A. Juels. Rfid security and privacy: A research survey. *Journal of Selected Areas in Communication (J-SAC)*, 24(2), 2006.

[13] J. Kim, H. Oh, and S. Kim. *Information Security Practice and Experience*, chapter A New Hash-Based RFID Mutual Authentication Protocol Providing Enhanced User Privacy Protection, pages 278–289. Springer, 2008.

[14] H. Knospe and H. Pohl. Rfid security. *Information Security Technical Report*, 5(4), 2004.

[15] Y.-Z. Li, Y.-B. Cho, N.-K. Um, and S.-H. Lee. *Computational Intelligence and Security*, chapter Security and Privacy on Authentication Protocol for Low-Cost RFID, pages 788–794. Springer Berlin / Heidelberg, 2007.

[16] V. Lockton and R. S. Rosenberg. Rfid: The next serious threat to privacy. *Ethics and Information Technology*, 7(4), 2005. `http://dx.doi.org/10.1007/s10676-006-0014-2`.

[17] M. Molnar and D. Wagner. *Privacy and security in library RFID*, chapter Issues, practices and architectures, pages 210–219. ACM Press, 2004.

[18] Ohkubo, M. Suzuki, and K. Kinoshita. Efficient hash-chain based rfid privacy protection scheme. int. conf. on ubiquious computing, workshop privacy: Current status and future directions.

[19] M. Ohkubo, K. Suzuki, and S. Kinoshita. Cryptograhpic approach to "privacy-friendly"' tags, in rfid privacy workshop, mit, usa 2003.

[20] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, and A. Ribagorda. *Personal Wireless Communications*, chapter RFID Systems: A Survey on Security Threats and Proposed Solutions, pages 159–170. Springer Berlin / Heidelberg, 2006.

[21] S. Weis, S. Sarma, R. Rivest, and D. Engels. Security and privacy aspects of low-cost radio frequency identification systems. in security in pervasive comp., volume 2802 of lcns.

[22] J. Westhues, S. Garfinkel, and B. Rosenberg. *RFID: Applications, Security, and Privacy*, chapter Hacking the prox card, pages 291–300. Addison-Wesley, 2005.

[23] K. K. Y. Cui, K. Matsuura, and H. Imai. Lightweight asymmetric privacy-preserving authentication protocols secure against active attack, proceedings of the pervasive computing and communications workshop, pp. 223-228, 2007.

Protection for Resource Constrained Devices

Prajwalan Karanjit
Helsinki University of Technology
pkaranji@cc.hut.fi

Abstract

Pervasive computing is an essential aspect of next generation computing. A lot of applications such as safety monitoring of buildings and huge spaces, tracking environmental pollution, health, scientific survey etc. are being developed based on several types of resource constrained devices. On one hand, these devices are providing some very unique services, while there is a huge potential that these devices may be compromised and misused by an adversary. Security options available for high end computers, needs modifications to be implemented for these resource limited devices. This paper, hence, presents a state of art survey on current researches on several techniques that can be implemented so as to build a security system for these devices. These techniques include key management, accelerating the cryptographic operations and specialized means of detecting attacks.

KEYWORDS: Resource constrained, Pervasive computing

1 Introduction

Internet is making rapid development and along with this, a lot of applications are being built over small devices. These devices are fully capable of networking and provide some unique services to the real world. Network smart card with built-in microprocessor [11, 19] is an example of such a device. Such smart cards are widely used as debit and credit cards, access control cards and health insurance. Sensor nodes are another popular resource limited devices. They are designed to measure certain phenomenon like temperature, vibration, light intensity etc. for over a large range of period. It then collects the data and transmits to the base station. It may send this data continuously with certain time interval, or may only send when it detects a change in phenomenon such as, change in temperature or light intensity etc. A network of such sensor nodes provides application to fields like scientific survey, health, military and other simillar areas.

It is possible for an adversary to obtain critical data by monitoring a sensor network, fabricate a false signal, modify the actual signals or even act as a fake sensor node. Besides these, several other types of attacks like attacking routing messages so that whole network gets disrupted, denial of service attacks, side channel attacks and many others are also possible. So, it is very necessary to provide protection means. However, the security methods perfectly suitable for large computers are not suitable for these small devices. There are several reasons to this.

One of the most prominent reasons is the resource limitation. Resource here basically means CPU capability, memory and power (mainly battery power). Also the bandwidth availability is limited [9]. Although there are some devices with microcontrollers that are capable of running advanced software like Java Virtual Machine [28], majority have low memory and CPU specifications. For example, ATmega128L, one of the common microcontrollers used in many sensor nodes, like those from MICAz Wireless Measurement System [8], has 8 MHz CPU, 8 bit word size, 4 KB RAM and 128 KB non volatile memory. Most widely used cryptographic algorithms like AES [32, 24] are designed for 32 bit CPUs and if used in lower bit CPUs then the performance is degraded. ATmega128L has 8 bit CPU, so AES will show degraded performance if applied directly. Also, algorithms like RSA demand a lot of CPU cycles and memory. This is because, for good security, RSA requires a large key size, such as 512 bit or 1024 bit and this increase both time and space complexity. Demand for more CPU cycles and need to read and write larger memory space means more engery will be required.

Most of these resource constrained devices, particularly sensor nodes, communicate through wireless channel. Many protocols and algorithms designed for wireless ad-hoc networks are not suitable for sensor networks. Despite this, wireless seems to be only the solution to incorporate a large number of sensor nodes into a network. A wired implementation could be unscalable and impractical. However, this makes it difficult to protect sensitive data. Ad-hoc nature makes it vulnerable in all layers of protocol stack [34] and implementing security is even more challenging. Particularly, secure routing and key management issues are far more complicated due to ad-hoc nature [20]. Security mechanisms based on cryptographic algorithms are required to deal with it. But, resource limited environment makes the situation peculiar.

Sensor network is basically a large scale deployment and the communication tends to be one to many or many to many or many to one. [25] highlights three categories of communication patterns, namely, node to base station, base station to some particular node and base station to all nodes. Current security standards are designed for two party communications i.e. one to one [29]. This does not scale properly when the scenario is other than one to one. Also usually the base station is taken as a trusted entity. As such it is likely that security mechanisms will be based on base station. Hence, there is high probability that base station will be bottlenecked [29, 5]. Besides these, there are other vulnerabilities like physical attack on the devices. Usually, sensor nodes are deployed in remote and unattained areas. So, it is possible that the devices may be physically harmed or may get stolen.

In many implementations of resource limited devices, and not just in sensor networks, the topology is not known prior to deployment. Even after deployment, the devices may be added or removed, thus affecting the topology. This unknown topology is also a problem with such networks [4].

Hence, it is extremely difficult to implement security in a network consisting of resource limited devices. Many of the current mechanisms require significant modifications, before they could be applied. It is very important to examine several options available to us so that these options could be tailored further. Objective of this paper is hence, to survey current researches going on in this field.

