TKK Dissertations 255
Espoo 2010

SECURING THE INTERNET WITH DIGITAL SIGNATURES

Doctoral Dissertation

Dmitrij Lagutin

Aalto University
School of Science and Technology
Faculty of Information and Natural Sciences
Department of Computer Science and Engineering

TKK Dissertations 255
Espoo 2010

SECURING THE INTERNET WITH DIGITAL SIGNATURES

Doctoral Dissertation

Dmitrij Lagutin

Doctoral dissertation for the degree of Doctor of Science in Technology to be presented with due permission of the Faculty of Information and Natural Sciences for public examination and debate in Auditorium T2 at the Aalto University School of Science and Technology (Espoo, Finland) on the 10th of December 2010 at 12 noon.

Aalto University
School of Science and Technology
Faculty of Information and Natural Sciences
Department of Computer Science and Engineering

Aalto-yliopisto
Teknillinen korkeakoulu
Informaatio- ja luonnontieteiden tiedekunta
Tietotekniikan laitos

Distribution:
Aalto University
School of Science and Technology
Faculty of Information and Natural Sciences
Department of Computer Science and Engineering
P.O. Box 15400 (Konemiehentie 2)
FI - 00076 Aalto
FINLAND
URL: http://www.cse.tkk.fi/
Tel. +358-9-470 23228
Fax +358-9-470 23293
E-mail: Dmitrij.Lagutin@tkk.fi

ISBN 978-952-60-3464-5
ISBN 978-952-60-3465-2 (PDF)
ISSN 1795-2239
ISSN 1795-4584 (PDF)
URL: http://lib.tkk.fi/Diss/2010/isbn9789526034652/

TKK-DISS-2841

Aalto-Print
Helsinki 2010

Aalto University

ABSTRACT OF DOCTORAL DISSERTATION	AALTO UNIVERSITY SCHOOL OF SCIENCE AND TECHNOLOGY P.O. BOX 11000, FI-00076 AALTO http://www.aalto.fi
Author Dmitrij Lagutin	
Name of the dissertation Securing the Internet with digital signatures	
Manuscript submitted 20.5.2010	Manuscript revised 12.11.2010
Date of the defence 10.12.2010	
☒ Monograph	☐ Article dissertation (summary + original articles)
Faculty Faculty of Information and Natural Sciences	
Department Department of Computer Science and Engineering	
Field of research Data Communications Software	
Opponent(s) Associate Professor Panagiotis Papadimitratos, KTH School of Electrical Engineering, Sweden	
Supervisor Professor Antti Ylä-Jääski	
Instructor D.Sc. (Tech.) Arto Karila, D.Sc. (Tech.) Hannu H. Kari, Professor Sasu Tarkoma	

Abstract

The security and reliability of the Internet are essential for many functions of a modern society. Currently, the Internet lacks efficient network level security solutions and is vulnerable to various attacks, especially to distributed denial-of-service attacks. Traditional end-to-end security solutions such as IPSec only protect the communication end-points and are not effective if the underlying network infrastructure is attacked and paralyzed.

This thesis describes and evaluates Packet Level Authentication (PLA), which is a novel method to secure the network infrastructure and provide availability with public key digital signatures. PLA allows any node in the network to verify independently the authenticity and integrity of every received packet, without previously established relationships with the sender or intermediate nodes that have handled the packet. As a result, various attacks against the network and its users can be more easily detected and mitigated, before they can cause significant damage or disturbance. PLA is compatible with the existing Internet infrastructure, and can be used with complementary end-to-end security solutions, such as IPSec and HIP. While PLA was originally designed for securing current IP networks, it is also suitable for securing future data-oriented networking approaches.

PLA has been designed to scale from lightweight wireless devices to Internet core network, which is a challenge since public key cryptography operations are very resource intensive. Nevertheless, this work shows that digital signature algorithms and their hardware implementations developed for PLA are scalable to fast core network routers. Furthermore, the additional energy consumption of cryptographic operations is significantly lower than the energy cost of wireless transmission, making PLA feasible for lightweight wireless devices. Digital signature algorithms used by PLA also offer small key and signature sizes and therefore PLA's bandwidth overhead is relatively low.

Strong security mechanisms offered by PLA can also be utilized for various other tasks. This work investigates how PLA can be utilized for controlling incoming connections, secure user authentication and billing, and for providing a strong accountability without an extensive data retention by network service providers.

Keywords	network security, future network technologies, denial-of-service attacks, Internet infrastructure, digital signature algorithms, elliptic curve cryptosystems		
ISBN (printed)	978-952-60-3464-5	ISSN (printed)	1795-2239
ISBN (pdf)	978-952-60-3465-2	ISSN (pdf)	1795-4584
Language	English	Number of pages	168
Publisher	Aalto University, Department of Computer Science and Engineering		
Print distribution	Aalto University, Department of Computer Science and Engineering		
☒ The dissertation can be read at http://lib.tkk.fi/Diss/2010/isbn9789526034652/			

Aalto University

VÄITÖSKIRJAN TIIVISTELMÄ	AALTO-YLIOPISTO TEKNILLINEN KORKEAKOULU PL 11000, 00076 AALTO http://www.aalto.fi

Tekijä Dmitrij Lagutin

Väitöskirjan nimi

Securing the Internet with digital signatures

Käsikirjoituksen päivämäärä 20.5.2010	Korjatun käsikirjoituksen päivämäärä 12.11.2010

Väitöstilaisuuden ajankohta 10.12.2010

☒ Monografia	☐ Yhdistelmäväitöskirja (yhteenveto + erillisartikkelit)

Tiedekunta Informaatio- ja luonnontieteiden tiedekunta
Laitos Tietotekniikan laitos
Tutkimusala Tietoliikenneohjelmistot
Vastaväittäjä(t) Apulaisprofessori Panagiotis Papadimitratos, KTH School of Electrical Engineering, Ruotsi
Työn valvoja Professori Antti Ylä-Jääski
Työn ohjaaja TkT Arto Karila, TkT Hannu H. Kari, Professori Sasu Tarkoma

Tiivistelmä

Internetin turvallisuus ja toimintavarmuus ovat elintärkeitä modernille yhteiskunnalle. Nykyisin Internetista puuttuu tehokkaat verkkotason tietoturvaratkaisut ja se on haavoittuvainen monenlaisille hyökkäyksille, etenkin hajauteituille palvelunestohyökkäyksille. Perinteiset päästä-päähän tietoturvaratkaisut, kuten IPSec, turvaavat vain liikenteen päätepisteet, ja eivät ole tehokkaita, jos alla oleva verkkoinfrastruktuuri halvaantuu hyökkäyksen seurauksena.

Tämä työ kuvaa ja analysoi Packet Level Authentication (PLA)-menetelmää, joka on uudenlainen ratkaisu verkkoinfrastruktuuriin suojaamiseen ja saatavuuden takaamisen julkisen avainten allekirjoitusmenetelmien avulla. PLA antaa jokaiselle verkossa olevalle solmulle mahdollisuuden itsenäisesti tarkistaa jokaisen paketin aitous ja eheys, ilman aiemmin muodostettua luottamussuhdetta paketin lähettäjään tai muihin solmuihin, jotka ovat käsitelleet pakettia. Tämän seurauksena erilaiset hyökkäykset verkkoa ja sen käyttäjiä vastaan voidaan havaita helpommin ja niitä voidaan lieventää, ennenkuin ne ehtivät aiheuttaa huomattavaa vahinkoa tai häiriötä. PLA on yhteensopiva olemassaolevan Internetin infrastruktuurin kanssa ja sitä voidaan käyttää täydentävien, päästä-päähän tietoturvaa tarjoavien tietoturvaratkaisujen, kuten IPSec ja HIP, kanssa. Vaikka PLA on alun perin suunniteltu turvaamaan nykyisiä IP-verkkoja, se soveltuu myös tulevaisuuden data-orientoituneiden verkkoteknologioiden suojaamiseen.

PLA on suunniteltu skaalautumaan kevyistä langattomista laitteista Internetin runkoverkkoihin, mikä on haaste, sillä julkisen avaimen menetelmät ovat laskennallisesti raskaita. Tästä huolimatta tämä työ osoittaa, että PLA:ta varten kehitetyt allekirjoitusmenetelmät ja niiden laitteistoläheiset toteutukset skaalautuvat nopeisiin runkoverkon reitittimiin. Lisäksi kryptograafisten operaatioiden lisäenergian tarve on merkittävästi pienempi kuin langattoman viestinnän energiantarve, minkä ansiosta PLA soveltuu myös kevyisiin langattomiin laitteisiin. PLA:n käyttämät allekirjoitusmenetelmät käyttävät myös lyhyitä avaimia ja allekirjoituksia, minkä ansiosta PLA käyttää suhteellisen vähän ylimääräistä kaistaa.

PLA:n tarjoamia vahvoja tietoturvamekanismeja voidaan käyttää myös muihin sovelluksiin. Tämä työ tutkii miten PLA:ta voidaan hyödyntää sisäänpäin tulevien yhteyksien hallitsemiseen, turvalliseen käyttäjän autentikointiin ja laskutukseen, sekä vahvan vastuunalaisuuden tarjoamiseen ilman teletunnistetietojen tallentamista operaattoreiden toimesta.

Asiasanat tietoverkkojen turvallisuus, tulevaisuuden verkkoteknologiat, palvelunestohyökkäykset, Internetin infrastruktuuri, degitaaliset allekirjoitusmenetelmät, elliptisten käyrien menetelmät

ISBN (painettu) 978-952-60-3464-5	ISSN (painettu) 1795-2239
ISBN (pdf) 978-952-60-3465-2	ISSN (pdf) 1795-4584
Kieli Englanti	Sivumäärä 168

Julkaisija Aalto-yliopisto, Tietotekniikan laitos

Painetun väitöskirjan jakelu Aalto-yliopisto, Tietotekniikan laitos

☒ Luettavissa verkossa osoitteessa http://lib.tkk.fi/Diss/2010/isbn9789526034652/

Table of Contents

List of Figures

List of Tables

List of the most important Abbreviations

AAA	—	Authentication, Authorization, and Accounting
AIP	—	Accountable Internet Protocol
ALPHA	—	An Adaptive and Lightweight Protocol for Hop-by-hop Authentication
ARPA	—	Advanced Research Project Agency
ASIC	—	Application Specific Integrated Circuit
CA	—	Certificate Authority
CAIDA	—	Cooperative Association for Internet Data Analysis
CGA	—	Cryptographically Generated Addresses
DHT	—	Distributed Hash Table
DNS	—	Domain Name System
DONA	—	A Data-Oriented Network Architecture
DoS	—	Denial-of-service
DDoS	—	Distributed Denial-of-service
DSA	—	Digital Signature Algorithm
EAP	—	Extensible Authentication Protocol
ECC	—	Elliptic Curve Cryptography
EU	—	European Union
FPGA	—	Field Programmable Gate Array
HIP	—	Host Identity Protocol
IP	—	Internet Protocol
IPSec	—	IP Security Protocol
ISP	—	Internet Service Provider
ITRS	—	International Technology Roadmap for Semiconductors
LAN	—	Local Area Network
MAC	—	Media Access Control
MIH	—	Media Independent Handover
MTU	—	Maximum Transmission Unit
NIST	—	National Institute of Standards and Technology
NTP	—	Network Time Protocol
PCB	—	Printed Circuit Board
PLA	—	Packet Level Authentication
PSIRP	—	Publish-Subscribe Internet Routing Paradigm

QoS	—	Quality of Service
RIPEMD	—	RACE Integrity Primitives Evaluation Message Digest
RSA	—	Rivest, Shamir and Adelman
SHA	—	Secure Hash Algorithm
TCP	—	Transmission Control Protocol
TESLA	—	Timed Efficient Stream Loss-tolerant Authentication
TTP	—	Trusted Third Party
URL	—	Uniform Resource Locator
VoIP	—	Voice over IP
VPN	—	Virtual Private Network
WLAN	—	Wireless LAN

Acknowledgements

The work on this thesis was supported by projects “Publish-Subscribe Internet Routing Paradigm (PSIRP)” and “Publish Subscribe Internet Technology (PURSUIT)” funded by the European Union, and by “Securing IP-based network infrastructures using Packet Level Authentication (PLA) - technique” funded by Tekes. The work was carried out in Helsinki Institute for Information Technology (HIIT), and in the Laboratory for Theoretical Computer Science of Helsinki University of Technology.

This thesis would not be possible without help from several people. I want to thank my supervisor Professor Antti Ylä-Jääski for supervising the thesis, and my instructor and superior at HIIT Arto Karila for his helpful comments and support.

I would also like to thank my instructor Hannu H. Kari for his suggestions, support and comments during my work in the PLA project and while writing this thesis. I also want to thank Professor Sasu Tarkoma from the University of Helsinki for his helpful comments.

Finally, I would like to thank the whole staff at HIIT and the Laboratory for Theoretical Computer Science for providing a great and supporting working environment.

Dmitrij Lagutin
Espoo, 12.11.2010

1. Introduction

The Internet is an integral part of everyday life and therefore its security is crucial for society. If a serious attack would disable a large part of the Internet, communications, monetary transactions, trade, and other important functions would be severely disrupted. Currently, users and the Internet infrastructure, such as routers, servers, and important services, are vulnerable to different kinds of malicious behaviour such as denial-of-service attacks, break-ins, phishing attacks, and unsolicited e-mail (Spam). Distributed denial-of-service attacks are especially dangerous and difficult to protect against.

Due to such frequent attacks, it is becoming increasingly difficult for benevolent users to use the Internet effectively, basically the signal to noise ratio is becoming too low. The reason for the Internet's vulnerability to various attacks lies within its original design goals, which assumed that the network would be used in a very different way than it is being used today. Originally, the Internet was designed to be used by a relatively small number of benevolent parties, thus the possibility of attacks originating from within the network was mostly ignored in its design. Nowadays, the situation is very different: the Internet is used by a large number of different users and virtually all attacks against it originate from within the network. Protecting the Internet against attacks is very difficult, since effective security measures against such attacks mostly do not exist.

In traditional end-to-end security solutions, only the end hosts can verify the validity and integrity of the traffic, which leads to several problems. First, they are not effective if the network infrastructure itself is under attack and unable to deliver packets. Second, they are not enough to provide sufficient security by themselves. The large amount of traffic on the current Internet is just unsolicited e-mail or other garbage, and the culprits behind attacks are rarely caught. Therefore, we feel that there is a clear need for security solutions, where the security policies are applied at every hop as the packet travels through the network. If the network infrastructure can verify the validity of the traffic, countermeasures against various attacks could be taken within the network, and not only by the end hosts. This would allow attacks to be stopped quickly and more efficiently, and would increase the chance of catching perpetrators.

After the network infrastructure has been secured, end-to-end solutions will be able to secure end-users and their services. The high level of security would greatly improve the efficiency of the network since only benevolent traffic would be forwarded, and make the network much more usable to end users who would not receive garbage traffic any more.

1.1 Problem statement

Before a problem statement can be presented, some key concepts related to network security and cryptographic systems need to be defined. From the network's and user's point of view, malicious and unwanted traffic can be classified into two categories. The first category includes the traffic that violates commonly accepted rules of the network (acceptable use policy). For example, traffic that has been forged in any way, or traffic that aims to overload or otherwise damage the recipient or the network falls in this category. The second category contains traffic that has been subjectively classified as being irrelevant by the recipient. Unsolicited e-mail (Spam) is one example of such traffic.

There exist several types of cryptographic systems. In symmetric cryptography, both sending and receiving parties share the same secret key. While symmetric cryptography is computationally efficient, it requires that the shared secret key is distributed to all recipients in a secure way. Public-key cryptography [34], also known as asymmetric cryptography, uses two keys: a public key that can be freely shared and a private key, which is mathematically tied to the public key. In public-key signature schemes [92], the creator of the content uses the private key to sign the content. Afterwards the public key and the signature are distributed along with the content, allowing the recipient to verify the content's authenticity.

The main objective of this thesis is to develop a mechanism for improving the security of the Internet by providing availability on the network layer. The mechanism aims to protect the network infrastructure and its users from various attacks, such as denial-of-service attacks, and is based on per-packet cryptographic signatures.

To achieve this objective, this thesis presents and evaluates a Packet Level Authentication (PLA) protocol, a novel method for improving the security of the

Internet by providing availability and protecting the network infrastructure and its users from various attacks. The solution presented in this thesis guarantees that benevolent users can use the network without disturbance, while the malicious and unwanted traffic in the network is detected and blocked as quickly as possible. The described solution works by giving every node in the network[1] the ability to verify each packet independently without existing trust associations utilizing public key cryptography and digital signature mechanisms. As a result, various attacks against the network and its users can be more easily detected, before they can cause significant damage or disturbance. Furthermore, a higher level trust management system is used to remove attackers from the network, and prevent them from causing further disturbance. Thus, the network will be able to fulfil its basic goal better: to deliver packets of valid users in a reliable and timely manner in all situations.

When high availability is achieved on the network layer, end-to-end security solutions can effectively protect network users against other threats by providing integrity and confidentiality at higher layers. Furthermore, a strong network layer security solution can be utilized by the higher layers to provide flexible user authentication and other functions.

This work assumes that public key cryptography can be used on the Internet scale, due to new cryptographic algorithms and advances in semiconductor technology. It is important to note that the aim of this work is to investigate how modern cryptographic solutions can be applied for securing the network infrastructure and services, and the advancement of cryptography is out of the scope of this thesis.

The original Internet was built on several assumptions, such as the scarcity of network resources, which are not valid any more. According to the Global Internet Geography report [107], the growth of average Internet traffic in 2008 was 53%, the available bandwidth grew by 62%, and the average network utilization was only 29%. Therefore, the Internet contains plenty of unused bandwidth. In addition, public key cryptography is becoming relatively less computationally intensive because of new, more efficient cryptographic algorithms, rapid advances in the semiconductor technology, and increases in the processing power. Therefore, this thesis assumes that it is feasible for a

1 In this thesis the network includes the network infrastructure and all other nodes located in the network.

network security solution to consume some amounts of bandwidth and computational resources.

Using modern public key cryptography algorithms, it is possible to build a network layer solution where every transmitted packet in the network carries a digital signature and has an undeniable owner. Such a system protects the network from various attacks and provides strong accountability that makes catching attackers much easier. This thesis aims to prove that the public key-based security solution is feasible and scalable to small portable mobile devices and high-speed Internet core network links with speeds of 40 Gbps[1] and above.

1.2 Scope of this thesis

The network security problem can be divided into three distinct levels [62] presented below. Each of these levels tries to answer different questions and solve different problems.

Technical-level security: How are threats and attacks noticed in the network, how can they be blocked or prevented on the network level? For example, a firewall is a technical-level security solution.

Policy-level security: How should the network be configured to decrease risks of attacks? Which actions should be taken in case of an attack? For example, the configuration of the firewall, implementing a security policy, falls into this category.

Juridical-level security: Which legal measures can be taken against attackers, e.g., after the firewall has detected an attack?

This work concentrates mostly on technical-level security solutions: how to detect and stop security threats on the network level. Some policy-level issues are also covered in the thesis, while the juridical level is mostly out of the scope of this research.

1 40 Gbps speed is the fastest commonly used network interface speed today. Faster routers usually use several 40 Gbps network interfaces in parallel.

1.3 Own contributions

The author's personal contributions in this thesis include:

- The PLA architecture is presented in Chapter 3. The architecture includes advanced means for stopping the unwanted traffic and reporting misuse in the network.
- Deployability analysis of PLA, including a consideration of the migration path towards PLA, as described in Chapter 9, and a performance and energy efficiency analysis presented in Section 5.3.
- Applicability of PLA to control incoming connections. Using PLA, it is possible to build a flexible system where connections are denied by default unless explicitly authorized as described in Section 6.1.
- Analysis of using PLA for implementing strong accountability, and secure billing and roaming on the Internet scale, additionally a trade-off between privacy and accountability is discussed. The study in Section 7.1 shows that a correctly implemented PLA-based network architecture provides better privacy with stronger accountability than the current Internet. Furthermore, Section 7.2 describes how PLA can be used to secure media independent handover (MIH) protocol.
- Applicability of PLA for securing the future data oriented publish/subscribe networks is discussed in Chapter 8. Since PLA is based on cryptographic identities and signatures, it is a natural option to secure also future network architectures.
- Comparison of PLA with hash chain and hash tree based security solutions. Section 5.4 includes an analysis of the security properties and power consumption of these solutions in a wireless environment. The results show that if an efficient cryptographic accelerator is available, PLA actually uses less energy in a wireless environment due to its lower bandwidth overhead.
- A proof of concept open-source implementation of PLA for Linux and FreeBSD (*libpla*), as described in Chapter 4.

Many of these results have been published in reviewed international conferences. In [69], the author's contributions include a mechanism how the rights for controlling incoming connections can be granted and revoked using PLA, and a comparison of such a mechanism against existing similar solutions..

In [70], the author described how the PLA-based network architecture can be used for providing strong accountability on the Internet, and compared the PLA-based system to the current IP address-based data retention mechanism in terms of accountability, security, and privacy. The author also investigated other applications of the PLA-based network architecture, including user authentication and roaming.

In [68], the author's contributions include a method for securing the rendezvous process in the publish/subscribe networking using PLA and cryptographic certificates, and an analysis of the security properties of such method.

In [71], the author's contributions include a comparison between PLA and lightweight hash tree and hash-chain-based security solutions in terms of security properties. Furthermore, the author analytically analyzed the security-related energy overhead of these methods in a wireless environment.

In [94], the author designed and analyzed a method for securing the media independent handover (MIH) process with PLA, in a co-operation with the first author.

1.4 Structure of the thesis

The thesis is organized as follows. Chapter 2 contains related work and background discussion, including the original design goals of the Internet, current security threats, and solutions. It also describes a set of new design goals for the next generation Internet. Chapter 3 introduces Packet Level Authentication (PLA), presents its design goals, and describes its architecture. Cryptographic solutions used by PLA and software and hardware implementation of PLA are described in Chapter 4. Chapter 5 analyses how PLA satisfies its design goals, analyses the performance and scalability of PLA, and it also includes a comparison with hash tree and hash chain based solutions. Network-level applications of PLA are discussed in Chapter 6, while Chapter 7 describes how PLA may be used at higher layers. Suitability of PLA for the future data-oriented publish/subscribe networks is studied in Chapter 8. Chapter 9 analyses PLA's deployment, Chapter 10 contains discussion about PLA and also discusses future work. Finally, conclusions are presented in Chapter 11.

2. Background

This chapter covers the original design goals, and their underlying assumptions, of the Internet and discusses its current security threats. Also previous work for solving these threats is presented. The aim is to highlight situations where current security solutions are inadequate and present requirements for the secure next-generation Internet.

2.1 Original design goals of the Internet

The Internet was originally designed for military use during the cold war era. In 1967, the Advanced Research Project Agency (ARPA), part of the US Department of Defence, proposed plans for a packet-switched network called ARPANET [119]. The proposal was approved, ARPANET started in 1969, and eventually ARPANET evolved into the Internet.

The original design goal [28] of the network, which later became the Internet was to connect different existing networks together. Other secondary design goals are listed below in order of importance:

1. The communication must continue despite the loss of networks or gateways
2. The network must support multiple types of communications services
3. The network architecture must accommodate a variety of networks
4. The network architecture must permit distributed management of its resources
5. The network architecture must be cost effective
6. The network architecture must permit host attachment with a low level of effort
7. The resources used in the network architecture must be accountable

Furthermore, one of the most important design principles of the Internet was the end-to-end principle [95][13]. The principle states that end-hosts should be responsible for end-to-end communication, while the lower network layers should be as simple and generic as possible, and that the more advanced application-dependent functionality should reside at higher layers.

Since the network was originally designed with military use in mind, the communication survivability was the top goal while accountability was at the bottom of the list. Had it

been developed for commercial purposes, accountability would have had a much larger importance.

It is important to note that protection from internal threats or attacks, which are caused by nodes attached to the network, was never mentioned in the list of design goals. Since the Internet was designed originally for military use, it was assumed that a possible attacker would always attack the network from the outside and would only try to destroy or sabotage the network infrastructure using a physical attack. All nodes that were connected to the network were assumed to be benevolent and always working for the common good of the network, never trying to intentionally cause any damage to the network. The design goals explicitly mention that the Internet must continue to operate despite the loss of infrastructure, like gateways. However, they completely ignored the situation where the gateway, or some other node in the network, is controlled by the attacker and is used to attack the network from the inside, for example, by broadcasting incorrect routing information or other means.

The original design goals also contain other assumptions. There was not any need to handle privacy and accountability since the Internet was to be used only by a small benevolent group. Eavesdropping was not an issue either because it was assumed that all communication would be transmitted using fixed lines that would be inaccessible to the external attacker.

It can be seen that many problems that affect the current Internet come from these original design goals and assumptions. Currently, the Internet is no longer solely used by military or government agencies, and virtually all attacks against the Internet originate from within the network itself, from nodes or routers that are attached to the Internet. Efficient mechanisms to defend against these attacks do not really exist, since they were not considered to be relevant in the original design goals.

Additionally, eavesdropping is a significant problem today, especially as wireless networks become more popular. Privacy issues on the Internet are also very important as the Internet's popularity is growing and the amount of personal data stored on the Internet is also increasing rapidly. Finally, the lack of accountability in the Internet prevents authorities from catching culprits behind attacks.

2.2 Current threats on the Internet

There exist several types of attacks that can be launched against Internet's users, servers, and infrastructure. A detailed description of these attacks, concentrating on attacks that target the Internet's infrastructure, follows below.

In a denial-of-service (DoS) attack, the attacker disturbs the victim in a such way that the victim is unable to continue normal operation. This can be accomplished using flooding: the attacker sends a large number of meaningless packets to the victim and these packets will use all or a majority of the victim's bandwidth or computing power, making it difficult or impossible for the victim to communicate with legitimate nodes. A DoS attack can also be accomplished by flooding the victim with legitimate requests. For example, the attacker could make thousands of requests for a web-page that is located on the victim's server.

By itself, a denial-of-service attack is usually not very dangerous. However, in a distributed denial-of-service attack (DDoS) [100] a large quantity of nodes simultaneously attack a single victim. Since a DDoS attack may originate from thousands of nodes that are located in different parts of the Internet, defending against such attacks is currently very difficult.

Denial-of-service attacks are usually launched against WWW servers, but there have been cases of DDoS attacks against DNS root servers. For example, statistics of the Finnish national Computer Emergency Response Team (CERT-FI) [23] show that the number of DoS attacks has increased fivefold between years 2006 and 2008.

A good example of a serious DDoS attack is an attack launched in Estonia in May 2007 [82]. This attack was launched against several government web sites, including the sites of different ministries. Overall, the attack consisted of 128 unique DDoS attacks that lasted from less than a minute to over ten hours. This attack highlighted two major problems that exist on the current Internet: the inability to react quickly to an attack and the inability to catch the perpetrators. The authorities were powerless to stop the attack which lasted more than one week overall. Years after the attack the culprits have yet to be caught.

In spoofing attacks, forged packets are used to attack the network. There exist different kinds of spoofing attacks. The Address Resolution Protocol (ARP) is used on the network layer to resolve a node's hardware address based on the network layer address like an IP address. In the ARP spoofing attack [8], the attacker sends fake ARP reply messages containing incorrect hardware address/network address mappings. The attacker can use ARP spoofing, e.g., to map his own hardware address to the gateway's network address, and thus a victim will send his packets to the attacker instead of the proper gateway. This would allow the attacker to launch a man-in-the-middle attack or simply cut all communications to the victim altogether.

TCP (Transmission Control Protocol) is the most commonly used transport protocol on the Internet. There are several TCP-related attacks such as TCP reset attacks and TCP SYN flooding attacks. In the TCP reset attack [114], the attacker attempts to terminate established TCP connections by sending spoofed TCP packets with an RST bit set. A TCP SYN flooding attack [98] is a form of denial-of-service attack where the attacker creates a large number of partially opened TCP connections to the victim, draining the victim of computational resources such as memory. The TCP protocol uses a three-way handshake to establish the connection: the initiator first sends a SYN message, the recipient replies using a SYN-ACK message, and finally the initiator sends an ACK message. The recipient also allocates resources to handle the connection after receiving the initial SYN message. These resources are freed after the recipient receives an ACK message and the connection is established. In a SYN flooding attack, the attacker sends a large amount of SYN messages using spoofed source addresses. This causes the victim to allocate resources to handle these connection requests, and because the source addresses are spoofed, the victim will never receive ACK messages and subsequently runs out of resources.

In a replay attack [104], the attacker captures valid communication between victims and replays it at a later stage. In the simplest form of a replay attack, the attacker simply creates several copies of a valid packet and sends the copies to a valid destination, consuming extra resources in the network. Such an attack can cause significant damage in wireless networks where resources like bandwidth and battery power are often scarce. Usually, the aim of a replay attack is to gain unauthorized access by replaying valid packets. In a classic replay attack, the attacker intercepts a message exchange and replays it fully at a later stage. For example, the attacker could intercept packets

containing a password exchange between a client and a server and then gain access to the server by resending intercepted packets. In more complex replay attacks, the attacker replays some part of a message exchange (which can be an exchange of cryptographic keys) simultaneously with an actual message exchange to gain unauthorized access or impersonate a victim. Usage of timestamps or one-time session tokens during message exchange offers protection against replay attacks. Such measures aim to guarantee that valid messages can be sent only once.

The Domain Name System (DNS) is used, among other things, to determine the IP address of a node based on its hostname. In a typical case, the client sends a DNS request containing an unknown hostname to the DNS server, and the server replies with the corresponding IP address using a DNS reply message. In the DNS cache poisoning attack [9], an attacker feeds incorrect hostname/IP address mappings to the DNS server. For example, the attacker could map a legitimate site like a website of a bank to his own IP address. As a result, clients wishing to access the bank's website will access the attacker's own server instead.

There are also other ways to attack the DNS. The DNS system is vulnerable to spoofing attacks because currently DNS messages are not authenticated or protected by any means. Thus, the recipient of a DNS reply message can not guarantee that the received message is authentic. There exist a DNS Security Extensions (DNSSEC) [7] mechanism to protect DNS messages using cryptographic signatures and other means. However, DNSSEC is not yet widely used on the Internet.

Phishing attacks differ from other attacks mentioned previously because they target users directly instead of targeting the network infrastructure or protocols. The aim of a phishing attack is to lure users to voluntarily disclose confidential information like passwords. For example, the attacker can contact a victim by phone, pretend to be a system administrator from the victim's place of work, and ask for a victim's password. On the Internet, one commonly used phishing approach is to create a look-a-like web page of an on-line bank and send its URL by e-mail to a large number of people in order to lure victims to reveal their bank account usernames and passwords. The amount of phishing cases has significantly increased in the last few years and in September 2009 alone over 40,000 phishing cases were reported [6]. Phishing attacks are often quite

effective since it is currently difficult for an ordinary user to guarantee the authenticity of a web page or the real sender of an e-mail message.

2.3 Current security mechanisms in the Internet

There are three main principles of information security which various security solutions aim to provide: confidentiality, integrity, and availability. The aim of confidentiality is to guarantee that only authorized parties can access data; confidentiality is usually accomplished via encryption. Integrity ensures that data is authentic and has not been tampered with. Availability aims to guarantee that data and services are available for legitimate users in all possible situations.

The Internet architecture can be divided into several layers according to the TCP/IP model; the application layer, transport layer, internet layer, and the network access layer as shown in Figure 1, the figure also contains a comparison to the Open Systems Interconnection (OSI) model [54].

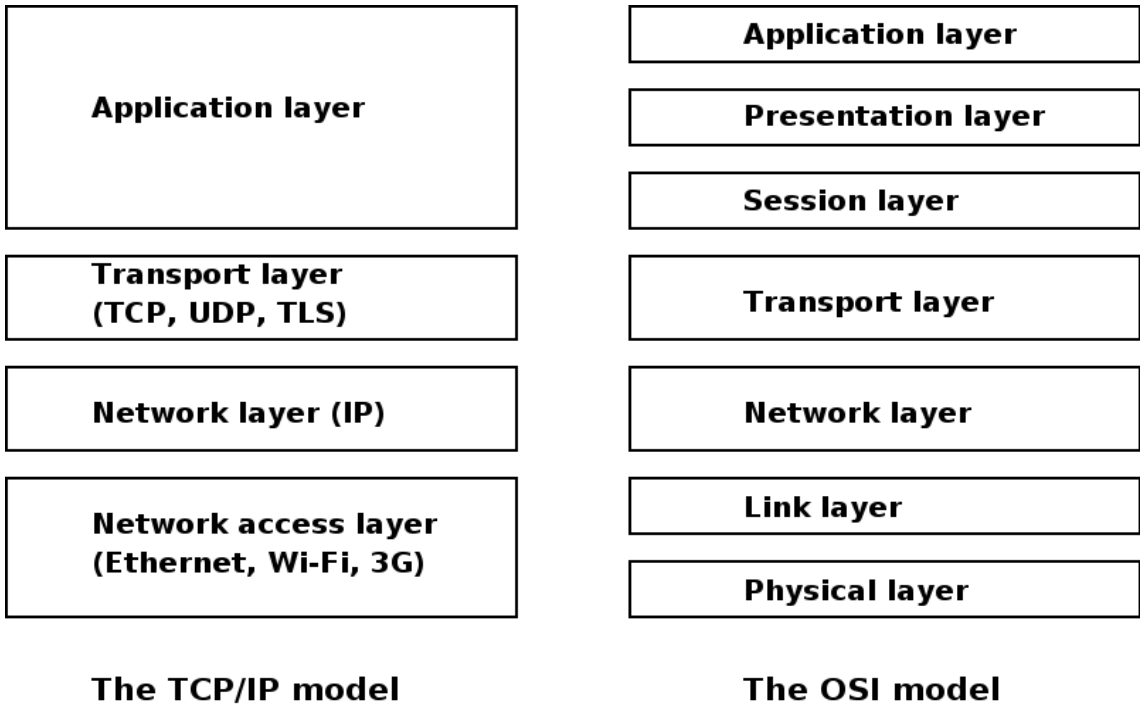


Figure 1. The Internet architecture according to the TCP/IP model and comparison to the OSI model

The network access layer of the TCP/IP model contains the physical and link layers from the OSI model while the application layer of the TCP/IP model includes application, presentation, and session layers from the OSI model.

There exist several solutions that work on the application layer of the TCP/IP model, for example, an application-level gateway (ALG) [101] that augments firewalls and network address translation (NAT) mechanisms. However, such application-level solutions are not covered here in detail, since this thesis concentrates on the network-layer security. Security solutions can also be classified based on their applicability as shown in Figure 2.

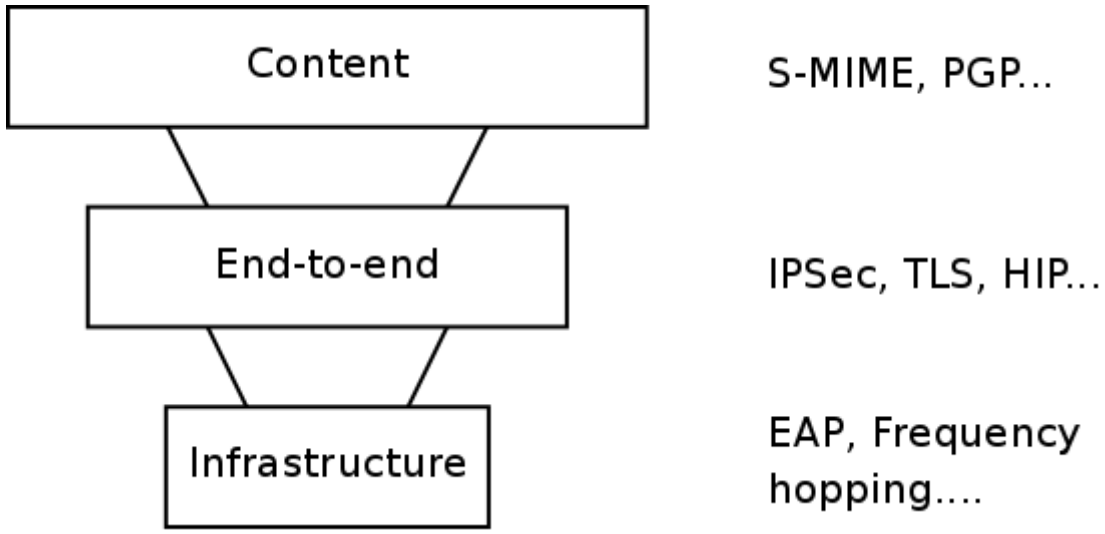


Figure 2. Security solutions divided into content-level, end-to-end level and infrastructure-level solutions

In this figure, content-level solutions refer to security solutions which work on the application layer in the TCP/IP model and aim to secure specific content, such as e-mail messages. End-to-end solutions usually work on the application and network layers and aim to protect all content which is transmitted between two end points. Infrastructure-level solutions aim to secure the underlying infrastructure on the link and physical layer against threats such as jamming or eavesdropping.

Existing security solutions for lower layers of Internet and countermeasures against DoS attacks are explained in more detail below.

2.3.1 Transport-layer mechanisms

The Transport Layer Security (TLS) [33] protocol and its predecessor, Secure Socket Layer (SSL), are end-to-end security solutions which aim to provide secure communication on the transport layer. They provide authentication and confidentiality using encryption and are widely used to secure web browsing and instant messaging

communications. In the case of web browsing, usually only the server is authenticated. During the establishment of such TLS connections, the server provides the client with a certificate that is signed by some trusted certificate authority (CA); the client may optionally contact the certificate authority to verify the certificate's validity. The client and the server also agree on a cryptographic cipher and hash function, and generate key material for encryption and decryption during the establishment phase. Since TLS works on the presentation layer of the OSI model, it does not protect traffic from attacks that occur at lower layers.

2.3.2 Network-layer mechanisms

The IP Security Architecture (IPSec) [63] protocol aims to improve the security of IPv4 and IPv6 protocols by providing integrity and confidentiality on the network layer. IPSec extends the IP header and uses the Authentication Header (AH) and Encapsulating Security Payload (ESP) protocols to accomplish its goals. The AH protects the packet's integrity while the ESP provides confidentiality through encryption. IPSec provides end-to-end security, allowing two hosts to establish a secure, encrypted IPSec connection between them. IPSec is a widely used protocol to secure virtual private networks (VPN) [45].

One major problem with IPSec and other traditional security solutions is that they concentrate on providing end-to-end security but they cannot protect the underlying network infrastructure. If the packet protected by IPSec or a similar solution has been modified, duplicated, or delayed, only the end point of the security association can detect this, and intermediate nodes will continue to forward these invalid packets. As a result, such packets will unnecessarily consume network resources. In addition, such end-to-end security solutions are useless if the underlying network infrastructure is attacked and is unable to deliver packets to the destination as a result of a DoS or other kind of attack.

The Host Identity Protocol (HIP) [80] aims to provide confidentiality, better support for mobility, and support for multihoming. Traditionally, an IP address determines both the topological location of a host in the network and its identity. This can cause problems in case of mobility where the host receives a new IP address upon changing networks. To solve this problem, HIP introduces host identifiers (HIs) to describe the identity of the

host, therefore the IP address is only used for describing the topological location of the host. Under HIP, connections are initiated to host identifiers instead of IP addresses and the host identifier is actually a hash over the host's public key, HIP also supports interoperability between IPv4 and IPv6. HIP provides confidentiality by encrypting all data traffic using the host's private key and the IPSec's ESP protocol.

HIP uses puzzles during its 4-way base exchange, which authenticates hosts before establishing a connection. The idea behind the puzzle is to require much more computational power from the initiator than from the recipient, this protects the recipient from some DoS attacks. However, such a puzzle mechanism can be abused in certain situations to launch DoS attacks against peers that are communicating with the victim using HIP. Imagine a situation where several devices with very low computational power are communicating with a certain HIP server. If an attacker starts flooding the HIP server with connection requests, the server will increase the difficulty of its base exchange puzzle, and those devices with very limited resources will be unable to solve the puzzle quickly enough to continue normal operation. HIP also possesses other security related disadvantages. While HIP offers some protection for the recipient during the establishment phase, it does not protect a recipient or underlying network after the connection has been established. After an attacker has discovered a victim's IP address, the attacker can freely launch, for example, a DoS attack against the victim. The attacker can also disclose IPSec's security association to other parties. The multihoming feature of HIP also improves the user's security, since a victim can change to another network interface and therefore stop the attack. However, this approach does not protect the network infrastructure since the malicious traffic sent to the victim will still reach the victim's previous network.

The experimental HICCUPS draft [19] defines a HIP DATA packet type, which allows secure communication by using cryptographic signatures without establishing base exchange. However, in this case hosts are not mutually authenticated.

Accountable Internet Protocol (AIP) [5] aims to improve the Internet's security by providing accountability on the network layer. AIP uses globally self-certifying unique end-point identifiers (EID) to identify and address the source and the destination of the connection, while still being based on IP. EIDs contain hashes of hosts' public keys which are communicating within the network. The network under AIP is divided into

several accountability domains (ADs), each of which has its own unique identifier which is also a hash of a corresponding public key. ADs can form a hierarchy within the network and the AIP address takes the form of AD_1:AD_2:...:AD_n:EID. By tying connection endpoints to public keys, AIP enforces accountability, since the initiator of the connection will be responsible for sent data.

AIP aims to prevent source address spoofing in the following way. If the router receives a packet from the unknown EID, the router will send a verification message back and the node will reply with a message signed by its private key. Since EID is a hash of the node's public key, this proves that the node owns a corresponding private key and thus has a right to use the EID. A similar method can also be used to authenticate ADs when a packet crosses the AD boundary. The downside of this approach is that additional control messages are required for source address spoofing protection.

Cryptographically Generated Addresses (CGA) [10] is a method to tie an IPv6 address to the user's public key. Basically, the hash of the user's public key is used to construct an interface identifier part of the IPv6 address. In order for CGA to be effective, packets must be signed by the user's private key and they must contain the corresponding public key. The main aim of CGA is to offer protection against IP address spoofing; every other node in the network can verify whether a hash of the public key matches an interface identifier.

2.3.3 Link-layer mechanisms

Link-layer security solutions aim to improve security by providing confidentiality and authentication on the link layer. They are mostly used in wireless networks, because wireless networks are very vulnerable to eavesdropping and are easy to attach to. Wired Equivalent Privacy (WEP) was the original solution to enhance the security of IEEE 802.11 wireless networks. However, WEP is a poor security solution for several reasons [20]. First, WEP uses shared secret keys for encrypting traffic and these keys are often stored in an insecure way on the device. In addition, WEP also lacks a key management protocol making it difficult to distribute new keys to all nodes. The biggest drawback of WEP is that its encryption can be easily broken by capturing enough encrypted packets. Several methods to break WEP encryption have been published and the fastest method [108] can discover the encryption key in less than one minute. To overcome these

limitations, WEP has been largely replaced by Wi-Fi Protected Access (WPA) and WPA2 technologies. WPA and WPA2 offer significantly better encryption and also support the Extensible Authentication Protocol (EAP) [1] for user authentication. EAP is a more secure and scalable way to manage authentication as compared to using pre-shared secret keys.

