IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 6 January 2023, accepted 18 March 2023, date of publication 27 March 2023, date of current version 3 April 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3262310

RESEARCH ARTICLE

ES-SECS/GEM: An Efficient Security Mechanism for SECS/GEM Communications

SHAMS UL ARFEEN LAGHARI[1], **SELVAKUMAR MANICKAM**[1], **AYMAN KHALLEL AL-ANI**[2], **MAHMOOD A. AL-SHAREEDA**[1], **AND SHANKAR KARUPPAYAH**[1]

[1]National Advanced IPv6 Centre (NAv6), Universiti Sains Malaysia (USM), Gelugor, Pulau Pinang 11800, Malaysia
[2]Faculty of Computing and Informatics, Universiti Malaysia Sabah, Kota Kinabalu, Sabah 88400, Malaysia

Corresponding authors: Selvakumar Manickam (selva@usm.my) and Shankar Karuppayah (kshankar@usm.my)

This work was supported by the Fundamental Research Grant Scheme (FRGS), Ministry of Higher Education, Malaysia, under Grant FRGS 203.PNAV.6712045.

ABSTRACT Industry 4.0, as a driving force, is making massive achievements, notably in the manufacturing sector, where all key components engaged in the production processes are being digitally interconnected. However, when combined with enhanced automation and robotics, machine learning, artificial intelligence, big data, cloud computing, and the Internet of Things (IoT), this open network interconnectivity renders industrial systems more vulnerable to cyberattacks. Cyberattacks may have a variety of different impacts and goals, but they always have negative repercussions for manufacturers. These repercussions include financial losses, disruption of supply chains, loss of reputation and competitiveness, and theft of corporate secrets. Semiconductor Equipment Communication Standard/Generic Equipment Model (SECS/GEM) is a legacy Machine-to-Machine (M2M) communication protocol used profoundly in the semiconductor and other manufacturing industries. SECS/GEM is mainly designed to be utilized in a trusted, controlled, and regulated factory environment separated from external networks. Industry 4.0 has revolutionized the manufacturing industry and has brought SECS/GEM back to the limelight, as SECS/GEM is completely devoid of security features. This research proposes ES-SECS/GEM, an Efficient Security mechanism that provides authentication, integrity, and protection against cyberattacks. The proposed mechanism is compared to other security mechanisms in terms of processing time, control overhead, and resilience against cyber-attacks. The ES-SECS/GEM demonstrated promising results, suggesting that it allowed SECS/GEM devices to only connect with authorized industrial equipment, maintained message integrity, discarded forged messages, and prevented cyberattacks on SECS/GEM communications. In terms of processing time and control, ES-SECS/GEM likewise outperformed other mechanisms and incurred the lowest values for these metrics.

INDEX TERMS SECS/GEM communications, machine-to-machine (M2M), Internet of Things (IoT), security mechanism.

I. INTRODUCTION

Industry 4.0, also known as the Industrial Internet of Things (IIoT) or smart manufacturing, is the result of rapid technological development over the past few decades [1], [2]. After the advent of the technological revolution, cyber-physical structures were praised for successfully mapping the real world into the digital one. The most significant innovations of modern technology are in the fields of cybersecurity, robotics, cloud computing, 5G networks, big data analysis, machine learning, the Internet of Things (IoT), and additive manufacturing [3], [4]. Industry 4.0 aims to increase industrial productivity and modernize the production process through better connectivity, machine learning, real-time data collection, machine-to-machine interaction on inexperienced mechanisms, automation, and advanced robotics. Despite

The associate editor coordinating the review of this manuscript and approving it for publication was Parul Garg.

their vast differences in size and scope, businesses, companies, and organizations all struggle with the same issue: a lack of connectivity and real-time insights into production, product development, supply chain management, and resource utilization to make effective decisions in a timely fashion [5], [6].

Manufacturers have traditionally prioritized the safety of their OT environment while paying much less attention to the IT security of their company. Criminals are well aware of this carelessness and are aware of how simple it is to break into and hack into industrial networks. Since industries place a high value on protecting sensitive data, hackers are becoming increasingly interested in gaining access to information like product specifications, recipes, materials used, system configurations, detailed logs of equipment use, communications patterns, etc. Since the supply chain is such a large and complex process, hackers and threat actors see it as an ideal setting for trying to infect a large number of suppliers and organizations simultaneously.

The manufacturing sector is the most vulnerable and targeted industry by attackers [7], [8] due to the epidemic level of recent security breaches and cybercrimes. A recent survey by EEF found that 48% of manufacturers had experienced a cybersecurity incident at some point, with 50% of those firms suffering financial losses or market disruption as a result. Cybercrime is expected to increase to an annual cost of $10.5 trillion for businesses around the world by 2025, up from $3 trillion in 2015 [9]. Meanwhile, manufacturing has been rapidly catching up in recent years, as evidenced by the Verizon Data Breach Investigation Report 2019 which detailed 352 incidents, 87 of which were among manufacturers. The computer virus attack on Taiwan Semiconductor Manufacturing Company (TSMC) was the worst breach of information security ever recorded in Taiwan. As the manufacturing sector embraces the fourth industrial revolution, or industry 4.0, with increased automation and data exchange, it exposed the information security weaknesses at production plants [10].

Despite the fact that SECS/GEM is widely regarded as the backbone of the manufacturing industry and has been in widespread use for several years, it is utterly devoid of security measures and, as a result, cannot be used in modern industry 4.0-compliant industrial networks. Given that, the proposed security mechanism is one of the first attempts ever made to protect SECS/GEM communications from cyber-attacks. Therefore, this paper proposes an efficient security mechanism for SECS/GEM Communication called ES-SECS/GEM mechanism. The following is a list of the most significant contributions upon the completion of this research:

- The proposed ES-SECS/GEM mechanism is highly configurable and allows adaptations to the desired security level.
- The proposed ES-SECS/GEM mechanism attains message integrity and ascertains that the communication over SECS/GEM protocol only takes place amongst the authorized devices in the manufacturing industry.
- A rule-based mechanism aims to prevent cyber-attacks (i.e., DoS attacks, False Data Injection Attacks, and Replay Attacks) carried out on SECS/GEM communications in the industrial network.

The rest of this paper is organized as follows. Section II reviews existing efficient security mechanisms based on digital signature-based mechanisms and SECS/GEM security mechanisms to secure SECS/GEM communication in Industry 4.0 landspace. Section III provides assumptions, threat model, and design objectives of this paper. Section IV proposes ES-SECS/GEM mechanism to secure SECS/GEM communication in Industry 4.0 landspace. Section V introduces a security analysis of the proposed ES-SECS/GEM mechanism in terms of security attributes and security comparison. Section VI shows the experiment and result. Finally, the conclusion of this paper is presented in Section VII.

II. LITERATURE REVIEW

This section reviews some efficient security mechanisms to secure SECS/GEM communication in Industry 4.0 landspace. This paper categorizes these mechanisms into digital signature-based mechanisms and SECS/GEM security. The description of this category is as follows.

A. DIGITAL SIGNATURE-BASED MECHANISMS

Authentication enables organizations to keep their networks secure by permitting only authenticated machines to communicate with other devices in the industrial network. Several Public-Key-Infrastructure (PKI)-based mechanisms provide authentication services for industrial networks. Based on Hash and XOR operations, the authors [11] propose a protocol for a lightweight authentication mechanism for M2M communication. There are two steps involved in the proposed mechanism to achieve authentication: (a) registration and (b) authentication. During the setup process, sensors are added to the Authentication Server (AS), and the AS generates and distributes pre-shared keys to the routers for use in subsequent phases. During the verification phase, the routers and sensors validate each other's identities. Because SECS/GEM is a point-to-point protocol, passively configured devices are limited to communicating with a single host at a time, while the proposed mechanism [11] relies on an authentication server to authenticate entities. Because of this, the proposed mechanism cannot be used for SECS/GEM equipment authentication.

Information authentication in IIoT systems was proposed using a certificate-less signature (CLS) scheme based on bilinear pairing in the aforementioned study [12]. Signature generation in this scheme calls for the signer to perform two exponentiations. However, in order for the verifier to verify a signature, two exponentiations and a pairing computation are needed. By presenting four different kinds of forgery attacks on signatures, the authors [13] proved that the CLS scheme does not provide the promised security. Therefore, the study [13] suggests improving the CLS scheme by introducing an elliptic curve cryptography–based Robust Certificateless Signature (RCLS) scheme. RCLS [13] is a robust cryptographic scheme that also provides defense against four signature forgery attacks, two of which are not addressed in the CLS scheme. Additionally, [14] have claimed RCLS is insecure by demonstrating how an attacker with the ability to replace a public key can easily impersonate other legitimate users in order to upload false messages. By forging the victim's valid signatures, the authors [14] demonstrated that this is possible, disproving the RCL scheme's claim that data integrity can be preserved indefinitely.

A lightweight authentication mechanism based on a hybrid Diffie-Hellman approach using AES and RSA for session key generation was proposed in [15]. The scheme allows for two-way authentication, securing messages against eavesdropping and replays while also protecting against Man-in-the-Middle (MITM) attacks. The advantage of a cryptographic hash-based message authentication code is used to ensure the message's cryptographic security. However, public-key encryption and reliance on Certificate Authority (CA) raise the total communication and computational overheads.

For the IoT ecosystem, [16] has developed a state-of-the-art authentication mechanism utilizing RSA public-key cryptography. The proposed mechanism provides a variety of security services, including X.509 certificate validation, RSA-based Public Key Infrastructure (PKI), and challenge/response protocols, in conjunction with a proxy-based security service provider. A novel system model, protocol design, architecture, and threat evaluation against established foes are all features of this approach. The proposed mechanism was selected for development as an ancillary service for a wide variety of mission-critical applications requiring X.509 certificates based on hard tokens, including smart cities, cyber-physical systems, etc. The proposed mechanism can be used with other security services, such as privacy, integrity, confidentiality, non-repudiation, and anonymity of the identities, thanks to the add-on service model.