Rest of the paper is organized as follows. Section 2 presents some of the security properties required by networks of resource limited devices. Section 3 discusses the key management issues and how applicable the solutions might be for both sensor networks and smart cards. Section 4 discusses some of the methods to enchance the cryptographic operations. Section 5 gives an insight on a different attack detection technique for resource constrained devices. Section 6 discusses other issues like secure routing and node revocation. And, finally, Section 7 concludes the work.

2 Security Requirements

Typically a secure system has following four basic requirements. First is confidentiality, which means the information is accessible only to authorized entities and not to others. Second is authentication, which is a property of proving the identity of entities involved. Third is integrity, which means the information should remain unmodified through the communication channel. Finally, fourth is availability, which means that the system should be accessible to the authorized entity whenever required.

These requirements apply well in context of resource limited device networking also. Confidentiality is needed so that a network may not leak any sensitive data. The channel must be secure and it should be very difficult, if not impossible, for an eavesdropper to monitor the channel. Similarly, authentication is also important. A third party can act as fake node in the network and perform attacks such as report fabrication or man in the middle. The data received from sensor network can be very critical and hence, it is very important to ensure that they are coming from trusted source. It is equally important that the data remains unmodified. So, data integrity is also necessary. Finally, the term availability means that the nodes as well as base station should be available for each other at any time necessary. This indicates protection against denial of service attacks.

As well as, there are some other unique requirements. Data sent by devices could be measurements or payment related information or some other critical information, in case of smart cards. Hence, freshness of data is also equally important [25]. Furthermore, capability to provide service in case of failure due to power or attack i.e. survivability and the ability to change the security level as requirement changes, are also important [4].

Possibility to detect malicious packets is also an important requirement. Traditional method of running a separate process for firewall is not applicable for resource limited devices [22]. Single process, i.e. the main process itself should have built it feature capable of separating between genuine packet and malicious packet.

In order to meet all these requirements, certain cryptographic operations need to be performed. But, as specified in the section 1, the devices have limited resources. So, certain architecture must be built to as to speed up the operations as well as not to overload the devices.

Besides these, there are other practical requirements like scalability in terms of number of keys and number of nodes, easiness in deployment of the system and possibility to add or remove nodes in future [10].

3 Key Management

Keys are crucial to any cryptographic operations. A slight compromise in selection of key or in mechanism of distributing it, might bring down even the most secure algorithms. It is worth nothing that even algorithms like RSA is secure only until we implement it correctly with right choice of keys [27] and of course, it must be distributed securely in a scalable way. So, key management is an important issue directly involved in any security mechanism.

There are different approaches for key management. This section further describes these approaches in context of resource constrained devices, particularly sensor nodes.

3.1 Single Key Approach

Simplest method will be to have one single key in all the nodes throughout the network. The nodes then will establish the communication link with the base station using this key. It has several advantages. Because of the single key, the memory requirement is low. The design of hardware as well as the software becomes very simple. It can even result into a power efficient design and provide resistance against certain types of denial of service attacks [6]. However, if this key is compromised or say the node is compromised, then the entire network will be compromised and not just that particular node.

Thus single key approach is generally used for initial key derivation purposes only, such as for determining pair-wise keys for later communications [10].

3.2 Public Key Approach

Public Key based approach for key distribution is widely used in modern computer networks. In context of networks involving resource limited devices also, it might be feasible. In this method, a master key pair (public and private) is generated before deployment [6]. The nodes will also generate their own key pair. In each node, a signature of master key is also placed. This signature is later used to verify the authenticity of public key of other nodes. After deployment, nodes exchange each with other their public keys, which is then verified with master's signature. Then after knowing each other's public key, nodes can create and exchange the session keys.

As long as the public key algorithm is used properly with carefully chosen keys, this method will result in a very secure system. The system is also scalable as the scheme works effectively regardless of the number of nodes in network [6]. Further, it has other desirable properties like resistant against node capture, possibility of ignoring compromised nodes [18]. In comparison to RSA, Elliptic Curve Cryptography (ECC) is a good candidate as a public key algorithm for resource constrained devices [16, 15]. ECC provide security of similar strength but with small key size than that of RSA, and hence less computational expense. [33] presents a password authenticated key exchange protocol for secure communication between two resource constrained wireless devices, based on ECC.

3.3 Centralized Approach

In centralized approach, devices request to a central trusted entity (Key Distribution Center or KDC) for the session key. The devices must also authenticate themselves to the KDC before it could provide them with the session key. In case of sensor networks, base station can play the role of KDC [6, 18]. Primary advantage of this approach is simplicity in computation as compared to public key based technique. Also less memory is required as the key size is usually 128 or 256 bits, which is again less than that required by algorithms like RSA. Session keys are used for short term. If a new communication starts then a different session key is required. So, in case a malicious or a compromised node is detected, then the central entity can ignore the request of those nodes. As such, it can easy to implement techniques like node revocation. But there is the problem of single point of failure. Different fault tolerance approaches might be required to compensate this. As well as there can be scalability issues, particularly because of communication overhead [6]. Also as KDC is required for every new session keys it can become a target of attack.

SPINS [25] is one such architecture based on this approach. It provides two building blocks, namely, SNEP and μTESLA. SNEP is designed for two party communications and provides data confidentiality, data authentication and data freshness properties. μTESLA is designed for broadcasting messages in network of resource limited devices.

3.4 Random Key Pre-Distribution Approach

Random key pre-distribution was first proposed by Eschenauer and Gligor in 2002 [13]. A universal key space is considered, from which, nodes receive a subset of keys. The members of this subset are randomly chosen. If sufficient numbers of random numbers are chosen for each subset, then there is high probability that each of the two nodes have at least one key common. Nodes will then go through key discovery phase where they attempt to find other nodes with which they share the key. This can be done by exchanging some key dependent signature with neighboring nodes. Alternatively, puzzles might be exchanges and nodes solving the puzzles correctly can be considered to have same key as the puzzle originator. Inspired by this idea, some new proposals have also been made [4, 10, 7].

The pair-wise key approach [6] is resilient against node capture, as each shared key is unique and capture of any node does not allow adversary to eavesdrop communication between other nodes. If a range extension of just two hops is considered, we can get network sizes comparable to the other schemes [7]. So, it is quiet scalable. It also supports node revocation and rekeying [18].

A comparative summary of above discussed key distribution techniques is shown in Table 1. Last column of the table indicates whether, the approach is suitable for smart cards as well. For smart cards, due to the nature of their application, random key pre-distribution is unsuitable. Single key approach is also vulnerable for the same reasons as with the sensor nodes. However, centralized key distribution and most importantly, hybrid encryption that involves public key and symmetric session key seems to be the most suitable for smart cards. For smart cards, heavy RSA based operations like X.509 certificate verifications are not much of problem. The reason is that these cards are mostly used for authentication purpose. They rarely operate all the time like sensor nodes. The only process that runs is for entity authentication and also for some secure data transfer but this data transfer is limited. So, unlike sensor nodes, smart cards can afford resource demanding operations.

It is clear that among these many methods of distributing keys, it is pointless to argue which one is better than whom. Each of these methods has their own advantages and disadvantages. So depending upon the application and size of network, as well as by taking future growth into account, one should select appropriate key distribution mechanism. Section 4 discusses approaches that may make public key cryptography efficient for several resource limited devices, as well as enhance the symmetric cryptography.

