

# Security Log Time Synchronization for High-Availability Systems

Martin Naedele  
ABB Corporate Research  
CH-5405 Baden-Dättwil, Switzerland  
martin.naedele@ch.abb.com

## Abstract

*An increasing number of factory automation systems are connected to the Internet or other public networks, and secured by firewalls, intrusion detection systems (IDSs), etc. In order to detect attacks, correlation of firewalls, router, proxy, and IDS logs is necessary. Successful correlation requires, among other things, synchronized time stamps for all the log entries created by different sources. The automation system usually contains a rather accurate time source, which could be used to derive the time base for all system components, including the above-mentioned security mechanisms. A number of standard protocols exist for time synchronization. In this work it will be shown that these protocols do not fulfill the necessary security requirements. In particular, they open up the automation system network to denial-of-service attacks from the outside. Various design alternatives and the requirements for an alternative time synchronization protocol are discussed.*

## 1. Introduction

### 1.1. Motivation

More and more factory automation systems are connected to the Internet or other public networks for remote maintenance, operation, and real-time data integration with the enterprise-level systems. A security zone including multiple firewalls, proxies, and network-based as well as host-based intrusion detection systems (IDSs) are standard best practice to secure the automation system against intrusions from the public network [8]. According to the principle of time-based security, a successful defense is based on the three elements of delay, detection, and response. Firewalls etc serve as delaying elements. In order to detect attacks, analysis

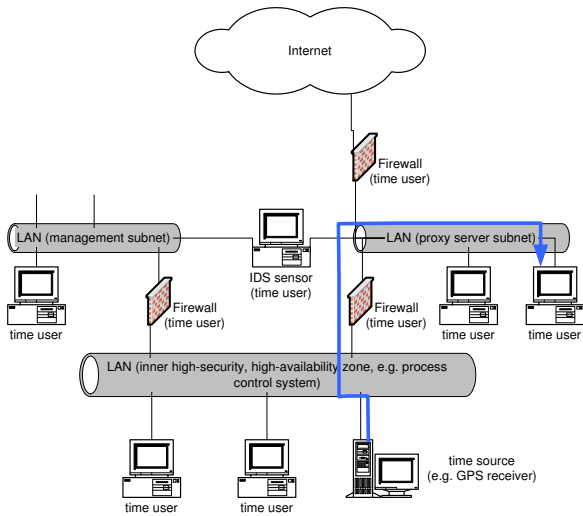
of firewalls, router, proxy, and IDS logs is necessary. Attacks can often only be detected if the corresponding entries in these logs can be correlated, as each concentrates on different cues for an attacks. Successful correlation requires, among other things, synchronized time stamps for all the log entries created by different sources [10]. The automation system usually contains a rather accurate time source, which could be used to derive the time base for all system components, including the above-mentioned security mechanisms.

A number of standard protocols exist for time synchronization. In this work it will be shown that these protocols do not fulfill the necessary security requirements. In particular, they open up the automation system network, the availability of which is highly critical for the functionality of the automation system, to denial-of-service attacks from the outside. Based on this fact, the requirements and design specifications for a new protocol, suitable for time synchronization of subsystems where the time source is located in a high-availability network, are derived and described.

### 1.2. Problem statement

Nowadays, accurate time sources are widely and unexpensively available (e.g. GPS). Many systems, especially critical infrastructure and manufacturing automation systems routinely incorporate such time sources instead of relying on external time providers. With such system, the time source is typically a component of the most inner, most trusted, and most safety/security critical part of the total system. The issue thus becomes how to distribute the time from the time source in the high security zone to clients in outer zones, without opening up channels that allow for attacks on the systems of the high security zone via the time synchronization protocol. This is particular relevant with respect to exposed perimeter elements of the security mechanisms, like routers, firewalls, etc, for

which a non-zero risk of subversion even during a successfully defeated attack is assumed. Figure 1 shows the network topology of this scenario.



