sustainability

MDPI

Article

Replay-Attack Detection and Prevention Mechanism in Industry 4.0 Landscape for Secure SECS/GEM Communications

Mahmood A. Al-Shareeda *, Selvakumar Manickam *, Shams A. Laghari and Ashish Jaisan

National Advanced IPv6 Centre (NAv6), Universiti Sains Malaysia, Gelugor 11800, Penang, Malaysia
* Correspondence: alshareeda022@usm.my (M.A.A.-S.); selva@usm.my (S.M.)

Abstract: Starting from the First Industrial Revolution to the current and Fourth Industrial Revolution (or Industry 4.0), various industrial machines are present in the market and manufacturing companies. As standardized protocols have become increasingly popular, more utilities are switching to Internet Protocol (IP)-based systems for wide-area communication. SECS/GEM is one of the standards that permit industries to collect information directly from the machines, either using RS323 or TCP/IP communication. TCP/IP communication is becoming more critical than ever, especially given our accelerated digital transformation and increasing reliance on communication technologies. The growth of IT is accelerating with cyberthreats as well. In contrast, security features in the SECS/GEM protocol may be neglected by some companies as it is only used in factories and not mostly used in the outside world. However, communication of SECS/GEM is highly susceptible to various cyberattacks. This paper analyzes the potential replay-attack cyberattacks that can occur on a SECS/GEM system. In replay attacks, this paper supposes an adversary that wants to damage an operation-based control system in an ongoing condition. The adversary has the ability to capture messages to watch and record their contents for a predetermined amount of time, record them, and then replay them while attacking in order to inject an exogenous control input undetected. The paper's objectives are to prove that SECS/GEM communication is vulnerable to cyberattack and design a detection mechanism to protect SECS/GEM communications from replay attacks. The methodology implements a simulation of the replay-attack mechanism on SECS/GEM communication. The results indicate that the design mechanism detected replay attacks against SECS/GEM communications and successfully prevented them.

Keywords: SECS/GEM communications; Industry 4.0 landscape; replay attack; detection and prevention mechanism

Citation: Al-Shareeda, M.A.; Manickam, S.; Laghari, S.A.; Jaisan, A. Replay-Attack Detection and Prevention Mechanism in Industry 4.0 Landscape for Secure SECS/GEM Communications. *Sustainability* **2022**, *14*, 15900. https://doi.org/10.3390/su142315900

Academic Editors: Ali Bidram, Amin Mahmoudi, Rahmatollah Khezri, Mostafa Shaaban and Solmaz Kahourzade

Received: 6 October 2022
Accepted: 25 November 2022
Published: 29 November 2022

1. Introduction

Industry 4.0 is the current transformation of conventional manufacturing and industrial operations using new smart machines. Large-scale Internet of Things (IoT) and machine-to-machine (M2M) communications are coupled for better automation, improved networking and self-monitoring, and intelligent computer manufacturing that can assess and diagnose issues [1–3].

Recently, the volume of IoT traffic has sharply increased and is predicted to continue to rise in the coming years. IoT devices are frequently the targets of cyberattacks with serious repercussions due to their vulnerability. Because of this, strong tools are required to ensure an adequate level of security in IoT networks. For such a complicated task in Industry 4.0, machine learning and deep learning techniques show promise to perform well [4,5].

SEMI SECS/GEM plays an important role in Industry 4.0. The photo-voltaic, semiconductor, solar cell manufacturing, surface mount technology (SMT), electrical construction, and device adopt SECS/GEM standards in their equipment [6–8]. With the introduction of SECS/GEM capability, it is quickly transforming its operations into a smart factory with

M2M communication, automation, and real-time data acquisition for data monitoring, control, and analytics. The SECS/GEM is an industry protocol that has been commonly used for decades in nearly all industrial sectors [9–11].

The Semiconductor Devices Communication Protocol is referred to as SECS, as shown in Figure 1. The term "GEM" refers to the SEMI standard E30, which represents a general model for equipment operation and communication by using a subset of the message types established in the SEMI standard E5. In this era of Industry 4.0, TCP/IP networking (SEMI Standards E37 and E37.1) has become the most general use in SECS/GEM interfaces [12,13].

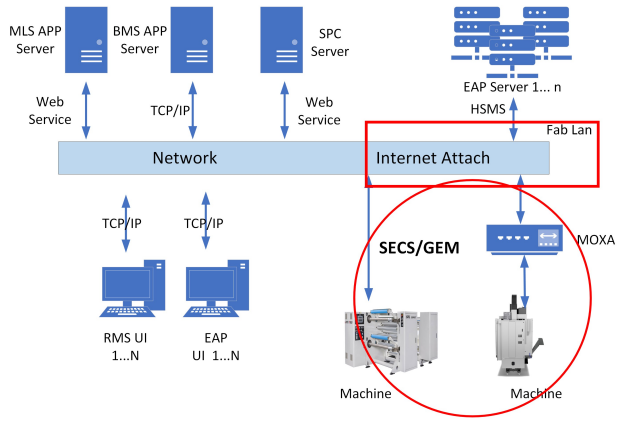

Figure 1. SECS/GEM SECI/HSMS connection.

A potential cyberthreat to SECGEM is anticipated. Once the computer system's protection is breached, the attacker can perform a cyberattack within an hour—simultaneous attacks from different locations are enabled by the Internet's lever of rising power. When an attacker accesses the SECS/GEM system and launches control acts that can inflict catastrophic damage, the most important effect of an attack occurs whether or not an attacker has access to it. Any successful attack could seriously impede the economy and the climate or even contribute to the loss of human life.

Today, cyberattacks on businesses, governments, and people are rising. The manufacturer/semiconductor industry is not safe from these cyberattack threats, although it is difficult to calculate the figures or success rates. Over time, cyberattacks have escalated. In a recent survey, McAfee saw an average of 375 new attacks per minute in the first quarter of 2020. Referring to one of the software engineers from Intel, the market demand for semiconductor equipment is USD 60 billion per year. We should consider cybersecurity threats to identify risk-based measures to protect equipment from them.

Although the semiconductor industry has laid out a clear argument for introducing cybersecurity standards, some concerns and issues remain. The introduction of cybersecurity requirements is not as simple as it sounds. In the field of security, there are various ways of breaching the network. If the goal of the intruder is to close down contact between the parties or to steal information from the unsuspected user, the protection mechanism in place must be able to protect itself against it. The only way to secure information that stops many of these threats from being possible is to encrypt all data sent from one location to another. However, what happens when the attacker does not need to know what the message says? This is the principle behind the replay attack [14].

