PTP Security Key Management Solutions

Martin Langer
Ostfalia University
of Applied Sciences
Wolfenbüttel, Germany
mart.langer@ostfalia.de

Steffen Fries
Siemens AG, CT RDA ITS
Munich, Germany
steffen.fries@siemens.com

Marius Rohde
MEINBERG Funkuhren
GmbH & Co. KG
Bad Pyrmont, Germany
marius.rohde@meinberg.de

Kai Heine
Ostfalia University
of Applied Sciences
Wolfenbüttel, Germany
ka.heine @ostfalia.de

Dieter Sibold
Physikalisch-Technische Bundesanstalt
Braunschweig, Germany
dieter.sibold@ptb.de

Rainer Bermbach
Ostfalia University
of Applied Sciences
Wolfenbüttel, Germany
r.bermbach@ostfalia.de

2021 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control, and Communication (ISPCS) | 978-1-7281-6484-7/21/$31.00 ©2021 IEEE | DOI: 10.1109/ISPCS49990.2021.9615270

Abstract—**The use of the Precision Time Protocol (PTP) is indispensable in many industrial and commercial areas. It enables the highly accurate synchronization of clocks within a network. In addition to other improvements, IEEE Std 1588-2019 (PTP v2.1) also includes an AUTHENTICATION TLV to protect the integrity and authenticity of PTP messages. However, the distribution of these security parameters is not part of the standard. Since 2020, various solutions are under development that intend to close this gap as automated key management systems. This paper presents the three approaches NTS4PTP, NTS4UPTP and GDOI4PTP being developed in collaboration with the IEEE 1588 Security Subcommittee. This document explains the protocol design and operation of these solutions, followed by a brief comparison of their functions.**

Keywords—PTP, NTS, GDOI, Network Security, Protocols

I. Introduction

The reliability of critical infrastructures such as electrical power distribution or telecommunications strongly depends on the time synchronization of computer systems. Many security mechanisms and process flows require time information as well to ensure the correct functionality. Besides the use of the Network Time Protocol (NTP) providing accuracies in the lower millisecond range, often the Precision Time Protocol (PTP) with time synchronization of the devices in the microsecond or even nanosecond range is required.

For a long time, PTP has synchronized clocks in an unsecured manner, thus making it very vulnerable to attacks on sensitive infrastructures. The first attempt to provide a security solution was made with the release of IEEE Std 1588-2008, which specifies PTPv2. The security approach described there in Annex K provided an experimental mechanism, but it was never implemented and thus not practically relevant. Its later discovered weaknesses and the increasing demand for security led to further development of the security design. With the publication of PTPv2.1 [1] in 2020, the standard now provides the first practical approaches. An essential component is the so-called AUTHENTICATION TLV (AuthTLV), a normative Type-Length-Value (TLV) extension, which ensures the integrity and authenticity of PTP messages. However, the distribution and application of the security parameters needed for this purpose are not specified in the standard. Since manual key management (pre-shared keys) is not a viable option for PTP due to poor scaling, automated key distribution is necessary.

This paper presents security solutions that can accomplish this goal. Since 2020, a total of three solutions have been under development in collaboration with the IEEE 1588 Security Subcommittee. These include two approaches based on the Network Time Security (NTS) protocol [2] (NTS4PTP and NTS4UPTP) and one that extends the Group Domain of Interpretation (GDOI) protocol [3] for PTP (GDOI4PTP). These security solutions differ in protocol flow, complexity, and supported PTP functionalities, so that each approach focuses on specific use cases or environments.

In the further course of this paper, Chapter II provides an overview of the development of PTP security and related work. Chapter III then presents basic information about the security concepts specified in PTPv2.1 and the main features of NTS and GDOI. The subsequent Chapters IV to VI depict the approaches NTS4PTP, NTS4UPTP and GDOI4PTP. Each section contains a brief description of the features and deployment areas, followed by the structural design and current protocol flow. Chapter VII then compares these three security concepts in a short overview. Finally, Chapter VIII discusses the conclusion and the further steps to be taken.

II. Related Work

The Annex K of IEEE Std 1588-2008 (PTPv2) offered message integrity, source authenticity, and anti-replay measures. The solution provided a three-step authentication process with a challenge response (CR) mechanism and a keyed-hash message authentication code (HMAC) for message integrity. However, analyses in [4] [5] [6] revealed various design flaws and weaknesses, for example the insufficient anti-replay protection and the complex process of authentication. Furthermore, the chosen HMAC proved inept for PTP in one-step mode.