**Figure 1. Topology of a multi-segment network where time values are distributed from a source in the most security and availability critical inner zone.**

As a concrete example for the envisioned attack, consider the following scenario: Clients obtain synchronized time from the time server via the popular Simple Network Time Protocol (SNTP) [5] in standard client/server request-response mode. This mode uses UDP, that is, connectionless traffic, therefore a firewall between the clients and the high security zone cannot filter incoming time synchronization requests based on connections established from the inside. Each incoming SNTP UDP packet from a legitimate, but potentially already subverted, host in the outer zone must be allowed to pass in, as far as the firewall rule set is concerned. This opens a potential channel for the attacker to flood the network of the high-security zone with traffic (SNTP packets) and thus execute a denial of service attack on the time server and the inner network, the availability of which may be safety critical.

Ideally, the system should be designed to reject all incoming packets which have not been requested from the inside, that is, which are not responses to previous outgoing messages. Thus nobody from the outside could prescribe the amount of traffic on the internal, high security automation system zone network.

### 1.3. Contributions

This work alerts to the risks of implementing standard time synchronization mechanisms in an availability-critical environment. It then presents and discusses various design alternatives for time synchronization with respect to reducing the threat of protocol-inherent denial-of-service risks.

## 2. Previous Work

### 2.1. Time protocols

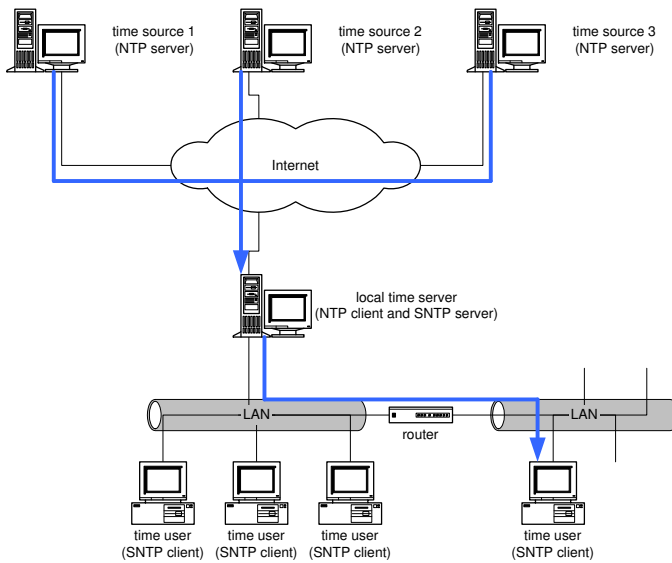
Network time synchronization protocols like Network Time Protocol (NTP) [3] and the lighter alternative SNTP have been developed in order to allow any system connected to a larger network, e.g. the Internet, to make use of accurate clocks available in this network, e.g. provided by national time keeping institutions, to regularly readjust the inaccurate, drifting clocks of computers.

The main problems in this context used to be to find suitable time sources on the network, to get time announcements from them, to ensure redundancy in case of temporary time source and/or network failures, and to detect and correct the influence of transmission delays and faulty time sources. NTP uses sophisticated algorithms for this purpose. Usually only one or, for redundancy, a small number of hosts need to perform the calculations required by a correction protocol like NTP. For time distribution/synchronization in the trusted local network with negligible transmission delay simpler protocols like SNTP or Time [9] can be used. Figure 2 shows the typical network topology of this type of scenario.

### 2.2. Time protocols and security

In [1], Bishop analyzes the security issues of this scenario for NTP v2. He assumes correct time sources and a non-subverted local network, so that all attacks happen in transmission only. The possible types of attacks identified in that work are:

- impersonation of a time server in order to inject a wrong time in the client,
- modification of transmitted messages in order to inject a wrong time in the client,
- replaying of legitimate transmitted messages in order to inject a wrong time in the client,
- delaying of legitimate transmitted messages in order to inject a wrong time in the client, and



**Figure 2. Topology of a multi-segment network which obtains its time information from multiple external time sources**

- intercepting and suppressing messages to prevent synchronization.

