SPsec101 – Concept

This document sets the scope for SPsec (Small-Packet Network Security Sublayer) by outlining where these networks are used, and which new or updated security requirements apply.

The SPsec specifications are divided into the following documents:

- **SPsec101** – Small-Packet Network Security Concept
  Introduction to the security concept of SPsec.
- **SPsec102** – Small-Packet Network Security Glossary
  Terminology and references used by SPsec.
- **SPsec201** – Small-Packet Network Security Generic Specification
  Network independent specification of the SPsec technology and methods.
- **SPsec301** – Small-Packet Network Security Serial Channel Mapping
  Mapping SPsec to generic serial point to point communication.
- **SPsec302** – Small-Packet Network Security CAN FD Mapping
  Mapping SPsec to CAN FD, supporting CANopen FD and J1939 FD.

Version 1.06 of 12-NOVEMBER-2025, jointly authored by

www.em-sa.com
Embedded Systems Academy GmbH
Bahnhofstraße 17
30890 Barsinghausen, Germany

ivesk.hs-offenburg.de
Hochschule Offenburg
Badstraße 24
77652 Offenburg, Germany

This project has been funded as part of the Central Innovation Program for SMEs (ZIM) by the Federal Ministry for Economic Affairs and Climate Action (BMWK).

Contents

1 Common security challenges for Small-Packet Networks

This chapter is an introduction to small-packet networks, their typical usage in embedded systems and the various security challenges associated with them.

1.1 Introducing Small-Packet Networks

From an industrial use case perspective, small-packet networks are typically used in the lowest layer of the automation pyramid. They connect sensors and actuators to control units in the same or upper layer. Many of these networks are also referred to as fieldbuses, although some of the more modern fieldbuses have packet sizes beyond “small. Physically, these are located deep within a machine, a production cell or within mobile machinery.

More generically the small-packet networks provide communication channels for microcontrollers to off-chip components. The serial or bus interfaces directly available on-chip with many microcontrollers include popular technologies such as I2C, SPI, LIN, Modbus, CAN and CAN FD.

As these communication interfaces are part of many microcontrollers, their usage is widespread among many applications – everywhere a microcontroller is used, that microcontroller at some point communicates off-chip with other components.

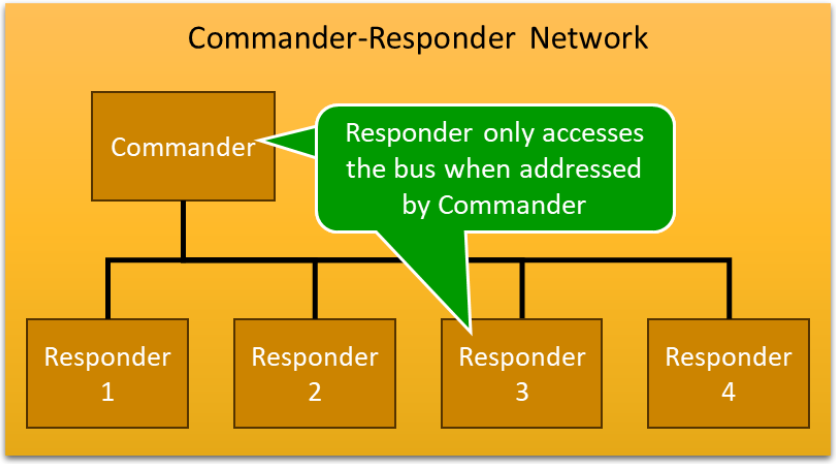
Historically, none of these small-packet networks was designed with security in mind. So neither the microcontrollers implementing the interfaces, nor the protocols used to exchange data provide any resources to make the communication “secure”. The use cases were typically “deeply embedded” into some machinery with no connection to other networks. Some communication would not leave

the local PCB (like SPI or I2C connected I/O) or at least stay within a machine with limited physical access.

