IEEE1588 Security Extensions: Requirements and proposed solutions

IEEE1588 Conference 2005 (v2)

Author: Stephan Schüler, Siemens Communications

Schueler.stephan@siemens.com

SIEMENS

Agenda

- Threats
- Countermeasures
- Implementation: Authentication and integrity protection, Message extensions
- Key Management

Threats on PTP systems

Eavesdropping

The attacker has access to the LAN and uses sniffing tools that enables him to eavesdrop and analyze PTP-Protocol (traffic analysis).

This passive attack is a precondition for further active attacks.

Man in the Middle attack

e.g. If a switch / transparent clock modifies a message with the goal of worse the accuracy

Replay attacks

Replaying eavesdropped / recorded PTP-Messages to disturb the PTP-Ports

Denial of Service attack

The attacker sends a flood of messages to overload the PTP-Ports

Masquerading

A hacker uses the identity of an other user to remain undetected.

Misuse of Service through Unauthorized access to management I/F of PTP-Ports,

e.g. change PTP-Parameters with PTP_MM_SET_.

Misuse of Service

through Unauthorized access to management I/F of PTP-Ports, e.g. change PTP-Parameters with PTP_MM_SET_.

through Unauthorized participating in PTP subdomain, e.g. taking over the PTP-Master role and send invalid time information.

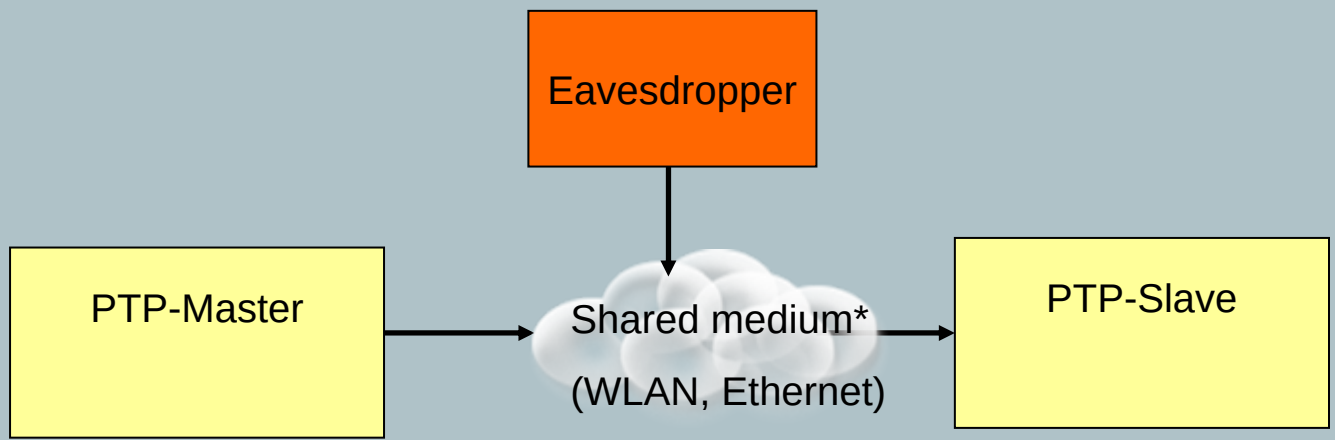
Countermeasures to Eavesdropping

- It is not required to have confidentiality for PTP-Traffic

  (timestamps are no sensitive data)

- Encryption would be the measure, but this makes the timestamping complicated

- Today there are no application areas requiring confidentiality for PTP-Traffic



*) Even in a switched medium the multicast PTP-Messages are send to all switch-ports

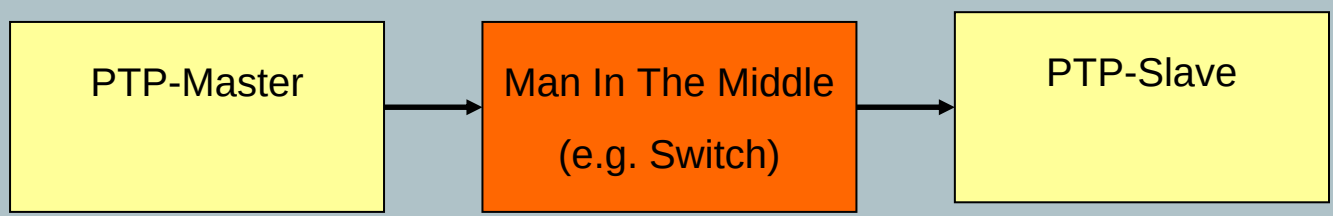
Countermeasures to Man in the Middle attacks

- Can’t be prevented in the PTP-Port, but:

- Checking the plausibility of the timestamps:

  filter algorithms in the PTP slave detecting timestamps

  out of range to improve the robustness



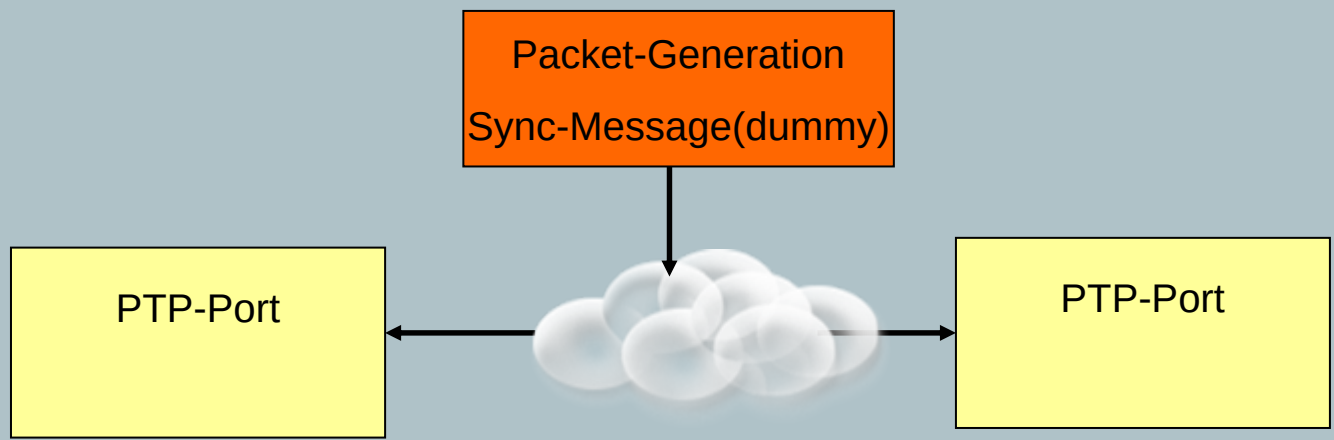
Countermeasures to Replay attacks

- Replay attacks can be identified when the same sequence number occurs twice
- Using sequence number field in the messages.
- Sequence numbers are already included in the header of all PTP-Messages and need not to be included in a security extension
- Checking the plausibility of the timestamps (filter algorithms in the PTP slave)

Countermeasures to DoS-Attacks

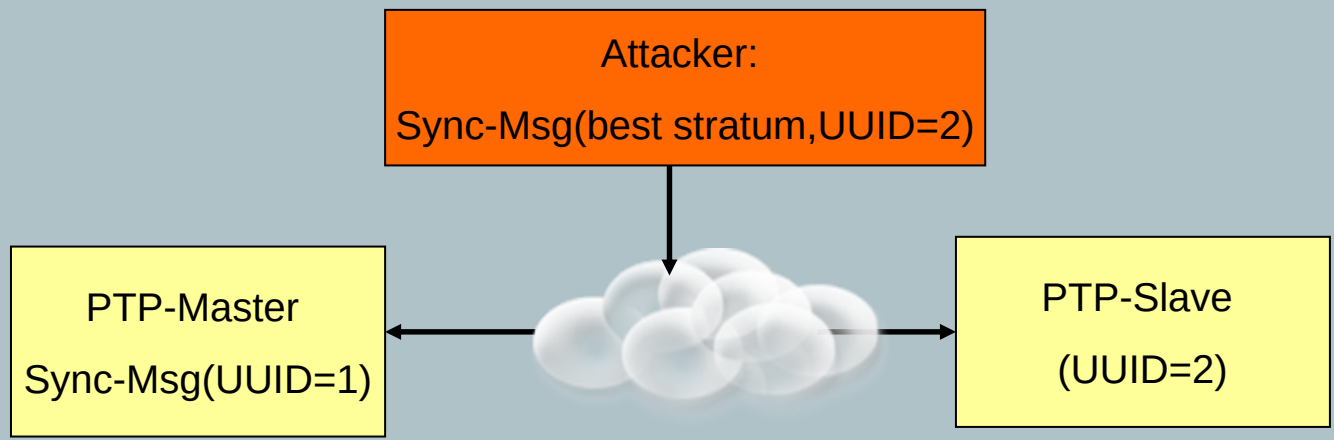
- Can’t be prevented in the PTP-Port, but:
- Send management alarm message (Alarm: “message discarded”),
  if DoS attack was detected (e.g. flood of packets)

- May be a new PTP management message or in case of SNMP a trap



Countermeasures to Masquerading

- Precondition: Attacker has eavesdropped a message from the PTP-Port with UUID=2
- Attacker takes the identity of the PTP-Slave with UUID=2
- Attacker may become PTP-Master



UUID= universal unique Identifier

Countermeasures to Masquerading

- **Authentication** can prevent masquerading

- Option 1: Group authentication of PTP-Ports
  - every PTP-Port gets the same cryptographic key (the „shared secret“)
  - The receiver can verify, that the message comes from one in the group

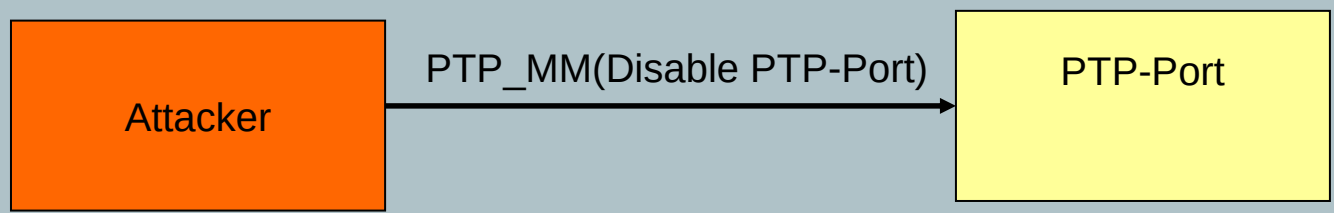
- Option 2: Key management system
  - centralized authentication
  - A central group controller (GC) is required
  - GC can be located in the grandmaster clock
  - (Details described later)