In subsequent versions (v3 and v4) of NTP, various cryptographic authentication schemes have been introduced to address these security concerns [6, 7].

All of the attacks treated in [1, 7] thus target the time service as such, they are attacks *on* the time synchronization protocol. The outcome of a successful attack would be a corrupted time value on the targeted host or subnet.

This paper, in contrast, is concerned with attacks *via* the time synchronization protocol. The outcome of a successful attack would be a host or subnet where operational functionality is not available. In the context of automation systems this is clearly the more severe scenario.

### 3. Design Alternatives

#### 3.1. Assumptions

This work makes the following assumptions:

- The system consists of multiple defensive zones (see Fig. 1) and is intended to be able to continue operation even while outer zones are attacked and subverted. The firewalls in front of the automation

system zone are the main means to avert denial-of-service attacks through flooding of the inner zone network by rejecting all illegitimate incoming packets.

- The mission-critical component of the system is the automation system within the automation system network. The external connectivity is "nice to have", but an interruption is not critical with respect to short-time plant performance or plant safety.
- The log consolidation/analysis and intrusion detection functionalities require that the sensors mechanisms like firewalls, HIDS, and NIDS are time synchronized.
- Relative time synchronization between all hosts and applications is the important goal, not absolute time correctness. Required relative accuracy has to be better than the millisecond resolution of, e.g., Syslog timestamps.
- The automation system is the time source for the system, including the security mechanisms (for completeness, some of the following design solutions disregard this assumption).
- The network topology and the configurations of the automation system and the security devices are relatively static, so that dynamic configuration of various networking parameters is not necessary and may be traded off against increased security.

#### 3.2. Time Protocol

Time [9] is a very simple request-response protocol via TCP/37 or UDP/37 for distributing time values with fractions of a second accuracy in a low-latency LAN, disregarding any systematic errors/corrections. The protocol does not have any built-in security mechanisms, so integrity and authenticity have to be provided e.g. via transport layer.

Advantages: none

Disadvantages: With the Time protocol, clients pull time values off the server, thus the connection is initiated from the client in the less secure zone, and therefore the firewall has to allow incoming non-established connections. A denial-of-service vulnerability exists.

#### 3.3. Precision Time Protocol

The IEEE-1588 Precision Clock Synchronisation Protocol for Networked Measurement and Control Systems (PTP) [2] is a recently created protocol for time

synchronization in automation systems. One of the design objectives of this standard was to avoid administration overhead, including any kind of security mechanism. PTP is suitable for underlying network protocols with segment multicast capability. A binding to UDP/Ethernet is explicitly given in the standard.

Advantages: none

Disadvantages: With the binding to UDP and client initiated messages, the denial-of-service vulnerability described above exists also for this protocol. In addition, it appears unlikely that future networking and network security equipment will implement this standard.

### 3.4. (Simple) Network Time Protocol

NTP [3] is a complex protocol containing filtering algorithms for consolidating time from multiple sources in a hierarchical time distribution scheme achieving low millisecond timing accuracy. The more lightweight, but fully compatible, protocol for distributing the time to hierarchy leaf nodes, without the correction capabilities is SNTP [4, 5], with its most recent version being v4 [5]. SNTP is the most commonly supported time synchronization protocol for PCs and network security appliances (firewalls, routers). Both NTP and SNTP use UDP (port 123).

There are several modes of operation of SNTP, which will be discussed in the following subsections. Independent of who initiates the communication, the convention will be used that the client is always the device/application in need of a time value and the server is the application providing the time value.

In [3] an access control and an authentication scheme to be used with unicast and multicast modes of (S)NTP are defined, which addresses issues of server spoofing, message replay, and message modification. The access control scheme is based on IP addresses and thus can not be regarded as secure. The authentication scheme is based on cryptographic message authentication using DES cipher block chaining. It requires key management efforts/protocols not specified within NTP. Recently additional extensions have been proposed to solve the server authentication and message integrity issues based on public key cryptography [6, 7]. As access control is still based on address and port filtering, these new schemes do not solve the problem investigated in this paper.