A multi-key-based mutual authentication mechanism was presented in [17]. In this method, the secret shared between the IoT server and the IoT device is stored in a vault full of keys all of the same size. The server and the IoT device agree on the initial contents of the secure vault and then exchange vault contents after each successful communication session.

Current M2M authentication protocols proposed for IIoT networks have serious security flaws that leave networks vulnerable to a wide variety of cyberattacks, such as denial-of-service (DoS) attacks [18], router impersonation attacks, and smart-sensor tracing attacks. Based on the findings, it is possible for an intruder to gain access to the router's secret key and the session key being used by another smart device to establish an encrypted connection to the router.

In [19], the authors proposed an authentication protocol for IIoT networks and discussed problems faced by IoT devices with limited resources. The proposed mechanism is thought to be relatively lightweight because it employs elementary operations like XOR, addition/subtraction, and the hash function. The proposed mechanism only needs four messages passed between principals in order to authenticate the communicating network entities. Its security was successfully evaluated using the Automated Validation of Internet Security Protocols and Applications (AVISPA) tool and Burrows-Abadi-Needham (BAN) logic, and its resistance to known attacks was also successfully evaluated in an informal study.

Fix the problems of authentication and message integrity that arise from using heterogeneous devices in production [20]. The authors discuss the locally cloud-based Service-Oriented Architecture (SOA) Arrowhead Framework. The local clouds offer a prerequisite and supporting set of core systems to allow for industrial automation programs. To verify ownership and control access to the devices in a private cloud, one of these required backbone systems must be an Authentication, Authorization, and Accounting (AAA) system. The AAA needs to allow for granular access control in an enterprise setting with many different users. An AAA solution based on Next Generation Access Control (NGAC) is proposed to implement fine-grained -service-level access control among IoT devices and machines in an industrial network.

B. SECS/GEM SECURITY MECHANISMS

The criticality of cybersecurity issues in manufacturing has been recognized recently [21], and several studies have been carried out to recommend appropriate security mechanisms for Industry 4.0 and IIoT [22]. Relying on devices participating in the IIoT network is crucial to the smooth functioning of the network. A single hacked node may become malevolent, bringing the whole production line to a halt or causing catastrophes. Therefore, it is vital that the equipment and machinery interacting in the IIoT environment establish a reliable relationship and communicate only with trusted and authorized devices. The first line of defense against cyber-attacks on industrial networks is to make it difficult or impossible for adversaries to establish unauthenticated communication links with legitimate manufacturing equipment. Various studies address cybersecurity issues in the industry and propose authentication mechanisms as a potential solution.

1) SECS/GEMsec MECHANISM

The digital signature method is employed by the SECS/GEMsec mechanism to encrypt the message hash at the sender's end and decrypt it at the receiver's [23]. Algorithms for digital signatures use public-key cryptography, which is different from the symmetric-key cryptography used in most other applications. All SECS/GEM message hashes are encrypted using the RSA algorithm with a key size of 2048. The 2048-bit key size was selected to lessen the burden of controlling each individual message sent between a host and a piece of hardware. Even though a 4096-bit key is significantly more secure than a 2048-bit one, its control overhead is twice as high. To decrypt the message, the legitimate manufacturing equipment on the receiving end is already configured with the sender's public key. The receiver can safely accept the message as a legitimate message transmitted by the authorized equipment because it can't be forged without the sender's private key.

Message integrity is protected by SHA-256 in the SECS/GEMsec mechanism [23]. As of this writing, SHA-256 has not been cracked, offers a decent amount of security, and runs quickly on 32-bit machines, so it was selected. Transmission-required messages in SECS/GEMsec were hashed using SHA-256. The message signature is generated by first computing the message's hash value with the SHA-256 algorithm and then encrypting that hash with the RSA algorithm.

The value of a 4-byte field called SystemBytes in the SECS/GEM header is incremented monotonically with every request message. There are two essential purposes served by this area. Each new SECS/GEM request message will always have a unique SystemBytes value, so this feature serves to first ensure that messages remain up-to-date. As a second point, the SystemBytes value in the response message will always match that of the corresponding request message. Since the message signature cannot be forged without the private key, even if the attacker crafts a new message with an increased SystemBytes value, the new message will be rejected. Therefore, the third design goal is met by the SECS/GEMsec mechanism, as it is able to detect and discard messages with duplicate, stale, or forged SystemBytes values.

To accomplish the fourth goal, the SECS/GEMsec makes use of the TCP flow and appends the signature to the end of the SECS/GEM message without changing any field in the message structure. Due to the inclusion of the message signature in the TCP payload, no additional control message is required for hash transmission. SECS/GEMsec checks the message length value, which can be derived from the first 4 bytes of any HSMS message, to determine where to place the signature in the final portion of the HSMS transmission. In addition to streamlining the process, there is also no need to alter the message structure because of this.

2) SECURED SECS/GEM

The research conducted by [24] addressed the confidentiality issues found in standard implementations of SECS/GEM communication protocol. All communications between devices and the host are transmitted as unencrypted binary code. This is because the SECS/GEM protocols, in their original standard form, make no provision for encryption of the message data. This vulnerability opens the door for adversaries to exploit and cause communication disruptions in the live environment.

Industrial equipment generates data that provides insight into the system and enables operators to control equipment locally or remotely. Nevertheless, transmitting this highly sensitive and critical data in plaintext is exceedingly risky and can entice cybercriminals to target industrial equipment. A Secured SECS/GEM security mechanism for industrial communication networks has been proposed to address this problem. The Secured SECS/GEM ensures data confidentiality and authenticity without adding complexity or compromising processing speed.

The plaintext data, a nonce, and the previously shared key are the three components needed to implement the encryption mechanism. A 256-bit symmetric encryption key is used for the pre-shared key (32 bytes). The nonce is a 128-bit random number (16 bytes). Both the encryption algorithm and the hashing function used to create the message verification tag use it as an initialization vector. The plaintext information is the original payload of the HSMS message. The 256-bit key is protected by the AES-GCM 256 encryption algorithm [25] because a longer key provides greater protection against exhaustive brute-force attacks. A pre-shared key and the plaintext data are sent to the encryption mechanism in a secure SECS/GEM exchange. The internal counter of the encryption algorithm is initialized with a pseudorandom nonce generated on the fly. It is secure to share the nonce along with the message, so the same nonce is written to the data payload and is required at the receiver end to decipher the ciphertext. The first 16 bytes of the data payload are where the nonce is stored. After receiving the key and a portion of the nonce to use as an Initialization Vector (IV) for its internal counter, the encryption scheme encrypts the data as 128-bit blocks. Following the nonce, the ciphertext information is tacked onto the end of the message. After the encryption process is complete, the encryption mechanism creates a tag and appends it to the final 16 bytes of the encrypted message's payload. Essentially, this label is just a hash that was calculated by the encryption system. On the receiving end, the tag is used to ensure that the message has not been tampered with.

The proposed mechanism relies on a specific packet format that is read at the receiving end to access the HSMS message's data payload. Nonces are derived from the first 16 bytes of a message's payload. Since the tag generated by AES GCM takes up the last 16 bytes of the encrypted payload, that number is subtracted from the total payload length before reading the data. The 128-bit blocks of cyphertext are processed by the decryption mechanism. Once encryption has been broken, a tag is created by the decryption system. Any information tagged with this would be identical to that obtained via the encryption process. If the tags are the same, the message is accepted; if not, the payload's authenticity is compromised, and the message is dropped.

3) REPLAY-RESISTED SECS/GEM

The research conducted by [27] addressed the detection and prevention of replay attacks on SECS/GEM communications. A replay attack is a type of cyber-attack in which an attacker intercepts a valid message and retransmits it at a later time in order to trick the receiving system into thinking the message is still valid. The purpose of a replay attack is to bypass security measures and gain unauthorized access to a system

TABLE 1. Summary of the strengths and limitations of the related works on securing SECS/GEM.

Mechanism	Strengths	Limitations
Standard SECS/GEM [26]	N/A	(I) Offers no security; (II) No authentication; (III) No protection against cyber-attacks at all; (IV) No mechanism to ensure message integrity; and (V) No support confidentiality.
SECS/GEMsec [23]	(I) Offers Authentication; (II) Ascertains integrity; and (III) Detection and prevention of cyber-attacks, including DoS attacks, Replay Attacks, and FDIA attacks.	(I) Very high processing time; (II) High control overhead (i.e., it appends control data with each message exchanged; (III) It is a digital signature-based solution; thus, it requires multiple keys; (IV) The mechanism is hardcoded and does not offer to choose algorithms for hashing and digital signature; (V) Fixed algorithm, as well as key sizes (RSA=2048, SHA2-256), cannot be changed.
Secured SECS/GEM [24]	(I) Offers Confidentiality; and (II) Authenticates only Data Messages.	(I) Only encrypts payloads, thus cannot protects privacy; (I) Vulnerable to replay attacks; (III) It does not authenticate Control Messages; (IV) Vulnerable to DoS attack
Replay-Resisted SECS/GEM [27]	Addresses replay-attacks	(I) Vulnerable to FDIA attacks; (II) Vulnerable to DoS attacks; (III) Does not offer authentication; and (IV) Does encrypt messages.

or network. For example, an attacker may intercept an authentication message and replay it in order to gain access to a secure network. To prevent replay attacks, M2M protocols often include mechanisms such as message authentication codes (MACs) or digital signatures, which allow the receiving system to verify the authenticity and integrity of the message. Time stamps and sequence numbers can also be used to prevent replay attacks by ensuring that messages can only be processed once and in the correct order. However, a timestamp alone is not sufficient to prevent replay attacks, instead, a system can use timestamps in combination with other measures such as authentication. For example, a system could require that each request includes a unique nonce (a number used only once) in addition to a timestamp. The system could then use the timestamp to verify that the request was made within a certain time window, and use the nonce to ensure that the request has not been replayed. The mechanism [27] only addresses replay attacks and lacks important features such as protection against DoS attacks and FDIA attacks. It also does not address authentication, confidentiality, or message integrity, making it not suitable for commercial use.