Countermeasures to Misuse of service (by User)

- Authenticate the user who wants access to the management I/F
  - Authentication of PTP_MM messages using

    encrypted hashed message authentication code (e.g. HMAC-SHA1)

  - Use other secure management interface like
    - SNMPv3 (requires a **MIB / private MIB** for PTP)
    - secure Web-I/F with ssl (https)



Countermeasures to Misuse of service (by Device)

- Encrypted Message Authentication Code (e.g. HMAC-SHA1)
- Group authentication (one cryptographic key for one subdomain)
- Checking the integrity of the message
  - recalculating the appended integrity check value (e.g. with HMAC-SHA1)

- Alternatives (if available):
- Configure separate VLANs for PTP ports
  - the switches and devices must be VLAN-aware

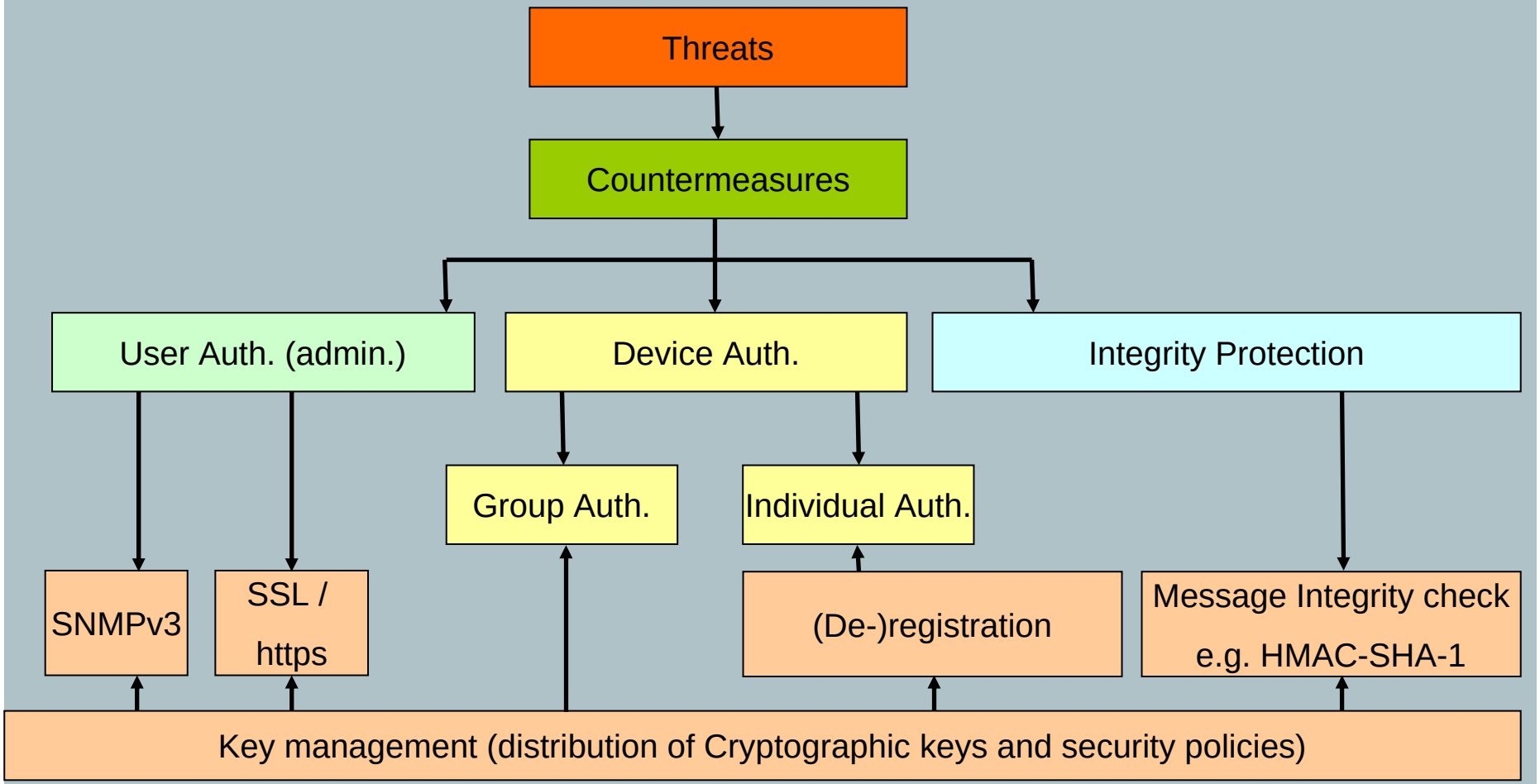
- Use IEEE802.1X for authentication
  - requires authentication server, e.g. RADIUS-Server in the network