4 Enhancing Cryptographic Operations

Several attempts have been made so as to boost the heavy operations in low resource environment. These attempts propose to carry out the operations like Public Key Cryptography (PKC) consuming less memory, less CPU cycles and less energy. In general three categories of such attempts are being researched. First is to rebuild the processor and associated software either by changing the way they work or by introducing a separate co-processor. Second is to explore the possibility of using different Public Key Schemes other than RSA. And the last one is outsourcing the cryptographic operations to some remote high-end server/proxy. These approaches are discussed below.

4.1 Specialized Hardware/Software

The idea here is to allow the main processor to perform the major operations and to introduce a secondary co-processor for cryptographic operations. [31] presents a co-processor designed for Advanced Encryption Standard (AES). The paper claims that their design is resistant against first order dif-

Approach	Scalability	Node Capture	Node Revocation	Resource Requirement	Smart Card
Single Key	High	No Resilience	Not Possible	Very Less	Unsuitable
Centralized	Medium	Resilience	Possible	Less	Suitable
Public Key	High	Resilience	Possible	High/Medium	Suitable
Random	High	Resilience	Possible	Medium	Unsuitable

Table 1: A comparision of key management approaches

ferential analysis by embedding a data masking countermeasures at hardware level. The authors have performed several simulations in order to verify their claim.

While accelerating the operations through the use of coprocessor seems to be an elegant approach, some are considering software and hardware optimization techniques for RSA and ECC. [16] presents such techniques based on implementations on two exemplary 8-bit microcontroller platforms: The 8051-based Chipcon CC1010 and the Atmel AVR ATmega128. The main advantage that we gain by such optimized techniques in comparison to co-processor based one is reduced hardware complexity and cost. So, such techniques can be beneficial for low cost implementations, in particular.

4.2 Alternative Public Key Cryptography

While RSA remains the most promising PKC algorithm for high end devices like computers, several other PKC algorithms are becoming good candidates for low resource devices. The reason is, of course, the heavy computation demand of RSA. Alternative schemes such as Rabin's schemes [26], NtruEncrypt [14] and Elliptic Curve Cryptography (ECC) are gaining importance. Studies on these selected low complexity PKC schemes show that these could be suitable for different types of applications[15]. All these seem to be the most promising candidates for resources limited devices. Several other papers also advocate on suitability of ECC [16, 15, 17].

Table 2 shows a brief comparison of execution times and memory consumed by ECC and RSA on an 8 bit CPU, ATmega128. The specifications of ECC i.e. secp160rl and secp192rl consists of both public key and private key along with global parameters [1]. It can be clearly seen that execution time for ECC is much less compared to RSA.

4.3 Outsourcing Cryptographic Operations

Another way to enhance heavy cryptographic operations could be to let some remote high-end machine do these operations on behalf of resource constrained devices. These devices will communicate to a centralized security server before sending or receiving packets from a non trusted network entity. All the cryptographic operations for most of the communications will be outsourced to this server. This approach, if implemented correctly, could minimize the resource usage by a great factor.

[30] presents a solution that utilizes the SSL as the security provider and uses SOCKS to redirect the traffic accordingly through such a security server. Secure Socket Layer (SSL), application protocol independent, provides the ability to establish private communication channel in a public network. In SSL, both the client and server are required to encrypt their data with one of the keys in an asymmetric key pair and decrypt with the other key of the pair. SOCKS performs at Session layer of the OSI reference model. It is a generic proxy protocol for TCP/IP-based networking applications. The protocol provides a flexible framework for developing secure communications by easily integrating other security technologies. With the usage of SOCKS, the resource constrained device will launch a connection to the remote secure server in the trusted network and then this server will establish a secure connection to any of the servers or users located in the not trusted network.

Results obtained by using such remote proxy are very promising. The analysis presented in [30] shows that running 1024 bit RSA algorithm is more than 13 times faster with this approach. Similar performance gain was there with AES algorithm also.

Usage of using remote proxy can go beyond just providing security. [2] explores possibility of implementing applications like natural language processing, speech recognition, face recognition over resource limited devices. The paper presents the design of Chroma, a tactics based remote execution system, and shows that it is able to achieve application performance that is comparable to execution on an ideal runtime system. This definitely has its challenges, but if successfully implemented then devices will not only be able to secure communicate but they will be able to provide smarter services too.

Public Key Cryptography tremendously simplifies the implementation of many typical security services and additionally reduces transmission power due to less protocol overhead [15]. Moreover, the capture of a single node would not compromise the entire network, since no globally shared secrets are stored on it. With the smart implementation approaches that involves either optimizing the existing hardware and software or introducing dedicated co-processor or even by outsourcing, PKC can be heavily used in resource constrained devices.

5 Attack Detection

Traditional approaches of detecting attacks on high end computers are also unsuitable for resource limited devices. The main reason for this is that they run all the time and as a separate process. Devices having limited resources cannot afford this luxury. So, a completely different detection method is required that runs as a part of the main process.

As a general idea, attack detection should be performed

Algorithm	Time (sec)	Memory (bytes)	Code (bytes)
ECC secp160r1	0.81	282	3682
ECC secp192r1	1.24	336	3979
RSA-1024 public-key e = 2^{16} + 1	0.43	542	1073
RSA-1024 private-key w. CRT	10.99	930	6292

Table 2: Average ECC and RSA execution times on the ATmega128 [16]

by the main process, at each layer as the data flows from physical layer up to the application layer. Simillar multilevel stage packet filtering technique is proposed in [22]. The method described runs as a part of TCP/IP stack, providing real time detection and thus conserving memory space. A packet typically goes through one layer of protocol stack after another until it is passed or dropped. The idea is to detect the malicious packets as early as possible. On top of that, the monitoring code runs only when the code execution passes through there. This alleviates the necessity for it to run all the time, thus resulting into a better security, reduced memory usage and enhanced performance. The method can, possibly, be used as a basis for revoking the compromised or infected nodes. However, implementation of such a technique would require modification of entire software. Further, the method will only drop the malicious packets upon detection of an attack. The system proposed in [22] does not describe how it will inform the base station about an attack. So, until attack is informed and compromised node is revoked, the bandwidth which is already limited, will only be wasted. The packets are declared malicious based on some measures. These measures could be heuristic based also. But there is a probability that even genuine packets are dropped.

There is one class of attacks that is quiet specific to such resource constrained devices. This attack is called battery exhaustion attack. Another similar attack is denial of sleep attack. As the devices such as sensor nodes, have limited battery, such attacks can effectively result into denial of service. However, it is also possible that battery exhaustion is just a side effect of some other attack and hence, could be used to detect these other attacks. For example, if a node is made to transmit a number of useless packets or a lot of packets are made to route through a particular node, then that node is likely to run out of battery very quickly. Battery exhaustion detection techniques then might be able to identify such attacks and hence locate the compromised node. Battery-Sensing Intrusion Protection System (B-SIPS) [3], which alerts on power changes detected on small wireless devices, is one such technique. B-SIPS uses dynamic threshold calculation algorithm. This system is scalable and complementary with existing detection systems. The system is also capable of sending notification to some central authority. This makes it useful tool for the base station or key distribution center to decide if the node is compromised and initialize node revocation operation.

