An Introduction to MACsec for Electric Protection and Control Devices

Colin Gordon, Prabhpreet Dua, Alex VanDeen, and Justin Clark
Schweitzer Engineering Laboratories, Inc.

Original release July 2022

An Introduction to MACsec for Electric Protection and Control Devices

Colin Gordon, Prabhpreet Dua, Alex VanDeen, and Justin Clark, *Schweitzer Engineering Laboratories, Inc.*

***Abstract*—Industrial control system (ICS) owners and operators are increasingly requesting the implementation of cryptographic protocols into critical energy system devices for securing data-in-motion against common threat scenarios. We argue that IEEE 802.1AE Media Access Control Security (MACsec) is the security solution for critical local-area networks (LANs) due to attractive qualities such as simplicity of design and use, and low maintenance requirements. Further, we demonstrate that MACsec may be combined with other cryptographic protocols into a complete secure transport solution for LANs and routed networks.**

I. Introduction

Media Access Control Security (MACsec) (IEEE Std 802.1AE-2018, *Media Access Control (MAC) Security*) is a standard for security in wired Ethernet local-area networks (LANs) [1]. MACsec offers authenticity and integrity for the entire Ethernet frame, as well as optional confidentiality (encryption) of the Open Systems Interconnection (OSI) Layer 2 (L2) data payload. As an OSI L2 specification, it provides these guarantees for Ethernet-based protocols in the typical industrial control system (ICS) LAN, including IEC 61850 Generic Object-Oriented Substation Events (GOOSE) and IEC 61850-90-2 Sampled Values (SV), which are difficult to secure using other common cryptographic protocols, such as Transport Layer Security (TLS) and Internet Protocol Security (IPsec).

MACsec may be used on its own or be combined with the IEEE Std 802.1X-2010, *Port-Based Network Access Control*, MACsec Key Agreement (MKA) [2] to automate secure key distribution and MACsec participant discovery. In this paper, we perform a technical evaluation of the functions and security benefits of MACsec and MKA protocols, IPsec, and TLS. Next, we compare the various attributes of those protocols with respect to how they secure critical data (their data plane attributes) and how those cryptographic protocols themselves are managed and controlled (their control plane attributes), along with some other general characteristics. Then, we discuss distribution cabinet, microgrid, and substation energy system architectures and evaluate how system operators may best apply MACsec and MKA, IPsec, and TLS in those applications. We conclude with advice to system owners and operators for the careful, condition-based assessment of the practicality of classes of cryptographic protocol implementations in energy system environments.

II. OSI L2 Data Plane Security Via MACsec

IEEE 802.1AE attempts to answer the question of how two or more devices securely communicate over a LAN through the proposal and definition of MACsec, a data plane cryptographic solution that secures inbound and outbound Ethernet packets tied to an interface of a device. MACsec supplies the following primary security attributes:

- Confidentiality (optional): MACsec obfuscates the Ethernet frame’s data payload to prevent unauthorized actors from viewing the contents of the frame.
- Integrity: MACsec protects the frame’s data payload to prevent unauthorized actors from manipulating the frame’s contents or from injecting new frames into the LAN.
- Endpoint authenticity: MACsec proves the identity of authorized hosts joined into a secure relationship in a LAN.
- Replay prevention: MACsec prevents unauthorized actors from attempting to duplicate and transmit Ethernet frames that originated from valid hosts.

MACsec enforces confidentiality, integrity, and endpoint authenticity by using a symmetric cryptographic key called a Secure Association Key (SAK) and enforces replay prevention through a sequentially increasing packet number (PN) attached to the original Ethernet packet.

MACsec offers two different modes of protection: Integrity Only or Integrity with Confidentiality. In the first case, Ethernet frames are transmitted without confidentiality protection but are protected using the other security attributes of MACsec. In the second case, MACsec uses encryption to provide confidentiality, which protects the original Ethernet data payload from snooping eyes.

A. MACsec Protocol Basics

A typical Ethernet Protocol Data Unit (PDU) contains a destination MAC address (DMAC), a source MAC address (SMAC), an EtherType, a payload, and frame check sequence (FCS). MACsec generates an authentication tag called an integrity check value (ICV) for each unique Ethernet frame and appends it to the end of the original PDU. MACsec also adds a MACsec protocol Security Tag (ST) logically after the SMAC of the original PDU, as illustrated in Fig. 1.

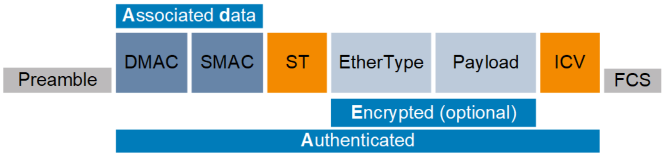


Fig. 1. Original PDU with MACsec ST and ICV attached.

MACsec uses the Advanced Encryption Standard (AES) Galois (pronounced gal-waa)/Counter Mode (GCM) cryptographic standard. The security of AES-GCM relies heavily on unique nonce values for each invocation of AES-GCM using the key (i.e., for each secured Ethernet frame). The implementation of AES-GCM therefore must use unique nonce values for each Ethernet frame. MACsec uses the PN as this nonce value, which we discuss in detail in the following subsections.

B. The MACsec Security Entity (SecY) and Secure Channels (SCs)

A logical instantiation of MACsec on a particular device's interface is called a SecY. The device communicates securely through the SecY with other peers (other SecYs) on a LAN. The following is a short description of how MACsec accomplishes this goal:

- Each SecY has its own cipher suite instantiation, which includes the cryptographic cipher (e.g., the standard's mandatory default suite of AES-GCM-128) and Confidentiality Offset, which indicates how much of each frame's payload will be encrypted.
- Each SecY creates one unidirectional transmit secure channel (TX-SC), which persists for the lifetime of the SecY. The device uses the TX-SC to transmit frames to all other peers on the LAN and thus is essentially point-to-multipoint (P2MP) in nature.
- Each SecY creates one unidirectional receive secure channel (RX-SC) for each MACsec peer on the LAN.

Fig. 2 is an example of a simple MACsec architecture on a LAN with three devices and their associated SecYs: Alice, Bob, and Carlos. Each SecY instantiates one TX-SC and a unique RX-SC for each peer.

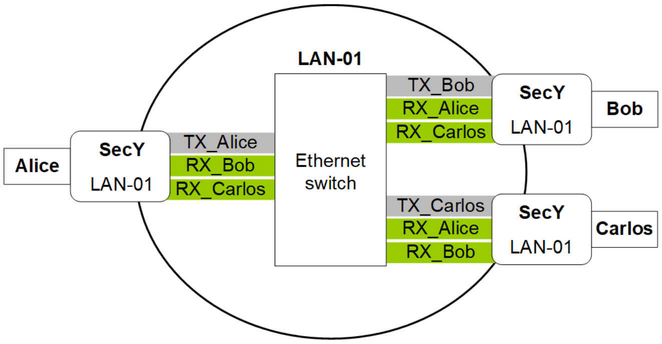


Fig. 2. Illustration of the interaction of MACsec SecYs on a LAN.

C. The Secure Channel Identifier (SCI)

SecYs use a unique SCI to allow themselves and peers to uniquely identify each other on a LAN. The SCI is eight octets in length and contains two distinct components:

- A system identifier is a globally unique MAC address (48-bit value) associated with the device.
- A port identifier is a 16-bit integer that generally represents the physical port of the device on which the SecY resides.

The standard implementations include the SCI in the MACsec ST attached to the original PDU on egress from the SecY. However, point-to-point links with only two SecYs do not explicitly require the SCI to be included in the ST because the TX-SCI can be gleaned from the SMAC when a default port identifier is used [1].

D. MACsec Security Associations (SAs)

Each SC has at least one and can have up to four unidirectional SAs. Each SA consists of two important parameters that relate to the SA:

- The SAK provides endpoint authenticity, integrity, and (optional) confidentiality to secure each SA.
- The PN provides replay prevention and a unique nonce value for each separate PDU, as required by AES-GCM.

Unlike the SC, each SA is transient in nature and persists only as long as a SAK, PN tuple are in use. The SecY distinguishes between SAs using an association number (AN) in the range [0, 3] (hence, each SC can have up to four SAs). Although there is no prohibition against using the same SAK across different SAs (and even across different SCs), successive SAs in the same SC must be identified by a different AN (termed the Secure Association Identifier [SAI], which is shown in Fig. 3) to allow the SecY to unique identify the specific SA.

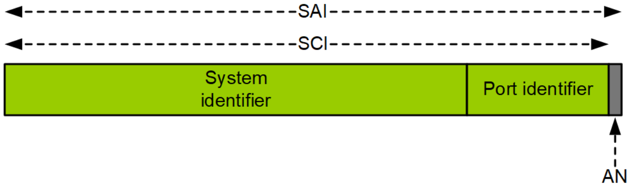


Fig. 3. The SAI [1].

Because the 32-bit PN acts as the nonce value for AES-GCM, the SC must install a fresh SA (SAK, PN tuple) before the PN limit is reached. The MACsec standard requires that the transition between SAs happen seamlessly without the loss of frames for the duration of an "interleave period" [1]. The risk of PNs expiring varies widely on a network; an energy system network supporting a moderate amount of supervisory control and data acquisition (SCADA) and engineering access communications does not use many Ethernet frames. For example, if a device receives 100 Ethernet frames (packets) over a 60-second period, then the PN will reach its maximum in roughly 81 years, as shown in (1).

$$\frac{2^{32}}{60\text{ min}\bullet 24\text{ hr}\bullet 100\text{ PPS}\bullet 365\text{ days}} \tag{1}$$

However, in the case of an IEC 61850-9-2 SV stream at a rate of 4,800 PPS, the PN will reach its maximum in just 10 days, as shown in (2).

$$\frac{2^{32}}{3{,}600\text{ s}\bullet 24\text{ hr}\bullet 4{,}800\text{ PPS}} \tag{2}$$

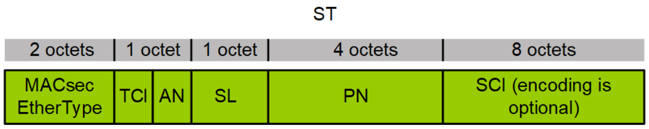
The MACsec standard recommends the MKA protocol as the primary method for renewing SAKs.

