Advances in ad hoc networking: Packet Level Authentication

Dmitrij Lagutin, dlagutin@cc.hut.fi

1. Introduction

The Internet was originally designed with an external threat in mind, it was not designed to be resistant against internal attacks where the attacker controls nodes connected to the Internet. As a result, the Internet is vulnerable against many kind of attacks. Denial of service and distributed denial of service attacks can be launched relatively easily against nodes in the network. Packets going through the network can be modified or duplicated. Replay attacks can be launched by delaying or duplicating legitimate packets. Such attacks consume network's resources and they are especially dangerous for wireless and ad hoc networks since such networks have usually very limited bandwidth and energy resources. Current security solutions, like IPSec, concentrate on providing end-to-end security and they do not protect underlying network infrastructure. If the packet protected by IPSec has been modified, only endpoints of the connection will notice the modification, and as a result these modified packets will consume network's resources unnecessarily. In addition, such security solutions are useless if the underlying infrastructure is attacked and is unable to deliver packets to the destination as a result of e.g. denial of service attack.

2. Packet Level Authentication

The Packet Level Authentication (PLA) [1][2] is a novel way to protect IP networks against different kinds of attacks. The basic idea of the PLA is to make it possible for every node in the network to verify authenticity of every packet without having any kind of contact with the sender of the packet. A good analogy for this principle would be a paper currency: salesperson in the shop does not need to contact the bank that has issued the note to verify the authenticity of the note. It is enough to just verify various security measures inside the note like watermark, metal string, and hologram. Similarly, using the PLA every node in the network can determine authenticity and validity of the packet just by looking at packet itself. If there is something wrong with packet, it can be discarded immediately before it can consume additional resources.

2.1 Design goals

Design goals of the Packet Level Authentication include:

Validation of packets: Every node in the network must be able to validate the authenticity of every packet without having contact with the sender of the packet.
Survivable in dynamic and hostile environment: The system must work in dynamic environment like wireless ad hoc networks where the topology of the network changes rabidly. The system must also work in hostile environment where attacks against system occur frequently.
Ability to add new nodes to the network and remove compromised nodes from the network: It should be possible to add new nodes to the system while at the same there should be a possibility to revoke rights to communicate from compromised or hostile nodes.
Scalability: The system must scale from small, wireless sensor networks to high speed Internet core network.
Minimum traffic overhead: The amount of extra packets and overhead (additional information in each packet) should be minimized.

Minimum trust between nodes: The system must work without additional security association setups between nodes.

In addition, the validation requirement contains the following:
Modification detection: Any node should be able to detect if the packet has been modified by intermediate nodes.
Duplicate detection: It should be also possible to detect duplicated packets, these packets will naturally be discarded by the node that detects them.
Delay detection: There should be a way to detect delayed packets since they are often used in replay attacks.

2.2 Packet Level Authentication header

The PLA accomplishes its goals by adding an additional PLA header on top of an IPv6 header in each packet. The PLA also utilizes a public key cryptography. The elliptic curve cryptography (ECC) [3] is used with the PLA because small ECC keys are cryptographically strong, a 160bit ECC key is roughly equivalent to 1024bit RSA key in terms of security. Small key sizes are important because public keys are included in the PLA header of each packet, if the size of a public key is too big, then the PLA header would produce a high bandwidth overhead.

The PLA header contains following fields:

Public key of the sender: Every packet that the sender sends contains sender's public key, together with signature this guarantees that the sender cannot deny sending a packet. The sender's public key in the header also makes possible for other nodes to drop packets from a hostile sender.
Certificate from a Trusted Third Party (TTP) to the sender: The aim of this certificate is to guarantee that the sender is a valid, trusted entity. The TTP can be for example an operator or a state authority.
Time stamp: The aim of the time stamp is to detect packets that have been significantly delayed. Delayed packets can be a sign of a replay attack.
Sequence number: Monotonically increased sequence number makes possible to detect duplicated packets.
Signature of the sender: The packet is signed by a sender with sender's private key. The signature guarantees that forged packets will be detected.

2.3 Trusted Third Parties

Trusted Third Parties (TTPs) add another layer of security to PLA. The TTP authorizes the nodes that are communicating using the PLA. If the node is hostile or compromised, TTP will not grant the node a new certificate, and without a proper certificate from TTP, the node cannot communicate fully using the PLA.

Thus, during the validity check of the packet, validity of the TTP certificate must be also checked, and in addition it should be checked that the TTP that has granted the certificate to the sender of the packet can be trusted. The exact method for verifying the validity of TTPs is under research and will not be covered here.

If the PLA packet has a correct signature, correct TTP certificate and the TTP that has given the certificate can be trusted, then the packet is authentic and it has been sent by a legitimate node. Thus the packet can be trusted and forwarded assuming that other fields of the PLA header like time stamp and sequence number are in order.