Certain attacks and their corresponding countermeasures could be relevant among different types of resource constrained devices. This relevancy is quiet stronger in case of smart cards and sensor nodes [12]. The routing attacks and DoS attacks that are possible in sensor network are also applicable to smart cards. Similarly, attacks such as IC reverse engineering, side channel attacks, micro-probing, interception of RF communications, jamming RF communications that are possible in smart cards are also applicable in sensor network. The respective countermeasures are also relevant to some extent. Such correlation can be an effective tool in understanding new possible threats and further enhancing the attack detection techniques.

6 Other Issues

This section discusses some other issues such as secure routing and node revocation. Both of these are more concerned with sensor networks.

6.1 Secure Routing

Several network layer attacks against sensor networks are possible. Most of these include, spoofing, altering, or replaying the routing information, selective forwarding, sinkhole attacks, sybil attacks, wormholes, HELLO flood attacks, acknowledgement spoofing [21]. Thus it is extremely important to consider a secure routing protocol, which should enable communication despite adversarial activities like these. Some protocols like directed diffusion and geographic routing assume trusted environment, which is usually not the case in a sensor network. Some protocols are based on public key cryptography and while some are based upon symmetric cryptography. Protocols like Ariadne, designed for ad hoc networks, prevent compromised routes consisting of uncompromised nodes, and also prevent a large number of DoS attacks [29]. However, such protocols may not be suitable directly for sensor networks and also other resource constrained device networks. [21] presents a detailed analytical study on this field.

6.2 Node Revocation

It is very important to eliminate a compromised node and restrict it from further participation. Key pre-distribution techniques have been found useful for this [7, 23]. In most of key pre-distribution mechanisms, the keying material or the key itself are associated with a node with a unique identifier. In order to revoke a given key or keying material, then it is enough to publish the identifier of the associated node in the revocation list to be revoked [23]. Attack detection techniques like the one discussed in [22], can also provide a basis for identifying a compromised node and help in revocation.

7 Conclusion

With an increase in widespread use of applications based on resouce limited devices, security issues have become a central concern. However, protection mechanisms should be such that they will optimize the available resource and bandwidth. This paper has presented a survey on "state-of-art" resource constrained device related security technologies that are currently available and that are under research. Particularly, survey and analysis on key management issues, ways to enchance cryptographic operations and attack detection techniques have been covered. This paper should provide a reference for building secure application on top of resource limited devices.

References

[1] The standards for efficient cryptography group (secg). `http://www.secg.org`.

[2] R. K. Balan, M. Satyanarayanan, S. Park, and T. Okoshi. Tactics based remote execution for mobile computing. Technical report, Carnegie Mellon University and Intel Research Pittsburgh, 2000.

[3] T. K. Buennemeyer, M. Gora, R. C. Marchany, and J. G. Tront. Battery exhaustion attack detection with small handheld mobile computers. IEEE, 2007.

[4] S. A. CAMTEPE and B. YENER. Key distribution mechanisms for wireless sensor networks: a survey. Technical report, Rensselaer Polytechnic Institute, March 2005.

[5] L. Chaithanya, M. Singh, and M. Gore. Secure data management in reactive sensor networks. In *ICISS 2006, LNCS 4332*. Springer-Verlag Berlin Heidelberg, 2006.

[6] H. Chan, A. Perrig, and D. Song. Chapter 1 key distribution techniques for sensor networks.

[7] H. Chan, A. Perrig, and D. Song. Random key predistribution schemes for sensor networks. In *In IEEE Symposium on Security and Privacy*, pages 197–213, 2003.

[8] Crossbow. Micaz datasheet. `http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MICAz_Datasheet.pdf`.

[9] A. Deligiannakis, Y. Kotidis, and N. Roussopoulos. Bandwidth-constrained queries in sensor networks. *The VLDB Journal, Springer Berlin / Heidelberg*, 2006.

[10] T. Dimitriou and I. Krontiris. A localized, distributed protocol for secure information exchange in sensor networks. In *19th IEEE International Parallel and Distributed Processing Symposium*. IEEE, 2005.

[11] DotDistribution. Microprocessor smartcards. `http://www.dotdistribution.com/shop/ilp/se~5/p/index.shtml`.

[12] K. Eagles, K. Markantonakis, and K. Mayes. A comparative analysis of common threats, vulnerabilities, attacks and countermeasures within smart card and wireless sensor network node technologies. In *WISTP 2007, LNCS 4462*. IFIP International Federation for Information Processing, 2007.

[13] L. Eschenauer and V. D. Gligor. A key-management scheme for distributed sensor networks. In *In Proceedings of the 9th ACM Conference on Computer and Communications Security*, pages 41–47. ACM Press, 2002.

[14] G. Gaubatz, J. peter Kaps, and B. Sunar. Public key cryptography in sensor networks - revisited. In *In 1st European Workshop on Security in Ad-Hoc and Sensor Networks (ESAS 2004*, pages 2–18, 2004.

[15] G. Gaubatz, J. peter Kaps, E. Öztürk, and B. Sunar. State of the art in ultra-low power public key cryptography for wireless sensor networks. In *In 2nd IEEE International Workshop on Pervasive Computing and Communication Security (PerSec 2005), Kauai Island*, pages 146–150. IEEE Computer Society, 2005.

[16] N. Gura, A. Patel, A. W, H. Eberle, and S. C. Shantz. Comparing elliptic curve cryptography and rsa on 8-bit cpus. pages 119–132, 2004.

[17] T. Hasegawa, J. Nakajima, and M. Matsui. A practical implementation of elliptic curve cryptosystems over gf (p) on a 16-bit microcomputer. In *In Public Key Cryptography PKC Š98, volume 1431 of Lecture Notes in Computer Science*, page 182Ű194. Springer-Verlag, 1998.

[18] P. Hämäläinen, M. Kuorilehto, T. Alho, M. Hännikäinen, and T. D. Hämäläinen. Security in wireless sensor networks: Considerations and experiments. In *LNCS 4017*. Springer, 2006.

[19] HowStuffWorks.com. What is a smart card? `http://computer.howstuffworks.com/question332.htm`.

[20] J.-P. Hubaux, L. Buttyan, and S. Capkun. The quest for security in mobile ad hoc networks. In *Mobile Ad-hoc Networking and Computing*. ACM, 2001.

[21] C. Karlof and D. Wagner. Secure routing in wireless sensor networks: Attacks and countermeasures. In *In First IEEE International Workshop on Sensor Network Protocols and Applications*, pages 113–127, 2003.

[22] H. K. Lu. Attack detection for resource-constrained network devices. In *Third International Conference on Systems*. IEEE, 2008.

[23] Y. Maeng, A. Mohaisen, and D. Nyang. Secret key revocation in sensor networks. In *UIC 2007, LNCS 4611*. Springer-Verlag Berlin Heidelberg, 2007.

[24] NIST. Aes alogrithm (rijndael) information. `http://csrc.nist.gov/archive/aes/rijndael/wsdindex.html`.