SECS/GM HSMS does not identify security as there is no identification of the connecting entities; no registration or certificate is necessary for the connection. Data do not have any standard encryption algorithms. In the past, SECS/GEM with TCP/IP communication did not seem to have a problem because factory networks were disconnected from the external internet connection. However, in this era, factory networks are slowly opening to the outside world. As Ethernet and the TCP/IP protocol stack are becoming a central part of plant and factory networks, the result is that linking those networks to more comprehensive

corporate structures is becoming more relaxed and more popular. The major contributions are listed below:

- This paper proves a replay attack that was conducted against standard SECS/GEM operations, which was required to be considered before deploying.
- The proposed mechanism uses timestamp technology to detect replay attacks once the recipient verifier checks the freshness of the timestamp.
- The proposed detection mechanism has the ability to prevent a replay attack in SECS/GEM communication.
- Security analysis shows the replay-attack analysis by changing the contents of System-Bytes.
- This paper uses a Testbed environment to implement and evaluate the proposed detection mechanism.

The rest of this work is categorized as follows. Section 2 introduces some literature review works. Section 3 provides the background of this paper in detail. Section 4.1 introduces the proposed replay-attack detection and prevention mechanism. Section 5 analysis the security and detection mechanism of replay attacks. Section 6 shows the experimental results of our proposal. Section 7 shows the conclusion of this paper.

2. Literature Review

This section reviews several research mechanisms proposed in IoT for the Industry 4.0 landscape. There are a huge number of mechanism protocols based on industrial IoT, such as Message Queuing Telemetry Transport (MQTT), Data Distribution Service (DDS), Open Platform Communications–Unified Architecture (OPC UA), Constrained Application Protocol (CoAP), SECS/GEM, and many more. These communication protocols are reviewed as follows.

2.1. MQTT-Based Mechanisms

Due to its openness environment, small footprint, and efficient publish/subscribe standard, MQTT has become the de facto standard protocol for IoT-based M2M communication. The security mechanism has captured the attention of both academics and business leaders [15]. Since MQTT's mode of operation is the root cause of security problems, it is especially susceptible to cyberattacks [16]. Additionally, many security solutions have been proposed to address the MQTT's security flaws [17–19].

2.2. DDS-Based Mechanisms

Like MQTT, DDS [20] is a publish/subscribe-based M2M protocol that aims to improve the efficiency of data transmission in real-time systems. DDS is built to operate on both TCP and UDP, giving users a choice between Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS) as the underlying security algorithm. For this reason, the Object Management Group (OMG) proposed the DDS security specification, which details a powerful security framework thought to be best suited for Internet of Things (IoT) devices. However, these algorithms are computationally intensive and, therefore, not suitable for devices with limited resources.

2.3. OPC UA-Based Mechanisms

The OPC Foundation is a group of companies that works together to set and uphold standards for the open connectivity of manufacturing equipment and software [21]. Although OPC UA has built-in security features, it still requires stringent security configurations to work properly, preventing attackers from gaining access to sensitive data through port theft, wireless eavesdropping, and other means [22,23].

2.4. CoAP-Based Mechanisms

CoAP [24], as defined by RFC 7252 [25], is another software framework optimized for low-power networks and devices. The server may send a verification query to ensure

the client machine's 'hello' message came from a legitimate location. Nonetheless, this additional safeguard protects against DoS attacks.

2.5. SECS/GEM-Based Mechanisms

Contrary to the other protocols we have covered, SECS/GEM lacks any sort of in-built security measures [26,27]. SECS/messaging GEM's protocol—the High-Speed SECS Message Services (HSMS)—runs atop TCP, which itself defines no security mechanism. The connecting entity is not validated, and no credentials or authorization are needed. There is no end-to-end encryption during the transfer of the payload, which means it is sent as plaintext. The information is encapsulated using a binary encoding process, making the message unintelligible to a human reader. However, the message can be decoded and the data extracted by anyone familiar with binary encoding and SECS/GEM.

Authentication of data in IIoT systems was proposed by Karati et al. [28] using a certificate-less signature (CLS) scheme based on bilinear pairing. In this scheme, the signer must perform two exponentiations before generating a signature. However, in order for the verifier to validate a signature, two exponentiations and a pairing computation are needed. For a lightweight authentication mechanism, K. Mahmood et al. [29] proposed a hybrid Diffie–Hellman approach using AES and RSA to generate session keys. Mutual authentication is provided by the scheme, which helps to safeguard against replay and Man-in-the-Middle (MITM) attacks and ensures the confidentiality of transmitted messages. To ensure the security of the IoT ecosystem, Mumtaz et al. [30] developed an authentication mechanism utilizing state-of-the-art industry standards and RSA public-key encryption algorithms. A multi-key-based mutual authentication mechanism was introduced by T. Shah et al. [31]. Secure vaults, which are collections of keys of the same size, are used to store a shared secret between the IoT server and the IoT device in this method. As a means of shielding IIoT networks from various forms of cyberattacks, such as denial-of-service (DoS) attacks, router impersonation attacks, and smart-sensor traceability attacks, S. F. Aghili et al. [32] investigated the security flaws in the current M2M authentication protocols proposed for these networks. In [33], E. Lara et al. proposed an authentication protocol for IIoT networks that takes into account the limitations imposed by the ubiquitous nature of IoT devices. Authentication and message integrity problems with heterogeneous systems in manufacturing were addressed by KK. Kolluru et al. in [34].

Recently, Laghari et al. [35] investigated the main operations of communications-based SECS/GEM and how potential adversaries could tamper with these operations to gain malicious or illegal access. According to their experiments, SECS/GEM communications are vulnerable to numerous attacks. Jaisan et al. [36] presented AES-GCM encryption to propose a security mechanism for SECS/GEM communication. Then, Laghari et al. [26] proposed a security mechanism based on the digital signature that satisfies the security requirements.

Authentication mechanisms that rely on digital-signature-based algorithms are more difficult to implement because they require Certification Authorities and multiple-key exchange mechanisms. More bandwidth and processing time will be needed to complete the processes or operations if the underlying mechanisms are more complex. The abovementioned mechanisms have been shown to introduce vulnerabilities and have a compromised defense mechanism in studies [32,37]. In other words, attackers will be able to take advantage of vulnerabilities in preexisting defenses by adapting the security mechanism presented above and launching attacks such as replay attacks on SECS/GEM communications. Damage to reputation, theft of sensitive information, and interruption of network connectivity are all possible outcomes. Therefore, this paper not only detects replay attacks but also prevents them from accessing SECS/GEM communication. The proposed mechanism of this paper will use timestamp technology through the sender, which sends a message with the current timestamp, and the recipient verifier, which checks the freshness of the received timestamp. Therefore, the proposed detection mechanism will be able to detect and prevent a replay attack in SECS/GEM communication.