For further handling in SPsec, we need to distinguish the two operation principles, the Commander-Responder Single Arbitration Networks and the Multi Access Arbitration Networks.

1.1.1 Commander-Responder Single Arbitration Networks

These network systems feature a single Commander that communicates with one or multiple Responder. If there are multiple Responders, then there is a method for the Commander to address a single Responder.

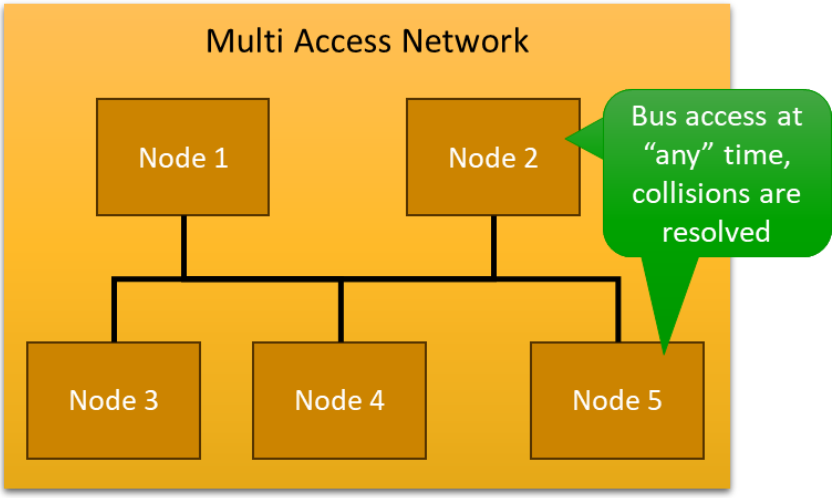


In SPI communication this is done by a chip select line, in I2C or Modbus an address value is used with which the commander can address a single Responder. Responders only communicate, if they received an appropriate command by the Commander. In these systems the Commander is fully responsible for the network access. Commaders decide when and why to address which of the Responders.

When adding security to such systems, each Commander-Responder communication can be regarded as a single point to point communication channel and many well established security methods can be used to protect such communication channels. Typically, it is enough to implement some optimizations or select existing lightweight methods to address the limited resources.

1.1.2 Multi Access Arbitration Networks

These networks do not have any single entity responsible for network access. Instead, all devices connected may request access to the network at any time. If collisions occur, there is an arbitration process ensuring that collisions are detected or even resolved.

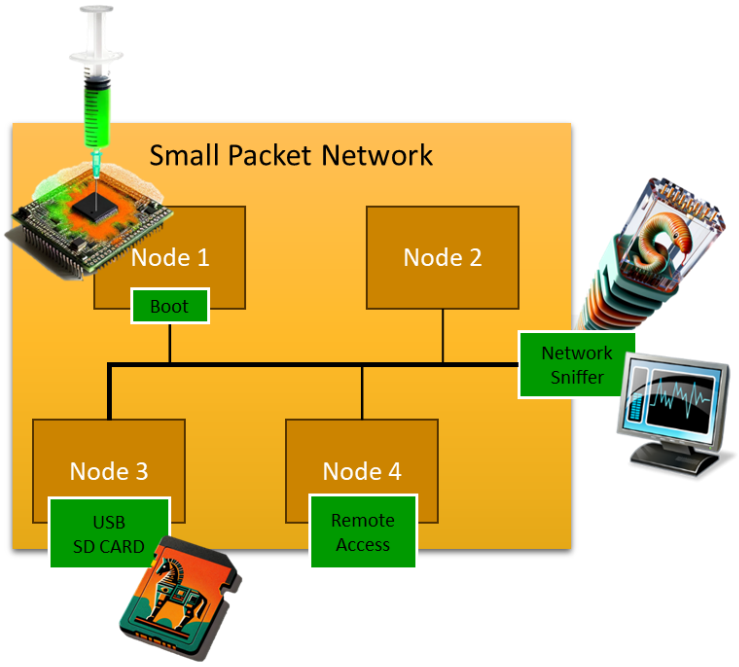


In CAN based systems, the arbitration process uses an address priority. The CAN frame with a higher priority wins the arbitration process.