Security for PTP (Overview 1)

- Defining message extension fields for security
- Defining cryptographic algorithms and keys to use
- Group authentication of PTP-ports (based on PTP subdomains)
- Secure Administration of PTP-Ports
- Protection against “replay attacks”

- Guidelines:
  - reuse suitable existing security concepts
  - Use standardized algorithms

Security for PTP (Overview 2)



Integrity protection and group authentication

- Integrity protection and authentication between the PTP-Ports
- A one way hash value is calculated over the PTP-Message (payload)
- The hash value is encrypted with the shared secret
- Use HMAC-SHA-1-96

HMAC

- IETF RFC 2104 (1997), *HMAC: Keyed-Hashing for Message Authentication*
- HMAC is used together with an hash algorithm, e.g. SHA1

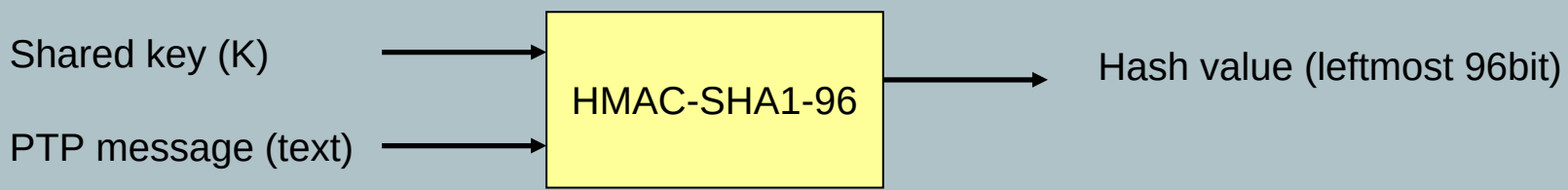
- **Example: HMAC-SHA1-96**

is the truncated 96-bit cryptographic hash value of the 160-bit SHA1 computation.

The 96 leftmost bits of the network byte order representation of the hash value

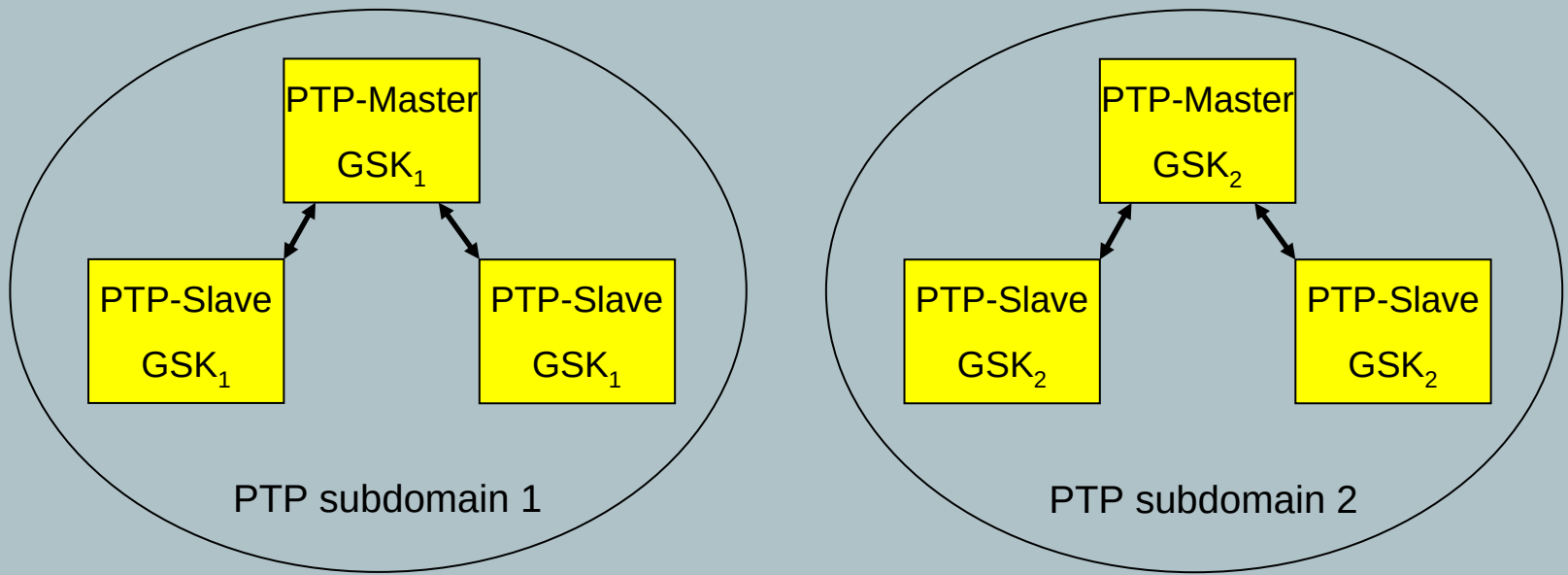
shall be used as the result.