3. Background

3.1. SECS/GEM Standard

Over 2000 industries from all over the world are members of Semiconductor Equipment and Material International (SEMI). It provides the goods, services, and machinery that manufacturers demand. It releases several standards—such as E4, E5, E30, and E37—that coordinate information exchange of host and machine components, as shown in Figure 2. SECS/GEM stands for Semiconductor Equipment Communication Standard/Generic Model for Communications and Control of Manufacturing Equipment and is the umbrella term for the SEMI communication standards listed as follows.

- E4 (SEMI Equipment Communications Standard-I (SECS-I), 1978): A way to exchange data over an RS-232 connection, which can be used to connect a wide variety of devices to a host. It is effective at the physical layer.
- E5 (SEMI Equipment Communications Standard-II (SECS-II), 1982): Facilitates the transmission of data between devices and hosts in the form of streams and function messages following a standard protocol.
- E30 (Generic Equipment Model (GEM), 1992): Aids in identifying SECS-II message usage and in monitoring device behavior during communication protocol with the host.
- E37.1 (High-Speed SECS Message Service Session Single (HSMS-SS), 1994): Within the Transmission Control Protocol/Internet Protocol network (TCP/IP), the protocol outlined in this document controls the flow of data between individual devices and the host computer.
- E37.2 (High-Speed SECS Message Service- Global Session (HSMS-GS)): The Global equivalent to E37.1 with the added ability to manage multiple sessions while using the same sophisticated hardware.

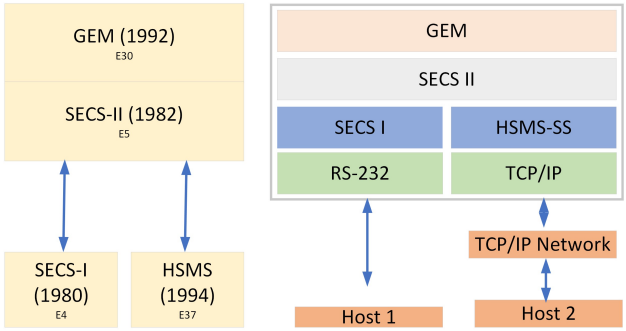

Figure 2. SECS standard.

The SECS/GEM is a widely used protocol in the manufacturing sector and has been so for many years [38]. From Intel and Samsung to Broadcom and UMC, IBM and Qualcomm, SK Hynix and Micron, NXP and Toshiba, TXN and Toshiba, and so on, the SEMI SECS/GEM has been the lifeblood of the semiconductor industry for years [39].

3.2. Replay Attack

Replay attacks, often referred to as playback attacks, are a sort of network assault in which honest data transfer has been repeated or purposefully delayed. The perpetrator of this assault could be the one who sent the original data, or an adversary who intercepted and retransmitted it, probably as part of an IP replacement packet spoofing attempt. This technique is regarded as one of the more basic man-in-the-middle attacks. "One way to characterize such an attack is": "an attack on a security protocol using a replay of messages in the intended (or initial and expected) sense from a separate context, thereby fooling the honest participant(s) into believing that the protocol run was completed" [40].

Figure 3 is a situation where Ann is trying to prove her identity to Ban via a post. Ban asks for her password to prove herself, to which Ann provides the password; meanwhile, El

sniffs the conversation and keeps the password. El (posing as Ann) interacts with Ban [41]. A replay attack can mimic the message after eavesdropping on the news from the network and replay it to the host and equipment. At the same time, the host and equipment do not seem aware of legitimate data.

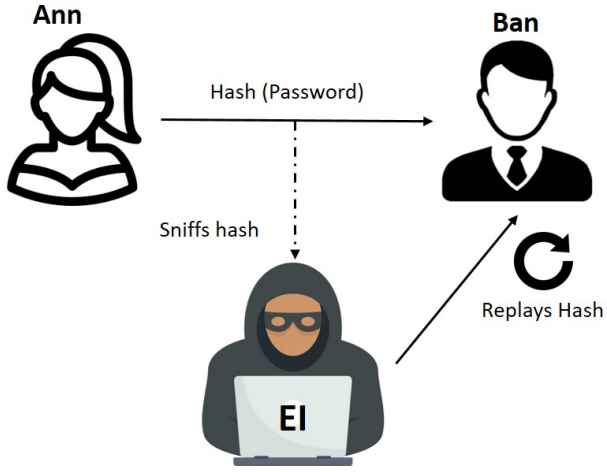

Figure 3. Illustration of a replay attack.

3.3. Scope of Study

The main focus of this paper is to prove that SECS/GEM communication is vulnerable to replay attacks. The replay attack will be demonstrated as the cyberattack in this paper and part of the main contribution to proving that SECS/GEM communication is vulnerable. This is due to this cyberattack being a major contributor that causes equipment to be stopped or data to be disclosed. For detection, the mechanism will only be limited to detecting the replay attack to show the ability of SECS/GEM's security to prevent the attack. Other types of cyberattack and detection will not be described in this paper.

3.4. Objective of Research

At the end of this section, the objectives achieved are as follows:

- Prove that SECS/GEM communication is vulnerable to cyberattack.
- Design a detection mechanism to detect replay attack.

3.5. Problem Statement

Since the SECS/GEM interface mostly uses TCP/IP networking (SEMI Standards E37 and E37.1) and HSMS does not have any security, SECS/GEM is vulnerable to cyberattack. As mostly semiconductor industries such as Intel, AMD, TSMC, TI, Infineon, etc. have SECS/GEM in their machines [17], it is crucial to secure SECS/GEM communications as one machine or device affected by a cyberattack may become harmful and trigger damage or loss of production, as shown in an example occurring on TSMC, X-Fab, and Tower Semiconductors.

The SECS/GEM protocol does not have any security features to create a system link with other network components that enable the opponent to launch an attack (core idea) and disrupt credibility and productivity. Besides integrity checking, both entities exchange binary-encoded SECS/GEM messages.

Attention to SECS/GEM security is quite rare as there are other types of security that are able to enhance network security. However, they are not 100% guaranteed security protection. Once a cyberattack breaks through these securities, it will be able to attack SECS/GEM communication in industry networks. We cannot lower our guard on SECS/GEM security even though there has been no report showing an attack on SECS/GEM. As SECS/GEM has a weakness itself in security, it is just a matter of time before a cyberattack is launched. If a cyberattack really happens on SECS/GEM, it will

be too late, and the impact will be huge, as described in the cyberattack cases previously. Compared with the previous cases, currently, most companies have moved to Industry 4.0. This is very worrying, as current research on SECS/GEM security and the development of security has still not been explored. This needs to be explored so that when an attack is able to penetrate primary network security, it still has secondary protection on itself.

4. Proposed Replay-Attack Detection and Prevention Mechanism

4.1. Plan and Implementation Flow

As shown in Figure 4, this paper follows 4 project steps, namely, Analysis, Design, Implementation and Testing, and Evaluation.