Publications followed in which improvements of Annex K and alternative key distribution mechanisms such as the Timed Efficient Stream Loss-Tolerant Authentication (TESLA) protocol, GDOI or NTS were presented. Naiara Moreira et al. summarized and compared these approaches in [4]. During the development of PTPv2.1, many of these proposals were incorporated into the standard and formed the current and normative security solution in clause 16.14 as well as the new Annex P (see III.A). The previous Annex K was therefore removed. The NTS protocol, which was in an early stage of development at the time, was not considered further.

The functionality of the so-called integrated security mechanism described in the PTP standard has already been confirmed by two implementations. Maftei et al. presented a PTPd-based implementation [7] and Shereen et al. integrated the protection mechanism into a Linux PTP implementation [8]. These realizations test both the immediate security and the delayed security processing approaches, but include no authentication or key distribution mechanisms. However, to ensure that the PTP protection mechanism is useful in practice, an automated key management system is required.

978-1-7281-6484-7/21/$31.00 ©2021 IEEE

III. Preliminaries

This chapter first provides an overview of the basic properties of the PTPv2.1 security concept [9] followed by a brief introduction and description of the two protocols NTS and GDOI which serve as basis for the new security solutions.

A. *PTPv2.1 Security Mechanisms*

The current PTPv2.1 standard describes various protection mechanisms in Annex P, divided into four so-called prongs. These can be used individually or in combination to protect the PTP network and the corresponding messages.

Prong A represents the built-in integrated security mechanism ensuring integrity and authenticity of PTP messages. It comprises the AUTHENTICATION TLV, the security processing method and the key management.

The AuthTLV is defined as a part of the PTP security specification in clause 16.14 [1] and is usually located at the end of a message. Besides other parameters and identifiers, it includes an integrity check value (ICV) that ensures the authenticity of the associated PTP message. Based on the parameters it contains, the recipient can identify the associated security policy (SP) and security association (SA). The PTP nodes receive SA and SP from the key management system which contain all the necessary information to construct the AuthTLV and compute the ICV. The ICV is formed over the entire PTP message including the AuthTLV (excl. ICV field).

When securing PTP messages, Prong A distinguishes between delayed security processing and immediate security processing. In the delayed variant, the master does not reveal the key used until a later point in time. TESLA is such a method and mentioned in the standard. Since TESLA seems to be difficult to implement, there is still no realization for PTP let alone detailed specification work.

In the immediate variant, the necessary security parameters and keys are transmitted before the start of the PTP communication. This allows all PTP nodes to check incoming PTP messages immediately. The variant supports Transparent Clocks (TC), but has reduced source authenticity in multicast mode, since all nodes sharing the same SA are able to modify the PTP messages. For the immediate variant, PTPv2.1 mentions the GDOI protocol as one key management system.

The PTPv2.1 specification allows the use of both manual and automated key management (KM) systems. While distributing keys manually may be practical for small PTP networks, it is not a solution for larger networks due to scaling issues and maintenance overhead. An automated KM system solves these problems and enables much better network management. However, PTPv2.1 does not define a specific key management system, but only names possible candidates.

Prong B uses external transport security mechanisms such as IPsec or MACsec. Though both approaches provide authenticity and integrity of transmitted messages as well as the optional encryption of the higher protocol layers, they have their specific limitations when used without prong A.

Prong C concentrates on architectural mechanisms and contains recommendations for increasing the resilience of a PTP network and identifies solutions for mitigating denial-of-service (DoS) attacks. These include redundant network paths as well as additional time sources and grandmaster clocks.

Monitoring the PTP network to detect problems or attacks in the infrastructure is the purpose of methods of prong D. This includes large changes in the measured peer-to-peer connection delays or unexpected offset jumps in the timing information. The use of a watchdog mechanism can prevent or mitigate the consequences of such delay attacks.

B. *Group Domain of Interpretation (GDOI)*

Group Domain of Interpretation is specified in RFC 6407 [3] and defines a key management for group communication. The general approach bases on group members (GM) connecting to a group controller/key server (GCKS), which acts as a key distribution center (KDC). It provides security associations containing the group key and its respective security policies. The security policy relates to group specific information like the cryptographic algorithms used and also lifetime information of the group keys.

GDOI utilizes UDP for transport and performs in two phases. In phase 1 the GM and the GCKS authenticate each other, utilizing peer specific X.509 certificates in conjunction with a public key infrastructure (PKI). Additionally, a connection specific key is established protecting phase 2 of the exchange, in which the actual group key for the target protocol and the connected security policy is distributed.