Frequency hopping [36] is a mechanism that is often used in military environments to protect wireless networks against jamming and to a lesser extent against eavesdropping. When frequency hopping is employed, the transmitter and receiver change their communication frequency rapidly across a wide frequency range. Since the attacker most likely has limited resources for jamming, the attacker will only be able to jam a small portion of the used frequency range. Thus, most communications will go through despite the jamming. In order for frequency hopping to be effective, the mechanism employed to select new frequencies must not be known by the attacker, therefore it can not be used for securing publicly available wireless networks.

Link-layer security solutions offer protection from external attacks, although they are not effective against internal attacks. For example, if the attacker can compromise the node that is communicating using pre-shared keys with an IEEE 802.11 base station, the attacker will gain access to relevant encryption keys and will be able to listen to traffic between other clients and this base station. A similar problem exists when frequency hopping is used; if the attacker compromises one node, the attacker is likely to be able to deduce the method for selecting new frequencies and will thus be able to jam or eavesdrop on the communication in the rest of the network.

Overall, existing security solutions have two main drawbacks. First, they do not protect the end user from receiving “garbage” traffic from the Internet. Second, they do not provide availability and do not effectively protect the network infrastructure from attacks, especially if these attacks occur within the network itself. For example, a traditional principle to protect the network against a denial-of-service attack is to have more resources available compared to the attacker. Naturally, this principle will not work against a distributed denial-of-service attack where the attacker can control thousands of nodes.

2.3.4 Defences against distributed denial-of-service attacks

Several solutions have been proposed for preventing and mitigating DDoS attacks. Stateless Internet Flow Filter (SIFF) [117] aims to prevent DDoS attacks with capabilities that are bound to the sender's IP address. During the negotiation phase routers on the path mark capabilities and later verify that data packets posses valid capabilities.

In a basic capability scheme hostile senders can flood capability request packets to the sender and simply limiting the amount of bandwidth reserved for capability requests may prevent valid users from communicating. Traffic Validation Architecture (TVA) [118] improves the basic capability approach by queuing capability requests based on path identifiers, which are approximate source locators. This reduces the possibility that a hostile sender can prevent valid users from sending capability requests.

The Fastpass [115] scheme uses cryptographic tokens together with capabilities. In this scheme each domain has its own public key, which is globally distributed through the BGP [91] protocol to each domain and router. The sender must first receive a token from the destination domain, this token is basically a certificate to some arbitrary identity, signed by the domain private key. The sender includes this token in the setup request packet that is sent to the destination. Routers on the path verify the token and mark the packet's capability header in a similar fashion as TVA or other capability scheme.

Fastpass allows tokens to be distributed separately from data traffic allowing more flexibility. It also prevents using setup messages for DoS attacks against the destination, since these setup messages must include a valid token. As a downside, the tokens are domain-specific, allowing the sender to send setup messages to any node within the domain with a valid token. The security of the scheme is not very strong either, since only the token is protected by the signature. This allows an attacker to reuse a valid capability given to another node in the same source network, and a malicious router can modify capabilities without chance of discovery.

ICING [99] is based on a policy that a consent is required from the receiver and every intermediate domain before the communication can take place. In order to enforce this

policy, the ICING packet header contains a public key and consent from each domain on the path. Instead of using cryptographic signatures, ICING uses a weaker MAC (message authentication code) to express the domains' consent in packets. Even though ICING uses public keys based on the elliptic curve cryptography, including a public key from each domain together with the consent produces a high bandwidth overhead.

The existing DoS protection mechanisms have several downsides. First, the capability to communicate is often given to the whole subnet or a domain. This allows attacks to originate from the subnet or domain with valid senders, and introduces a problem with mobility. The security is also questionable, since in most cases security mechanisms are weak. For example, in some approaches the capability field within the packet's header can be modified by a hostile router without others noticing. While Fastpass uses signed tokens, it has the same basic problem as other schemes that rely only partially on cryptographic signatures. The attacker can flood routers with connection setup packets that contain invalid signatures, and routers must verify all these signatures to check whether such packets are valid. Such an attack will overload the routers, since it is assumed that routers are not able to perform signature verifications at wire speed.

The current Internet architecture where by default anyone can contact anyone else makes it difficult to implement flexible and efficient mechanisms against DoS attacks. According to [47] several changes must be made to the Internet's architecture in order for it to be really secure against denial-of-service attacks. The study proposes the following changes: 1. Separation of client and server addresses, 2. Non-global client addresses, 3. Reverse path forwarding checking of server addresses, 4. State setup bit to distinguish different types of traffic, 5. Nonces and puzzles, 6. Middlewalls in addition to firewalls, and 7. Efficient multicast.

2.3.5 An example of a real-life security solution

This section describes a proposal for a secure Internet banking solution, which was described in [90]. The proposal includes a duplicated infrastructure and several layers of security to secure the service. In the proposal, the computing system of the bank is duplicated on three levels. First there exist two copies of the bank's computing centre located in different geographical locations. This is necessary to protect the system from physical attacks and natural disasters. Secondly, ISP connections are duplicated; there

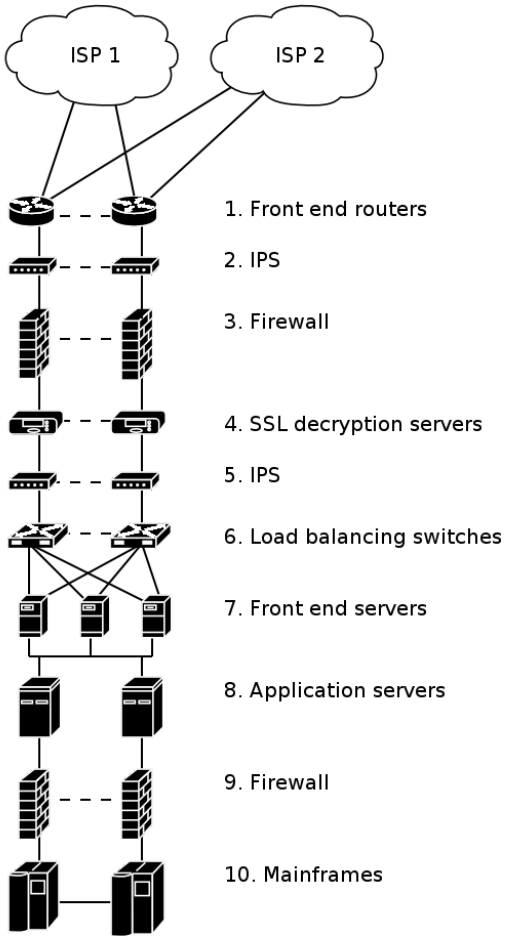


Figure 3. An example of a security solution for Internet banking [90]

are two ISPs which provide services to both computing centres. Therefore, even if one ISP goes off-line, the service is not interrupted. Thirdly, each computing centre has at least two sets of identical hardware (firewalls, routers, servers, mainframes) in order to cope with hardware failures. An overview of the proposal is shown in Figure 3 above.

At the top layer there are front end routers that are connected to both ISPs. These routers forward traffic to IPS (Intrusion Prevention System) [96] systems that reside in the front of firewalls. The aim of these systems is to stop trivial flooding attacks before they can cause damage to other components. On the fourth layer there are separate servers to handle SSL decryption. Since SSL encryption and decryption are computationally intensive operations, it is better to handle these operations in dedicated servers. Otherwise the attacker could flood mission-critical servers with a high amount of SSL connection requests. There is another set of IPS systems in front of load balancing switches. Load balancing switches forward traffic to an available front-end server which

is connected to the actual application servers. Finally, there is another set of firewalls in front of the mainframes that contain the critical data.

Such a security solution is very complex, but this complexity is necessary to ensure good protection against different kinds of attacks. Duplication of servers means that a single server failure will not interrupt operations, while having multiple security layers adds protection for servers that are actually handling critical transactions.

2.4 Consequences of the insecure Internet

Due to insecurity of the current Internet, users are susceptible to various attacks that are very easy to launch. This problem exists on many levels; not only are attacks easy to launch, but stopping or limiting the attack quickly is difficult from the victim's point of view. Finally, the culprits behind those attacks are rarely caught, allowing them to continue attacks in the future without fear of retribution.

There are also additional problems. Insecurity of the Internet has led to overuse of firewalls to curtain attacks. Firewall rules are often so strict that they block valid traffic making the Internet less usable for benevolent users. For example, currently there are no good means to verify whether the incoming traffic is desired or malicious, therefore firewalls often block all incoming traffic to end hosts, making it difficult to initiate a remote session to the personal computer. These strict firewall rules also contradict the fundamental goal of the network: to convey data for benevolent users. For example, voice over IP (VoIP) applications like Skype must use various tricks to circumvent strict firewall rules in order to make their services usable [97].

Furthermore, a flexible user authentication is difficult to implement in the current Internet, since there is no strong network-layer security solution available. Therefore, different services often use their own authentication solution, and as a result users must have multiple different usernames and passwords that are difficult to maintain. One good example of this problem is an authentication to the publicly available wireless LAN network. Even thought some progress has been made in this area, wireless LAN authentication is often still a time-consuming process that requires the user to navigate thought multiple pages before access to the network is granted.

2.5 Redesigning security of the Internet

Since the Internet suffers from serious inherent security problems, simply patching the current system is not good enough [74]. Instead, a novel network-layer security solution should be created.

This section describes the requirements for the secure next-generation Internet. The fundamental goal is availability: to deliver packets of valid users in a reliable and timely manner in all situations. Other requirements for the secure next-generation Internet are listed below. The network should support these requirements, although the applied security policies will determine how strictly these requirements will be enforced in various situations.

Only valid packets are forwarded in the network and the malicious traffic is stopped as soon as possible

Since the Internet is currently very vulnerable to internal attacks, it is very important to detect and react to those attacks as soon as possible. Thus, only valid, verified packets should be allowed to be transmitted in the network. Invalid packets that have been modified, delayed, or duplicated are considered malicious since they can be a sign of DoS, replay or other attacks. Therefore, these packets should be discarded by the first possible entity of the network before they can cause damage or unnecessarily consume network resources elsewhere.

This requirement is analogous to detecting forged currency as quickly as possible, instead of verifying its authenticity only at end-points, i.e., banks.

Every packet has an owner and all packets originate from trusted entities

Vehicles are allowed to use roads only after they pass an inspection and receive a numberplate that uniquely identifies the car and its owner. Similarly, there should be a way to determine the owner of every packet that is sent on the network.

This is an important requirement in order to limit various attacks and effectively remove entities behind those attacks from the network. In addition, every entity that sends data to the network must be authorized by a some authority that is responsible for managing

the network. It is important to provide such traceability and accountability in order to make it easier to catch attackers, and to offer protection against phishing attacks.

Malicious nodes should be removed quickly from the network

A benevolent node may become dangerous to the network for several reasons. For example, the node could be hijacked by a malicious party or its software could malfunction. Therefore, there should be an effective way to quickly remove malicious nodes from the network before they can cause significant damage.

This requirement would especially help against DDoS attacks that can utilize millions of compromised nodes. Removing these nodes from the network mitigates the attack and makes future attacks more difficult to launch with the same compromised nodes.

Prioritizing traffic

In case of emergency, the network's bandwidth may become very limited. Therefore, there should be a way to prioritize traffic in order to make sure that high priority traffic will get through in all situations. This is analogous to giving a priority to emergency calls in mobile phone networks.

Manageability

The whole network cannot be managed by a single entity. Hence, there should be a way for different operators and authorities to effectively manage different parts of the network without requiring centralized control. There should also be an easy and flexible way to add new nodes and users to the network. This requirement resembles the fourth requirement of the original Internet but covers a wider scale. The network may be very dynamic containing a large amount of nodes which are constantly leaving and entering the network.

Controlling incoming connections

In the current Internet architecture the initiator of the connection[1] is completely in control of the connection. The initiator of the connection can decide whom he will contact and when a connection is made. However, such a policy presents many problems. The recipient of the connection might be using a wireless access network with a limited bandwidth, and the recipient might even have to pay for all incoming traffic. In

1 In this context, the term connection denotes the situation where the initiator is sending data to the recipient over the network by any possible means.

addition, the recipient might be in a situation where he does not want to be disturbed by unnecessary connections, while at the same time, the recipient may want to receive very important connections from specific initiators.

This requirement would also make DoS attacks more difficult to launch, and is analogous to the “do not call” registry for telemarketers.

Privacy protection

The accountability requirement mentioned above does not mean that users should completely abandon their privacy. For example, users must have a way to create pseudonyms in such way that they can maintain their anonymity to the network in normal situations. Basically, the user should not be forced to disclose unnecessary information to other parties in the network, for example, it is not necessary to disclose a real identity to participate in an online discussion, and even the access network provider does not always need to know the real identity. However, if the user breaks the law, authorities should have a way to determine the real identity behind the user's pseudonym.

Compatibility with future data-oriented networks

Traditionally, the Internet architecture has been a host- and connection-oriented one where users establish connections to specific hosts. Such a model is awkward and unnecessarily complex for many applications since users are more interested in receiving the actual data content than in establishing a connection to some host. Furthermore, an efficient multicast [31] is practically impossible to implement on the Internet scale. Data-oriented publish/subscribe networks aim to solve these problems by giving users the means to retrieve relevant data without having the information about its topological location within the network. The future network-layer security solution should also be compatible with principles used by data-oriented networks.

None of the existing security solutions completely satisfy the above-mentioned requirements. For example, end-to-end security solutions, such as IPSec and HIP, do not allow intermediate nodes to detect malicious or unwanted traffic without pre-existing trust associations with the sender. Furthermore, while IPSec and HIP tie traffic to cryptographic identities, there is no way to determine a real identity behind it, or verify that the cryptographic identity can be trusted.

In addition to these security-related requirements, the next-generation network architecture must naturally take other issues into account, such as compatibility with different kinds of access mediums, these other requirements are not covered in this thesis.

3. Packet Level Authentication (PLA)

In order to solve the aforementioned security problems, we introduce the Packet Level Authentication (PLA) protocol [21]. The main aim of PLA is to enhance network security by providing availability and protecting the network from several kinds of attacks, like denial-of-service attacks. The main principle is that benevolent traffic should go through while malicious traffic should be detected and stopped as quickly as possible. The major difference between traditional security solutions and PLA is that PLA gives the ability for nodes in the network to detect attacks immediately by checking the authenticity and integrity of every packet. In comparison, when traditional end-to-end security solutions like IPSec are used, only the end point of the security association can verify the authenticity of the packet. Unlike traditional link-level solutions, PLA allows every node to verify the packet independently without having to trust nodes that have previously handled the packet. It is important to note that PLA aims to complement existing security solutions instead of completely replacing them.

The security measures in Packet Level Authentication resemble those present in paper currency. Anyone can verify whether or not a paper bill is authentic without having to contact the bank that issued the bill. It is enough to verify various security measures inside the bill, such as its watermark, a metal strip, or a hologram. The same principle applies to PLA. When PLA is used, any node in the network can verify the authenticity and the integrity of every packet without having any kind of contact with the sender of the packet because PLA includes in every packet all the necessary data to carry out such verification. Such a system has a significant advantage compared with traditional security solutions that concentrate on providing end-to-end security. Because PLA allows various attacks to be immediately detected, the network can take countermeasures against them in a more effective way, before attacks can cause a significant amount of damage. To accomplish its goals, PLA utilizes digital signatures based on public key cryptography. The public key cryptography is very computationally intensive, but it can be used with a sufficient performance as long as dedicated hardware is used to handle cryptographic operations.

Integrity protection allows nodes in the network to easily detect modified traffic, however, detecting the situation where a benevolent node has turned hostile is a more difficult problem and therefore PLA contains two distinct layers:

- The integrity protection through digital signature techniques and cryptographic identities.
- The trust management layer for granting, delegating, and revoking rights to above-mentioned cryptographic identities.

This chapter is organized as follows: Section 3.1 discusses design criteria behind PLA and an overview of the PLA architecture is presented in Section 3.2. Section 3.3 introduces the PLA header structure while Section 3.4 describes a lightweight variant of the PLA header that can be used to reduce PLA's bandwidth and computational overhead. Section 3.5 describes trusted third parties in the context of PLA. Section 3.6 describes how the authenticity of the packet is verified using the information contained in the PLA header, and Section 3.7 shows how the network may react to attacks. Finally, the bootstrapping of a new node is discussed in Section 3.8.

3.1 PLA design criteria

The design criteria of PLA are listed below. They are classified as either mandatory criteria, that are essential, important criteria that are important to have, but not essential, or optional criteria that are less important but are still useful.

Mandatory

Compatibility with the existing Internet. The system shall work with existing IP networks without requiring any major changes to the network. The system shall also be compatible with existing security solutions like IPSec. This requirement is essential to allow a gradual deployment of the system to the current IP networks.

Deployability. The system must be easy to deploy on a wide scale. It shall be possible to easily add more nodes to the system and the system must also work without any additional security association setup between nodes.

Malicious nodes should be removed quickly from the network. A benevolent node can become hostile for the network for several reasons. For example, the node could be hijacked by a malicious party, or the node's software could malfunction. Therefore, to mitigate severity of attacks there shall be a way to remove malicious nodes from the

network effectively and quickly before they can cause significant damage to the network. This requirement was also present in Section 2.5.

Validation of packets. Every node in the network must be able to validate the authenticity of every packet without prior communication with the sender of the packet. This is especially important in wireless ad-hoc networks where packets take different routes.

It should be possible for any node to detect if a packet has been modified, duplicated, or delayed. The last two validation requirements are important to protect the network from replay attacks which use duplicated or delayed packets.

Important

Scalability. The system should be scalable from small wireless ad-hoc networks to large networks on the Internet scale. The system should also be usable with small and portable devices. This requirement is not mandatory, since the system would still be useful in certain situations, such as in mission critical networks, even if this requirement is not satisfied.

Optional

Small power consumption and bandwidth overhead. It is preferable for the system not to introduce significant bandwidth overhead and not consume large amounts of power. This requirement is especially important for mobile networks and small mobile devices, because such networks and devices are usually bandwidth and energy constrained. The requirement is marked as an optional, since it is not compulsory for the functionality of the system. Even without fulfilling this requirement, the system would still be useful in certain situations, e.g., wired networks with plenty of available bandwidth and no shortage of power.

Free of patents. The system should not use patented technologies. Fulfilling this requirement would ease the widespread adoption of the system.

It is important to note that the PLA is not designed to provide confidentiality in end-to-end communication. Other security solutions such as IPSec and HIP can be used together with PLA to provide confidentiality.

3.2 Overview of the PLA architecture

The basic idea behind PLA is to ensure the authenticity of packets by using public key digital signatures to sign every packet sent over the network. When public key cryptography is used, only the holder of the private key can sign the packet, but every party can verify the authenticity of the packet using the packet's signature and the sender's public key. PLA accomplishes its goals by adding its own header to the every IP packet, which contains all necessary information for verification of the packet. An overview of an example PLA architecture is presented in Figure 4.

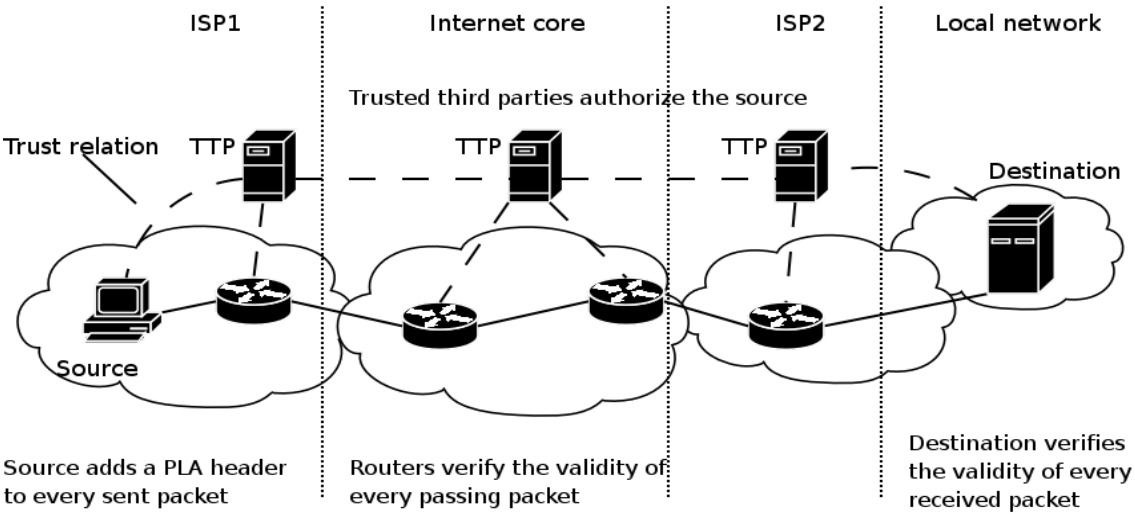


Figure 4. An example of usage configuration of the PLA-based network architecture

In a simplified case, the PLA architecture consists of four major elements: a source and destination that are communicating with each other, routers between them and of trusted third parties (TTPs). In comparison to the plain Internet architecture, PLA adds trusted third parties and the ability to verify the PLA header information at every node. The example figure is vertically divided into four networks; the network of the source, the Internet core network, the network of the destination's operator, and the destination's local network. Solid lines denote the actual data connection while dashed lines denote trust relations between different entities.

The basic operation of PLA is as follows. The source that sends a packet adds a PLA header to each sent packet. This header is added just after the IP header using a standard IP header extension mechanism [32]. As the packet travels through the network, all routers that handle the packet verify the packet's validity using information from the PLA header[1]. If this validity check fails, the packet is discarded immediately. Finally, the packet arrives at the destination, which will also perform the validity check.

In addition, there exist entities called trusted third parties (TTPs) which will authorize the sender. In this example, it is assumed that such authorization between the TTP and sender has already been carried out. In the scope of PLA, a TTP is responsible for the tasks of a certificate authority (CA) and registration authority (RA) from a traditional public key infrastructure architecture [24]. While TTPs are not necessary for checking the validity of the packet, they add another layer of security. As the packet travels through the network, intermediate routers and the destination can contact trusted third parties to verify that the sender of the packet is a valid and trusted entity in the network.

The position of PLA in the TCP/IP model is shown in Figure 5 below.

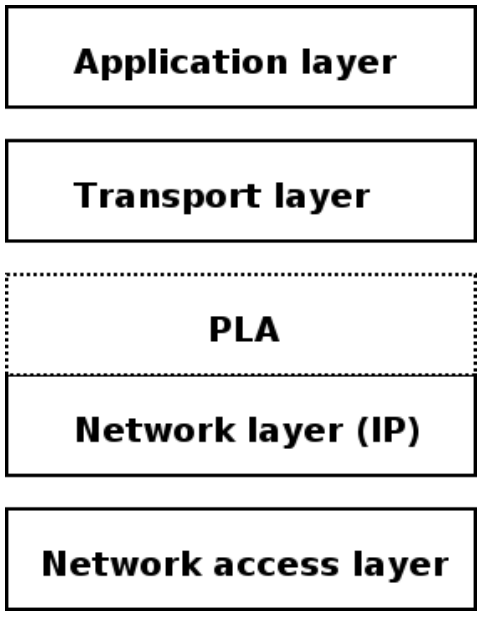


Figure 5. Position of PLA in the TPC/IP model

PLA is completely transparent to higher layers, thus PLA is able to work along with any current or future higher layer protocols like TCP, UDP, and HIP. In this figure PLA is

1 In case the router does not understand a PLA header, it routes packets simply based on IP header information. This enables a gradual deployment of PLA.

positioned on top of a network layer as the PLA header resides after the IP header. However, PLA is not dependent on the network-layer protocol used, thus it could be argued that PLA can also be positioned below the network layer. Such a case would require that every router in the network would support PLA.

It is important to note that PLA is a simple and basically stateless protocol that relies on public key digital cryptographic signatures, which are assumed to be strong. Unlike the stateful and quite complex TCP protocol, PLA is resilient against attacks on itself. The security analysis of higher level protocols that utilize PLA, and cryptographic algorithms and their hardware implementations is out of the scope of this thesis.

3.3 PLA header

Figure 6 describes an example of how the addition of the PLA header affects the structure of a normal IP packet which also utilizes IPSec.

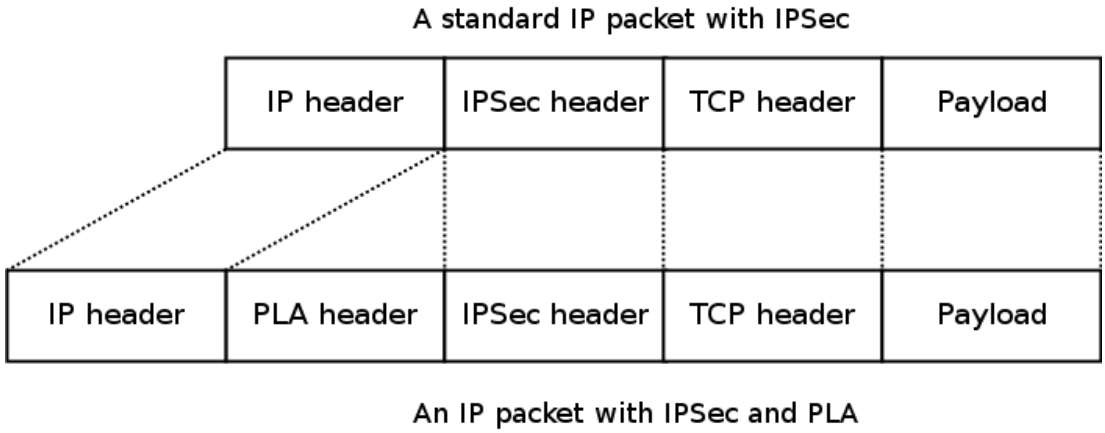


Figure 6. A comparison between a standard and PLA-enabled IP packet

In this figure, "IP header" refers to a standard IPv6 header containing source and destination addresses and ports among other fields as described in RFC 2460 [32]. In an IP header extension mechanism, the newest extension header is always added immediately after the IP header. In order to protect a whole packet, the PLA header must be added last, and thus it is positioned in the packet directly after the IP header before any additional extension headers like an IPSec header. Finally, the example packet contains a standard TCP header and a payload. The aim of this figure is also to show that PLA does not affect any higher level protocols.

An overview of the PLA header structure is shown in Figure 7. More detailed specifications of the header can be found in Appendix D. The actual PLA header is marked as a bold box and it contains the following fields.

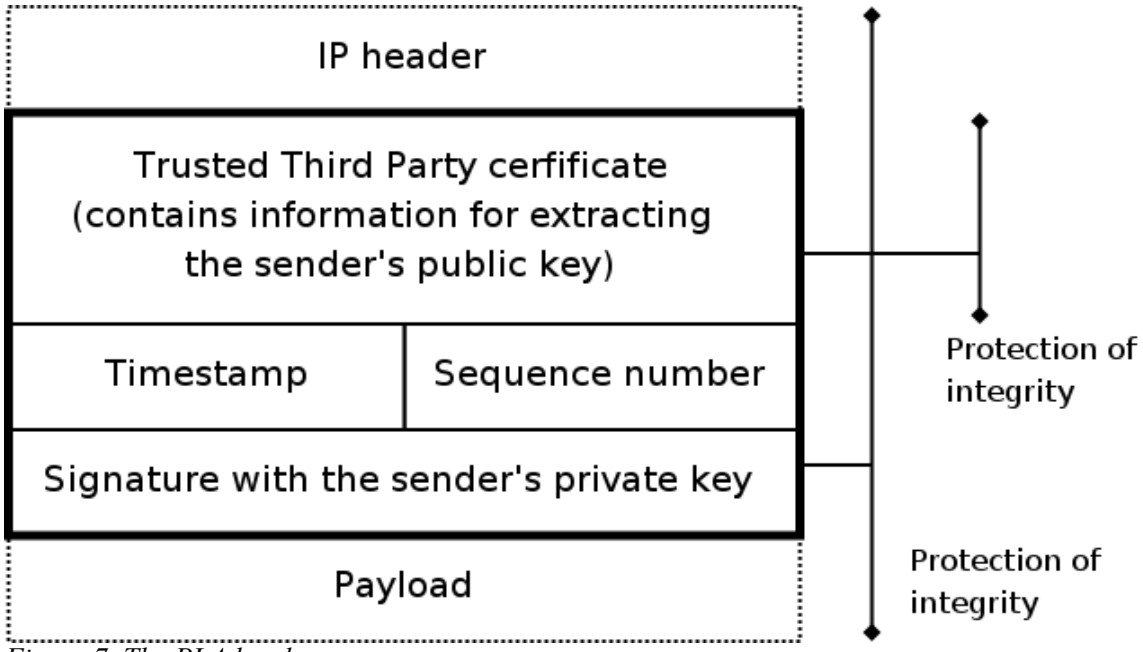


Figure 7. The PLA header

Trusted Third Party certificate

Usage: This is a certificate from the TTP to the sender. Its aim is to corroborate the binding between the sender's identity and its public key. It is also used to guarantee that the sender is a valid, trusted entity that has been authorized by some trusted third party. A TTP certificate is described in more detail in the next section.

To reduce computational and bandwidth overhead, PLA uses identity-based implicitly certified keys which are described in more detail in Section 4.1. Thus, the sender's public key is not explicitly present in the header, but can be calculated using information present in the TTP certificate. The sender's public key, together with a signature, protects the integrity of the packet and guarantees that any modifications of the packet will be detected, it also guarantees that the sender cannot deny sending a packet.

Example: The task of a TTP can be handled by a network operator, organization or a state authority.

Timestamp

Usage: The aim of the timestamp field is to detect packets that have been significantly delayed. Delayed packets can be a sign of a replay attack.

Sequence number

Usage: A monotonically increasing sequence number makes it possible to detect duplicated packets. It can also be used for per packet billing purposes.

Signature

Usage: The packet is cryptographically signed by the sender with the sender's private key. The signature guarantees the integrity of the packet; any future modification of the packet will be detected because such modification would break the signature. Since the signature is also calculated over the PLA header, the attacker will not be able to modify other fields in the PLA header like the timestamp or sequence number. PLA uses elliptic curve cryptography (ECC) [65][79] for cryptographic operations because ECC offers good security with small key sizes.

The signature calculation ignores some fields in an IP header, like the hop limit field, since that field can change during the lifetime of the packet. The PLA packet is considered fully valid if all fields in the PLA header are in order. The signature must be correct, the TTP certificate must be correct and issued by a valid trusted third party, the sequence number must be a monotonically increasing number, and the timestamp should be recent enough according to the security policy used.

Figure 8 describes relationships between various identities in the UML format. The trusted third party is responsible for storing the user's real identity. Both the TTP and the user may posses an arbitrary number of cryptographic identities, which in turn are included in the PLA header. Such an approach allows a good trade-off between the anonymity and security since the user's cryptographic identities act as pseudonyms. In a case of misuse, the user's real identity can be determined using the PLA header information and TTP records.

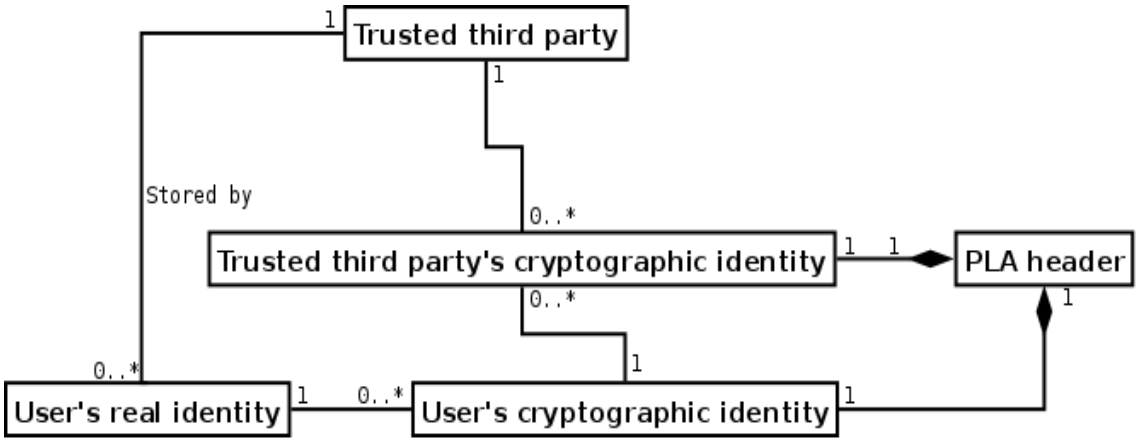


Figure 8: Relationships between PLA components.

3.4 Lightweight PLA header

In order to reduce both the bandwidth and computational overhead of PLA, there is an option to use a lightweight PLA header. Otherwise it contains the same security properties as the full PLA header, but the TTP certificate is omitted. The idea is that nodes which are handling packets cache information about the TTP certificate and thus the sender does not need to include those fields in each header. However, it is up to the sender to decide how many packets are sent with a lightweight PLA header. The lightweight PLA header is presented in Figure 9.

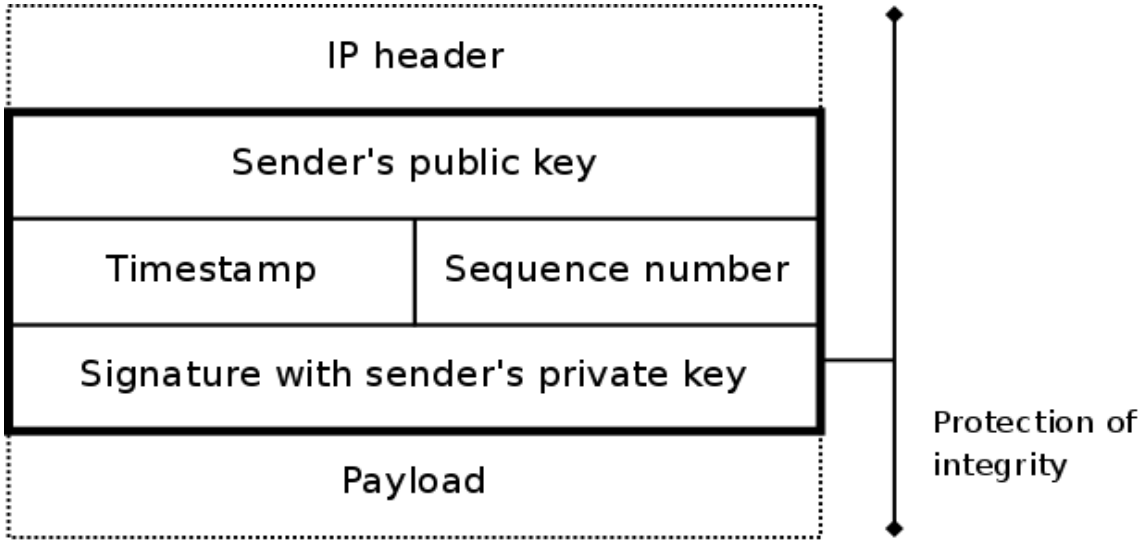


Figure 9: The lightweight PLA header

As a downside, the lightweight PLA header increases the complexity of the packet handling in routers, since they need to cache the TTP certificate. However it decreases the bandwidth overhead, and since the sender's public key is included directly in the

lightweight PLA header it also decreases the computational complexity of the packet verification. In order to make it possible for routers to validate the packet, the sender must naturally include a full PLA header in the first packet of the connection and occasionally send packets with a full PLA header to refresh routers' caches.

3.5 Trusted Third Parties

Simply including a sender's public key with signature in a PLA header is not enough. An attacker could generate a large amount of different public keys for itself and launch an attack using thousands or even millions of different keys. Therefore, even if only a small amount of packets are sent using a single public key, the attacker could paralyse its victim or the network by flooding. To protect the network infrastructure from such attacks, separate trusted third parties are required.

In the scope of PLA, a trusted third party (TTP) is an entity that provides a binding between the user's identity and its public key and authorizes users who want to communicate using PLA by granting them certificates[1]. The TTP can be, for example, an operator, in which case the TTP would grant certificates to valid users of this operator, or a state authority, which would grant certificates to its citizens. The TTP certificate is included in the PLA header and it can be viewed as proof that the node is a well behaving entity that can be trusted. The TTP certificate has a limited validity time, after which it must be renewed. If the user has engaged in malicious behaviour, then its TTP certificate will simply not be renewed and the user will not be able to communicate using PLA. A TTP certificate information is utilized when validating packets going through the network, the packet must have a valid TTP certificate from a trusted TTP in order for it to be considered fully valid.

An overview of the TTP certificate format is shown in Figure 10. More detailed certificate specifications are presented in Appendix A using the S-expressions [93] format. Fields of the TTP certificate are:

Signature part of TTP certificate. This field is used for calculation of the sender's public key, and it also implicitly signs the TTP certificate. If the calculated sender's

1 In the scope of PLA, TTP certificates contain rights and are similar to standard authorization certificates.

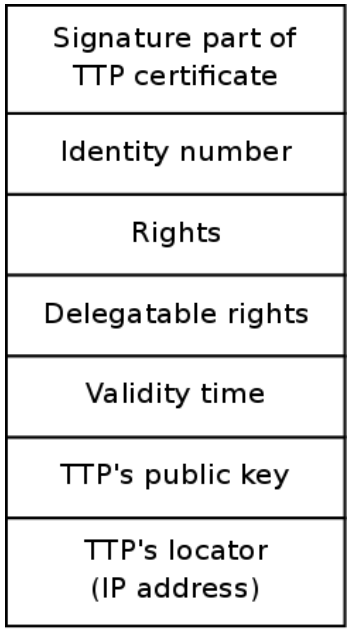


Figure 10. The TTP certificate format

public key successfully verifies the whole packet, then it also means that the TTP certificate is valid.

Identity number. This is the unique identity number given by the TTP to the user. The identity number is necessary for identity-based implicitly certified keys and it is utilized during the calculation of the sender's public key.

Rights granted by the certificate. TTP certificates can have different rights that are expressed in this field.

Delegatable rights. This field contains information about which rights can be delegated forward to another party. Delegation of rights is useful in some cases. For example, if a user wants to use another device temporarily as his own, he could delegate his existing TTP certificate to that device.

Validity time of the certificate. This field contains not-before and not-after timestamps that denote the time frame during which the certificate is valid.

Public key of the TTP. The TTP's public key is used for calculation of the user's public key.

Locator of the TTP. Which contains the IP address of the TTP in order to allow nodes to contact the TTP that has authorized the sender.

In order for the TTP system to be effective, there should exist a way to guarantee that malicious users will not be able to have a valid TTP certificate for a prolonged period of time. Since there are hundreds of millions of users on the Internet, having a centralized revocation mechanism for all TTP certificates is not feasible. On the other hand, users who have been offline for extended periods of time should be able to continue normal communication without major issues. This problem can be solved using multiple certificate types with different rights and validity times. The type of the TTP certificate is checked at every node and the packet is prioritized accordingly. Different TTP certificate types are subsequently explained.

Since PLA relies on public key signatures, it is important that the private key is kept private. There are several mechanisms to accomplish this. For example, the private key can be stored in cryptographic hardware, or on a external start card of memory stick, encrypted by the user's password. PLA is indifferent to how private keys are stored, as long as they are stored securely. In case the private key is lost, the user should contact the TTP that has issued the key to revoke the corresponding certificate. Basically, this is equivalent to cancelling a lost credit or SIM card.

3.5.1 TTP certificate types and their usage

Table 1 shows an overview of TTP certificate types and their usage. The basic aim of the TTP certificate types is to allow the network to distinguish different kinds of traffic. Then the network's policy can decide how exactly different traffic should be handled.

Table 1: Overview of the most common TTP certificate types

Certificate Type	***Rights***	***Validity time***	***Usage***
Normal	Normal traffic	Hours or minutes	Normal end-user traffic
Signalling	Limited bandwidth	Years or months	Retrieval of the normal TTP certificate
Self-signed	Only allowed to the first PLA enabled router	Any	Beginning of the bootstrapping phase
Priority	Highest priority	Hours or minutes	Network management, emergency traffic

It is important to note that the TTP certificate mechanism is flexible, and it is easy to create additional certificate types. For example, in addition to the normal TTP certificate, the operator may offer its users a certificate with a higher service quality and priority. The most common certificate types are explained below in detail.

A signalling certificate with limited bandwidth and long validity time.

This certificate type is designed to be used for retrieval of short-term certificates instead of normal traffic. Such a certificate is needed in cases where a device has been offline for a long time and its normal certificate has expired. The signalling certificate has a long validity time but a very limited bandwidth in order to reduce risks associated with such a long-term certificate. For example, routers can reserve a small amount of their bandwidth, such as 1%, for traffic that uses long-term TTP certificates[1].

Since the validity time of the certificate is long, there should be a method to revoke certificates. Each TTP should maintain a list of signalling certificates that have been issued and revoked by this TTP. Then, the status of the certificate can be verified by querying the TTP that has issued the certificate. This method resembles the Online Certificate Status Protocol (OCSP) [81] revocation method. The validity time for such a long-term certificate is on the order of months or years. For increased security, an issuer of the long-term TTP certificate may require that the user renews such a certificate offline in person, or by other means.

Self-signed (issued by the sender) certificate

A self-signed certificate is designed to be used in situations where the node does not have any kind of TTP certificate. This can occur when a signalling certificate has expired or when a new node is connected to the network for the first time. The traffic sent using self-signed certificates should be allowed only to the nearest router, for example, a WLAN access point. Afterwards, the router should encapsulate these packets in its own PLA header. The PLA encapsulation mechanism is discussed in more detail in Section 3.5.3.

Such encapsulation provides several benefits. There will be no packets in public networks that cannot be traced since the sender of every packet will be authorized by

1 Routers can detect different TTP certificate types and prioritize traffic accordingly.

some valid TTP. In addition, the router will be responsible for the packets that it encapsulates, which means that it will choose carefully how much traffic that uses self-signed certificates the router will encapsulate. The router could allow only a few connections per minute using self-signed certificates with a very limited bandwidth per connection. This should be sufficient to retrieve a valid TTP certificate for those who need it, like a completely new computer or a computer whose long-term certificate expired while it was offline.

Priority certificate

This certificate type has the highest priority in the network, otherwise it resembles the normal traffic certificate. The priority certificate can be used internally for network management messages since these messages should be transmitted in a reliable way even when the network is under attack and has limited resources.

This certificate type may also be used by authorities in case of emergencies, to guarantee that relevant messages get through even if the network is overloaded.

3.5.2 Management of Trusted Third Parties

In order for the TTP system to be effective, there must be a way to check the validity of the TTP itself. Otherwise, the attacker could create its own TTP which would issue valid TTP certificates to a large amount of the attacker's nodes.

We use a system where TTPs form a tree-like hierarchy similar to DNS servers. Each TTP is trusted by some other TTP that resides on a higher level. On the top level the number of TTPs is relatively small and thus they can form explicit security associations between them. When a new TTP enters the network, it must be authorized by some existing TTP using the TTP certificate format discussed previously. As a result, each TTP has a certificate chain that contains certificates for all TTPs in the chain starting from the root TTP.