When adding security to such systems, one of the challenges is the multicast aspect of the communication. Data transmitted by one Node might reach multiple other nodes (multicast), so strictly point to point security mechanisms can not directly be adopted. Preferably some secure grouping functionality is provided.

1.2 Security Threats to Small-Packet Networks

The typical threats the small-packet networks face is illustrated in the figure below.



Anyone with physical access to the networking cable can easily install additional network interfaces that can "sniff" (read) all communication and inject or replay commands and other communication objects.

This can also be used to get access to the bootloader system of connected devices. If not protected further, this would allow an attacker to introduce own malicious code.

Special attention needs to be paid to devices with further network or removable memory interfaces. USB or SD cards are popular attack vectors and every remote access or gateway to other networks is another potential attack vector.

It is important to recognize that if a node is compromised it typically means that also all security measures are compromised. If an attacker has full internal access to a node – running own code – then cryptographic protections are useless if they are part of the code executed in that node.

1.3 Requirements of Acts and Regulations

The IoT (Internet of Things) trend introduced vulnerabilities. Even as a machine manufacturer that purposefully disregards this trend and ensures that its machines are NOT Internet connected, you can not be sure that this remains true. Online shops worldwide offer cheap interfaces that when connected to any of the small packet networks provide full access. Any owner or user of a machine could install such an interface without the machine's manufacturer knowledge.

Worldwide new or updated security related acts, regulation and standards demand security in depth and defence in depth. Depending on risk assessments, threat level and security level desired, unprotected small-packet communication is no longer acceptable for all applications.

Several acts and regulations specifically demand that "all data in rest or motion" needs to be "encrypted and authenticated". Technical details are not part of these documents and related standards will typically use a wording like "must use state-of-the-art security mechanisms".

Several applications might still be able to produce a risk assessment where this is not required, but soon most embedded systems newly designed will have to adhere to these requirements. Small-Packet networks without a security layer might need to be replaced with those that offer one.

In summary, cybersecurity measures for all networks typically include:

- Apply security-by-design and defence-in-depth strategies
- Design to limit attack surfaces
- Ensure that security updates can be applied in a timely fashion
- Provide security related event monitoring and logging
- Protection from any unauthorized access
- Protect confidentiality and integrity of data or commands transmitted or stored
- Protect the availability of essential functions

It depends on the network specific cybersecurity risk assessment how well individual measures have to be implemented. Regulations typically require the use of "state-of-the-art" methods.

1.4 Resource Constraints

As mentioned before, most small-packet networks were invented years ago, without security in mind. At the time, these were designed to communicate sensor data effectively. Like a temperature or

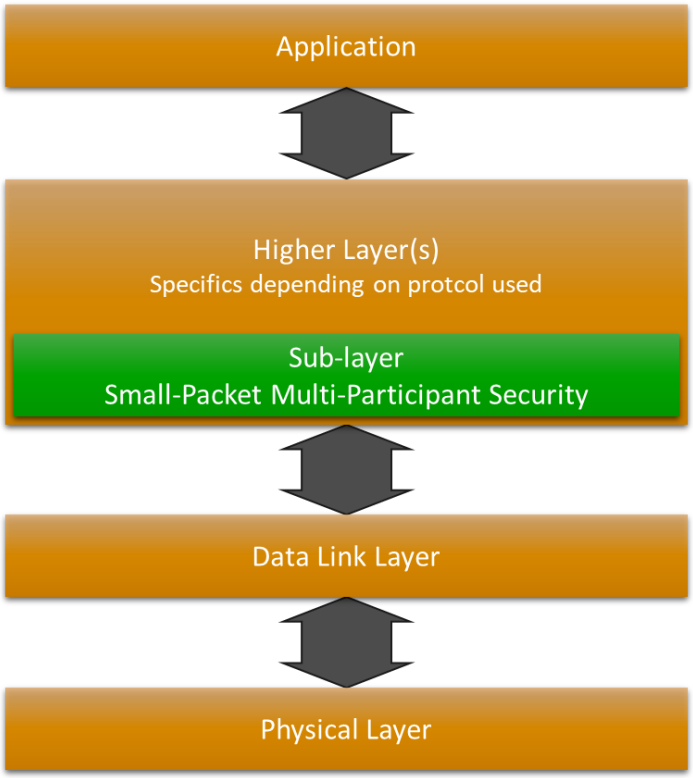
pressure value coming from a single sensor – so a single value of 16 or maybe 32 bit that is communicated cyclically. Looking at Internet style security where the authenticity of data communicated can be protected by a cryptographic checksum of 256 bit we have to realize that many small packet networks have “no room” for such an overhead on security related data.