E. MACsec ST and ICV

As previously mentioned, MACsec prepends an ST and generates and appends a trailer (ICV) to Ethernet PDUs that it secures.

The ST consists of the following parts (as shown in Fig. 4):

- The MACsec EtherType (88-E5) indicates to Ethernet middleboxes (switches) and hosts that the Ethernet frame is MACsec-secured.
- Tag control information (TCI) and short length (SL) inform a SecY about how to handle the MACsec-secured frame.
- AN, PN, and SCI provide security and inform the receiving host about the nature of the sender and MACsec key.



ST

2 octets	1 octet		1 octet	4 octets	8 octets
MACsec EtherType	TCI	AN	SL	PN	SCI (encoding is optional)

Fig. 4. MACsec ST [1].

The SL is used only when the original Ethernet frame is shorter than a certain length (such as in the case of Address Resolution Protocol [ARP] queries). The TCI contains information about whether an SCI is used, whether an Ethernet Passive Optical Network scenario is in effect, or whether confidentiality is in effect.

F. MACsec Encryption Modes

MACsec allows the device to specify whether the original Ethernet frame will be both encrypted and authenticated or simply authenticated. MACsec's two primary encryption modes are No Confidentiality, where the original Ethernet frame is authenticated but not encrypted, and Confidentiality with Offset = 0, where the original Ethernet frame is both encrypted and authenticated.

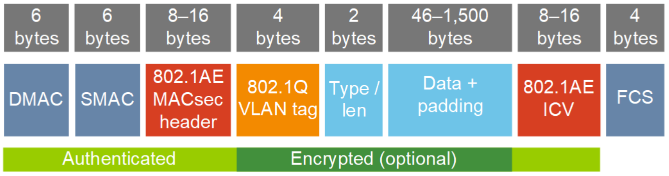
MACsec is a nonroutable protocol. Traditional routers change the original SMAC and DMAC, which causes a mismatch between the ICV generated on egress and the ICV generated on ingress. If users desire to route MACsec-protected frames between two or more MACsec-aware devices, then those devices must implement additional L2 tunneling methods, such as a Virtual Extensible LAN, a Generic Network Virtualization Encapsulation, or an L2 Tunneling Protocol [3].

G. MACsec Security Guarantees

MACsec provides authenticity and integrity guarantees for all portions of the original Ethernet frame, including the MACsec ST, the original SMAC, and the original DMAC. (The only portion of the frame not protected is the FCS, which would be redundant.) Any attempted modification of the new MACsec-protected Ethernet frame is mitigated by the ICV; the SecY receiving the frame detects any attempt to change either the ICV or the MACsec-protected Ethernet frame by performing the decryption and verification functions.

H. Virtual Local-Area Network (VLAN) Handling

Because MACsec secures the entire original Ethernet frame, MACsec treats VLAN EtherTypes and headers (including QinQ) like any other non-VLAN Ethernet header, as shown in Fig. 5.



6 bytes	6 bytes	8–16 bytes	4 bytes	2 bytes	46–1,500 bytes	8–16 bytes	4 bytes
DMAC	SMAC	802.1AE MACsec header	802.1Q VLAN tag	Type / len	Data + padding	802.1AE ICV	FCS

Fig. 5. MACsec-secured Ethernet frame with VLAN header.

Devices implementing IEC 61850 GOOSE attach a VLAN header to the frame so that Ethernet middleboxes can prioritize traffic or supply other filtering functions. This is a common scenario in substations using IEC 61850 GOOSE for protection or automation functions. In these scenarios, Ethernet middleboxes may remove the outside VLAN header or may leave it attached before MACsec processing occurs. In the latter case, manufacturers can update the MACsec authentication functions to automatically detect and bypass the addition of the VLAN header to the ICV generation and validation function through a process called "VLAN-in-the-clear" [4]. In those scenarios, the VLAN header is attached immediately after the SMAC header and before the MACsec ST (in contrast to the VLAN header placement in Fig. 5).

III. OSI L2 MACsec Control Plane Via MKA Protocol

A. The Control Plane and Introduction to MKA

Whereas MACsec is a protocol designed to securely transfer data between devices [1], MKA is a set of extension protocols to facilitate and automate the commissioning, management, and scalability of MACsec on a LAN [2]. In this case, MKA serves as the control plane protocol that facilitates the operation of MACsec without interrupting the flow of information in the data plane. Acting on the control plane, rather than the data plane, allows working details of the underlying system to be hidden from its configuration. This can help facilitate development and maintenance without disruption and keeps the data plane as simple and fast as possible. To aid in this, MKA provides the following ease-of-use attributes:

- Network discovery: Hosts can discover other MKA-supporting devices attached to the same LAN.
- Mutual authentication: Hosts can confirm mutual possession of a Connectivity Association Key (CAK) and provide common access to a Connectivity Association (CA).
- Key management: MKA automatically generates new SAKs for all authorized MACsec hosts joining a CA and rotates SAKs when they near expiration. MKA can also distribute new CAKs to all authenticated hosts who possess a previously shared CAK.

- MACsec parameter management: MKA enables the automatic creation of SCIs and facilitates the synchronization of the cipher suites and confidentiality offsets used by all authorized MACsec hosts.
- Bounded receive delay: MKA can guarantee that a frame is not delivered after a known bounded time (typically 2 seconds) with a lowest acceptable PN.

MKA automates most of the commissioning and management overhead of the MACsec. It does this initially by facilitating commissioning between devices with a key server election process and makes its initial connection by using pre-shared keys (PSKs). It then seamlessly monitors and rotates keys for all participants without losing any data secured by MACsec by communicating on a channel separate from the data plane. It does all this without excessive input by any individual managing the network. Once MKA is configured between Key Agreement Entities (KaYs), it runs and exchanges SAKs, and if needed CAKs, for as long as the network exists without issue or maintenance.

B. *MKA Discovery and Initial CAK Value*

All KaYs participating in MKA first identify each other on a network by recording MACsec Key Agreement Protocol Data Unit (MKPDU) broadcasts. All active KaYs then elect a key server by choosing the participant advertising the highest key server priority (an 8-bit integer) encoded in each MKPDU. If two or more KaYs are broadcasting the same value, then the MAC address is used as a tiebreaker. Lower numerical values for key server priority and MAC address are accorded the highest priority. All peers are then grouped into the same CA by owning the same CAK.

There are two ways to distribute a CAK, which also includes a corresponding public Connectivity Association Key Name. The first way is to manually enter the pairwise (or group, if more than two devices are being initialized) CAK on each peer by way of a PSK. The other is to distribute the pairwise CAK while it is protected by AES Key Wrap by a Master Session Key that has been previously established by existing IEEE 802.1X Extensible Authentication Protocol methods. The CAK can be predefined, derived from something such as a user password, or generated at the time of distribution using an available cryptographically secure random number generator (RNG). The Connectivity Association Key Name is always distributed in cleartext and serves as an identifier of the CA and CAK it is associated with. All peers in possession of the same CAK are now considered part of the same CA, and the CAK may also serve as the root key for all further key generations among the CA, as illustrated in Fig. 6. To lessen the chance that the initial pairwise CAK is a weak value, we recommend that it be rotated to a new randomly determined value before distributing further cryptographic keys associated with MACsec.

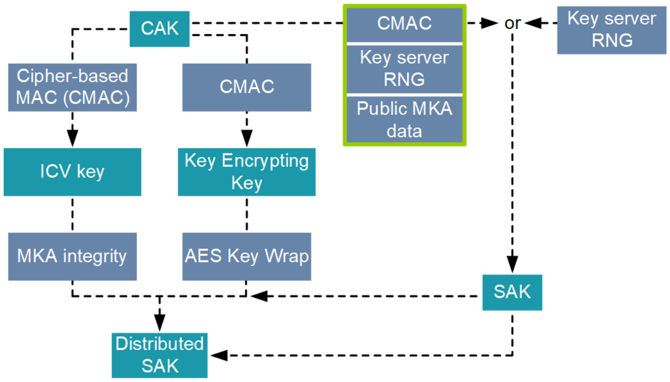


Fig. 6. MKA key hierarchy [2].

Each participating KaY is also responsible for maintaining and advertising a list of potential peers and a list of live peers. Potential peers include all individual hosts that have sent an MKPDU that the participant received. Live peers are determined by all other participants believed to be in current possession of the CAK and their liveness, determined by whether they received a sufficiently recent MKPDU broadcast in combination with a message number that is greater than the last transmitted by that peer. After any MKPDU is received, any future MKPDU with a prior message number is discarded.

C. *Establishing MACsec Over the MKA CA*

The next step in establishing a MACsec-secured channel is for the MKA key server to distribute a SAK to each participant in the CA. The SAK can be derived from the CAK or generated independently using a cryptographically secure RNG. The SAK must be encrypted so that only peers in possession of the CAK can use it. To encrypt and decrypt it in such a way, a Key Encrypting Key must be derived from the CAK the same way for each peer in the CA and then must be used as an input of AES Key Wrap around that SAK before distribution. Each SAK is also identified in the clear by a 128-bit Key Identifier to identify the corresponding SAK for network management and personnel to observe and diagnose MKA operation without needing access to the secret key. The MKA key server then distributes the SAK by using an MKPDU that includes a distributed SAK parameter set. Each MKPDU that is received is verified by checking an ICV key, which is also derived from the CAK, against its ICV at the end of the packet. After the SAK is distributed and verified, each peer establishes its TX-SA as well as an RX-SA. At this point, MACsec communications are established and can be used. See Fig. 7 for a visual representation of this initial commissioning process using a pairwise CAK that is immediately rotated.