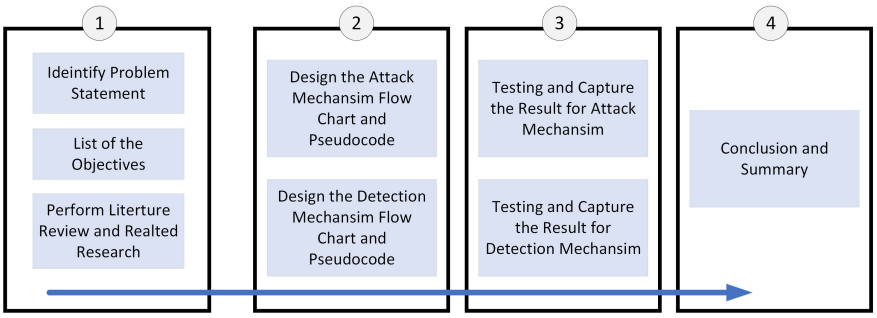

Figure 4. Details of project implementation flow chart.

- Requirement Analysis Stage: This stage analyzes the SECS/GEM SEC-II message and HSMS message, understanding the message encoding between these two standards. Besides, it analyzes in which situations the replay attack can occur.
- Design Stage: Based on requirement analysis, simulation concepts can be designed to simulate the replay attack in SECS/GEM communication to prove objective number 1—that is, SECS/GEM is vulnerable—and design a detection mechanism for objective number 2.
- Implementation and Test Stage: In this stage, the first implementation is the setup SECS/GEM simulator. Secondly, it develops an attack pseudocode. Lastly, a detection mechanism algorithm is produced.
- Evaluation Stage: This is the most critical part of the study to demonstrate the attack on SECS/GEM and how the detection mechanism detects the attack. The result of the attack and detection will be shown at this stage.

4.2. Design Concepts

SEGS/GEM E37 HSMS provides a binary code message consisting of 4 Byte Message length + 10 Bytes Header + Data Message (0–8 Mbytes). For attack mechanism simulation, byte 9 consisting of SType is used to identify whether the message is in linktest.reg (SType = 5) or linktest.res (SType = 6). Code is written to analyze these areas and trigger an attack message to have SType = 9 to send to entities. SType = 9 is known as separate.req, which terminates the connections.

For attack mechanism simulation, byte 9 consisting of SType is used to identify whether the message is in linktest.reg (SType = 5) or linktest.res (SType = 6). Code is written to analyze these areas and trigger an attack message for SType = 9 to send to entities. SType = 9 is known as separate.req, which terminates the connections.

The 24 h communication pattern between a host and equipment in a production environment is depicted in Figure 5. SECS/GEM messages are paired with a primary message, which is a request message, and a secondary message, which is a response message. Over 14 thousand data are transferred in S6F11 and S6F12, and all can be exposed to the attacker.

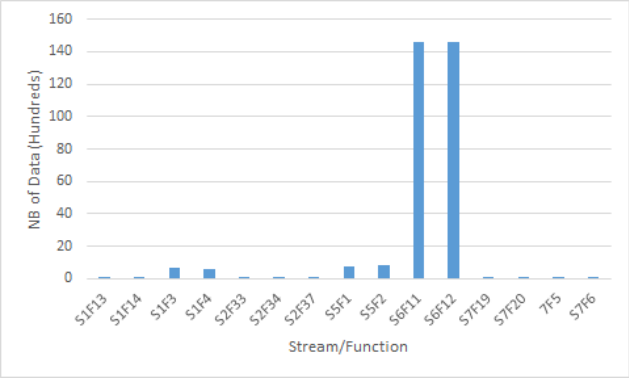

Figure 5. SECS/GEM message data transfer in one industry equipment.

According to the SECS/GEM standard, equipment can only connect to one host device at once. There is just one active point-to-point communication link between the host and production equipment. Thus, if the attacker manages to break the host connection and establish a relationship with equipment, the host will be unable to reconnect to the equipment unless the attack terminates the connection. Data will transfer to the attacker once connections are established with equipment. The attacker can eventually perform a replay attack.

5. Replay-Attack Analysis

The equipment will return to a NOT-CONNECTED state after a failed TCP connection. On the specified port address, it will begin to wait for a new incoming connection request (i.e., usually port 5000). At this point, the attacker would connect to the equipment and assume control of it as the host. These data may be exchanged between the attacker host and equipment. The attacker can collect significant stream function data, as shown in Figure 6.

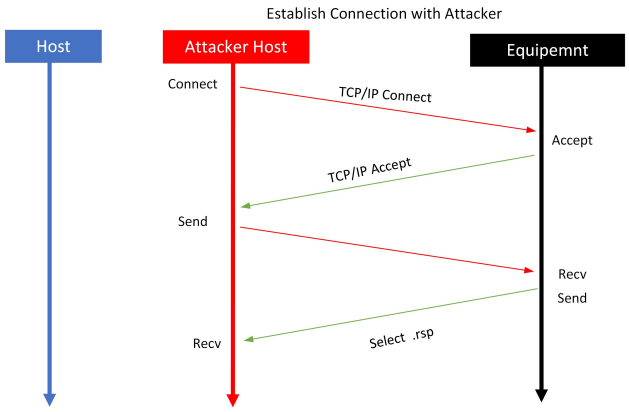

Figure 6. Attacker establishing a connection with equipment.

The attack also can stand behind and listen to port 5000 on all the HSMS messages being transferred. The attacker can mimic the message and replay the message by sending it back to the host or equipment. These, eventually, will confuse the host and equipment. This attack is a replay attack.

In a production environment, as in Figure 7, the host can connect to various equipment types that are equipped with SECS/GEM. An attack is able to launch separate.reg to all the equipment and terminate their primary host communication, and establish attack host connections to all the equipment. The machine may stop and behave abnormally. The impact and damage are huge when an attack involves the entire company's SECS/GEM equipment. Data collection from equipment to host is also doubtable due to the replay

message by the attacker to host. Data such as process control data, machine alarm events, or machine OEE may be polluted by replay attack and unable to be used.

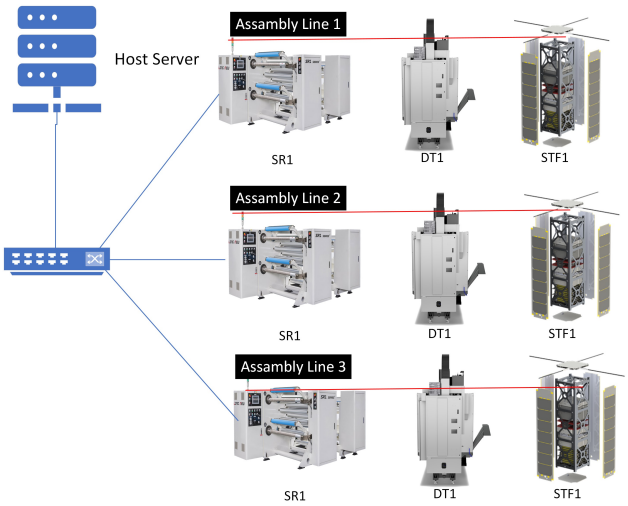

Figure 7. Production network environment—host with equipment.