C. CRITICAL REVIEW

This paper reviews digital signature-based mechanisms and SECS/GEM security mechanisms. A further layer of complexity is added to the aforementioned authentication mechanisms by the use of digital signature-based algorithms, which rely on Certification Authorities and multiple-key exchange mechanisms. Since the link is point-to-point and remains stable for weeks [28], there's no need for a Certificate Authority [29], [30]. When the mechanisms involved in a process or set of operations are more intricate, more time and data transfer capacity will be needed to complete them. It has been shown in studies [14] and [17] that the aforementioned mechanisms create new security holes. That is to say, adapting the security mechanism described above will make it easier for attackers to exploit the vulnerabilities of the current mechanisms and conduct attacks on the SECS/GEM communications, such as denial-of-service (DoS) attacks, impersonation attacks, and replay attacks. Because of this, data theft, a loss of trust in the network's security, and public disrepute are all possible outcomes.

While the customizability of security mechanisms enables their security to be adjusted in response to changing circumstances. Both SECS/GEMsec and Secured SECS/GEM are devoid of this freedom and do not provide flexibility to choose operations best suited to the situations. Thus, the situation demands a different solution, a security mechanism that encompasses tons of operations and presents users with features deemed fit to the scenarios without compromising performance or jeopardizing the message structure of the standard SECS/GEM protocol. Therefore, this thesis proposes a better solution - a security mechanism - that supersedes both SECS/GEMsec and Secured SECS/GEM and offers much more than these two mechanisms (i.e., SECS/GEMsec and Secured SECS/GEM) combined. Table 1 summarizes the strengths and limitations of the related works on securing SECS/GEM.

III. BACKGROUND

A. ASSUMPTIONS

Several assumptions are established for the development of a comprehensive security mechanism that is intended to accomplish the research objectives of this study and validate the expected outcomes. These assumptions are given as follows.

- Industrial equipment with SECS/GEM interface may exchange messages using either the SECS-I or the HSMS protocol; however, since the SECS-I protocol is outdated and only exists on legacy machines, this research is limited to the HSMS protocol.
- It is assumed that all participating entities (i.e., hosts and equipment) in the industrial network are configured with the ES-SECS/GEM mechanism.
- Cryptographic hash functions require a key to generate the message digest; thus, the two devices engaged in

communication must have each other's keys to verify the received message. Since ES-SECS/GEM is built atop SECS/GEM standard – a point-to-point protocol – secret keys are preconfigured on communicating devices during the initialization process; hence there is no need to employ a complex key distribution mechanism.
- It is assumed in this research that SECS/GEM communication takes place in an environment (wired or wireless) that is vulnerable to cyber-attacks, and adversaries can eavesdrop on the communication and launch attacks.

B. THREAT MODEL

In order to propose a sophisticated and comprehensive security mechanism that accomplishes the goals of this study, the threat model is defined as under:

- The attacker can monitor network traffic and is fully aware of the precise port number used for communication between a host and equipment. Given that the SECS/GEM messages have a non-secure design, an attacker can modify contents in the captured messages (both control message and data messages) to launch various attacks, including DoS attacks, Replay attacks [31], and FDIA attacks.
- In order to launch an attack, the adversary may examine every message exchanged between the two communicating entities and keep track of the most recent SystemBytes, knowing that the SystemBytes are incremented monotonically.
- The attacker may impersonate a host or equipment and transmit malicious messages, such as a recipe change or the termination of a communication connection, which results in an FDIA or DoS attack.
- Given that only an entity configured in active mode can initiate the communication establishment process, the attacker can masquerade and send a request on behalf of the legitimate host or equipment and prevent connection establishment, resulting in a DoS attack.

C. DESIGN OBJECTIVES

These three requirements are; a lightweight, multi-featured, and customizable scheme to perform authentication, message integrity, and prevention of cyberattacks, which can be achieved as follow:

- Lightweight: The lightweight requirement may be achieved by reducing the complexity of existing mechanisms for generating the message digest and then encrypting it in such a way that it is no longer readable. Due to the fact that complicated mechanisms such as SECS/GEMsec use SHA-256 and RSA to offer authentication, which ultimately requires more processing time. Thus, the complexity in SECS/GEMsec may be greatly reduced when merely a cryptographic hash function is used to accomplish the desired design objective. The resulting mechanism becomes lightweight and efficient by excluding RSA entirely from the proposed system. Furthermore, there will be no need to manually configure three distinct keys (i.e., one key for SHA2-256; two keys for RSA) on each computer in the industrial network.
- Multi-featured Solution: When testing and debugging an offline machine, it is not necessary to have sophisticated security mechanisms in place; instead, the standard SECS/GEM implementation is sufficient for this task. In addition, the mechanism should be adaptable and provide a variety of options depending on the scenarios. SECS/GEMsec only offers authentication while Secured SECS/GEM offers only confidentiality. However, a robust mechanism should provide multiple features as a bundle. This way, the mechanism would allow the user to select between several modes of operation. Hence, the proposed mechanism will incorporate multiple features (such as authentication, confidentiality, integrity, etc.) and will be suitable for a wide range of applications.
- Customizable: Because hardcoded security mechanisms are infeasible and do not allow for selecting other solutions, a comprehensive security mechanism is required to resolve these issues. The security mechanism will include its own message structure, header, and options fields, such as algorithm and key size, to be used to allow for selection among options without changing anything in the existing message structure of the HSMS protocol. Thus, the resulting mechanism will circumvent the limitation of having a hardcoded solution and provide a plethora of possibilities.

IV. PROPOSED ES-SECS/GEM MECHANISM

The proposed ES-SECS/GEM mechanism is intended to protect communications in industrial networks against cyber-attacks in order to achieve the research objectives. The proposed ES-SECS/GEM mechanism consists of three main stages. The first stage is called ACB (i.e., Authentication Code Block) Generation and Authentication. This stage aims to generate the ACB message structure and initialize it with the values supplied during the system configuration to authenticate equipment in the industrial networks. The second stage is achieving authentication and message integrity, which ascertains that the messages exchanged between the two communicating devices are not altered while in transit. The third stage, Attack Prevention, aims to define a rule-based mechanism and classify messages into legitimate and illegitimate for attack detection and prevention. The performance is measured in terms of processing time, control overhead incurred, and resilience against cyber-attacks. Figure 1 shows the main stages of the proposed ES-SECS/GEM security mechanism.

A. ACB GENERATION AND INITIALIZATION (STAGE 1)

This stage aims to achieve the first objective of this research, which is to prevent unauthorized devices from communicating and disrupting SECS/GEM communications in the manufacturing industry. To achieve this goal, the authentication

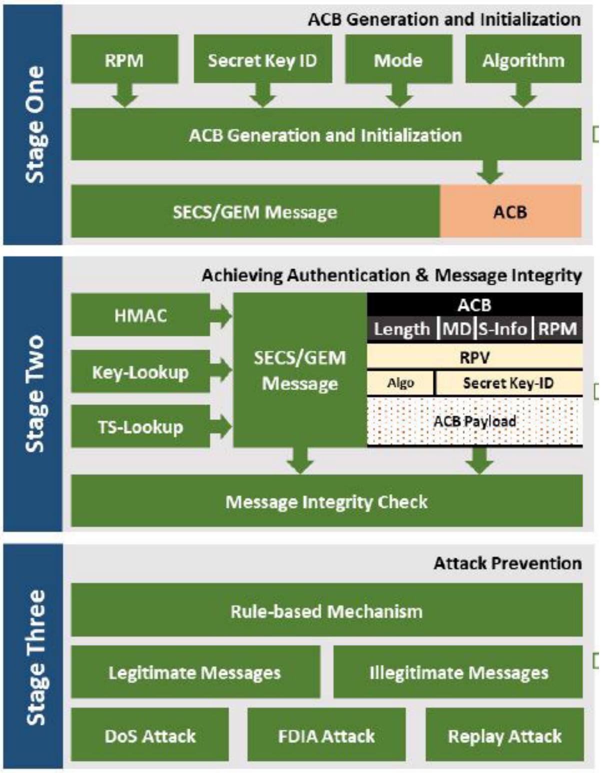
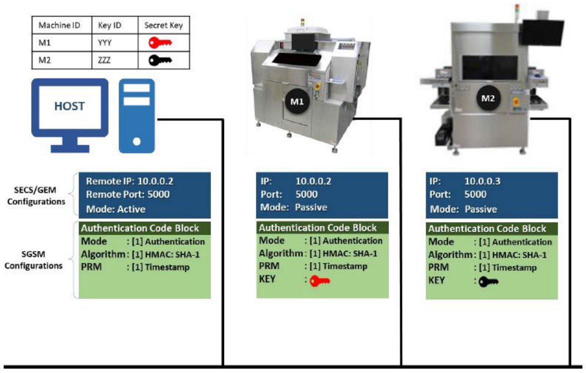


FIGURE 1. Architecture of ES-SECS/GEM mechanism.

mechanism is devised that only allows legitimate industry equipment to communicate once authenticated. The HMAC algorithm is adopted to generate a message digest utilizing various hashing algorithms (e.g., SHA-1, SHA-2, SHA-3, etc.). The generated message digest is thereafter appended to the messages exchanged between the two communicating devices. First of all, the sender is required to generate ACB with the specifications provided during the system configurations. The ACB is initialized with zeros because there is no control information or data to initialize it with during the initialization phase. The Secret-Key required to compute the hash value is stored in a protected file and is referenced with the Secret-Key ID. For security reasons, the secret key is stored separately in a file and is never shared between the host and the equipment; instead, the secret-key-id is referenced whenever the hash for a given message has to be computed. As mentioned previously, the mechanism for key distribution is beyond the scope of this study; keys are installed manually during system configuration. The host is usually communicating with several factory tools at a time, whereas the equipment is only connected to a single host. Figure 2 depicts the anticipated factory equipment equipped with the proposed ES-SECS/GEM mechanism on the shop floor.