In the following, various ways of configuring/using (S)NTP will be discussed:

#### 3.4.1 SNTP unicast (client/server) mode

Unicast mode uses outside-initiated conversations and thus induces exactly the denial-of-service vulnerability discussed in Section 1.2.

#### 3.4.2 SNTP unicast mode with compensating controls

SNTP is used in standard unicast mode. The resulting denial-of-service vulnerability through flooding of the automation system network as described above is mitigated by compensating mechanisms:

- High bandwidth network with low operational utilization, which makes exhaustive flooding difficult.
- Switched Ethernet in the automation system zone, which restricts the flooding to the segments on which reside the firewall network interface and the time server, which should execute on a dedicated host.
- Throttling of incoming UDP messages at the firewall, if supported by the firewall product. Various algorithms are possible, such as dropping every  $n$ -th message per source host, dropping all messages within a time window after a passed message per source host, etc. Note that throttled time synchronization messages must be dropped, not delayed, in order to ensure freshness (minimum latency) for passing messages. This behavior differs from strategies chosen for policing other types of traffic.
- Use of a redundant network in the high-security automation system zone of which only one path is connected to the firewalls to the outside. In case of successful flooding of this path the system can switch to the uncongested backup path.

Each of these compensating mechanisms can individually be overcome by an attacker, but together they offer reasonable security for less critical systems.

Advantages:

- Uses the "office IT" standard mode of operation for all devices, applications, and protocols.
- Cost efficient, if the mentioned compensating mechanisms are anyway available in the system.

Disadvantages:

- The vulnerability mitigation by providing enough unused bandwidth is not easily and securely testable: The test has to actually stress the system. The mitigation strategy will break down unnoticed if the internal operational traffic increases

over time or the attacker's injection bandwidth increases.

### 3.4.3 SNTP multicast mode

In multicast mode the SNTP server sends out in regular intervals unsolicited messages containing the current server time. In order to reach subnets other than the one where the server is residing, IP multicast using the multicast group address 224.0.1.1 is used. The clients register for and listen to this group address.

Advantages:

- Based on standards (SNTP, IP multicast)

Disadvantages:

- In a pure multicast mode the client has no means to determine the network propagation delay between the server and the client. The standard suggests to use one or multiple unicast conversations initially to establish these values on the client, and then switch to multicast mode. This, however, would open up the system to the unicast denial-of-service vulnerability described above.
- All the hosts and routers/firewalls in the system need to support IP multicast. According to the relevant newsgroups, this seems to be a non-trivial issue for common firewalls.
- Opening the firewall for multicast and/or running the registration protocol (Internet Group Management Protocol (IGMP)) on the firewall host creates additional complexity and may open up additional security vulnerabilities.

### 3.4.4 SNTP broadcast mode

SNTP broadcast is a variant of multicast using instead of a IP multicast address the broadcast address of the local subnet of the SNTP server. Due to the propagation restriction to the subnet of the server, this mode can not be used in the scenario discussed in this paper, where the security devices and applications reside in multiple subnets separated from the automation system zone with the time server by routers and firewalls.

### 3.4.5 SNTP anycast mode

Anycast mode is used by a client to dynamically discover one of multiple time servers in a network. This mode is not relevant for the discussion in this paper.

### 3.4.6 (S)NTP proxy server on the firewall

On each dual-homed firewall and router a SNTP or NTP proxy is installed that acts as client on the inside and as server on the outside [11]. Within each of the outer subnet either the server-initiated (S)NTP broadcast mode or (S)NTP unicast is used.

Advantages:

- No messages are crossing the firewall. The number of time requests to the time server in the automation system zone is bounded by the number of firewalls directly connected to this inner zone, and their request rate. The denial-of-service vulnerability is thus removed, as long as no attack can be found that would stimulate the proxy on the firewall to increase its request rate on the automation system network.