RFC 2104 describes the procedure with the **secret key *K*** set to the ***shared secret***
(20 byte SHA1-hashed password) and ***text*** set to the ***PTP message***.



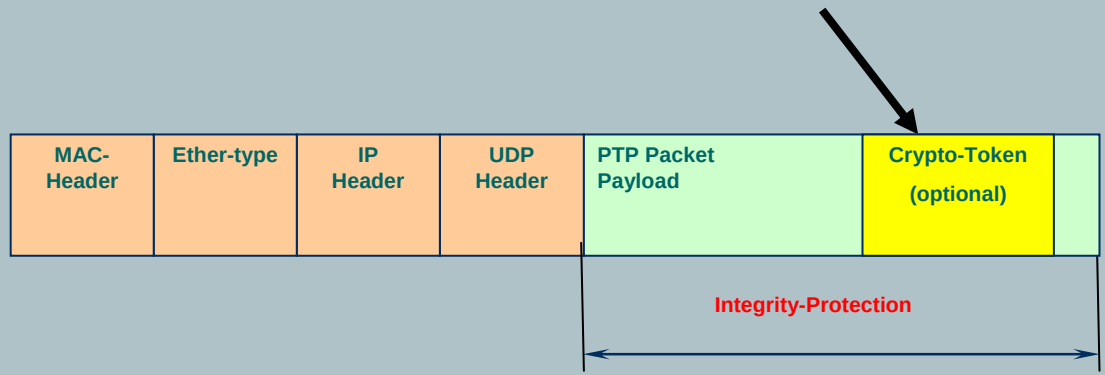
Keys: Shared Secret

- The shared secret is a common Group shared key (GSK=K) for all PTP-Ports of one PTP subdomain
- A symmetric key is used for fast calculation
- Only one key K for the whole PTP subdomain
- The key is stored in every PTP-Port as a 20 byte SHA1-hashed password



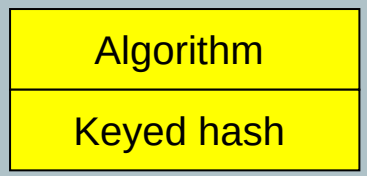
Message Extension: Crypto Token

structure of a PTP packet with the new security extension (Crypto Token):



- Crypto Token can be included in every PTP-Message
- Crypto Token is optional
- It carries the keyed hash value, which is calculated over the complete PTP packet payload
- (without UDP/IP-header) and additional information about the used algorithm
- Before calculation the Crypto Token-Fields will be set to zero

Elements in the Crypto Token



Algorithm to be used with HMAC:

0= SHA-1 (NIST FIPS PUB 180-1)

1= SHA-256 (NIST FIPS PUB 180-2)

Keyed hash value calculated over the complete PTP packet payload (without UDP/IP-header)

truncated to the leftmost n byte

(HMAC-<algorithm>-<n*8>),

n=12. Default for long messages only, e.g. HMAC-SHA1-96*

n=2. Default for short messages, e.g. HMAC-SHA1-16*

*) discussed today, but truncated hash should **not be < n=10 byte** (collision problem: the attacker can make an offline analysis and find a key which produces a hash with the same leftmost n bits.

HMAC:

- IETF RFC 2104 (1997), *HMAC: Keyed-Hashing for Message Authentication*
- The message will be hashed with a hash-algorithm, e.g. SHA1
- The hash value will be encrypted with the shared secret

Key Management: Purpose

First approach (as described before):

- We establish a Group Security Association (GSA), using a group key (GSK)
- We use a symmetric GSK (to achieve low computational workload, …)
- We configure the GSK and the security policies (SP) separately and manually in each GM

SP = manually configured parameters like algorithm to use, Key length, …

Second approach (adding the key management):

- We introduce a key management for automatic key distribution (and rekeying) of the group session key (GSK)
- Requirements:
  - a Group Controller / Key server (GC/KS)
  - establish a secure tunnel between the GC/KS and the GM for the key distribution
- The secure tunnel can be established in two ways:
  - using symmetric keys (the Master Key, MK)
  - using asymmetric keys (public and private keys)

Key Management: Group controller model

General Requirement for PTP key management:

The proposed group key management architecture to be used for PTP is based upon a G**roup Controller Model, described in:**