Figure 11 contains an example of how the validity of the unknown TTP can be verified. In this example, a router or a destination has received a packet with an unknown TTP X and wants to verify its validity. First, the router sends a verification request to its local TTP E containing the public key and locator of the unknown TTP X. If TTP E does not

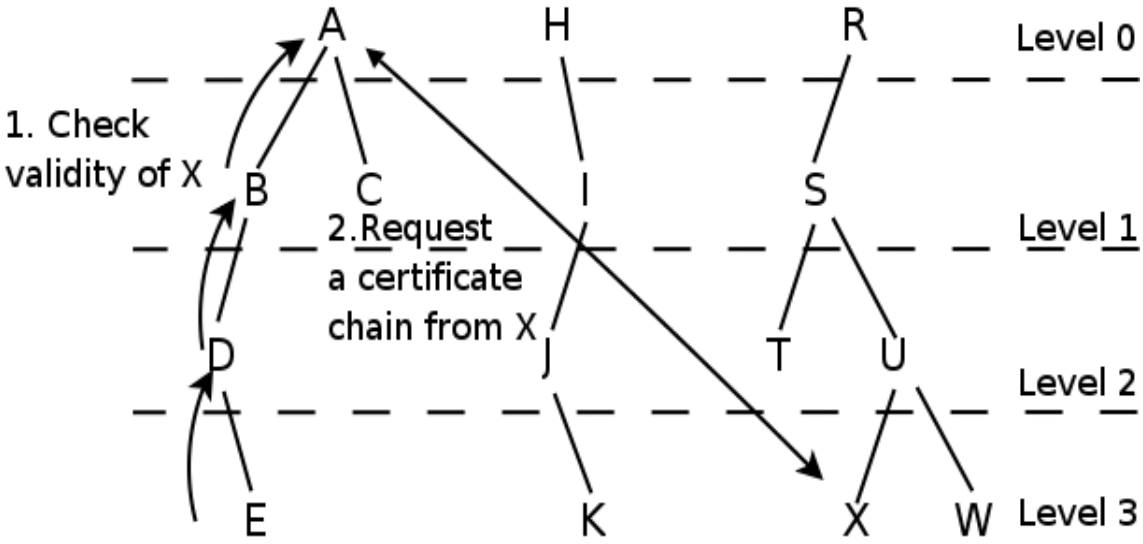


Figure 11. An example of verifying the validity of the unknown TTP

have the validity information about the unknown TTP X in its cache, it forwards the verification request to its parent. If the parent has the validity information about the requested TTP X it sends it back in a reply, otherwise it sends a request upwards to its parent and so on. When the request reaches the top, the top level TTP A will contact TTP X to request X's certificate chain. This chain contains a list of TTPs from the top level down to TTP X, i.e., TTP R => S => U => X. TTP A checks that all the certificates in the chain are valid and have not been revoked. Since TTP A trusts the top level TTP R, it can also trust other TTPs in the chain including X. TTP A adds validity information about X in its cache and sends a reply to the verification request downwards and the reply eventually reaches the router that made the original request. When a TTP or a router receives a verification reply, it stores the verification information (validity status and duration) in its cache, thus it will not be necessary to send another request as long as the cached validity information does not expire.

The large-scale management, revocation, and renewal of cryptographic keys is a very challenging task. The mechanism described above is a one possible way to accomplish it, and this field requires a further study.

3.5.3 Encapsulation of PLA headers

The overall number of TTPs on the Internet can be very high (several millions), which presents a problem with the performance and scalability of PLA. If a router receives packets from senders that have been authorized by a large number of unknown TTPs,

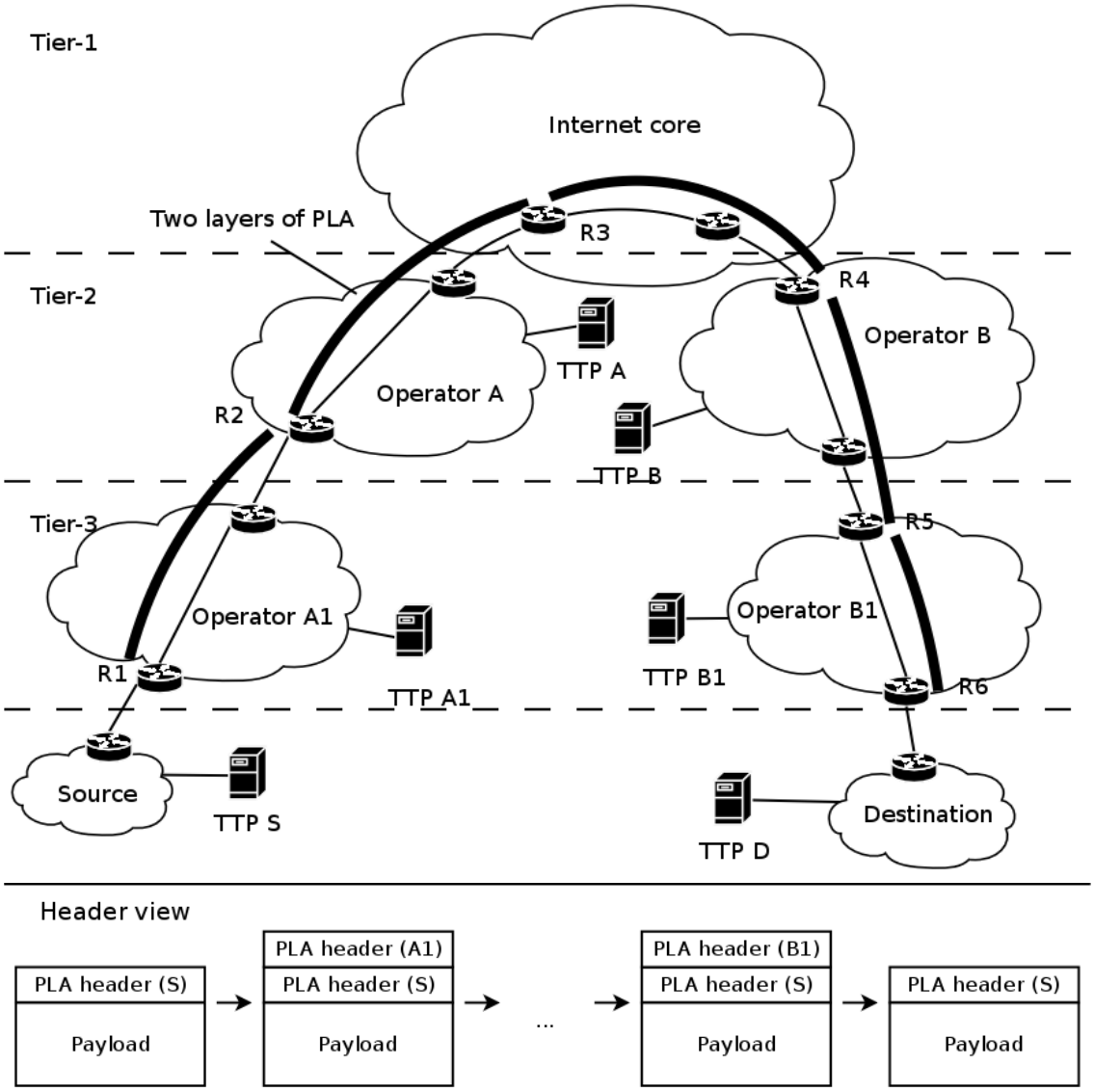


Figure 12. An example of PLA encapsulation

the router must verify each TTP independently and such verifications will use a lot of time and network resources. This problem can be mitigated by encapsulating PLA packets that travel through the Internet into another PLA header. After a packet has been encapsulated, only the outer PLA header will be verified, significantly reducing the number of trusted TTPs. An example of such encapsulation is shown in Figure 12.

In the figure, clouds denote different networks which form a hierarchy: customers are connected to the Internet through their operators, small tier-3 operators are connected to larger tier-2 operators and a limited amount of the largest tier-1 operators form the Internet core network. For simplicity, only two layers of operators are shown in the figure; in reality, the hierarchy would be deeper and big operators would serve a large

number of smaller operators. Thin lines in the figure denote ordinary PLA connections while bold lines denote encapsulated connections where two layers of PLA are used. The lower part of the figure shows PLA headers of packets in different parts of the network.

In this example, a node in the “source” network which is connected to the Internet through operator A1 is communicating with a node in the “destination” network, which uses operator B1 for Internet connection. The sending node is authorized by a local TTP S. The encapsulation works as follows: as the packet arrives to router R1, which is an edge router of operator A1, the router verifies the packet's validity including the validity of TTP S which has authorized the sender of the packet. If this validity check is successful, the router R1 encapsulates the packet in its own PLA header and sends the packet forward. This means that the new header includes R1's public key and TTP certificate received from TTP A1, since TTP A1 is responsible for operator A1's network[1]. As the packet travels through A1's network, intermediate routers will only check the outer PLA header of the packet, thus they will not need to trust TTP S that has authorized the original packet. Eventually the packet reaches router R2 which resides at the edge of operator A's network. Router R2 performs a validity check on the packet, including a check on validity of the TTP A1, strips the outer PLA header from the packet, encapsulates the packet in its own PLA header and sends the packet forward. The new PLA header naturally contains a TTP certificate from TTP A. The example continues in a similar way, as the packet arrives in a new network, an edge router replaces the outer PLA header with its own header. Eventually the packet arrives to the operator of the destination network, where router R6 strips the outer PLA header and sends the original packet to its final destination.

The advantage of such a system is that only the end point of the connection needs to perform the costly validity check of TTP S that has authorized the sender of the packet, this validity check resembles a reverse DNS check performed today. Operator edge routers need to trust in the TTPs of their “child” and “parent” operators, while intermediate routers inside operator networks need only trust in their own local TTP, since the encapsulation guarantees that every packet that is transmitted in an operator's

1 Use of encapsulation would also mean that CGA address checks would fail, since CGA addresses are generated from the sender's public key which is different from the public key used for encapsulation. However, this does not pose a significant problem if the CGA address check is performed only in access networks where packets are sent without encapsulation.

network is sent by a node which is certified by that operator's own TTP. Inside the core network, routers would only need to trust the TTPs of major operators. In this example, router R3 would need to trust TTP A. Therefore, this approach significantly reduces the amount of TTPs that intermediate routers must trust, in most cases a router need only trust its local TTP, and thus the overhead produced by TTP verification queries is greatly reduced. On the downside, the encapsulation produces some extra bandwidth overhead because there are two PLA headers in each packet, and requires more computational power in routers which perform the encapsulation since a new signature must be generated to every packet. However, encapsulation is still feasible to use within operator networks since they usually have a large amount of bandwidth available and routers will have powerful dedicated hardware to perform cryptographic operations.

Theoretically, routers could remove the original PLA header altogether and just replace it with their own header. Such a system would satisfy the requirement that only valid packets are forwarded in the network. However, in order to satisfy the rest of the requirements presented in Sections 2.5 and 3.1, the sender of the packet must be identifiable and thus the original PLA header must be present in every packet.

The use of encapsulation also means that there is no need to have a centralized revocation scheme for TTPs of major operators. If an operator's TTP becomes compromised, then the operator simply informs its “parent” and “child” operators of this and their nodes will stop trusting the compromised TTP. The number of minor TTPs can be very high, and thus it is more effective to use certificates with short validity times (couple of hours) to manage TTPs. Each TTP will receive a short-time certificate from its parent TTP and this certificate must be renewed. If the TTP becomes compromised, its parent will not renew the certificate and the TTP will not be trusted by other parties.

3.6 PLA header verification procedure

The verification of the PLA header consists of several steps and a basic case is shown as a state diagram in Figure 13. The aim of this example is to show what steps are always present in full PLA header verification and what are possible outcomes. This example assumes that the TTP certificate inside the header is a normal short-term certificate without bandwidth limitations.

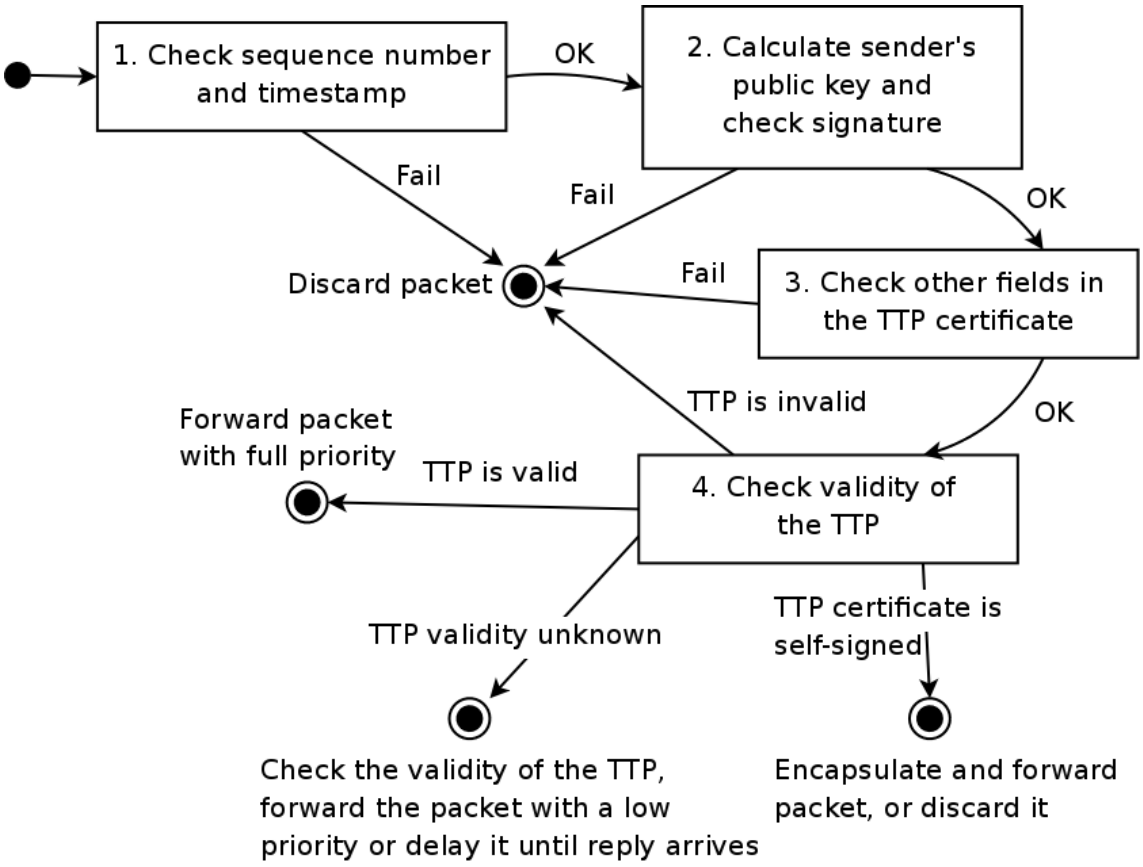


Figure 13. A state diagram of PLA header verification

The exact order of steps may differ, in certain cases performing header verification in different order might be more efficient in terms of computational resources or offer better protection against denial-of-service attacks. The main aim of PLA header verification is to check whether a packet is authentic and how much a sender of the packet can be trusted. Based on the outcome of these tests, a node will determine how to handle the packet.

In the beginning of the verification procedure, the timestamp and the sequence number are checked. If either of them is invalid then the packet has been significantly delayed or duplicated and is thus discarded. In the second step, the sender's public key is extracted from the TTP certificate and the cryptographic signature of the packet is verified using the extracted public key. If this verification fails, the packet is inconsistent and is therefore discarded. In the third step, other fields of the TTP certificate, such as the validity time and rights of the certificate, are checked. If the TTP certificate has expired or if it does not have valid rights, the packet is also discarded.

In the final step of the verification procedure, the validity of the TTP which issued the certificate is checked. If the TTP is invalid (cannot be trusted), the packet is discarded. If the TTP is known to be valid, everything is in order and the packet is forwarded with full priority. If the TTP's validity is unknown, the node checks the validity of the TTP by sending a request to a TTP of its own. While waiting for a response, the node forwards the packets in question with low priority, or alternatively, the node could delay the packets while awaiting a reply regarding the TTP validity. The latter option would enhance security, although it would introduce an additional latency for the initial packet. Finally, if the TTP certificate is generated by the sending node of the packet (i.e., it is self-signed), the node has two options. Either the node simply discards the packet, or the node can encapsulate the packet in its own PLA header and forward it. Encapsulating the packet would mean that the node is taking the responsibility for the packet, and thus the node should only allocate a small portion of bandwidth for such packets with self-signed TTP certificates. Self-signed certificates are only used when the sending node has just started operating and does not have any valid certificate from a TTP yet.

The router has a lot of flexibility in terms of performance/security trade-off when deciding how to prioritize packets. The router can adopt a “paranoid” policy, in which case it would:

- Delay or discard packets which were sent by a user that has been authorized by an unknown TTP, until the validity of the TTP has been verified.
- Query the status of encountered signalling certificates from TTPs that have issued those certificates, and allocate very limited bandwidth to packets that use a signalling TTP certificate.
- Discard all packets that use self-signed TTP certificates.

Or a more relaxed policy, like:

- Forward all packets immediately, even if their TTP is unknown.
- Encapsulate packets that use self-signed certificates in router's own PLA header.
- Allocate more bandwidth to self-signed and signalling certificates.

Most importantly, PLA offers routers a consistent way to decide how different kinds of packets should be handled. Routers can then make a decision depending on their policy. The optimal policy for each router depends on several factors and determining the optimal policy is beyond the scope of this thesis. For example, routers in civilian and

military networks may have a very different policy for handling packets, just like routers in access networks may have a different policy compared to routers in core networks.

3.7 Reacting to attacks and reporting misuse

To achieve a high level of security in the network, two goals should be fulfilled. First, the attack should be stopped as quickly as possible, preferably close to the source. Secondly, there should be a way to identify culprits behind the attack in order to prevent them from continuing the attack, and to make them responsible. This section provides an overview how attacks can be stopped and misuse reported with PLA. Sections 6.2 and 7.1 contain more detailed description of these issues.

3.7.1 Stopping attacks

If the network utilizes CGA addresses or some other mechanisms to tie the cryptographic identities into network-layer identifiers, such as IP addresses, stopping the attack with PLA is straightforward. A node that has come under an attack sends the following *STOP* message back to the attacker, indicating that the node does not want any additional traffic from the attacker.

Victim ⟶ Attacker:

{STOP, $PK_{ATTACKER}$}

Or

{STOP, $PK_{ATTACKER}$, Timeframe}

Routers on the path will verify the integrity of such a message, and will check that the source IP address is not spoofed. If everything is in order, routers would add the <$PK_{ATTACKER}$, IP_{VICTIM}> pair to the internal blacklist, therefore stopping the traffic from the attacker to the victim. Naturally, it is not reasonable to expect every router on the path to co-operate with such messages. The return path may differ in some parts, fragmentation may occur within the network, the size of the blacklist may be significantly limited in intermediate routers, or some routers may just ignore *STOP* messages. However, such a scheme does work even if just one router takes note of the stop message, and therefore the traffic to the victim is stopped before it reaches its destination. Since the amount of intermediate hops on the Internet is usually about ten or

more, there is a high probability that at least one router will conform to the *STOP* message and will stop the unwanted traffic.

Even if only one router conforms to *STOP* the effectiveness of the scheme can be gradually improved. Suppose the traffic between the attacker A and the victim V flows along the path: A – X – Y – Z – V. Router Z near the destination conforms to the *STOP* message, while other routers ignore it. Router Z can easily notice the situation, since it is receiving traffic that should have been blocked already. In this situation, router Z would send another *STOP* message towards the attacker, either its own *STOP* message or the cached original *STOP* message from the victim. As a result, the point where the unwanted traffic is stopped would be moving towards the source of the traffic, freeing resources in the rest of the network.

Using *STOP* messages also introduces accountability between the operators. Let us assume for simplicity that in the previous each router is managed by the distinct operator. If operators W, X and Y would continue to ignore the stop message, then operator Z would have proof of such negligence, since operator Z would possess both the original *STOP* message from the victim containing the attacker's cryptographic identity and the timestamp, and subsequent packets from the attacker to the victim with later timestamps. Such a strong accountability provides incentives to all operators to behave according to the rules.

3.7.2 Reporting misuse

Since PLA is based on cryptographic identities and signatures it provides strong accountability features. The victim of the attack possesses packets sent by the attacker, which contain the attacker's cryptographic identity together with the TTP's public key and locator. Therefore, the victim can report misuse to the TTP that has certified the attacker, and present the original packet sent by the attacker as proof. Such reporting can be automated to some extent, and handled by, for example, the firewall software.

If the TTP receives enough proof[1] against the attacker, it will not renew the attacker's certificate, preventing him from continuing the attack after the current certificate

1 There is no one correct solution for how the TTP should determine whether it should not renew the user's certificate. This is more of the policy- and legislation-level issue, which is out of the scope of this thesis.

expires. The rights of the attacker can be revoked either permanently or temporarily by the TTP depending on the policy. For example, if the misuse is serious, the TTP may require the user to contact the TTP offline before a new certificate is granted. Additionally, packets sent by the attacker provide evidence to start an investigation if the misuse has been serious enough.

In some cases, for example if the computer has been infected with viruses, it may not be possible to reveal identities of the real attackers. However, in this case the infected computer will be prevented from sending data to the network, and therefore PLA still fulfils its goal of providing availability.

3.8 Bootstrapping a new node to use PLA

PLA relies on certificates from trusted third parties and a new device must somehow retrieve all relevant certificates for communication. This section describes one possible bootstrapping procedure for a new device, i.e., how a new device can retrieve certificates to communicate fully using PLA. There are basically three distinct use cases for bootstrapping. First, the TTP certificate can be already pre-installed, for example, the customer buys a laptop or makes a contract with the operator. In this case the TTP certificate would be present on the device or the SIM card. The second alternative is that the user is a completely new customer of the TTP. The last case is the situation where a new device is acquired by the user that already possesses a valid TTP certificate.

The first use case where the certificate is pre-installed on the device is trivial from the bootstrapping point of view, therefore this section concentrates on the second and third case. The bootstrapping example shown in Figure 14 assumes that the device is brand new, it does not have any certificate from a TTP and it does not even have a public/private key pair. It is also assumed that the device stores its private key locally. It would also be possible to store a private key in some kind of secure portable device, in which case the user could carry his private key with him and use it with several devices.

After the new device is turned on, it moves to the key generation state where it generates a public/private key pair for itself. In step 2, the device generates a self-certificate that has the same format as a TTP certificate but is signed with the device's own private key. Now the device has all of the necessary keys and certificates for limited communication,

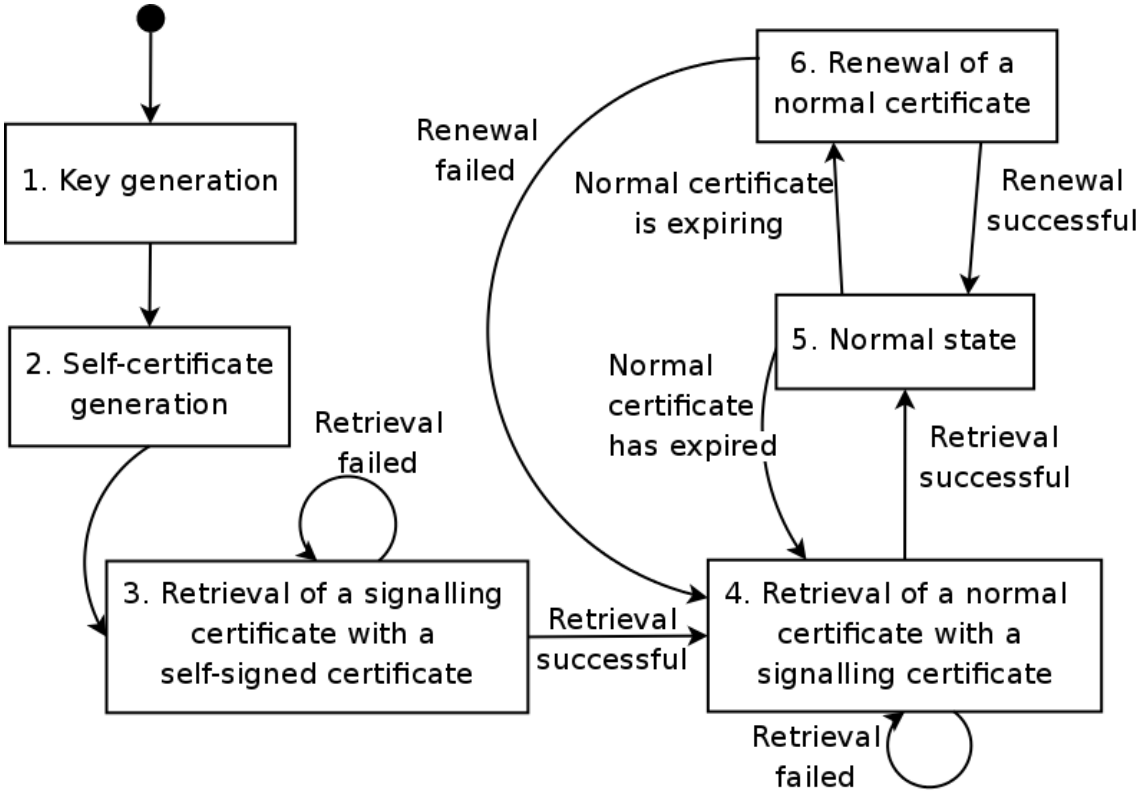


Figure 14. The bootstrapping procedure of a new node

and it can retrieve a long-term signalling certificate from a TTP with its own self-signed certificate in state 3. Afterwards, the device retrieves a normal short-term certificate from a TTP using the previously received signalling certificate. If this retrieval is successful, the device enters normal state 5, where it can communicate normally using PLA without any bandwidth limitations. When this normal short-time certificate is about to expire, the device needs to request renewal from the TTP in state 6. If the renewal fails or if the normal certificate has already expired, for example because the device was turned off for a long period of time, then the device moves back to state 4 where it needs to retrieve a normal certificate. Theoretically, also the long term signalling certificate can expire, in which case the device would move back to step 3. This step (retrieval of the initial signalling certificate from the TTP) is discussed in more detail in the next section.

3.8.1 Retrieval of the initial certificate from the TTP

Figure 15 illustrates how a new device can retrieve an initial certificate from the TTP. The certificate format used in the figure is an S-Expression format which is described in Appendix A. In this case, the TTP service is provided by an operator with whom the

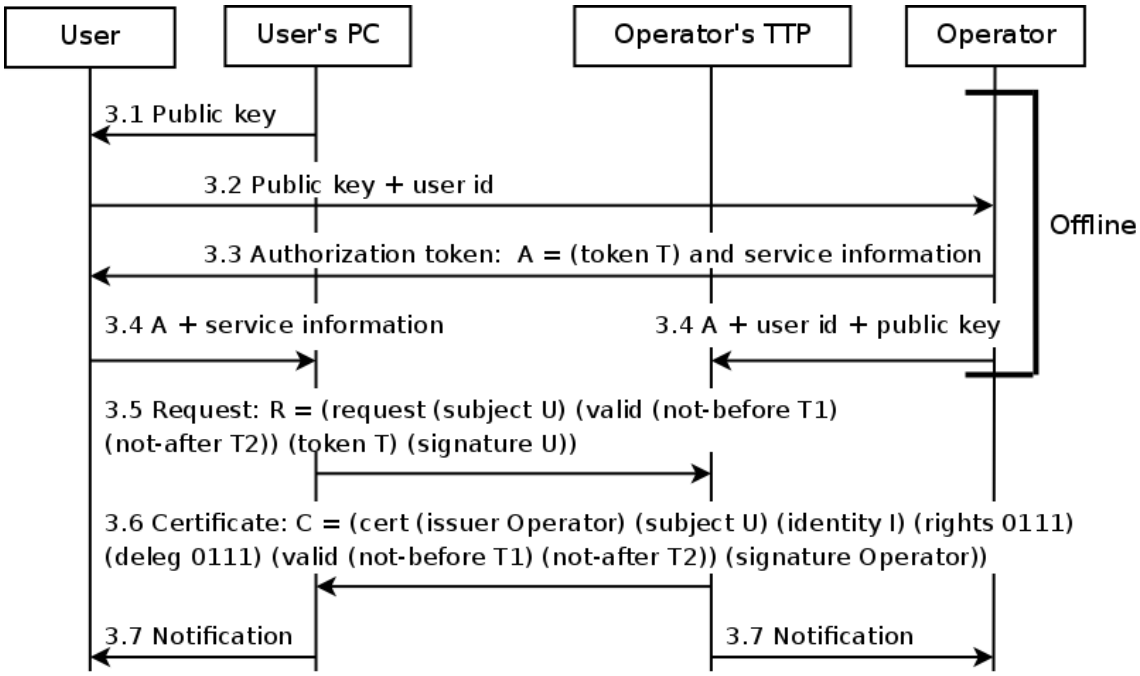


Figure 15. Process of retrieving initial certificate from the trusted third party

owner of a computer has a contract and the user requests a TTP certificate for his PC. The idea is that the operator provides an authentication token to the user, and the token is used by the user to prove his identity during the certificate retrieval. The token acts as a one-time password which is bound to a specific public key and the token can be exchanged offline between the user and the operator.

The retrieval process proceeds as follows. In steps 3.1 and 3.2, the user retrieves the public key[1] of the PC and sends it together with some kind of user id, such as a customer number, to the operator. In step 3.3, the operator gives the user an authorization token containing service information such as the address of the operator's TTP. Then in step 3.4 the user enters the service information to their PC while the operator sends the given authentication token, user id, and the public key to its TTP. In step 3.5, the request for a new certificate is sent. This request contains a public key, a requested validity time for the certificate, the authentication token, and a signature. The operator's TTP now verifies that the public key and authentication token in the request match the values sent by the operator. If they match, the TTP sends a final TTP certificate back to the user in step 3.6[2]. In the final step 3.7, the user and the operator are notified that the certificate has been issued and received successfully.

1 The public key mentioned here is the preliminary key used during the certificate retrieval process. A final public key is extracted from information received from a TTP in step 3.6 based on formulas mentioned in Section 4.2.2.

2 To protect privacy, traffic in steps 3.5 and 3.6 must be encrypted by some means, such as TLS.

3.8.2 Delegation of TTP certificate to another device

In some cases it is useful to have the ability to delegate rights to communicate, such as rights to request a valid TTP certificate, to another device either permanently or temporarily. For example, if a user wants to use a friend's computer as his own for a while, he could temporarily delegate his rights to that computer. Or if the user wants to take into use some home appliance which lacks traditional input methods, he could permanently delegate his rights to that appliance.

In principle, delegation of rights works as follows. A valid user authorizes another device with a certificate, thus creating a certificate chain: TTP => user => device. Using this certificate chain, a third party can then request a new certificate from the user's TTP. An example of such delegation is shown in Figure 16.

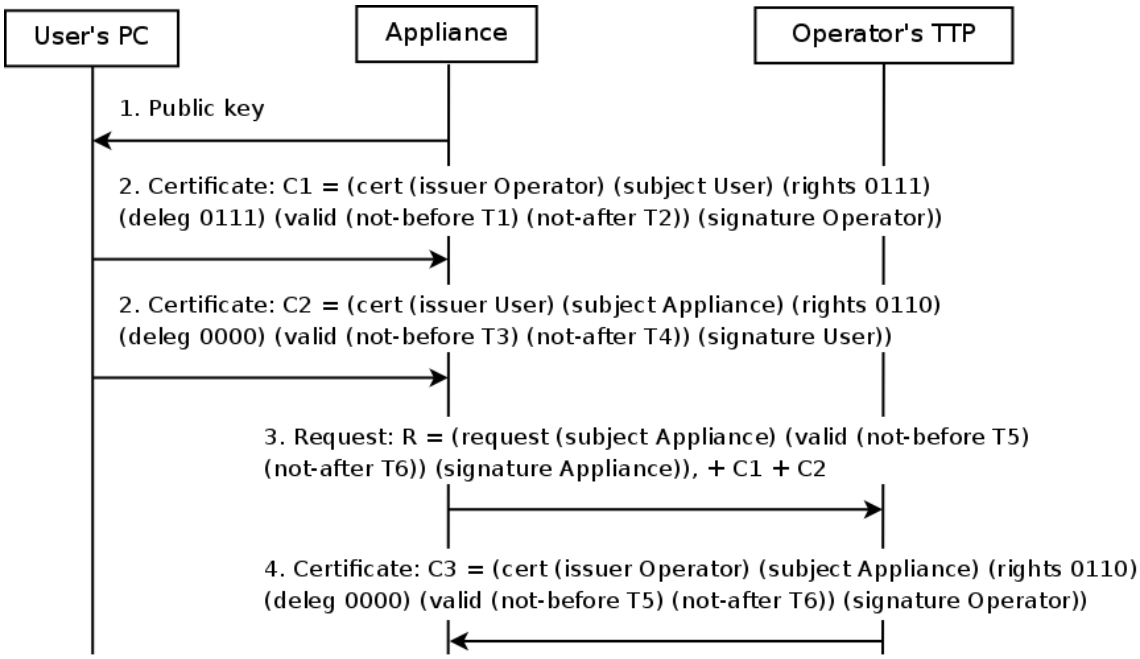


Figure 16. An example of delegation rights to another device.

In the first step, an appliance generates a public key for itself which is then transmitted to the user. The means by which data is transmitted in steps 1 and 2 is irrelevant, it can be transmitted using a network or offline using a separate memory stick. In the second step, the user transmits two certificates to the appliance: the first is the user's TTP certificate which states that the user is a valid and trusted entity, and the second is the certificate issued by the user to the appliance. These certificates form a certificate chain: TTP => user => appliance and thus with these certificates, the appliance can request its

own certificate from the TTP in steps 3 and 4. After this certificate C3 has been received in step 4, the appliance will be able to communicate using PLA.

4. Implementing PLA

This chapter describes cryptographic solutions used by PLA. It also describes proof of concept software and hardware implementations of PLA.

4.1 Software implementation

A proof-of-concept Linux implementation of PLA has been made [86]. The aim of this implementation is to show that the basic idea of PLA is implementable in practice, and PLA can be used together with existing applications. The implementation uses IPv6 and supports packet authenticity checking and offers support for hardware acceleration for signature verification. The Linux implementation consists of a kernel module and user space applications. The implementation supports three modes of operation:

1. The PLA header is added to a plain IP packet and the PLA-enabled packet is sent forward.
2. The PLA header is removed from an existing PLA-enabled packet (after checking the packet's validity) and the plain IP packet is sent forward.
3. The PLA-enabled packet is simply forwarded after performing validity checks.

The implementation also supports several PLA layers (PLA encapsulation). The proof-of-concept implementation makes it possible to build a network consisting of non-PLA-enabled nodes and PLA-enabled nodes. A general architecture of the PLA software implementation is presented in Figure 17.

As the packet arrives to the network interface, it is handled by the network stack of the operating system. The operating system contains a PLA module which will handle PLA-related operations such as adding a PLA header to a plain IP packet and verifying the validity of the PLA header. The PLA kernel module communicates with a separate cryptography module which resides in the user space. For testing purposes this cryptography module supports both software and hardware based cryptographic solutions. In addition, the implementation supports adding a PLA header to packets with a blank signature without making any cryptographic computations. This feature is useful for testing the non-cryptographic overhead of the PLA implementation.

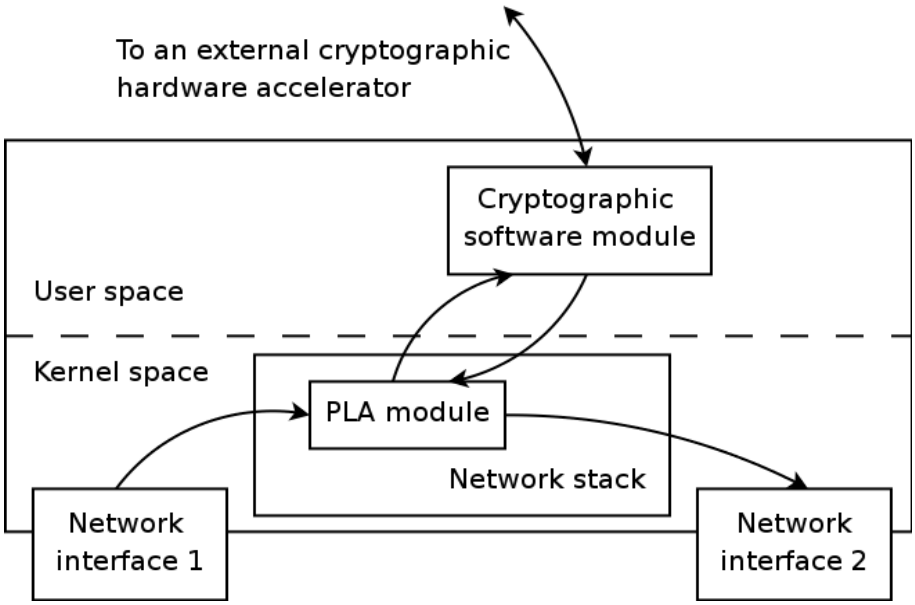


Figure 17. PLA software implementation for Linux

To allow easy experimentation, the PLA functionality has also been implemented as a separate user space *libpla* library that runs under Linux and FreeBSD [87]. The *libpla* library provides functions for adding and verifying a PLA header over a memory area. Therefore separate libraries, such as *libpcap* [106] and *libnet* [72], should be used for sending and receiving packets from the network interface.

4.2 Cryptographic solutions

Since PLA is based on public key cryptography with public keys and signatures included in every packet, the cryptographic solution used should offer good security with relatively small key and signature sizes. Therefore, instead of using the most commonly used public key algorithms such as RSA [92] or DSA [43], PLA uses Elliptic Curve Cryptography (ECC) for cryptographic tasks. ECC is a very efficient algorithm in terms of security per key size, since a 163-bit ECC key has roughly the same cryptographic strength as a 1024-bit RSA key or an 80-bit symmetric key [77]. PLA itself is not dependent on the cryptographic solutions used and can be used with different cryptographic algorithms and key lengths. However, the ECC algorithm is currently the only feasible solution, since other solutions would require significantly longer cryptographic keys and signatures to achieve the same level of security.

The PLA implementation uses a standardized Kobliz curve K-163 [43] defined over a binary field because binary field operations are fast in both hardware and software. PLA

uses 164-bit compressed public keys, 163-bit private keys, and 326-bit signatures. These key sizes provide security roughly equivalent to that of 80-bit symmetric algorithms. According to the U.S. National Institute of Standards and Technology (NIST) recommendations [11], such a level of security is sufficient for data storage until the year 2010. However, in the case of PLA, a somewhat lower level of security is acceptable, since most of the keys and packets in the network are relatively short lived.

As computational power increases in the future it will become easier to use a brute force attack to break cryptographic algorithms and stronger cryptographic solutions will be required. In this respect, ECC is a good solution because its efficiency in comparison to RSA increases as key sizes become larger. For example, increasing an ECC key size from 163 to 233 bits (an increase of 43%) would increase security as much as doubling a key size in the RSA algorithm from 1024 to 2048 bits. According to the NIST recommendations, such security should be sufficient until the year 2030.

4.2.1 Per packet signature generation and verification

Per packet signatures are generated as follows using ECC. Let a base point generator G and prime order r be elliptic curve related global parameters, s a private key, and $u \in_R \mathbb{Z}_r^*$. A public key W is generated by multiplying the base point and the private key: $W = sG$. A cryptographic signature on message m consists of two values (c,d) and is computed using the private key as follows:

$$c = [uG]_x + \mathrm{H}(m) \ (mod\ r) \quad (1)$$

$$d = u - sc \ (\mathrm{mod}\ r), \quad (2)$$

where $[\mathrm{P}]_x$ is the x-coordinate of the elliptic point P converted to an integer and H is a collision-resistant hash function. The PLA implementation uses a 160-bit RIPEMD [35] as the hash function. It would also be possible to use other hash functions offering the same level of security, such as a 160-bit SHA-1 [44]. In the case of PLA the hash is computed over the whole packet ignoring the hop limit field of an IPv6 header.

Any other party can verify the packet's signature (c,d) with a public key W by checking that:

$$\mathrm{H}(m) = c - [dG + cW]_x \ (mod\ r). \quad (3)$$

4.2.2 Trusted Third Party certificates

In order to reduce packet overhead and save computational power, PLA uses identity-based implicitly-certified keys [17], which means that it is not necessary to include the sender's actual public key in the PLA header. Generating public keys with implicit certificates for the sender works as follows. Let G and r again be elliptic curve related global parameters, ID the user's identity[1], s_{TTP} a TTP's private key, and W_{TTP} a TTP's public key. At the beginning, the user generates and sends kG to the TTP where $k \in_R \mathbb{Z}_r^*$. The TTP calculates:

$$(\bar{r}, b) = \text{COMPRESS}(kG + k_T), \text{ where } k_T \in_R \mathbb{Z}_r^* \quad (4)$$

$$r_u = \bar{r} + \text{H}(ID) \quad (5)$$

$$\bar{s} = k_T - r_u s_{TTP} \pmod r, \quad (6)$$

where COMPRESS is the point compression function giving the x-coordinate of $kG + k_T$ and the compression bit b. The TTP sends its signature $(r_u, b), \bar{s}$ back to the user who calculates his private key $s = k + \bar{s} \pmod r$ and public key $W = sG$. The signature part of a TTP certificate which is included in the PLA header is (r_u,b). In this approach the TTP does not learn the user's private key even though the TTP is involved in the key generation process, therefore the user's privacy is not compromised.

Afterwards, a verifying party can extract the user's actual public key W from the signature part of the TTP certificate (r_u,b), the user's identity ID, and the TTP's public key W_{TTP} by calculating:

$$W = \text{DECOMPRESS}(r_u - \text{H}(ID), b) - r_u W_{TTP}, \quad (7)$$

where DECOMPRESS is the point decompression function given an x-coordinate and compression bit b. If the extracted public key W successfully verifies the packet, then also the TTP certificate is valid. On the other hand, if the signature verification fails, it is not possible to determine whether the signature or the TTP certificate is incorrect. The signature verification would fail if the signature is authentic but TTP certificate information incorrect, or vice versa. This does not actually matter in the case of PLA, since the packet must contain both the valid TTP certificate and signature in order to be considered valid.

1 String ID contains several fields of the TTP certificate presented in Figure 10: identity, rights, delegatable rights, and validity time.

4.2.3 Improving the efficiency of cryptographic operations

Cryptographic solutions developed for PLA also utilize novel ideas for increasing the efficiency of various ECC-related calculations. A new method for performing the public key extraction and verification mentioned above using only a single three-term simultaneous scalar multiplication is described in [14]. Another method for improving the performance is discussed in [15]. PLA also utilizes a new method of computing the integer equivalent of random Frobenius expansions [16], which significantly speeds up the generation of cryptographic signatures. Fast signature generation is beneficial for nodes that are sending a large amount of PLA packets and for nodes that encapsulate PLA traffic.

Overall, the most computationally complex operation in the ECC algorithm is the elliptic curve point multiplication. Other operations including hash calculations are significantly less complex. The ECC signature generation requires a single point multiplication, the standard signature verification used by a lightweight PLA requires two point multiplications, and the full PLA verification consisting of the public key extraction and signature verification requires three point multiplications. Therefore, the signature generation is theoretically three times as fast as the full PLA verification.