Disadvantages:

- This approach violates the rule that no services should run on the firewall and makes firewalls vulnerable to SNTP/NTP based attacks [10], which may even result in complete firewall subversion.
- It is required that the firewall product supports/provides an SNTP server in broadcast mode.
- With the firewall acting as server in the unicast mode there is the risk that a denial-of-service attack can be executed against the firewall using the time-synchronization protocol. This is, however, an attack of much less consequence compared to attacking the automation system network in the same way, as it only affects the non-mission-critical external connectivity.

## 3.5. Obtaining time from the outside network

The time data for synchronization for the security mechanism devices and applications are not obtained from the automation system time source on the inner network as shown in Fig. 1, but from a network on the outside, e.g. a national timekeeping institution, according to the standard operation mode of time synchronization protocols (see Fig. 2).

Advantages: none

Disadvantages:

- The firewall between automation system zone and security management zone shown in Fig. 1 does not have direct access to the outside network. Data would have to flow via the automation system zone network, and thus the goal of isolating

this zone from externally induced time synchronization traffic is not achieved.

- This approach is also not suitable for systems that are not connected to an outside network with an authoritative time source - though they might still be connected to an outside network that is dangerous from a security point of view, e.g. a large company intranet.
- The outside firewalls have to be opened to pass (S)NTP traffic, which makes the whole security zone, and together with the previous item the whole system, vulnerable to (S)NTP based attacks.

### 3.6. Time source in each subnet

Each time client host has its own time source, which could either be a very accurate clock set manually at configuration time and certain intervals afterwards, or a GPS or DCF77 receiver. The relative time synchronization of all time client hosts is achieved by synchronizing all clients not to each other, but to the "real time". This approach avoids any time synchronization traffic on the network. Alternatively, one could have one time source per subnet and distribute it within each subnet via a protocol like SNTP or Time.

Advantages:

- No time synchronization traffic would need to cross from one subnet into another, in particular not from a less secure zone into a more secure zone. The denial-of-service vulnerability is effectively removed.

Disadvantages:

- This approach incurs a high cost for providing multiple time receivers or accurate clocks and installing their antennas, and/or the manual time setting and maintenance effort in case of free-running clocks without common time base.

## 4. Requirements for a secure time synchronization protocol

Standard time synchronization protocols like (S)NTP are optimized towards three requirements:

- The organization has no internal precision time sources and wants to use external ones.

- The internal topology of time clients is highly dynamic and time synchronization should incur a minimum of manual configuration and administration effort.
- The relevant security objectives are message integrity and source (time server) authentication.

For the automation system, in contrast, the boundary conditions are:

- There is an internal precision time source. It is part of the most sensitive subnet in the system.
- The system topology is comparatively static.
- Besides message integrity and source authentication it is a major security objective to avoid denial-of-service attacks on the subnet containing the time source.

This clear mismatch in requirements suggests to propose a different time synchronization protocol for the automation domain which trades off management flexibility for increased security.

A custom protocol for time synchronization in safety-critical systems with availability requirements should have the following security features:

- Payload format: In order to reuse existing concepts and implementations, the message formats and time synchronization/filtering algorithms should follow (S)NTP.
- Authentication/integrity: The cryptographic scheme of [6, 7] can be used for server and message authentication. Alternatively, transport level security mechanisms like SSL (requires TCP) or IPSec may be used, if necessary together with hardware-based accelerators.
- Availability/denial-of-service prevention: A connection-based protocol like TCP is used, which allows to filter for established connections at the firewall. A time synchronization conversation is initiated from the time server on the inside. It sends a "permission for time synchronization" message to each time client. Each well known time client - there will only be a relatively small and static number of them - is directly addressed in unicast mode. Once the client receives this permission message, it sends a time synchronization request according to (S)NTP format, which the firewall will pass, as it is part of the connection established previously from the inside. The server will reply with another (S)NTP conforming message, and close the connection. From the message

it sent and received the client can calculate the network delay and account for it when adjusting its internal time based on the server reply. With this scheme, the time server in the security and availability critical zone determines the network load incurred by time synchronization traffic through its scheduling of permission messages.