[25] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. D. Tygar. Spins: Security protocols for sensor networks. In *Wireless Networks*, pages 189–199, 2001.

[26] M. O. Rabin. Digitalized signatures and public key functions as intractable as factorization. Technical report, Massachusetts Institute of Technology, 1979.

[27] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21:120–126, 1978.

[28] R. Roman, C. Alcaraz, and J. Lopez. A survey of cryptographic primitives and implementations for hardware-constrained sensor network nodes. In *Science + Business Media, LLC*. Springer, 2007.

[29] E. SHI and A. PERRIG. Designing secure sensor networks. In *WIRELESS SENSOR NETWORKS*. IEEE, December 2004.

[30] Y.-S. They, S.-Y. Phang, S. Lee, H. Lee, and H. Lim. Cpop: Cryptography process offloading proxy for resource constrained devices. In *International Conference on Information Security and Assurance*. IEEE, 2008.

[31] E. Trichina and T. Korkishko. Secure aes hardware module for resource constrained devices. In *ESAS 2004, LNCS 3313*. Springer-Verlag Berlin Heidelberg, 2005.

[32] Wikipedia. Advanced encryption standard. `http://en.wikipedia.org/w/index.php?title=Advanced_Encryption_Standard&oldid=249106597`, 2008 (accessed 2-November-2008).

[33] D. S. Wong, A. H. Chan, and F. Zhu. Password authenticated key exchange for resource-constrained wireless communications (extended abstract). In *ICN 2005, LNCS 3421*, pages 827–834. Springer-Verlag Berlin Heidelberg, 2005.

[34] H. YANG, H. LUO, F. YE, S. LU, and L. ZHANG. Security in mobile ad hoc networks:challenges and solutions. In *TOPICS IN WIRELESS SECURITY*. IEEE, February 2004.

Suitability of Open Source Solutions for Electronic Voting Systems

Mikko Niemi
Helsinki University of Technology
mjniemi@cc.hut.fi

Abstract

In this paper we examine the suitability of the open source software development model for producing electronic voting systems. We discuss what kind of security requirements an electronic voting scheme needs to fulfill, we examine the risks and benefits of full source code disclosure, and we consider how well or badly existing open source business models support the development of electronic voting systems. We find that the requirements are very difficult to satisfy, and while an open source approach has some benefits, there are some drawbacks as well. We conclude that an active community of volunteers is important for maximizing the benefits of open source development, but whether an e-voting software project is attractive enough for volunteer developers remains an open question.

KEYWORDS: electronic voting; open source software; information security

1 Introduction

Electronic voting, often also known as e-voting, has become a seriously debated subject since the turn of the millennium. E-voting promises several benefits over traditional paper ballots: the results of an election would be available almost instantly when the polls close; organizing elections would become cheaper since fewer election officials would be needed for conducting the vote; electronic voting machines can be designed to be accessible to people with disabilities and to accommodate multiple languages.

Despite the possibilities offered by electronic voting, many open questions remain regarding the security and transparency of such voting schemes. Researchers have found several weaknesses in existing commercial e-voting systems [1, 12, 17]. For example, in paper ballot voting, a single corrupt election official would not be able to easily manipulate the result of the whole election. In a computerized system however, malicious code could spread automatically and invisibly like a virus. This kind of attack has been demonstrated by the Center for Information Technology Policy at Princeton University [5].

One approach for increasing the transparency in electronic voting schemes is making the source code available to the public. However, the effects this has on security are by no means self-evident. On the one hand, the more people there are scrutinizing the source code, the more likely it is that a vulnerability is found. On the other hand, access to the source code makes the attacker's job easier as well, and while the defenders need to find all the security holes, the attacker only needs to find one effectively exploitable. Many open source projects have been successful in security critical areas. Such projects include for example the Sendmail mail transfer agent, the MySQL relational database system, the Linux operating system and the Apache web server. Nevertheless, development of a complete secure electronic voting system poses some unique challenges, which makes it difficult to compare to other software projects.

There are two fundamentally different kinds of electronic voting schemes. In poll-site voting schemes the voting terminals are located at the polling stations, where the voting takes place under the supervision of the election officials. The other type of electronic voting scheme is remote voting, where the voters may cast their votes, for example, over an Internet connection or by telephone. In this paper we consider issues that are pertinent to both types of electronic voting schemes.

2 Security and Other Requirements in Electronic Voting

The classification of requirements in this section is mostly based on the framework for comparison of electronic voting schemes described by Sampigethaya and Poovendran [18]. In their framework, the requirements of electronic voting schemes are organized in three categories. General security requirements are those requirements that a secure system must satisfy before even considering adversarial attacks. Requirements for resilience against attacks are categorized under adversary counter-attack requirements. Finally, fulfilling system implementation requirements ensures that the voting scheme is implementable in practice. Other requirements that are not addressed by them are listed here under miscellaneous requirements.

2.1 General Security Requirements

- Eligibility: Effective mechanisms exist for ensuring that only valid voters are allowed to vote and each voter gets to cast only the permitted number of votes. The authentication and uniqueness requirements are covered by this definition of eligibility. [3, 8, 18, 20, 29]

- Secrecy and Privacy: The voting scheme enables a secret ballot where a vote, after it has been cast, can not be traced back to the voter. With maximal privacy, a voter's privacy may be breached only by collusion of all the other entities (voters and authorities). [2, 3, 8, 18, 20, 29]

- Verifiability: A system that satisfies individual verifiability requirement enables a voter to assure herself that her vote was correctly recorded and counted in the final tally. In the case of universal verifiability, anyone can verify that the result was correctly computed from the valid votes and that the tally process was accurate. This requirement conflicts with the secrecy and privacy requirements. [3, 8, 18, 20]

- Accuracy, Integrity and Correctness: All the correctly cast votes are recorded and counted in the final tally. Invalid voters' votes are not counted. Votes can not be deleted, modified or forged without detection. These requirements are closely related to the universal verifiability requirement. [2, 3, 18, 20, 29]

- Fairness: A partial tally can not be exposed while the election is still in progress. [18, 20, 29]

2.2 Adversary Counter-attack Requirements

- Robustness and Reliability: The voting scheme must be resilient against attacks from corrupt voters and authorities. The system as a whole should be successful even in the event of a fault leading to a partial failure. [3, 8, 18, 20, 29]

- Receipt-freeness: The voter must not receive or be able to generate a receipt that could be used to prove to other entities the content of her vote. This requirement is related to the privacy requirement and untraceability. [18, 20]

- Incoercibility: An adversary must not be able to manipulate a voter to cast her vote in a certain way, nor should it be possible to coerce a voter to abstain from voting. [2, 3, 8, 18, 29]

2.3 System Implementation Requirements

- Scalability and Efficiency: The voting scheme must be scalable in terms of storage, computation and communication needs to be implementable in a large scale. [18, 29]

- Practicability: The voting scheme must not make any unrealistic assumptions or have requirements that would be difficult to implement. The voting scheme must be cost-effective. [3, 18]

2.4 Miscellaneous Requirements

- Flexibility: The election equipment must be compatible with existing standards and flexible enough to allow various different formats of ballot questions. [3]

- Usability and Accessibility: The election equipment must be easy to use and accessible to people with disabilities. [3, 8]

- Certifiability: It must be possible to test the system so that it can be ensured to meet the set criteria. [3]

- Transparency: It must be possible for the voters to understand how the voting scheme works. [3, 8]

3 Risks and Benefits of Open Source

Use of open source licensing in electronic voting software has some clear benefits, but also some serious drawbacks. The main and most obvious benefit is the increased transparency, which in turn increases trust to the system [13]. Other benefits include freedom from many of the licensing and intellectual property issues that may cause problems with the use and maintenance of proprietary e-voting systems.