4.3 Hardware acceleration of cryptographic calculations

Since public key cryptography is very computationally intensive, it is not feasible to use general purpose CPUs for such cryptographic computations. One of PLA's design goals is to scale to high capacity networks and to devices with low computational resources, thus dedicated hardware acceleration is required to handle cryptographic tasks. In the future, such a cryptographic accelerator can be integrated into technologies such as Trusted Platform Module (TPM) [110] and Trusted Network Connect (TNC) [109], which aim to enhance the security of personal computers and networking by using an additional security hardware module.

There are two computationally intensive problems where hardware acceleration is useful: validation of packet signatures, and generation of signatures for packets. The former operation is performed by any node that forwards PLA traffic while the latter is performed by nodes that send packets or encapsulate traffic. Both problems are very

parallel in nature since packets received by a node can be verified independently. Thus, the throughput (the amount of packets that a node can process in a given time frame) is more important than the speed of a single operation (the latency) as long as the latency is low compared to the latency of the network itself.

Figure 18 describes one possible architecture for the PLA hardware accelerator. It assumes that the accelerator is integrated on a networking hardware and PLA-related operations are performed in a transparent manner. The left side of the figure contains PHY and MAC interfaces that are also present in normal network interface chips, the right side contains a generic network processor and buffers, and the PLA-related functionality is located in the middle. While Figure 18 shows the PLA accelerator as a separate module, the accelerator could also be integrated inside the main network processor.

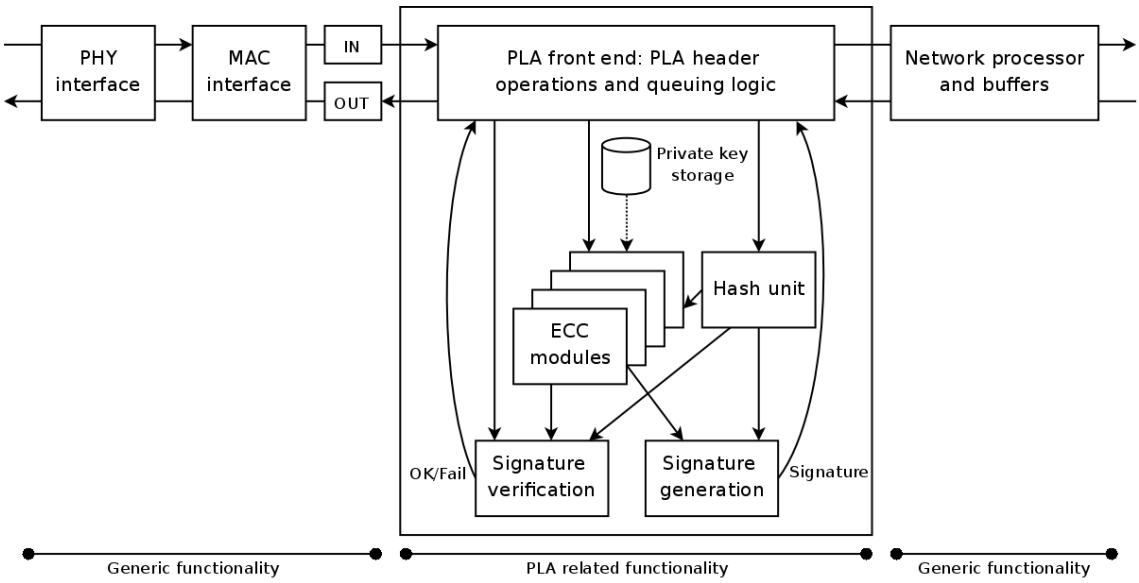


Figure 18. An example hardware acceleration architecture for PLA

The example PLA hardware accelerator works as follows. After an incoming packet passes the PHY and MAC interfaces, it arrives at the PLA front end, which is responsible for performing various PLA-related operations including the PLA header extraction and modification, verification of timestamp and sequence numbers, and queuing. For incoming packets, the front end extracts relevant information from the PLA header, such as the TTP's public key, TTP's implicit certificate, identity string and the packet's signature.

Afterwards, the relevant information from the packet is forwarded to the hash unit and ECC modules that perform ECC-related operations, and finally to the signature verification module. The module calculates equations (7) and (3) to extract the sender's public key and to verify the packet's signature, and sends the outcome of the verification back to the front end. If the verification is successful, the front end forwards the packet to the generic network processor.

Outgoing packets follow the reverse path. After passing through a network processor, they arrive at the PLA front end, and are forwarded to the hash unit and ECC modules. There also exists a storage for private keys that are used for signing. The signature generation module calculates the signature according to equations (1) and (2), and sends the corresponding signature to the front end. The front end adds the signature to the packet's header and the packet is sent out through the MAC and PHY interfaces.

Such an accelerator would work without an active intervention from the host computer. The host would just need to set up policies on how packets should be treated if they do not pass all verifications fully, and signal what private keys should be used for signing outgoing packets.

4.3.1 Proof of concept PLA hardware accelerator

A proof of concept implementation of the hardware accelerator for ECC-related calculations has been implemented for PLA [59]. The initial hardware accelerator is based on a field programmable gate array (FPGA). An FPGA is a chip containing a large amount of programmable logic, including logic gates, I/O, and memory. The main advantage of an FPGA is its flexibility. The FPGA can be programmed to perform different tasks, therefore FPGAs are very suitable for prototyping purposes. On the downside, FPGA offers significantly lower performance than the application-specific integrated circuit (ASIC) [67]. An FPGA usually runs at relatively low clock speeds but contains a large amount of computational units which can perform several computations in parallel, making it a suitable solution for the acceleration of PLA cryptographic computations.

The proof-of-concept accelerator supports the calculation of an elliptic curve point multiplication $Q = kP$, where Q and P are points on an elliptic curve and k is an integer.

The elliptic curve point multiplication is the most demanding operation of the ECC algorithm. Basically, the proof-of-concept accelerator covers ECC modules from Figure 18. An overview of the accelerator is presented in Figure 19.

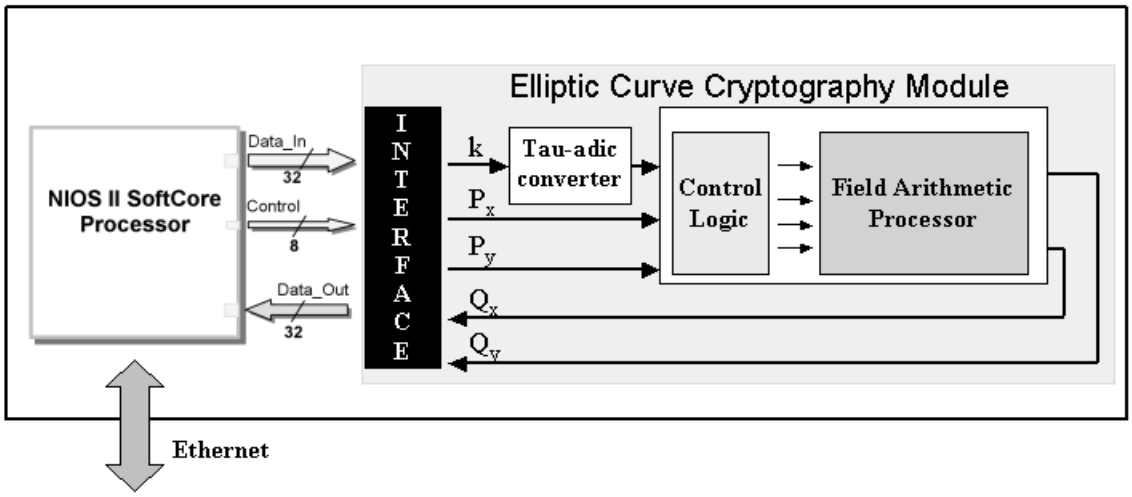


Figure 19. An overview of the proof of concept PLA hardware accelerator

The accelerator is divided into two main parts: an I/O part and the actual elliptic curve cryptography module which accelerates ECC calculations. The elliptic curve cryptography module consists of tau-adic converters and field arithmetic processors (FAPs). A FAP processor is the part of the accelerator that is responsible for performing the elliptic curve point multiplication.

Koblitz curves, which are used in the ECC implementation, require that the scalar k used in the point multiplication $Q=kP$ is converted into a so-called tau-adic expansion [64]. This conversion is non-trivial and needs to be performed by the hardware in order for the accelerator to achieve its full potential. Hence, the elliptic curve cryptography module contains separate tau-adic converters that are described in more detail in [60].

A Nios II soft-core processor is connected to the cryptography module over an internal bus and is used for I/O communication with the host computer over Ethernet. This I/O part is necessary for the proof-of-concept implementation where a majority of the workload is done on a main CPU. In a final accelerator design, the cryptography module would be directly connected to the network hardware.

The proof-of-concept accelerator consist of a separate board containing the FPGA chip. This board is connected to the host computer by Ethernet and communication between

the host and the accelerator is handled using TCP. At a high level, the accelerator works as follows; the host computer sends values of k and P to the accelerator, which performs the point multiplication and returns Q to the host computer.

Currently, the proof-of-concept hardware accelerator has some limitations. The Ethernet interface of the accelerator is limited to 100 Mbps speed, which becomes a bottleneck in certain situations. Furthermore, the PLA software implementation is not optimized and the communication latency between the host PC and the proof-of-concept accelerator is relatively high.

4.3.2 Performance of the proof-of-concept PLA hardware accelerator

An optimized design for performing verifications has been implemented based on an Altera Stratix II EP2S180C3 FPGA board. The FPGA chip used is built on a 90 nm manufacturing process and it contains 96 digital signal processing elements and approximately 1 Megabyte of memory. According to simulations the FPGA accelerator achieves 166,000 verifications per second with a latency of 114 μs per verification [61] when optimized for throughput. In this case, 19 FAP processors are used in parallel at a clock speed of 164MHz. The improved design increases the performance to about 242,000 verifications and this design can also be optimized for latency, in which case the latency per verification decreases to 35 μs [58].

The same design can also be used for performing signature generations and for lightweight PLA verifications where the public key extraction is not necessary. Theoretically, the performance should increase by 50% and 200% for lightweight verifications and signature generations respectively, since these operations require two and one point multiplications instead of three. However, due to optimizations, the actual performance increase is 17% and 166% respectively, i.e., the design can perform 283,000 lightweight verifications or 645,000 signature generations [58]. The performance implications and scalability of cryptographic operations is discussed in Section 5.3.

4.4 Summary

The PLA functionality has been implemented as a software-based prototype that runs under Linux and FreeBSD in user space. There also exists a kernel space implementation for Linux. Furthermore, a FPGA-based hardware prototype has been created to accelerate generation and verification of cryptographic signatures.

The software-based proof-of-concept prototype shows that PLA can be implemented and deployed in real-life systems. The hardware accelerator achieves a promising performance, and its performance implications on the scalability of PLA will be further evaluated in the next chapter.

5. Analysis of the PLA implementation

The first two sections of this chapter analyse how well PLA meets its design criteria presented in Section 3.1 and the requirements for the next-generation Internet presented in Section 2.5. Section 5.3 contains an analysis of PLA's performance and an energy overhead. Finally, Section 5.4 includes a comparison with hash tree and hash chain based security solutions in terms of security properties and energy efficiency in wireless networks. Hash tree and hash chain based solutions are included in the comparison since they offer similar hop-by-hop security properties as digital signatures.

5.1 PLA design criteria analysis

The design criteria of PLA are divided into three parts: mandatory, important, and optional criteria.

5.1.1 Mandatory criteria

Compatibility

Since PLA simply adds its own header to an IP packet using a standard extension mechanism, it is compatible with existing IP networks. Routers that do not support PLA can forward PLA packets simply based on IP header information. In such cases, traffic would continue as usual but features of PLA would not be utilized. PLA can also be used together with other IP-based protocols and security solutions such as IPSec or HIP.

In a hypothetical situation, a router could add a new header on top of the PLA header, i.e., between the IP header and the PLA header. In such cases, there are two possible outcomes for the rest of the network. The first alternative is that the packet is forwarded as a plain IP packet, ignoring the PLA header altogether. Alternatively, if the routers are intelligent enough, they can retrieve the PLA header from the packet even if there exist other additional headers between the IP header and the PLA header. Thus, they could still use information from the PLA header to check the integrity of the original PLA secured packet. However, in such situations it would not be possible to protect the integrity of additional headers, because these headers would have been added after the PLA header.

Deployability

PLA is compatible with current IP networks, thus it can be deployed gradually. In addition, PLA does not require separate security associations between nodes because all information required for packet verification is included in the PLA header. This is very useful in dynamic network environments such as ad-hoc networks, where the network topology can change frequently. Even if the path along which packets travel changes significantly, every node along the path can fully verify packets if PLA is used.

To facilitate easier deployment, two different kinds of PLA-enabled routers can be used during the early deployment phase as shown in Figure 20. The network would contain a limited amount of smart routers which can fully verify PLA header information and are able to perform signature verifications at wire speed. In addition, there would be simple routers without dedicated hardware for cryptographic operations that could verify various PLA header fields like timestamp and sequence number. Simple routers would also support traffic throttling and blocking based on various parameters like the sender's public key or the trusted third party that has authorized the sender. They can be implemented by making small modifications to existing IP routers. Since PLA is based on the IP header extension mechanism, the network can also contain ordinary routers that are not aware of PLA.

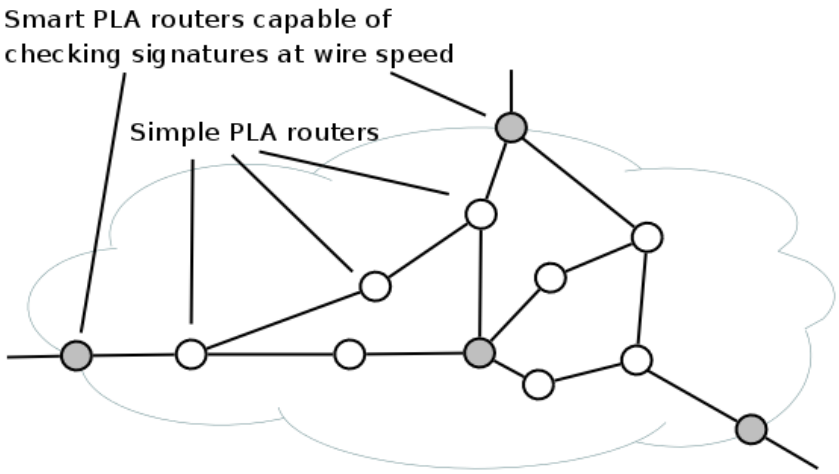


Figure 20: An example of gradual deployment of PLA inside a domain

In the early stage of deployment, most PLA-enabled routers in the network would be simple routers, while smart routers would be located on the border of the network and at strategic points. Therefore, all traffic coming to the network from outside sources will be verified by smart routers and smart routers inside the network would improve detection of attacks originating within the network.

To improve security in the rest of the network the following approach can be used. After detecting invalid or malicious traffic, the smart router sends a control message to stop or limit the flow of such traffic to simple routers which reside on the return path as shown below. This message is similar to the *STOP* message described in Section 3.7.1. The message is protected by PLA and therefore contains the sender's signature[1].

Smart router → Simple routers:
{STOP, PK_{SOURCE}, Timeframe}
Or
{LIMIT, PK_{SOURCE}, Timeframe, Amount}

Simple routers will verify that such a *STOP* or *LIMIT* message has arrived from a local smart router and will limit or block the flow described in a message accordingly. This is just one possible case of the control message format; traffic can also be limited or blocked based on an IP addresses, or the trusted third party that has authorized the source. Basically in this approach smart routers control the behaviour of the rest of the network by using PLA-protected control messages. Such a system would work regardless whether the attacker is located within the local network or not. The main advantage of this approach is the ability to stop or limit unwanted traffic as close to the source as possible, even when a majority of routers are not fully PLA-enabled smart routers.

Such a scheme can be attacked by sending a large amount of bogus *STOP* or *LIMIT* messages to simple routers. Since simple routers have limited resources for performing signature verifications, they would not be able to process authentic control messages coming from smart routers. To alleviate this problem, simple routers can be configured to accept control messages only from trusted sources. It is reasonable to assume that the amount of smart routers within the network is relatively small, and their IP addresses and identities are known by other routers within the network.

Overall, such a system offers better security compared to a plain network without PLA support and it requires only a low initial investment, since a vast majority of routers in the network can be slightly modified ordinary routers. The system would also work with ordinary, non-PLA-aware routers, in which case smart routers would use SNMP [22] or

1 Other mechanisms like a shared secret or hash message authentication code (HMAC) can also be used to authenticate control messages.

a similar protocol for traffic management. The security of the network can be improved in a straightforward way simply by adding more smart PLA-enabled routers.

Malicious nodes should be removed quickly from the network

TTP certificates with a limited validity time offer a way to remove malicious nodes from the network. If a node becomes dangerous to the network, i.e., it does not behave according to the rules, the TTP will not renew its certificate and without a valid TTP certificate, the traffic that the node sends will be blocked at the first router. In addition, routers can block public keys which are used to send large amounts of invalid packets, thus preventing malicious nodes from communicating further.

The validity time of short-time TTP certificates is basically a trade-off between security and overhead. A shorter validity time means better security since a potential attacker has a smaller window of opportunity to launch an attack, although it would also increase the overhead because certificates would need to be renewed frequently across the whole network. A longer validity time increases the window of opportunity to launch an attack, but it decreases the overhead of the certificate-renewal process. The effect of different TTP certificate validity times on security and PLA-related overhead is discussed in more detail in Chapter 10. The validity time could be changed on the fly depending on the situation, for example if the network is under constant attack, decreasing the validity time of TTP certificates would make sense.

A mission critical network, like a military network with a limited amount of nodes and plenty of available bandwidth, could use a centralized revocation scheme to instantly revoke certificates of compromised nodes or TTPs. In this case revocation messages can be sent through the one-way channel, such as a radio link or a satellite connection, that covers a large area.

Validation of packets

The PLA header contains all necessary information to perform validity checks on packets. The signature together with the public key allows detection of forged packets. The timestamp and the sequence number can be used to detect delayed or duplicated packets.

5.1.2 Important criterion

Scalability

The most crucial matter regarding the scalability of PLA is the amount of certificate-related traffic. Nodes and TTPs within the network must renew their certificates periodically and their certificates may be revoked, and the amount of this type of traffic should remain reasonable even with a very large number of nodes and TTPs. In the most extreme case, every TCP session could utilize different TTP certificate. The effect of this traffic is analysed in the next section in more detail under the topic of manageability. The performance of cryptographic operations also affects PLA's scalability and will be discussed in Section 5.3.

The TTP architecture presented in Section 3.5 allows TTPs to be added and removed in a flexible manner. In order to limit the overhead of TTP certificate verification traffic, two PLA layers should be used within Internet core networks as mentioned in Section 3.5.3. While using encapsulation and two PLA layers will produce higher bandwidth overhead in core networks, it will significantly decrease the amount of TTPs in which routers must trust. This reduces the overhead produced by TTP verification requests and will thus improve overall scalability. In addition, a router which encapsulates traffic into its own PLA header basically also takes responsibility for that traffic, meaning that routers should only encapsulate traffic in which they fully trust either directly, or through the chain of trust.

The scalability of PLA for small and portable devices depends on the power consumption produced by PLA and it is discussed below.

5.1.3 Optional criteria

Small power consumption and bandwidth overhead

The overhead caused by PLA can be divided into two main categories: a bandwidth overhead, which is caused by the addition of the PLA header to each packet, and a power consumption overhead, which is caused by public key cryptography operations like signing and verifying the packets. PLA's power consumption is analysed in Section 5.3.

Since PLA adds its own fixed-length header to every packet, the average size of sent packets determines the relative bandwidth overhead of PLA. To estimate an average bandwidth overhead of the PLA, network traffic traces from the Cooperative Association for Internet Data Analysis (CAIDA) [113] were used to determine an average packet size. Traces from year 2009 contained information about 5.6 million packets and an average packet size was about 581 bytes or 4650 bits. The size of the PLA header with all IP header extension fields and padding is 80 or 136 bytes, depending on whether the TTP certificate is included. If we assume that 10% of packets are sent with a TTP certificate, then the average overhead produced by PLA would be about 86 bytes per packet. This means that the PLA header adds roughly a 15% bandwidth overhead per PLA layer compared to plain IP traffic. The overhead would increase to 30% in cases where two layers of PLA are used as a result of encapsulation. If packets are fragmented during their journey, and the node performing the fragmentation adds an own PLA header to fragmented packets, the overhead would increase further. However, such overhead is not critical, considering that the Internet has a spare bandwidth available, especially in core networks [107] where encapsulation would be used. Figure 21 describes the bandwidth overhead of the PLA header with different packet sizes.

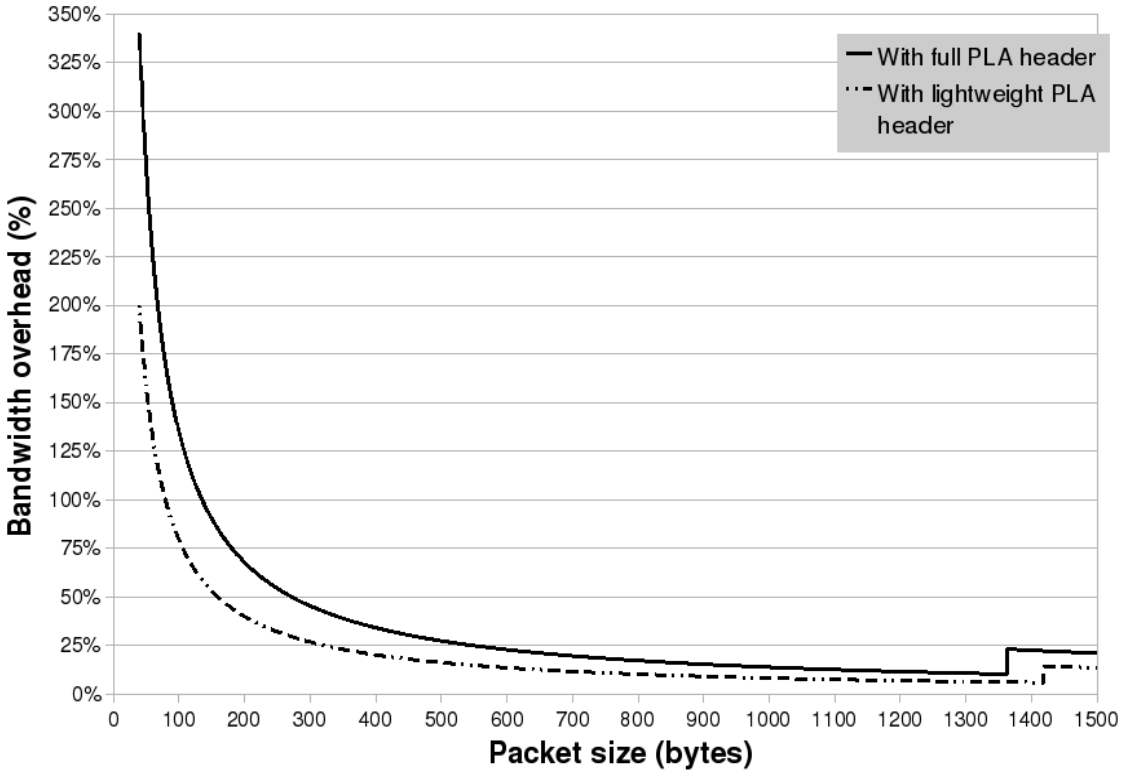


Figure 21. Relationship between the packet size and the relative PLA bandwidth overhead

Since PLA adds a fixed-length header, the relative overhead is larger with small packet sizes and decreases as the packet size increases. When the packet size is between 500 and 1000 bytes, the relative overhead is roughly 13-27% for the full PLA header and 8-16% for the lightweight PLA header. An increase in the relative bandwidth overhead occurs with packet sizes of about 1400 bytes and above since adding a PLA header increases the packet's size above the commonly used maximum transmission unit of 1500 bytes. Therefore, the packet needs to be fragmented and additional PLA and IP headers are needed. The figure assumes that the packet contains only a standard IPv6 header of 40 bytes in addition to the PLA header; if other headers are also present, the overhead produced by fragmentation would slightly increase.

To further reduce a bandwidth overhead caused by PLA, a method similar to a TCP header compression [55] can be used. In the TCP header compression mechanism communicating parties store constant fields of the TCP and IP headers in their cache and transmit only fields that change, saving bandwidth. The same principle can be used to omit the sender's public key from the lightweight PLA header and transmit only timestamp, sequence number and signature fields in each packet. This would save roughly 20 bytes of header space in comparison to the lightweight PLA header.

In order to maintain compatibility with the Internet the compressed header should be reconstructed at the first non-bandwidth constrained node, such as a base station. For example, a mobile node with limited bandwidth would send the first packet with a full PLA header. Afterwards, it would calculate the signature over the full PLA header, but would only transmit timestamp, sequence number and signature fields in the PLA header. The base station would take the static PLA header information from its cache, reconstruct the header, and then forward the packet with a full PLA header. Since the original signature was calculated over the full PLA header, nodes in the rest of the network can verify these packets normally. The same principle would also work for receiving data. In that case the base station would perform the header compression and a mobile node would reconstruct the header based on cached information.

Free of patents

Various implementation of ECC algorithms are covered under several patents. Most ECC-related patents are owned by the Canadian company Certicom, which claims to have over 300 issued and pending patents related to ECC and public key cryptography.

Most ECC-related patents are set to expire by the year 2020, thus, for now, PLA implementation uses algorithms that are patended and does not satisfy this requirement. While PLA is not dependent on the cryptographic solution used, ECC is currently the only feasible solution for PLA because ECC can achieve a strong security with relatively small key and signature sizes. Therefore, it would be necessary to license ECC-related patents for a commercial PLA implementation, which could hamper the PLA's deployment. However, the existence of patents does not necessarily pose a significant threat to the wide scale deployment of PLA. There exist several system such as modern mobile phones and cellular networks that extensively utilize patented technologies.

Overall, PLA satisfies its design requirements well. Mandatory criteria are fully satisfied. The scalability and low power overhead requirements are satisfied quite well and these can be further improved in the future. The free-of-patents requirement is the only requirement that cannot be satisfied until ECC-related patents expire.

5.2 Redesigning security of the Internet criteria analysis

This section contains an analysis of how PLA satisfies the requirements for the secure next-generation Internet that were presented in Section 2.5. Some of these requirements are equivalent to those presented in the previous section and thus they will not be covered here again.

Only valid packets are transmitted in the network

PLA fully satisfies this requirement, every node that forwards the packet is capable of checking the integrity of the packet using the information contained in the PLA header. Only unique, authentic packets will be sent forward. Delayed, forged, or duplicated packets are discarded immediately.

Every packet has an owner and all packets originate from trusted entities

PLA satisfies this requirement by including information for calculating a public key of the sender in every packet and by using trusted third parties. Since every packet is cryptographically signed by the sender of the packet and contains the sender's public key, the sender is not able to deny sending the packet. Thus, every sent packet in the network has an owner which can be traced. The aim of the trusted third parties is to

guarantee that the sender is really a valid and trusted entity in the network. Therefore, untrusted entities will not be able to send data through the network.

Prioritizing traffic

PLA allows traffic to be divided into different groups which can be prioritized as necessary. These traffic groups are: valid data traffic from valid users, priority traffic from valid users, valid data traffic from unverified users (the TTP which has authorized the user has not been fully verified yet), and valid signalling traffic. A router could reserve a small amount of bandwidth for the latter two categories and allocate a majority of available bandwidth for valid traffic originating from valid users. Such a mechanism is very flexible, and would enable the network to prioritize traffic more effectively. In case of a serious emergency when only a small amount of bandwidth is available, routers could delay or even discard low-priority packets.

Using this principle, PLA could be used to implement a more secured version of Virtual LAN (VLAN) [50] technology. The main aim of VLAN is to make network management easier by allowing LANs in different physical locations to be grouped into virtual LANs. VLAN adds a separate tag to the Ethernet frame containing a VLAN identifier and a priority level of the frame. The downside of the current VLAN solution is poor security; any router can change the value of the VLAN tag within the Ethernet frame, potentially disrupting the network. Since the VLAN tag is included in the Ethernet frame, network-layer security solutions like IPSec cannot be used to protect it.

In the scope of VLAN, PLA could help as follows. Each virtual LAN would be managed by a separate TTP and the TTP's public key would act as a VLAN identifier. Since the TTP's public key and locator are included in a full PLA header, a separate VLAN identifier would not be necessary. Priority information of the frame could be expressed using the rights field of the TTP certificate. Basically, all VLAN-related information would be included in a PLA header, and thus it would be protected against forgery by the packet's signature. Such a mechanism would also prevent unauthorized access to the VLAN network since all users wishing to use the network would need to receive a valid TTP certificate from the TTP that manages the VLAN.

Manageability

The TTP certificate mechanism presented in Section 3.5 allows an efficient management of a large number of nodes. The TTP certificate includes a validity time and several levels of rights and it is also possible to specify which rights can be delegated forward. Section 3.8 presents a mechanism for bootstrapping and managing various devices, including devices which lack traditional input methods.

Since users will continuously renew their TTP certificates, they will use the TTP's bandwidth and computational resources. However, the load produced on the TTP is not significant, especially since the TTP can use the same PLA's hardware accelerator for cryptographic operations. Let us assume that a major TTP would have around one million customers. If each customer requests a new TTP certificate every hour, then there would be roughly 278 certificate requests per second, which is not a high amount of requests to handle for a large server. The bandwidth overhead would not be significant either. A certificate request and reply would each require a single packet, consuming roughly 3000 bits of bandwidth in total (2000 bits for two PLA headers and 1000 bits for other information including a new TTP certificate). Thus, 278 requests per second would consume less than 1 Mbps of bandwidth.

Even if users will use a different TTP certificate for each TCP session, and renew their certificates every few seconds generating 1 Gbps of traffic related to certificate requests, the computational load caused by cryptographic operation will still be minor. In this case there will be about 278,000 certificate requests per seconds, and this amount of cryptographic operations can be easily handled by a dedicated cryptographic accelerator.

Controlling incoming connections

PLA together with traditional certificates can be used to control incoming connections. The recipient of a connection could grant certificates to trusted initiators allowing them to make incoming connections. PLA is used to enforce that only the data from valid initiators is allowed to be transmitted to the recipient. Section 6.1 contains a more detailed discussion about using PLA to control incoming connections.

Privacy protection

Including information about the sender's public key together with an identity number given by a TTP in every packet naturally produces a privacy risk, since the sender of the

packet can be easily tracked in the whole network using a unique public key and identity received from a TTP. However, this problem can be solved by using multiple public keys for each user which will act as pseudonyms. For example, the user can generate a large amount of public keys and request short-time TTP certificates for these keys from the TTP using his main public key which is already authorized by the TTP. After the user makes a connection using his new public key, he will be practically anonymous to the rest of the network. The user's TTP will be the only entity in the network that will know the mapping between the user's real identity and all of his public keys. Thus, privacy will not be a problem as long as the user changes his public key frequently and as long as the above-mentioned TTP certificate retrieval processes is protected by encryption. Such a process could also be automated in software to make it easier to use for the user. The user's key management software could be configured to periodically generate a new public key, request a TTP certificate for it, and use it for future traffic without any user interaction.

Another option to enhance privacy and provide flexibility is to utilize several public keys simultaneously for performing different tasks. For example, the user could use one public key to read work-related e-mail, while using another for instant messaging with his friends. Based on the desired level of privacy the user can decide how many key pairs to use and how often to change keys. Thus, with the help of PLA, it is possible to produce a good trade-off between the security and privacy. If the user engages in illegal behaviour, the authorities can determine the real identity of the user by contacting the TTP that has authorized his public keys. Otherwise, the user will be able to maintain reasonable privacy[1].

One problem with privacy in the current Internet is the lack of trust during anonymous communication. It is hard to communicate effectively while retaining complete anonymity because other parties on the Internet do not have the means to verify effectively whether that anonymous person can be trusted or whether the data is really coming from the same anonymous person or from someone else that impersonates him. PLA provides help to this issue because it allows for users to have several pseudonyms that can be validated by other parties. For example, suppose that A and B want to communicate with each other. A wants to retain his anonymity, thus he discloses one of

1 It is important to note that a network layer solution like PLA cannot offer a complete privacy solution by itself. Higher layer protocols and applications may leak information about the user. Therefore, also other methods must be used together with PLA to provide a complete privacy.

his numerous public keys together with the associated TTP certificate to B without disclosing any personal details. Now B can contact A's TTP and verify that A is really a valid entity and his TTP certificate is still valid. As a result, B can trust A without knowing A's real identity. Since A has numerous public keys in use, he can allocate one for communication with B and A will not need to change that key frequently. In addition, by checking the signatures of received packets, B can be sure that those packets are coming from A and not from some other party that impersonates A.

Compatibility with future data-oriented networks

Since PLA utilizes cryptographic signatures and identities extensively, it is a natural solution for securing data-oriented networks.

Most data-oriented networking approaches rely on self-certifying identifiers, meaning that the receiver of the data item should be able to verify its authenticity and integrity. PLA accomplishes the same functionality on the packet level through cryptographic signatures. Similarly to the idea of controlling incoming connections, data-oriented networks aim to prevent unwanted traffic by transferring control to the receiver. Using PLA to secure data-oriented networks is further discussed in Chapter 8.

5.3 Performance and energy consumption of PLA

This section discusses the performance and power consumption of PLA's cryptographic operations in various situations. The performance should be high enough in order for PLA to scale to Internet core networks. While the power consumption is especially important in wireless environments, it should also be at reasonable levels in core networks.

5.3.1 Performance of cryptographic operations

In stand-alone simulations, the FPGA-based cryptographic accelerator achieved 166,000 full signature verifications per second [61]. This corresponds with verifying 2 Gbps of traffic if maximum sized 1500-byte packets are used. With a minimum packet size of about 1000 bits (a minimal IPv6 packet with a PLA header), such an accelerator could handle 166 Mbps of traffic. According to Altera's PowerPlay simulation tool, the power consumption for this FPGA implementation would be about 18.6 W under load.

Therefore, the energy consumption per validated packet is about 112 µJ. It is important to note that these results were achieved with a programmable FPGA, which achieves significantly lower performance compared to a dedicated application specific integrated circuit (ASIC) made on the same manufacturing process. In addition, the FPGA used in testing was released in 2005 and is manufactured on a relatively old 90 nm process and is thus not the fastest FPGA available on the market today.

Altera offers a Hardcopy structured ASIC technology [4] which allows an existing FPGA design to be easily converted into a structured ASIC. Such a structured ASIC would achieve a significantly higher performance and lower power consumption compared to a programmable FPGA. However, the structured ASIC would still not be as an efficient solution as a fully customized ASIC. Based on Altera's simulation tools, converting the existing FPGA PLA accelerator design to a Hardcopy II HC240F1508I 90nm structured ASIC would result in a performance of 850,000 verifications per seconds with an estimated power consumption of 22.4 W [41]. The Hardcopy ASIC contains four times as many computational units and runs at the higher clock speed of 210 MHz, therefore the latency of a single verification would decrease to about 89 µs. Hence, such an ASIC could verify 0.85 - 10 Gbps of traffic depending on the packet size, while the energy consumed per packet would decrease to about 26 µJ.

These results are based on the power consumption under a full load and due to idle power consumption of ASICs, the power consumption of a single cryptographic verification may be higher under a low load. However, since the cryptographic accelerator is based on multiple parallel computational blocks, it is relatively easy to lower the idle power consumption by turning off unnecessary computational units.

Currently, the most modern manufacturing process available is the 32 nm process which is three generations ahead of the 90 nm process used in the current FPGA implementation. The transistor density of an integrated circuit roughly doubles with each new process generation while the supply voltage of the circuit decreases, which reduces overall power consumption. Thus, due to a more advanced manufacturing process and because a customized ASIC achieves better performance and lower power consumption than a structured ASIC, it is reasonable to estimate that a customized ASIC solution manufactured on a 32 nm process would be able to verify tens of gigabits

of average traffic with a power consumption of less than a couple of watts per gigabit of traffic[1].

Another important PLA-related performance issue is the additional latency produced by PLA. This latency can be divided into three main parts: the latency of cryptographic computations, the latency produced by communication between the main processor and cryptographic accelerator, and the latency added by non-cryptographic tasks such as checking the timestamp and the sequence number within a packet. The latency of communication between the network processor and the cryptographic accelerator would not be significant in an optimized case, since in a dedicated PLA-enabled router, the hardware accelerator would reside either on a PCB connected directly to the main network processor by a fast, low latency bus, or inside the network processor itself. Thus, when measuring the latency impact of the PLA in an optimized case, the latency of actual cryptographic operations is the most crucial. This latency amounts to 90 μs per verification using a structured Hardcopy ASIC and this latency depends linearly on the clock speed of the circuit. Thus, in an optimized case this latency would be decreased further.

Overall, the latency impact of PLA is not significant. If we assume that a typical connection uses about 12 hops and the total latency impact of packet verification in an optimized case is around 100 μs per router (70 μs for cryptographic operations and 30 μs for other operations including communication overhead), then this would produce only a 1.2 ms extra delay in one direction and a 2.4 ms increase in round-trip time.

5.3.2 Improving the efficiency of PLA

One way to further increase the efficiency of the PLA implementation at high network speeds is to use jumbo or super jumbo frames, which have a frame size of 9000 or even 64,000 bytes instead of a normal frame size of 1500 bytes. Therefore, inside high speed networks, several PLA packets could be encapsulated into a single jumbo frame, significantly decreasing the amount of verifications needed per unit of traffic. For example, if six packets are encapsulated into a single jumbo frame, then the amount of

1 The current FPGA design is optimized for a specific elliptic curve type and length, this is feasible since the FPGA design can be easily updated. A customized ASIC should include support for several ECC related parameters which would slightly decrease its efficiency, still it would be far more efficient than an FPGA solution.

PLA verifications would drop to one-sixth of the original. Using jumbo frames would provide three main benefits: less computational resources would be required for packet verification, power consumption would also be decreased, and bandwidth overhead produced by encapsulation would be lower, since an additional PLA header would use only a very small part of a jumbo frame payload.

Depending on the security policy used, routers do not necessarily need to check every passing packet. In a normal situation, it could be sufficient to check every tenth or every twentieth packet randomly without significantly reducing the security of the network. If the attacker floods the network with a large amount of invalid packets, routers would catch at least some of the invalid packets sent by the attacker even if they do not check every packet that they forward. In addition, connections usually go through more than ten routers, and thus it is probable that every packet will still be checked at least once as it travels through the network.

According to the International Technology Roadmap for Semiconductors (ITRS) [53] the performance of semiconductors will continue to exponentially increase in the near future while costs will continue to decrease. On average between 2009 and 2024, the amount of transistors per price unit will increase by 41% per year while the clock speed of circuits will increase by 8% per year. As a result, the overall performance at a given price point will increase by about 52% per year, which is in line with the growth of the Internet's traffic. These results shows that per-packet cryptographic signatures used by PLA are a viable solution also in the future. Furthermore larger MTU sizes are likely to be adopted in the future, which would further increase PLA's efficiency for the reasons mentioned above.

Overall, PLA is a very scalable solution which can be used in very high bandwidth networks as long as dedicated hardware is used for accelerating cryptographic operations.

5.3.3 Energy consumption in core networks

High-speed core network routers have plenty of available electrical power. However the power consumption is still important, since there is a limit to how much power the

router can consume with reasonable cooling. Preferably, PLA should not significantly increase the power consumption of the current routers.

Table 2 below presents a summary of the performance and power consumption of a proof-of-concept hardware acceleration in core networks. For comparison, the table also includes the Cisco CRS-1 core network router [25], which supports sixteen full-duplex 40 Gbps interfaces offering 640 Gbps of bandwidth in one direction and 1280 Gbps of bandwidth in total.

The average packet size is assumed to be 4650 bits for normal packets and 70,000 bits when jumbo frames are used. While some traffic on the Internet uses significantly smaller packets, the average packet size is the most important for the performance of PLA, since it determines the average load on routers. It is important to note that these results show the performance of the Hardcopy ASIC manufactured on the old 90 nm process. An optimized ASIC on a modern 32 nm manufacturing process would achieve significantly higher energy efficiency.

Table 2. Performance and power consumption of cryptographic operations in core networks

Hardware	***Verifications per second***	***Throughput***	***Power consumption***	***Power/ throughput***
90 nm FPGA	166,000	0.77 Gbps	18.6 W	24.2 W/Gbps
90 nm Hardcopy ASIC	850,000	3.95 Gbps	22.4 W	5.67 W/Gbps
90 nm Hardcopy ASIC, 10% of packets verified	850,000	39.5 Gbps	22.4 W	0.57 W/Gbps
90 nm Hardcopy ASIC with jumbo frames	850,000	59.5 Gbps	22.4 W	0.38 W/Gbps
90 nm Hardcopy ASIC with jumbo frames, 10% of packets verified	850,000	595 Gbps	22.4 W	0.04 W/Gbps
Cisco CRS-1 Carrier Routing System with 16 slots [25]		1280 Gbps	9,630 W	7.52 W/Gbps

Even with the 90 nm Hardcopy ASIC, the power consumption of PLA's cryptographic verifications is lower than the power consumption of current core routers. If jumbo frames are used, or only 10% of packets are fully verified, then adding PLA support to the core network router would increase the total power consumption by about 5-7%. Therefore PLA does not consume excessive amounts of power and is scalable for core networks.

In this case the power efficiency is actually more important than the raw performance, since as long as the power efficiency is good, the performance can easily be increased by using larger ASICs or multiple ASICs in parallel. These results show that a 90 nm Hardcopy ASIC provides enough performance for the currently used 40 Gbps network interface as long as jumbo frames are used, or only 10% of packets are verified. With a more optimized ASIC built on a modern 32 nm manufacturing process, the performance should easily scale to 100 Gbps interfaces.

Next we investigate how much energy PLA consumes on the Internet's scale. According to Cisco the total IP traffic will amount to 21,367 PB per month in 2010 [26], this figure also includes non-Internet traffic and traffic generated by mobile devices. With an average packet size of 4650 bits, this equals to about 14,182 million packets sent every second. Based on the energy consumption figures mentioned in Section 5.3.1, signing this amount of traffic with a 90 nm Hardcopy ASIC would use only 142 kW of power. If we assume that an average flow on the Internet goes through 12 hops, then the total power consumption of using PLA for signing and verifying every IP packet in the world would be about 4567 kW, which is the power output of a large wind turbine. On a yearly basis this would imply the energy consumption of about 40 GWh, which is an insignificant amount on a worldwide scale.

5.3.4 Energy consumption in wireless networks

Before PLA's energy overhead in wireless networks can be discussed it is important to determine energy consumption of the wireless transmission itself. This is not an easy task. Several power-efficient transmission methods for IEEE 802.11a/h wireless LANs were analysed in [89] and the best method achieved a power efficiency of approximately 70 Mbits of traffic per Joule of consumed energy over a transmission range of 10 meters using the star network topology. This is equivalent to a power consumption of about 14.3 nJ/bit. As the transmission distance was increased to 25 meters, the efficiency dropped to about 9 Mbit of traffic per Joule corresponding to 111 nJ/bit. However, this result assumes that the networking hardware utilizes a special power conservation scheme, and does not take into account interference from weather or buildings.