This scheme, of course, has the consequence that dynamic subscription is not possible, but that all receivers (clients) need to be manually configured at the server.

When designing the protocol, care has to be taken that the exposure window during which incoming messages are allowed is minimized. The stateful firewall has to judge session state from the messages it sees. It normally considers a session as established on seeing the first ACK (second part) of the three-way handshake to open the connection. That means, that after this message it will pass any incoming message for this connection until the connection is closed, independent of any further action from the inside. Also, it will only consider the session as closed once it has seen FIN packets from both sides, which means that a malicious outside communication partner could keep the firewall open by not sending or suppressing the incoming FIN.

To reduce the exposure, the internal server could send a RST directly after its time synchronization reply message. On receiving a RST, a firewall will usually regard the connection as closed after a much shorter time-out period (seconds instead of minutes).

In order to be able to use standard devices and applications like firewalls and IDS, a converter between the proposed custom protocol and a standard protocol like SNTP may need to reside on each subnet (see Fig. 3). This would even remove the need for full manual configuration of all time clients in the server.

## 5. Conclusion

Various design alternatives for providing time synchronization in a high-availability network have been presented and evaluated.

Obviously, a defense-in-depth approach to secure system design prescribes to rely not exclusively on the firewall but to provide additional means like switched Ethernet, redundant network in the inner zone, high transmission capacity margin, etc (see Section 3.4.2).

The problem of time synchronization in an availability-critical network is one example for how the

security requirements and resulting solutions and protocols differ between "office IT" and automation systems, and that new architectural approaches and security mechanisms may be necessary to deal with these differences.

## References

- [1] M. Bishop. A security analysis of the NTP protocol version 2. In *Proceedings of the Sixth Annual Computer Security Conference*, 1990.
- [2] IEEE Standards Organization. Standard IEEE 1588-2002: Precision clock synchronization protocol for networked measurement and control systems. <http://ieee1588.nist.gov>, Sept 2002.
- [3] D. Mills. Network time protocol (version 3) specification, implementation and analysis. RFC 1305, March 1992.
- [4] D. Mills. Simple network time protocol (SNTP). RFC 1769, March 1995.
- [5] D. Mills. Simple network time protocol (SNTP) version 4 for IPv4, IPv6 and OSI. RFC 2030, October 1996.
- [6] D. Mills. Cryptographic authentication for real-time network protocols. *AMS DIMACS*, 45:135–144, 1999.
- [7] D. Mills. Public key cryptography for the network time protocol. Electrical Engineering Report 00-5-1, University of Delaware, May 2000.
- [8] M. Naedele. IT security for automation systems - motivations and mechanisms. *atp*, 45(5), 2003.
- [9] J. Postel and K. Harrenstien. Time protocol. RFC 868, May 1983.
- [10] B. Rothke. The criticality of network time synchronization and its effect on information systems security. *Computer Security Institute Journal*, 14(3):7–16, Summer 1998.
- [11] E. D. Zwicky, S. Cooper, and D. B. Chapman. *Building Internet Firewalls*. O'Reilly, 2 edition, 2000.