Considering how little experience there is of open source e-voting software, it is not clear whether an open source project would attract significant contributions from volunteers. The danger is, that if there is not a large and active enough community performing peer review, publishing the source code gives greater benefits to the attackers than to the defenders.

3.1 Benefits of Open Source

A scheme where security is based on the secrecy of the implementation is often called security through obscurity [25]. In the field of computer security, it is well known and widely accepted, that security through obscurity is not a viable principle. If the attacker is a trusted insider, or if there is a leak that exposes the systems inner design to an adversary, security through obscurity fails. Furthermore, in a transparent scheme where the implementation can be peer reviewed, flaws and vulnerabilities are more likely to be found before deployment. The most prominent example of favoring transparency over obscurity comes from the field of cryptography. In cryptography, the idea that a cryptographic system must be secure even if the details of its implementation fall in the hands of the attackers, is known as Kerckhoffs' principle [25]. A well designed cryptographic system uses well known public algorithms that have gone through extensive analysis by the cryptographic community. In such a scheme, the only secrets that need to be kept are the secret keys, such as pass phrases.

An open source approach to producing an e-voting scheme would seemingly exclude security through obscurity. Since the source code would be available to the public, it would not be possible to use solutions that rely on the secrecy of the implementation. This would force the design to use published and well tested algorithms instead of secret proprietary solutions that may or may not be secure. The main argument that commercial e-voting vendors use for keeping the source code closed is protection of trade secrets, not hiding the inner workings of the voting scheme from would-be attackers [26]. Yet, some independent third party analysis of the source code is the only way to convince the public that an electronic voting scheme is secure and leaves no chance for effecting the result of an election through a security flaw or a back door [14]. While a third party source code analysis can be accomplished with limited disclosure of the code, the most convincing approach would be publishing the whole source code under some open source license for the whole world to examine.

There are also some other ways in which open source can improve security. Simply knowing that the code is going to be available for others to scrutinize may motivate the developers to write cleaner and more human readable code that is easier to debug and maintain. When flaws are found, open source enables anyone with programming skills to find the exact problem and submit a solution. This is quite the opposite approach compared to proprietary solutions where intellectual property claims can be used to silence criticism [6]. As an example, in 2004, a number of students were served with cease-and-desist letters after publishing internal emails showing that Diebold Inc. had been using uncertified software on their voting machines [28].

In addition to hindering independent researchers' attempts to analyze and expose security vulnerabilities, proprietary licenses can also be used to lock-in customers to a certain vendor. To do this, the vendor only needs to use licensing that is strict enough to leave the vendor in control of maintaining, servicing and further developing the system. With this much dependency, it becomes very expensive for the customer to change to a different vendor. However, changing the vendor might become necessary, if for example the original vendor goes out of business. In this event, the customer would likely have to invest into a completely new system. With open source licensing, the situation could be very different. If the original vendor withdrew from the market or the support fees became exorbitant, the customer could choose a different provider, which could then continue the development of the original software.

Finally, there is the argument of open source being more cost effective than proprietary software. However, given the very limited experience from open source e-voting software, it is quite difficult to speculate on issues such as total cost of ownership. It is likely, though, that under open source licensing, savings could be made in at least the creation, modification and licensing of the software [21]. Additionally, use of open source might eliminate the costs and administrative workload associated with tracking the number of software copies in use, work that is typically necessary with proprietary software.

3.2 Risks and Open Questions

In cryptography, the peer review process works well and increases the trust for algorithms that pass the scrutiny. This is true, because there exists a large community of cryptographers who are eager to participate in examining new ideas. Breaking the newest encryption algorithm is considered a great scientific achievement. In many other areas however, it is not quite clear how significant a part peer review has in improving security.

As an example, it might be tempting for an open source advocate to claim that transparency is the main reason for the Linux operating system having a better security record than Microsoft's Windows. While there may be some truth in this claim, there are some other factors to consider as well. Firstly, Linux is based on the well designed user level security architecture of the Unix operating system. Secondly, Windows is much more commonly used, making it a more attractive target for many attackers.

Producing secure electronic voting software differs in some fundamental ways from other projects where the open source model has been successful. Experience has shown that the most attractive projects for community developers are those that aim to produce software that is used by programmers [15]. Unlike operating systems or web servers that are used daily, electronic voting software is used very rarely, perhaps only once every four or six years. Whether an e-voting project could attract an active community is still an open question. According to Jason Kitcat, the founder of a free e-democracy project, extensive media coverage and backing from major organizations were not enough to attract many open source developers to contribute to the GNU.FREE Internet Voting project led by him [15]. Without an active community, publishing the source code arguably only helps the adversaries and has no positive effects on security.

Source code transparency does not solve everything. Certain problems would still remain, even in the case where an open source e-voting project would be successful enough to draw a large community of qualified volunteers to perform peer review. Source code reviews could be circumvented, for example, by inserting the malicious code at compile time, using a compiler specifically crafted for this purpose. There would therefore be a need for some mechanism for guaranteeing that the published source code would not be secretly modified before it is installed on the voting machines. On the other hand, if a vulnerability is exposed shortly before the election day, some last minute patching might be necessary. Other alternatives would be falling back to paper ballots and manual counting of the votes, or rescheduling the election to a later date. Of course, postponing the election might be considered a successful attack. Moreover, controlling the last minute changes to the system is a major challenge, whether an open source license is used or not.

Another inadequacy with open source e-voting software is that while the application software is available for review, there are generally many other components in an electronic voting scheme that are not necessarily available for scrutiny. Implementation details of the hardware and the integrated code running it, firmware, are typically not released to the public. There may also be need for some proprietary binary drivers. Finally, even if an open source operating system is used, it is likely that large portions of the legacy code included is not going to be reviewed, given the size and complexity of operating systems [21]. While these are not arguments for proprietary e-voting schemes, they do show that using an open source model in the development of the application code is not enough to guarantee full transparency of the final e-voting system.

Commercial vendors of e-voting solutions argue that their source code contains valuable trade secrets that need to be protected [26]. If the transparency of the open source approach would be considered valuable enough to mandate all e-voting software source code disclosure to the public, it could mean that several e-voting providers would withdraw from the market. With them, their experience and expertise gained from high quality research would be lost [21].

4 Viability of Open Source Business Models

There have been numerous studies investigating why software developers participate in open source projects [7, 9, 10, 19]. Some have ideological reasons, such as the belief that software should not be a proprietary good. Limiting the power of large software companies by providing free alternatives is another ideological motive. Then there are more practical reasons, for example, the need for a solution for a problem for which the proprietary solutions were in some respect lacking, too expensive, or nonexistent. Some other significant reasons are self-improvement by learning and developing new skills, sharing knowledge and the joy of being a part of a community.