B. AUTHENTICATION AND MESSAGE INTEGRITY (STAGE 2)

This stage aims to achieve the second objective of this research, which ensures the integrity of the SECS/GEM messages by preventing modifications to the messages during transit. It is essential to protect message integrity because attack actors can modify message contents and launch various attacks on the factory equipment/host. In order to protect message integrity, the proposed mechanism utilizes the cryptographic hash function to ascertain message integrity. The hash value is computed using a preinstalled secret key, and the generated hashing code is transmitted along with the SECS/GEM message. Upon receiving a message, the receiving entity will extract the hash value from the message and recompute the hash value for the verification process. The message will be considered authentic and unaltered only and only if the embedded hash value matches with the newly computed hash value on the receiving side.



Machine ID	Key ID	Secret Key
M1	YYY	🔑
M2	ZZZ	🔑

Remote IP: 10.0.0.2
Remote Port: 5000
Mode: Active

Authentication Code Block
Mode : [1] Authentication
Algorithm : [1] HMAC: SHA-1
PRM : [1] Timestamp

IP: 10.0.0.2
Port: 5000
Mode: Passive

Authentication Code Block
Mode : [1] Authentication
Algorithm : [1] HMAC: SHA-1
PRM : [1] Timestamp
KEY : 🔑

IP: 10.0.0.3
Port: 5000
Mode: Passive

Authentication Code Block
Mode : [1] Authentication
Algorithm : [1] HMAC: SHA-1
PRM : [1] Timestamp
KEY : 🔑

FIGURE 2. Architecture of ES-SECS/GEM mechanism.

1) PLACEMENT OF ACB IN HSMS MESSAGE

The ideal place to put ACB in an HSMS message is at the very end of the message. There are two methods that may be used to accomplish this. The following are the specifics of the two alternative methods that may be used:

- The HSMS length field is used to determine the entire message length, including the header and payload. Thus, it is conceivable to include the ACB length in the HSMS length bytes and generate enough space inside the message itself for the ACB.
- Instead of adding ACB size in the length field of the HSMS protocol, it is better to append the ACB to the HSMS message directly after the final byte of the HSMS message. This way, the ACB will start precisely where the HSMS message ended; hence, the receiver must inspect the HSMS length field and add 1 to the value to determine the location of the ACB.

Method-2, on the other hand, appends ACB to the TCP stream and makes no changes to the HSMS message structure or content; therefore, it does not suffer from overflow problems as Method-1 does. In order to keep things simple, Method 2 is adopted in this study. Figure 3 shows the ACB that has been attached to the HSMS message.

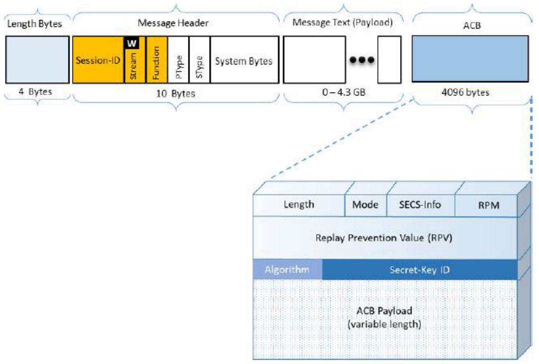


FIGURE 3. The Piggybacked ACB with SECS/GEM message.

2) HASHING-BASED MECHANISM

SECS/GEM operates in a trusted network; therefore, any device in the network may initiate a connection request, which the receiving entity is compelled to accept because there is no way to detect whether the incoming connection request originated from a legitimate entity or an attacker. This blind trust enables attackers to inject malicious data or conduct a DoS attack to disrupt communication. This can be avoided by exploiting security mechanisms that authenticate devices during the connection establishment phase; help in detecting messages that have been tampered with while in transit.

The Hash-based Message Authentication Code (HMAC) is essential for ensuring that the data exchanged between the two devices have not been tampered with during transit. The concept behind HMAC is straightforward; the sender computes the cryptographic hash over the block of data and a pre-shared key employing algorithms such as SHA-1, SHA-2, or SHA-3. The sender then sends both the data and the generated hash value to the receiver, which repeats the process and generates the hash over the data using the same key as the sender. The receiver then compares the two hash values and accepts the message if there is a match; otherwise, the message is discarded on the assumption that it was tampered with during transit.

An HMAC can be computed using any cryptographic hash function, such as SHA-2 or SHA-3; the resulting MAC algorithm is referred to as HMAC-X, where X is the hash function used in the calculation (e.g., HMAC-SHA256 or HMAC-SHA3-512). The HMAC's security is determined by the security of the underlying hash function, the size of the hash output, and the size and quality of the key. With HMAC, a hash is calculated twice. First, we use the secret key to generate the inner key and then the outer key. Initial processing of the algorithm results in a hash value computed from the message and the secret key. The final HMAC code is generated in the second pass from the inner hash result and the outer key. As a result, the algorithm is more resistant to length extension attacks.

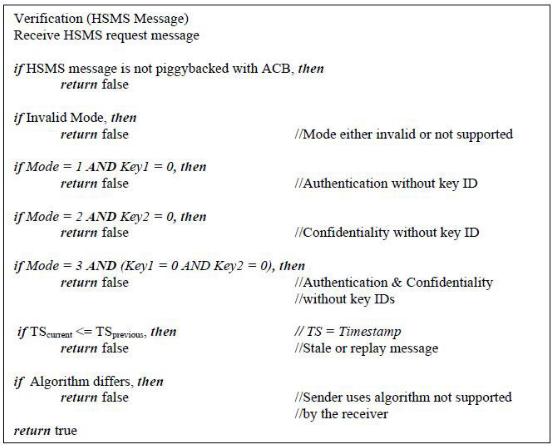
An iterative hash function uses a compression function to repeatedly iterate over message fragments of a predetermined size. SHA-256, for instance, uses chunks of data that are 512 bits in size. HMAC's output will be the same length as the hash function's input (256 bits for SHA-256, 512 bits for SHA-312, etc.), though it can be truncated if necessary.

C. PREVENTION OF CYBER-ATTACKS (STAGE 3)

This stage aims to prevent cyberattacks on SECS/GE communications in industrial networks by designing and implementing a rule-based mechanism that enables the sender and receiver to validate data and control messages without any intermediatory authority. Therefore, the third objective (i.e., attack prevention) is achieved through a rule-based mechanism that validates data and controls messages.

Rule-based Mechanism: The rule-based mechanism intends to enable the SECS/GEM entity to validate incoming messages independently of any other party or agent. The receiving entity is responsible for verifying the authenticity of both data and control messages in this mechanism. When performing a connection establishment process, the sender (i.e., usually host configured in active mode) must first generate the ACB with the required preconfigured settings and append it to the HSMS message. The ACB has several fields, including Length, which is the length of the entire ACB; Mode, which specifies the operation mode such as authentication; Algorithm, which specifies the HMAC algorithms such as MD5, SHA-1, etc.; and RPM, which specifies Nonce or timestamp value to retain the freshness of the message.

The sender must compute the hash using key1, embed it in ACB, and append it to HSMS messages for transmission. Upon receiving a message, the receiver must first check the presence of ACB accompanied by the message. The receiver then examines the ACB and validates the mode value and other various checks before computing the hash for verification. Once all checks are validated, the receiver will then compute the hash and compare it with the hash received with the message. The received message will be accepted if the resulting hash value matches the hash value embedded in the message; else, the message will be discarded. Figure 4 depicts the process of verifying the received message.



```
Verification (HSMS Message)
Receive HSMS request message

if HSMS message is not piggybacked with ACB, then
        return false

if Invalid Mode, then
        return false                        //Mode either invalid or not supported

if Mode = 1 AND Key1 = 0, then
        return false                        //Authentication without key ID

if Mode = 2 AND Key2 = 0, then
        return false                        //Confidentiality without key ID

if Mode = 3 AND (Key1 = 0 AND Key2 = 0), then
        return false                        //Authentication & Confidentiality
                                            //without key IDs

 if TS_current <= TS_previous, then         // TS = Timestamp
        return false                        //Stale or replay message

if  Algorithm differs, then
        return false                        //Sender uses algorithm not supported
                                            //by the receiver
return true
```

FIGURE 4. Rule-based controls for verifying HSMS messages.