In a simulated study of IEEE 802.11 power-saving mechanisms [57], 50 nodes were placed in a 1000 x 1000 meter area. The best case energy consumption was about 2.22

μJ/bit. According to [120], power consumption for a GSM cell phone is about 13.8 μJ/bit for the upload and 5.6 μJ/bit for the download. For wireless LAN these numbers are 1.1 μJ/bit and 0.75 μJ/bit respectively.

As a comparison, the verification power consumption of the Hardcopy ASIC PLA cryptographic accelerator would be significantly lower, only around 26 μJ per packet, or 5.6nJ/bit if average 4650-bit packets are used. The power consumption of a customized ASIC manufactured on a modern manufacturing process would be even lower. Therefore, the power consumption of PLA's cryptographic operations is much lower than even theoretical results for the wireless LAN power consumption. It is also important to note that signature generation uses less computational power, while in the wireless networks the opposite holds true; sending packets consumes more power than receiving it. Therefore, for a wireless sending or forwarding node the energy overhead of cryptographic operations is even less significant.

The power consumption of cryptographic operations is also insignificant when compared to the battery capacities of ordinary cell phones. A standard cell phone battery rated for 3.7 V operating voltage and 1000 mAh capacity contains about 13,320 Joules of energy. Even with the power efficiency of a cryptographic accelerator based on the Hardcopy ASIC, this amount of energy would be enough to validate about 512 million packets, which would correspond to 298 Gigabytes of average-sized traffic. With a 1 Mbps network connection it would take almost one month of continuous usage to receive such an amount of data.

The real-life power consumption of a Nokia N810 Internet tablet was studied in [121]. N810 consumes 0.509 W of power when using power-saving mechanisms and downloading data through a wireless LAN interface at the rate of 16 kB/s. When the download rate is increased to 256 kB/s, the power consumption grows to 1.057 W. The rate of 256 kB/s corresponds to about 451 4650-bit packets per second, verifying signatures of these packets with Hardcopy ASIC would consume less than 12 mW of power, increasing the total power consumption only by about 1%. Actually, the major part of PLA's related energy overhead in wireless networks comes from the fact that PLA produces some bandwidth overhead, and this extra data must be transmitted wirelessly, consuming some extra energy. For example, with a 4650-bit average packet

size and full PLA header, PLA's bandwidth overhead is about 23%, and therefore the total energy overhead with N810 is about 24%.

Even thought PLA produces some power and bandwidth overhead, it may achieve better power efficiency in mobile networks in certain cases. For example, mobile ad-hoc networks are very vulnerable to various attacks such as DoS or route spoofing. A successful attack can quickly drain the whole ad-hoc network of its resources like a battery power or bandwidth. Thus, if the ad-hoc network is frequently attacked, PLA can actually save power in the network by limiting attacks quickly, before they can bring the whole network down by consuming all its available energy.

According to these results, PLA is an energy-efficient solution, which scales from small mobile devices to Internet core routers. The next section evaluates the power efficiency of PLA in wireless networks with hash chain and hash tree-based security solutions.

5.4 Comparison with hash tree and hash chain-based solutions

Hash tree and hash chain-based security methods are comparable to PLA, since they offer hop-by-hop integrity protection on the network level. This section compares the security properties of these approaches and also includes an analytical analysis of the power consumption in a wireless environment. This comparison assumes that every packet is verified at every hop, since it is the worst case situation for PLA. More detailed discussion is available in [71].

5.4.1 Hash trees

The hash tree is a tree where leaves are hashes of data blocks, like packets. Nodes higher in a tree are hashes of their respective children. A Merkle tree [78] is a complete binary hash tree.

Hash trees can be used to protect the integrity of the network traffic [116], [73]. The idea is to create a Merkle tree which consists of packets' hashes, calculate the root hash of the tree, and sign it with a cryptographic signature. Then packets are sent along with the tree's root signature and the necessary amount of tree hashes to reconstruct the root

hash. The verifier of the packet can reconstruct the root hash and verify the root signature.

As an advantage, using hash trees reduces the amount of required cryptographic signature verifications to verify the integrity of the packet. This saves computational resources, since hash calculations are significantly faster to perform than signature verifications. In order to give the ability to independently verify the integrity of a single received packet, additional hashes along with a root signature must be included into every sent packet. A hash tree with width w requires $log_2(w)$ additional hashes to be included in sent packets.

5.4.2 Hash chain-based solutions

A hash chain is formed by hashing a random seed value s multiple times with a one-way hash function H. The result of the hash operation is used as an input for the next element of hash chain, i.e., $h_1 = H(s)$, $h_2 = H(h_1) = H(H(s))$, and so on. The final element of the hash chain is the anchor. The hash chain is used in the reverse order to create signatures, if the length of the chain is n, then the anchor, h_n, is used first, followed by h_{n-1} and h_{n-2}.

By tying identities to hash chains, hash chains can be used to authenticate nodes and traffic in the network [48]. Security properties of hash chains come from the fact that the hash function H is a one-way function, therefore it is not possible to determine x if only $H(x)$ is known. Therefore, only the owner of the hash chain can know previous values of the chain, while any other party can easily verify that $h_n = H(h_{n-1})$ after the value h_{n-1} has been disclosed. There exist three different secure ways to distribute values of the hash chain to other parties: one-time signatures, time-based approaches and interaction-based approaches.

An example of a time-based hash-chain approach is Timed Efficient Stream Loss-Tolerant Authentication (TESLA) [88], which is a protocol designed to authenticate broadcast and multicast traffic. TESLA assumes that sender and receiver have loosely synchronized clocks.

Adaptive and Lightweight Protocol for Hop-by-hop Authentication (ALPHA) [49] is an interaction-based hash chain scheme. ALPHA uses a three-way signature process where

both the sender and receiver use two hash chains, it also allows any node on the path to verify and authenticate the protected traffic. First, the sender sends an *S1* packet containing its hash chain anchor and a message authentication code (MAC) calculated over the payload with a previous hash chain value. The receiver replies with an *A1* packet indicating that he is willing to receive the packet. Finally, the sender sends the actual data together with a previous hash chain value. Intermediate nodes can verify the payload's integrity using the MAC from the cached *S1* packet. Therefore ALPHA requires two control packets to be transmitted for each data packet, to reduce such bandwidth overhead there exist two variants of ALPHA. ALPHA-C transmits multiple MACs in a single *S1* packet, while ALPHA-M constructs a hash tree over *w* packets and the whole tree is authenticated with a single *S1* packet.

5.4.3 Security Properties of PLA, hash tree and hash chain-based approaches

In this section we evaluate the security properties of PLA, the lightweight PLA that does not contain a TTP certificate in the header, the basic hash tree approach described in Section 5.4.1, ALPHA-C, and ALPHA-M. Another authentication scheme, TESLA, was omitted from the comparison since it depends on the synchronized clocks and is a less flexible solution, while the basic ALPHA scheme would produce a high bandwidth overhead. The results of evaluation based on various criteria are summarized in Table 3. The width of the hash tree and the amount of MACs sent simultaneously in ALPHA-C is denoted by *w*. By authentication we do not mean simply tying the packet to a specific cryptographic identity, but determining whether the cryptographic identity is a trusted one and has permissions to use the network. For measuring the bandwidth overhead we assume that a 163-bit ECC public key and 160-bit hash function are used for adequate security [11]. With a compression bit, the public key uses 164 bits of the header space.

The first criterion (C1) is the independent transmission of packets. This is especially important for a real-time communication, like a video or voice conferencing, where a low latency is preferred. PLA allows each packet to be sent immediately, while other approaches utilize hash trees or cumulative transmissions of MACs, which require that the sender must cache *w* packets before sending the first one.

The C2 criterion evaluates the support for the fully independent integrity verification in intermediate nodes. PLA and a basic hash tree approach allow every node to independently verify each packet, since all necessary information is already present in every sent packet. However, a basic hash tree approach must cache intermediate hashes and already verified root signatures for an optimal performance. ALPHA-M and -C do not support this criterion and require states for integrity verification.

With the C3 criterion we evaluate whether the solution supports fully independent sender authentication. The results are the same as above, except that lightweight PLA does not satisfy this criterion since it does not include a TTP certificate in every packet.

Table 3: Evaluation of security mechanisms

Criteria	***Lightweight PLA***	***PLA***	***Basic hash trees***	***ALPHA-M***	***ALPHA-C***
C1. Independent transmission of packets.	Yes	Yes	No	No	No
C2. Independent integrity verification	Yes	Yes	Yes, but needs caching for best performance	No	No
C3. Independent authentication	No	Yes	Yes, but needs caching for best performance	No	No
C4. Transit path independence	Yes	Yes	Yes, at a lower performance	No	No
C5. Bandwidth overhead per data packet	Lightweight PLA header (586 bits)	Full PLA header (1024 bits)	Public key (164 bits) + root signature (326 bits) + $log_2(w)$ hashes	1 + $log_2(w)$ hashes	1 hash (160 bits)
C6. Other bandwidth overhead	None	None	None	4/w hashes	(w + 3)/w hashes
C7. Per packet computational requirements	Signature verification	Key extraction and signature verification	At most 1 + $log_2(w)$ hash calculations, 1/w signature verifications	At most 1 + $log_2(w)$ hash calculations	1 hash calculation
C8. Load on verifying node	Constant	Constant	Variable	Constant	Constant

The C4 criterion evaluates whether packets can travel along multiple transit paths between the sender and receiver. PLA allows every node to independently verify each packet, regardless which path the packet takes. A basic hash tree-based approach also supports independent verification, but the performance will degrade if packets belonging to the same hash tree will take multiple paths. ALPHA schemes require intermediate

nodes to cache *S1* packets and therefore all further data packets must travel along the same path in the network.

The C5 criterion is about bandwidth overhead per data packet. A simple PLA header contains the sender's public key, signature, timestamp, and a sequence number. In addition to the above, a full PLA header contains trusted third party (TTP) certificate. In order to support independent verifications using a basic hash tree approach, $log_2(w)$ hashes must be included in packets in addition to the public key and the root signature. ALPHA-M requires the undisclosed hash chain element to be present in data packets in addition to $log_2(w)$ hashes. For ALPHA-C including the undisclosed hash element is enough. In all cases some extra fields are necessary for IP extension headers, these are not included in comparison.

The next criterion (C6) contains other bandwidth requirements. Both ALPHA schemes require two control packets (*S1* and *A1*) to be transmitted per w packets. In ALPHA-M w denotes the width of the hash tree while with ALPHA-C it is the amount of MACs transmitted in a single *S1* packet. ALPHA-C transmits a MAC for every packet, while ALPHA-M uses a single MAC for the whole tree.

The criterion C7 is per-packet computational requirements, excluding minor checks like verification of a timestamp or sequence number. Lightweight PLA requires a signature verification to be performed. Full PLA utilizes implicit certificates, where the sender's public key is extracted and the packet's signature is verified in a single operation, which also verifies the TTP certificate. Basic hash tree and ALPHA-M approaches require $1 + log_2(w)$ hash calculations if the received packet is the first one from the tree. Intermediate hashes can be cached and therefore verification of subsequent packets will require less hash calculations. Additionally, the basic hash tree approach requires a single signature verification per hash tree. ALPHA-C does not utilize hash trees, therefore only a single hash calculation needs to be performed.

The last criterion (C8) evaluates whether the load on verifying nodes is constant or variable. For PLA and ALPHA approaches, the load is constant; intermediate nodes must perform a signature verification or hash calculations per each packet respectively. In the basic hash tree approach the signature must be verified for each packet from a

previously unknown hash tree. Therefore the verifier's load may differ significantly depending on the order in which packets arrive.

This drawback introduces a significant denial of service vulnerability. The attacker can simply flood packets belonging to different hash trees, since the verifying node will not have the packets' hashes in its cache, it must must calculate the whole hash tree and perform the root signature verification for each malicious packet. As a result, the verifying node will be overloaded, since it does not have the capacity to verify the signature of each packet[1].

The main difference between PLA and hash chain-based approaches is that PLA supports fully independent verifications of packets by any node in the network: a node which has received the packet can verify its authenticity independently without having any kind of contact with the sender of the packet. This feature is especially useful in ad-hoc networks. In addition, PLA does not require nodes to store per-packet or per-sender states for the basic authentication. While ALPHA supports hop-by-hop authentication, it requires that some data is cached by intermediate nodes and all packets take the same path in the network. A basic hash tree approach supports independent verifications of packets, but its performance will degrade unless all packets take the same path in the network and verifying nodes cache intermediate hashes. Basically all solutions are trade-offs between usage of bandwidth, computational resources, features and implementation complexity.

ALPHA is a less robust solution in a case of route changes or failures. If the route changes after the first *S1* packet, the rest of the packets from the same hash tree or cumulative transmission cannot be verified unless the *S1* packet is retransmitted. Since intermediate route changes are usually not visible to the recipient or the sender of the traffic, further data packets in the session will probably be dropped by intermediate nodes located on the new path.

Using large values of w with ALPHA or a hash tree approach increases the latency of communication and therefore is not suitable for real-time traffic. For example, VoIP applications send usually from 34 to 50 packets per second [27]. Therefore, using a 16

1 This is the fundamental assumption of hash tree based approaches.

packets wide hash tree, or sending 16 MACs simultaneously with ALPHA-C, would introduce up to 470 ms of extra latency in one direction.

In order to support the strong source authentication with ALPHA, the hash chain anchors used must be authenticated by some other means, e.g., public key signatures. ALPHA requires different hash chains to be used for every sender/receiver pair, therefore such authentication can introduce a significant overhead if the receiver is communicating with a large number of senders.

5.4.4 Energy consumption in a wireless environment

This section investigates energy requirements of various security solutions for wireless communication. In order to make the analysis thorough, energy costs for wireless transmission, reception and forwarding are evaluated separately. For wireless LAN power consumption, the absolute best case values from Section 5.3.4 are used; 14.3 nJ/bit for a 10-meter transmission distance and 111 nJ/bit for a 25-meter distance [89]. We also include higher wireless LAN power consumption values that are much more realistic [120], marked as "real-life WLAN" in the figures. PLA is assumed to use a cryptographic accelerator based on a 90 nm Hardcopy ASIC technology, which consumes 26 μJ of energy per signature verification and 10 μJ of energy per signature generation [41].

Since we do not have a real ASIC available for cryptographic calculations, the following results are calculated analytically. We calculate the bandwidth overhead of various security approaches and compare the energy cost of transmitting such extra traffic versus the cost of performing cryptographic operations. The energy cost of hash calculations was ignored since it is not significant compared to the cost of wireless transmission or signature verifications. The per-packet energy advantage of various approaches in comparison to PLA can be summarized using the following formula.

$$E_{total} = B * E_{wireless} + E_{crypto} . \quad (1)$$

Where B is an average per-packet bandwidth advantage in bits against PLA (negative values denote bandwidth disadvantage), $E_{wireless}$ is the energy cost of wireless reception per bit, and E_{crypto} is the energy cost of the signature verification.

5.4.5 PLA versus hash tree approach

First we compare PLA with a basic hash tree approach for integrity protection, since both techniques support fully independent packet verification by any node. In this respect, they offer identical security properties.

In the hash tree approach a single signature is generated and verified for the whole hash tree while with PLA every packet's signature is verified separately. In order for both schemes to offer the same level of protection and thus to make comparison fair, the PLA header information, like a sequence number, timestamp, and TTP certificate, must be included in every packet in the hash tree approach. In this respect, the bandwidth overhead is the same for both of these two approaches. The only difference is the size of extra hashes that must be included in the hash tree approach to make independent verifications of packets possible.

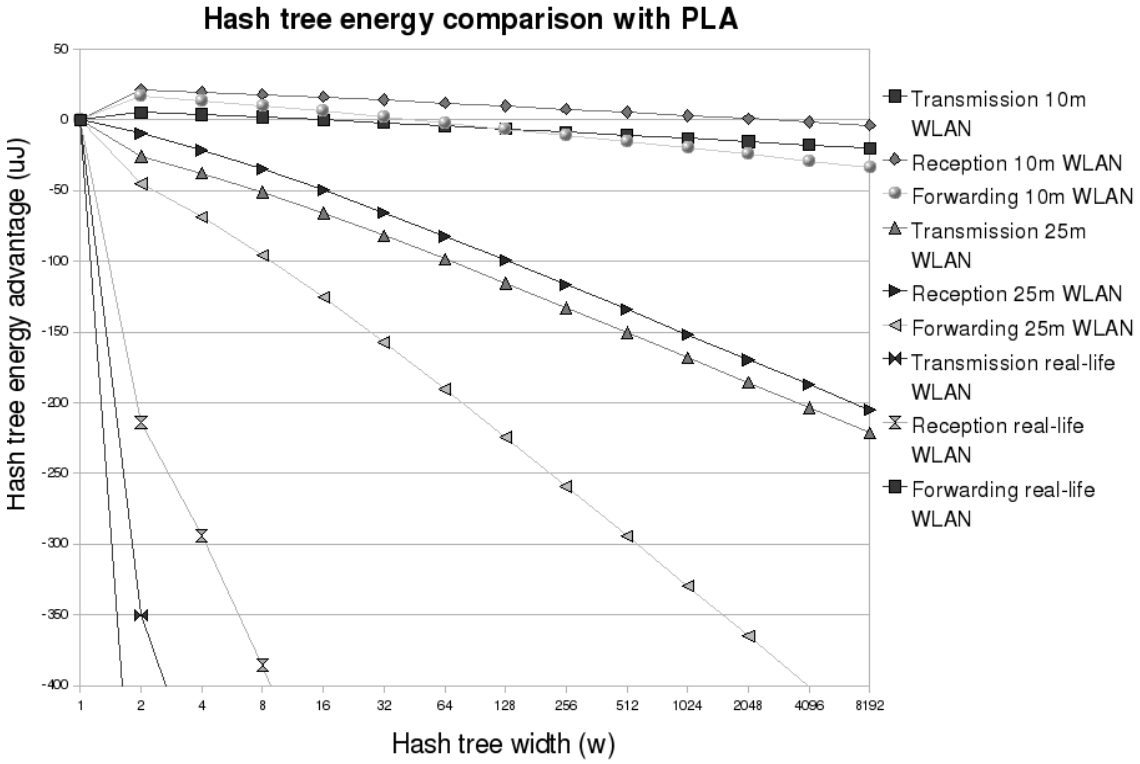


Figure 22: Per packet energy advantage of hash trees versus PLA in a wireless LAN

The results are shown in Figure 22, when the hash tree width $w = 1$ hash trees are not used and the signature verification is performed for each packet. Y-axis values denote per-packet energy advantage, positive values mean that a hash tree approach has an energy advantage over PLA, while negative values describe the reverse situation.

The following observations can be made from the figure, if the cost of wireless transmission is very low then hash trees save energy in some cases. However, as the width of the hash tree grows to thousands of packets, the total energy consumption actually increases in comparison to PLA, since large hash trees have a significant bandwidth disadvantage due to inclusion of extra hashes. As the cost of wireless transmission increases, even using narrow hash trees increases the total energy consumption. If real-life estimates for the power consumption of wireless LAN are used, the total energy consumption becomes completely dominated by the cost of wireless communication. As a result, the hash tree approach becomes very energy inefficient due to their large bandwidth overhead.

Due to the nature of elliptic curve cryptography, the signature generation requires significantly less resources than the signature verification, while in wireless networks upload energy consumption is usually higher than download consumption. Therefore, a wireless reception is actually the worst wireless scenario for PLA. If the data is transmitted, then the PLA's energy advantage grows since the cost of cryptographic operations decreases while the transmission costs of extra hashes required by hash trees increases. The same applies for forwarding nodes that need to perform two wireless operations, reception and transmission, per single signature verification.

These calculations do not include the energy required for performing hash calculations, and they also do not take into account packet fragmentation, which is introduced by the inclusion of extra hashes to the packet. Fragmentation would increase the amount of sent packets, and therefore the amount of signature verifications, since additional packets must be protected by hash tree signatures. Both of these would further increase the energy consumption of the hash tree approach.

5.4.6 PLA versus ALPHA-M and -C

ALPHA variants use additional control packets that have their own headers. For bandwidth overhead calculations, we assume that packets contain standard IPv6 and UDP headers, therefore the length of these control packets, excluding IP header extension fields, is 704 bits.

In addition, ALPHA-M includes hashes to data packets. As a result, if the width of the hash tree *w* is very small, the bandwidth overhead per packet becomes large due to additional control packets per each tree. While with big values of *w* the bandwidth overhead is also large due to extra hashes included to data packets. In the case of ALPHA-C, *w* denotes the number of MACs sent simultaneously in the *S1* packet. Due to IP packet size limitations roughly 70 MACs can be included in a single *S1* packet, therefore *w* is limited to 64 in the figure for ALPHA-C.

In order to make an apples-to-apples comparison, we compare the bandwidth overhead of ALPHA-M to the lightweight PLA ignoring PLA's timestamp and sequence number fields. I.e, for PLA we take into account the bandwidth overhead of a public key and signature. This is fair, since ALPHA does not contain these security measures but they can be added to ALPHA at the expense of additional bandwidth overhead. Similarly, ALPHA lacks a TTP certificate and trust mechanism, therefore a comparison to full PLA would not give an accurate picture. Using a lightweight PLA decreases computational requirements of signature verification by about 17% compared to the full PLA verification as mentioned in Section 4.3.2.

Results are shown in figures 23 and 24 in the same format as before, positive values denote the situation where ALPHA has an energy advantage over lightweight PLA. Situations where $w = 1$ denote the case of basic ALPHA where hash trees or cumulative transmission are not utilized at all. It can be seen in the figure that ALPHA-M achieves the smallest bandwidth overhead per packet when $w = 8$. When the width of the hash tree is smaller, the bandwidth overhead is higher due to control packets, while large trees introduce higher overhead because of extra hashes in data packets.

If the cost of wireless communication is very low, then ALPHA-M has a small energy advantage when the hash tree is not very wide, and the traffic is received wirelessly. When the cost of wireless communication is closer to real-life figures, then it once again dominates the overall energy costs and PLA becomes a much more energy-efficient solution. Once again, wireless transmission and forwarding favours PLA even more, since the energy cost of wireless operations increases in those cases. These results do not take into account the energy costs of hash calculations and packet fragmentation, which would slightly increase the energy consumption of ALPHA-M.

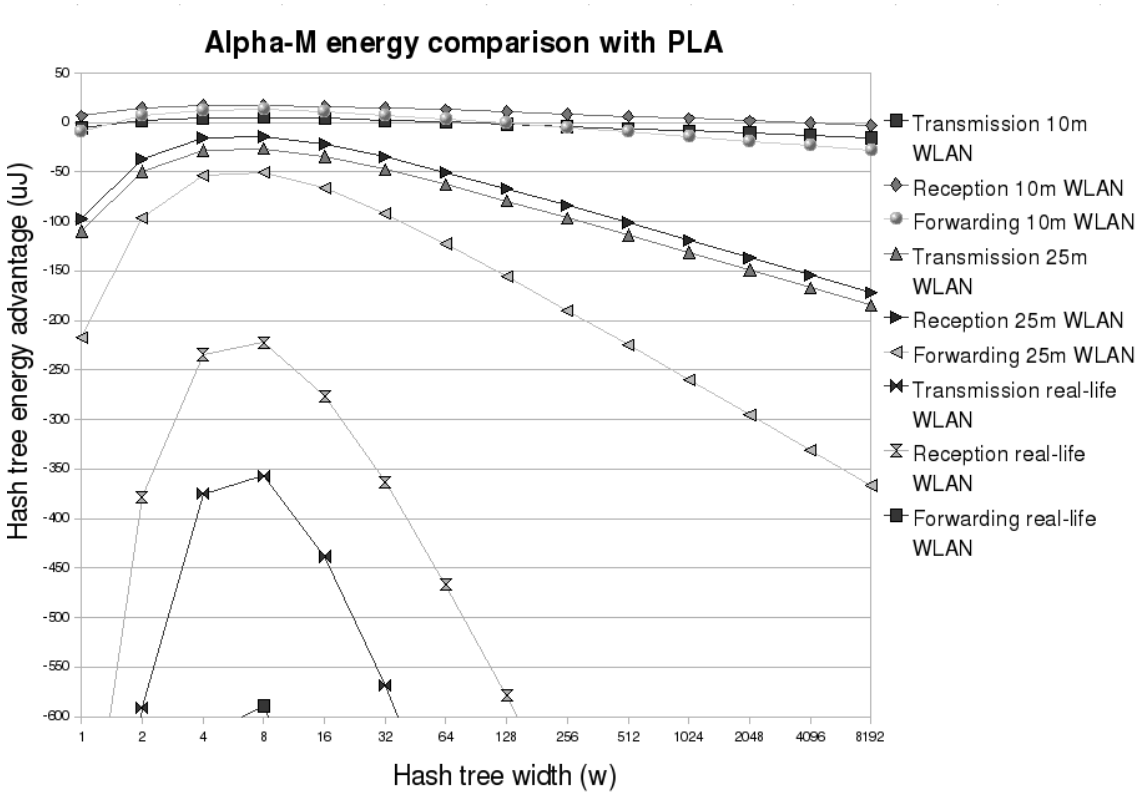


Figure 23: Per packet energy advantage of ALPHA-M versus lightweight PLA in a wireless LAN

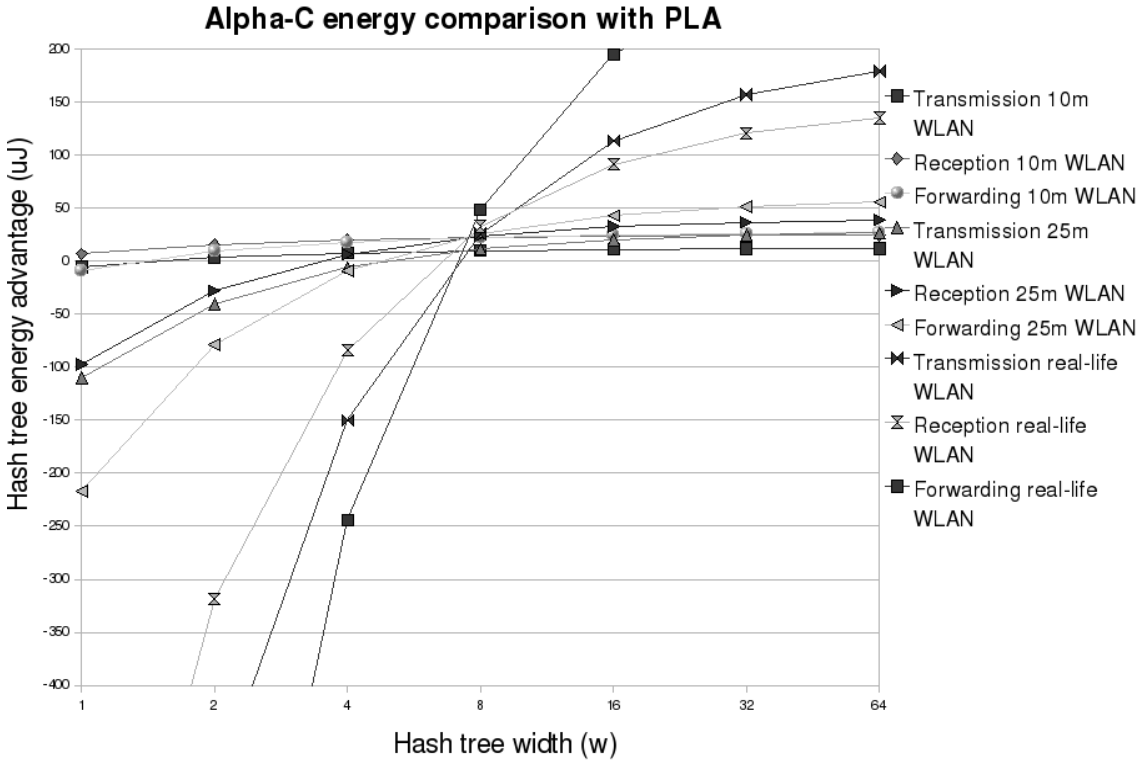


Figure 24: Per packet energy advantage of ALPHA-C versus lightweight PLA in a wireless LAN

Because the bandwidth overhead of ALPHA-C in data packets is constant, increasing *w* decreases the overall overhead. It can be seen in the figure that lightweight PLA's bandwidth overhead becomes higher than ALPHA-C's when *w* is 8 or higher. ALPHA-C achieves the best energy efficiency when the maximum amount of MACs is included in a single S1 packet.

It is important to note the disadvantages of ALPHA-C. It is a path-dependent solution that does not support independent verifications, and it also requires intermediate nodes to cache one MAC per each data packet. ALPHA-C offers a significant bandwidth and energy advantage only when *w* is 12 or higher, such large values can cause problems with a latency sensitive communication. For example, to keep the latency of VoIP traffic below 100 ms, not more than four or five packets can be buffered, and in that case ALPHA-C has an energy disadvantage against PLA.

5.4.7 Summary of the comparison

In the past it has been assumed that public key-based security solutions are inherently inefficient in terms of energy consumption and therefore are unsuitable for mobile and wireless devices. The analysis shows that with an efficient hardware accelerator for cryptographic operations, the cost of wireless transmission will dominate the verification-related power consumption. Therefore PLA, which is based on public key digital signatures, is actually more energy efficient since it uses less bandwidth compared to hash tree or hash-chain-based solutions like ALPHA.

The reason for PLA's advantage is the following. In real-life cases wireless reception uses almost 1 μJ/bit of energy, while the cost of performing a signature verification is only 26 μJ. Therefore, even if the PLA can reduce a bandwidth overhead only by 100 bits per packet, it already becomes a much more energy-efficient solution. While some studies [89] have achieved orders of magnitude better energy efficiency, they have achieved those in a special simulated environment without taking into account natural obstacles and buildings, therefore they can not be considered a realistic estimate of energy consumption of the wireless communication.

Due to the nature of elliptic curve cryptography, a signature generation requires significantly less resources than a signature verification, while in wireless networks the

upload energy consumption is usually higher than download consumption. Therefore, for a sending node the energy consumption of cryptographic operations would be even less. The same applies to the wireless ad-hoc networks where nodes often forward the traffic and perform two wireless operations, reception and transmission, per a single signature verification. In these cases the PLA is an even more attractive solution.

The idle power consumption of the chip performing signature verifications has not been taken into account here. If the node receives only a handful of packets per second, then a power consumption per signature verification will be higher. On the other hand, in this case the power consumption would also increase for a wireless reception, since the wireless interface also has a fixed idle power consumption. In any case, the cost of wireless communication would still dominate the overall power consumption in most cases, even if we double or triple the energy consumption of a signature verification.

In the future, the power consumption of semiconductors will likely continue to decrease at a rapid pace [53]. The same does not apply to the energy cost of wireless transmission since it also depends on physical properties of the transmission medium that can not be changed. Therefore, the energy consumption of cryptographic computations will likely decrease faster in the future compared to energy consumption of the wireless communication. This would make PLA even more attractive in the future, since it decreases the bandwidth overhead at the expense of the computing power.

If buffering a large amount of packets is feasible then ALPHA-C becomes a very energy-efficient solution due to its lower bandwidth overhead. However, PLA is overall a much more flexible solution, since it does not depend on the transit path and does not require caching of packets at the sender's end for optimal performance. Therefore, PLA can also be used with a latency sensitive communication and dynamic wireless ad-hoc networks where the network topology changes frequently.

5.5 Summary

Overall PLA satisfies most of requirements presented in chapters 2 and 3. The performance and power consumption of PLA is also good. With a dedicated hardware acceleration built on the modern manufacturing process PLA is scalable to future 100 Gbps network interfaces. PLA is also suitable for mobile devices since the per-packet

power consumption of cryptographic operations is lower than the energy cost of transmitting an average-sized packet wirelessly. PLA also achieves better energy efficiency in wireless networks than hash tree and hash chain-based security solutions, due to PLA's lower bandwidth overhead.

6. Network-layer applications of PLA

This chapter describes PLA's applications on the network layer. These applications include prevention of unwanted traffic on the network through controlling incoming connections and securing the network infrastructure.

6.1 Controlling incoming connections

One way to solve the problem of unwanted incoming connections presented in Section 2.5 is to block all incoming connections to the recipient that are not explicitly allowed. Such blocking can be naturally done in a personal firewall, but in that case incoming connection attempts would still consume resources in the recipient's access network. Thus, the network should provide means to block unwanted incoming connections already at the access network level, before the unwanted traffic can even reach the destination. Such blocking would also make it much more difficult to launch DoS attacks against the recipient or the recipient's network.

PLA together with a traditional certificate mechanism can be used to solve this problem [69]. Such a method would work as follows: the potential recipient grants explicit certificates to trusted initiators to initiate incoming connections to the recipient. PLA is used to guarantee on the packet level that connections to the recipient really originate from trusted initiators. An example of such a system is shown in the figure below.

The system presented in Figure 25 consists of four main parts. The initiator is the party that initiates the connection. The proxy is an entity in which the recipient trusts. The task of the proxy is to give certificates to trusted initiators for making incoming connections to the recipient. The proxy also keeps track of the recipient's IP address if the recipient is changing networks. In order to eliminate a single point of failure, proxies form a Distributed Hast Table (DHT) [46] network. There are two firewalls located in the initiator's and recipient's access networks. They take notice of certificates that are passing through it and firewalls block all outgoing and incoming connections to the network unless the recipient is a valid entity within the network and the incoming connection to the recipient has been explicitly allowed via certificates.

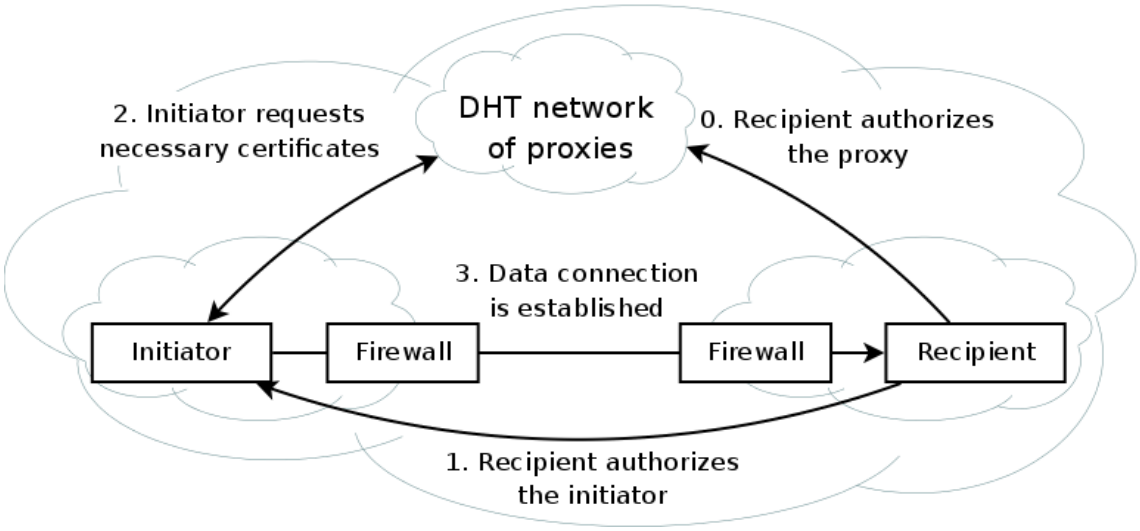


Figure 25. On overview of a system for controlling incoming connections

In the very beginning, the recipient authorizes the proxy using a certificate. The recipient also authorizes a trusted initiator by giving him a certificate that certifies the initiator's public key, this certificate exchange can also be carried off-line and it is shown as step 1 in the figure. After the initiator has received a certificate from the recipient, the initiator can contact the proxy to request all necessary certificates for making an incoming connection. These certificates include a certificate that was given by the recipient to the proxy in step 0, a certificate from the recipient to the initiator, and a certificate from the firewall (i.e., the access network) to the recipient. In addition, the proxy reports the recipient's current IP address to the initiator. In the last step, the initiator first sends all those certificates to the recipient using a control message and afterwards the data connection can be established. Firewalls check that the certificates form a valid certificate chain: recipient's firewall => recipient => proxy => initiator. Such a chain shows that the recipient, which has a right to use the access network, has authorized the proxy that has authorized the initiator. If this certificate chain is in order, firewalls will allow the initiator to establish a connection to the recipient. The main task of the initiator's firewall is to stop unwanted connections as soon as possible. Naturally, revocation and delegation of certificates for making incoming connections is also supported under such a system.

In such an application, PLA is necessary to ensure that the data sent to the recipient is really coming from a trusted initiator. The initiator's public key inside the PLA header of data packets must match the initiator's public key present in a certificate chain, only in that case will the firewall let traffic through. Hence, a malicious party will not be able to

make incoming connections by using captured certificates that have been granted to other trusted initiators.

Subsequent sections contain examples of how such a system works in practise, and how rights for creating incoming connections can be delegated and revoked.

6.1.1 Controlling incoming connections example

Figure 26 presents a more detailed example of controlling incoming connections. The certificate format is explained in Appendix B.

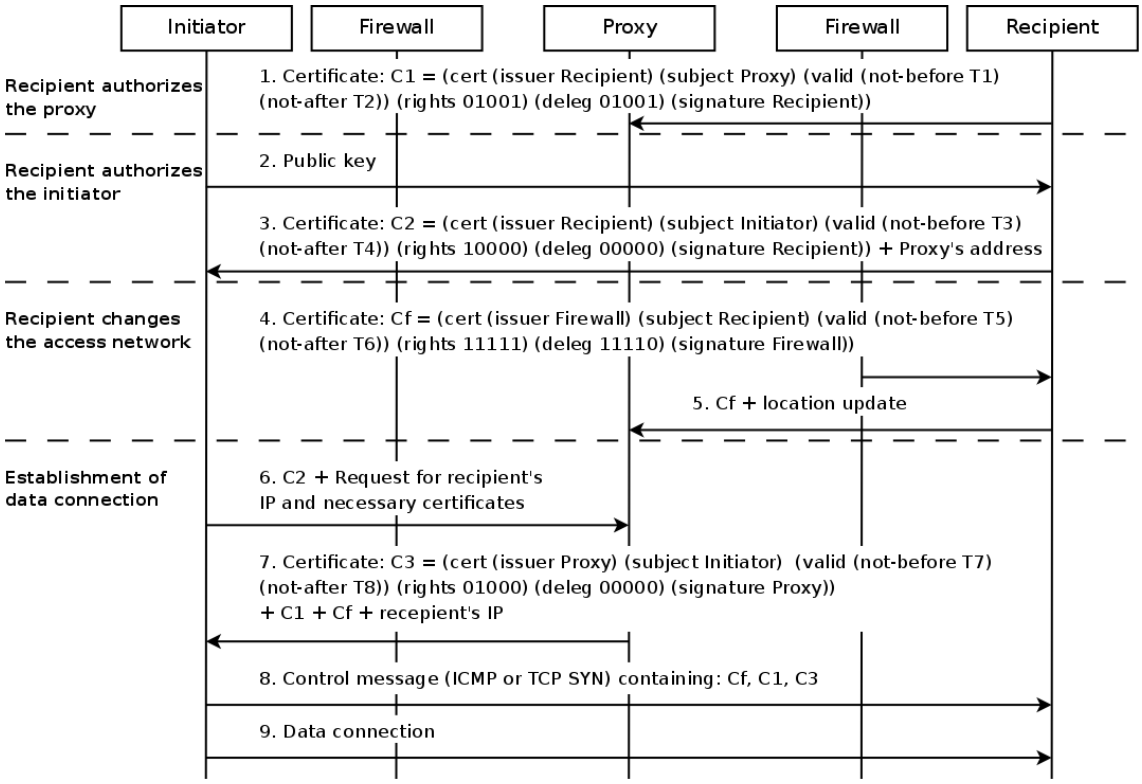


Figure 26: Flow diagram for controlling incoming connections

In the first step, the recipient gives a C1 certificate to the proxy[1], therefore the proxy can disclose the recipient's location to certain trusted initiators and the proxy can also authorize them to initiate connection to the recipient. Based on traditional certificate formats, in this certificate the recipient is the issuer of the certificate, the proxy is the subject and the certificate is signed by the recipient.

1 The proxy may be protected by additional firewalls not shown here.

In steps 2 and 3 the recipient authorizes the initiator, these steps can also be carried out offline. The public key given in step 2 can either be a real public key, or the signature part of the TTP certificate as described in Section 3.5. The aim of the C2 certificate given in step 3 is to allow the initiator to request the recipient's current location (IP address) and necessary certificates from the proxy (the right for session initialization management bit is set to one in C2 certificate).

In step 4 the recipient changes the access network. The recipient gets a certificate Cf from the network as part of the network's AAA procedure (denoted in the figure as "firewall") that allows the recipient to use the new access network. The recipient sends a location update together with this new Cf certificate to the proxy in step 5.

Steps 6-9 describe establishment of a data connection between the initiator and recipient. In step 6, the initiator contacts the proxy. The initiator sends a C2 certificate to the proxy and requests the IP address of the recipient together with all required certificates. The recipient might change the network frequently, thus the initiator must retrieve the recipient's latest IP address from the proxy.

Based on the C2 certificate, the proxy knows that the initiator is authorized to get the recipient's latest IP address and to receive a certificate for sending data to the recipient. As a result, the proxy creates a new C3 certificate in step 7 that allows the initiator to initiate the incoming connection to the recipient. Certificates Cf, C1, C3 together with the recipient's IP address are then sent to the initiator.

In step 8, the initiator first sends a control message, using e.g. an ICMP or TCP SYN message, with certificates Cf, C1 and C3 to the recipient. This allows the firewall of the recipient's network to check that the recipient is a legitimate node in the access network and the recipient is willing to receive the incoming traffic from the initiator. The Cf certificate tells the firewall that the recipient is authorized by the access network and the recipient is a valid entity. The C1 and C3 certificates create a certificate chain together: recipient => proxy => initiator, which denotes that the initiator has the right to initiate the connection to the recipient. Finally, the data connection is established in step 9.

The C3 certificate that allows the initiator to create an incoming connection is given through the proxy because that way the recipient can easily revoke it. For example, if

after step 2 the recipient does not trust the initiator any more, the recipient can notify the proxy that the C2 certificate is revoked and thus the initiator would not be able to receive the C3 certificate from the proxy. If the recipient would have sent the C3 certificate directly to the initiator, then the revocation of the C3 would be much more difficult.

If the recipient uses several network interfaces simultaneously, the proxy could return several different IP addresses and associated Cf certificates to the initiator. In addition to permitting certain initiators to send data to itself, the recipient could also specify more general policies to the proxy, such as "now I do not want to be disturbed, do not allow anybody to contact me" or "anybody can contact me, even without a valid C2 certificate".