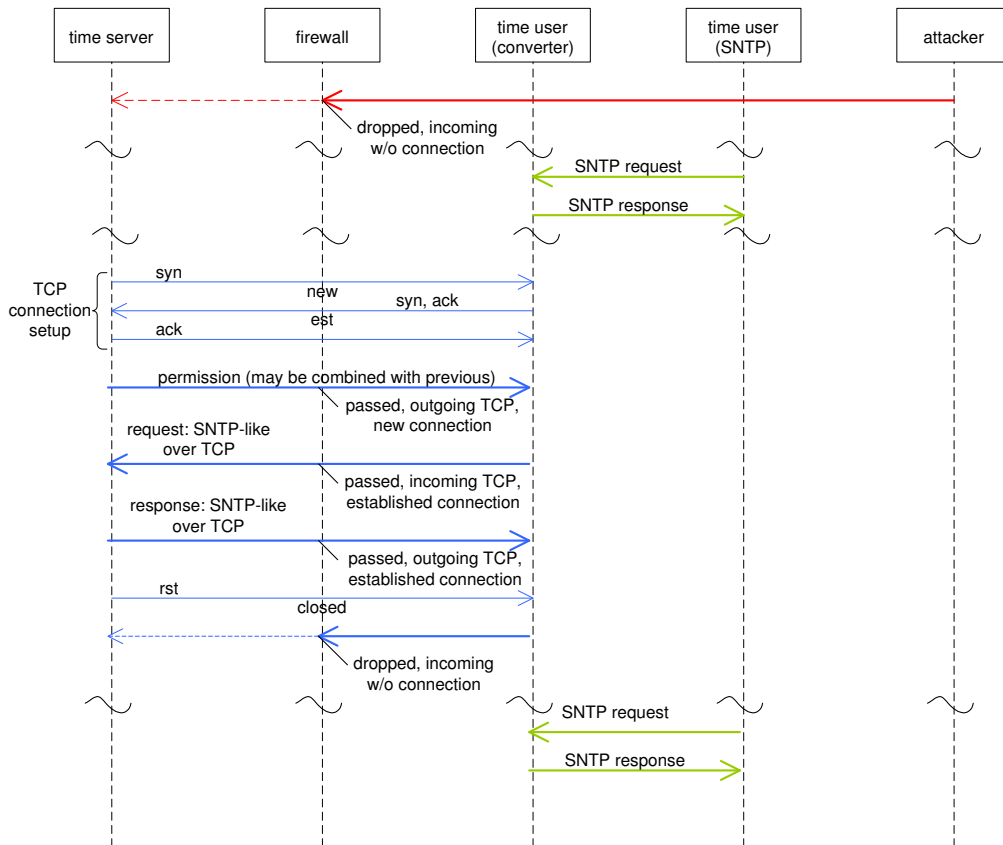
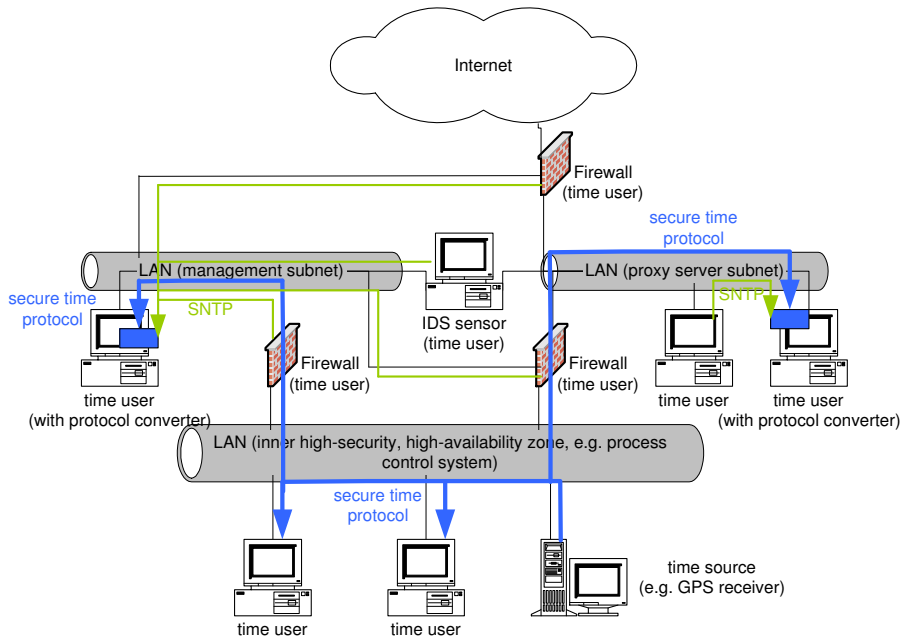


Figure 3. Topology and message sequences for a secure time synchronization protocol.