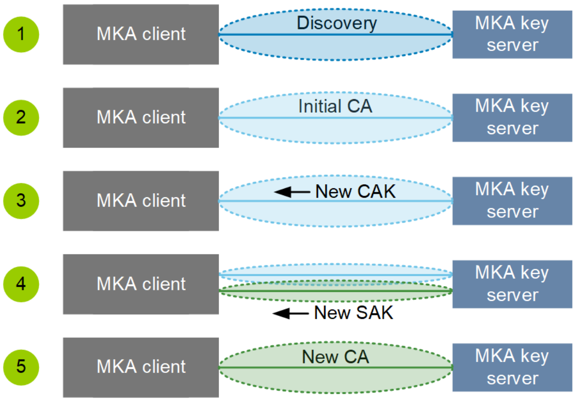


Fig. 7. Initial MKA PSK with rotation.

When the key server must rotate all participants on the CA to a new SAK, which occurs when the key server observes the lowest acceptable PN for the latest key (here, the latest key refers to the latest SAK) or when an external trigger requests it, the key server generates and then distributes new SAKs to each peer using the same MKPDU mentioned above. The new SAK is identified by a new Key Identifier value, which is used to coordinate the correct SA between KaYs. Once each participant is in possession of the new SAK, it advertises the status of the receive SAs and its transmit SA through MKPDU broadcasts. Then, when the key server sees that all peers are ready to receive, it begins transmitting over its new transmit SA. Once all KaYs see the key server transmitting over the new SA, they switch their transmit SAs to the new one as well. That describes the handshake of transition from using one SAK to the next without ever losing data in the process.

D. *MKA CAK Rotation*

Although not as typical of a use case in most information technology (IT) scenarios, it is sometimes a requirement, or a desire, to rotate any long-standing keys in an operational technology (OT) environment, according to National Institute of Standards and Technology (NIST) guidelines [5]. To ensure the secure operation of an existing network connection is not affected, there must be a way to seamlessly distribute a new CAK to each participant in the CA as well. Although there is currently limited literature on this process, the IEEE 802.1X Clause 9 MKA specification does support this operation [2]. The process is similar to that of an MKA key server distributing new SAKs, even down to the reuse of the same MKPDU packet, but this time using a distributed CAK parameter set value. Every KaY then stores this second CAK and broadcasts back that it is also in possession of that CAK along with its corresponding Connectivity Association Key Name. This, in effect, creates a new CA, independent of the one that was previously being used. Now, the key server can distribute new SAKs for the new CA rather than the previous one. Once all SAs and connections are set up, communications can begin using that SA, now under a new CA. The key server continues to keep track of all live peers as well as potential peers, and it is critical that each KaY only remove the old CA if all live peers are already broadcasting on the new CA. The key server also keeps track of the ANs of the old SAK and properly increments the AN of the newly distributed SAK to allow all MACsec communications to continue without disruption. Fig. 8 visualizes this bumpless key rollover process. This process of rotating SAKs, and CAKs if desired, keeps secure communications across an Ethernet network for the lifetime of said network, after the initial commissioning process.

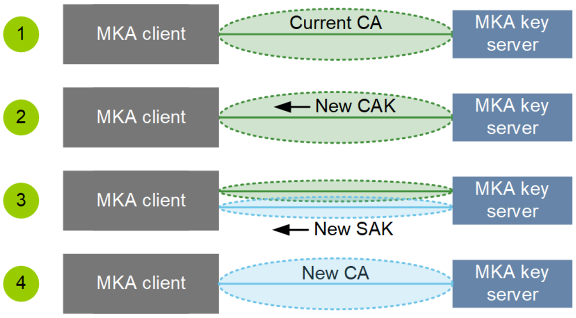


Fig. 8. MKA CAK rotation process.

IV. OSI Layer 3 (L3) Security Via IPsec

Next, we discuss the technicalities of the two cryptographic protocols that are commonly used (or considered for use) in energy system networks.

A. *An Introduction to IPsec*

IPsec is a popular security standard for securing Internet Protocol (IP) packets traversing through IP routable networks. It works in tandem with its key management protocol to establish a secure session for a limited lifetime and renews the secure sessions on a periodic basis.

IPsec offers a suite of security protocols that provide granularity and configurability for both the data plane and the key management session. Configurability can include selecting the type of data to secure, selecting the type of security function to support (e.g., confidentiality or data origin authentication), choosing authentication methods, and choosing cryptographic algorithms for key management and the data plane session.

To secure the data plane, IPsec offers two major protocol options: Encapsulating Security Payload (ESP) and Authentication Header (AH). In this paper, the scope is limited to the more common ESP protocol, which offers both authentication and encryption, unlike AH, which offers only authentication. The ESP protocol secures packets through directional secure sessions (SAs). Each SA is identifiable by a unique Security Parameter Index (SPI). Therefore, for two-way communication, each peer has a transmitting SA and a receive SA, each with a unique SPI value labeling the SA (roughly equivalent to MACsec's SAI).

An important aspect of IPsec is the portion of the topology that can be protected. IPsec is very flexible in this regard such that both end hosts and network gateways can participate in an IPsec connection. For example, a network gateway can participate and communicate with another subnet to another gateway or communicate directly with an end host. Similarly, an end host can communicate with another end host or with a gateway. For this paper, we limit the scope to a connection between two end hosts, such as between a control or protective device to a remote terminal unit (RTU) or SCADA master end host, possibly in a substation or control center.

IPsec also offers flexibility in the portion of the IP packet that is protected. When an IPsec connection is configured in tunnel mode, the contents of the entire original IP packet (including IP headers) are encapsulated and sent to a receiving gateway (or end host). The receiving gateway of the IPsec tunnel strips the IPsec header and routes the packet as per the original inner IP header. In ESP mode, the original encapsulated IP headers also appear obfuscated, as opposed to transport mode, where only the payload of the original IP packet is encapsulated and secured. For this paper, we limit the scope to a tunnel mode configuration. In this mode, a new IP header is added, with source and destination IP addresses of the IPsec clients taking part in the tunnel on a nonsecure channel. Fig. 9 gives an illustration of IPsec ESP mode.

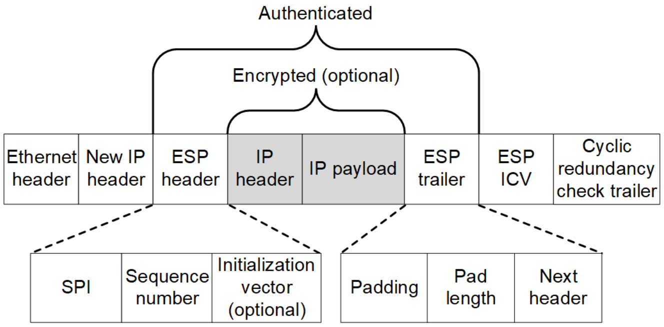


Fig. 9. IP packet encapsulated with ESP tunnel mode.

B. *IPsec Cryptographic Considerations*

Cryptographic algorithms used for the secure session and for the management protocols are also important configurable parameters in IPsec. Both IPsec clients mutually arrive at the same cryptographic algorithms supported on both ends using IPsec's key management protocol.

For an ESP SA, NIST has suggestions for the latest best practices for cryptographic algorithms [6].

Key management, authentication, and negotiation of the ciphers is commonly conducted with the combination of Internet Security Association and Key Management Protocol and Internet Key Exchange (IKE) version 2 protocol. IKEv2 specifies two distinct roles for participants involved in establishing an IPsec connection: an initiator, which commences a request to create an initial IKE secure session, and a responder, which waits for the initial establishment of the connection. In our example, the protection and control (P&C) device assumes the role of responder and the head-end station takes the role of initiator, like in Fig. 10.

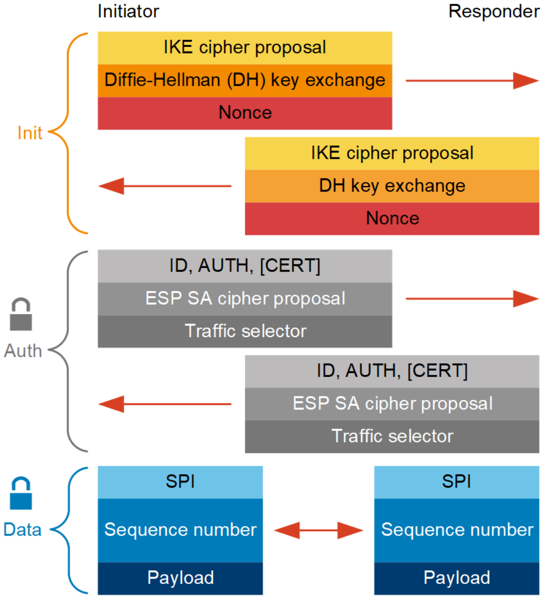


Fig. 10. Init, Auth, and Data stages of IPsec.

Internet Security Association and Key Management Protocol and IKEv2 key management is composed of two essential stages [7]:

1. The Init stage, where the peers negotiate encryption and integrity ciphers, exchange nonces, and exchange public keys to derive a shared secret using DH. This is done to establish an encrypted session for the next stage (Auth) of the key management.
2. The Auth stage, where the peers communicate on an encrypted channel using the shared secret derived in the Init stage, authenticating each other through various methods, such as a PSK or X.509 certificates. The peers also establish parameters for creating an IPsec SA, including negotiating ciphers for the IPsec session and deriving keys for the SA.

IKEv2 cryptographic parameters that are also mutually negotiated among IPsec clients and ciphers supported by both the clients are used. NIST also recommends best practices for IKEv2 ciphers [6].