Secondly, if cryptographic functions are to be used, then it needs to be evaluated if these can be executed in an external hardware security module or if they should execute within the microcontroller also handling the small-packet network communication. Any microcontroller executing cryptographic functions should have a secure non-volatile memory for key storage, a true random number generator and preferably further security features such as a secure code zone and extra code protection preventing code read outs, even if physical extraction methods are used.

In summary, it is challenging to secure an existing non-secure design (e.g. microcontroller with sensor and small-packet network communication interface) without hardware modifications.

1.5 Network Modelling

When talking about security by design or defence in depth for any network technology, one needs to examine security measures at all network layers. Our SPsec Small-Packet Multi-Participant Security Sublayer sits above the Data Link Layer. While this is usually the Network Layer, not all small-packet networks completely implement the 7 layer OSI reference model.



To achieve highest security levels and follow the security by design principle, additional security measures must be considered in the higher layer protocols. Here especially communication for configuration and network management require additional protection. Devices must be able to recognize the integrity of their own configuration.

1.6 Attack Detection and Reaction

Depending on security measures implemented, attacks are detected during different stages of the communication.

1.6.1 After-the-Fact Attack Detection

Methods that use continuous system monitoring report attacks “after they happened”. The attacker’s manipulated messages might get through but are detected and can be reported. In some systems this can be within less than a second of the attack. In others it might take longer until the incident is reported to higher levels, as multiple manipulated messages might be required to detect an attack reliably. This is an acceptable security measure for several use cases, for example in long-term data collecting or where data is only used with some delay because a complete data set is gathered from multiple sources. Some of these security measures can be implemented without cryptographic methods.

1.6.2 Immediate Attack Detection

In cases where even the smallest delay or a single manipulated message might cause severe damage, individual messages need to be properly authenticated in order to discard them before their data content is used. This might be required if critical commands are transmitted that have an immediate effect. This usually requires the use of cryptographic methods where some authentication tag is added to every message transmitted to ensure its authenticity.

1.6.3 Attack Reaction and Reporting

While reactions to attack detections are highly application specific, the minimum requirement is that all data potentially manipulated gets earmarked as “possibly manipulated” or even completely discarded. For after-the-fact detections, this means that there must be a method in place to still report this manipulation for data that already passed through the system.

In cases where immediate detection is used, the messages in question can be directly discarded. Similar to an Internet firewall, manipulated network traffic simply does not pass the security sub-layer.

On a larger scale, it needs to be decided how an entire system should react to such detections and what kind of information is entered into security logs for later analysis. A few isolated manipulated messages should not cause a system shut down, otherwise the scenario would be similar to a denial of service (DoS) attack, where a few manipulated messages would cause an entire system shutdown.

However, depending on the application, if the system is part of some critical infrastructure, where is the line between a few isolated manipulated messages and a continuous ongoing attack that could cause even more damage than only a temporary shutdown?

Unfortunately, there is no generic formular to apply here. For each application, the system designers need to decide where that line is between “acceptable level of discarded messages” and a safe shut-down to prevent even bigger damage.

