Enhanced Inf-TESLA Protocol: A Continuous Connectivity and Low Overhead Authentication Protocol via IoT Devices

KHOULOUD ELEDLEBİ1, AHMED ADEL ALZUBAIDI2, CHAN YEOB YEUN1,2, (Senior Member, IEEE), ERNESTO DAMIANI1,2, (Senior Member, IEEE), VICTOR MATEU3,4, AND YOUSOF AL-HAMMADI2

1Center for Cyber-Physical Systems, Khalifa University, Abu Dhabi, United Arab Emirates
2Department of Electrical Engineering and Computer Science, Khalifa University, Abu Dhabi, United Arab Emirates
3Technology Innovation Institute, Abu Dhabi, United Arab Emirates
4Department of Mathematics, Universitat de Lleida, 25003 Lleida, Spain

Corresponding author: Chan Yeob Yeun (chan.yeun@ku.ac.ae)

This work was supported in part by the Center for Cyber-Physical Systems (C2PS), Khalifa University; and in part by the Technology Innovation Institute (TII) under Grant 8434000386-TII-ATM-2035-2020.

ABSTRACT Continuous and low-cost broadcast authentication is a fundamental security service for distributed sensor networks. This paper presents a novel development of a continuous and low-overhead broadcast authentication protocol named enhanced Infinite timed-efficient stream-loss tolerant authentication (enhanced Inf-TESLA) protocol, based on the Inf-TESLA protocol, whose continuous authentication is limited to the duration of its keychains. The enhanced Inf-TESLA protocol satisfies important security properties, including lower communication and computational overhead; a continuous generation of key-chains without the need to establish synchronization packets; scalability to a large network; and resistance to masquerading, modification, man-in-the-middle, and replay attacks. We also highlighted an unaddressed authentication issue in the last packets of the original TESLA protocol and proposed a corresponding solution. We performed a simulation analysis using JAVA and proved that, compared to the Inf-TESLA protocol, the enhanced Inf-TESLA protocol can continuously authenticate packets for the entire lifetime of the receiver. We also compared the enhanced Inf-TESLA protocol with the original TESLA protocol in terms of time complexity and critical authentication processes. The results revealed the superiority of the enhanced Inf-TESLA protocol over the original TESLA protocol in terms of the message authentication code (MAC) value generation time and packet authentication time, which we believe can significantly improve the lifetime and lower the energy expenditure of Internet of Things devices with limited power sources.

INDEX TERMS Continuous authentication, Internet of Things, low overhead, TESLA protocol, time complexity.

I. INTRODUCTION

The development of the Internet of Things (IoT) technology has enabled billions of devices around the world to be connected to the Internet to collect and share data, create a level of digital intelligence, and support real-time communication of data [1]. Majority of devices that contribute to the IoT are constrained devices that have access to user information and daily life changes, which makes them vulnerable to cybersecurity attacks. Counter actions include using IoT devices as entry points to access other parts of the network or as a bait to turn down the attacker’s system down. Constrained devices, such as sensors or smart devices, have limited CPU, memory, and power resources, which restricts the use of...
security protocols in protecting the privacy of their transferred data [1], [2].

The main challenges in securing broadcast communication are source and integrity authentication, verifying that the received data comes from a legitimate source and has not been altered en-route [3]. Furthermore, maintaining continuous authentication between the source and destination during its lifetime is another security and resource management challenge to avoid wasting energy while needlessly repeating the source verification process [4], [5]. Consequently, security protocols have begun to focus on important services to be protected while maintaining a reliable lifetime of the devices. The development of the TESLA (timed efficient stream loss-tolerant authentication) protocol in [6] and its variants introduced broadcast authentication protocols for constrained devices with low communication and computational overhead, relying on symmetric cryptographic functions (symmetric keychains and message authentication code (MAC) functions) to be exchanged for authentication of the sender and its data [7]. TESLA protocols also enable asymmetric properties by establishing a loose synchronization between the sender and the receiver [8] and with the disclosure delay available to protect the exchanged keys. However, the issue of continuous authentication between the sender and the receiver has not been addressed properly in any variant of the TESLA protocol.

It is claimed that a variant of the TESLA protocol named Inf-TESLA [9] provides continuous authentication by using two keychains in offset alignment, allowing one keychain to authenticate the data while the other is being generated. It only synchronizes the time at the beginning of the connection and relies on symmetric cryptographic functions to authenticate the packets. However, once the two keychains expire, the synchronization between both parties needs to be restarted, which wastes energy, memory space, and time while communicating with the same sender. Moreover, there is an unaddressed issue in authenticating the last packets in the variant TESLA protocols, which we revealed for the first time. These challenges motivated us to propose a continuous and low-overhead authentication protocol for IoT devices.