V. OSI Layer 4 (L4) Transmission Control Protocol (TCP) Security Via TLS 1.3

A. *An Introduction to TLS*

TLS is a ubiquitous protocol for connections between internet-based services and consumer electronics. TLS is particularly popular for protecting HyperText Transfer Protocol (HTTP) web browser data, and the conjunction of the two is called HyperText Transfer Protocol Secure (HTTPS). TLS is used with TCP-based protocols because TLS relies on an underlying stream-based transport layer to provide reliable transmission and packet reassembly, as shown in Fig. 11. There are other connectionless implementations of TLS, such as Datagram TLS, but discussion of the separate Datagram TLS protocol is outside the scope of this paper. This paper also limits the scope of TLS to its most recent iteration, version 1.3 [8].

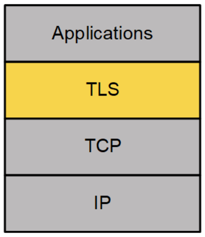


Fig. 11. TLS protocol relies on TCP.

A TCP connection is established between a client and a server. In the context of a protection control device, the client could be a SCADA master, and the server could be a SCADA outstation on the RTU. TLS protects application-layer communications through symmetric key encryption of the data between clients and servers.

TLS functionality is supported by a protocol suite rather than one single protocol. As seen in Fig. 12, TLS is made up of the following protocols:

- Record: All the contents of TLS key exchange, alert messages, and application payload are encapsulated in this format.
- Handshake: All messages exchanged regarding the initial setup of the TLS connection are labeled as handshake messages.
- Alert: Any erroneous issues in a TLS connection are labeled as alert messages.
- Application data: Encrypted application data exchanged during a TLS session are labeled as application data.

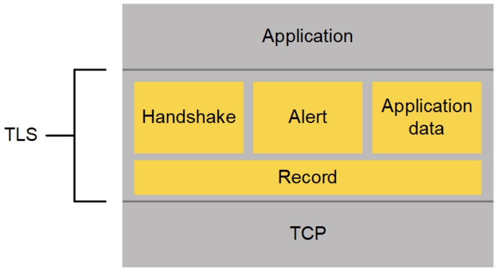


Fig. 12. TLS protocol suite.

B. TLS Cryptographic Considerations

An instance of a TLS connection can be referred to as a TLS session. To secure a TLS session, authentication of at least the server is performed. The three methods for authentication are:

- Public-key infrastructure (PKI).
- An external PSK.
- A session resumption key that employs session tickets issued by the server.

For control plane authentication of the end hosts on the internet, PKI is relied on for most of its current instantiations. PKI provides a chain of trust from centralized authorities, called Certificate Authorities. During the authentication process, the TLS server presents an X.509 certificate signed by a Certificate Authority when proving its authenticity, along with a signature of itself. The TLS client and server may use PSKs in lieu of PKI as a less common form of authentication.

To ease reauthentication of the existing client with a server, a server has the option of generating a key for use by a TLS client for newer TLS sessions. This is referenced as a session ticket. When used by the client, this generated key serves as a form of PSK authentication (hereafter referred to as a resumption PSK).

To secure the control and data planes, TLS employs Authenticated Encryption with Associated Data (AEAD) ciphers with different symmetric keys for both the server and the client control and data planes. However, this symmetric key originates from a common master shared secret. This common master secret is arrived at based on the authentication method used and whether another shared secret is generated to introduce forward secrecy (FS), using an Elliptical Curve Diffie-Hellman Exchange (ECDHE) during the establishment of the session. The possible combinations for the common shared secret based on the authentication method are listed in Table I.

TABLE I
INPUTS FOR TLS 1.3 MASTER SHARED SECRET

Authentication Method Used	**Possible Inputs for Common Master Shared Secret Derivation**
PKI	The shared secret derived from ECDHE
External PSK	No ECDHE: only the PSK secret ECDHE performed: the PSK secret and a shared secret derived from ECDHE
Session resumption key (session ticket; resumption PSK)	No ECDHE: only the session resumption key ECDHE performed: the session resumption key and a shared secret derived from ECDHE

If the TLS implementation does not use ECDHE, then there is no guarantee of FS for the link [9].

From a packet exchange standpoint, TLS sessions are normally established in three exchanges involved in the handshake. Fig. 13 illustrates that data can begin to exchange in 1-Round Trip Time (1-RTT) during the handshake process. In the first packet, the TLS client offers a short list of AEAD ciphers and pseudo-random functions (PRFs) it supports for negotiation. If the authentication type is not based on PSKs, public information for ECDHE is exchanged to arrive at a shared secret.

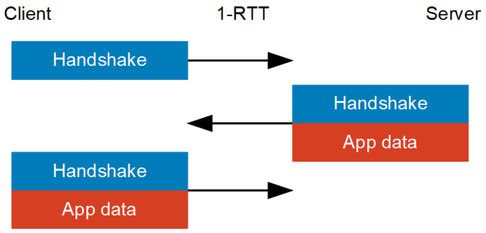


Fig. 13. Packet exchange during 1-RTT TLS initialization.

The server also responds with public information for the ECDHE, AEAD ciphers, and PRFs. If the server is being authenticated with an X.509 certificate, then the certificate (signed by a trusted Certificate Authority) along with a digital signature and message authentication code are sent encrypted. If authentication from a TLS client is needed, the TLS server also adds a request to the client for a similar certificate and a signature. Encrypted data can also be sent along with the handshake data from the server because keys have been derived from either the ECDHE or the PSK.

If the certificates from the client were requested, the certificate and a digital signature are sent encrypted in the final exchange of the handshake. Also encrypted is a message authentication code and application data from the client. At this point, the TLS connection is successfully created.

If the clients and the servers use a PSK or session resumption key for authentication, it is also possible to use a variant of the handshake exchange, where user data can be sent out immediately during the initial handshake packet exchanges. Fig. 14 illustrates these data being exchanged instantaneously, known as a 0-Round Trip Time (0-RTT) connection. However, the 0-RTT connection has weaker security properties in terms of packet non-replay, and the initial data exchanged along with the handshake do not have FS guarantees if ECDHE is not used.

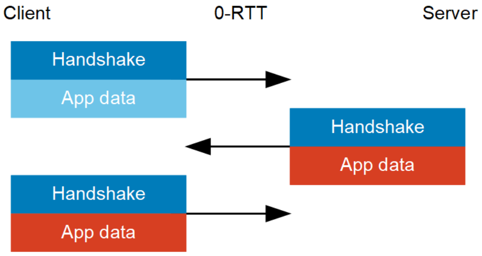


Fig. 14. Packet exchange during 0-RTT TLS initialization.

VI. Cryptographic Protocol Comparison

In the following subsections, we draw several comparisons between IPsec, TLS, MACsec, and their key management protocols (IKEv2, TLS handshake, and MKA) for suitability to electric P&C devices. We focus on the security attributes provided by the control plane and data plane, as well as some additional attributes.

A. *Comparison of Cryptographic Protocol Control Plane*

The control plane of a security protocol involves functions such as facilitating authentication of peers in the connection, as well as key generation and distribution. There are also differences with how strictly the control plane of a cryptographic protocol couples with the data plane. MKA acts as a separable, extrinsic control plane for MACsec while the control plane mechanisms for IPsec (IKEv2) and TLS (handshake protocol) are intrinsic and inseparable from those protocols. In the following subsections, VI.A.1–VI.A.7, we contrast the control plane and its suitability to applications in energy system networks.

1) *Control Plane Authentication*

The control plane manages how the cryptographic protocol verifies the identity of remote hosts with which it wishes to securely communicate. This first process of authenticating peers in a connection and establishing a secure control plane is essential for the holistic security of the protocol. MKA can authenticate the control plane either via normal IEEE 802.1X methods outside the scope of this paper (such as Extensible Authentication Protocol-TLS) or via a PSK. IPsec can use many methods to mutually authenticate each endpoint in the control plane, including a variety of Extensible Authentication Protocol methods, X.509 asymmetric key pairs, and a PSK. TLS 1.3 can use X.509 certificates with its associated PKI controls, PSKs provided by external means, or resumption PSKs generated by the TLS server and distributed to the TLS client after a previously established handshake via session tickets [8].

P&C devices generally prefer PSKs when it comes to fast and simple control plane authentication, in part due to the ability to establish secure communications by requiring a one-time physical presence, known in the industry as a trust-on-first-use approach, without requiring complex supporting security infrastructure [9].

2) *Control Plane Confidentiality and Integrity*

MKA, IPsec, and TLS all supply integrity controls for the control plane. The control planes need confidentiality for certain functions, such as the secure transport of keys with MKA. MKA is an example of such a protocol that integrates minimal confidentiality, except when distributing new SAKs for MACsec or CAKs for MKA, which it encrypts using AES Key Wrap. IPsec using IKEv2 provides additional confidentiality controls at the control plane for privately negotiating traffic selectors without exposing IP addresses and subnets and for providing secrecy for certificates during authentication. TLS goes even further, where it provides an encrypted Server Name Indication field to prevent those snooping the wire from seeing what eventual endpoint the TLS client is desiring to communicate with. Confidentiality of the TLS control plane has led to a demand for TLS decryption device usage in critical infrastructure to prevent threat actors from utilizing that protocol as a method for remote command-and-control and data exfiltration [10].

3) *Control Plane Key Management*

Some cryptographic protocol control planes have intrinsic methods to manage their own keys after a manual initialization period, whereas others require extrinsic key management. MACsec without MKA requires extrinsic key management, while MACsec with MKA can manage its own keys intrinsically after initial configuration by an operator, since the MKA protocol contains methods to generate and distribute CAKs for MKA, even to the point where no initial commissioning secrets remain. IPsec relies on extrinsic key management for its control plane keys after initial configuration by an operator, as does TLS for the most common scenarios; however, TLS does have methods for some key management when using certain PSK modes.

As detailed previously, MKA has special parameter set values for the distribution of CAKs over an existing CA. The distribution of the new CAK must be controlled by the specific implementation, but that process may execute a regularly scheduled CAK distribution event to refresh the cryptographic context of the CA to aid system operators in following cybersecurity standards for key renewals [5].