It will be regarded as the control message, in addition to these. An attacker who is familiar with the HSMS protocol can sniff a message and respond with a different payload in order to harm or interrupt production-line equipment. The SystemBytes sequence number field in HSMS's header is monotonically incremented for each transaction. A transaction is a set of request and response messages sent from the host to the equipment or the other way around. Each transaction has its own sequence number; however, each succeeding transaction will have a number that corresponds to the one before it. Because of this predictability, attackers can launch a replay attack by changing the contents of SystemBytes. Replay attacks would be detrimental to general communication; it may tear down equipment communication in the production line or adversely impact equipment behavior, which can cause product quality issues and accidents. Figure 8 depicts the replay-attack scenario.

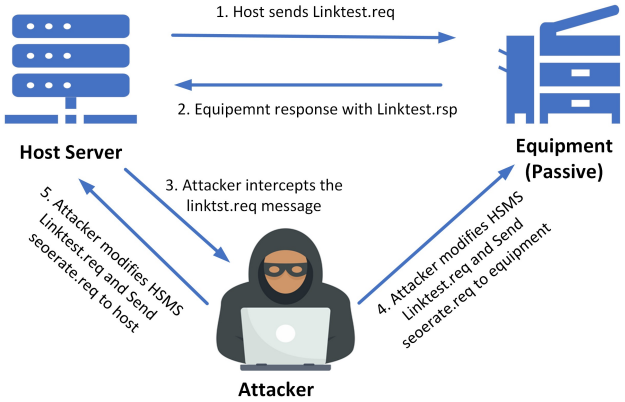

Figure 8. Attacker while capturing, intercepting, and attacking SECS/GEM communication.

6. Experiment Result

6.1. Hardware Simulation Concept

In this section, numerous attacks are launched against various SECS/GEM communication processes occurring between the equipment and host. The hardware for the testing attack simulation is shown in Table 1. Between a host and equipment in the assembly line, SECS/GEM communication first begins.

Table 1. Simulation hardware and spe ification.

Device	Specifications	OS	Software
Host	CPU: Intel® Ci7-9750H @ 2.6Ghz × 8 RAM: 24 GB	Win10	PyChram
Equipment	CPU: Intel® Ci7-3770M @ 3.4Ghz × 8 RAM: 8 GB	Win10	PyChram
Attacker	CPU: Intel® Ci3-330M @ 2.13Ghz × 2 RAM: 8 GB	Ubuntu 2020.3	PyChram Wireshark
Switch	Cisco Catalyst 2960 Fast Ethernet		

6.2. Testbed Environment

The trials were performed to watch how SECS/GEM acted under various attacks. Python is used to implement SECS/GEM host and equipment, which simulates the behavior and capabilities of SECS/GEM entities. The required code-level adjustments have been made in order to replicate the host-side emulator's attack behaviors. It is crucial to eavesdrop and collect the packets while they are in transit and then modify them to execute assaults such as replay attacks in order to successfully conduct cyberattacks on SECS/GEM communications.

As in Figure 9, we will simulate an attack on the assembly production line for equipment STF1. The production line arrangement is almost the same in the work area of my company's production floor. All these machines SR1, DF1, DEK1, and STF1 are equipped with SECS/GEM features to transfer data to PCS (process control system), RMS (Recipes Management System), MMS (Material Management System), TMS (Tool Management System), OTA, and CAMSTAR.

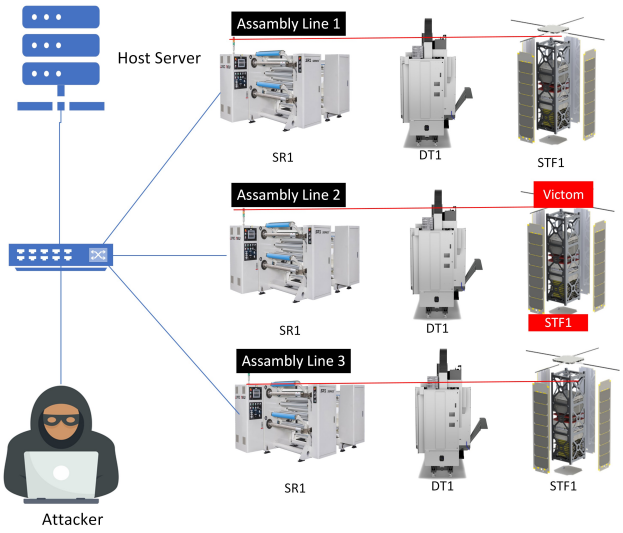

Figure 9. Testbed environment—an attack scenario.

6.3. Implementation

Figure 10 is the attack flow chart. Attack will start to sniff SEC-II messages. If the header message contains Byte 9 (SType) = 6, it will change to create a header message with SType = 9 that represents separate.reg. If SType is not equal to 6, it will keep listening to the SEC-II message. The next flow after the Set SType = 9 will be changing the source and destination IP. It then will send the separation.reg to equipment. Once the equipment is terminated, it will connect with equipment until the connection is successful.

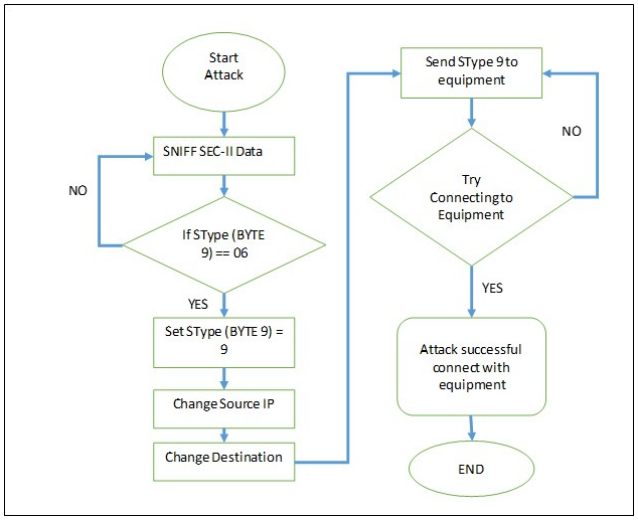

Figure 10. Attack flow chart.