The initial connection between the GM and the GCKS is always initiated by the GM using the group key pull protocol. Re-keying may be achieved either by reusing the same pull approach or providing updated SA information to the GM by the GCKS applying a group key push protocol. The latter is much faster and only requires a single message sent via an IP multicast address to the group. As the push message does not supply information to the GCKS if the GM has successfully received an SA update, RFC 8263 [10] defines an add-on to enable the GCKS to request a confirmation of the updated SA as well as the acknowledgement message itself.

GDOI primarily targeted the management of IPsec group security associations in multicast communication scenarios. As GDOI is specified in an extensible way, it is also applied in other protocols like for the power system domain as outlined in section VI.

C. *Network Time Security (NTS)*

The NTS protocol is generally a security extension for time protocols. Released in 2020 as RFC8915 [2] it since secures the NTPv4 protocol in (unicast) client-server mode. Many NTP implementations already offer NTS-secured NTP. NTS provides strong cryptographic protection against packet tampering, prevents tracking, scales well, and is robust against packet loss. The protected time protocol remains untouched as much as possible, so an extension for PTP is obvious.

The NTS Key Establishment server (NTS-KE server) distributes keys and security parameters. Communication with the time server (currently implementation-specific) can run in a common service, or can be logically or physically separated. Thus, one NTS-KE server can manage multiple time servers as well, allowing free time server negotiation and load balancing. An NTS-secured communication consists of three parts: The Transport Layer Security (TLSv1.3) handshake, the distribution of the security association, and the secured time transfer. TLS connection and SA distribution together form the NTS Key Establishment protocol (NTS-KE) and act as a key management system. Here, NTS records are used to transmit the payload. TLS is utilized to authenticate and protect these records. After the data exchange the TLS channel is closed and the secured time transmission begins. The application of security parameters to the time protocol depend on the time protocol itself. In the case of PTP, this is defined by the two approaches NTS4PTP and NTS4UPTP.

IV. NTS FOR PTP

One of the solutions to provide an automated key management for PTPv2.1 is NTS for PTP (NTS4PTP). Its development started even before the NTS standard was finalized and initially focused on PTP multicast connections [11]. As development progressed, collaboration with the IEEE 1588 Security Subcommittee began and in early 2021 NTS4PTP was published as an initial IETF draft [12].

A. Overview

The current NTS4PTP draft provides a complete KM solution for PTPv2.1 that follows the immediate security processing approach and supports all communication and transport modes in PTP. It adds new functionalities to the NTS standard and thus allows the independent and secure operation of PTP besides NTP. It provides key rotation, free negotiation of servers and algorithms in unicast mode and group control in multicast connections. In PTP networks, the NTS-KE server is a central entity and communicates with all PTP nodes, regardless of the state (master/slave) of the respective PTP ports. For full coverage of all PTP functions, NTS4PTP defines two different methods, explained in more detail below.

B. The Group-Based Approach (GrBA)

The group-based approach (GrBA) is suitable for securing multicast and mixed multicast/unicast connections. GrBA supports all PTP functions and clock types in the network. In this variant, the PTP network is divided into one or more groups, each of them protected by an individual security association. Only authenticated and authorized PTP nodes are allowed access to these groups, so key distribution takes place only to trusted PTP nodes. The group formation is based on PTP domain and profile so that collisions between domains can be eliminated. In addition, the administrator of the PTP network can define further sub-groups to create group-of-two (Go2) constellations that are suitable for unicast connections and consequently include only the PTP requester and grantor.

The communication flow in GrBA is divided into two phases (see Figure 1) and applies to all PTP nodes. In phase 1, the NTS-KE protocol is used to distribute the group SAs. For this purpose, a PTP node establishes a TLS connection to the NTS-KE server executing the NTS-KE protocol subsequently. The PTP node and the NTS-KE server authenticate each other based on X.509 certificates. After establishing the TLS channel, the PTP node sends a PTP Key Request message. It essentially contains the PTP group that the PTP node wishes to join. The NTS-KE server checks the authentication and authorization of the PTP node and then generates a PTP Key Response message. It contains various security parameters, the group key and the validity period. Thereafter both sides close the TLS channel finishing phase 1. In phase 2, the PTP node applies the received parameters and can check secured PTP messages and generate them itself. Shortly before the SA expires, the PTP node reconnects to the NTS-KE server and requests new parameters by executing phase 1. After the previous SA expires, the new SA can be used directly.