For TLS, the use of external PSKs and X.509 certificates for control plane authentication requires the extrinsic management of those keys after initial operator configuration. However, TLS 1.3 can use trust-on-first-use methods for initial operator configuration (such as initial configuration with X.509 certificates or an external PSK) and then have the TLS server manage the control planes by issuing resumption PSKs to the TLS client via session tickets [8]. This method can effectively establish permanent trust (if configured by the implementation) since a full handshake using an external PSK or X.509 certificate is no longer required, so long as new resumption PSKs (via session tickets) are issued regularly by the TLS server to avoid the seven-day maximum lifetime of session tickets [8]. As a downside to this approach, resumption PSKs used with 0-RTT handshakes are vulnerable to replay attacks [11]. Further, if the implementation does not use the resumption PSK with ECDHE mode when invoking the TLS handshake, there is no guarantee of FS for the data plane [8]. In some circumstances (real-time, high-volume communications), it is best for P&C devices to use resumption PSKs without ECDHE mode due to computational considerations.

4) Generation and Establishment of Data Plane Keys

A primary duty of the control plane is to generate and establish the cryptographic keys used for the security of the data plane. MKA automatically generates and distributes keys for MACsec based on PN exhaustion, while IPsec and TLS use DH for offline key generation (except for a non-DH mode for resumption PSKs used in TLS).

The MKA key server uses the CA as the secure channel for communication and establishment of keys for MACsec. To do this, the MKA key server encrypts the SAK when distributing it to CA participants, which those participants use for establishing MACsec SAs with all other MKA peers for secure MACsec communications. The SAK is a group shared key and thus is not unique to each CA participant. The MKA standard specifies two methods for the generation of SAKs: generation entirely from the key server's RNG or from a mixture of RNG and metadata taken from other parameters available on the CA [2].

The IPsec control plane uses an intermediary key to generate symmetric data plane keys via a PRF. IPsec generates the intermediary key itself from a PRF using contributions from several parameters exchanged during initialization of a new IKEv2 connection (known as an IKE_SA_INIT message exchanged during the Init stage described previously). Those contributions include nonces exchanged in the clear over the connection, a common shared secret derived via DH, and other inputs, such as the SPI of the ESP connection, also exchanged in the clear. The control plane then combines the intermediary key with further nonces to generate the keying material for the first IPsec ESP SAs.

TLS generates client and server data plane keys (application keys) from a common master secret, itself generated either via ECDHE (logically after the control plane authentication via a PSK or X.509 certificates) or directly from a PSK.

5) Data Plane Session Management

MKA performs SAK renewals (affecting both transmit and receive SAs) under three conditions:

- The PN for either transmit or receive associations (SAs) reaches a threshold limit (pending PN exhaustion).
- The specific MKA key server implementation forces a refresh of the SAK either directly or indirectly via a CAK distribution event.
- The MKA key server detects that a new participant has joined the CA.

IKEv2 is responsible for the renewal of IPsec transmit and receive sessions (SAs). There are two major conditions that cause these data plane key renewals [12]:

1. Periodic inline rekeying happens when the expiration of timers or counters associates with the lifetime parameters configured by either peer. These configurable parameters can be either a lifetime in seconds, a number of bytes, or a certain number of packets, depending on the IPsec client used.
2. Reauthentication of either peer participating in the IPsec connection through the IKEv2 management protocol. Reauthentication verifies that peers still retain access to their authentication credentials. On reauthentication, the IPsec SAs also renew. Based on the IPsec client implementation and configuration, reauthentication events are conducted on a set lifetime.

The TLS control plan can renew application keys during the session itself as part of a key update procedure, and either end of the link may initiate this process. To maintain the security of the data plane session, the TLSv1.3 standard dictates a limit on the amount of data that can be exchanged on a data plane key, nearing which the end host must renew the key. When using the AES-GCM cipher, this limit comes to 24 million full-size record protocol units [8]. Therefore, the key renewals can be performed anytime by either client or server. However, the standard mandates key renewal for the client or server before they reach the data transfer limit.

6) Transition Between Data Plane Sessions

The transition during key renewals is a distinct process between the security protocols. System operators desire seamless data plane session transitions because P&C devices use GOOSE and SV, which rely on a link with minimal interruption and no packet loss for communications-based protection applications.

MKA handles transitions between data plane sessions for MACsec for both control plane and data plane key renewals. In the event of a key change of an SA, the MKA key server ensures that a transitory state exists where communications on the old key are active on all peers while the new key is installed and begins to be used. This ensures bumpless key transfer without any loss of packets, commonly known as a make-before-break connection. Due to their ability to seamlessly transition between data plane sessions, MACsec and MKA are practical solutions to protect high-availability communications with

3-millisecond transfer time requirements across an Ethernet network, as described in IEC 61850-90-4 [13].

For IPsec, the renewal of SAs can be configured for make-before-break such that a transitory state exists where both the peers in the connection (initiator and responder in IKEv2 terminology) can ensure access to old SAs while establishing new receive and transmit SAs. Older behavior in some implementations of IPsec is break-before-make, in that it causes the endpoints to immediately move to the new SA, which could cause data traffic loss when the responder sends the traffic on the old SA during the transitory state and the initiator is awaiting a confirmation on the new SA creation [12].

For TLS, the end host renewing its transmission keys sends a key update message and switches the transmission of the data plane over to the new key. Since TCP ensures in-order reliable packet delivery, there is no data loss when changing the keys since TCP ensures the stream is delivered in order, although there may be a small latency added during the key transition process. However, on a packet loss, this may delay the timely processing of subsequent data until the previous data have been successfully retransmitted. As an additional consideration, it may be possible for TLS to deliver data continuously even during a control plane rekey (TLS handshake). So long as the handshake implements a 0-RTT method, it is possible that hosts will not drop packets. A specific TLS implementation may also use multiple session tickets and use multiple resumption PSKs simultaneously across multiple TCP streams to ensure data continuity. However, as previously mentioned, the use of 0-RTT methods comes with additional security risks (specifically to confidentiality), and this may limit the application of TLS to only SCADA-based protocols and not those with critical timing requirements (e.g., IEEE Std C37.118 synchrophasors [14]).

7) Some Additional Important Control Plane Differences Noted

An important distinction between MKA and both IPsec and TLS is that the MKA key server handles creating and distributing the session keys for MACsec SA, in contrast to the IPsec and TLS DH method where both peers in a connection arrive at a shared symmetric session key without the control plane distributing that key over the control plane connection. Thus, MACsec and MKA keys that are distributed over the CA are not capable of FS since they rely on the secrecy of the CAK for their confidentiality. In contrast, IPsec and TLS provide FS because the encrypted symmetric key used for the data plane is not sent over the control plane channel, and thus, IPsec and TLS are not reliant on the privacy of the control plane cryptographic keys. For systems that do not have a strong requirement for confidentiality of already-communicated data, FS is not a benefit.

There are benefits to not supporting FS. DH functions can be computationally intensive compared with encryption or message authentication code operations, so support for MACsec and MKA may not require as much processing power as support for TLS or IPsec. Further, manufacturers may designate MACsec-supporting P&C devices as clients-only so that they never become a key server. This ensures that all key renewal processes are offloaded to the key server device (a gateway or switch that can support more complex cryptographic processors).

Another important distinction between MKA and both IPsec and TLS is which devices need good RNG capabilities. Because the MKA key server generates and distributes keys, the clients (which can be P&C devices) do not need good entropy sources or RNG functions. IPsec and TLS require the generation of cryptographic random values by both endpoints and thus require cryptographically secure pseudo RNGs [8]. For P&C devices, this imposes additional hardware and supply chain requirements when minimal device updates to a P&C device can prevent early technology-based obsolescence.

TLS 1.3 supports multiple cipher suites, signature generation functions, and hashing algorithms, and IPsec supports vastly more (for compatibility purposes). MACsec and MKA both support only a limited set of cipher suites (AES-GCM for MACsec and AES-CMAC for MKA), which further reduces implementation burden and implementation maintenance.

Finally, “vanilla” MACsec (MACsec without MKA) is the only cryptographic protocol that does not require sessions establishment at the control plane level—a process that delays the transmission and reception of secure application data by a small amount of time. The 0-RTT mode of TLS also supports a similar function, but this comes with security downsides (such as a risk of replay attacks).

B. Comparison of Cryptographic Protocol Data Planes

In this section, we conduct a comparison of various security protocols with respect to their data planes.

In cryptographic protocols, to protect the data, the following security services can be analyzed:

- Integrity: Does the cryptographic protocol provide integrity controls for the data plane? This is a mandatory requirement for security for energy system protocols.
- Confidentiality: Does the cryptographic protocol provide confidentiality controls for the data? For certain applications of P&C devices, monitoring via an intrusion detection system may require packet contents to be in the clear.
- Ethernet protocol coverage: What kinds of Ethernet protocols does the cryptographic protocol protect?
- Frame replay, reorder protection: Does the cryptographic protocol prevent replay attacks against the data stream?
- Message holdback protection: Does the cryptographic protocol provide the ability to detect and discard stale data (data older than a certain period of time)?
- FS: Does the cryptographic protocol guarantee that the exposure of all long-term keys will not compromise the confidentiality of prior encrypted communications?

MACsec protects the payload of Ethernet frames by requiring mandatory integrity but offering optional encryption. MACsec secures OSI L2+ Ethernet protocols and above on an endpoint-to-endpoint direct connection. This offers the advantage of using one encryption and authentication solution to protect L2 protocols and above on the link. This makes MACsec suitable for use with IEC 61850 GOOSE and SV without requiring mapping to other layers of the OSI model. MACsec is compatible with existing Ethernet networks (with the ability to expose VLAN headers on existing Ethernet switches) and compatible with Ethernet redundancy protocols, such as IEC 62439-3 Parallel Redundancy Protocol [15]. MACsec also protects ancillary but essential protocols to functioning of the IP stack, such as ARP. This reduces the attack surface for attack vectors, such as ARP poisoning.