D. THE WORKFLOW OF THE PROPOSED ES-SECS/GEM MECHANISM

ES-SECS/GEM addresses the main security-related issues faced in industrial networks and offers authentication, message integrity, and protection against cyber-attacks. The proposed mechanism specifies its own structure dubbed ACB to maintain control as well as data related to the security features. The ACB is required to be appended with all SECS/GEM messages on the sender in order to achieve the intended objectives. The validation is performed on both sender and receiver by distinguishing legitimate messages from illegitimate ones. The timestamp is used to maintain the freshness of the messages in order to avoid replay attacks. The workflow of the proposed ES-SECS/GEM mechanism is discussed as follows:

- The SECS/GEM entities can be configured as active or passive, which means only active entities can initiate connection establishment requests. Thus, a connection establishment request must be initiated by the trusted active entity (i.e., usually the host). The host must generate the ACB and must initialize it with values supplied during the configuration setup.
- The host must select the mode and security level for communication. There can be several modes supported by the proposed mechanism; however, this study limits to only authentication. The other optional mode for testing and debugging is supported without security, in which mode the proposed mechanism will emulate the functionality of standard SECS/GEM protocol without security features.
- RPM field is used to provide flexibility to provide a variety of options to choose appropriate identifiers to maintain the freshness of the message. The proposed mechanism uses timestamp as default; however, nonce or any other mechanism can be adapted in the future.
- The current time of the system is retrieved and placed in the RPM value field in ACB each time a message is required to be sent.
- The algorithm determined the first time when communication starts based on the selected algorithm and key size.
- The ACB's length field is then specified with the total size required for ACB. The length field is assigned a value in the end because it depends on the algorithm and key size chosen for the HMAC algorithm.
- At this stage, the ACB is initialized with the default values, and the sender requires a secret key in order to compute the hash for the given message. It is required to check whether the sender is acting as an active entity or not because usually, the host is communicating with several devices; thus, it requires a secret key already known to the receiver. The host has several keys, each for a separate device; thus, keys are stored in a secure vault and are mapped with a secret-key id. The equipment, on the other hand, only has one key; thus, it does not require a key retrieval process.
- The sender then examines the first four bytes of the HSMS message and determines the length of the message. The sender then appends ACB to the HSMS message.
- Once all required information is available to the sender and ACB is appended with the HSMS message, the sender then computes the Hash of the entire block, including the HSMS message and ACB combined.
- The generated Hash is then placed in the Payload area in the ACB, and it transmits the message to the intended destination.
- Upon receiving an HSMS message, the receiver checks for the ACB and immediately discards the message if ACB is not found.
- The receiver then extracts the Hash from ACB and stores it in a temporary variable while also zeroing off the ACB Payload.
- The receiver examines the ACB and determines whether the receiver entity is active; if yes, it refers to the secret-key id field of the ACB and retrieves the associated secret key from the vault in order to generate the Hash value on the receiver. This step is bypassed if the receiver is a passive entity and the key is directly known to the passive entity, so there is no need to retrieve it from the vault.
- The receiver then computes the Hash of the received message (both HSMS message and ACB) on the receiver and compares it with the Hash accompanied with the message received. The message can only be processed further if both Hash values match; the message is otherwise discarded.
- The receiver then extracts the timestamp value and checks whether the message received is the first message or not; if it is the first message, then it will accept the message and save the timestamp for future reference.
- In cases if the message is not the first message, then the receiver will retrieve the saved timestamp value extracted from the previous message and compare it with the timestamp of the current message; the message will only be accepted if the current timestamp value is newer than the value of the previous message.

This way, the authentication of the communicating entities is achieved because the sender is required to generate a hash for the connection establishment request message, without which the receiver will simply discard the message. The presence of ACB and hash for all received messages are counter-verified on the receiver; thus, message integrity is maintained and achieved. Without knowing the secret key, the attacker cannot inject anything into the message or modify it; consequently, the proposed mechanism protects SECS/GEM communications and prevents cyber-attacks.

V. SECURITY ANALYSIS

A. SECURITY ATTRIBUTE

1) ATTAINMENT OF AUTHENTICATION AND INTEGRITY

A security mechanism must first and foremost ensure that communication takes place only among authorized SECS/GEM-enabled devices. Failure to do so would result in various attacks, including the DoS attack. Keeping this under consideration, the ES-SECS/GEM appends the ACB to each message, providing the receiver with the authentication information necessary to verify the received message. To accomplish this, the sender uses an HMAC algorithm to generate a hash using a secret key that is pre-shared between the two communicating devices. The receiver regenerates a new hash using the same pre-shared key and compares the two hash values. If two hash values match, the message is accepted; otherwise, it is rejected. The attacker cannot communicate with the receiver without knowing the key, and breaking the key from a hash value is computationally infeasible. Thus, ES-SECS/GEM assures that communication takes place only among authenticated devices.

The hash attached to each message uniquely identifies the message. The attacker can still intercept the message and can extract relevant information from it, but they cannot modify the message contents. This is because attackers are unaware of the secret key, and without it, computing a new hash on the given message is not possible.

Secured SECS/GEM, on the other hand, attaches a nonce and a tag with each outgoing data message, which is required by the receiver to decrypt and authenticate the message. Secured SECS/GEM focuses on encrypting just data messages, preventing attackers from obtaining valuable information that, if revealed, could result in financial losses and damage the reputation. Because control messages are left unauthenticated and unencrypted, attackers have the opportunity to exploit these messages and conduct a variety of attacks, including a DoS attack.

SECS/GEMsec ensures that the communication takes place only between authenticated devices, and to achieve this, it attaches a signature with each message. Even though SECS/GEMsec provides authentication and successfully prevents attacks, it does so by encrypting the hash with the RSA algorithm, which makes it compute-intensive and exceedingly slow compared to the ES-SECS/GEM and Secured SECS/GEM protocols.

2) LIGHTWEIGHT

SECS/GEM-enabled devices produce massive amounts of data that provide valuable insight into equipment utilization, material consumption, and so on. This data must be generated and processed immediately to minimize errors and material waste. SECS/GEM standard is extremely quick and gives near-real-time insights. Hence, security mechanisms proposed for SECS/GEM must also be fast and have a low processing time overhead.

3) MULTI-FEATURED

The ES-SECS/GEM is versatile and offers a rich set of features that makes it superior to other mechanisms. ES-SECS/GEM has several operational modes which enable it to adapt to different situations according to industrial needs. For example, it can be configured in authentication mode in which it authenticates devices, maintains integrity, and prevents cyber-attacks. The messages in authentication mode are still transmitted in plaintext, which renders communication vulnerable to eavesdropping attacks, and attackers can steal business secrets. To prevent this, the ES-SECS/GEM can be configured in dual operational mode in which it not only authenticates messages but also encrypts them as well. It is computationally infeasible for attackers to launch a brute-force attack and breach ES-SECS/GEM security in order to damage the organization's reputation and steal proprietary information.

In contrast, Secured SECS/GEM and SECS/GEMsec are designed to offer confidentiality and authentication, respectively. There is no support for other features and the freedom to choose features as desired in a given situation with these mechanisms. Secured SECS/GEM only encrypts data messages and transmits control messages in plaintext, making it vulnerable to several attacks, including the DoS attack. SECS/GEMsec, on the other hand, authenticates devices and prevents cyber-attacks; however, it is devoid of confidentiality support; thus, attackers can monitor communication and steal business secrets.

4) ATTAINMENT OF CUSTOMIZABILITY

Industrial devices stay connected and operational on the shop floor for several weeks, if not months, and are powered off only during regular maintenance and hardware or software upgrades. Unnecessary shutdowns would result in financial losses for the industry; hence, a security mechanism for such devices must be customizable to avoid frequent upgrades or shutdowns. Considering that, the ES-SECS/GEM is designed to support different modes, and each mode provides different algorithms with varying key sizes to be selected depending on the desired security level. For example, for minimum security, ES-SECS/GEM can be operated in authentication mode with HMAC-SHA256. Based on the results, it is observed that HMAC-SHA256 incurs little processing time overhead, i.e., 0.31 and 0.17 for sending and receiving messages, respectively. However, the more robust security, the HMAC-SHA3-512, can be used without software upgrades. Customizability offers different security levels and avoids software upgrades which ultimately increased throughput and benefits businesses.

In contrast, SECS/GEMsec and Secured SECS/GEM address specific security issues and are rigid in their design. SECS/GEMsec uses the RSA algorithm with a fixed-size key (i.e., 2048 or 4096), which cannot be changed once configured. Similarly, Secured SECS/GEM is hardcoded with AES and does not allow a change of algorithm or key size.

This rigid design pattern limits the application of these two mechanisms on a commercial scale.

5) DoS ATTACK RESILIENT

One of the primary design objectives of ES-SECS/GEM is to prevent cyber-attacks, especially the DoS attack because the implications of such attacks are severe and cause substantial financial losses. DoS attacks are successful on standard SECS/GEM because it accepts and processes every message it receives, assuming that it is sent from legitimate factory equipment. Attackers leverage this vulnerability and inject a separate-req message, which terminates the connection immediately without waiting for an acknowledgment. ES-SECS/GEM appends an ACB with each message it transmits, which contains a hash value that determines whether the message is sent from a legitimate device or not. The receiver regenerates the hash and compares it with the hash contained in the received message. The message is accepted only if two hashes match; otherwise, it is rejected, considering a potential DoS or FDIA attack. DoS attacks carried out on different mechanisms were repeated 20 times. Table 2 depicts the results of the DoS attack prevention analysis.

TABLE 2. DoS attack prevention analysis.

Mechanism	Experiment Count [N]	Failed Preventing Attack [F]	APSR
Standard SECS/GEM [26]	20	20	0
SECS/GEMsec [23]	20	0	1
Secured SECS/GEM [24]	20	20	0
Replay-Resisted SECS/GEM [27]	20	20	0
ES-SECS/GEM (SHA256)	20	0	1

6) REPLAY ATTACK RESILIENT

It is possible to detect and prevent replay attacks using the ES-SECS/GEM mechanism because it is equipped with the necessary functionality. It is trivial for attackers to capture a packet and then send it to the victim at a later stage. In order to determine whether the message is fresh or replayed, the timestamp is extracted and saved by the receiver after each message is received and accepted. This is required to validate the freshness of the next message received. The messages are only accepted if the timestamp of the current message is greater than the timestamp of the previous message that the system has previously accepted.