6.1.2 Revocation of rights for incoming connections

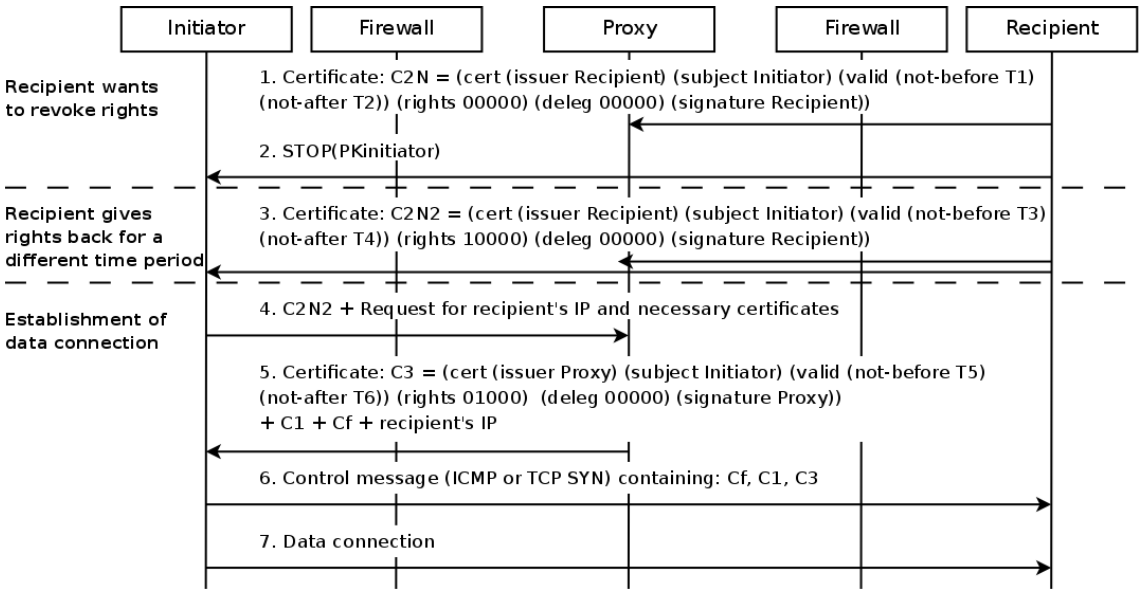


Figure 27: Revocation of rights for making an incoming connection

Figure 27 describes a situation where the recipient revokes and later reinstates a right for incoming connections. The revocation is done by notifying the proxy, firewall and the initiator. It is possible for the recipient to change rights by sending a new version of the C2 certificate (called C2N and C2N2 in this example) to the proxy. The proxy will keep only the newest version of this kind of certificate in its database and this certificate always overrides the certificate received from the initiator. The example goes as follows.

In the first step, the recipient creates a C2N certificate, which is similar to the C2 certificate in the previous use case but all rights are cleared. This C2N certificate is then sent to the proxy and it will override any previous similar certificate. If the initiator requests the recipient's IP address or C3 certificate from the proxy (step 6 in the previous use case), the proxy denies the request because it possesses a valid C2N certificate that has its rights set to zero. In order to stop the data flow from the initiator, the recipient also uses a *STOP* message as outlined in Section 3.7.1.

In the second step the recipient wants to give rights back for a different time period. A new certificate, C2N2, is created and this certificate has a right for session initialization management bit set to one. This certificate is sent to both the proxy and the initiator. This certificate will also pass the firewalls which will take note of it.

In step 3, the initiator can request the recipient's IP address with necessary certificates from the proxy. If the initiator will make a request to the proxy using the original C2 certificate outside the validity time of the C2N2 certificate, the request will be denied since the proxy has the C2N2 certificate in its database and this certificate automatically overrides other certificates.

Just like in a previous case, the proxy sends the recipient's IP address with necessary certificates to the initiator in step 4. In step 5 certificates Cf, C1 and C3 show that the initiator has a valid rights for making an incoming connection to the recipient and the C2N2 certificate does not refute this right. Finally, the data connection is established in step 6.

6.1.3 Delegation of rights

The previous examples assumed that the recipient always has a direct trust relationship with the initiator. However, such a requirement is often not necessary. Most of the people are members of some organizations, such as companies, schools, hobby groups, social networking groups etc., and the basic idea of controlling incoming connections can be easily extended to handle indirect trust relationships. For example, instead of authorizing all initiators directly, the recipient could authorize all members of a trusted organization.

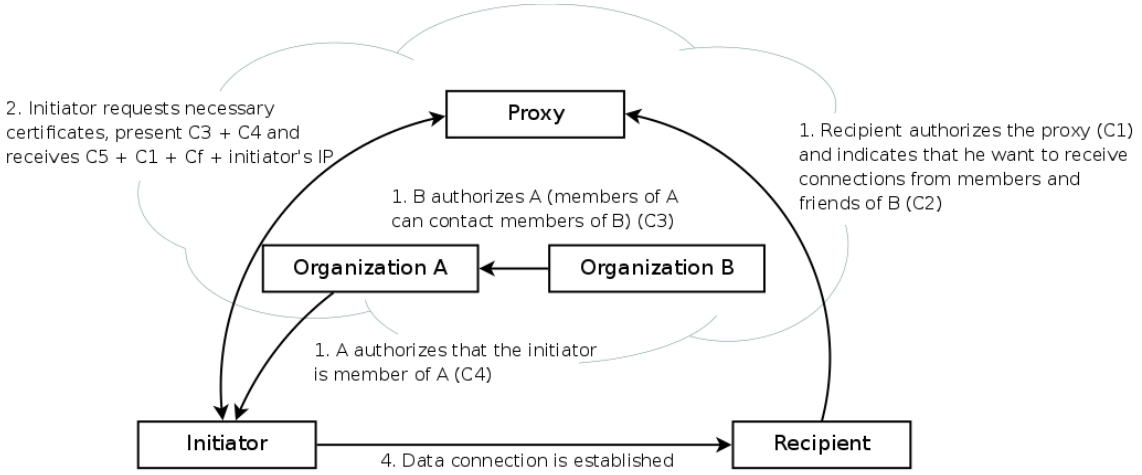


Figure 28: Delegation of rights for incoming connections between different organizations

The example in Figure 28 describes how the rights for controlling incoming connections can be delegated between different organizations in a flexible way. In the first step preliminary authorizations take place. The recipient authorizes the proxy with the C1 certificate as in previous examples, he also indicates with the C2 certificate that he trusts organization B, and wants to receive connections from members and friends of B. Additionally, organization B authorizes A with the C3 certificate, meaning that members of A can contact members of B, and A authorizes that the initiator is its member (C4 certificate).

In the second step, the initiator contacts the proxy and presents certificates C3 and C4. The proxy, which already possesses the C2 certificate, checks that certificates C2, C3, and C4 are in order and form the chain of trust: Recipient => B => A => Initiator. As a results, the initiator receives a traffic certificate and can start the data connection as in previous examples.

Such delegation is very flexible and can be suitable for a wide range of real-world applications. The recipient can control how far memberships propagate, e.g., the recipient can decide to accept connections from members of B, but not from A's members. It is also possible to build a traditional blacklist-based system where all connections are allowed by default unless the sender has been explicitly blocked.

6.1.4 Implications of controlling incoming connections

While the idea of requiring an explicit permission for connections may sound very radical, it is suitable for many applications. For example, almost all e-mails received from completely unknown senders are Spam. Valid e-mails usually originate from people with whom the receiver has at least some kind of relationship or contact through work, family ties, or common friends and hobbies. Therefore, controlling incoming connections with a flexible delegation system as outlined above does not prevent valid communication.

In the first phase, such a method could be used to prevent unwanted traffic in e-mail applications, instant messaging systems, VoIP calls and other similar application. Controlling incoming connections for most of Internet's traffic would produce higher overhead, but it would also prevent most denial-of-service attacks since malicious nodes could not freely send traffic to the target.

6.2 Securing the network infrastructure

PLA can be effectively used to secure the network infrastructure due to its ability to immediately detect and react against different kinds of attacks. In contrast, on the current Internet, unwanted or malicious traffic is usually detected and blocked in the firewall near the destination. Therefore, such traffic consumes resources in the whole network and induces the load on the destination's network, possibly preventing the destination from serving benevolent users.

The PLA-enabled node that receives a forged, duplicated, or delayed packet can discard the packet immediately, preventing this packet from consuming resources in the rest of the network and stopping a possible attack quickly. Furthermore, since the PLA header contains information for calculating a sender's public key which is certified by some trusted third party, the attacker can be traced and caught easier compared to current solutions.

In the current Internet, reacting to attacks, i.e. trying to remove malicious nodes from the network, requires a lot of manual work and communication, which consumes some time and makes it difficult to react to the attacks quickly. Since PLA is based on strong

cryptographic techniques, and PLA headers contain all necessary information for verification, it is possible to detect and react to attacks in a more automatic way, lowering the window of vulnerability to attacks. The applicability of PLA for the prevention of different kinds of attacks is discussed below.

6.2.1 Denial-of-service attacks

PLA offers protection against denial-of-service attacks on many levels. First, the mechanism for controlling incoming connections presented above can be used as a pre-emptive measure against most of the denial-of-service attacks, since potential attackers can not freely attack a random victim. Furthermore, *STOP* messages offer a way for the victim to mitigate the attack by blocking the traffic from the attackers. Finally, the TTP mechanisms allow wrongdoers to be removed from the network, shortening the timespan of attacks and preventing the same attacker from attacking multiple targets in a row. It is important to note that these security mechanisms are independent from each other and it is not compulsory to use all of them at the same time. The network's security would be significantly improved even if only one or two of these mechanisms are used. A more detailed explanation of these security mechanisms offered by PLA follows.

STOP messages, which are outlined in section 3.7.1, offer a flexible way for the victim to prevent the attacker from sending further traffic. Since it is not feasible for all routers to maintain a blacklist for all passing connections, *STOP* messages should be stored at the PLA-enabled router closest to the source.

Figure 29 describes the situation where the source from domain A sends unwanted traffic to the destination in domain D, and the destination replies with a *STOP* message stating that it does not want to receive any further traffic from the destination. Such a *STOP* message is stored at the router that is closest to the destination. In this way the unwanted traffic is stopped close to the source before it can consume resources in the rest of the network.

Since it is not realistic to assume that all networks on the Internet always conform to all *STOP* messages, routers located in other domains on the path may also store *STOP* messages in order to check whether other parties are really conforming to these messages. For example, if domain C receives a packet from domain B that has been

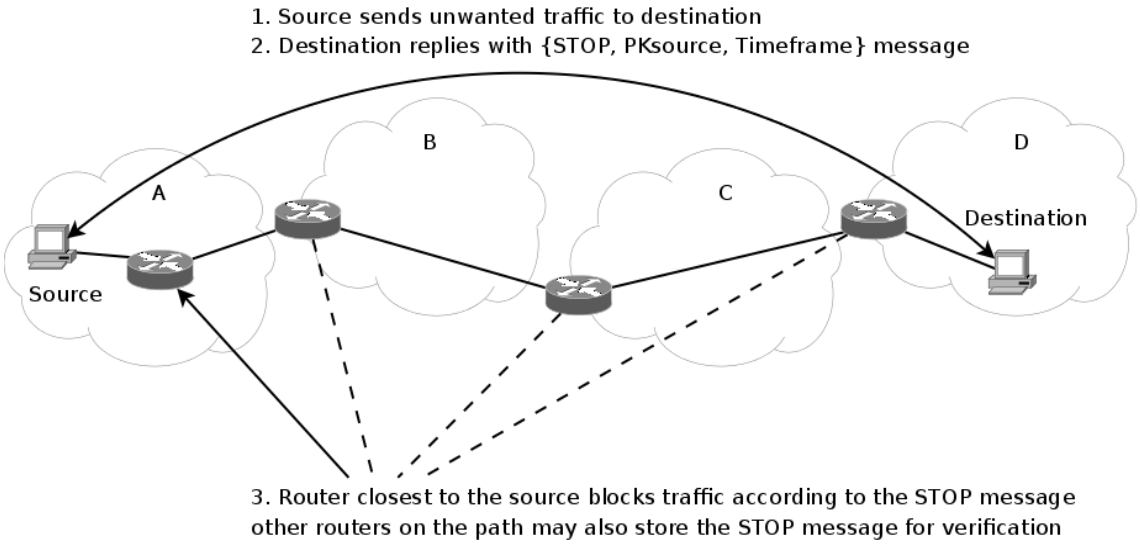


Figure 29: An example of stopping unwanted traffic with STOP messages

blocked by a previous *STOP* message, then domain C has undeniable proof[1] that domains A and B have not respected the *STOP* message.

Such checks do not need to be done for every *STOP* message or the packet. On the public Internet it should be enough for the intermediate routers to store just a small fraction of *STOP* messages to the blacklist, and check a fraction of incoming packets against the blacklist. Since packets travel through multiple domains, and serious attacks consist of large amounts of packets, the non-conformance to *STOP* messages will be likely to be noticed at some point of the network. After a domain has noticed that other domains are not conforming to *STOP* messages, it can take their traffic under a stricter scrutiny and consider sanctions against these for not adhering to the rules. Such accountability offers a strong incentive for every domain to obey rules, and to block unwanted traffic near its origin.

A hostile node can try to poison a router's caches by sending invalid *STOP* messages. Such an attack can be mitigated by various means. Since the goal of *STOP* messages is eventually to stop the unwanted traffic as close to the origin as possible, the router near the edge can use aggregation techniques to reduce the necessary state. For example, if a node sends a large amount of separate *STOP* messages to every node located in the subnet, the router can block all traffic from the subnet to the node. Hence, there is no need to store each STOP message separately.

1 Since all traffic is protected by timestamps and cryptographic signatures, it is easy to detect the situation where an incoming packet violates the previously sent *STOP* message.

The TTP mechanism provides several means to fight against DoS attacks. In case of misuse, a user's short-term TTP certificate will not be renewed, preventing him from continuing the attack after expiration. Additionally, there is a bigger chance that perpetrators will be caught since the TTP knows each user's real identity. Section 7.1.5 contains a more detailed discussion on catching culprits in the case of misuse. Routers inside domains may be also configured to only accept traffic authorized by a local TTP.

The TTP mechanism also offers benefits to the end user. In a situation where the user's computer has been compromised, and floods the network with garbage, one currently used policy is simply to cut off the user's Internet access altogether. With PLA, only the infected computer would lose access to the network. The user's other computers with different cryptographic identities would be able to access the network normally.

A denial-of-service attack can also be launched by an attacker who resides in the middle of the network and makes a large amount of copies of valid packets. Such an attack is especially dangerous and easy to carry out in wireless networks where the attacker can easily intercept valid traffic. Even if a traditional security solution like IPSec is used, intermediate nodes can not determine that the extra packets are just copies of the original, because they cannot look inside an encrypted IPSec packet. With PLA, intermediate nodes can easily detect duplicates by checking the timestamp and sequence number fields in the PLA header.

It is important to note that protection against DDoS attacks is an extremely challenging task on the Internet, which is composed of thousands of distinct domains and hundreds of millions of users. While the presented mechanisms prevent and mitigate DDoS attacks in most cases, they may not offer complete protection against DDoS attacks in all real-life situations. Nevertheless, significantly reducing the amount and severity of DDoS attacks is a major improvement compared to the current Internet.

PLA implements some of the key principles presented in the study that outlines necessary changes for the DoS-resistant Internet [47]. The rights field of the TTP certificate is basically a more flexible version of the state setup bit proposed in the study, and it is used to distinguish various types of traffic. Stopping unwanted traffic near its origin using the PLA header information and *STOP* messages is equivalent to the proposed middlewalls. Since PLA is based on strong cryptographic signatures,

which can be independently verified at wire speed, we feel that using puzzles and nonces during the connection establishment phase as proposed in the study is unnecessary. Other methods mentioned in the study, such as the separation of client and server addresses, and efficient multicast are out of the scope of this work.

Several existing DoS protection solutions are based on capabilities that are tied to a sender's IP address or the subnet. Such an approach is problematic when users utilize mobile devices and are changing networks frequently. Since PLA is based on cryptographic identifiers, it does not have this limitation.

6.2.2 Phishing attacks

PLA helps protect against phishing attacks in several ways. First, combining PLA-level cryptographic keys with application-level identities would offer protection against identity thefts which are often used in a phishing attack, since the attacker would be unable to send packets in the victim's name without gaining access to the victim's private key. Second, since the sender's public key is included in every packet, a recipient can check the validity of packets that it receives from, e.g, a WWW server. The recipient can also check whether the server's public key is trusted by a reliable TTP. This decreases the probability of a successful phishing attack, since the attacker participating in a phishing attack would eventually lose its TTP certificate.

6.2.3 Spoofing-related attacks

PLA includes information about a sender's public key in every packet along with a signature that is generated using the sender's private key. This offers good protection against various spoofing attacks. The attacker is unable to send spoofed packets in the victim's name and the attacker will also be unable to launch an attack by taking a valid packet and changing some fields in it because such modification would break the signature of the packet.

6.2.4 Replay attacks

Replay attacks can be easily detected with PLA, since the PLA header contains a time stamp and a sequence number. Thus, packets which are duplicated or significantly delayed will be detected and discarded.

6.3 Benefits of PLA in a real-life security solution

In this section, we will revisit the Internet banking security example which was presented in Section 2.3.5 to discuss what benefits PLA could provide in a such a situation. In order to provide maximum security and reliability, the original proposal contained several layers of security solutions. With the help of PLA, it is possible to reduce the number of security layers as presented in Figure 30.

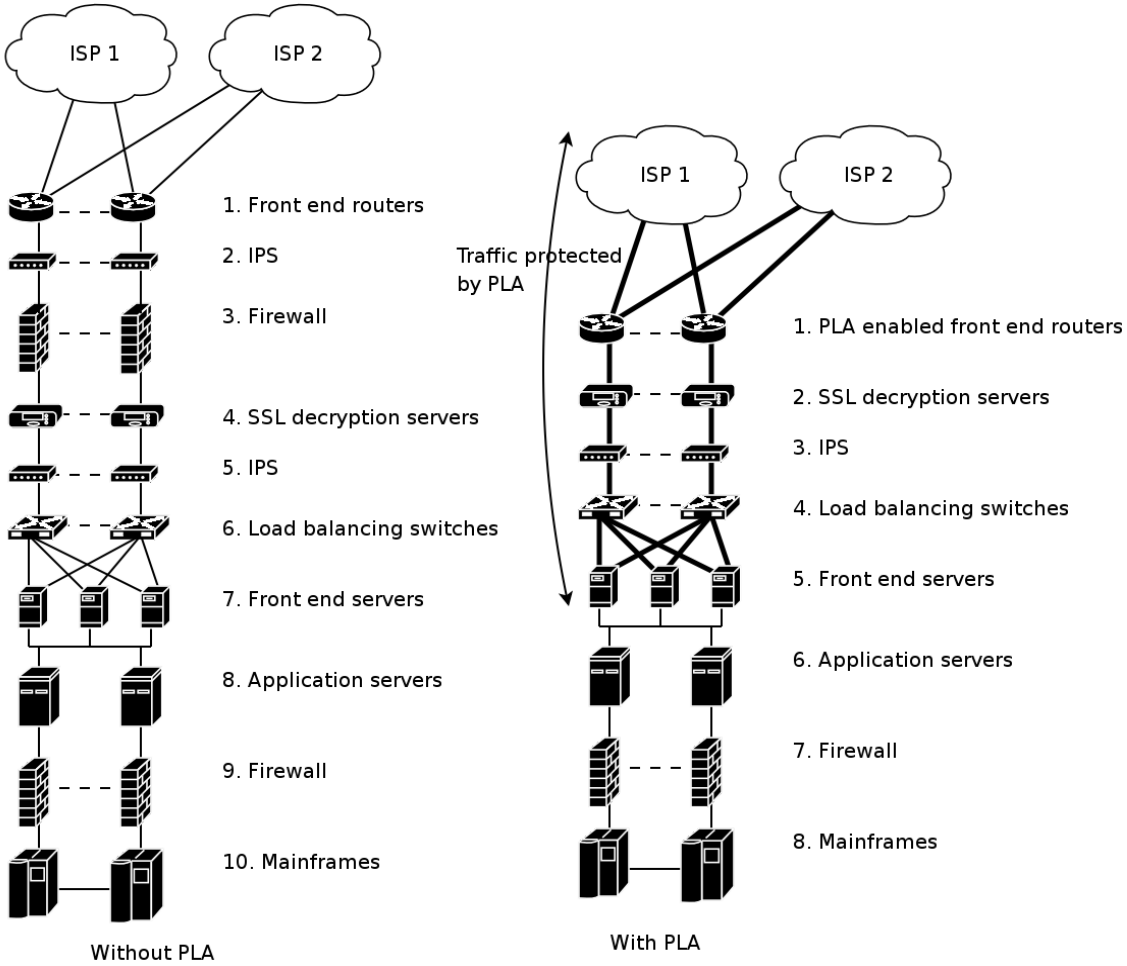


Figure 30. An example of a security solution for Internet banking utilizing PLA

The original proposed security solution had five security-related layers before actual application-related servers: front-end routers, two Intrusion Prevention Systems (IPS),

firewall, and SSL decryption servers. Having PLA-enabled front-end routers could simplify the solution by removing the need for a separate IPS and a firewall. These front-end routers would check the authenticity of incoming packets and the validity of users and let only valid packets from trusted users through. Because SSL decryption is a resource-intensive operation, having dedicated servers to handle decryption tasks is preferred in any case. In this example, PLA-enabled traffic would be transmitted all the way to the front-end servers, which would then remove the PLA header from the incoming traffic and send plain IP packets forward to the application servers.

To further improve security, the public key of the PLA header could be utilized during the user authentication process of an online banking session. When obtaining an online banking account, a user would present his public key to the bank and the user's username/password combination would be usable only if it is accompanied by the user's public key in the PLA header. This would provide additional security, since even if a hostile party is able to capture the username/password combination, it would not be able to gain access to the victim's bank account without having access to the private key of the victim. Such a system would naturally support delegation of rights as presented in Section 3.8.2, thus a user could temporarily transfer his rights to another computer and use it to access his bank account.

7. Using PLA at higher layers

This chapter describes how PLA can be used to provide security and accountability at higher layers. Potential use cases of PLA include flexible billing, securing the mobility management related signalling, and Internet-wide user authentication and roaming.

7.1 Strong accountability, Internet-wide roaming and flexible billing

This section describes how PLA can be used to implement strong accountability, and a secure and flexible Internet-wide roaming and billing. More discussion is available in [70].

To alleviate the lack of built-in accountability on the Internet, several countries have introduced or proposed data retention laws that require Internet service providers (ISPs) to store their users' identity and information related to users' communication for months or years. Such legislation is problematic for several reasons; first it introduces a significant privacy issue, since users' private details are stored in multiple places for long periods of time. It also increases costs of the ISPs since they must maintain and store sensitive records. And finally it is not really secure, since the data retention is based on IP address records and not on strong security mechanisms.

7.1.1 Status of the current data-retention legislation

According to European Union (EU) directive 2006/24/EC [37], providers of publicly available electronic communications services or of public communications networks must store data related to communication for 6-24 months. In case of computer networks this data includes the user's name, address, the IP address allocated to the user, duration of the communication and other related information.

EU defines the communication network as "*transmission systems and, where applicable, switching or routing equipment and other resources which permit the conveyance of signals by wire, by radio, by optical or by other electromagnetic means, including satellite networks, fixed (circuit- and packet-switched, including Internet) and mobile terrestrial networks, electricity cable systems, to the extent that they are used for the purpose of transmitting signals, networks used for radio and television*

broadcasting, and cable television networks, irrespective of the type of information conveyed" [38]. Basically, even a public wireless LAN base station would fall under this definition.

Similar legislation has also been proposed in the United States [30], requiring any network access provider to store communication-related data for at least two years.

The current EU data-retention legislation presents several problems. First, it places a heavy burden on the access network providers, i.e. ISPs, since they must securely store a large amount of private data for several months or even years. Such an additional overhead increases costs and significantly increases the barrier of entry to become an access network provider. This in turn decreases the competition since small ISPs may not have sufficient resources to conform with the legislation. Such legislation also makes community networks [42], [40] impractical in many cases, since the administration overhead of running the network becomes very high.

The second issue is the privacy of users, since personal details of users will be stored in all access networks that they use. For users that travel a lot there may be tens or even hundreds of such networks. This significantly increases the possibility that at least one of the networks leaks users' personal details. Such leaks can occur for a variety of reasons, e.g, through discarded or lost computers [51], [12]. The access network may also leak or sell personal information on purpose. Since the amount of access networks on the Internet is very large, it is not possible to effectively monitor every one of them to guarantee that they are handling users' personal information in a secure manner.

Finally, the security of such a system is questionable. The legislation relies on identifying users through their IP addresses, however, IP addresses can be spoofed relatively easily defeating the main goal of the system. Furthermore, the current data retention mechanism only tries to prove that a user has been using a certain IP address at a certain time frame. Data related to the actual communication is not covered by the legislation, therefore there is no way to prove that the user has really participated in the attack and sent malicious packets to the victim. This opens a possibility for parties connected to the Internet to frame a certain user by faking their logs and claiming that an attack has occurred from the user's IP address. In order to provide a strong accountability through data retention, the access network would also need to store

information about destination IP addresses and the amount of transferred data for each connection that the user establishes. This would not be a feasible approach since the amount of collected data would be very large, and such an approach would also produce huge privacy risks.

7.1.2 Implementing strong accountability with PLA

Figure 31 describes identity relationships between the traffic and users on the current Internet and when PLA is used. In the current Internet the network traffic is not directly tied to any user's identity, the traffic just contains a sender's IP address. Therefore, the data retention legislation often requires network access providers to establish a link between these IP addresses and users' real identities through retention of necessary data. Basically ISPs must store "user's IP address – user's real identity" mappings. This link is marked as a dashed line in the figure, since it is not a strong one. It is possible to spoof IP addresses and additionally, not all ISPs may have a sufficient knowledge or resources to accurately register their users and securely store data for an extended time period. As a result, some of the stored data may be incorrect or can be lost.

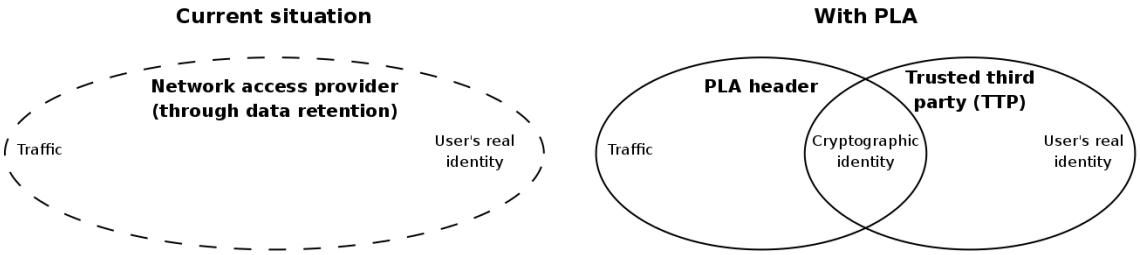


Figure 31: Relationships between identities and network traffic in the current Internet (left) and with PLA (right)

By introducing cryptographic identities PLA separates this problem into two distinct cases: mapping the network traffic into cryptographic identities, and mapping cryptographic identities into real ones. The PLA header ties all traffic sent by the user to the user's cryptographic identity with a cryptographic signature. This signature offers non-repudiation; the sender of the packet can not deny sending it. The task of the trusted third party is to link the user's cryptographic identity with his real identity. Both of these bindings are strong since they are based on cryptographic techniques and are marked as a solid line in the figure. To allow the use of pseudonyms, a single user may possess multiple cryptographic identities, but a single cryptographic identity always refers to the single user.

This approach significantly reduces the burden of the access network providers since they do not need to store users' real identities for accountability purposes. Also the privacy of the system is improved. The network access provider does not necessarily need to store users' private information for months or years. Actually, in some cases the access network provider does not need to see users' real identity at all. Similarly, the TTP does not see the actual traffic, hence, there is not a single entity in the network that sees a direct mapping between user-generated traffic and the user's real identity.

PLA also improves the security properties of the system since the user's cryptographic identity is bound to every sent packet by the cryptographic signature providing inherently strong accountability. This prevents spoofing attacks that are relatively easy to perform for IP addresses.

With PLA, the network access provider may, but is not obliged to, provide a TTP service. Therefore, it would be logical for TTPs to be significantly outnumbered by the access network providers; large companies with more resources would provide TTP services, but small access network providers could opt out. As a result, parties providing TTP services would have much stricter requirements regarding identifying their users and storing personal identities securely for a long period of time. In addition it would be easier to audit and control that TTPs are really storing personal data securely.

7.1.3 Managing user identities

PLA allows a flexible trust-management system where TTPs act as identity providers. A single user can use services of multiple TTPs, and access network providers are also connected to the global TTP system through the trust relationship with their own TTP.

Figure 32 describes an example of the trust architecture of the system. The left side contains access networks marked as grey clouds. Each access network has at least one customer relationship with a TTP, for example the access network AN2 is a customer of TTP D while AN3 is a customer of both TTP A and B. These customer relationships are marked as a bold line. TTPs also form a global system through trust relations with each other. Finally, users are customers of one or several TTPs. Basically, TTPs act as middlemen between the users and providers of access networks to facilitate trust relations between them. For example, all users can use access network AN1 since its

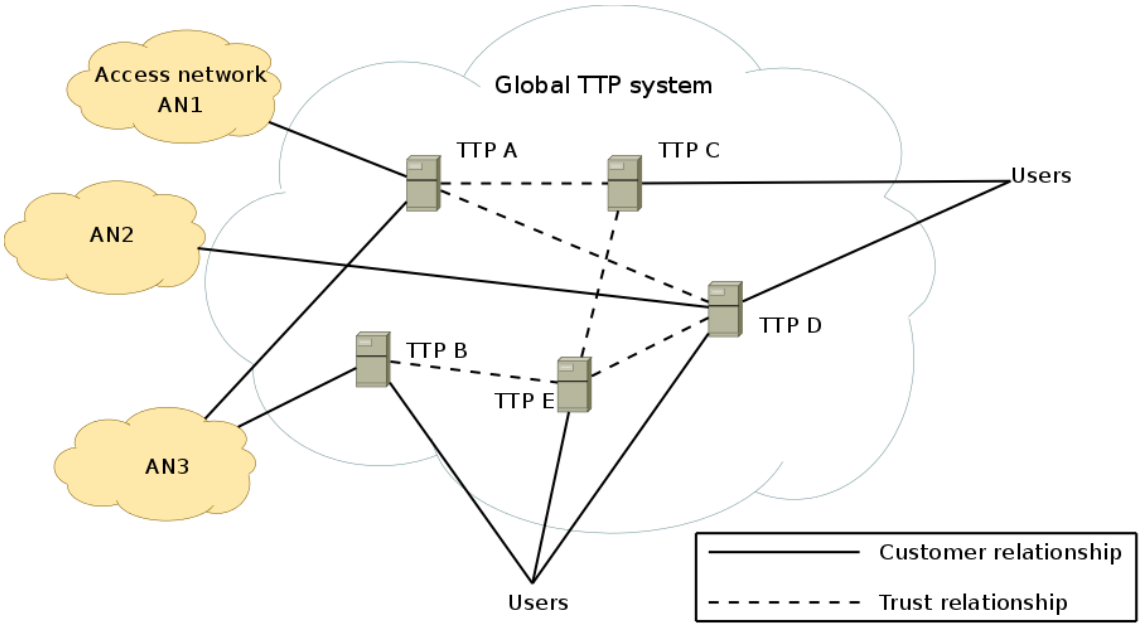


Figure 32: An example of trust relationships in a PLA-based network architecture

TTP A has direct or indirect trust relationships with all TTPs that have authorized the users. Such an approach eliminates the need for users to establish direct customer relationships with access network providers. TTPs can also be used to handle billing in a transparent manner, further increasing the flexibility and reducing the burden of access network providers. The system also works on an international scale since TTPs can create contracts with foreign users and access network providers.

Such a system would allow users to use access networks without revealing their real identity to the access network owners. In principle, this would work as roaming with current mobile phones. For example, GSM phones utilize SIM cards [39] to identify users on the network, and other mobile phone systems use similar technologies. The SIM card does not contain the user's real identity, instead it contains an IMSI number, which identifies both the user and its operator. The mobile phone user is able to utilize a foreign network as long as the network owner has an agreement with the operator that has issued the SIM card, there is no need to have a direct customer relationship between the user and the foreign operator. Billing is handled directly based on IMSI number and the real identity of the user is not revealed to the foreign access network. Another analogy to the system is an international credit card. Users can utilize their credit cards abroad as long as the store has a direct or indirect customer relationship with the credit card provider.

7.1.4 Authentication and bootstrapping

Section 3.8 described the general case of bootstrapping a completely new node to use PLA. This section describes how the authentication and bootstrapping is handled when a PLA-enabled node arrives to the foreign access network.

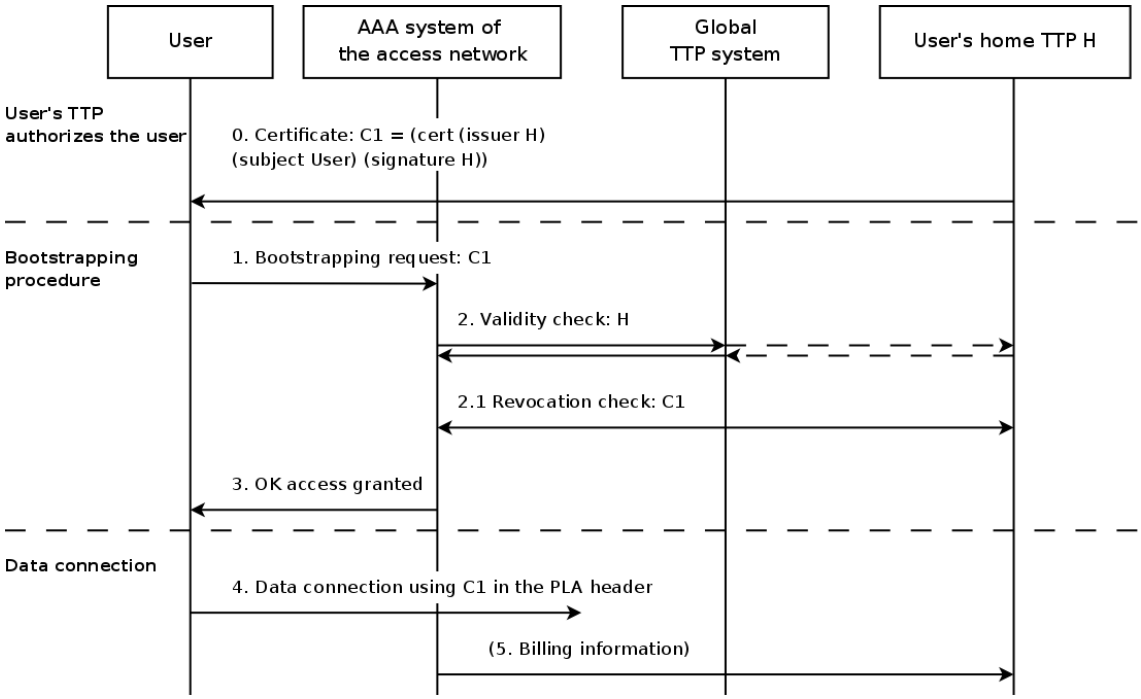


Figure 33: An example of the bootstrapping procedure with PLA

The bootstrapping shown in Figure 33 is basically equivalent to the AAA procedure. First, we assume that the user has already received a certificate (C1) from some TTP, in this case a home TTP H. The C1 certificate denotes that the user is a valid entity and is authorized to use the network. This certificate exchange is shown as step 0 and it can also be carried out offline. We do not describe in detail how the user is authenticated when acquiring the initial certificate, an Internet banking account, showing a physical id in person, or some other form of identification may be used in this case. The user initiates the bootstrapping procedure by presenting its C1 certificate to the access network. Before the access network can grant access, it must verify the integrity of the certificate itself, the validity of the TTP H that has authorized the user, and check the revocation status of the certificate. In step 2 the access network checks TTP H's validity by contacting the global TTP system through its own TTP. The access network will receive a reply whether TTP H can be trusted. After receiving a positive reply, the access network contacts TTP H directly using its locator that is present in the PLA header, and checks the revocation status of the C1 certificate. If the certificate has not

been revoked, the access network grants access to the user as shown in step 3. As a result, the user is able to start communicating globally by using the C1 certificate in his PLA header.

When such an approach is used, the network access provider does not need to store any kind of data about the user to guarantee security and accountability. The responsibility of the access network provider will be limited to checking that the user has a valid certificate from a valid trusted third party. While the TTP validity check and certificate revocation check introduce some latency, they are not performed frequently. The access network can also cache the TTP validity information. Therefore, it does not need to perform a TTP validity check per every user if multiple users certified by the same TTP are accessing the network.

It is important to note that the C1 certificate is issued to the user and is not tied to the hardware. Therefore the user can utilize any available device, like a computer in the Internet cafe, for accessing the network with his credentials.

On the current Internet, billing is another reason why access network providers need to store users' personal information, and this requirement can also be removed with the help of PLA. As described before, in a mobile phone network billing is handled using a SIM card and IMSI number. With PLA, the certificate from a trusted third party to the user is equivalent to the SIM card. As the user enters a foreign network, it presents its own TTP certificate as a part of the authentication process. The owner of the network checks that it has an agreement with the TTP that has issued the certificate. The network owner also checks during step 2 that the user is able to pay for network access. If everything is in order, the user is granted access to the network, and the billing is handled by periodically sending necessary information to the user's TTP as shown in step 5.

Since PLA is based on cryptographic techniques, billing can be simply and securely implemented to prevent abuse by both the access network and the user. The PLA header itself allows flexible billing options on traffic or time basis since it contains a monotonically increasing sequence number and a timestamp. To provide proof that the user has been using network services, the access network simply sends a copy of the user's first and last packet to the TTP. Since all PLA packets are protected by the

signature signed by the user's private key, the access network can not forge any billing-related information to overcharge the user[1]. At the same time, the user can not deny sending packets to the access network. This prevents cheating in both ways.

Some large access network providers may want more control over traffic, which can be accomplished by issuing a separate traffic certificate (C2) to their users. In that case, the user's initial C1 certificate would only be used for bootstrapping, and all further data traffic sent by the user would use the C2 certificate in the PLA header. This would allow the network provider to filter out all data traffic that does not have its own certificate in the PLA header, potentially simplifying the network management. On the downside, the access network provider would need to securely store C1 – C2 certificate mappings for billing and accountability purposes since the original C1 certificate would not be present in the data traffic.

7.1.5 Catching culprits and revocation of certificates

The previous section described a bootstrapping procedure for accessing the network. This section and Figure 34 describes how misuse is handled in the system.

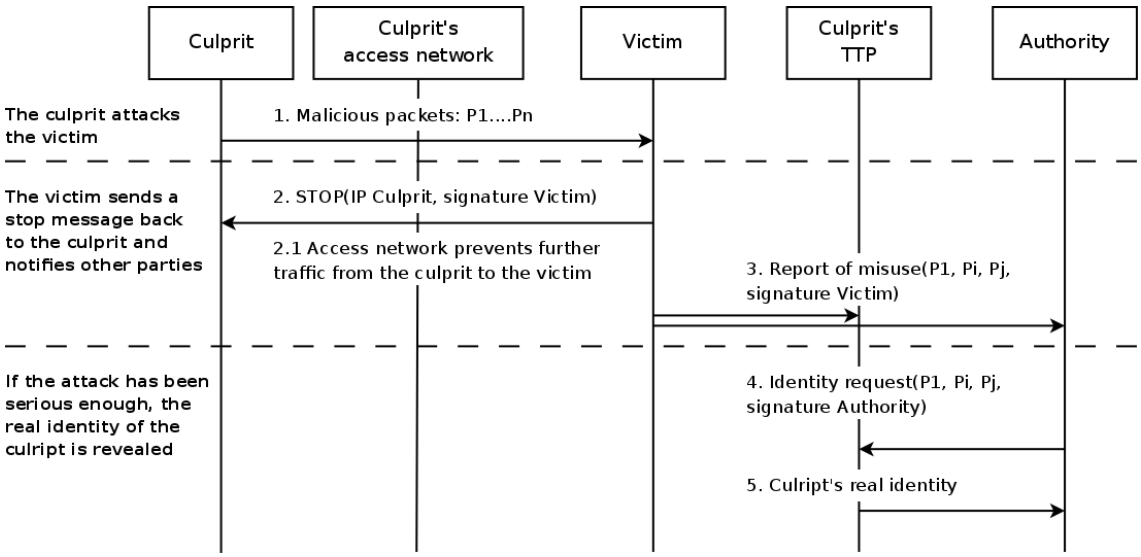


Figure 34: An example of catching culprits with PLA

In the first step of the example, the culprit initiates the attack by sending malicious packets P1...Pn to the victim. After the victim notices the attack, it caches at least some

1 The signature also protects the source IP address, therefore the access network cannot take a packet sent by the same user from a different network, and claim that it was sent from its network.

malicious packets and sends a *STOP* message back to the culprit. This message contains the culprit's IP address and victim's signature, and it signals to the culprit's access network and other routers on the path that the victim does not want to receive any further traffic from the culprit. The culprit's access network will take note of this *STOP* message and will prevent further traffic from the culprit to the victim. The *STOP* message prevents unwanted traffic even if just one router on the path adheres to it, therefore it is quite an efficient solution even if the return path is partially different. In step 3, the victim notifies the culprit's trusted third party (TTP) and authorities of the attack, providing them with cached packets P1, Pi, and Pj as a proof. These packets contain the TTP's cryptographic identity, the culprit's cryptographic identity and the culprit's signature making it impossible for the culprit to deny sending them. If the attack is considered to be serious enough, authorities will request the culprit's real identity from his TTP as described in steps 4 and 5. If the misuse has been a very serious one, the culprit's TTP may also revoke the culprit's certificate and notify the culprit's access network of the revocation. This would prevent the culprit from sending traffic to any destination.

This approach has several advantages. First, the victim is able to stop an attack on itself quickly. As long as there is a single router on the path which conforms to the *STOP* message, the victim will not receive further traffic from the culprit. It will also become more difficult for the victim to frame other parties for the attack. In many cases a lengthy investigation of the victim's logs is not necessary if cached packets already contain proof of the attack. It is also important to note the role of the access network. Since the culprit's identity is stored by the TTP, the access network does not need to store any personal details of the culprit or to be in a contact with authorities, it just needs to perform PLA validity checks on outgoing and incoming traffic and conform to received *STOP* messages.

As a result, there is no need for data retention by the access network provider. The proposed system is both much simpler and much more secure compared to the current solution, where access network providers store identity – IP address mappings for months or years.

The scheme outlined above is flexible and gives a lot of power to the user. However, it requires that the user's private key is kept safe and no other party can access it. Since the

user might lose his private key for various reasons, or his computer can be infected with viruses or malware, there should be a way to revoke the certificate tied to that key.

The revocation process is straightforward: first the user contacts the TTP that has issued the certificate, e.g., using a phone or other means, and informs it of the revocation. The exact details of are not covered here, since this step is equivalent of cancelling a lost SIM or credit card. The TTP in turn sends a revocation notice to access networks where the certificate has been recently used. This step is useful, since some recently used access networks may still trust the certificate. After receiving the revocation notice, these access networks will prevent further traffic with a revoked certificate. If some party tries to access a new network with the revoked certificate, then the revocation check during the bootstrapping phase will fail and the access will not be granted.