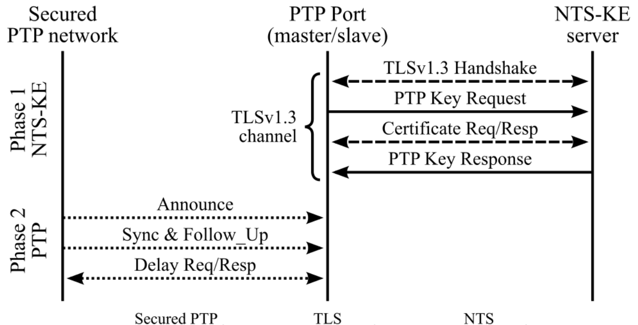

Fig. 1. NTS4PTP protocol flow in the group-based approach (GrBA)

The advantages of GrBA are the simple implementation and support of all PTP features and Transparent Clocks. The group constellation also poses some limitations. Group modes principally provide slightly reduced source authenticity, since every participant in a group can modify the PTP packets. Thus, in the case of a hijacked PTP node, this may threaten the entire group. A Go2 unicast mode scales poorly because of the manual subgroup configuration by the administrator.

C. The Ticket-Based Approach (TiBA)

The ticket-based approach (TiBA) solves the scaling problems of GrBA for unicast connections by avoiding group binding. TiBA implements end-to-end security between two PTP nodes (requester and grantor) and is therefore exclusively suitable for PTP unicast. In addition, TiBA allows free message authentication code (MAC) and server negotiation, which also avoids the need for manual configuration of the unicast master table by the administrator at each individual PTP node. However, the disadvantages of TiBA are its higher complexity compared to GrBA and the currently missing support for Transparent Clocks, since they cannot write a correction field due to end-to-end (E2E) security.

The TiBA communication model uses a ticket system to transfer the SAs. For this purpose, the time servers (grantors) must register in advance with the NTS-KE server. Since NTS did not define a communication between those two, NTS4PTP specifies another sub-protocol called NTS Time Server Registration (NTS-TSR). The communication flow is therefore divided into three phases (see Figure 2).

In phase 1, the grantor connects to the NTS-KE server via a secured TLS channel, signaling the use of the NTS-TSR protocol. Both sides authenticate each other and complete the handshake. Next, the grantor sends a PTP Registration Request message to the NTS-KE server, which includes among other things its addresses and supported algorithms. The NTS-KE server generates a symmetric ticket key storing it together with other grantor information. Then, the NTS-KE

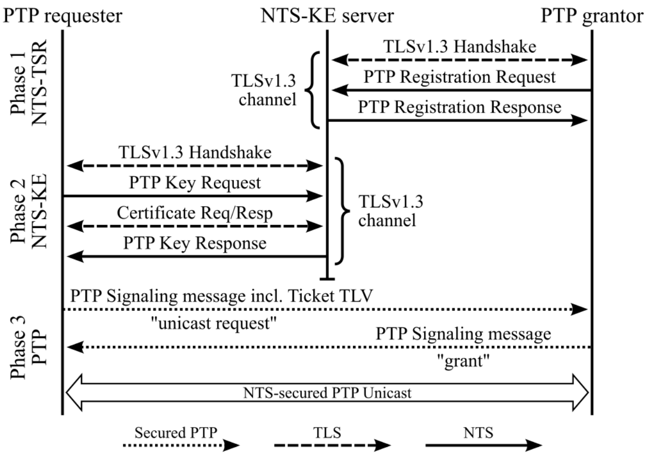

Fig. 2. NTS4PTP protocol flow in the ticket-based approach (TiBA)

server generates a PTP Registration Success message containing this ticket key, validity information and algorithms. Upon receiving this message, both sides close the TLS channel and the grantor is now available as a time server.

In phase 2, a requester can connect to the NTS-KE server requesting an SA for this grantor. The requester executes the NTS-KE protocol and passes the address of the grantor. The NTS-KE server may also assign a grantor. It then generates a PTP Key Response message containing a unicast SA for the connection to the grantor and a ticket. This ticket contains this SA as well and is encrypted with the ticket key that only the NTS-KE server and the grantor know. After reception, the requester ends the TLS connection and is able to contact the respective grantor with a secured PTP signaling message in which the ticket is embedded (phase 3). This PTP message is used to negotiate unicast contracts. Upon receipt, the grantor can decrypt the ticket and use the SA it carries to verify the PTP message and authenticate the sender. The grantor then stores the SA for the duration of its validity, so that resending the ticket in subsequent messages is unnecessary. As in GrBA, in TiBA the grantor and requester must regularly report to the NTS-KE server and collect new parameters.