In this paper, we present a new variant of the Inf-TESLA protocol named enhanced Inf-TESLA protocol, which allows continuous communication and authentication between the sender and the receiver by regenerating new offset keychains during their communication time window without the need to establish a new synchronization. Our proposed protocol relies on the inclusion of an additional symmetric key, called the master key, to be exchanged between the sender and the receiver during their first synchronization, performing one-way hashing and symmetric encryption to protect the secret keys exchanged between the parties to generate new keychains. While the Inf-TESLA protocol tends to cut the connection between the sender and the receiver after the expiration of the first two keychains, the enhanced Inf-TESLA protocol can keep the communication channel between the sender and the receiver available for authenticating the broadcasting packets, starting from a single synchronization packet, and regenerating the keychains as long as the communication between the parties is not terminated. We also propose a solution for the authentication issue of the last packets in variant TESLA protocols, with no additional computational overhead. Therefore, we listed the contributions of our study as follows:

1. Developing an enhanced Inf-TESLA protocol to provide continuous authentication and low computational overhead to the receiver during the generation of the keychains, without re-establishing synchronization packets between the sender and the receiver.
2. Exposing the unaddressed issue of authenticating the last packets in the existing TESLA protocols and proposing a minimal computational overhead solution.
3. Analyzing the time complexity of the original TESLA and enhanced Inf-TESLA protocols through simulation results and proving it theoretically.
4. Performing comparative analysis between the original TESLA and enhanced Inf-TESLA protocols in terms of the computational overhead on the sender and receiver sides.
5. Studying resistance against cybersecurity attacks including masquerading, modification, man-in-the-middle, and replay attacks.

The remainder of this paper is organized as follows: Section II discusses a related-work review of variant TESLA protocols. Section III demonstrates the discontinuity problem of TESLA protocols, including the Inf-TESLA protocol. Section IV discusses the proposed enhanced Inf-TESLA protocol. Section V explains the authentication issue of the last packets in existing TESLA protocols and our proposed solution. Section VI presents a comparative analysis between the enhanced Inf-TESLA protocol and the original TESLA protocol. Finally, Section VII summarizes the conclusions.

II. RELATED WORK
The TESLA protocol is a broadcast authentication protocol that possesses important properties that allow it to be implemented in constrained IoT devices [6]. It relies on symmetric cryptography, wherein there is a symmetric key shared between the sender and the receiver, and the MAC function that uses the symmetric key to output a MAC number with the original message as an input. The broadcasted packets contain the original message and its MAC value to allow the receiver to authenticate the sender and maintain the integrity of the message. TESLA also provides asymmetric cryptography through a time-delay interval to disclose the symmetric key between the sender and the receiver. In other words, the symmetric key will not be disclosed when the message is being sent, but there is a certain time-delay called disclosure delay (d), during which the receiver needs to wait until the sender reveals the key to be used for authentication of the previous message. Moreover, the loose synchronization time provided between the sender and the receiver helps reduce the computational demands and energy consumption.
of constrained devices [8]. Recent implementation of TESLA protocols involved the authentication of GPS navigation messages and event-driven traffic between the VANET network members [10]–[12]. TESLA protocol is proposed to be used during the real-time nature of VANETs as it uses symmetric key encryption schemes, which are verified by the receiver in a shorter time as compared to using asymmetric digital signatures [13].

The drawbacks of the TESLA protocol lie in the lack of scalability of new IoT devices joining the network or the loss of predefined keychain packets, due to weak communication. It also tends to waste time and energy while establishing synchronization with the same sender, owing to the expiration of the keychain during the communication time window associated with the receiver. Therefore, improvements and updates have been proposed for the original TESLA protocol to increase security, connectivity, and scalability.

Tesla ++ was developed to simplify the messages being transferred between the sender and the receiver to reduce the computational overhead and packet loss [14]. The MAC value is first sent to the receiver, and after the disclosure delay, the original message and disclosed key are sent for authentication purposes. An attacker will not have prior knowledge about the message before the key disclosure and, therefore, will be incapable of manipulating the message. It also reduces the buffering size of messages waiting until the key disclosure. However, the protocol still follows the synchronization establishment between network members upon termination of the keychain, which lacks immediate and continuous authentication.

Staggered TESLA was proposed to reduce the time required to filter the received packets and the probability of buffering overflow while waiting for the key to be disclosed [15]. Additional MAC values of non-authenticated messages are included in the broadcast packet to protect it from being manipulated. These MAC values are related to the time intervals for which their keys are not yet disclosed, and the number of extra MAC values added depends on the network applicability. The inclusion of the MAC values can partially authenticate the packet before disclosing the key by recognizing a familiar pattern. If unusual MAC values are received, the receiver immediately drops the packet without buffering it until the disclosure of the key, which can reduce the buffer overflow in the system. Although staggered TESLA improves the scalability of the IoT network, it increases buffering issues and packet losses when an attacker floods the buffer with replicas of MAC numbers. In addition, it does not support continuous authentication between network members as the key chain terminates.

The $\mu$TESLA protocol aims to simplify the functionality of the Tesla protocol from a broadcast authentication into a unicast authentication, where the sender authenticates one receiver at a time [1], [9], [16]. This process reduces the computational power and the communication bandwidth usage of the receiver receiving unnecessary authentication packets that do not belong to them. An enhancement to the $\mu$TESLA protocol introduces a third trusted party (key server) between the sender and the receiver, responsible for sending the symmetric key while the sender sends the message [17]. This increases the difficulty for the attacker while forging, especially because the key server is hard to compromise. Nevertheless, unicasting the initial key and security parameters delays the joining of new members to the network, which does not support scalability. Moreover, it does not resolve the lack of immediate and continuous authentication presented in the original TESLA protocol.