7.1.6 Wireless LAN authentication with PLA

PLA would be especially useful in wireless LAN authentication, since users change wireless networks frequently, and are often using a single network only for a limited period of time. Therefore, the authentication in wireless networks must be a flexible and lightweight process. For example, while currently most larger airports provide Internet access through a wireless LAN, it is usually not free and in most cases users are required to use a credit card to handle the authentication and billing. This creates several problems; e.g., the user's personal details are stored by the access network provider, the access network provider has an opportunity to overcharge the user or the user can try to evade the charges. With PLA, the situation would become much simpler, the users could authenticate with a certificate from the TTP that has an indirect trust relationship with the wireless LAN provider at the airport.

Another use case would be wireless LAN network at a university campus. The university would issue TTP certificates to its students and employees, and the network traffic sent with those certificates will be automatically allowed to the wireless LAN network. Such a system would not need a complicated authentication solution and users would not need to perform a time-consuming sign-in process when they want to access the network. The system can also be easily extensible, for example, the wireless LAN base stations can be configured to accept TTP certificates issued by other universities, also allowing their students and employees to utilize the wireless network.

During the authentication process PLA can also be used to secure initial link layer network discovery and bootstrapping messages. Therefore, the user authentication is performed in a transparent manner before the client even receives an IP address, which provides extra security to the wireless network. The authentication would also work even if PLA is only used during the bootstrapping phase. After the bootstrapping has been completed, the mobile node and the base station will generate a symmetric session key to secure the rest of the traffic. Basically, a PLA-based authentication system would be simpler, faster, and more efficient than currently widely used EAP-based authentication solutions.

7.1.7 Analysis

The current Internet is inflexible since the access network provider has two tasks. First, it must convey traffic on behalf of its users, and secondly, it must store users' personal details and bind them to IP addresses. These two tasks are very different in nature, and it is not sensible to always manage them with the same entity. The PLA-based network architecture uses principles from cellular networks; the user should be able to utilize multiple physical networks using a single contract. However, we extend this principle on the Internet scale using PLA as the enabling technology. With the proposed system, the access network provider can concentrate on its fundamental task, to provide telecommunication services to customers. Billing and mapping between cryptographic and real identities would be handled by TTPs. While some larger access network providers can offer TTP services, such services can also be offered by other parties, such as banks or credit card companies.

From a security point of view, the biggest difference is that under the current system the access network provider is practically responsible for its users' traffic. In a case of misuse, the provider must be able to point to the guilty user through data retention mechanisms. By using per-packet cryptographic techniques and trusted third parties, PLA transfers the responsibility from the access network provider to the user and TTP. The access network is just responsible for performing PLA-related checks: to verify that the sent packet is valid, has been sent by the authorized entity, and is not delayed or duplicated. If the packet passes these checks, then the access network provider is not responsible for the consequences, even if the packet is used for malicious purposes. As

outlined in a previous section, the authorities can determine the user's real identity without any additional co-operation from the access network provider.

Naturally, no system is completely secure. While the PLA ties traffic to cryptographic identities, these identities might leak to a hostile party, for example if the computer has been compromised. Therefore, the purpose of network accountability, whether it is accomplished by PLA or by a traditional data retention mechanism, is to provide a clue towards the origin of the attack. Evidence provided by such accountability may not always be decisive in a legal sense. In a case of key leakage, using PLA may not be able to reveal the real culprits behind the attack, but PLA limits the scope of the attack, since routers can easily block the traffic sent with a compromised cryptographic identity and the misused user's certificate can be revoked.

PLA would also make it much more difficult for hostile parties to frame users for imaginary attacks. If the victim of the attack can not present packets signed by a user's private key as evidence, then the user will be innocent by default. On the current Internet it is possible for the service provider to frame the user by forging its own logs and claiming that an attack originated from the user's IP address.

There is always a trade-off between security and privacy: if complete privacy is allowed, then most of the attacks on the network can be stopped to a certain degree, but it will not be possible to catch perpetrators behind those attack. PLA actually improves users' privacy through the use of multiple cryptographic identities acting as pseudonyms. If PLA is also used for billing, the access network provider does not need to know the user's real identity, and there is no single entity in the network that can directly tie the user-generated traffic to his real identity. This offers a good balance between security and privacy; as long as users are behaving well, they can maintain their privacy. However, real identities of users engaging in malicious behaviour can be retrieved with the help of trusted third parties.

The proposed scheme would also increase the competition between access network providers since the barrier of entry to market will be significantly lower for two reasons. First, access network providers do not need to store users' personal details for security purposes, and additionally, PLA and the TTP system can be utilized for handling billing in the network. Basically this would decouple the ownership of the transmission

medium from billing and higher level services, allowing TTPs and access network providers to compete for customers independently of each other. Such a method would allow the emergence of very lightweight access network providers, which do not need to directly identify their users at all. For example, a person could share their Internet connection with neighbours or passers by for a nominal fee, and communal networks would once again become a viable option.

The solution relies on indirect trust relationships between operators and TTPs. Since it is not feasible to require all parties to trust in each other for billing purposes, the amount of trust hops between TTPs can be a deciding factor whether to trust them to do the billing. For example, the access network can decide that it will accept customers from its own TTP and from TTPs that form a direct trust relationship with the access network's TTP.

The latency of the bootstrapping phase is also low. The access network needs to consult its own TTP and the user's TTP before the access can be granted and under normal circumstances such checks would take a few seconds of time. Actually, using a PLA-based certificate mechanism would significantly reduce the overall latency compared to current commonly used solutions where the user is authenticated by manually entered passwords or credit card information. For example in case of a wireless LAN, the mutual authentication can be performed during the network discovery and initial bootstrapping messages, and the client is already authenticated when it gets an IP address.

7.1.8 Potential weaknesses in the solution

Using PLA for Internet-wide roaming and billing is based on trust relationships between access network providers, trusted third parties and users. Potential situations where various parties may try to abuse the system with corresponding solutions are presented below.

User

- Trying to evade charges for using the access network. The PLA header's signature effectively prevents this, the operator will possess packets signed by the user's private key.

- Engaging in illegal behaviour. Malicious traffic can be filtered based on cryptographic identities. The authorities can also determine the user's real identity with the help of the trusted third party. Therefore, the user can be prosecuted for his actions.

Access network provider

- Overcharging the user. The PLA header's signature effectively prevents this.
- Not performing PLA-related checks for outgoing packets, or not conforming to received *STOP* messages[1]. Such behaviour will eventually be noticed by the parent operator of the access network provider since there will be a large amount of invalid or undesired packets originating from the access network. Therefore, the parent operator can take actions against the access network provider by charging higher fees for transit or limiting the bandwidth until the access network provider will start to perform the necessary checks.
- Not performing necessary checks during the bootstrapping procedure, i.e., not checking the revocation or balance status of the user's certificate. Such behaviour would have the following consequences on security and billing. First, if the user manages to cause damage with a revoked certificate, then the access network provider could also be held responsible, since it allowed the user to send traffic to the network without a valid certificate. Secondly, the access network provider will not be compensated by the TTP for the user's traffic since necessary revocation and balance checks were not performed.

Trusted third party

- Not compensating access network providers for the network usage. Access network providers would break the contract with the TTP and would not allow users of such a TTP to access their networks anymore.
- Not revoking certificates of malicious users. In such cases other trusted third parties in the network may lose trust in the TTP in question. Therefore the TTP's customers would not be able to access as many networks as before, and in the long run the TTP in question would lose customers.

It can be seen from these examples that all parties have strong incentives to behave according to the rules of the system. The user must not engage in illegal behaviour in

1 *STOP* messages can be independently verified by any router on the path, therefore a co-operation of the access network is not strictly necessary to stop the undesired traffic to the receiver.

order to retain the ability to use the network. The access network provider must perform necessary checks during the authentication and for outgoing data packets in order to get compensated for the user's traffic. The trusted third party must co-operate with authorities and other TTPs in order to retain its customers.

The existence of three distinct parties and strong cryptographic identities and signatures makes it easier to detect the abuse in the system and resolve potential disagreements. Let us consider a previous example where the user accesses the wireless LAN network at the airport, and the credit card is used for the user authentication and billing. If the network tries to overcharge the user, or the user tries to evade charges, then the process of resolving disagreements on the current Internet is complex and time consuming. It will require a lot of manual work, since both the user and the access network provider must try to prove their points with evidence, like IP address mappings and traffic logs, which are not inherently strong. Therefore, the evidence must be thoroughly investigated by an actual person. However, if PLA is used, the situation becomes much more simple. The access network provider simply sends a copy of the user's first and last sent packet to the user's TTP and billing will be handled based on these packets as was discussed in Section 7.1.4. The access network can not claim any additional compensation and at the same time the user can not deny sending that packet since it was signed by his private key. This makes the process of resolving disagreements more automatic and reduces the overhead of all involved parties: the user, the access network provider, and the entity that is responsible for billing.

7.2 Securing media independent handover (MIH)

Media independent handover (MIH) [76] is a framework to enable handovers in IP networks between various link-layer access technologies. For example, MIH can be used to transfer the connection seamlessly from the 3G cellular network to the wireless LAN, or between two different wireless LAN networks. However, MIH does not define concrete security measures to protect such handovers.

In order for MIH to be useful, both the end user and point of access of the new network should mutually authenticate each other. Otherwise, the user could utilize the network's resources without permission, or a bogus point of access could trick users to initiate the handover to itself. The handover signalling should also be tamper-proof.

A method for securing MIH with PLA is described in [94]. In the example presented in Figure 35 it is not compulsory to use PLA for all traffic, protecting the signalling traffic with PLA is enough.

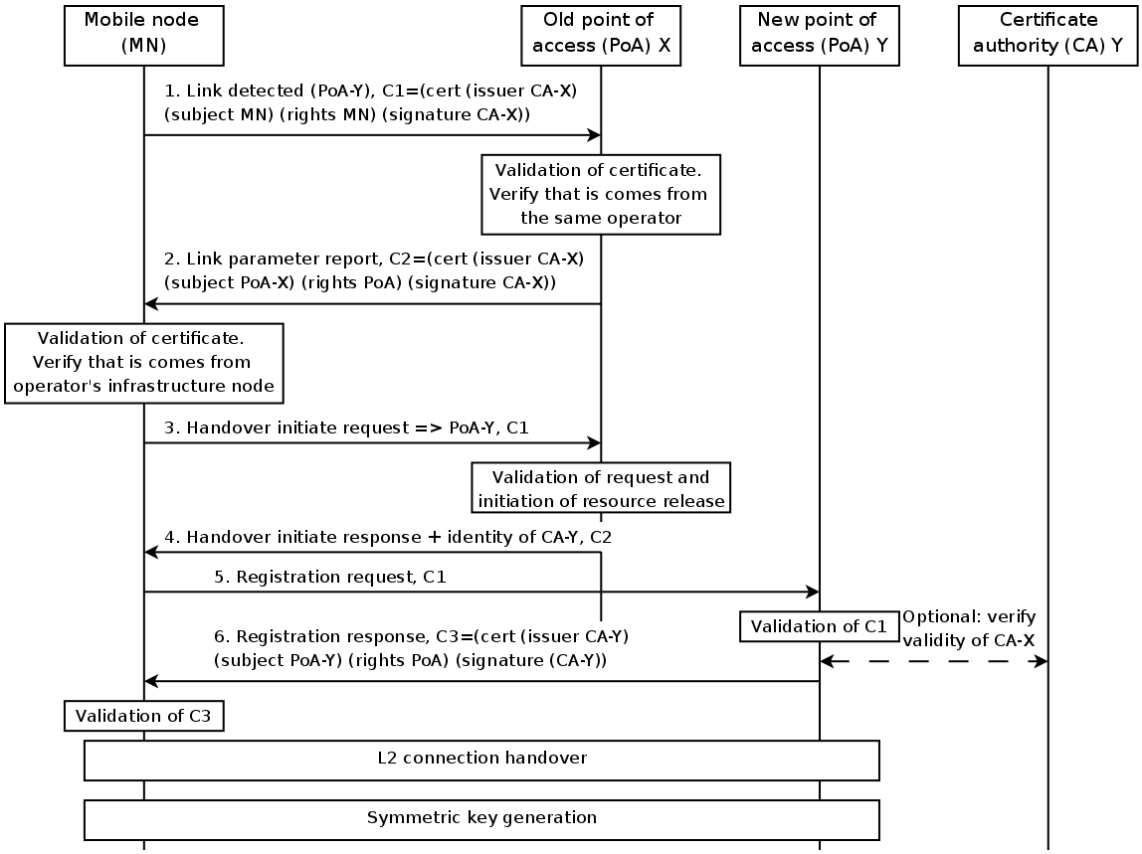


Figure 35: Using PLA to secure media independent handover (MIH) [94]

In this example, the mobile node (MN) is connected to network X and initiates the handover to network Y, which has an agreement with X. The example assumes a normal PLA architecture, where the mobile node already possesses the certificate from the certificate authority X, which basically is the trusted third party (TTP). The rights field inside a TTP certificate are used to distinguish the network access points from ordinary users. All signalling messages in the example are protected by PLA.

In the first step, the mobile node initiates the handover by sending a MIH message to the existing point of access (PoA X). Since this message is protected by PLA, it contains a certificate from the CA-X to the mobile node. PoA X verifies the message, and checks that the mobile node has been certified by the same operator. In step 2 PoA X replies with a link parameter report message. The MN can now verify that the reply is valid, and has arrived from the infrastructure node of the same operator.

In steps 3 and 4 the handover is initiated. It is assumed that operators, which allow handovers between themselves, already know the cryptographic identities of each others' TTPs. Such an identity is given to the mobile node in step 4. In step 5 the new point of access verifies MN's certificate and optionally verifies the identity of CA-X if it is not already trusted. Finally, in step 6 the mobile node can verify that the new access point has really been certified by CA-Y. Such a check prevents MN from making the handover to the bogus point of access. After the handover has been completed, there is an option to generate session keys to encrypt the wireless traffic.

Using PLA to secure MIH signalling messages offers good security without a need for extra messaging and external authentication. For example, in steps 1 and 2 both the mobile node and the point of access authenticate each other directly based on contents of the PLA header, while the similar authentication between the MN and the point of access (PoA Y) happens in later steps. The authentication occurs as a part of the normal MIH signalling, and does not require any extra packets to be transmitted. Furthermore, there is no need to contact an external authentication server. PLA also protects signalling messages from being tampered with.

8. PLA and future data-oriented networks

This chapter explores the applicability of PLA for securing future data-oriented networking approaches.

Data-oriented networking aims to solve the inefficiency of the current host- and message-oriented Internet. Instead of making connections to individual hosts, operations are performed to data. In practice data-oriented networks are based on the publish/subscribe principle and work as follows. First, the publisher publishes some data to the network, usually identified by a flat label identifier. Then interested parties may subscribe to the identifier to receive the data item. If there are multiple subscribers for the same data, multicast is automatically used for an efficient data delivery. The rendezvous service acts as a middleman between publishers and subscribers.

Identifiers used within data-oriented networks are often self-certified in nature, they are derived from cryptographic identities in order to give subscribers a mechanism to easily verify that the data item is valid and has originated from the correct publisher. Since data items are uniquely named, caching within the network becomes straightforward. Caches can be added to the network dynamically, and because of the self-certifying nature of identifiers, the subscriber do not need to care whether the data has been sent by the original publisher or the cache, the subscriber can verify the authenticity of the data in any case.

Data-oriented networks adhere to the trust-to-trust principle [29], which argues that users on the Internet are interested in communicating with trusted entities, instead of arbitrary hosts. Data-oriented approaches also aim to transfer some control from the initiators of the connection to the receivers since users request relevant data instead of contacting other hosts directly.

Well-known data-oriented systems include Internet Indirection Infrastructure (i3) [102], Data-Oriented Network Architecture (DONA) [66], Routing on Flat Labels (ROFL) [18], Content Centric Networking (CCN) [56] and Publish-Subscribe Internet Routing Paradigm (PSIRP) [2]. PSIRP differs from other similar solutions since it aims to implement a publish/subscribe-based network from scratch, without relying on existing network-layer technologies such as the IP protocol.

8.1 PSIRP background

One example architecture for PSIRP is presented in [112]. The main components of the system are:

- *Namespace owners* manage namespaces for publication identifiers. The namespace owner authorizes publishers to use part of the namespace for their publications.
- *Publishers* create the actual publication, which in turn is delivered to *subscribers*.
- *Data sources* serve the actual publication in the network.
- *Scopes* control the dissemination of data. Scopes have trust relationships with data sources and rendezvous system.
- *Rendezvous system* is used by data sources and subscribers to register and subscribe to the publication.

In many cases the namespace owner, the publisher, and the data source are a single entity. However, for additional flexibility they may be separate. For example, the namespace owner may be a big newspaper, which would authorize writers to use part of the namespace for their articles. These articles would then be served by dedicated data sources.

On the network layer, PSIRP publications are identified by rendezvous identifiers (Rid), while scope identifiers (Sid) denote scopes, both of these identifiers are 256-bit flat labels. To access the data the subscriber must know both the Rid and Sid[1]. Rids and Sids have a <P:L> (public key:label) structure similar to DONA [66]. In the payload messages the label part of Rid contains a hash of the arbitrary label, while the rendezvous requests and subscription messages can include the original variable length label to enable dynamically generated content. Similarly to PLA, PSIRP also utilizes ECC-based keys, therefore unlike in DONA, the whole public key can be included in the identifier simplifying the system.

The rendezvous process goes as follows. At the beginning, the data source publishes the publication to the rendezvous system. The interested subscriber first sends a rendezvous request and receives the topological location of the data in reply from the rendezvous system. Next, the subscriber sends a separate subscription request towards the data

1 A separate mechanism is necessary to resolve human readable names into <Sid:Rid> pairs

source. Caches on the path can take note of this request, and serve the publication directly to the subscriber if it is already in cache.

8.2 Securing the PSIRP rendezvous process with PLA

As mentioned in Section 5.2, PLA is a natural alternative for securing data-oriented networks since it is based on cryptographic identifiers and signatures. This section discusses in more detail how PLA can be used to secure a data-oriented networking system. A more detailed discussion about various aspects of PSIRP security is available in [68].

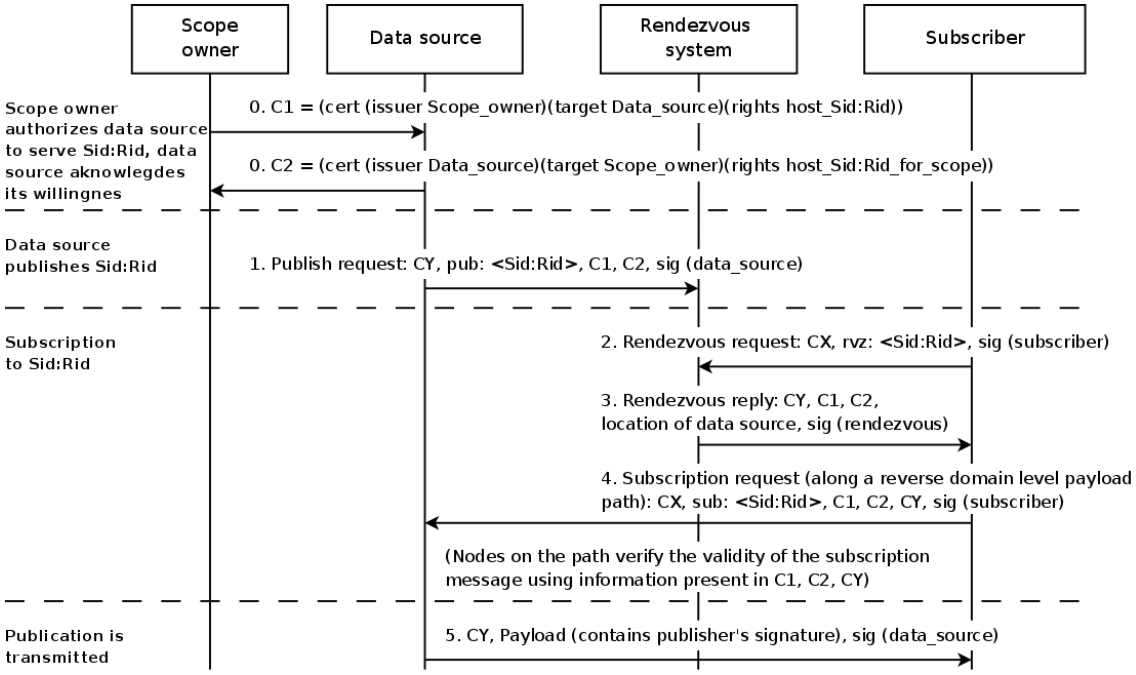


Figure 36: Securing rendezvous process with PLA in data-oriented networks [68]

Figure 36 shows an example of how PLA can be used to secure the rendezvous process, which covers the most important functions of the publish/subscribe networking.

When a node, for example a data source or subscriber, is bootstrapped, it will receive a certificate from the local access network, basically this certificate is equivalent to the PLA's TTP certificate. In the figure, CX denotes such a certificate from the access network to the subscriber and CY denotes a similar certificate to the data source.

In step 0, the scope authorizes the data source to serve the publication (<Sid:Rid>) with a C1 certificate, while the data source acknowledges its willingness with a C2

certificate. The C2 certificate is required since without it a hostile scope could induce load to the data source by claiming that the publication can be found from the target data source.

In the first step, the data source sends a publication advertisement concerning <Sid:Rid> to the rendezvous system. This message contains the data source's certificate from the access network (CY) along with C1 and C2 certificates. The whole message is protected by the data source's signature. In the next steps, the subscriber initiates the subscription process and the rendezvous system returns a data source's location within the network along with all relevant certificates.

The final subscription message is sent in step 4. This message contains CX, CY, C1 and C2 certificates, which together with the <P:L> structure of identifiers offer a proof to intermediate nodes that this message is valid. Any node on the path can verify that the scope and data source have authorized each other to serve <Sid:Rid>, and the data source is really a valid entity in the network.

Finally, in step 5, the publication is delivered from the data source to the subscriber. The payload delivery may be optionally protected by the data source's signature using PLA.

For some use cases, it is important to have access control for subscribers for a certain scope, therefore unauthorized subscription messages must not reach the data source. If it is also important to keep the location of data sources confidential, additional checks can be implemented in the rendezvous system.

The example presented in Figure 36 can be augmented to support the access control. In this case the C1 certificate would contain different rights, denoting that access control is required and therefore a subscriber must possess a separate certificate (C3) from the scope. This C3 certificate acts as a capability. In step 2 the rendezvous request will go all the way to the scope, and the subscriber must authenticate itself using a password or other means. Such authentication may require additional signalling that is not shown in the example. After a successful authentication, the subscriber will receive the above-mentioned C3 certificate from the scope, and will include it to the subscription message in step 4.

In step 4 intermediate nodes will see from the C1 certificate that the access control check is present, and will verify that the subscription message will contain a valid C3 certificate. The issuer field of the C3 certificate must match the public key present in Sid, while the subject field must match the subject field of the CX certificate. This way a hostile subscriber can not reuse a valid C3 certificate.

8.3 Analysis

Basic PLA security features, such as integrity protection, detecting duplicated and delayed traffic, removal of malicious users from the network through the TTP mechanisms, and strong accountability also apply to data-oriented networks. No significant modification of PLA is necessary. While the above-mentioned example applies to PSIRP, PLA can also be used to secure other kinds of data-oriented networks.

Furthermore, PLA allows nodes on the path to independently verify the validity of rendezvous and subscription messages, and distinguish those messages from the data traffic. In this case, validity means that subscriber and data source are valid entities in the network, the subscriber wants to receive the publication, the data source is willing to serve the publication and is authorized by the scope. Independently verifiable per-subscriber capabilities are also supported. Invalid messages can then be dropped immediately.

PLA's cryptographic identities and signatures allow the network to differentiate users and traffic easily. Therefore the access network can limit the amount of subscription messages sent by a single node within a given timeframe, and subscription messages can also be limited per destination. Since the size of the subscription message with all certificates is a few kilobits at most, this effectively prevents severe DoS attacks against the data sources. Subscription messages can also be prioritized higher than the data traffic. Therefore in a case of congestion, subscriptions will get through at the expense of bulk data transfer, improving the overall availability of the network. A similar limitation can be enforced for rendezvous messages to prevent hostile nodes from overloading rendezvous nodes.

Since PSIRP already includes cryptographic keys as a part of identifiers, the extra bandwidth overhead of certificates and PLA is lower, since keys do not need to be

included twice. For example, when the scope authorizes the data source to host <Sid:Rid>, the certificate and the PLA header does not need to explicitly contain the scope's public key since it is already a part of the Sid.

In a publish/subscribe approach such as PSIRP, the rendezvous system is a crucial part of architecture since basic publish and subscribe operations depend on it. Hence, the rendezvous should be able to function properly in all possible situations. Using the above-mentioned principles, PLA can also be used to secure rendezvous-related control traffic.

9. Deployment of PLA

This chapter explores incentives and challenges related to the large-scale deployment of PLA. Real-world deployment of new networking protocols on the Internet scale is an extremely challenging task. For example, the IPv6 specification was completed over 10 years ago, and in spite of a serious IPv4 address space shortage a wide-spread deployment of IPv6 still has not happened yet. Host Identity Protocol (HIP) is another example of a promising technology that has been under development for over a decade, but has not yet been widely deployed.

In order for a new protocol to be deployable, incremental deployment should be possible in a flexible way, since it is not realistic to assume that everyone is willing to adopt the new technology at once. Furthermore, the benefits of the new solution should significantly outweigh drawbacks to justify the cost of transition.

9.1 Incentives for PLA deployment

PLA offers several benefits to the network operators and users. It provides integrity protection, protection against DoS attacks, ability to stop unwanted traffic, QoS possibilities, and it also offers flexible billing options and the strong accountability without extensive data retention.

According to the analyses presented in Section 5.3.3, using PLA globally in every router for every packet would consume about 40 GWh of energy annually. For comparison, activities related to unsolicited e-mail (Spam) consume 33 TWh of energy annually according to McAfee estimates [75]. Therefore, even if PLA could reduce the amount of Spam by 10%, it would produce massive energy savings.

9.1.1 Availability and DoS protection

PLA provides availability on the network layer by offering an integrity protection and strong authentication. This allows network operators to quickly detect and block invalid or malicious traffic.

PLA improves availability and prevents unwanted traffic in multiple ways. First, the unwanted traffic can be blocked based on the sender's cryptographic identity, the destination or the TTP that has authorized the sender. Additionally, the malicious node can be removed from the network by revoking or not renewing its TTP certificate; this limits the time frame when the malicious node can cause the load on the network. Finally, strong accountability provided by PLA allows culprits to be caught, preventing them from carrying out future attacks.

9.1.2 Quality of service and traffic prioritization possibilities

The current Internet lacks support for fine-grained packet prioritization and therefore it can not be used for conveying really critical messages. In a case of DoS attack or congestion, there is no guarantee that these important messages will go through. As a result, several countries have built dedicated networks for authorities and emergency services.

The TTP certificate mechanism used by PLA offers a great flexibility for prioritizing traffic and implementing the quality of service (QoS) support. The rights field within the TTP certificate can be used to denote a traffic priority. Routers would allocate a small amount of the bandwidth to high priority packets, and in case of congestion they would drop low priority packets first.

Such an approach can be used in multiple ways. First, the higher priority can be sold to users as an extra service. Furthermore, the operator can reserve highest priority to itself and for authorities. This would allow transmission of critical messages, such as network control messages or emergency announcement, in virtually all situations, even if the network is under a severe load or a DoS attack. Therefore, such a system would reduce the need for having a separate dedicated network infrastructure for authorities and emergency use.

9.1.3 User authentication, accountability and billing

Currently operators face increasing requirements from authorities regarding data retention and user identification. These requirements significantly increase the costs and reduce the flexibility of operators. For example, the current EU legislation requires

operators to provide accountability through data retention. Such a requirement raises the barrier of entry and effectively prevents lightweight operators, which only want to convey traffic, from functioning.

As mentioned in Section 7.1, PLA provides strong data-origin authentication and accountability, which eliminates the need for data retention by operators. This reduces the costs of operators and adds flexibility to the network. PLA allows operators to concentrate on their fundamental task, to convey traffic on behalf of their users, and leave other tasks, such as the user identification and authorization, to other parties. The basic responsibility of operators would be limited to performing PLA checks for outgoing traffic. In a case of misuse, the real identity of the culprit can be determined with the help of the TTP; co-operation of the operator is not required.

Since PLA offers non-repudiation, it can be used for flexible billing schemes. For example, the user could be charged based on the amount of sent packets or bandwidth used. Tariffs may also depend on time and operators could offer a cheaper bandwidth at night. PLA's certificate mechanism together with the sender's signature guarantee that such a billing scheme is secure. The user can not evade charges since his signature is present in every sent packet, and the operator can not impose additional charges on the user since it can not forge the user's signature. The non-reputability also to some extent allows an automatic resolution of disagreements between various parties, such as operators and end-users.

9.2 Challenges for PLA deployment

The biggest challenge for PLA deployment is the cost associated with the additional hardware for cryptographic operations. While the cost of the actual accelerator ASIC is not very high [53], significant resources are needed to develop, evaluate and deploy new routers and computers on the Internet scale. These costs mostly affect the early PLA deployment, as the production volumes of the necessary hardware grow, the costs will be greatly reduced. Furthermore, there is a tendency to support cryptographic operations through dedicated hardware in general purpose products. Even though the ECC algorithm is not widely supported, VIA and Intel processors accelerate some cryptographic algorithms in hardware [111][52], and the Symbian^3 [103] operating

system developed for mobile phones provides support for hardware acceleration of cryptographic algorithms.

There are also additional challenges. Some of the current Internet routers often block packets that use less known IP extension headers, therefore there is a risk that also PLA packets will be blocked. A simple configuration modification is required for these routers to let PLA packets go through.

Some parties can view PLA's advantages as disadvantages and may hinder PLA's deployment. For example, since PLA offers flexible and secure billing and user management, it lowers the barrier of entry for the operators and increases competition. Therefore, some dominant operators may view PLA as a threat to their position. Some countries may also oppose the idea of the Internet-wide roaming that was presented in Section 7.1, since it allows users to be authorized by TTPs residing in a foreign country

Since PLA is based on a strong cryptographic techniques, identity spoofing and impersonation becomes much more difficult to achieve. This might annoy intelligence agencies and other entities.

9.3 Wireless PLA

Wireless PLA [3] combines the principles of PLA, such as integrity protection, with hash chain-based solutions to reduce computational overhead of PLA. In order to fully use hash chains to protect packets, hash chain anchors must first be authenticated. WPLA accomplishes this by using a full PLA header with initialization packets.

Afterwards, WPLA offers two modes of operation. In the lightweight verification mode only hash chains are used to secure the traffic. In the adaptive mode, WPLA includes both the PLA header and the hash elements in its header. Therefore, the intermediate nodes can decide whether to use hash chains or signatures to verify the traffic. While the adaptive mode increases the bandwidth overhead, it is very flexible. For example, the downside of hash chain-based security solutions is path dependency. However, if some packets protected by adaptive WPLA take another path in the network, intermediate nodes can still verify their authenticity by using PLA's public key signatures.

WPLA also provides support for confidentiality. In this case the sender creates a separate hash chain, which is used as a keying material for symmetric key encryption. The sender periodically changes the encryption key by using the next element from the chain. Afterwards, the sender reveals the previously used symmetric key to the receiver by encrypting it with the receiver's public key. When the receiver receives an encrypted packet, it can verify their integrity using either the hash chain or signature method. However, the receiver can not decrypt those packets until the sender reveals the encryption key used.

While wireless PLA is a more complicated solution, it does not require dedicated hardware for performing cryptographic operations. Therefore it is a suitable alternative to the full PLA in the early adoption phase.

9.4 Migration path to PLA

PLA does not require support from every node in the network, making incremental support possible as described in Section 5.1.1.

PLA provides benefits to several parties, therefore migration paths originating from both the end users or small operators, and large operators are possible. For example, the flexible user authentication offered by PLA would be especially useful for small access network providers. It also allows companies that already manage user identities, such as banks and credit card companies, to enter a new field and manage user identities for the access network providers. In this case migration towards PLA would start from these companies and their users, and eventually grow into the global system envisioned in Section 7.1.

Larger operators could deploy PLA inside core networks to utilize QoS and other possibilities offered by PLA. For example, PLA can be used to implement a resilient control plane for the Internet without building a separate network. Afterwards, PLA can be gradually deployed to the forwarding plane.

Overall, it is reasonable to assume that PLA needs to achieve a critical mass after which widespread adoption would happen naturally. For example, since PLA simplifies the network security problem and saves resources in various ways, such as stopping attacks

quickly, a major operator that is already using PLA would save resources if its customers would also adopt PLA. Therefore, the major operator could demand higher charges from its customer operators and users if they are not using PLA. Eventually, after the critical mass is reached, everybody would need to adopt PLA in order to get a network connectivity at a reasonable price.

10. Discussion and future work

This chapter discusses various aspects of PLA and outlines future work. Overall, PLA offers strong technical-level mechanisms for securing the network infrastructure. PLA provides strong integrity protection and accountability and allows every packet to be independently verified by any node in the network. This work shows that mechanisms offered by PLA are technically solid, can be used in real-life environments, and also benefit higher-layer protocols and applications.

Widespread adoption of PLA would have several implications on everyday life. The increased security would make the network more efficient, and decrease the amount of Spam and other garbage on the Internet. Users would be able to use transparent, simple and secure authentication methods that do not require excessive manual intervention. For example, the user possessing a valid certificate would be able to log into wireless networks in all parts of the world without entering passwords or credit card information. Usage of cryptographic identities also makes identity management and delegation much easier. For example, it would be easy to delegate rights to use a computer or the network connection to another person in a secure manner.

With a strong technical-level solution in place, the next step is to create efficient and flexible policies to take full advantage of the security features offered by PLA. While this work has also covered some policy-level issues, i.e., necessary types of TTP certificates, an example of scalable TTP trust architecture for the Internet, and examples of using PLA at higher layers, there is still a lot of work to be done in this area. For example, in some cases alternative trust management solutions, such as OpenID [85], can be used instead of the example TTP architecture.

Since PLA significantly changes the security and accountability properties of the Internet, legislation should also be updated to take this into account. For example, legislation should reflect that operators do not need to store information about the IP addresses used to achieve good accountability in the PLA-based network architecture.

As more and more personal information is transmitted over the Internet, privacy on the network becomes increasingly important. This work has outlined several mechanisms for achieving a good anonymity and privacy in a PLA-based network architecture, such

as the use of pseudonyms and the separation of user identity management and network provision. However, the security and accountability features of PLA can be easily misused, therefore a great deal of care must be used when creating policy- and legislation-level mechanisms in order to achieve good privacy in the whole system. For example, the user should be able to use multiple cryptographic identities in parallel, access network providers should not be required to identify their users, the user should be able to choose the TTP provider independently from the access network, and the TTPs themselves must be well secured.

The TTP certificate system used by PLA is flexible and there are several options regarding certificate validity times. One possible way to further improve security is either to make normal TTP certificates revocable or reduce their validity time. The first alternative would be to keep the current validity time of hours, but make certificates revocable (thus other routers should verify the revocation status of the certificate). A second alternative would be to use irrevocable traffic certificates with a validity time of only a few minutes. Since the validity time is so short in this case, good security can be achieved without support for revocation of such certificates. The bandwidth required for requesting a new certificate and checking the revocation status of a certificate is roughly equivalent, in both cases two packets are sent containing a TTP certificate and some additional information. In the first case, nodes with whom the user is communicating must contact his TTP to check the revocation status of the user's TTP certificate, while in the second case, the user must contact his TTP frequently to renew his certificate. Thus overall, the load on TTPs is quite similar. However, in the first case where the TTP certificate has a longer validity time and can be revoked, the user can more easily attempt to attack his TTP by sending a large amount of garbage data to a large number of different recipients. Those recipients in turn would verify the validity of the user's TTP certificate and thus create a load on the user's TTP.

In order to offer protection against replay attacks, PLA-enabled routers should check the timestamps of packets and only forward packets which have not been significantly delayed. Thus, PLA assumes that all participating nodes have their clocks somehow synchronized. How strict a synchronization is necessary depends on the policy and applications used. For example in a military network clocks should be strictly synchronized, but in a normal Internet differences of a few seconds would be acceptable. While the clock synchronization can be managed easily with the Network

Time Protocol (NTP) [84] or a similar solution, in certain cases the clock synchronization requirement may cause some problems. A node might have an incorrect clock and it will not be able to send data to the network, and therefore will also be unable to synchronize its clock. One way to overcome this problem is to allow limited traffic using certain protocols, such as NTP, even if the sending node does not have a properly synchronized clock. Use of encapsulation would also mitigate this problem since only the router in the access network would see the incorrect timestamp. Afterwards, the router would encapsulate the packet in its own PLA header containing a valid timestamp.

There are further ways to reduce the PLA's overhead. For example, an idea for selecting the cryptographic algorithm and the key length at runtime depending on the application is presented in [105]. The same idea could be used by PLA especially in a resource constrained networks. When a higher level of security is required, a sending node would apply a stronger PLA signature, and a weaker signature could be used when a lower level security is sufficient. Since such a mechanism requires that all parties support all possible cryptographic alternatives, it is not really suitable for the Internet but could be used in closed networks.

10.1 Additional uses of PLA

The strong security properties offered by PLA can also be used for various other tasks. Since the PLA header already includes information for calculating the sender's public key, PLA is suitable to be used together with Cryptographically Generated Addresses (CGA). A PLA packet along with an CGA IP address would provide proof that the sender has the right to use his IP address. Actually, the principles of CGA addresses can be taken further. Including a subnet prefix in the CGA address calculation would tie the created address to the subnet, and offer proof that the user has the right to use a specific subnet. This would benefit mobility management schemes since sending a mobility update with a bogus subnet prefix would become much more difficult. Furthermore, the MAC address of the network interface could also be derived from the public key. These mechanisms would simplify the network attachment process since there would not be a need for a separate IP address negotiation when entering the network, and prevent the MAC and IP address spoofing.

To provide stronger protection against phishing attacks, cryptographic identities used by PLA could be stored in DNS records, using the idea of HIP DNS extensions [83]. Therefore, during the DNS lookup a client would receive a server's public key and could immediately verify whether the server is authentic. The TTP certificate mechanism provides an additional way to verify that the server's public key is valid and authentic. The same principle can also be applied to subnets; when querying the subnet through the DNS, the client would receive the public key of the router that manages the subnet. This information can then be used to verify the authenticity of the network management and control messages.

The cryptographic hardware accelerator utilized by PLA can also be used to perform various tasks on the operating system and application levels. For example, all documents such as web pages and spreadsheets could be automatically signed by their creators, and the file system could automatically sign all files stored on the hard drive. Basically, a traditional cyclic redundancy check (CRC) mechanism could be replaced by strong cryptographic signatures improving overall security. The performance would not be a problem in this case, since applications and operating systems operate on larger data blocks than IP networks. For example, if the Hardcopy accelerator that can perform 850,000 verifications per second would be applied to verifying a file system with a standard 4 kilobyte block size, it would be able to verify almost 4 GB of data per second. Such a speed is significantly higher than the performance of the hard drives.

10.2 Future work

As mentioned before, PLA requires efficient policies to utilize it fully. Important policy-related questions include: how should the router handle packets whose senders have been authorized by unknown TTPs? How much traffic should routers reserve for different TTP certificate types? What percentage of packets should be checked at routers, for a reasonable security? How quickly should potential hostile nodes be removed from the network? The answers to these question will depend on the network's usage, for example, civil and military networks would have very different security policies, and other circumstances like whether the network is under attack.

The deployment process of PLA must be studied further. For example, standardization of PLA's core functionality would help future adoption. Furthermore, the future

deployment would benefit from the large-scale validation of the system taking various network environments into account, including wireless and sensor networks. It would also be beneficial to perform a more rigorous security analysis on potential higher layer protocols that utilize PLA.

The current PLA implementation is a proof of concept one, and can be significantly optimized. The hardware accelerator used by PLA can be improved by transferring the design to more modern and faster FPGAs. The current performance and power consumption estimates are based on FPGA and Hardcopy ASIC simulations. An ASIC design on a more modern manufacturing process would offer more accurate information about the cost, performance and power consumption of the cryptographic accelerator.

Currently, a full TTP certificate contains a TTP public key and its locator which is a 128-bit IPv6 address. Such a separate locator is necessary for contacting the TTP in the current solution. A more efficient solution could utilize a DHT overlay network. The public key of the TTP could be used as its DHT key and a DHT network would be used for contacting TTPs, decreasing the space requirements of the TTP certificate. The downside of such a solution would be increased latency when contacting TTPs.

PLA could also be combined with the Host Identity Protocol (HIP). In a such case PLA would guarantee the integrity of packets while HIP would provide confidentiality and support for mobility and multihoming. Combining HIP with PLA could also simplify HIP. The 4-way base exchange mechanism of HIP will not be necessary, since the PLA header already authenticates the initiator. PLA and HIP headers could also be merged into a single header to reduce bandwidth overhead.

11. Conclusions

The security problems of the Internet are a result of its design, which lacked built-in security mechanisms and assumed that all network nodes are benevolent. In order to really improve the security of the Internet a novel network-layer solution should be sought, using higher layer security mechanisms to patch up the drawbacks of the network layer is not enough.

This thesis described the Packet Level Authentication architecture and investigated its applications. PLA is a novel way to secure the network infrastructure by providing availability on the network layer. The fundamental principle of PLA is that every transmitted packet in the network has an undeniable owner and every node within the network can verify the authenticity and validity of the packet independently. PLA allows any node in the network to verify that a passing packet has not been modified, has not been delayed and is not a duplicate of another packet. This gives PLA an advantage compared to traditional end-to-end security solutions in which only the end points of the connection can verify the validity of the packet.

PLA is based on public key cryptography and it protects every packet in the network by a cryptographic signature. PLA is compatible with existing IP networks and can be used together with other security solutions, such as HIP or IPSec; PLA is also a natural security solutions for future data-oriented networking approaches.

The main benefit of PLA is that various attacks against the network and its users can be detected immediately. Hence, attacks can be confined before they can cause significant damage to the network. PLA also provides strong accountability that allows removal of malicious nodes from the network, and offers a way to catch culprits behind attacks without extensive data retention by operators. Despite the accountability features of PLA, users are able to maintain their privacy through the use of pseudonyms.

Public key cryptographic operations consume a significant amount of computational resources, but this work shows that PLA is scalable to high-speed networks and low-power devices as long as a dedicated hardware accelerator is used for cryptographic operations. The energy overhead of PLA is very insignificant compared to the overall energy cost of wired and wireless networking. Since various security problems, such as

unsolicited e-mail and denial-of-service attacks, waste a large amount of energy and other resources, PLA often saves overall energy since it is able to stop attacks in a more efficient way.

The strong security mechanisms offered by PLA can also be utilized to enhance security at higher network layers. PLA can be used as an enabling technology to implement a flexible user authentication and Internet-wide roaming, and a system where incoming connections are denied by default without an explicit authorization.