V. NTS for Unicast PTP

The second solution, NTS for negotiated Unicast PTP (NTS4UPTP) [13], is designed to reuse as much as possible from NTS. A main aspect of NTS4UPTP is to mitigate amplification and replay attacks inherent to PTP's unicast communication model for which NTS do not provide sufficient protection.

A. *Overview*

Unicast PTP uses a stateful subscription-based communication, which contrasts with the request-response communication of NTP. As described above, a PTP client (requester) sends a request to a PTP server (grantor) which sends a grant or deny response subsequently. Then, the server can start sending time information packets to the client if the connection was granted. This established state is called contract in [1]. A contract has a lifetime in which time-messages are sent and both parties can cancel it at any time. Like NTS4PTP this results in three communication phases: Phase 1 with the NTS key establishment, phase 2 comprising the PTP unicast transmission negotiation and phase 3 performing the PTP unicast packet transmission. Figure 3 illustrates the connections between the NTS-KE server, PTP server and PTP clients.

B. *Phase 1*

In general, the NTS-KE phase is identical to the use with NTP. The PTP client connects to the NTS-KE server via TLS, negotiates the Authenticated Encryption with Associated Data (AEAD) algorithm, a Server to Client (S2C) and Client to Server (C2S) key and gets an initial cookie. A key derivation function generates the keys from the TLS connection. The cookie contains both keys and is encrypted with a server key that is only known to PTP server and NTS-KE server. As in NTS the exchange of the server key is not specified. The MAC function of the AEAD algorithm, the keys, a nonce and the PTP message are used to generate the ICV of the AuthTLV in the following phases. The C2S and S2C keys are used according to the direction of message transmission.

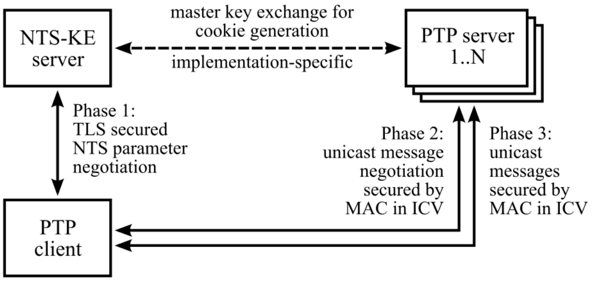
NTS-KE server
master key exchange for cookie generation
implementation-specific
PTP server 1..N
Phase 1: TLS secured NTS parameter negotiation
Phase 2: unicast message negotiation secured by MAC in ICV
Phase 3: unicast messages secured by MAC in ICV
PTP client

Fig. 3. NTS4UPTP infrastructure

C. *Phase 2*

In phase 2 the PTP client must establish a contract with the PTP server. Every packet type needs its own contract and includes rates and durations for sending this type of message. The contracts are stored at the server. NTS4UPTP uses the table of contracts and enhances the information stored to provide a better replay and amplification protection.

The first PTP message of the client to the server includes the cookie generated by the NTS-KE and is protected by the ICV calculated with the C2S. The transmitted cookie is decrypted by the server to obtain the S2C and C2S keys to check and generate the ICVs. If it is a legitimate request the server answers with a nonce included in the message. These nonces are used substituting the cookie in follow-up signaling messages to establish a challenge-response like authentication mechanism. One benefit of this mechanism is the included replay protection. The next expected nonce is always stored in the server and will be used to check the next message received from the client. In addition, the source IP address of the client must be included in the PTP packet as the UDP header is not protected by the ICV.

To protect the first cookie-based request after phase 1 against replay and DoS attacks, an additional challenge-response roundtrip is implemented. The server must not change or cancel any existing contract until a full roundtrip has succeeded. If a timeout of either the server key or the C2S/S2C keys is reached the PTP server tells the client to obtain new keys and a new cookie from the NTS-KE server. If this happens the client must repeat phases 1 and 2.

D. *Phase 3*

If the server has granted the transmission in phase 2, it starts sending the requested messages in the determined rate. Because phase 3 requires a client to successfully complete phase 2, it is sufficient to protect the authenticity of the messages in the transmission phase. The built-in sequence number replay protection of PTP is sufficient as an additional timing check for rollover events is implemented. This check further reduces the risk of replayed packets during a rollover event of the sequence number.