As was discussed in the previous section, there is a real chance that developing electronic voting software is not a very attractive project for open source developers. There is certainly a need for e-voting software. The failures of proprietary e-voting software also support ideological opinions favoring transparent open source solutions. However, unlike many successful community oriented open source projects, such as the Linux operating system, e-voting software is not software that can be used by the developers daily. In the case of Linux, fixing a bug or writing a new device driver has often immediate benefits for the individual developer. Effort committed to an e-voting software project, on the other hand, does not necessarily have very visible payoffs before the election day.

It may be unreasonable to expect a large number of volunteers to commit to an open source e-voting software project. For this reason, it is important to consider how viable the open source development model would be for commercial e-voting vendors before, for example, ruling public disclosure of all e-voting software source code mandatory. Many business models that take advantage of open source software have been identified [4, 16, 27]. In the rest of this chapter, we examine how well these various business models adapt to the field of electronic voting.

4.1 Selling Support and Services

Providing support services and consulting is one way of generating revenue from open source software. Possible service options include installation and integration support, technical and legal certification, training, ongoing maintenance and support services, and migration services [4, 16, 27]. The idea behind this business model is that a specialized company can provide these kind of services at a lower price than the cost of the customers doing it themselves.

Without an active community of volunteers, the initial costs of producing the software have to be covered by the developer company. The problem with this business model is, that once the source code is made publicly available, there is nothing to prevent the original developer's competitors entering the support market. In the area of electronic voting software, it may well be, that the initial development costs of the software and the risks involved are too high for this business model.

4.2 Proprietary Components

In this business model, majority of the software is published under an open source license but some important components are kept under a proprietary license [4, 27]. Revenue comes from selling the proprietary components and possibly also support services. This business model gives the developer company a clear advantage over its competitors on the support market. However, keeping essential functionality under a proprietary license may alienate some developers from the community.

With e-voting software, this business model seems more realistic than the support selling business model. It is essentially a compromise between transparency and protecting the intellectual property rights of the developer. The risk of this business model is, that a competitor could develop their own versions of the proprietary components, or those components might even be produced by open source developers. On the other hand, with e-voting software, the proprietary components would likely have to pass an expensive auditing and certification process. This would raise the cost of entering the market and probably discourage at least some competitors.

4.3 Selling Hardware

Providing free software may help in selling hardware [16]. For example, a company might release open source device drivers for Linux. By investing in the development of open source drivers, the company extends the market of the hardware. This business model is comparable to the traditional loss-leader commercial model where one product is sold at low cost, perhaps even at a loss, in order to stimulate other profitable sales.

The challenge of this business model is, that if the software is not specific enough for the hardware that the software developer is selling, then the competitors might, with minor adjustments, take advantage of the open source code to sell their hardware instead. This would most probably be the case with e-voting software, where significant amount of the source code would likely be independent of the underlying hardware.

4.4 Dual Licensing

Some open source licenses, such as the GNU General Public License, are said to be viral. This means that they set restrictions on the licensing schemes of software components that are combined with the open source components. Software under the GPL, for example, may not be linked to proprietary libraries. With the dual licensing business model, the developer company offers software under two different licenses, one of them an open source license, the other a proprietary license [4, 16, 27]. The latter one is targeted to customers who are willing to pay a fee for the license to use the open source software in combination with their own closed source proprietary software.

This business model seems poorly suited for e-voting software. Dual licensing works best with software components that can be easily combined with other software to build larger systems. It is unlikely that there exists a market for partial e-voting software solutions or independent e-voting software components. Additionally, if parts of the software are under a proprietary license, the transparency of the voting scheme suffers. On the other hand, if nobody is interested in buying the commercial license, dual licensing produces no revenue.

4.5 Advertising

Revenue from advertisement is a fairly new way of financing open source projects [16]. For example, in 2006, 85% of Mozilla Foundation's revenue came from fees payed by the search engine company Google, for being by default listed first in the quick search box of the Firefox web browser [11].

While the possibility of reaching the whole voting public would most certainly be of great interest to many advertisers, it is difficult to imagine that this kind of financing strategy could be seriously considered for e-voting. Corporate interests and commercial messages must be kept apart from the voting process. Advertisement would be very difficult to use for financing in a way that would not interfere with the democratic process, thus it seems clear that this business model can not be used with electronic voting software.

4.6 Public Funding

There may be cases where an open source project for developing some software would be considered valuable to the society, but unlikely to draw volunteers from the open source community. This situation may arise in specialized scientific areas, such as radio astronomy, computational chemistry or biology, where the developers often need to be experts on the subject in order to understand the problems being solved. One way to finance such projects is public funding through universities or national grants [4]. In this business model, the funding institution does not expect to profit directly from the investment.

Public funding would be a natural way to finance open source e-voting software, considering that in the end, the costs of voting are in any case covered by the taxpayers. The open question is, where to find the developers willing to commit to a publicly funded open source e-voting project. One possible idea is to leverage the knowledge in universities and other educational institutions. Including an e-voting project to the curriculum would help to expand the community of volunteer developers. Furthermore, students introduced to an e-voting software project during their studies might be interested in the project even after graduating.

5 Discussion

As we have seen, the requirements for a secure electronic voting scheme are numerous and, to some extent, conflicting. Protecting the integrity of the voting process requires that the results can be verified, but the privacy of individual voters must also be considered. One approach to implementing verification without weakening voter privacy is the use of voter verified paper audit trails [22]. In paper trail e-voting schemes, the voting machine prints out a record of the cast vote. The voter then verifies this paper ballot before the vote is recorded electronically and finally, the verified paper ballots are collected and safely stored. After the election, verification can be performed by comparing the tally of the electronic votes and the tally calculated from the paper ballots. The problem with paper ballot schemes is that calculating paper ballots is very expensive and time consuming. The efficiency of calculating the election results electronically is one of the most important benefits of e-voting. If the results are verified with a paper trail system, this benefit is mostly lost.

Securing electronic voting is difficult even when the voting machines are under the control of the election officials. When considering remote voting, where the voting system can be accessed through an Internet connection from a regular desktop computer, the challenges are much more difficult. For starters, there are no technical solutions for preventing vote selling or voter coercion in remote voting schemes where the voter and the actual event of voting can not be monitored by the election officials. Other problems with remote voting are, for example, securing availability, as public networks are vulnerable to denial of service type of attacks. Furthermore, it has been claimed that the software and hardware of computers used by the public are still too insecure for enabling secure Internet voting [24].

Adopting the open source development model does not change the fundamental challenges of implementing secure electronic voting systems. Open source has some benefits and some drawbacks. Increased transparency is the main benefit. If a voting scheme is based on proprietary components, the public can not be entirely assured of the accuracy of the voting process. In fact, a proprietary electronic voting scheme might not even need to be technically flawed for some negative effects. Suspicions and doubt of the systems integrity alone could erode public trust to the election process. With less trust, there is less incentive to vote, and lower voter turnout would directly harm the democratic process.