Overall, this work has shown that PLA is a technically valid solution and is useful for several purposes. However, deployment and a commercial adoption of novel network-layer solutions is always a significant challenge. Hence, future work should concentrate on real-life deployment issues and pave a way towards adoption of PLA.

References

[1] Aboba, B. *et al.* Extensible Authentication Protocol (EAP). The Internet Society, Network Working Group, Request for Comments: 3748. June 2004.

[2] Ain, M. *et al.* PSIRP Deliverable D2.4 - Update on the Architecture and Report on Security Analysis [online]. Available from: http://www.psirp.org/files/Deliverables/FP7-INFSO-ICT-216173-PSIRP-D2_4_ArchitectureUpdateAndSecurityAnalysis.pdf [Accessed 10th August 2009].

[3] Al Hasib, A. Design and implementation of Wireless Packet Level Authentication (WPLA). Master's thesis, Helsinki University of Technology, Espoo, Finland, 2009.

[4] Altera. About HardCopy ASIC Series [online]. Available from: http://www.altera.com/products/devices/hardcopy-asics/about/hrd-index.html [Accessed 21st May 2008].

[5] Andersen, D. G. *et al*. Accountable Internet protocol (AIP). In Proceedings of SIGCOMM 2008, pp. 339-350, Seattle, USA, August 2008.

[6] Anti-Phishing Working Group. Phishing Activity Trends Report - 3rd Quarter 2009 [online]. Available from: http://www.antiphishing.org/reports/apwg_report_Q3_2009.pdf [Accessed 22nd February 2010].

[7] Arends, R. *et al.* DNS Security Introduction and Requirements. The Internet Society, Network Working Group, Request for Comments: 4033. March 2005.

[8] Atkins, D. *et al.* Internet Security. New Riders, pp. 257-316. 1996.

[9] Atkins, D. and Austein, R. Threat Analysis of the Domain Name System (DNS). The Internet Society, Network Working Group, Request for Comments: 3833. August 2004.

[10] Aura, T. Cryptographically Generated Addresses (CGA). The Internet Society, Network Working Group, Request for Comments: 3972. March 2005.

[11] Barker, E. *et al.* Recommendation for Key Management – Part 1: General. National Institute of Standards and Technology, NIST Special Publication 800-57, March 2007.

[12] BBC. Bank customer data sold on eBay [online]. Available from: http://news.bbc.co.uk/1/hi/uk/7581540.stm [Accessed 24th April 2009].

[13] Blumenthal, M. S. and Clark, D. D. Rethinking the Design of the Internet: The End-to-End Arguments vs. the Brave New World. ACM Transactions on Internet Technology (TOIT), Volume 1, Issue 1, pp. 70-109. August 2001.

[14] Brumley, B. Efficient Three-Term Simultaneous Elliptic Scalar Multiplication with Applications. In Proceedings of the 11th Nordic Workshop on Secure IT Systems--NordSec '06, pp. 105-116. Linkoping, Sweden, October 2006.

[15] Brumley, B. Left-to-Right Signed-Bit τ-adic Representations of *n* Integers (short paper). In Proceedings of Information and Communications Security, 8th International Conference - ICICS '06, pp. 469-478. December 2006.

[16] Brumley, B. and Järvinen, K. Koblitz Curves and Integer Equivalents of Frobenius Expansions. Revised Selected Papers of the 14th Annual Workshop on Selected Areas in Cryptography, SAC 2007, pp. 126-137. Ottawa, Canada, August 2007.

[17] Brumley, B. and Nyberg, K. Differential Properties of Elliptic Curves and Blind Signatures. In Proceedings of Information Security, 10th International Conference - ISC '07, volume 4779 of Lecture Notes in Computer Science, pp. 376-389. Springer-Verlag, 2007.

[18] Caesar, M. *et al.* ROFL: Routing on Flat Label. In Proceedings of SIGCOMM 2006, pp. 363-374, Pisa, Italy, September 2006.

[19] Camarillo, G. and Melen, J. HIP (Host Identity Protocol) Immediate Carriage and Conveyance of Upperlayer Protocol Signaling (HICCUPS). The Internet Society, Network Working Group, Internet Draft. March 2010.

[20] Cam-Winget, N. *et al.* Security flaws in 802.11 data link protocols. Communications of the ACM, Volume 46, Issue 5, pp. 35-39. May 2003.

[21] Candolin, C. Securing Military Decision Making In a Network-centric Environment. Doctoral dissertation, Department of Computer Science and Engineering, Helsinki University of Technology, Espoo, Finland, 2005.

[22] Case, J. *et al.* A Simple Network Management Protocol. The Internet Society, Network Working Group, Request for Comments: 1067, August 1988.

[23] CERT-FI. CERT-FI yhteydenotot nimikkeittäin [online]. Available from: http://www.cert.fi/katsaukset/tilastot.html [Accessed 22th February 2009].

[24] Chokhani, S. and Ford, W. Internet X.509 Public Key Infrastructure - Certificate Policy and Certification Practices Framework. The Internet Society, Network Working Group, Request for Comments: 2527. March 1999.

[25] Cisco. Cisco CRS-1 Carrier Routing System [online]. Available from: http://www.cisco.com/en/US/products/ps5763/index.html [Accessed 12th January 2010].

[26] Cisco. Cisco Visual Networking Index: Forecast and Methodology, 2008-2013, 9th June 2009 [online]. Available from: http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360_ns827_Networking_Solutions_White_Paper.html [Accessed 7th January 2010].

[27] Cisco. Voice Over IP - Per Call Bandwidth Consumption [online]. Available from: http://www.ciscosystems.com/en/US/tech/tk652/tk698/technologies_tech_note09186a0080094ae2.shtml [Accessed 10th February 2009].

[28] Clark, D. The design philosophy of the DARPA internet protocols. ACM SIGCOMM Computer Communication Review, Symposium proceedings on Communications architectures and protocols SIGCOMM '88, Volume 18 Issue 4, pp. 106-114. August 1988.

[29] Clark, D. D. and Blumenthal, M. S. End-to-end Arguments in Application Design: The Role of Trust. In Proceedings of TPRC, 2007.

[30] CNN. Bill proposes ISPs, Wi-Fi keep logs for police" [online]. Available from: http://www.cnn.com/2009/TECH/02/20/internet.records.bill/index.html [Accessed 5th June 2009].

[31] Deering, S. Host extensions for IP multicasting. The Internet Society, Network Working Group, Request for Comments: 1112. August 1989.

[32] Deering, S. *et al.* Internet Protocol, Version 6 (IPv6) Specification. The Internet Society, Network Working Group, Request for Comments: 2460. December 1998.

[33] Dierks, T. The TLS Protocol Version 1.0. The Internet Society, Network Working Group, Request for Comments: 2246. January 1999.

[34] Diffie, W. and Hellman M. E. New Directions in Cryptography. IEEE Transactions on Information Theory, Volume 22, Number 6, pp. 644-654. November 1976.

[35] Dobbertin, H. *et al.* The hash function RIPEMD-160 [online], 2010. Available from: http://homes.esat.kuleuven.be/~bosselae/ripemd160.html [Accessed 23th March 2010].

[36] Ephremides, A. *et al*. A design concept for reliable mobile radio networks with frequency hopping signaling. Proceedings of the IEEE, Volume 75, Issue 1, pp. 56-73. January 1987.

[37] European Parliament. Directive 2006/24/EC [online]. Available from: http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:32006L0024:EN:HTML [Accessed 6th May 2009].

[38] European Parliament. Directive 2002/21/EC [online]. Available from: http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2002:108:0033:0050:EN:PDF [Accessed 6th May 2009].

[39] European Telecommunications Standards Institute. Digital cellular telecommunications system (Phase 2+); Specification of the Subscriber Identity Module - Mobile Equipment (SIM - ME) interface (GSM 11.11) , December 1995.

[40] Flickenger, R. Building Wireless Community Networks, O'Reilly, 2002.

[41] Forsten, J. *et al.* Packet Level Authentication: Hardware Subtask Final Report. Technical report [online]. Available from: http://www.tcs.hut.fi/Software/PLA/new/doc/PLA_HW_final_report.pdf [Accessed 4th February 2009].

[42] FreeNetworks.org. Volunteer cooperative association dedicated to education, collaboration, and advocacy for the creation of FreeNetworks [online]. Available from: http://www.freenetworks.org/ [Accessed 6th March 2009].

[43] Gallagher, P. Digital Signature Standard (DSS). National Institute of Standards and Technology, Federal Information Processing Standards Publication FIPS 186-3, June 2009.

[44] Gallagher, P. Secure Hash Standard (SHS). National Institute of Standards and Technology, Federal Information Processing Standards Publication FIPS 180-3, October 2008.

[45] Gleeson, B. *et al*. A Framework for IP Based Virtual Private Networks. The Internet Society, Network Working Group, Request for Comments: 2764. February 2000.

[46] Gribble, S. D. *et al*. Scalable, Distributed Data Structures for Internet Service Construction. In proceedings of the 4th Symposium on Operating System Design and Implementation (OSDI 2000), pp. 319-332. 2000.

[47] Handley, M. and Greenhalgh, A. Steps Towards a DoS-resistant Internet Architecture. In Proceedings of ACM SIGCOMM workshop on Future directions in network architecture, pp. 49-56. New York, USA, 2004.

[48] Hauser, R. *et al*. Reducing the Cost of Security in Link State Routing. In Proceedings of the Network and Distributed System Security Symposium, NDSS'97, 1997.

[49] Heer, T. *et al*. ALPHA: An Adaptive and Lightweight Protocol for Hop-by-hop Authentication. In Proceedings of ACM CoNEXT 2008. Madrid, Spain, December 2008.

[50] IEEE Computer Society. IEEE Standard for Local and metropolitan area networks - Virtual Bridged Local Area Networks. May 2006 [online]. Available from: http://standards.ieee.org/getieee802/download/802.1Q-2005.pdf [Accessed 20th May 2008].

[51] Infowatch. The biggest data leak ever: 26.5 million U.S. veterans face risk of ID theft [online]. Available from: http://www.infowatch.biz/threats?chapter=148831545&id=187279348 [Accessed 24th April 2009].

[52] Intel. Intel® Advanced Encryption Standard (AES) Instructions Set - Rev 3 [online]. Available from: http://software.intel.com/en-us/articles/intel-advanced-encryption-standard-aes-instructions-set/ [Accessed 30th April 2010].

[53] International Technology Roadmap for Semiconductors (ITRS). ITRS 2009 Edition [online]. Available from: http://www.itrs.net/Links/2009ITRS/Home2009.htm [Accessed 12th January 2010].

[54] International Telecommunication Union. Information technology - Open Systems Interconnection - Basic Reference Model: The basic model, Recommendation X.200 (07/94) [online]. Available from: http://www.itu.int/rec/T-REC-X.200-199407-I/en [Accessed 28th April 2010].

[55] Jacobson, V. Compressing TCP/IP Headers for Low-Speed Serial Links. The Internet Society, Network Working Group, Request for Comments: 1144. February 1990.

[56] Jacobson, V. *et al.* Networking named content. In Proceedings of ACM CoNEXT 2009, pp. 1-12. Rome, Italy, December 2009.

[57] Jung, E. and Vaidya, N. H. Improving IEEE 802.11 power saving mechanism. Wireless Networks, Volume 14, Issue 3, pp. 375-391. June 2008.

[58] Järvinen, K. and Skyttä, J. High-Speed Elliptic Curve Cryptography Accelerator for Koblitz Curves. In Proceedings of the 16th IEEE Symposium on Field-programmable Custom Computing Machines, FCCM 2008, pp.109-118. Palo Alto, USA, April, 2008.

[59] Järvinen, K. and Skyttä, J. On Parallelization of High-Speed Processors for Elliptic Curve Cryptography. IEEE Transaction on Very Large Scale Integration (VLSI) Systems, Volume 16, Number 9, pp. 1162-1175. September 2008.

[60] Järvinen, K. *et al*. Efficient Circuitry for Computing tau-adic Non-Adjacent Form. In Proceedings of the 13th IEEE International Conference on Electronics, Circuits and Systems, ICECS 2006, pp. 232-235. Nice, France, December 2006.

[61] Järvinen, K. *et al*. FPGA Design of Self-certified Signature Verification on Koblitz Curves. In Proceedings of the Workshop on Cryptographic Hardware and Embedded Systems, CHES 2007, pp. 256-271. Vienna, Austria, September 2007.

[62] Kari, H. H. Attacking Internet. Public presentation [online]. Available from: http:// www.huoltovarmuus.fi/documents/7/TIVA_20090506%20Hannu%20Kari %20Attacking_Internet.ppt [Accessed 12th January 2010].

[63] Kent, S. and Atkinson, R. Security Architecture for the Internet Protocol. The Internet Society, Network Working Group, Request for Comments: 2401. November 1998.

[64] Koblitz, N. CM curves with good cryptographic properties. In Proceedings of the 11th Annual International Cryptology Conference on Advances in Cryptology, pp. 279-287. August 1991.

[65] Kobliz, N. Elliptic Curve Cryptosystems. Mathematics of Computation. Volume 48, pp. 203-209. 1987.

[66] Koponen, T. *et al.* A Data-oriented (and beyond) network architecture. In Proceedings of SIGCOMM 2007, pp. 181-192, Kyoto, Japan, August 2007.

[67] Kuon, I. and Rose, J. Measuring the gap between FPGAs and ASICs. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. Volume 26, no. 2, pp. 203-215. February 2007.

[68] Lagutin, D. *et al.* Roles and Security in a Publish/Subscribe Network Architecture. In Proceedings of the IEEE Symposium on Computers and Communications (ISCC'10), pp. 68-74. Riccione, Italy, June 2010.

[69] Lagutin, D. and Kari, H. H. Controlling incoming connections using certificates and distributed hash tables. In Proceedings of the 7th International Conference on Next Generation Teletraffic and Wired/Wireless Advanced Networking (NEW2AN 2007), pp. 455-467. St. Petersburg, Russia, September 2007.

[70] Lagutin, D. and Tarkoma, S. Cryptographic Signatures on the Network Layer - an Alternative to the ISP Data Retention. In Proceedings of the IEEE Symposium on Computers and Communications (ISCC'10), pp. 87-93. Riccione, Italy, June 2010.

[71] Lagutin, D. and Tarkoma, S. Public Key Signatures and Lightweight Security Solutions in a Wireless Environment. In Proceedings of the 9th International Conference on Next Generation Teletraffic and Wired/Wireless Advanced Networking (NEW2AN 2009), pp. 253-265. St. Petersburg, Russia, September 2009.

[72] Libnet [online]. Available from: http://libnet.sourceforge.net/ [Accessed 15th March 2010].

[73] Lundberg, J. A Wireless Multicast Delivery Architecture for Mobile Terminals. Doctoral dissertation, Department of Computer Science and Engineering, Helsinki University of Technology, Espoo, Finland, 2006.

[74] Markoff, J. Do We Need a New Internet? The New York Times, Week in Review, 14th February 2009 [online]. Available from: http://www.nytimes.com/2009/02/15/weekinreview/15markoff.html [Accessed 20th February 2009].

[75] McAfee. The Carbon Footprint of Email Spam Report [online]. Available from: http://img.en25.com/Web/McAfee/CarbonFootprint_28pg_web_REV.PDF [Accessed 12th January 2010].

[76] Melia, T. Mobility Services Transport: Problem Statement. The Internet Society, Network Working Group, Request for Comments: 5164. March 2008.

[77] Menezes, A. Elliptic Curve Public Key Cryptosystems, Kluwer Academic Publishers, 1993.

[78] Merkle, R. Secrecy, authentication, and public key systems. Doctoral dissertation, Department of Electrical Engineering, Stanford University, 1979.

[79] Miller, V. Use of Elliptic Curves in Cryptography. Lecture notes in computer sciences; 218 on Advances in cryptology - CRYPTO 85, pp. 417-426. Springer-Verlag, 1986.

[80] Moskowitz, R. *et al*. Host Identity Protocol. The Internet Society, Network Working Group, Request for Comments: 5201. April 2008.

[81] Myers, M. *et al.* X.509 Internet Public Key Infrastructure - Online Certificate Status Protocol (OCSP). The Internet Society, Network Working Group, Request for Comments: 2560. June 1999.

[82] Nazario, J. Estonian DDoS Attacks - A summary to date. The Arbor Networks Security Blog [online]. Available from: http://asert.arbornetworks.com/2007/05/estonian-ddos-attacks-a-summary-to-date/ [Accessed 21st May 2008].

[83] Nikander, P. and Laganier, J. Host Identity Protocol (HIP) Domain Name System (DNS) Extension. The Internet Society, Network Working Group, Request for Comments: 5205. April 2008.

[84] Mills, D. L. Network Time Protocol Version 4 Reference and Implementation Guide. NTP Working Group, Technical Report 06-6-1. June 2006.

[85] OpenID Foundation [online]. Available from: http://openid.net/ [Accessed 10th February 2010].

[86] Packet Level Authentication [online]. Available from: http://www.tcs.hut.fi/Software/PLA/new/ [Accessed 10th May 2008].

[87] Packet Level Authentication (PLA) library [online]. Available from: http://www.psirp.org/files/Deliverables/libpla-standalone-0.1.tar.gz [Accessed 12th January 2010].

[88] Perrig, A. *et al*. The TESLA Broadcast Authentication Protocol, CryptoBytes, Volume 5, no. 2, pp. 2-13, 2002.

[89] Qiao, D. *et al.* Interference Analysis and Transmit Power Control in IEEE 802.11a/h Wireless LANs. IEEE/ACM Transactions on Networking, Volume 15, Issue 5, pp. 1007-1020. October 2007.

[90] Rajaniemi, A. Verkkopankin toimintavarmuuden turvaaminen tietoverkon näkökulmasta. Master's thesis, Helsinki University of Technology. 2005.

[91] Rekhter, Y. *et al*. A Border Gateway Protocol 4 (BGP-4). The Internet Society, Network Working Group, Request for Comments: 4271. January 2006.

[92] Rivest, R. L. *et al.* A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Communications of ACM, Volume 21, Issue 2, pp. 120-126. February 1978.

[93] Rivest, R. L.. SEXP (S-expressions) [online]. Available from: http://theory.lcs.mit.edu/~rivest/sexp.html [Accessed 22st February 2010].

[94] Saha, S. and Lagutin, D. PLA-MIH: A Secure IEEE802.21 Signaling Scheme. In Proceedings of the IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob 2009), pp. 252-257. Marrakech, Morocco, October 2009.

[95] Saltzer, J. H. *et al.* End-to-end Arguments in System Design. ACM Transactions on Computer Systems (TOCS), Volume 2, Issue 4, pp. 277-288. November 1984.

[96] Scarfone, K. and Mell, P. Guide to Intrusion Detection and Prevention Systems (IDPS). National Institute of Standards and Technology, NIST Special Publication 800-94, February 2007.

[97] Schmidt, J. The hole trick, How Skype & Co. get round firewalls [online]. Available from: http://www.h-online.com/security/features/How-Skype-Co-get-round-firewalls-747197.html [Accessed 28th January 2010].

[98] Schuba, C. *et al*. Analysis of a Denial of Service Attack on TCP. In Proceedings of the 1997 IEEE Symposium on Security and Privacy, pp. 208-223. Oakland, USA, May 1997.

[99] Seehra, A. *et al.* A policy framework for the future Internet. In Proceedings of HotNets VIII, New York, USA, October 2009.

[100] Specht, S. M. and Lee, R. B. Distributed Denial of Service: Taxonomies of Attacks, Tools and Countermeasures. In Proceedings of the 17th International Conference on Parallel and Distributed Systems, 2004 International Workshop on Security in Parallel and Distributed Systems, pp. 543-550. San Francisco, USA, September 2004.

[101] Srisuresh, P. and Holdrege, M. IP Network Address Translator (NAT) Terminology and Considerations. The Internet Society, Network Working Group, Request for Comments: 2663. August 1999.

[102] Stoica, I., Adkins, D., Zhuang, S., Shenker, S., and Surana, S. Internet Indirection Infrastructure. In Proceedings of ACM SIGCOMM 2002, pp. 73-86. Pittsburgh, USA, August, 2002.

[103] Symbian Foundation. Symbian^3 - Symbian Developer Community [online]. Available from: http://developer.symbian.org/wiki/index.php/Symbian%5E3 [Accessed 30th April 2010].

[104] Syverson, P. A Taxonomy of replay attacks. In Proceedings of the 7th IEEE Computer Security Foundation Workshop (CSFW 94), pp. 187-191. Franconia, USA, June 1994.

[105] Taddeo, A. V. *et al.* Negotiation of Security Services: a Multi-criteria Decision Approach. In Proceedings of the 4th Workshop on Embedded Systems Security (WESS'09). Grenoble, France, October 2009.

[106] Tcpdump/libpcap public repository [online]. Available from: http://www.tcpdump.org/ [Accessed 15th March 2010].

[107] TeleGeography Research. Global Internet Geography, Executive Summary. [online]. Available from: http://www.telegeography.com/products/gig/samples08/gig.zip [Accessed 24th February 2009].

[108] Tews, E. *et al*. Breaking 104 bit WEP in less than 60 seconds [online]. Available from: http://eprint.iacr.org/2007/120.pdf [Accessed 22nd December 2009].

[109] Trusted Computing Group. Trusted Network Connect, Specification Version 1.0. May 2009.

[110] Trusted Computing Group. TPM Work Group [online]. Available from: https://www.trustedcomputinggroup.org/groups/tpm/ [Accessed 20th May 2008].

[111] VIA. VIA Padlock Security Engine [online]. Available from: http://www.via.com.tw/en/initiatives/padlock/hardware.jsp [Accessed 30th April 2010].

[112] Visala, K., Lagutin, D., and Tarkoma, S. LANES: An Inter-Domain Data-Oriented Routing Architecture. In Proceedings of ReArch'09, pp. 55-60. Rome, Italy, December 2009.

[113] Walsworth, C. et al. The CAIDA Anonymized 2009 Internet Traces - Anonymized 2009 dataset [online]. Available from: http://www.caida.org/data/passive/passive_2009_dataset.xml [Accessed 18th February 2009].

[114] Watson, P. Slipping in the Window: TCP Reset Attacks (white paper). In Proceedings of CanSecWest. 2004.

[115] Wendlandt, D. *et al*. FastPass: Providing First-Packet Delivery [online]. CyLab technical report, March 2006. Available from: http://www.cylab.cmu.edu/files/pdfs/tech_reports/cmucylab06005.pdf [Accessed 22nd December 2009].

[116] Wong, C. and Lam, S. Digital signatures for flows and multicasts. In proceedings on the 6th International Conference on Network Protocols (ICNP '98), pp. 198—209, 1998.

[117] Yaar, A. *et al.* SIFF: A Stateless Internet Flow Filter to Mitigate DDoS Flooding Attacks. In Proceedings of 2004 IEEE Symposium on Security and Privacy, pp. 130-143. Oakland, USA, May 2004.

[118] Yang, X. *et al.* A DoS-limiting Network Architecture. In Proceedings of SIGCOMM 2005, pp. 241–252, Philadelphia, USA, August 2005.

[119] Zakon, R. Hobbes' Internet Timeline v8.2 [online]. Available from: http://www.zakon.org/robert/internet/timeline/ [Accessed 22nd May 2008].

[120] Zhong, L. Make "sense" for Computing. Public presentation [online]. Available from: http://www.ece.rice.edu/corp/annualmtg/mtgarchive/2008archive/lzhong_affil08.pdf [Accessed 5th February 2009].

[121] Xiao, Y. *et al.* Practical Power Modeling of Data Transmission over 802.11g for Wireless Applications. In Proceedings of the e-Energy 2010, 1st International Conference on Energy-Efficient Computing and Networking, pp. 75-84. Passau, Germany, April 2010.

Appendix A: TTP certificate format

The high level structure of the certificate is presented below in the S-expressions format.

```
<cert>:: "(" "cert" <issuer> <subject> <identity>
<validity> <rights> <deleg> <sig> ")";
```

The certificate is granted by `<issuer>` to `<subject>` and `<validity>` determines the time period when the certificate is valid. Fields `<rights>` and `<deleg>` determine what rights the certificate contains and which rights can be delegated to other parties. Finally, there is an issuer's signature over the certificate. Each field is described in more detail below.

PLA uses identity-based implicitly-certified keys, thus the actual subject's public key is calculated using information present in a TTP certificate as described in Section 4.2, and the subject's public key is not physically included in the certificate.

The “issuer” field

```
<issuer>:: "(" "issuer" <issuer-details> ")";
<issuer-details>:: "(" "pub-key" <pub-key> "locator"
     <locator> ")";
<pub-key>:: <byte-string>;
<locator>:: <byte-string>;
```

The `<issuer>` field consists of two elements, the public key of the issuer and the issuer's locator (e.g., an IP address). If the certificate is issued by a normal user instead of a trusted third party, the locator field is zero.

The “subject” field

```
<subject>:: "(" "subject" <pub-key> ")";
```

Subject's public key is calculated using information of other fields of the TTP certificate.

The “identity” field

```
<identity>:: "(" "identity" <id> ")";
<id>:: <byte-string>;
```

This field contains a user's identity which is given by a trusted third party. The TTP gives an unique identity to each of its users.

The "validity" field

```
<validity>:: "(" "valid" <not-before> <not-after> ")";
<not-before>:: "(" "not-before" <timestamp> ")";
<not-after>:: "(" "not-after" <timestamp> ")";
<timestamp>:: <byte-string>;
```

The validity field contains timestamps that determine the period when the certificate is valid. Both fields `<not-before>` and `<not-after>` are always present. If the `<not-before>` field is zero, the certificate is valid immediately. The `<timestamp>` uses Unix timestamp format and it contains number of seconds after 1st January 1970.

The "rights" field

```
<rights>:: "(" "rights" <bits> ")";
<bits>:: <string>;
```

The `<rights>` field determines the rights given by the certificate. For trusted third party certificates, there are three different rights expressed as bits:

xxx1 – The right to delegate other rights.

xx1x – The right to request a new certificate.

x1xx – The right to send data to the network at a normal priority.

1xxx – The right to send data to the network at a high priority.

The "deleg" field

```
<deleg>:: "(" "deleg" <bits> ")";
```

This field determines which rights can be delegated to other parties. The format is the same as with the `<rights>` field:

xxx1 – The right to delegate the right to delegate rights.

xx1x – The right to delegate the right to request a new certificate.

x1xx – The right to delegate the right to send data to the network at a normal priority.

1xxx – The right to delegate the right to send data to the network at a high priority.

The aim of the first right is to let the issuer have more control over delegability of other rights. If the "rights" field is in a form of xxx1 and the "deleg" field is xxx1 then it

means that the rights can be delegated indefinitely forward. If the “rights” field is xxx1 but the “deleg” field is xxx0 then the subject can delegate rights to another subject, but this another subject cannot delegate rights forward any more. If both fields are in a form of xxx0 then it means that no rights can be delegated to other parties. Thus, the issuer can allow rights to be delegated only once or indefinitely or not allow delegation at all.

The “signature” field

```
<sig>:: "(" "signature" <signature> ")";
<signature>:: <byte-string>;
```

The signature field contains the `<issuer>`'s cryptographic signature over the certificate ignoring the `<issuer>` field.

Appendix B: Certificate format for controlling incoming connections

The structure of the certificate is presented below in the S-expressions format.

```
<cert>:: "(" "cert" <issuer> <subject> <validity> <rights>
<deleg> <sig> ")";
```

The certificate is granted by `<issuer>` to `<subject>` and `<validity>` determines the time period when the certificate is valid. Fields `<rights>` and `<deleg>` determine what rights the certificate contains and which rights can be delegated to other parties. Finally, there is an issuer's signature over the certificate. Each field is described in more detail below.

The "issuer" field

```
<issuer>:: "(" "issuer" <issuer-details> ")";
<issuer-details>:: "(" "pub-key" <pub-key> "locator"
     <locator> ")";
<pub-key>:: <byte-string>;
<locator>:: <byte-string>;
```

The `<issuer>` field consists of two elements, the public key of the issuer and the issuer's locator (e.g., an IP address). If the certificate is issued by a normal user instead of a trusted third party, the locator field is zero.

The "subject" field

```
<subject>:: "(" "subject" <pub-key> ")";
```

The subject field contains a public key of the subject.

The "validity" field

```
<validity>:: "(" "valid" <not-before> <not-after> ")";
<not-before>:: "(" "not-before" <timestamp> ")";
<not-after>:: "(" "not-after" <timestamp> ")";
<timestamp>:: <byte-string>;
```

The validity field contains timestamps that determine the period when the certificate is valid. Both fields `<not-before>` and `<not-after>` are always present. If the

<not-before> field is zero, the certificate is valid immediately. The <timestamp> uses Unix time format and it contains number of seconds after 1st January 1970.

The "rights" field

```
<rights>:: "(" "rights" <bits> ")";
<bits>:: <string>;
```

The <rights> field determines the rights given by the certificate. There are five different rights expressed as bits:

xxxx1 – The right to delegate other rights.

xxx1x – The right to send the data to the network.

xx1xx – The right to request a new certificate.

x1xxx – The right to create an incoming connection.

1xxxx – The right for session initialization management. This right might be necessary when creating an incoming connection.

The "deleg" field

```
<deleg>:: "(" "deleg" <bits> ")";
```

This field determines which rights can be delegated to other parties. The format is the same as with the <rights> field:

xxxx1 – The right to delegate the right to delegate rights.

xxx1x – The right to delegate the right to send data to the network.

xx1xx – The right to delegate the right to request a new certificate.

x1xxx – The right to delegate the right to create an incoming connection.

1xxxx – The right to delegate the right for session initialization management. This right might be necessary when creating an incoming connection.

The aim of the first right is to let the issuer have more control over delegability of other rights. If the "rights" field is in a form of xxxx1 and the "deleg" field is xxxx1 then it means that the rights can be delegated indefinitely forward. If the "rights" field is xxxx1 but the "deleg" field is xxxx0 then the subject can delegate rights to another subject, but this another subject cannot delegate rights forward any more. If both fields are in a form of xxxx0 then it means that no rights can be delegated to other parties. Thus, the issuer can allow rights to be delegated only once or indefinitely or not allow delegation at all.

The “signature” field

```
<sig>:: "(" "signature" <signature> ")";
<signature>:: <byte-string>;
```

The signature field contains the signature over the whole certificate with the `<issuer>`'s private key.

Appendix C: Format for certificate requests

```
<request>:: "(" "request" <subject> <validity> <token>
     <sig> ")";
<request>:: "(" "request" <subject> <validity> <sig> ")";
```

The certificate request contains subject's public key and requested validity time. The request may also contain an optional authorization token. Token field is explained below while subject and validity fields are identical to fields mentioned previously.

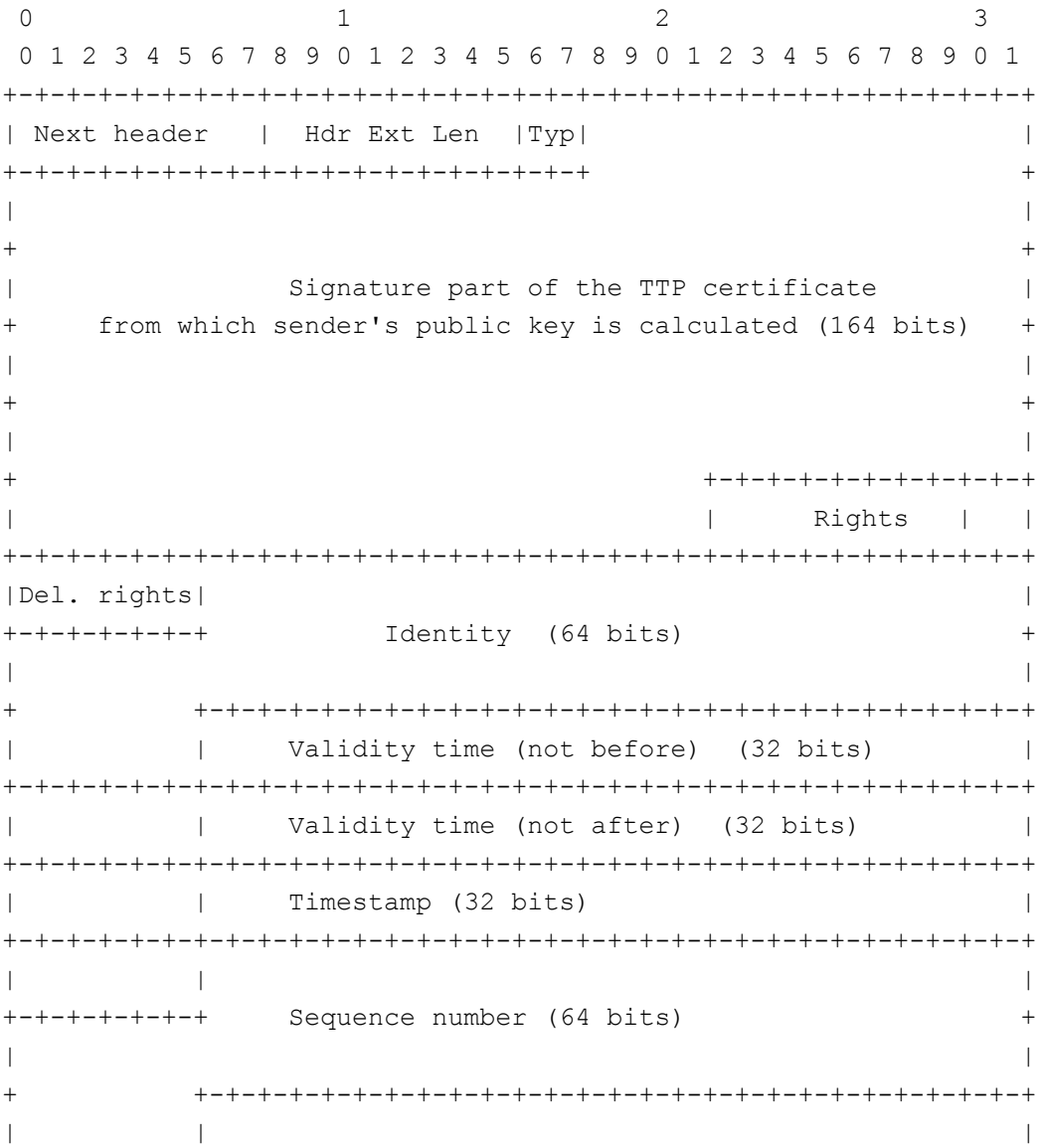
The “token” field

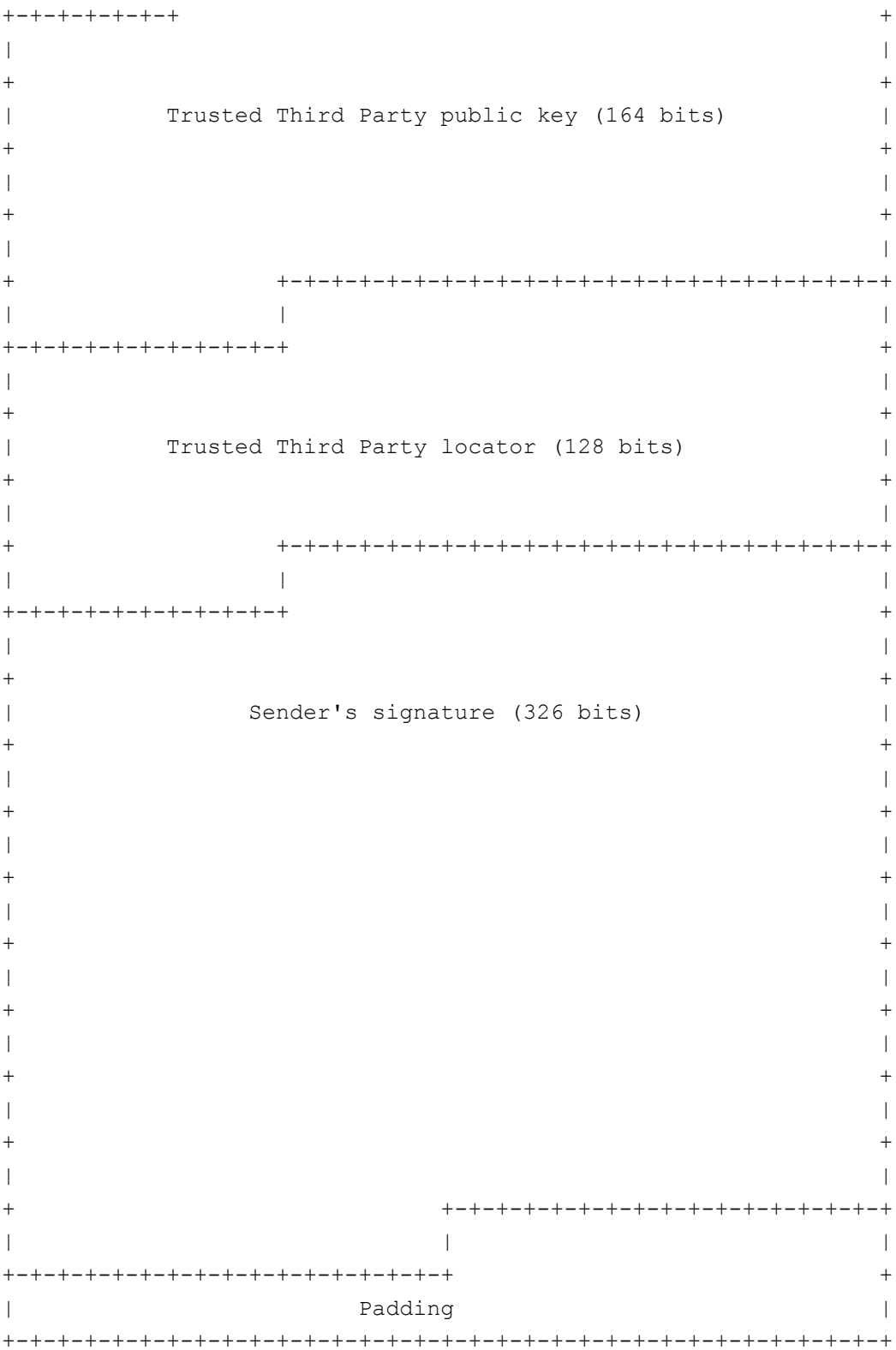
```
<token>:: "(" "token" <t> ")";
<t>:: <byte-string>;
```

This field contains an authorization token which may be present in a certificate request. The aim of the token is to guarantee that the requesting party has a right to request a certificate.

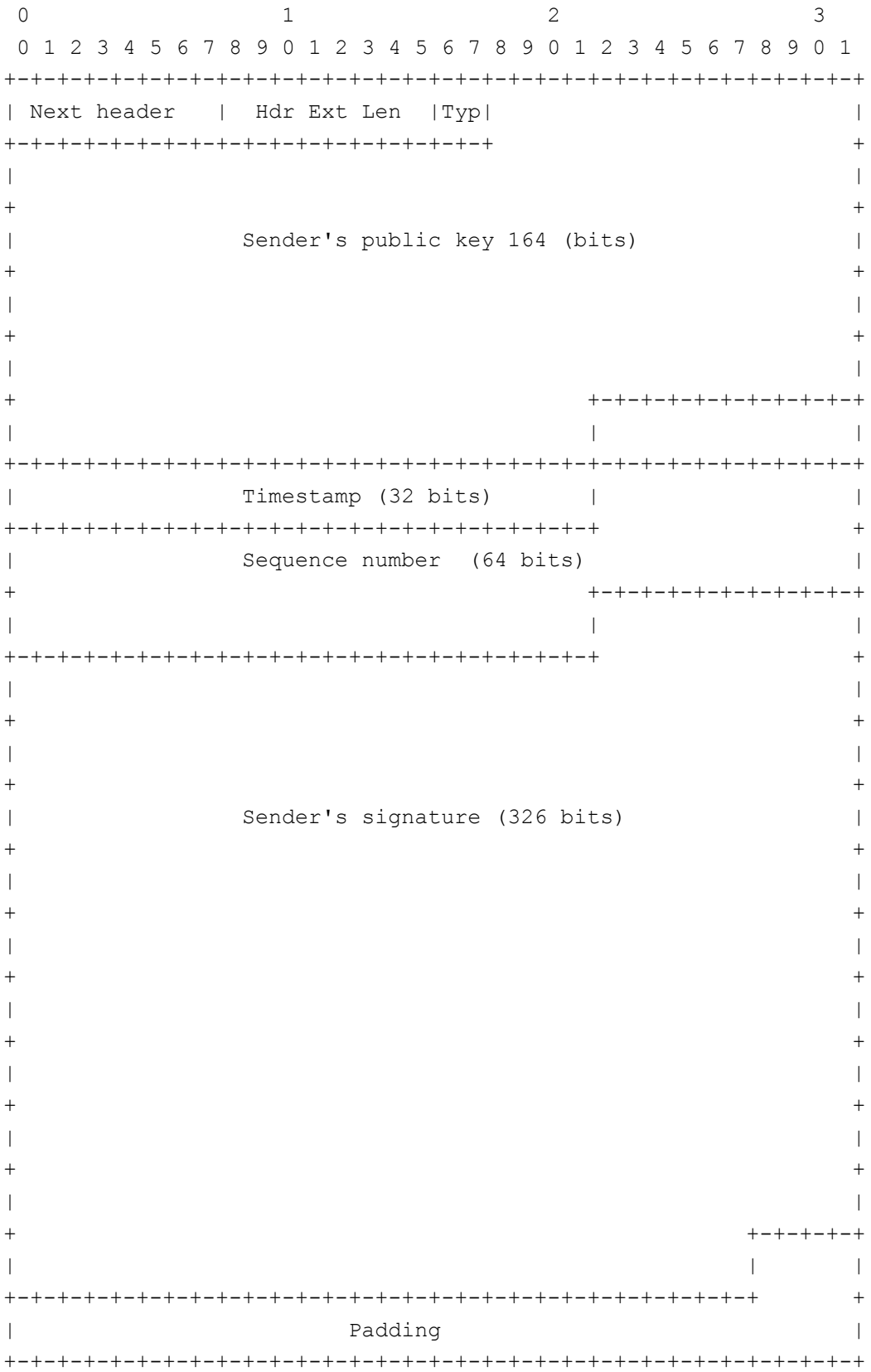
Appendix D: PLA header

The PLA header is added on top of an IPv6 header using an extension header id 143, there was no specific reason for choosing this number, it was just a first id number available. The next header and the header extension length (Hdr Ext Len) fields are standard fields from the IPv6 extension header format, they contains the id of the next header and the length of the PLA extension header. The type field refers to PLA header type, 00 denotes a full PLA header while 01 denotes a lightweight header without the TTP certificate information. The size of PLA related fields in the header is 1024 bits (128 bytes). Since the length of the IPv6 extension header must be divisible by 64, the total size of the header with IPv6-related fields and padding is 1088 bits (136 bytes) as presented below.




Trusted Third Party public key (164 bits)
Trusted Third Party locator (128 bits)
Sender's signature (326 bits)
Padding

Below is presented a lightweight PLA header with the total length of 640 bits (80 bytes)



ISBN 978-952-60-3464-5
ISBN 978-952-60-3465-2 (PDF)
ISSN 1795-2239
ISSN 1795-4584 (PDF)