IPsec protects OSI L3+ IP payloads. However, like MACsec, it offers configurability-optional encryption. TLS, however, secures OSI L4+ payloads that use TCP and requires confidentiality to be mandatory. TLS 1.3 always enforces encryption through use of AEAD ciphers. However, for use with intrusion detection systems, TLS offers no standardized version to support null ciphers and provide optional encryption in the current TLS standard. However, there are nonstandard implementations and a push in the industry to support null cipher suites for applications requiring the Internet of Things or the Industrial Internet of Things [16]. MACsec supports this requirement to be compatible with intrusion detection systems with optional confidentiality.

While all the cryptographic protocols provide replay protection, IPsec requires a 32-packet window to gracefully handle packet reordering events on public networks and thus is susceptible to reordering events. MACsec has an optional packet reordering window, while TLS lacks a packet reordering window entirely. (TLS relies on TCP to provide reordering.) MACsec also adds the unique but optional ability to prevent stale data attacks with message holdback protection when used in conjunction with MKA.

In the context of internet communications, FS is important to avoid exposing past recorded information and compromising long-term keys. IPsec and TLS both use DH to introduce random data plane keys on new sessions. In contrast, MACsec and MKA do not provide FS.

C. *Additional Considerations*

Some attributes of the various cryptographic protocols that we discuss here do not admit to strictly classifying by control plane or data plane functions.

1) *Trusted Platform Module (TPM) Support*

A secure enclave, such as a TPM—typically implemented as a separate, hardened physical component—provides secure storage for cryptographic materials on a device. TPMs provide maximum security for asymmetric public-private key implementations due to their ability to fully hide the private key, which negates the need to expose that key by routing it to other areas of the system. TPMs only partially support symmetric key pairs, since those keys do need to be sent to different areas of a system for usage in the cryptographic system and can only fully secure symmetric keys when the system is powered off. TPMs therefore provide better protection for IPsec and TLS under normal circumstances (when those protocols use asymmetric control plane authentication methods) rather than MACsec or MKA.

2) *Quantum Cryptanalysis Resistance*

If researchers can expand the capabilities of quantum computers, quantum cryptanalysis will be able to defeat most modern asymmetric cryptographic methods [17]. The same quantum computers would have only quadratic speedup against symmetric cryptographic ciphers, which reduces 128-bit symmetric keys to an effective 64 bits, and 256-bit keys to an effective 128 bits. Therefore, MACsec and MKA (in their 256-bit variants) are naturally resistant to quantum attacks, while most (if not all) asymmetric ciphers used by IPsec and TLS will be vulnerable. Currently there is no timeline for the introduction of quantum-based cryptanalysis. However, given the difficulty of predicting the introduction of new technologies, quantum computing should still be a consideration for devices that will have lifespans of decades.

3) *Routed Network Support*

Routers strip SMAC information from Ethernet frames and sometimes change IP and TCP headers. MACsec and MKA cannot communicate natively over a routed network since they check those headers' data as part of their integrity checking functions. IPsec can communicate over public routed networks, and TLS can even communicate through load balancers and Carrier-Grade Network Address Translation devices that predominant public cellular networks use. When P&C devices are needed to communicate with other devices located on routable networks with IP-based protocols, infrastructure operators can use IPsec and TLS in tandem with MACsec on nonlocal networks in the topology. For example, system engineers can implement IPsec for the security of communications between network gateways as a tunneling solution, and MACsec can be used to secure the local network path from the gateway to a P&C device. Gateway devices can handle the frequent firmware updates required to support IPsec and TLS protocols, and this dual-cryptographic mode (either IPsec or TLS with MACsec) alleviates the need for similar firmware updates on P&C devices.

4) *Complexity*

A critical part of the success of the implementation of any technology is whether it is easy to operationally maintain and use by its intended user base. MACsec and MKA (in PSK mode) are relatively simple protocols to develop, program, and use [18]. (MACsec's official Linux kernel driver is 3,110 lines of code without headers and comments.) IPsec and TLS both require complex implementations to support a variety of general-purpose scenarios and scenarios befitting a substantial number of internet-based applications and devices, and both are difficult to program and use correctly by automation professionals.

5) *In-Memory Attack Surface*

A concern with hop-by-hop cryptographic protocols (such as MACsec and MKA) is the need to perform encryption and decryption operations by several devices on the same data flow. If a threat actor were to compromise a device performing the encryption and decryption functions, then they would have the ability to view and manipulate the unprotected data flow during its traversal across the device, or the attacker could compromise the cryptographic keys to view or manipulate the protected data flow on the wire. This latter scenario also affects complex asymmetric protocols that rely largely on trust relationships using PKI and X.509 certificates, including possible embedded hardware components, such as TPMs. The impact of the targeted compromise of naming infrastructure, trust infrastructure, and certificate key generation and revocation systems can be enormous for any number of devices implementing TLS. This problem is especially acute when supporting infrastructure (such as Domain Name Services) and trust infrastructure are in the corporate environment, which is more likely to be attacked by adversaries. IPsec using PSK implementations may be least susceptible to this specific scenario when implemented end-to-end without tie-ins to centralized trust infrastructure.

At the end of this paper, we include a full comparison table of security attributes for easy reference.

VII. A Static and Dynamic Framework and Architectural Recommendations

A. The Dynamic, Static Element Framework

When attempting to evaluate the use of cryptographic protocols in energy system environments, we first recommend sorting energy system elements into static and dynamic types [19]. Static elements prioritize reliability and availability by providing automated telemetry and high-speed protection functions (or other automated controls) as well as routine and automatic operation for lengthy periods of time without human interaction. Static system elements are physically protected and isolated, and all communication flows are known and designed in purposefully. Dynamic elements support business requirements for the control and supervisory functions used to manage the system. They are dynamic in that applications, communications, and configurations change more frequently to serve changing business needs. Dynamic system elements are more exposed to other hosts, unauthorized personnel, and novel threats, increasing the need for up-to-date cybersecurity support.

The usual cryptographic choices for dynamic elements include protocols that are built for a wide variety of uses and thus are subject to frequent standard changes and implementation updates. They are complex to develop and require considerable management functionality. These kinds of cryptographic protocols require substantial configuration and operational expertise and are designed to complicate application monitoring efforts through heavy reliance on confidentiality. When applied in OT environments, their complexity tends to negatively affect the overall safety, reliability, and availability of critical energy system elements due to the need for frequent firmware updates. Our framework advises against IT cryptographic security controls for data and commands exchanged with other dynamic system elements only (not for static system elements). Examples of these kinds of protocols include TLS and IPsec.

The best cryptographic choices for static elements are protocols and algorithms that are not subject to frequent changes, are simple to develop and apply, do not block modest monitoring efforts, and also do not negatively affect the overall safety, reliability, and availability of critical energy system elements by requiring frequent changes. This framework also recommends either abstaining altogether from cryptographic protocols for static elements, or, if justified by threat analysis, using static-oriented cryptographic protocols. An example of a cryptographic security control that is suitable for static environments due to its simplicity and longevity is MACsec and MKA with a simple form of authentication.

As Fig. 15 shows, any digital signals that must flow between dynamic and static elements are handled by a third element type—a device type called a mediator. A mediator incorporates both IT and OT security controls and communicates with both dynamic and static infrastructure elements. A mediator already exists in most energy systems, since advanced RTUs, jump boxes, protocol converters, proxies, gateways, and embedded terminal servers often take on the role of a mediator. The mediator effectively acts as a cryptographic protocol break, if not as an application-layer inspection chokepoint or "protocol break" (which cybersecurity engineers often use for North American Electric Reliability Corporation Critical Infrastructure Protection [NERC CIP] applications [20]).

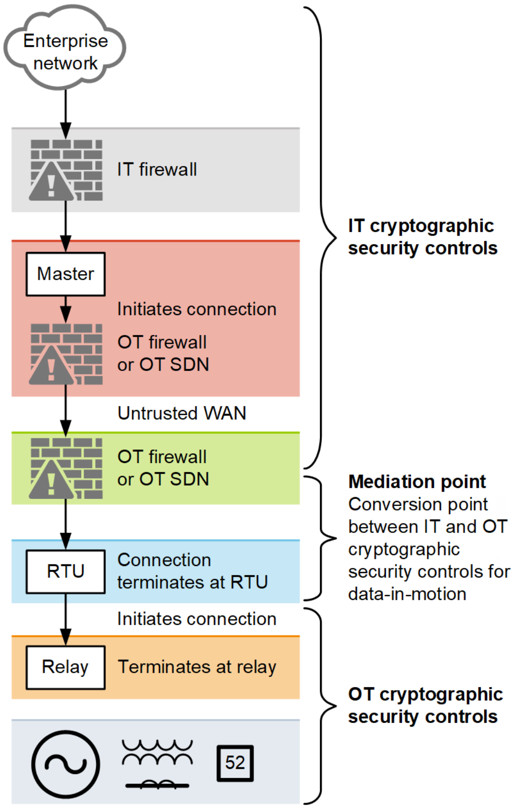

Enterprise network
IT firewall
Master
Initiates connection
OT firewall or OT SDN
Untrusted WAN
OT firewall or OT SDN
RTU
Connection terminates at RTU
Initiates connection
Relay
Terminates at relay
52
IT cryptographic security controls
Mediation point
Conversion point between IT and OT cryptographic security controls for data-in-motion
OT cryptographic security controls

Fig. 15. Example energy system data flow with cryptographic protocol overlay.

This paper next examines the use of MACsec with MKA, TLS, and IPsec in three different architectures: distribution cabinet, microgrid, and substation. Keep in mind that as background to the architectural discussion, TLS and IPsec are end-to-end protocols that are suitable for routed infrastructure (such as public internet or cellular networks). On the other hand, MACsec and MKA are LAN-only technologies that cannot natively flow over routed networks. MACsec is either hop-by-hop on a LAN or end-to-end between LAN devices if middlebox Ethernet elements (switches) do not implement MACsec.