ES-SECS/GEM successfully detected and prevented attacks carried out on SECS/GEM communications on each attempt making it superior when compared with other mechanisms. SECS/GEMsec also prevented replay attacks; however, its processing time is very high, i.e., around 20 milliseconds, whereas ES-SECS/GEM has a processing time lower than one millisecond. Secured SECS/GEM, on the other hand, is vulnerable to replay attacks and fails to detect and prevent such attacks. Results of Replay attacks are shown in Table 3.

TABLE 3. Replay attack prevention analysis.

Mechanism	Experiment Count [N]	Failed Preventing Attack [F]	APSR
Standard SECS/GEM [26]	20	20	0
SECS/GEMsec [23]	20	0	1
Secured SECS/GEM [24]	20	20	0
Replay-Resisted SECS/GEM [27]	20	0	1
ES-SECS/GEM (SHA256)	20	0	1

7) FDIA ATTACK RESILIENT

The FDIA attack requires that malicious content must be injected into an already established connection between a host and the equipment. In order to achieve this, the attacker must craft an attack message that contains precise information such as SystemBytes, the SType value, the SessionID, etc. Once the message is ready to be injected, the attackers must generate the hash and place it into the ACB; otherwise, the receiver will discard the message. The attacker cannot generate the hash because the key is unknown to the attacker. Alternatively, the attacker can capture the message transmitted over the network and manipulate it with malicious information. However, altering the content of the message results in a different hash value when computed on the receiver; consequently, the received message will be discarded due to the mismatch of the two hash values. The ES-SECS/GEM's authentication process prevents attackers from establishing connections with authorized industry equipment, and the captured message cannot be manipulated and sent to the victim. Thus, ES-SECS/GEM successfully detects and prevents FDIA attacks.

Table 4 Summarized Results of Cyber-attacks summarizes the findings of an experiment conducted to determine the robustness of ES-SECS/GEM against FDIA attacks. As seen in the table, all attempts to inject malicious content were detected and prevented effectively.

TABLE 4. FDIA attack prevention analysis.

Mechanism	Experiment Count [N]	Failed Preventing Attack [F]	APSR
Standard SECS/GEM [26]	20	20	0
SECS/GEMsec [23]	20	0	1
Secured SECS/GEM [24]	20	20	0
Replay-Resisted SECS/GEM [27]	20	20	0
ES-SECS/GEM (SHA256)	20	0	1

TABLE 5. Security attributes comparison.

Attribute	Standard SECS/GEM [26]	SECS/GEMsec [23]	Secured SECS/GEM [24]	Replay-Resisted SECS/GEM [27]	ES-SECS/GEM
Multi-featured	✗	✗	✗	✗	✓
Customizable	✗	✗	✗	✗	✓
Authentication	✗	✓*	✓	✗	✓
Lightweight	-	✓	✗	✓	✓
DoS Attack Resilient	✗	✗	✓	✗	✓
Replay Attack Resilient	✗	✗	✓	✓	✓
FDIA Attack Resilient	✗	✗	✓	✗	✓

B. SECURITY COMPARISON

Secured SECS/GEM is entirely vulnerable to cyber-attacks because it only encrypts and authenticates data messages; the control messages are left unprotected. Although the SECS/GEMsec mechanism prevents cyber-attacks, it is highly compute-intensive and imposes 19.05 and 20.36 milliseconds processing time overhead on sending and receiving messages, which makes it infeasible for commercial use. ES-SECS/GEM is a feature-rich security mechanism enabling various modes and algorithms to be selected depending on the security level required. It suppresses other security mechanisms and prevents cyber-attacks. Results presented in Table 5 indicated that ES-SECS/GEM outperformed other mechanisms in terms of preventing attacks such as DoS attacks, Replay attacks, and FDIA attacks.

Table 5 summarizes each mechanism's performance measured under different attributes. It can be observed that ES-SECS/GEM obtained better results in all aspects covered in this study.

VI. EXPERIMENTS AND RESULTS

Several metrics were used to compare the proposed Match-Prevention method to the state-of-the-art, and two experiment scenarios (normal and attack) were used to determine if the method met the criteria set forth in the study. Time to process, data transfer rates, and successful DDoS mitigation attempts were all measured in the same ways as in earlier studies.

A. TESTBED SETUP AND IMPLEMENTATION TOOLS

SEMI's SECS/GEM specifications served as the basis for the development of the proposed ES-SECS/GEM mechanism. An industrial IT security lab assesses the effectiveness of the scenarios used to gauge resistance to cyberattacks. Two computers, one serving as the host and the other as the equipment, are set up with Python SECS/GEM implementations [32]. Because the attackers in the experiment have broken through the network's defenses and the firewall, they can listen in on the conversation and sniff the data being transferred between the host and the device. The Testbed environment, as shown in Figure 5, represents a typical industrial network architecture with attackers present.

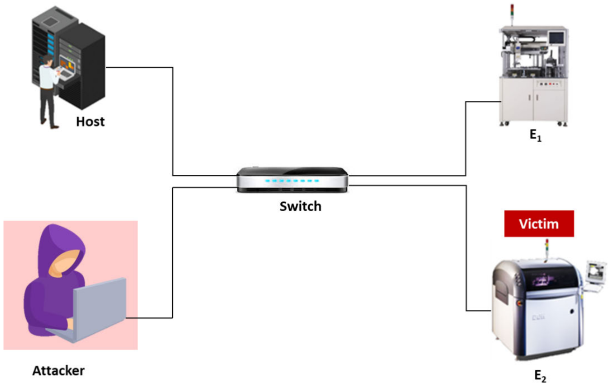


FIGURE 5. Testbed environment for SECS/GEM communications.

Since SECS/GEM is a point-to-point protocol, the connection establishment request must come from one of the communicating entities (either the host or the equipment). While it is possible to set up either entity in active mode, in industrial networks the host is typically configured that way, so that is what we have done. Equipment setup is typically passive (i.e., the entity configured in passive mode will open up a port and listen for incoming connection requests). The hostile actor is ready to launch a denial-of-service attack against SECS/GEM channels. The specifications of both the host and the equipment are identical to those of newly purchased machines in the semiconductor industry in 2020. The following is a list of the hardware and software used in the experiments and their respective specifications.

- Host: This device role has the specifications of Intel(R) Core(TM) i7-9750H @ 2.60GHz x 6, the memory of 8 GB, and worked on the operating system (Win-10).
- Equipment: This device role has the specifications of Intel(R) Core(TM) i5-6500 @ 3.20 GHz x 4, the memory of 2 GB, and worked on the operating system (Win-10).
- Attacker: This device role has the specifications of Intel(R) Core(TM) Ci3-330M @ 2.13GHz x 2, the memory of 8 GB, and worked on the operating system (Kali Linux 2020.3).
- Switch: This device role has the specifications of Cisco Catalyst 2960 Fast Ethernet.

B. PROCESSING TIME

This section discusses the processing time required by the ES-SECS/GEM's processes for generating and verifying the control and data messages exchanged between sender and receiver in the industrial network

The processing time of ES-SECS/GEM is measured with HMAC-SHA1, HMAC-SHA256, HMAC-SHA512, HMAC-SHA3-256, and HMAC-SHA3-512. Although SHA1 is broken and may not be suitable for commercial usage, however, without knowing the key, the hash values generated with HMAC-SHA1 are relatively difficult to break. Therefore, due to the smaller key size (i.e., 160 bits or 20 bytes), ES-SECS/GEM is also evaluated with SHA1. The processing time for measuring these processes at the sender and receiver is explained in the sub-sections below.

1) SENDER MESSAGES

Figure 6 shows that for ES-SECS/GEM(SHA256) and ES-SECS/GEM(SHA512), the average processing times are 0.52 and 0.55 milliseconds, respectively. While ES-SECS/GEM(SHA512) doubles the key size and provides more robust security when compared to ES-SECS/GEM(SHA256), the performance penalty of 0.03 milliseconds, can be considered negligible in most circumstances.

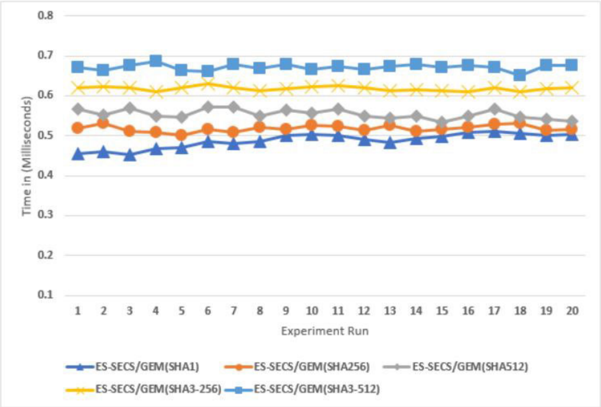


FIGURE 6. Timing of ES-SECS/GEM Mechanism Processing at the sender.

The advantage of using ES-SECS/GEM is that it offers several options to choose from, making it the most suitable candidate for commercial use. For example, when speed is the most critical aspect of the system, ES-SECS/GEM can be used in conjunction with either SHA1 or the SHA256 mechanism because both of these mechanisms have a shorter processing time, i.e., 0.49 and 0.52 milliseconds, for sending and receiving messages, respectively. On the other hand, the ES-SECS/GEM with SHA3-512 is ideal for situations where security is the utmost important factor. The ES-SECS/GEM (SHA3-512) is recommended for the highest security because it is fast and has not been compromised yet.

2) RECEIVING MESSAGES

As with the sender, the experiments were repeated 20 times at the receiver, and the results were averaged in order to satisfy computer science requirements (Devore, 2016). The receiver must detach the ACB upon receiving a message and undertake verification and validation to decide whether to accept or reject the message.

Measuring the processing time required to receive a message entails taking into account all of the procedures mentioned above performed by the receiver, which ultimately contributes to attaining appropriate security at the expense of higher processing time. As expected, the shortest processing time for receiving request messages at equipment is 0.04 milliseconds for standard SECS/GEM, whereas the processing time of 0.2 and 0.27 milliseconds are observed for ES-SECS/GEM(SHA1) and ES-SECS/GEM(SHA3-512), respectively. The processing time for messages received at the receiver is depicted in Figure 7.