3. Implementation of Packet Level Authentication

3.1 Software implementation

Experimental PLA implementation exists for Linux, the implementation contains Linux kernel module and userspace applications. The implementation adds a PLA header to outgoing packets. When the PLA packet is received, fields like signature are checked in the PLA header, and if everything is in order, the packet is processed by upper layers of the operating system. The source code of the experimental PLA implementation is publicly available [4]. ECC operations like signing and verifications of signatures require a lot of computational resources and calculating those operations with a general purpose processor is not feasible. Preliminary results have shown that a round trip time for a PLA packet that travels through two PLA enabled routers (2GHz Athlon64 PCS) is about 60 milliseconds. This suggests that such hardware can only verify roughly tens of packets per second. In the future, PLA software implementation will be able to utilize external hardware accelerators for handling cryptographic operations.

3.2 Hardware acceleration of cryptographic operations

Verification of cryptographic signatures requires lot of computational resources as mentioned previously. Since every node should be able to check the signature of every packet, general purpose processors are not suitable for this task. Instead, Field Programmable Gate Arrays (FPGA) can be used for improving performance of specialized tasks like cryptography. Since the FPGA can be programmed for a specific task, FPGAs produce significantly higher performance in specific tasks compared to general purpose processors.

Preliminary results of accelerating elliptic curve cryptography with FPGA chip have produced promising results. With Altera Stratix II EP 2S180 FPGA chip it is possible to achieve roughly 165000 ECC verifications per second. With a maximum payload per packet, about 10 kbits, 165000 packets equal to 1.68Gbps of traffic and with 5kbit payload, 165000 packets equal to 840Mbps of traffic. Since dedicated hardware processors achieve significantly higher performance compared to programmable FPGA chips, these results show that the PLA is scalable for very high speed networks as long as dedicated hardware is used to handle cryptographic operations.

In addition, there are several hardware implementations of elliptic curve cryptography in the literature that have produced a good performance with a low power consumption [5][6][7].

4. Applications of Packet Level Authentication

The main aim of the PLA is improve security and protect the network against various attacks. However, the PLA can also utilized for other tasks. For example, the sequence number that is present in the PLA header can be used by operators for per packet billing: the sender of packets will be charged by the amount of packets he sends. Since the sequence number increases monotonically and every packet is signed by the sender's private key, the sender cannot deny sending packets. The sequence number could also be increased by the size of the packet, this would allow to implement billing based on the amount data that the sender has sent.

The PLA can also be utilized for controlling incoming connections and to increase lifetime of wireless ad hoc networks. These cases will be described below in more detail.

4.1 Controlling incoming connections

In the current Internet architecture the initiator of the connection is in charge of the connection. The initiator of the connection can decide to whom he will make a connection and when the connection is

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

One way to solve this problem is to block all incoming connection that are not explicitly allowed to recipient. 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, it is better to block unwanted incoming connections already at the access network level, before they even reach the destination. This kind of blocking would also make it much more difficult to launch denial of service attacks against the network or recipient.

This goal can be accomplished using certificates and the PLA: the recipient of the connection grants explicit certificates to trusted initiators and the PLA is used to guarantee that connections to the recipient really originate from trusted initiators. An example of such a system is shown in figure below.

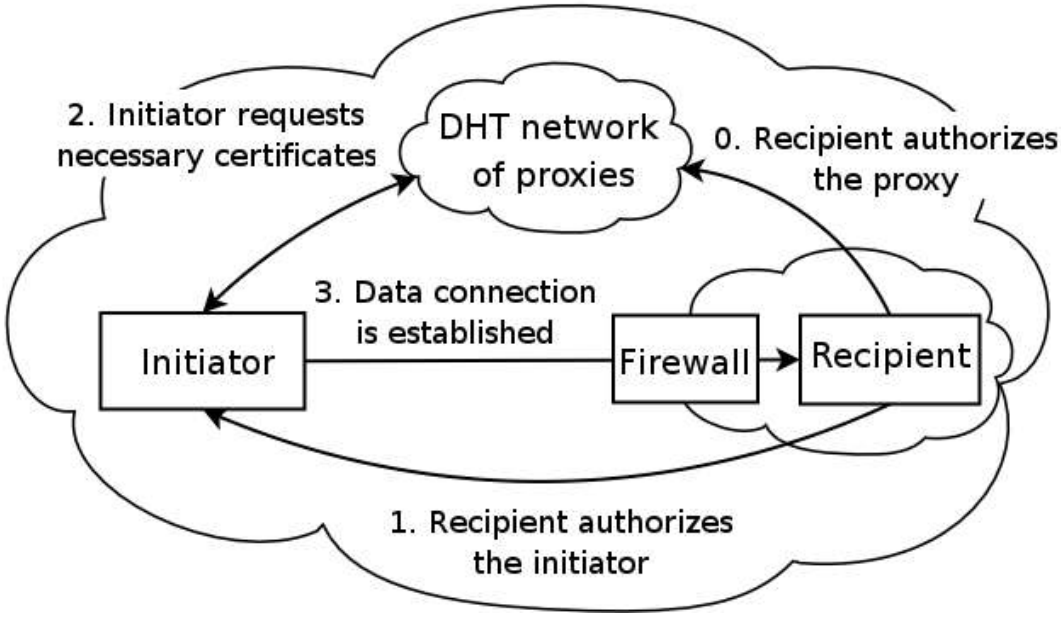

Figure 1. Controlling incoming connections with certificate and PLA

In the figure, the **proxy** is an entity in which the recipient trusts. The task of the proxy is to give certificates to other trusted parties for making incoming connections to the recipient. The proxy also keeps track of 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) network. The **firewall** is located in recipient's access networks. It takes notice of certificates that are passing through it and the firewall blocks all 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.