Scapy Tool, a Python-based program, is used to capture, alter, and send forwarded packets to the intended recipient. For each procedure, the attacks were carried out thirty times. Equation (1) measures the ability of the SECS/GEM processes to prevent the attacks carried out on SECS/GEM communication [42]:

$$ATTACK_{SR} = 1 - \frac{F}{N} \quad (1)$$

$ATTACK_{SR}$ represents the specific attack's success rate, N represents the number of times forged messages are injected, and F signifies that the particular attack type failed several times. The algorithm found that if $ATTACK_{SR}$ is 1, then the attack is successful. If $ATTACK_{SR}$ is not 1, an attack is specified on the SECS/GEM's as not a success, such as in control messages and data messages. The number of times testing and Formula (1) are a reference to a research study called "Denial of Service Attack on Neighbor Discovery Protocol Processes in the Network of IPv6 Link- Local" [42]. For this paper, we increase our number to 30 times to prove that the replay-attack success rate is still high even after 30 times of testing.

Table 2 shows that the success rate for replay attack is high, and there is no failure even in 30 times testing. In replay attack, the control message is also able to listen without any failure, and the duplicate data are able to be injected without any failure. Since there is no failure, $ATTACK_{SR}$ = 1. Attack failures are manually calculated. The high assault success rate demonstrated how thoroughly exposed the SECS/GEM communication is to these cyberattacks.

Table 2. Experiment result.

Message Type	Experiment Count (N)	Attack Failure (F)	$ATTACK_{SR}$
Control	30	0	1
Data	30	0	1

6.4. Detect Mechanism

For detection of replay attack, we explored implementing a timestamp. Figure 11 is a Detection flow consisting of Send and Receive flow. This flow needs to be put into host and equipment. The concept is that the sender will append a timestamp while the receiver will append an extra timestamp. In the sender flow chart, before sending any HSMS package, it will save the current timestamp and inject it into the data message. The receiving entity will receive these packages with a timestamp in the message data. The receiving entity

will add an extra timestamp and compare whether the received timestamp is less than or equal to the last time it recorded. If the timestamp is less than or equal to the previous timestamp, it will trigger "Duplicate package" and drop the package; otherwise, it will accept the package and proceed as usual.

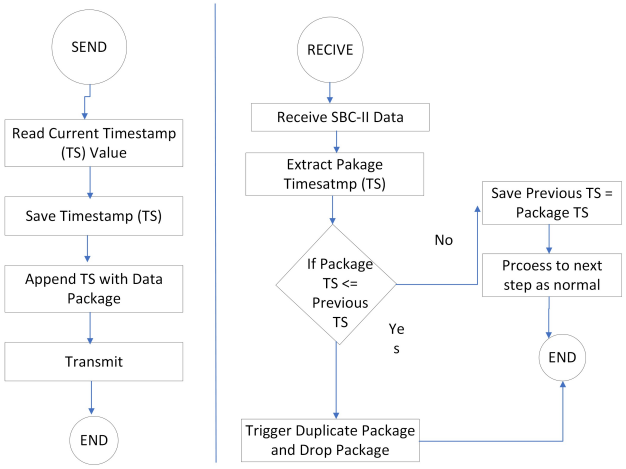

Figure 11. Detection flow chart.

Figure 12 presents the pseudocode that explains the timestamp that was saved and appended to the data package. *self.last_ts* is a timestamp, and data represents data combined with the timestamp.

```
def send_packet(self, packet):
    """
    Send the ASCII coded packet to the remote host.

    :param packet: encoded data to be transmitted
    :type packet: string / byte array
    """
    # encode the packet
    data = packet.encode()

    # append timestamp to packet
    self.last_ts = int(time.time())
    data += struct.pack(">L", self.last_ts)

    if not self.send_packet_as_blocks(data):
        return False

    # randomly call the send duplicate function
    # if bool(random.getrandbits(1)):
        # self.send_duplicate_packet(packet)

    return True
```

Figure 12. Pseudocode for Save and Append timestamp.

Using the pseudocode as in Figure 13, the Python-based Scapy tool is run to hijack and detect any suspicious package intercepted and injected into the message. The result shows that the suspected message was duplicated and the packet was dropped with a triggering Duplicate detected.

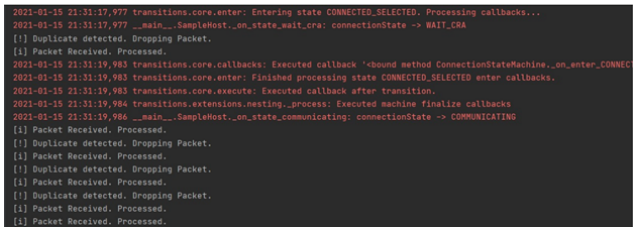

Figure 13. Result of detection on duplicate package.

As a result, this section shows the implementation to perform the attack and detection was performed as expected. The result of the attack and detection concludes that SECS/GEM is vulnerable to replay attack and the need for security prevention to detect the attack.

6.5. Security Comparison

This subsection compares the security of our proposed mechanism with eight recently proposed mechanisms [28–34] for secure SECS/GEM communications. Let SC-1, SC-2, SC-3, and SC-4 denote threats to cybersecurity, preventing replay attacks, detecting replay attacks, and efficient mechanism, respectively. Table 3 compares the different mechanisms in terms of security.

According to Table 3, none of the nine recently proposed mechanisms can achieve all four security features (SC-1, SC-2, SC-3, and SC-4). In contrast, our mechanism could achieve all four security features for secure SECS/GEM communications.

Note that the main aim of an efficient mechanism is to reduce the performance overhead during prevention and detection. Unlike Laghari et al. [26], K. Mahmood et al. [29], and Laghari et al. [35] using encryption algorithms, the proposed mechanism is more effective at securing replay attacks by using timestamps without any extra algorithms.

Table 3. Security comparison of past mechanisms and our mechanism.

Mechanisms	SC-1	SC-2	SC-3	SC-4
Laghari et al. [26]	✓	✓	✓	✗
Karati et al. [28]	✗	✗	✗	✗
K. Mahmood et al. [29]	✓	✓	✓	✗
Mumtaz et al. [30]	✗	✗	✗	✗
T. Shah et al. [31]	✗	✗	✗	✗
S. F. Aghili et al. [32]	✗	✗	✗	✗
Lara et al. [33]	✗	✗	✗	✗
KK. Kolluru et al. [34]	✗	✗	✗	✗
Laghari et al. [35]	✓	✓	✓	✗
Our mechanisms	✓	✓	✓	✓

7. Conclusions

SECS/GEM is mostly used in semiconductor industries and it has been highly used since its introduction. SECS/GEM can quickly turn their activities into a smart factory for data monitoring, regulation, and analytics through M2M connectivity, automation, and real-time data acquisition. As binary-encoded communications are exchanged unencrypted without any authentication or encryption, SECS/GEM does not provide any security measure. The results of the experiment demonstrate that cyberattacks can target the SECS/GEM message. Therefore, a thorough security structure is needed to guard against these cyberattacks. For secure and dependable communication in the future, SECS/GEM security should include authentication, confidentiality, and integrity. The challenges discussed in this paper are the topic of our next step, which offer a comprehensive security architecture to safeguard SECS/GEM communication against online attacks.