E. *Differences to NTS*

In contrast to NTS, NTS4UPTP drops the objective of unlinkability which results in less cryptographic computation. Because no information in the PTP package is considered confidential, no need has remained to encrypt. Therefore, just the MAC function of the negotiated NTS AEAD is currently in use. In addition, the stateful functionality of PTP makes it possible to simplify the cookie exchange to a nonce based challenge-response mechanism after the initial contract establishment. NTS4UPTP is a lightweight, simple, and fast adoptable approach that improves the key management and the overall security of negotiated unicast PTP. The biggest advantage is the reuse of the NTS-KE server implementations without big adaptions and the resulting infrastructure synergy effects.

VI. GDOI FOR PTP

The third solution bases on GDOI, which can be extended to distribute SA information also for other protocols, applying a group key to protect the target protocol - PTP exchanges.

One further GDOI application originates from the power system automation domain, which can be directly leveraged also for PTP. Here, multicast communication is applied to communicate measurement and control information between protection devices within a substation or between phasor measure units. As the information is crucial for the safety and reliability of the power distribution system, this communication must be secured appropriately. Because of strict timing requirements, the application of asymmetric cryptography was ruled out. The integrity protection of the communicated information is achieved by utilizing a group key in conjunction with symmetric algorithm. This is specified in IEC 62351-6 [14], while the distribution of the SA information is defined as part of IEC 62351-9 [15].

IEC 62351-9 and IETF RFC 8052 [16] enhance GDOI to be applicable in the power system automation domain. Some of these enhancements can be directly utilized for distribution of SA information in PTP multicast scenarios to the PTP instances forming a group. Figure 4 shows the potential interaction of two PTP instances A and B of a common group with the GCKS (also known as KDC).

Phase 1 of GDOI to mutually authenticate the GCKS and the PTP instances can be directly applied as defined in RFC 6407 [3]. It requires all peers to possess a X.509 certificate and a private key as well as sufficient information to verify the certificates. Once GDOI phase 1 is finished, a pairwise key has been established to protect GDOI phase 2 between GCKS and the PTP instances, which provides information specific to the target protocol SA, here PTP.

For GDOI phase 2 a closer look into RFC 8052 [16] is necessary to identify enhancements, which have been done to utilize GDOI in the power system domain. They comprise payload definitions to protect power system communication using IEC 61850 and relate to:

- Identification of a group belonging to IEC 61850 as ID payload,

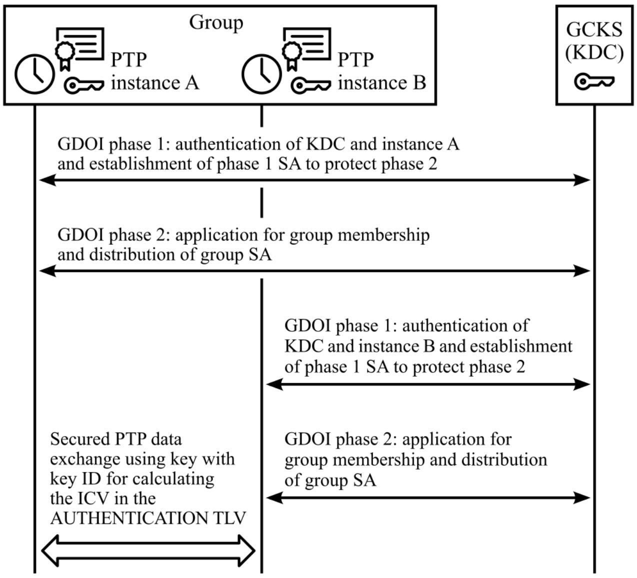
Group
PTP instance A
PTP instance B
GCKS (KDC)
GDOI phase 1: authentication of KDC and instance A and establishment of phase 1 SA to protect phase 2
GDOI phase 2: application for group membership and distribution of group SA
GDOI phase 1: authentication of KDC and instance B and establishment of phase 1 SA to protect phase 2
Secured PTP data exchange using key with key ID for calculating the ICV in the AUTHENTICATION TLV
GDOI phase 2: application for group membership and distribution of group SA

Fig. 4. GDOI application for PTP security (enhanced from [9])

- Provisioning of security attributes for the group key or Traffic Encryption Key (TEK) as SA TEK payload,
- Provisioning of the group keys relating to the policies from the SA TEK payload as Key Distribution (KD) payload,
- Registration of object identifiers (OIDs) for additional cryptographic algorithms, specific payloads, and protocol identification.