The big question is, how to encourage open source development of e-voting software. There are examples of open source projects, such as the Linux operating system, where much of the work is carried out by unfunded volunteer workers. This type of development model relies on the network effect of many volunteer developers personally committing to the project. Their motivation can stem, for example, from the developer's personal need for the software, ideological reasons or simply technical curiosity. Important for this kind of project seems to be, that the developers are rewarded by seeing the constant improvement in their work [23]. Development of e-voting software might not be rewarding enough to support the unfunded community based development model.

Companies such as MySQL and Red Hat have demonstrated that it is possible to use open source software as a basis for profitable business. However, it appears that most of the existing open source business models are not very well suited to the area of electronic voting. The question is, where do the open source alternatives come from, if an open source e-voting project is not attractive to community developers and there is no incentive for commercial software vendors to voluntarily move to the use of open source licensing. One solution is to force the vendors to publish their source code by legislation that mandates e-voting software to be open source licensed. Supporting open source projects with public funding is another alternative.

6 Conclusions

Satisfying the requirements for a secure electronic voting scheme is difficult. An open source licensing model does not change the fundamental challenges. The open source approach has the benefit that if the system is flawed, the public can find out about it. This, however, requires that there are enough people willing to commit time and effort to scrutinizing the published source code. How attractive an electronic voting project is to open source developers is still an open question. Without an active community of volunteers, the risks of publishing the source code may outweigh the benefits. Nevertheless, open source software in electronic voting systems is a worthwhile goal, because of the importance of transparency in democracy.

References

[1] A. W. Appel, M. Ginsburg, H. Hursti, B. W. Kernighan, C. D. Richards, and G. Tan. Insecurities and Inaccuracies of the Sequoia AVC Advantage 9.00H DRE Voting Machine. `http://coblitz.codeen.org/citp.princeton.edu/voting/advantage/advantage-insecurities-redacted.pdf`, October 2008.

[2] J. Benaloh and D. Tuinstra. Receipt-free secret-ballot elections (extended abstract). In *STOC '94: Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*, pages 544–553, New York, NY, USA, 1994. ACM.

[3] J. C. D. Mote. Report of the national workshop on internet voting: issues and research agenda. In *dg.o '00: Proceedings of the 2000 annual national conference on Digital government research*, pages 1–59. Digital Government Society of North America, 2000.

[4] C. Daffara. Guide for SMEs, April 2008. `http://flossmetrics.org/sections/deliverables/sections/deliverables/docs/deliverables/WP8/D8.1.2-SMEsGuide.pdf`.

[5] A. J. Feldman, J. A. Halderman, and E. W. Felten. Security Analysis of the Diebold AccuVote-TS Voting Machine. Technical report, Princeton University, September 2006. `http://citp.princeton.edu/pub/ts06full.pdf`.

[6] E. Felten. Interesting Email from Sequoia, March 2008. `http://freedom-to-tinker.com/blog/felten/interesting-email-sequoia`.

[7] R. Ghosh, R. Glott, B. Krieger, and G. Robles. Free/Libre and Open Source Software: Survey and Study, Part IV, June 2002. `http://www.infonomics.nl/FLOSS/report/FLOSS_Final4.pdf`.

[8] D. A. Gritzalis. Principles and requirements for a secure e-voting system. *Computers & Security*, 21(6):539–556, October 2002.

[9] A. Hars and S. Ou. Working for Free? Motivations for Participating in Open-Source Projects. *Int. J. Electron. Commerce*, 6(3):25–39, 2002.

[10] G. Hertel, S. Niedner, and S. Herrmann. Motivation of software developers in Open Source projects: an Internet-based survey of contributors to the Linux kernel. *Research Policy*, 32(7):1159–1177, 2003.

[11] Hood & Strong, LLP. Mozilla foundation and subsidiary – independent auditors' report and consolidated financial statements, 2006. `http://www.mozilla.org/foundation/documents/mf-2006-audited-financial-statement.pdf`.

[12] H. Hursti. Critical Security Issues with Diebold Optical Scan Design. *Black Box Voting Project, July*, 4, 2005. `http://www.blackboxvoting.org/BBVreport.pdf`.

[13] Joseph Hall. Transparency and Access to Source Code in E-Voting. In *USENIX/ACCURATE Electronic Voting Technology (EVT'06) Workshop*, 2006. `http://josephhall.org/papers/jhall_evt06.pdf`.

[14] A. M. Keller and D. Mertz. Privacy issues in an electronic voting machine. In *In Proceedings of the ACM Workshop on Privacy in the Electronic Society (WPES*, pages 33–34. ACM Press, 2004.

[15] J. Kitcat. Source availability and e-voting: an advocate recants. *Commun. ACM*, 47(10):65–67, 2004.

[16] J. Koenig. Seven open source business strategies for competitive advantage. *IT Manager's Journal*, May 2004.

[17] T. Kohno, A. Stubblefield, A. Rubin, and D. Wallach. Analysis of an Electronic Voting System. In *IEEE SYMPOSIUM ON SECURITY AND PRIVACY*, pages 27–42. IEEE COMPUTER SOCIETY, 2004.

[18] Krishna Sampigethaya and Radha Poovendran. A framework and taxonomy for comparison of electronic voting schemes. *Computers & Security*, 25(2):137–153, March 2006.

[19] K. R. Lakhani and R. G. Wolf. Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects. *SSRN eLibrary*, 2003.

[20] B. Lee and K. Kim. Receipt-free electronic voting through collaboration of voter and honest verifier. In *Proceedings of JW-ISC2000*, pages 101–108, January 2000.

[21] B. McPherson. A Report to the California Legislature on Open Source Software in Voting Systems, 2006. http://ss.ca.gov/elections/open_source_report.pdf.

[22] R. Mercuri. A Better Ballot Box? *IEEE Spectrum*, 39(10):46–50, 2002.

[23] E. Raymond. The Cathedral and the Bazaar. *Knowledge, Technology, and Policy*, 12(3):23–49, 1999.

[24] A. D. Rubin. Security considerations for remote electronic voting. *Commun. ACM*, 45(12):39–44, 2002.

[25] B. Schneier. Secrecy, Security, and Obscurity. http://www.schneier.com/crypto-gram-0205.html.

[26] M. Shamos. Paper v. Electronic Voting Records – An Assessment. In *Proceedings of the 14th ACM Conference on Computers, Freedom and Privacy*, 2004.

[27] D. Spiller and T. Wichmann. Free/Libre and Open Source Software: Survey and Study, Part III, June 2002. http://www.berlecon.de/studien/downloads/200207FLOSS_Basics.pdf.

[28] United States District Court, N.D. California, San Jose Division. OPG, Pavlovsky & Smith v. DIEBOLD, 337 F.Supp.2d 1195, 72 U.S.P.Q.2d 1200. http://www.eff.org/files/filenode/OPG_v_Diebold/OPG%20v.%20Diebold%20ruling.pdf.

[29] Yu-Yi Chen and Jinn-Ke Jan and Chin-Ling Chen. The design of a secure anonymous Internet voting system. *Computers & Security*, 23(4):330–337, June 2004.