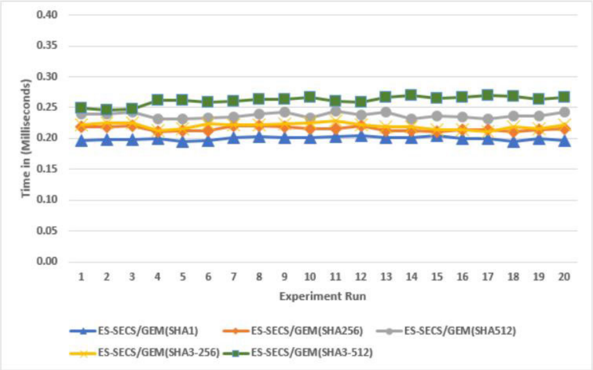


FIGURE 7. Timing of ES-SECS/GEM Mechanism Processing at the Recipient.

3) TOTAL PROCESSING TIME

The experiments conducted under a normal scenario are repeated 20 times, where each experiment yielded a different processing time. Thus, the average processing time (mean), standard deviation (SD), and overhead induced in generating and verifying data and control messages are calculated and presented. The overhead is computed by using the mean processing time of Standard SECS/GEM as the baseline, as shown below:

$$Overhead = Mean(X_{mechansim}) - Mean(StandardSECSGEM) \quad (1)$$

where Overhead is the processing time induced by different security mechanisms, Mean (XMechanism) is the average processing time measured during 20 different experiments for the given mechanism (i.e., SECS/GEMsec, Secured SECS/GEM, and ES-SECS/GEM), and Mean (StandardSECSGEM) is the average processing time computed for Standard SECS/GEM mechanism.

The experimental results illustrated in Figure 8 indicate that the ES-SECS/GEM mechanism has the lowest processing time (i.e., 0.31 and 0.17 milliseconds for sending and receiving messages, respectively) compared with Secured

TABLE 6. Summary of control overhead.

	Control Message (10 bytes)		Data Message (100 bytes)	
Mechanism	Actual	Overhead	Actual	Overhead
Standard SECS/GEM [26]	14	0	114	0
SECS/GEMsec [23]	14	512	612	512
Secured SECS/GEM [24]	14	0	132	32
Replay-Resisted SECS/GEM [27]	14	8	108	8
ES-SECS/GEM	14	48	148	48

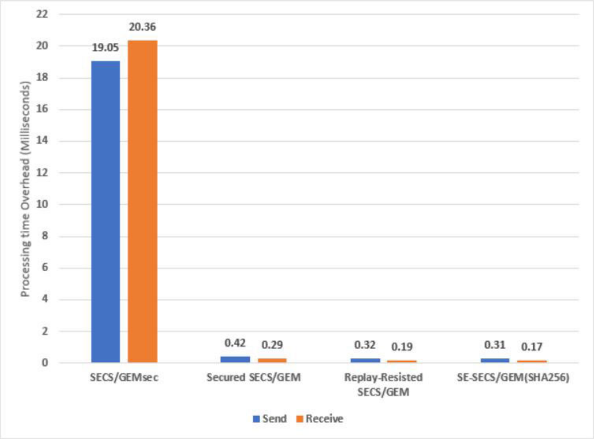


FIGURE 8. Overhead processing time comparison.

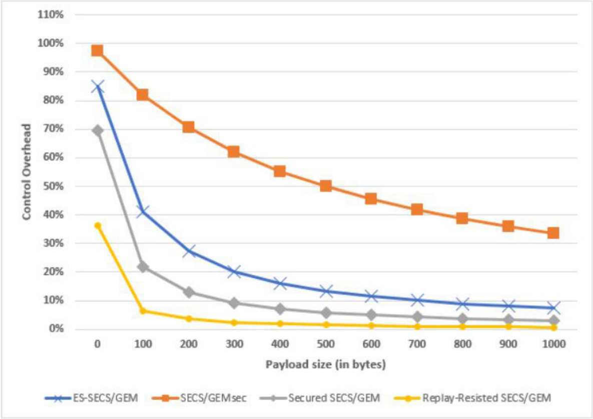


FIGURE 9. ES-SECS/GEM: control overhead Vs. message size.

SECS/GEM and SECS/GEMsec mechanisms. Hence, the ES-SECS/GEM fulfills the lightweight requirements. Compared with the standard SECS/GEM, the ES-SECS/GEM's processing time is slightly higher, which is expected given the cost of securing the SECS/GEM communications from cyber-attacks.

C. CONTROL OVERHEAD

SECS/GEM is an incredibly efficient protocol in terms of data packaging and achieves high data density; hence, it transports data efficiently and consumes less network bandwidth. For this very reason, a security mechanism for SECS/GEM must adhere to the underlying principle and avoid imposing a significant control overhead on devices to attain authentication and prevent cyber-attacks.

Standard SECS/GEM does not incur control overhead because it is devoid of security features and requires no need to transmit control data to the receiver for the verification process. In contrast, the security mechanisms outlined in the preceding subsections provide security features, which entail adding control information to the message, hence incurring control overhead. Table 6 summarizes the control overhead observed in all mechanisms studied in the preceding subsections.

Table 6 illustrates the control overhead of the given mechanisms for both control messages and data messages (100-byte payload). SECS/GEMsec is believed to have the highest control overhead because of its large key size (i.e., 4096-bit key). Secured SECS/GEM 32-bytes has the lowest control overhead; unfortunately, it only secures data messages, making it subject to cyber-attacks and unsuitable for usage in a production environment. ES-SECS/GEM protects both data and control messages and protects against cyber-attacks, with a control overhead of 48 bytes when used in conjunction with a 256-bit HMAC algorithm. Secured SECS/GEM and ES-SECS/GEM(SHA256) control overheads decrease rapidly as message size increases; however, the control overhead for SECS/GEMsec with payload size=1000 bytes is roughly 34%, which is regarded as significantly high in comparison with other mechanisms. Figure 9 illustrates the effect of increased message size on control overhead.

VII. CONCLUSION

The industry 4.0 ecosystem is built on integrating all functional units to create a streamlined production environment with high throughput and little or no human interaction required to perform daily tasks. While the machine-to-machine communication protocols used in the industrial ecosystem are lightweight and highly optimized for performance, the majority of these protocols lack security features, leaving them vulnerable to cyberattacks. Thus, utilizing security-less protocols in today's digital environment is extremely dangerous and may compromise crucial corporate secrets if precautions are not taken. This research is the first of its kind in this direction, addressing security vulnerabilities found in standard SECS/GEM and proposing an efficient, lightweight, multi-featured, and customizable mechanism for preventing cyber-attacks in the manufacturing industry.

The proposed ES-SECS/GEM mechanism is accomplished by appending an Authentication Code Block (ACB) to each message exchanged between a host and equipment in an industrial network. Maintaining the message structure while ensuring security was a challenge that was overcome by attaching ACB to each message transmitted over the network. Meanwhile, the proposed ES-SECS/GEM mechanism designed a rule-based mechanism as a knowledge-based system concept to allow proposal-enabled devices to decide whether the received data or control messages are coming from a legitimate or illegitimate device. Only authorized devices (i.e., the sender and receiver) have access to a secret key, without which attackers cannot regenerate the hash. Finally, the evaluation was conducted in two different scenarios: normal and attack. The normal scenario was used to evaluate the proposed mechanism's performance in terms of processing time and control overhead, while the attack scenario was used to evaluate the ES-SECS/GEM's resilience to cyber-attacks.

REFERENCES

[1] M. Javaid, A. Haleem, R. P. Singh, and R. Suman, "Artificial intelligence applications for industry 4.0: A literature-based study," *J. Ind. Integr. Manage.*, vol. 7, no. 1, pp. 83–111, Mar. 2022.

[2] A. Kumar, R. Agrawal, V. A. Wankhede, M. Sharma, and E. Mulat-Weldemeskel, "A framework for assessing social acceptability of industry 4.0 technologies for the development of digital manufacturing," *Technol. Forecasting Social Change*, vol. 174, Jan. 2022, Art. no. 121217.

[3] A. Haleem, M. Javaid, R. P. Singh, S. Rab, and R. Suman, "Perspectives of cybersecurity for ameliorative industry 4.0 era: A review-based framework," *Ind. Robot, Int. J. Robot. Res. Appl.*, vol. 49, no. 3, pp. 582–597, Apr. 2022.

[4] R. Raman and A. Kumar, "Potential, scope, and challenges of industry 4.0," in *A Roadmap for Enabling Industry 4.0 by Artificial Intelligence*. Hoboken, NJ, USA: Wiley, 2023, p. 201.

[5] M. Zhu, X. Peng, Y. Sun, S. Fuyang, and D. Jiao, "Simulation study of semiconductor communication protocol SECS/GEM," in *Proc. Int. Conf. Wireless Commun. Smart Grid (ICWCSG)*, Aug. 2021, pp. 148–152.

[6] S. A. Laghari, S. Manickam, and S. Karuppayah, "A review on SECS/GEM: A machine-to-machine (M2M) communication protocol for industry 4.0," *Int. J. Electr. Electron. Eng. Telecommun.*, vol. 10, no. 2, pp. 105–114, 2021.

[7] B. Williams, M. Soulet, and A. Siraj, "A taxonomy of cyber attacks in smart manufacturing systems," in *Proc. 6th EAI Int. Conf. Manage. Manuf. Syst.* Cham, Switzerland: Springer, 2023, pp. 77–97.

[8] H. Giberti, T. Abbattista, M. Carnevale, L. Giagu, and F. Cristini, "A methodology for flexible implementation of collaborative robots in smart manufacturing systems," *Robotics*, vol. 11, no. 1, p. 9, Jan. 2022.