Some of these phase 2 enhancements can be directly reused for applying GDOI to provide the group key and the security policy parameter to PTP instances. Some details specific to PTP are to be defined for:

- ID payload carrying an object identifier to identify PTP as the target protocol. Note that when applied in the power system domain, one may re-use the existing IEC 61850 identifier defined by IETF RFC 8052 and treat PTP as the associated protocol.
- SA TEK payload carrying PTP security specific attributes (security policy) as there are:

  Remaining lifetime of the group key, integrity algorithm applied to protect PTP messages, length of the integrity check value, key length information of the disclosed key (when using delayed security) as well as information about used optional fields in the AuthTLV (e.g., enhancement of sequence number information in the header or the reserved field usage), information about the selected processing of the AuthTLV, immediate or delayed processing and KD payload to provide the group key(s) related to the PTP security policies from the SA TEK payload.

RFC 8052 registers additional authentication algorithms directly usable for PTP (e.g. HMAC-SHA256-128 or AES-GMAC-128). Also the newly defined identification payload to allow the selection of the target protocol based on an OID.

As selection of domain specific options of PTP in general and the selection of a key management approach specifically is done by a PTP profile, it is expected that the final definition of the enhancements stated above will be done in the context of a PTP profile selecting GDOI. For the power system domain, this could be addressed in a revision of IEC 61850-9-3 [17], likely in conjunction with IEC 62351-9. There is an ongoing effort to amend the existing Annex P of IEEE Std 1588-2019 with more details about the necessary GDOI enhancements to ease its adaptation for PTP application.

VII. COMPARISON OF THE SECURITY SOLUTIONS

This chapter summarizes the features of the three security solutions and discusses their advantages and limitations. Table 1 provides an overview of the essential key points.

NTS4PTP provides a complete solution for PTPv2.1 and covers all transport and communication modes in PTP. It uses the widely available TLSv1.3 as a phase-1 frame protocol to distribute group SAs or end-to-end SAs (unicast) over it and uses an SA update mechanism, without interrupting the PTP connections. Here, the two modes GrBA and TiBA are employed to serve the different communication types. GrBA focuses on grouping for PTP multicast connections, while TiBA implements a scalable concept for unicast connections based on a ticket solution. The TiBA mode is designed in such a way that only one ticket needs to be transferred between grantor and requester per rotation period. Another advantage of NTS4PTP is the capability to use NTS-secured NTP and

TABLE I. PROPERTIES OF THE KEY MANAGEMENT SOLUTIONS

Supported PTP Feature	**Key Management Solution**			
	NTS4PTP		***NTS4UPTP***	***GDOI4PTP***
	GrBA	***TiBA***		
Multicast	yes	no	no	yes
Mixed multicast/unicast	yes	no	no	yes
Unicast	limited	yes	yes	limited
Transparent Clocks	yes	no	no	yes
Further Properties				
Phase 1 frame protocol	TLSv1.3		TLSv1.3	ISAKMP/IKE
Authentication	PKI		PKI/CR	PKI
Security association	group	E2E	E2E	group
Group control (Multicast)	yes	---	---	yes
Additional PTP TLV	no	yes	yes	no
SA updates	cyclical		cyclical	key pull/push
Srv2Srv comm. spec.	---	yes	no	---

PTP together and independently with a common key management server. In addition, it fully specifies the grantor/NTS-KE server (Srv2Srv) communication (NTS-TSR protocol). As the design defines all necessary security schemes for PTP it appears a bit comprehensive in its entirety. Due to E2E security, TiBA mode does not support TCs. Furthermore, this mode requires a TLV to transport the ticket. On purpose, no support for a forced SA rotation outside the update period (e.g., in case of a detected security incident) is provided, to avoid increasing protocol complexity.

NTS4UPTP offers pure PTP unicast support and also utilizes the NTS protocol as a foundation. Thus, it uses TLSv1.3 as a phase 1 framework protocol as well, which is already supported in many appliances and therefore easier to deploy. The protocol design focuses on a procedure that is as close as possible to the NTS [2] process and consequently provides initial cookies to transport the keys and parameters. This makes it easier to handle, since NTS and NTS4UPTP have an almost similar behavior at the communication level. The NTS key exchange is only needed for the first unicast contract that is established during initialization, after a failure or a timeout of the keys. Another feature is the authentication, which is not only certificate-based but also uses a challenge-response mechanism. Further advantages of NTS4UPTP are the lightweight protocol structure and the also common and independent use of NTS-secured NTP and PTP in a common service. Disadvantages are of course the limitation to PTP unicast mode, the lack of support for TCs, due to end-to-end security and also the need for a TLV to transport the cookies.