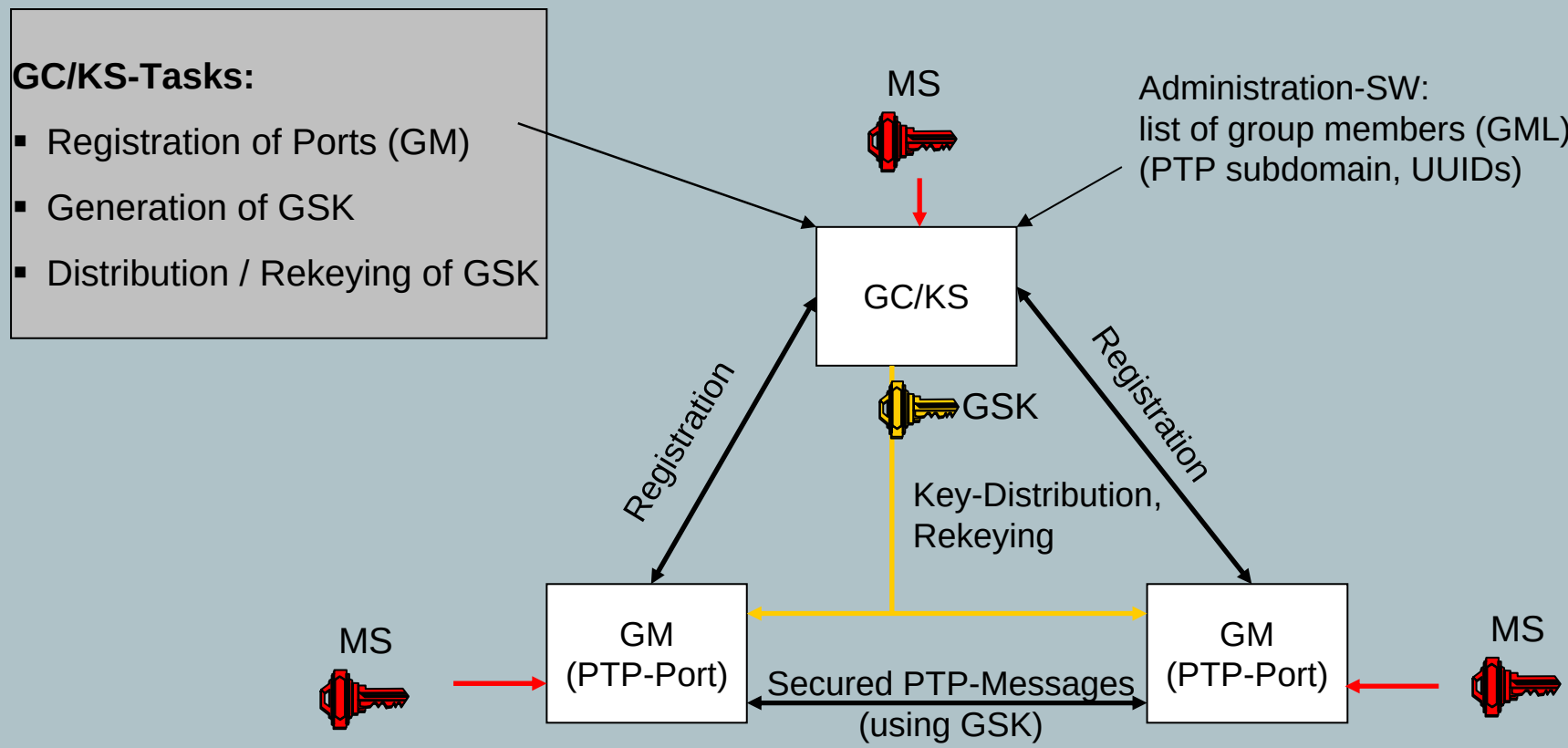
- RFC 2093 Group Key Management Protocol (GKMP)
- RFC 2094 GKMP Architecture
- RFC 4046 MSEC Group Key Management Architecture

The group owner designates a group controller (GC) for member registration and rekeying. with a (**floating**) single group owner as the root-of-trust.

Special Requirement for PTP key management:

It shall be possible that the GC is floating (changing with the PTP-Master change to achieve redundancy

Key Management: Group controller model



- GC/KS: group controller / key server
- MS: Master secret (using symmetric keys). May be public / private key if using asymmetric keys
- GSK: Group Session key

Key Management: Registration Protocol

Purpose:

- Distribution of Group key
- Rekeying of group key

2 Alternatives:

1.) Pull: GM registers at GC to get the GSK (e.g. to request a valid GSK)

2.) Push: CG registers all GMs (GC requires a list of GMs which has to be configured)

Key Management: MIKEY-Protocol (1)

- Published in RFC3830: MIKEY: Multimedia Internet KEYing
- Supports the Group key management architecture (GKMARCH, RFC4046)
- Can be used for **peer-to-peer** and **group** communication
- Defined in Multicast Security Working Group (MSEC WG)
- It is suitable for heterogeneous (mix of wired and wireless) networks

Key Management: MIKEY-Protocol (2)

- MIKEY is a **general purpose** key management protocol
  - It defines the basic messages, and packet blocks
  - Transport of MIKEY in SIP, RTSP defined in RFC3830
  - Transport of MIKEY in TESLA is defined (draft-ietf-msec-bootstrapping-tesla-01.t
  - Transport of MIKEY in SRTP is defined in a separate document
  - Transport of MIKEY in PTP has to be defined
- MIKEY allows also a generic use through dedicated payload types
- MIKEY defines the basic messages, and packet blocks
- MIKEY doesn‘t require a special transport protocol

Key Management: MIKEY-Protocol over PTP

Proposed to use MIKEY-Protocol for PTP:

- Registration of PTP-Ports to GC/KS
- Distribution of GSK and security policies from GC/KS to GMs
- Rekeying of GSK