2 Non-Cryptographic Security Measures

Depending on network technology, various non-cryptographic security measures can be used. This section summarizes those typically available.

This information is provided as a reference, SPsec itself does not define specifics about non cryptographic measures that should be taken.

2.1 Physical Access Limitation

If the small packet network is within a limited space and that space is physical enclosed or potentially even sealed, then physical attacks are limited. Depending on the machinery and application, a risk assessment might show that this provides sufficient security to reach a lower security level. When it comes to higher security levels, this will not be sufficient, but it also might still be a requirement. It provides an additional security layer when applying defence in depth strategies.

2.2 Access Tracking and Event Logging

Standards like IEC 62443 require that system access is limited to authorized users and that accesses are logged providing audit data to determine who did when and what with/to the system. The communication protocols used on small-packet networks do typically not support user access or login, so such tacking would need to be provided on a higher level. For any system configuration or maintenance action, an auditable log should be created. Typically, this happens within the tools used for configuring the system, running diagnostics or loading firmware updates.

When it comes to logging of events, protocols like CANopen already provide a minimal mechanism like the error history. This is a list maintained by each device logging the errors that occurred. However, the information provided in this history is minimal. A proper security log would include date and timestamp and not only errors but also configuration information. Therefore, all this information should go into the auditable log mentioned above.

2.3 Network Zoning and Security Gateways

Standards like IEC 62443 request that networks are “zoned” and access across the zones is highly limited and protected. Therefore, every device or interface connected to the small packet network offering a connection to another network needs to implement a security gateway / firewall. Only with adequate authentication should it be possible to “cross the gateway” from one zone to another.

2.4 Continuous Network Monitoring

Even if this measure cannot prohibit an attack, it is a valuable tool to detect and analyse attacks. By continuously monitoring and analysing all network traffic unusual traffic or patterns can be detected and reported. This is especially easy to implement, when the network traffic expected is very predictable, like data produced on a fixed cyclic time.

2.5 Address Injection Monitoring

Depending on network technology used, it can be possible to detect when an unknown 3rd party injects messages. If participants exchange addressed data units, then the addresses are uniquely

assigned to producers. If injection monitoring is used, all participants shall monitor the network for addresses assigned to them. If a Participant detects an address assigned to itself, then this is an injection detection. The Participant shall produce an injection alert and if a Secure Heartbeat is used, update the status shared in the heartbeat accordingly.

NOTE: if an attacker takes the Participant in question offline, then the Participant's Secure Heartbeat will no longer be produced. Therefore, absence of a Secure Heartbeat must be treated as a potential attack / injection scenario.