GDOI4PTP takes its strength from the GDOI protocol, which is a well-established standard and typically used in combination with IPsec. Since GDOI is already specified as key management for multicast communication in power system automation, GDOI4PTP is particularly suitable here. GDOI4PTP is designed to secure multicast and mixed multicast/unicast connections, which also covers the majority of PTP traffic. Similar to NTS, GDOI also requires a phase 1 frame protocol for the secure transmission of SAs. For this purpose, Internet Security Association and Key Management Protocol (ISAKMP) or Internet Key Exchange Protocol version 1 (IKE) is used, which are also proven standards in practice. As in NTS4PTP, the group formation is done in advance and the authentication of the group members is performed via the public key infrastructure. The advantage of GDOI4PTP is the support of TCs and the better possibilities for group control. For example, key distribution and group control can be done via both key-pull and key-push mechanisms. Moreover, this approach can be used directly in PTP as no additional TLV is needed. However, a known disadvantage is the poor scaling when setting up native unicast connections, as these must be manually predefined as a group.

VIII. CONCLUSION

The security approaches defined in PTPv2.1 can only exist in practice if they are served by a suitable key management system. The solutions considered in this paper meet this requirement, implement the immediate security processing approach and offer individual strengths depending on the application area (see Table I). The best approach for a particular application might depend on the PTP communication mode utilized and the presence of key management technology in the network for other purposes. Since all approaches are still in the specification process, further optimization will follow in the future.

REFERENCES

[1] "IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems," IEEE Std 1588-2019, Jun. 2020.

[2] D. Franke *et al.*, "Network Time Security for the Network Time Protocol," RFC 8915, doi 10.17487/rfc8915, Sep. 2020.

[3] B. Weis, S. Rowles and T. Hardjono, "The Group Domain of Interpretation," RFC 6407, doi 10.17487/rfc6407, Oct 2011.

[4] N. Moreira *et al.*, "Security mechanisms to protect IEEE 1588 synchronization: State of the art and trends," 2015 IEEE International Symposium on Precision Clock Synchronization for Measurement, Control, and Communication (ISPCS), Beijing, 2015.

[5] C. Önal and H. Kirrmann, "Security improvements for IEEE 1588 Annex K: Implementation and comparison of authentication codes," ISPCS 2012, San Francisco, CA, USA, Sep. 2012.

[6] A. Treytl and B. Hirschler, "Security flaws and workarounds for IEEE 1588 (transparent) clocks," ISPCS 2019, Brescia, Italy, Oct. 2009.

[7] D. Maftei *et al.*, "Implementing Proposed IEEE 1588 Integrated Security Mechanism," ISPCS 2018, Geneva, 2018.

[8] E. Shereen, F. Bitard, G. Dán, T. Sel and S. Fries, "Next Steps in Security for Time Synchronization: Experiences from implementing IEEE 1588 v2.1," ISPCS 2019, Portland, OR, USA, Sep. 2019.

[9] K. O'Donoghue, S. Fries, and D. Sibold, "New security mechanisms for network time synchronization prototcols," ISPCS 2017, Sep 2017.

[10] B. Weis *et. al.*, Group Domain of Interpretation (GDOI) GROUPKEY-PUSH Acknowledgement Message, RFC 8263, Nov 2017.

[11] M. Langer *et. al.*, "A Network Time Security Based Automatic Key Management for PTPv2.1", 2020 IEEE 45th Conference on Local Computer Networks (LCN), Sydney, Australia, Nov 2020.

[12] M. Langer *et. al.*, "NTS4PTP - Key Management System for the Precision Time Protocol Based on the Network Time Security Protocol," Internet Draft, draft-langer-ntp-nts-for-ptp-01, Mar. 2021.

[13] H. Gerstung *et. al.*, "Network Time Security for the Unicast Mode of the Precision Time Protocol," IETF Internet Draft, draft-gerstung-nts4uptp-03, Jun 2021.

[14] IEC 62351-6, "Power systems management and associated information exchange - Data and communications security - Part 6: Security for IEC 61850," Oct 2020.

[15] IEC 62351-9, "Power systems management and associated information exchange - Data and communications security - Part 9: Cyber security key management for power system equipment," May 2017.

[16] B. Weis *et. al.*, "Group Domain of Interpretation (GDOI) Protocol Support for IEC 62351 Security Services," RFC 8052, June 2017.

[17] IEC 61850-9-3, "Communication networks and systems for power utility automation - Part 9-3: Precision time protocol profile for power utility automation," May 2016.