B. *Distribution Cabinet Architectures*

A simple distribution cabinet network consists of a recloser controller and a gateway device (such as a cellular modem) that connects back either to a SCADA master in a distribution substation or a distribution front-end processor in a control center, as shown in Fig. 16. Typical traffic flows between head-end and remote-end include engineering access (in the form of Telnet, Modbus, or HTTP) and Distributed Network Protocol (DNP3) for SCADA. In this network, the connections are typically as follows:

- C1 is a flat L2 Ethernet network between the master and the head-end gateway.
- C2 is a routed L3 cellular network. In a minority of cases, this connection can also be an L2 fiber or wireless connection.
- C3 is an L2, point-to-point Ethernet connection.

A growing number of architectures include peer-to-peer fiber or wireless connections between recloser controllers.

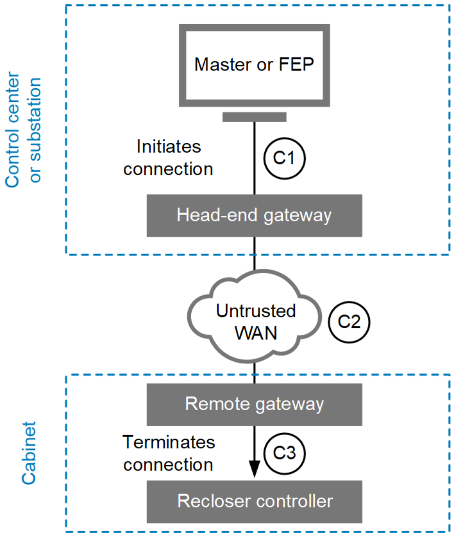


Fig. 16. Distribution cabinet architecture.

A typical application of cryptography on this network is to use IPsec to secure C2 via a gateway-to-gateway solution or to secure both C2 and C3 using an embedded IPsec endpoint in the recloser controller itself. Other common solutions involve the securing of individual protocols using TLS as a protocol wrapper or through secure application extensions, such as DNP3 Secure Authentication. Based on the framework from the previous subsection, we argue that both these solutions are nonideal for static, embedded devices, such as recloser controllers, given that both IPsec and TLS are designed for the security of dynamic system elements and are not best suited for the longevity and reliability of the recloser controller.

Engineers considering the use of MACsec on standard architectures can best use the far-end gateway as a cryptographic mediator and implement MACsec on C3 between the remote gateway and the recloser controller. This implementation best serves the reliability and security of the so-called last foot cable by both providing integrity and authenticity controls while reducing firmware churn and settings complexity associated with TLS and IPsec. Further, since MACsec can protect all L2 protocols, it is an ideal candidate for the security of IEC 61850 GOOSE connections in cases where recloser cabinets are communicating peer-to-peer or the C2 connection is an L2 network. This recommendation has the added benefit of eliminating in-cabinet gateway devices and reducing the overall maintenance burden.

C. *Microgrid Architectures*

Microgrid architectures vary substantially in size and scope. Given a scenario with distributed energy resources (DERs) that includes renewable generation assets, such as solar panels, onsite DER controllers, and a centralized DER management system (DERMS), a typical microgrid architecture is represented in Fig. 17.

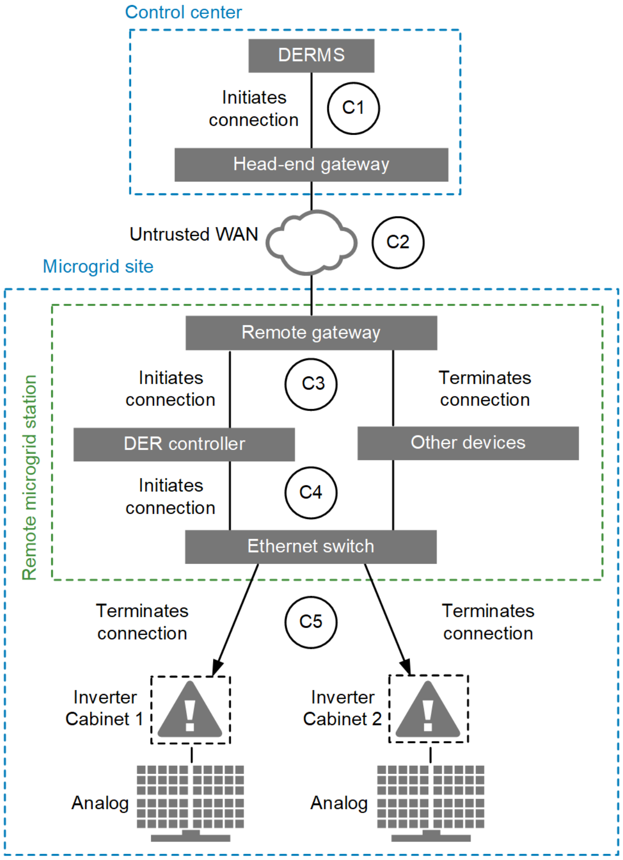


Fig. 17. Microgrid architecture.

In this microgrid representation, the DERMS lies in a centralized location (such as a control center) with one or more remote microgrids under its supervision. The typical traffic flows include single or redundant DNP3 SCADA data streams to the DER controller on site, which itself communicates DNP3, Modbus, or EtherNet/IP to individual, intelligent DER endpoints (such as inverters). HTTPS or Modbus is typical for engineering access. Network connection points are as follows:

- C1 is a flat L2 Ethernet network between the master and the head-end gateway.
- C2 is a routed L3 public network, with access provided by various well-known telecommunication companies.
- C3 and C4 are L2 Ethernet connections through an onsite industrial switch.
- C5 is L2 fiber or wireless Ethernet connections over a midsized geographical area.

Like in the distribution scenario, IPsec is used to secure C2 via a gateway-to-gateway solution. Some scenarios involve the use of either DNP3 Secure Authentication or TLS-wrapped DNP3 to provide end-to-end security between the DERMS and the DER controller and subsequently to secure C1 and C3. However, this is not the majority scenario—generally, C1 and C3 are plaintext and reliant on the protection of the physical structures in which they reside. Since DER controllers generally count as dynamic devices or mediator devices in our framework, we acknowledge the appropriate use of IPsec, TLS, or MACsec to secure C1, C2, and C3 links.

Connections C4 and C5 at the microgrid site are generally plaintext, especially if communication links use hardwired communication media. (Modern Ethernet radios generally offer link encryption.) System owners may choose to rely on the security provided by the physical control house structure for the security of C4. Because the less-protected C5 last-mile connections are L2, they are ideal candidates for MACsec as it would allow individual DERs to maintain reliability in the long term while providing authentication and integrity to data flows that are currently mostly plaintext.

D. *Substation Architectures*

A modern substation network typically features a substantial mix of older and newer technologies, media converters, and different protocols used for SCADA, teleprotection, engineering access, and instrumentation traffic, such as time synchronization. The NERC CIP cybersecurity standards have informed cybersecurity controls for the transmission and generation assets that compose the North American bulk electric system [21]. NERC CIP requirements mandate the use of encryption and multifactor authentication for engineering access requests and an examination of all data flows ingressing or egressing electronic security perimeters.

SCADA data flows (which are, again, typically DNP3 protocol in North American substations) from a SCADA master or front-end processor typically egress a head-end gateway from a centralized control point over a leased telecommunication circuit or utility-owned fiber backbone. At the substation, data flows ingress a far-end gateway (typically owned by the organization's IT business unit) and route to a gateway owned by the OT business unit that establishes a CIP electronic security perimeter to eventually terminate at an RTU. (See Fig. 18.) Engineering access flows separately terminate inside the outer gateway prior to the electronic security perimeter gateway at a so-called Intermediate System acting as a jump host. From the Intermediate System, the engineering access flow regenerates and may traverse the electronic security perimeter gateway and terminate at the RTU or travel directly to an intelligent electronic device (IED) or protective relay. The connection list is as follows:

- C1 is an L2 or L3 network between the master and the head-end gateway.
- C2 is an L2 utility-owned fiber or a telecommunication Multiprotocol Label Switching or Carrier Ethernet link.
- C3 is an L2 Ethernet connection through an onsite hardened switch owned by the organizational IT business unit.
- C4 and C5 are L2 Ethernet connections through an onsite industrial switch owned by the OT business unit.

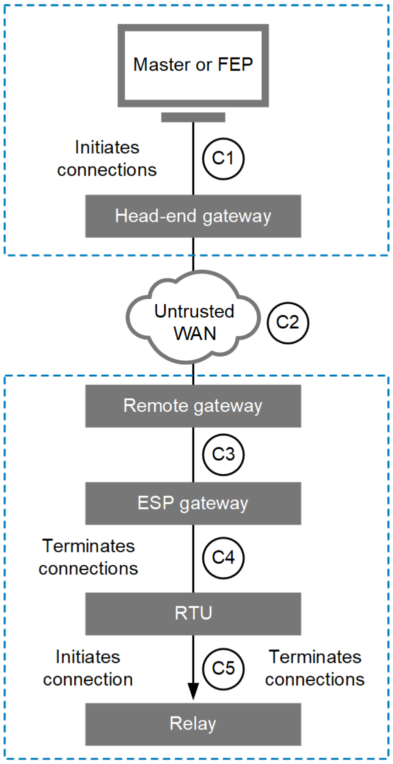


Fig. 18. Substation architecture.

Like the microgrid, utilities generally use IPsec to secure C2 and C3 connections to the remote gateway or the electronic security perimeter gateway or both via a gateway-to-gateway solution. With many telecommunication providers offering L2 connections (or utilities offering running fiber), MACsec is also a possible candidate for C2 and C3 links. Some utilities use TLS to provide end-to-end security of SCADA signals from C1 to C4, but this is currently somewhat rare, with most system owners opting for plaintext on C1 and C4 links. While encryption for interactive remote access is mandatory with NERC CIP, TLS is generally used to secure Remote Desktop or virtual interface traffic with encryption ending at the Intermediate System either before or after the remote gateway. System owners may use the RTU as a mediator device, as a termination point for TLS or IPsec, and as an initiation point for

MACsec down to subsequent devices. Thus, TLS and IPsec are acceptable at C1, C2, and C3 connections, with MACsec being an ideal candidate for C4 and C5 links. Unlike in the microgrid scenario, system owners will want to perform a threat analysis to see whether links protected by the physical control house structure will require cryptographic controls or should remain plaintext due to emphasis on reliability and availability.