In the beginning of the example, the recipient must authorize the proxy using a certificate. The recipient also authorizes trusted initiator by giving him a certificate, this certificate exchange can be also carried offline and it is shown in step 1 in the figure. After the initiator has received certificate from the recipient, the initiator can contact proxy to request all necessary certificates for making an incoming connection. Exact details of those certificates are not covered here, but they include a certificate that was given by recipient to proxy in step 0, a certificate from recipient to initiator and a certificate from the firewall (access network) to the recipient. In addition proxy reports recipient's current IP address to the initiator. In the last step, the initiator sends first all those certificates to the recipient using a control message and afterwards the data connection can be established. The firewall

checks that certificates form a valid certificate chain: firewall => recipient => proxy => initiator. Such a chain shows that the recipient which has a right to use an access network has authorized the proxy that has authorized the initiator. If this certificate chain is in order, the firewall will allow the initiator to establish connection to the recipient. Naturally, revocation and delegation of certificates is also supported under such a system.

The PLA is necessary to ensure that the data is really coming from a trusted initiator. The initiator's public key inside the PLA header of data packets must match with the initiator's public key present in a certificate chain, only in that case the firewall will let traffic through. Thus, malicious party will not be able to make incoming connections using certificates that are granted to trusted initiators.

4.2 Wireless ad hoc networks

Wireless ad hoc networks have very limited resources like bandwidth and batter power. Thus various attacks, like denial of service attacks, can consume significant portion of network's resources and thus decrease the lifetime of the network significantly or make the network unusable by consuming all available bandwidth. In addition, the attacker may break routing of the ad hoc network by advertising e.g. a very cheap route to itself.

The Packet Level Authentication produces some processing power and bandwidth overhead, but it allows attacks to stopped relatively quickly by detecting and dropping forged, modified or delayed packets. For example, if some node floods the network with garbage or copies of legitimate packets, these packets will be dropped at the next hop and they will not consume resources in the whole network. In addition, using the PLA it is possible to revoke rights to communicate from hostile or compromised nodes. Thus the PLA can prolong the lifetime of the wireless ad hoc network in situations where the network is frequently attacked.

5. Conclusions

The current Internet architecture has not been designed with a security in mind and it is inherently very insecure. Different kinds of attacks like denial of service attacks can be launched easily. There exist many security solutions, but they mostly concentrate on providing end-to-end security, thus they do not protect the network's infrastructure and they do not help if the infrastructure is attacked and is unable to deliver packets.

The PLA aims to solve this problem by providing means for every node in the network to check the validity and authenticity of every packet without having any kind of contact with the sender of the packet. Thus, forged, duplicated or delayed packets can be dropped immediately before they can inflict serious damage to the network or consume additional network's resources. This is important especially in wireless ad hoc networks where resources like bandwidth and energy are very limited. The PLA relies on a public key cryptography and it introduces an additional header on top of IPv6 header.

Preliminary results have shown that the PLA does not produce high computational overhead if a hardware acceleration is used for cryptographic calculations. Elliptic curve cryptography which is used with the PLA allows small public keys that are cryptographically strong, this limits bandwidth overhead of the PLA.

The PLA has also other potential applications beside protecting the network from attacks. For example, the sequence number present in the PLA header can be used for billing and the PLA together with certificates can be used for controlling incoming connections.

References

[1] C. Candolin. Securing Military Decision Making In a Network-centric Environment. Doctoral dissertation, Espoo 2005.
[2] J. Lunberg. Packet level authentication protocol implementation. In Military Ad Hoc Networks, Series 1, No 19, Helsinki 2004.
[3] N. Kobliz. Elliptic Curve Cryptosystems. Mathematics of Computation, Volume 48, pp. 203-209, 1987.
[4] PLA software package [online]. Available from: http://www.tcs.hut.fi/Software/PLA/new/Download.shtml [Accessed 30th April 2007]
[5] A. Satoh and K. Takano. A Scalable Dual-Field Elliptic Curve Cryptographic Processor. IEEE Transactions on Computers, Volume 52, Number 4, pp. 449-460, April 2003.
[6] G. Gaubatz, J. Kaps, E. Öztürk, and B. Sunar. State of the Art in Ultra-Low Power Public Key Cryptography for Wireless Sensor Networks. In proceedings of the third International Conference on Pervasive Computing and Communications Workshops, Hawaii, USA, March 2005.
[7] J. Goodman and A. Chandrakasan. An Energy-Efficient Reconfigurable Public-Key Cryptography Processor. IEEE Journal of Solid-State Circuits, Volume 36, Number 11, pp. 1808-1820, November 2001.