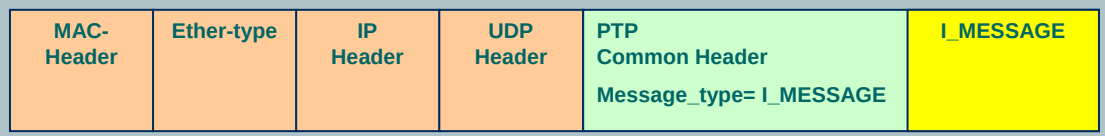
Key Management: MIKEY-Messages over PTP

- MIKEY has only two messages: Initiator Message, Responder Message
- MIKEY messages can be transported over any transport-protocol

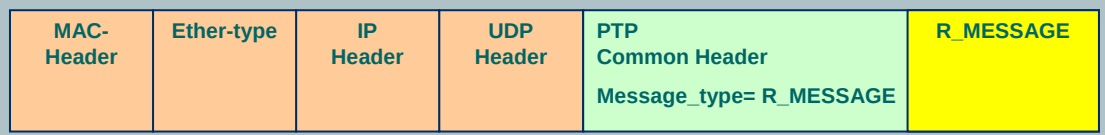
  -> Just use the common PTP-Header for transportation

I_MESSAGE = Header, Timestamp, IDi, IDr, Security Policy, E(encr_key, {TGK}) || MAC

With encr_key = master key and TGK = session key



R_MESSAGE = HDR, Timestamp, IDr, Valid-Flag



Key Management: MIKEY-Modes

MIKEY defines 3 options for the authentication and negotiation of session keys

(All as 2 way handshakes):

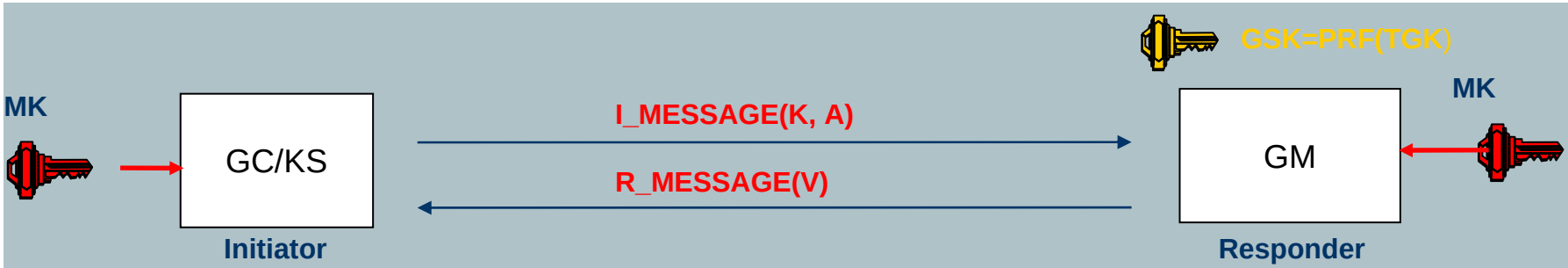
- Symmetric key distribution (pre-shared keys, MAC for integrity protection)
- Asymmetric key distribution (based on asymmetric encryption)
- Diffie Hellman key agreement protected by digital signatures
  - Creates a DH-key, which is used as the TGK
  - cannot be used to create group keys; only single peer-to-peer keys

Two further versions exist, which are not part of RFC3830 itself

- Diffie Hellman key agreement protected by symmetric pre-shared keys
- Asymmetric key distribution (based on asymmetric encryption) with in-band certificate provision

- The default and mandatory key transport encryption is **AES in counter mode** [RFC3711]
- The default and mandatory keyed hash algorithm is **HMAC-SHA-1**

Key Management: MIKEY – Symmetric key distribution



Initialization:

Rand, **TGK**:= Random()

encr-key, auth-key := PRF(MK,…|| Rand)

Protocol execution:

K := [ID_A] || [ID_B] || T || Rand || $E_{encr\text{-}key}$(TGKs [|| KEK]) || {SP}

A := HMAC-SHA1(auth-key, K)

Retrieve **TGK** from K

auth-key := PRF(MK, …|| Rand)

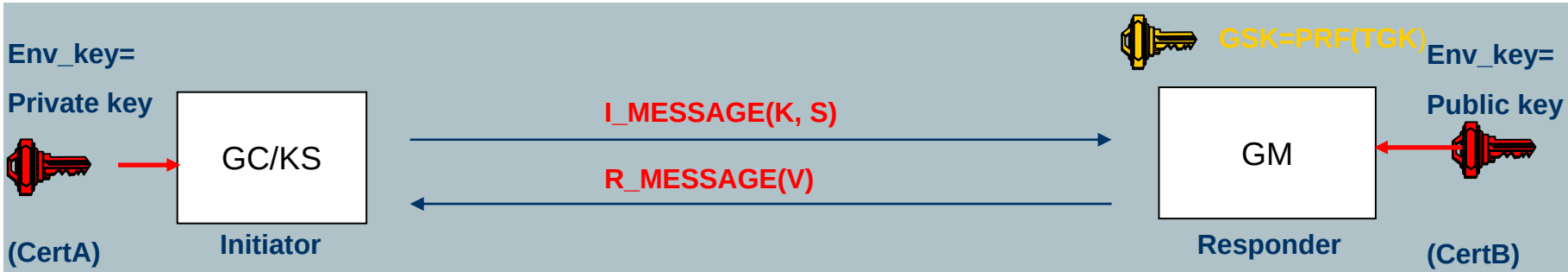
V := HMAC-SHA1(auth-key, ID_A || ID_B || T ||), [ID_B])