VIII. Conclusion

In general, we recommend carefully considering either a cryptoless implementation or the use of MACsec for static-oriented Ethernet communication infrastructure connected via OSI L2 links, and the use of simple, well-engineered IPsec or TLS implementation (or other appropriate methods) for dynamic-oriented communication Ethernet infrastructure connected via routed OSI L3+ links. This recommendation is based upon the reliability, availability, and longevity constraints of static infrastructure and the appropriateness of the use of IPsec and TLS on public networks used by modern telecommunication devices. Given the simplicity and flexibility of MACsec for securing all protocols—including protocols, such as IEC 61850, that are more specialized for target applications—it is an ideal candidate for securing last-mile or last-foot communications infrastructure.

We conclude this paper with a summary comparison of MACsec, MACsec with MKA, IPsec, and TLS cryptographic protocols, as shown in Table II.

TABLE II
TLS, IPsec, MACsec Comparison

Security Attribute	**MACsec**	**MACsec With MKA**	**IPsec**	**TLS 1.3**
Control plane authentication	NA	802.1X or PSK	PSK, X.509, Extensible Authentication Protocol	PSK or X.509
Control plane confidentiality and integrity	NA	MKA is integrity only; minimal confidentiality for key distribution	Both	Both
Control plane key management	Extrinsic	Intrinsic with distributed CAK support	Extrinsic	Extrinsic; intrinsic in some key modes
Generation and establishment of data plane keys	Extrinsic	Generated and distributed by MKA key server	DH	DH or PSK
Data plane session management	NA	Managed solely by the key server	Managed by either end	
Bumpless key renewals	No	Yes, both MACsec and MKA	Yes, with make-before-break	Yes, with resumption PSKs
Data plane integrity	Yes			
Data plane confidentiality	Yes, optional		Yes, optional	Yes, mandatory*
Session establishment	No	Yes		Partial, with 0-RTT
Ethernet protocol coverage	OSI L2+		OSI L3+	TCP
Frame replay, reorder protection	Yes, optional		Mandatory window of 32 packets	Yes
Message holdback protection	No	Yes	No	
Forward secrecy	No		Yes	
TPM support	Partial		Full	
Quantum cryptanalysis resistance	Yes		No	
Routed network support	None		Yes	Yes
Complexity	Lower	Low	High	Higher
In-memory attack surface	Wide	Wide	Narrow	Wider

* RFC 9150 adds a null cipher suite option to TLS 1.3 [16].

Before the use of any cryptographic protocols in energy system networks, please perform a threat analysis to make sure the perceived risk of plaintext communications outweighs the possible downsides of cryptography on the reliability, availability, and economical nature of the system.

IX. References

[1] IEEE Std 802.1AE-2018, *IEEE Standard for Local and Metropolitan Area Networks-Media Access Control (MAC) Security*.

[2] IEEE Std 802.1X-2010, *IEEE Standard for Local and Metropolitan Area Networks–Port-Based Network Access Control*.

[3] L. Krattiger, "Multi-Site Data Center Networking With Secure VXLAN EVPN and CloudSec," Cisco, August 2020. Available: blogs.cisco.com/datacenter/multi-site-data-center-networking-with-secure-vxlan-evpn-and-cloudsec.

[4] Microchip Technology, "VSC MACsec PHYs Tag-in-the-Clear–Bypassed Tags and Authentication," June 2020. Available: microchipsupport.force.com/s/article/VSC-MACsec-PHYs-Tag-in-the-Clear-Bypassed-Tags-and-Authentication.

[5] E. Barker, "Recommendation for Key Management: Part 1 – General," National Institute of Standards and Technology, Gaithersburg, Maryland, May 2020. Available: csrc.nist.gov/publications/detail/sp/800-57-part-1/rev-5/final.

[6] E. Barker, Q. Dang, S. Frankel, K. Scarfone, and P. Wouters, "Guide to IPsec VPNs," National Institute of Standards and Technology, Gaithersburg, Maryland, June 2020. Available: csrc.nist.gov/publications/detail/sp/800-77/rev-1/final.

[7] A. Basu and J. Young, "IKEv2 Packet Exchange and Protocol Level Debugging," Cisco, March 2013. Available: cisco.com/c/en/us/support/docs/security-vpn/ipsec-negotiation-ike-protocols/115936-understanding-ikev2-packet-exch-debug.html.

[8] E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.3," *IETF Datatracker*, August 2018. Available: datatracker.ietf.org/doc/html/rfc8446.

[9] R. Housley, J. Hoyland, M. Sethi, and C.A. Wood, "Internet-Draft: Guidance for External PSK Usage in TLS," *IETF Datatracker,* December 2021. Available: datatracker.ietf.org/doc/html/draft-ietf-tls-external-psk-guidance#section-6.1.

[10] S. Dallas, "TLS/ SSL Decryption – One of the Main Pillars of Zero Trust Model," *Information Security Buzz*, May 2020. Available: informationsecuritybuzz.com/articles/tls-ssl-decryption-one-of-the-main-pillars-of-zero-trust-model/.

[11] WolfSSL Inc., "How to Use the 0-RTT Rope to Climb, Without Hanging Yourself!," September 2017. Available: wolfssl.com/use-0-rtt-rope-climb-without-hanging/.

[12] StrongSwan, "Expiry and Replacement of IKE and IPsec SAs." Available: wiki.strongswan.org/projects/strongswan/wiki/ExpiryRekey.

[13] M. van Rensburg, D. Dolezilek, and J. Dearien, "Case Study: Using IEC 61850 Network Engineering Guideline Test Procedures to Diagnose and Analyze Ethernet Network Installations," proceedings of the PAC World Africa Conference, Johannesburg, South Africa, November 2015.

[14] IEEE Std C37.118.1, *IEEE Standard for Synchrophasor Measurements for Power Systems,* 2011.

[15] IEC 62439-3, *Industrial Communication Networks – High Availability Automation Networks – Part 3: Parallel Redundancy Protocol (PRP) And High-Availability Seamless Redundancy (HSR)*, 2021.

[16] N. Cam-Winget and J. Visoky, "RFC 9150: TLS 1.3 Authentication and Integrity-Only Cipher Suites," *IETF*, April 2022. Available: ietf.org/rfc/rfc9150.pdf.

[17] S. Jordan and Y. Liu, "Quantum Cryptanalysis: Shor, Grover, and Beyond," *IEEE Security & Privacy*, September/October 2018, pp. 14–21. Available: computer.org/csdl/magazine/sp/2018/05/msp2018050014/17D45Xq6dCs.

[18] L. Torvalds, "MACsec.c," *GitHub, Inc*. Available: github.com/torvalds/linux/blob/master/drivers/net/macsec.c.

[19] J. Carlson, D. Gunter, C. Roberts, C. Gordon, and G. Masters, "Do IT Cryptographic Security Controls Work for Energy Systems?" May 2021. Available: selinc.com.

[20] North American Electric Reliability Corporation, "Cyber Security – Security Management Controls," January 2016. Available: nerc.com/pa/Stand/Reliability Standards/CIP-003-6.pdf.

[21] A. Jones, "NERC CIP and the Importance of Consistent Compliance," I.S. Partners. Available: ispartnersllc.com/blog/nerc-cip-standards-overview/.

X. Further Reading

J. P. Aumasson, *Serious Cryptography: A Practical Introduction to Modern Encryption*, No Starch Press, San Francisco, Figure 8-2.

H. Grasset, "Parallel Redundancy Protocol (PRP): An In-Depth Look," *Schneider Electric*, June 2015. Available: blog.se.com/electricity-companies/2015/06/12/parallel-redundancy-protocol-prp-an-in-depth-look/.

Range Commanders Council Telecommunications and Timing Group, "Overview of IRIG-B Time Code Standard," May 2016. Available: web.archive.org/web/20220328094035/https:/itsamerica.com/assets/publications/TN-102_IRIG-B.pdf.

EndRun Technologies, "Precision Time Protocol (PTP/IEEE-1588)." Available: endruntechnologies.com/pdf/PTP-1588.pdf.

M. Dworkin, "Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC," National Institute of Standards and Technology, November 2007. Available: nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38d.pdf.

XI. Biographies

Colin Gordon is a senior research engineer with over a decade of experience at Schweitzer Engineering Laboratories, Inc. (SEL), currently active in its research and development division. Colin's work experience includes the engineering and implementation of cybersecure communication networks and the research and design of embedded cryptographic security controls for risk mitigation and regulatory compliance purposes. Colin joined SEL in January 2008 and holds a bachelor's degree in computer engineering from the University of Idaho.

Prabhpreet Dua is an application engineer in automation at Schweitzer Engineering Laboratories, Inc. (SEL). He holds a master's degree in electrical power systems engineering from North Carolina State University and a bachelor's degree in electronics and communications from NIIT University, India. Preet joined SEL in August 2019 as an associate integration and automation engineer in research and development.

Alex VanDeen is a lead software engineer in the communications division at Schweitzer Engineering Laboratories, Inc. (SEL). He holds a bachelor's degree in electrical engineering from the University of Washington. Alex joined SEL in September 2016 and has been working on various networking projects since, including Media Access Control Security, software-defined networking, and traditional networking.

Justin Clark is an engineering manager with the Schweitzer Engineering Laboratories, Inc. (SEL) research and development division, specializing in the development of secure communications appliances for critical infrastructure. His work experience includes implementing, testing, and providing support for solutions using Ethernet and serial media. Justin joined SEL in November 2009 and holds a Bachelor of Science degree in computer engineering from the University of Idaho.

20220701 • TP7038-01