In future work, we will extend this work by proposing a strong mechanism to resist security attacks such as DoS, forgery, modify, and man-in-the-middle attacks. Additionally, the overhead efficiency in terms of communications and computational costs will be considered as well.

Author Contributions: Conceptualization, writing—review and editing, M.A.A.-S.; writing—original draft preparation, investigation, supervision, S.M.; funding acquisition, software, visualization, S.A.L.; methodology, funding acquisition, resources, A.J.; project administration, funding acquisition, software. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by Vice Chancellor Initiative Allocation, Universiti Sains Malaysia, grant number 311/PNAV/4119101.

Institutional Review Board Statement: Not Applicable.

Informed Consent Statement: Not Applicable.

Data Availability Statement: Not Applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Prasad, R.; Rohokale, V. Internet of Things (IoT) and machine to machine (M2M) communication. In *Cyber Security: The lifeline of Information and Communication Technology*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 125–141.
2. Al-Shareeda, M.A.; Manickam, S.; Saare, M.A.; Arjuman, N.C. Proposed security mechanism for preventing fake router advertisement attack in IPv6 link-local network. *Indones. J. Electr. Eng. Comput. Sci.* **2023**, *29*, 518–526. [CrossRef]
3. Mazhar, M.S.; Saleem, Y.; Almogren, A.; Arshad, J.; Jaffery, M.H.; Rehman, A.U.; Shafiq, M.; Hamam, H. Forensic Analysis on Internet of Things (IoT) Device Using Machine-to-Machine (M2M) Framework. *Electronics* **2022**, *11*, 1126. [CrossRef]
4. Nascita, A.; Cerasuolo, F.; Di Monda, D.; Garcia, J.T.A.; Montieri, A.; Pescape, A. Machine and Deep Learning Approaches for IoT Attack Classification. In Proceedings of the 2022 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), New York, NY, USA, 2–5 May 2022; pp. 2–5.
5. Al-Shareeda, M.A.; Manickam, S.; Mohammed, B.A.; Al-Mekhlafi, Z.G.; Qtaish, A.; Alzahrani, A.J.; Alshammari, G.; Sallam, A.A.; Almekhlafi, K. Provably secure with efficient data sharing scheme for fifth-generation (5G)-enabled vehicular networks without road-side unit (RSU). *Sustainability* **2022**, *14*, 9961. [CrossRef]
6. Sun, Y.; Peng, X.; Zhu, M.; Jiao, D.; Fuyang, S. Design and Implementation of OPC UA Server Based on SECS/GEM Protocol. In Proceedings of the 2021 International Conference on Wireless Communications and Smart Grid (ICWCSG), Hangzhou, China, 13–15 August 2021; pp. 38–42.
7. Al-Shareeda, M.A.; Manickam, S.; Mohammed, B.A.; Al-Mekhlafi, Z.G.; Qtaish, A.; Alzahrani, A.J.; Alshammari, G.; Sallam, A.A.; Almekhlafi, K. Cm-cppa: Chaotic map-based conditional privacy-preserving authentication scheme in 5g-enabled vehicular networks. *Sensors* **2022**, *22*, 5026. [CrossRef]
8. Li, F.; Qiao, Y.; Zhao, H.; He, S.; Chen, X. Analysis of SEMI EDA standards for semiconductor equipment data acquisition. In Proceedings of the Third International Conference on Computer Communication and Network Security (CCNS 2022), Hohhot, China, 28 October 2022; Volume 12453, pp. 13–19.
9. Terng, E.F.; Yeoh, S.C.; Tong, K.C.; Yeo, K.S. Data analysis on SMT reflow oven with SECS/GEM communication protocol. In Proceedings of the 2020 IEEE 10th Symposium on Computer Applications & Industrial Electronics (ISCAIE), Penang, Malaysia, 18–19 April 2020; pp. 118–124.
10. Al-Shareeda, M.A.; Manickam, S.; Mohammed, B.A.; Al-Mekhlafi, Z.G.; Qtaish, A.; Alzahrani, A.J.; Alshammari, G.; Sallam, A.A.; Almekhlafi, K. Chebyshev polynomial-based scheme for resisting side-channel attacks in 5g-enabled vehicular networks. *Appl. Sci.* **2022**, *12*, 5939. [CrossRef]
11. Zhu, M.; Peng, X.; Sun, Y.; Fuyang, S.; Jiao, D. Simulation study of semiconductor communication protocol SECS/GEM. In Proceedings of the 2021 International Conference on Wireless Communications and Smart Grid (ICWCSG), Hangzhou, China, 13–15 August 2021; pp. 148–152.
12. Yaw, S.C. Development of SECS/GEM LabVIEW Toolkit for GEM-compliant Semiconductor Equipment. Ph.D Thesis, Tunku Abdul Rahman University College, Pulau Pinang, Malaysia, 2019.
13. Al-Shareeda, M.A.; Manickam, S. Man-In-The-Middle Attacks in Mobile Ad Hoc Networks (MANETs): Analysis and Evaluation. *Symmetry* **2022**, *14*, 1543. [CrossRef]
14. Al-shareeda, M.A.; Anbar, M.; Hasbullah, I.H.; Manickam, S.; Abdullah, N.; Hamdi, M.M. Review of prevention schemes for replay attack in vehicular ad hoc networks (vanets). In Proceedings of the 2020 IEEE 3rd International Conference on Information Communication and Signal Processing (ICICSP), Shanghai, China, 12–15 September 2020; pp. 394–398.
15. Roldán-Gómez, J.; Carrillo-Mondéjar, J.; Castelo Gómez, J.M.; Ruiz-Villafranca, S. Security Analysis of the MQTT-SN Protocol for the Internet of Things. *Appl. Sci.* **2022**, *12*, 10991. [CrossRef]
16. Hernández Ramos, S.; Villalba, M.T.; Lacuesta, R. Mqtt security: A novel fuzzing approach. *Wirel. Commun. Mob. Comput.* **2018**, *2018*, 8261746. [CrossRef]
17. Munshi, A. Improved MQTT Secure Transmission Flags in Smart Homes. *Sensors* **2022**, *22*, 2174. [CrossRef]
18. Patel, C.; Doshi, N. A novel MQTT security framework in generic IoT model. *Procedia Comput. Sci.* **2020**, *171*, 1399–1408. [CrossRef]