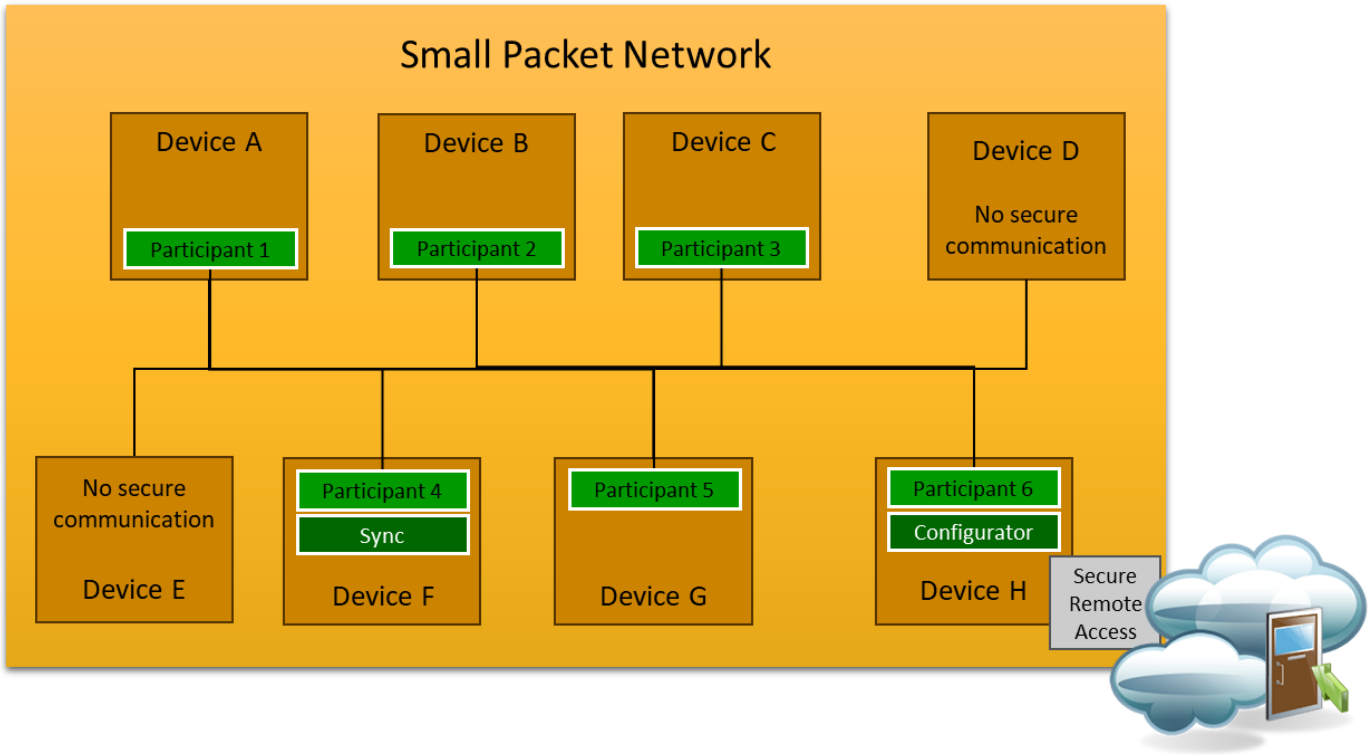
3 SPsec Basic Operating Principles

This chapter outlines the operating principles of SPsec.

3.1 Roles

In SPsec the following roles are defined:

- Participant
  Devices participating in secure communication. Have a unique Participant ID.
- Sync Role
  Participant providing the sync value, either a counter value or a timestamp.
- Configurator
  Participant that can configure other Participants. At any time, only one Configurator may be active. During regular operation, the Configurator is not required.



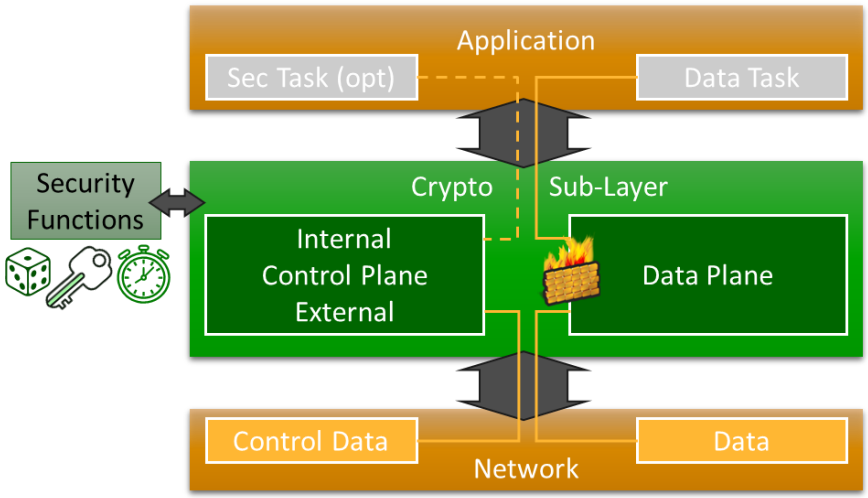
3.2 Control and Data Planes

SPsec divides the communication exchanged into the following types:

- data plane
  The regular communication as it would happen in the network without SPsec. Based on addressed data units. These get converted into secured addressed data units by the SPsec sublayer.
- External control plane
  Sublayer communication required to maintain the security. Synchronization values, heartbeats and configuration communication.
- Internal control plane
  Optional communication path from Sublayer to host or application above sharing SPsec status information and alerts.