[9] S. Morgan, "Report: Cyberwarfare in the C-suite," in *Cybercrime Magazine*, 2021. [Online]. Available: https://cybersecurityventures.com/wp-content/uploads/2021/01/Cyberwarfare-2021-Report.pdf

[10] S. Peng, "The real reason behind the TSMC cyber attack," CommonWealth Mag., Nov. 2018. [Online]. Available: https://english.cw.com.tw/article/article.action?id=2194

[11] A. Esfahani, G. Mantas, R. Matischek, F. B. Saghezchi, J. Rodriguez, A. Bicaku, S. Maksuti, M. G. Tauber, C. Schmittner, and J. Bastos, "A lightweight authentication mechanism for M2M communications in industrial IoT environment," *IEEE Internet Things J.*, vol. 6, no. 1, pp. 288–296, Feb. 2019.

[12] A. Karati, S. K. H. Islam, and M. Karuppiah, "Provably secure and lightweight certificateless signature scheme for IIoT environments," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3701–3711, Aug. 2018.

[13] Y. Zhang, R. Deng, D. Zheng, J. Li, P. Wu, and J. Cao, "Efficient and robust certificateless signature for data crowdsensing in cloud-assisted industrial IoT," *IEEE Trans. Ind. Informat.*, vol. 15, no. 9, pp. 5099–5108, Sep. 2019.

[14] W. Yang, S. Wang, X. Huang, and Y. Mu, "On the security of an efficient and robust certificateless signature scheme for IIoT environments," *IEEE Access*, vol. 7, pp. 91074–91079, 2019.

[15] K. Mahmood, S. A. Chaudhry, H. Naqvi, T. Shon, and H. F. Ahmad, "A lightweight message authentication scheme for smart grid communications in power sector," *Comput. Elect. Eng.*, vol. 52, pp. 114–124, May 2016.

[16] T. Shah and S. Venkatesan, "Authentication of IoT device and IoT server using secure vaults," in *Proc. 17th IEEE Int. Conf. Trust, Secur. Privacy Comput. Commun./12th IEEE Int. Conf. Big Data Sci. Eng. (TrustCom/BigDataSE)*, Aug. 2018, pp. 819–824.

[17] S. F. Aghili and H. Mala, "Breaking a lightweight M2M authentication protocol for communications in IIoT environment," Cryptol. ePrint Arch., Rep. 2018/891. Accessed: Mar. 29, 2023. [Online]. Available: https://eprint.iacr.org/2018/891.pdf

[18] M. A. Al-Shareeda, S. Manickam, and M. A. Saare, "DDoS attacks detection using machine learning and deep learning techniques: Analysis and comparison," *Bull. Electr. Eng. Informat.*, vol. 12, no. 2, pp. 930–939, Apr. 2023.

[19] E. Lara, L. Aguilar, M. A. Sanchez, and J. A. García, "Lightweight authentication protocol for M2M communications of resource-constrained devices in industrial Internet of Things," *Sensors*, vol. 20, no. 2, p. 501, Jan. 2020.

[20] K. K. Kolluru, C. Paniagua, J. van Deventer, J. Eliasson, J. Delsing, and R. J. DeLong, "An AAA solution for securing industrial IoT devices using next generation access control," in *Proc. IEEE Ind. Cyber-Phys. Syst. (ICPS)*, May 2018, pp. 737–742.

[21] M. Shahin, F. Chen, H. Bouzary, A. Hosseinzadeh, and R. Rashidifar, "Classification and detection of malicious attacks in industrial IoT devices via machine learning," in *Proc. Int. Conf. Flexible Automat. Intell. Manuf.* Cham, Switzerland: Springer, 2023, pp. 99–106.

[22] X. Yu and H. Guo, "A survey on IIoT security," in *Proc. IEEE VTS Asia Pacific Wireless Commun. Symp. (APWCS)*, Aug. 2019, pp. 1–5.

[23] S. U. A. Laghari, S. Manickam, A. K. Al-Ani, S. U. Rehman, and S. Karuppayah, "SECS/GEMsec: A mechanism for detection and prevention of cyber-attacks on SECS/GEM communications in industry 4.0 landscape," *IEEE Access*, vol. 9, pp. 154380–154394, 2021.

[24] A. Jaisan, S. Manickam, S. A. Laghari, S. U. Rehman, and S. Karuppayah, "Secured SECS/GEM: A security mechanism for M2M communication in industry 4.0 ecosystem," *Int. J. Adv. Comput. Sci. Appl.*, vol. 12, no. 8, pp. 1–11, 2021.

[25] M. N. Alenezi, H. Alabdulrazzaq, and N. Q. Mohammad, "Symmetric encryption algorithms: Review and evaluation study," *Int. J. Commun. Netw. Inf. Secur.*, vol. 12, no. 2, pp. 256–272, 2020.

[26] SSCP Agreements. (2020). *Semi E30—Specification for the Generic Model for Communications and Control of Manufacturing Equipment (GEM)*. [Online]. Available: https://www.semi.org/en

[27] M. A. Al-Shareeda, S. Manickam, S. A. Laghari, and A. Jaisan, "Replay-attack detection and prevention mechanism in industry 4.0 landscape for secure SECS/GEM communications," *Sustainability*, vol. 14, no. 23, p. 15900, Nov. 2022.

[28] B. Adrien, P. Isabel, and G. João, "Artificial intelligence, cyber-threats and industry 4.0: Challenges and opportunities," *Artif. Intell. Rev.*, vol. 54, pp. 3849–3886, Feb. 2021.

[29] E. F. Terng, S. C. Yeoh, K. C. Tong, and K. S. Yeo, "Data analysis on SMT reflow oven with SECS/GEM communication protocol," in *Proc. IEEE 10th Symp. Comput. Appl. Ind. Electron. (ISCAIE)*, Apr. 2020, pp. 118–124.

[30] O. A. Amodu and M. Othman, "Machine-to-machine communication: An overview of opportunities," *Comput. Netw.*, vol. 145, pp. 255–276, Nov. 2018.

[31] M. A. Al-shareeda, M. Anbar, I. H. Hasbullah, S. Manickam, N. Abdullah, and M. M. Hamdi, "Review of prevention schemes for replay attack in vehicular ad hoc networks (VANETs)," in *Proc. IEEE 3rd Int. Conf. Inf. Commun. Signal Process. (ICICSP)*, Sep. 2020, pp. 394–398.

[32] P. W. D. Charles. (2023). *Simple Python SECS/GEM Implementation*. [Online]. Available: https://github.com/bparzella/secsgem

SHAMS UL ARFEEN LAGHARI received the B.Sc. (Hons.) and M.Sc. degrees in computer science from the University of Sindh, Jamshoro, Pakistan, and the M.S. degree in computer science from PAF-KIET, Karachi, Pakistan. He is currently pursuing the Ph.D. degree in network security with the National Advanced IPv6 Centre, Universiti Sains Malaysia. His research interests include cybersecurity, Industry 4.0, distributed systems, cloud computing, and mobile cloud computing.

SELVAKUMAR MANICKAM is currently the Director of the National Advanced IPv6 Centre and an Associate Professor specializing in cybersecurity, the Internet of Things, Industry 4.0, cloud computing, big data, and machine learning. Previously, he was with Intel Corporation and a few start-ups working in related areas before moving to academia. He has authored or coauthored more than 220 papers in journals, conference proceedings, and book reviews. He has graduated with 18 Ph.D. students in addition to bachelor's and master's students. He has given several keynote speeches and dozens of invited lectures and workshops at conferences, international universities, and industry. He has given talks and training on internet security, the Internet of Things, Industry 4.0, IPv6, machine learning, software development, and embedded and OS kernel technologies at various organizations and seminars. He also lectures in various computer science and IT courses, including developing new courseware in tandem with current technology trends. He is involved in various organizations and forums locally and globally. While building his profile academically, he is still very involved in industrial projects involving industrial communication protocol, robotic process automation, machine learning, and data analytics using open-source platforms. He also has experience in building the IoT, embedded, server, mobile, and web-based applications.

AYMAN KHALLEL AL-ANI received the Ph.D. degree in advanced computer networks from Universiti Sains Malaysia (USM). He is currently a Senior Lecturer with the Faculty of Computing and Informatics, Universiti Malaysia Sabah (UMS). His current research interests include malware detection, web security, intrusion detection systems (IDS), intrusion prevention systems (IPS), network monitoring, the Internet of Things (IoT), IPv6 security, artificial intelligence, machine learning, data mining, and optimization algorithms.

MAHMOOD A. AL-SHAREEDA received the Ph.D. degree in advanced computer networks from University Sains Malaysia (USM). He is currently a Postdoctoral Fellow with the National Advanced IPv6 Centre (NAv6), USM. His current research interests include network monitoring, the Internet of Things (IoT), vehicular ad hoc network (VANET) security, and IPv6 security.

SHANKAR KARUPPAYAH received the B.Sc. degree (Hons.) in computer science from Universiti Sains Malaysia, in 2009, the M.Sc. degree in software systems engineering from the King Mongkut's University of Technology North Bangkok (KMUTNB), in 2011, and the Ph.D. degree from TU Darmstadt, in 2016. His Ph.D. dissertation is titled "Advanced Monitoring in P2P Botnets." He has been a Senior Lecturer with the National Advanced IPv6 Centre (NAv6), Universiti Sains Malaysia, since 2016. He has also been a Senior Researcher/Postdoctoral Researcher with the Telecooperation Group, TU Darmstadt, since July 2019. He is currently working on several cyber-security projects and groups, such as the National Research Center for Applied Cybersecurity (ATHENE), formerly known as the Center for Research in Security and Privacy (CRISP).