19. Rahman, A.; Roy, S.; Kaiser, M.S.; Islam, M.S. A lightweight multi-tier S-MQTT framework to secure communication between low-end IoT nodes. In Proceedings of the 2018 5th International Conference on Networking, Systems and Security (NSysS), Dhaka, Bangladesh, 18–20 December 2018; pp. 1–6.
20. Friesen, M.; Karthikeyan, G.; Heiss, S.; Wisniewski, L.; Trsek, H. A comparative evaluation of security mechanisms in DDS, TLS and DTLS. In *Kommunikation und Bildverarbeitung in der Automation*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 201–216.
21. Pu, C.; Ding, X.; Wang, P.; Xie, S.; Chen, J. Semantic Interconnection Scheme for Industrial Wireless Sensor Networks and Industrial Internet with OPC UA Pub/Sub. *Sensors* **2022**, *22*, 7762. [CrossRef]
22. Profanter, S.; Tekat, A.; Dorofeev, K.; Rickert, M.; Knoll, A. OPC UA versus ROS, DDS, and MQTT: Performance evaluation of industry 4.0 protocols. In Proceedings of the 2019 IEEE International Conference on Industrial Technology (ICIT), Melbourne, Australia, 13–15 February 2019; pp. 955–962.
23. Matischek, R.; Bara, B. Application study of hardware-based security for future industrial IoT. In Proceedings of the 2019 22nd Euromicro Conference on Digital System Design (DSD), Kallithea, Greece, 28–30 August 2019; pp. 246–252.
24. Iglesias-Urkia, M.; Orive, A.; Urbieta, A.; Casado-Mansilla, D. Analysis of CoAP implementations for industrial Internet of Things: A survey. *J. Ambient Intell. Humaniz. Comput.* **2019**, *10*, 2505–2518. [CrossRef]
25. Gong, X.; Feng, T. Lightweight Anonymous Authentication and Key Agreement Protocol Based on CoAP of Internet of Things. *Sensors* **2022**, *22*, 7191. [CrossRef]
26. Laghari, S.U.A.; Manickam, S.; Al-Ani, A.K.; Rehman, S.U.; Karuppayah, S. SECS/GEMsec: A Mechanism for Detection and Prevention of Cyber-Attacks on SECS/GEM Communications in Industry 4.0 Landscape. *IEEE Access* **2021**, *9*, 154380–154394. [CrossRef]
27. Laghari, S.A.; Manickam, S.; Karuppayah, S. A review on SECS/GEM: A machine-to-machine (M2M) communication protocol for industry 4.0. *Int. J. Electr. Electron. Eng. Telecommun.* **2021**, *10*, 105–114. [CrossRef]
28. Karati, A.; Islam, S.H.; Karuppiah, M. Provably secure and lightweight certificateless signature scheme for IIoT environments. *IEEE Trans. Ind. Inform.* **2018**, *14*, 3701–3711. [CrossRef]
29. Mahmood, K.; Chaudhry, S.A.; Naqvi, H.; Shon, T.; Ahmad, H.F. A lightweight message authentication scheme for smart grid communications in power sector. *Comput. Electr. Eng.* **2016**, *52*, 114–124. [CrossRef]
30. Mumtaz, M.; Akram, J.; Ping, L. An RSA based authentication system for smart IoT environment. In Proceedings of the 2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), Zhangjiajie, China, 10–12 August 2019; pp. 758–765.
31. Shah, T.; Venkatesan, S. Authentication of IoT device and IoT server using secure vaults. In Proceedings of the 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), New York, NY, USA, 1–3 August 2018; pp. 819–824.
32. Aghili, S.F.; Mala, H. Breaking a lightweight M2M authentication protocol for communications in IIoT environment. *Cryptol. Eprint Arch.* **2018**, *19*, 891.
33. Lara, E.; Aguilar, L.; Sanchez, M.A.; García, J.A. Lightweight authentication protocol for M2M communications of resource-constrained devices in industrial Internet of Things. *Sensors* **2020**, *20*, 501. [CrossRef]
34. Kolluru, K.K.; Paniagua, C.; van Deventer, J.; Eliasson, J.; Delsing, J.; DeLong, R.J. An AAA solution for securing industrial IoT devices using next generation access control. In Proceedings of the 2018 IEEE Industrial Cyber-Physical Systems (ICPS), St. Petersburg, Russia, 15–18 May 2018; pp. 737–742.
35. Laghari, S.; Manickam, S.; Karuppayah, S.; Al-Ani, A.; Rehman, S.U. Cyberattacks and vociferous implications on SECS/GEM communications in industry 4.0 ecosystem. *Int. J. Adv. Comput. Sci. Appl.* **2021**, *12*. [CrossRef]
36. Jaisan, A.; Manickam, S.; Laghari, S.; Rehman, S.U.; Karuppayah, S. Secured SECS/GEM: A Security Mechanism for M2M Communication in Industry 4.0 Ecosystem. *Int. J. Adv. Comput. Sci. Appl.* **2021**, *12*. [CrossRef]
37. Yang, W.; Wang, S.; Huang, X.; Mu, Y. On the security of an efficient and robust certificateless signature scheme for IIoT environments. *IEEE Access* **2019**, *7*, 91074–91079. [CrossRef]
38. Stoop, F.; Ely, G.; Menna, R.; Charache, G.; Gittler, T.; Wegener, K. Smart factory equipment integration through standardised OPC UA communication with companion specifications and equipment specific information models. *Int. J. Mechatronics Manuf. Syst.* **2019**, *12*, 344–364. [CrossRef]
39. Rubow, B. SECS/GEM, SECS/GEM Features & Benefits Series, Figshare. 2017. Available online: https://www.cimetrix.com/blog/features-and-benefits-of-the-secs-gem-communication-standards (accessed on 11 October 2022).
40. Malladi, S. On Preventing Replay-attacks on Security Protocols, Figshare. 2020. Available online: https://en.wikipedia.org/wiki/Replay_attack# (accessed on 11 October 2022).
41. Schuba, C.L.; Krsul, I.V.; Kuhn, M.G.; Spafford, E.H.; Sundaram, A.; Zamboni, D. Analysis of a denial of service attack on TCP. In Proceedings of the Proceedings. 1997 IEEE Symposium on Security and Privacy (Cat. No. 97CB36097), Oakland, CA, USA, 4–7 May 1997; pp. 208–223.
42. Al-Ani, A.K.; Anbar, M.; Al-Ani, A.; Ibrahim, D.R. Match-prevention technique against denial-of-service attack on address resolution and duplicate address detection processes in IPv6 link-local network. *IEEE Access* **2020**, *8*, 27122–27138. [CrossRef]