The figure below illustrates the communication paths available to the SPsec sublayer. Towards the network it exchanges unprotected data plane and internal control plane communication and towards the network protected data plane and external control plane communication.

In addition, the sublayer requires access to security related functions like random numbers, key storage and timer functions.



3.3 key Management

SPsec security is based on various pre-shared keys with a different level of priority. One of the highest priority keys is the Integrator key which typically gets installed when the network is powered up for the first time. While SPsec does not include full scale key management, it provides configuration functionality to install or remove keys. Sorted by priority, the keys defined are:

- Provisioning key
  Optional device specific key, installed by the manufacturer.
- Integrator key
  Single network system key, installed when network is first powered up.

- Seed key
  Shared key base for Communication key derivations, updated with every maintenance cycle.
- Session key
  Key used during time limited 1:1 sessions.
- Communication key
  Key used for secure grouping.

3.3.1 Pre-determined Communication key Derivation/Rolling

The current Communication key used is automatically derived from the Seed key. key derivation is pre-defined on timer or counter overrun. For example, “on every hour” or “on counter overrun of 10000”. This ensures that a Communication key does not get over exposed while offering a method to refresh keys during regular operations.

3.4 Uniqueness values

To protect repeating values in addressed data units, a uniqueness value is added to the cryptographic functions. This ensures that previous secure commands cannot simply be replayed. Depending on communication mode used, SPsec uses a counter (typically for 1:1 operation) or a synchronized timer (typically for secure grouping operations). Details are defined in the mapping documents.

3.5 Security stamp

Addressed data units are turned into secured addressed data units by adding a Security Stamp to them. The Security Stamp may include SPsec operational information, uniqueness information and an authentication tag. Effectively this means that additional security data needs to be added to the addressed data unit. Depending on network technology it can be challenging to provide that additional information as it potential takes away bytes from the available data field. Sometimes the information can also be partially encoded into the address field. The mapping documents describe in detail how the Security Stamp is structured for the network technology the document applies to.

4 SPsec Cryptographic Security Functions

This chapter summarizes the security functions introduced by SPsec.

4.1 Cryptographic Primitives Required

The cryptographic functions required by any device handling SPsec communication include

- Secure key storage
- True Random Number Generator
- AEAD (authenticated encryption with associated data) function
- KDF (key derivation function)

Which specific cryptographic algorithms are used for AEAD and KDF is defined in the network mapping documents.

4.2 SPsec session for Mutual Authentication and key Agreement

This service provides a secure 1:1 session between two Participants or the Configurator and a Participant. The session opening is similar to a combination of cTLS (compact TLS) and TLS-PSK (Transport Layer Security with Pre-Shared key) session and includes the key derivation and verification. In its most optimized form only 4 times 8 bytes are exchanged during a Hello and Finished stages to derive and confirm a session key.

4.3 SPsec Parameter Authentication

If synchronized timestamps are used, extra effort needs to be taken to authenticate that timestamp. The method introduced by SPsec allows any Participant to request the current timestamp. The timestamp role then creates a response that includes both the timestamp, and an authentication confirmation of the challenge included in the Participants' request.

4.4 SPsec Secure Heartbeat

Depending on configuration, participants produce a Secure Heartbeat. The heartbeat includes basic information about the Participant's state and is authenticated based on current timestamp and Communication key. If a Participant stops producing the Secure Heartbeat, it is considered removed from the network and all addressed data units from it need to be ignored (as they are likely to be injections from an attacker).