GM builds the group session key GSK from TGK

PRF = Pseudo Random Function

- **Pre-shared secret based distribution**
- May proceed as 2-way handshake (optional second message)
- Only the initiator influences the key generation
- No PKI support necessary

Key Management: MIKEY – Asymmetric key distribution



Initialization:

Rand, **TGK** := Random()

encr-key, auth-key := PRF(env-key,…|| Rand)

Protocol execution:

O := E(encr-key(IDA || TGKs || [KEK]))

P := HMAC-SHA1(auth-key, O) ; T=Timestamp SP = Security Policy

K := EPK-B(env-key), O, P, T, Rand, [(IDA || CertA)], [H(CertB)] || {SP}

S := SignSK-A(H(K))

Retrieve **TGK** from K

auth-key := PRF(env-key,…|| Rand)

V := HMAC-SHA1(auth-key, ID_A || ID_B || T ||), [ID_B])

GM builds the group session key GSK from TGK

PRF = Pseudo Random Function

Cert = Certificate

- May proceed as 2-way handshake (optional second message)
- Only the initiator influences the key generation
- PKI support necessary

Key Management: Reasons for Rekeying / Key lifetime

- The group key may need to be changed on demand
  - if it is determined that the key has been compromised
- Maximal lifetime of a key depends on many factors, e.g. used algorithms
- Example key lifetime of 2^{32} PTP-packets:
  - 1000 packets/sec (worse case: short sync with 1ms interval)
  - results in max. 50 days of communication
- A rekeying has to be done before the end of lifetime !
- A rekeying has to be done after N PTP-Sync-Messages
- N should be << 2^{32} packets

Key Management: Rekeying Protocol (2)

- The rekey protocol periodically updates or changes the Group session key (GSK)
- The group members can request re-synch at the GC/KS

  (if their keys expired and an updated key has not been received)
- For a synchronous key change, the rekeying will be done in 2 steps:

  1.) distribute new Goup Session Key (GSK) (and get acknowledged)

  2.) include information on switch-over time
  using the **sequence number of PTP sync_msg**

  E.g.: the new session key is valid starting with PTP_sequence_number 751
- MIKEY: GC/KS: Send I_MESSAGE(encypted(GSK), PTP_sequence_number=751)
  - GMs: Send R_MESSAGE(Valid-Flag)
- In MIKEY rekeying protocol is the registration protocol. Intiator is the GC/KS
- **to avoid implosion problems in large scale installations the rekey message can be sent in multicast (push). This requires that all group members use the same master Key (MK). The Acknowledge mustn't be requested (GC overload). If a GM didn't get the new key, it can request it separately at the GC/KS (pull)**

Key Management: Rekeying Protocol (3)

- Rekeying Message:
  The Group Controller distributes the new GSK to all GMs
- rekeying starts in configured time intervals
- Synchronous activation of the new key after complete distribution
- A sequence number of the PTP-sync_message will be distributed with the rekeying_message
- For efficiency the “MIKEY symmetric key distribution scheme” shall be used for rekeying

Conclusion

For securing the PTP-Protocol the two presented approaches have to be implemented:

- A Group Shared Key (GSK) for message integrity protection
- A Key management is required for larger installations of PTP-netwoks

Next Steps:

Specify the details for transport of MIKEY over PTP:

- Specify the Security Policy Parameters to be transferred in the I_MESSAGE:
  - Hash algorithm to use , Key length
  - Activation timepoint for the new key (use Ptp_sequence_number)
  - Key lifetime (use Ptp_sequence_number)
- Enable Fault tolerance for GC/KS: Locate GC/KS in PTP-Master
  - Dynamically changing master role has to be considered for Registration with pairwise shared secrets

Thank You

Presented by: Stephan Schüler

Siemens Communications

Schueler.stephan@siemens.com

Backup

Abstract

The PTP-Protocol has no security mechanism to protect the transmission of PTP-messages in the current revision of the standard IEEE1588-2002. But there is a strong demand to extend the standard by security mechanism since the most customers already have their security policies which have to be fulfilled if they want to introduce new applications which include the PTP-Protocol. The security policies result from known threats as there are passive and active attacks like eavesdropping, man-in-the middle-attacks or replay-attacks.

The presentation gives an overview for countermeasures to these threats which should be defined in the next revision of the IEEE1588 standard, especially mechanisms for authentication, authorization and integrity protection. The currently active P1588 working group is already taking these requirements into account.

After the overview, a concrete proposal for a security extension will be presented. It covers two main countermeasures: The first one is the integrity protection and authentication of the PTP-messages between the PTP-Ports and the second one is a mechanism for a role based access control which is required for the secure administration of PTP-Ports.

The presented solution describes the required elements for a message extension, concrete algorithms which are suitable for the integrity protection and authentication and the authorization mechanism which is based on shared secrets.