Multilevel $\mu$TESLA is introduced to reduce the authentication time-delay between the sender and the receiver and the probability of having a denial of service attack [18], [19]. Two keychains are generated, where the high-level keychain is directly connected to the base station, and the low-level keychain is responsible for authenticating the packets transferred between the sender and the receiver. The high-level keychain has a long-time interval to cover the entire lifetime of the receiver without requiring the establishment of a new keychain to reduce the computational complexities and demands. Each time interval in the high-level key chain is further divided into short intervals that correspond to the low-level keychain. The advantage of short time intervals is that they reduce the time required to receive and authenticate the packet, so that the time-delay is tolerable. However, multilevel $\mu$TESLA does not support continuous authentication among the network members. Moreover, the inclusion of two-level keychains increases the computational overhead in the network.

The last improvement to the TESLA protocol is called infinite TESLA protocol (Inf-TESLA), which claims to provide continuous authentication between the sender and the receiver even after the keychain has expired [9]. Inf-TESLA introduces two keychains that are in offset alignment with each other so that if one keychain expires, the other chain continues working and maintains synchronization between the sender and the receiver. These two keychains can follow either the two key modes, where both keys will be sent in the broadcast packet, or an alternating mode, where a key from either chain is presented alternately; if one keychain corresponds to the odd intervals, then the other chain corresponds to the even intervals. The drawback of Inf-TESLA is the limited authentication period between the sender and the receiver, which depends on the keychain duration. In other words, once the two keychains expire, the overall authentication is terminated, and a new synchronization must be established if the sender is still communicating with the receiver. This leads to the absence of continuous authentication between the sender and the receiver during the communication time window. Hence, we propose our enhanced Inf-TESLA protocol that guarantees continuous authentication and regeneration of the required offset keychains as long as the communication channel is running between the sender and the receiver, without the need to establish a synchronization process that drains the battery and buffer space of constrained devices. Table 1 in section VI
summarizes the advantages and disadvantages of the previously discussed variant TESLA protocols, and the proposed Enhanced Inf-TESLA protocol that is discussed later in section IV. A more detailed explanation of the security properties provided by these protocols can be found in [20].

III. DISCONTINUITY PROBLEM IN THE INF-TESLA PROTOCOL

In variant TESLA protocols, when the key level reaches the last key element, the system must re-establish a new synchronization between the same sender and receiver as if they are new to the connection. These unnecessary establishments increase computational demands and energy wastage. To overcome this problem, Infinite-TESLA introduces two keychains that are in offset alignment with each other, so that if one keychain expires, the other chain still works, maintaining synchronization between the sender and the receiver. The commitment keys of the two keychains are sent within the synchronization packet, which is sent only once to the receiver, to reduce the time and energy wasted while signing the packet with a digital signature and verifying the packet at the receiver side. This leads to the condition wherein all receivers that want to connect to the sender need to be available at the beginning of the connection to receive the desired synchronization packet and achieve the necessary security parameters; otherwise, they will miss the opportunity to authenticate the received packets.

Once the two keychains of the Inf-TESLA expire, the connection is terminated, and a new synchronization message is required when the receiver wants to reconnect to the server [21]. This is because during the exchange between the end of the keychain and the generation of another keychain, to continue authentication, a commitment key must be exchanged between the sender and the receiver every time a new keychain is to be generated. Recall that in the Inf-TESLA protocol, the synchronization packet is only sent once at the beginning of the simulation; therefore, the claim of continuous resynchronization in the Inf-TESLA is not valid for more than two keychains.

IV. ENHANCED INF-TESLA PROTOCOL

In this study, we propose a new enhancement to the Inf-TESLA protocol, named enhanced Inf-TESLA, which allows the protocol to maintain communication continuity between the generated keychains without the need to always distribute the synchronization packet to the receiver. The enhanced Inf-TESLA protocol relies on symmetric encryption and hashing functions, which have low computational demands on protocol performance.

Our enhancement begins by generating a master key ($K_m$) to be shared between the sender and the receiver in the synchronization packet, where this master key is going to be among the master keychains responsible for encrypting and decrypting the commitment keys that will be hidden in the last packets of the keychains throughout the lifetime of the receiver. Let us assume that the enhanced Inf-TESLA protocol is processing “keychain $m$” and “keychain $m-1,”” as depicted in Fig.1; once the enhanced Inf-TESLA keychains reach the last elements used to exchange messages, the sender will start generating two new keychains and extracting their commitment keys.

The last packet from keychain $m-1$ and keychain $m$ is used to send the commitment key of the next keychain ($K^m_0$ and $K^{m+2}_0$) encrypted with the master key ($e_{K_m}$). The last packets will contain four elements as follows:

$$P_i \rightarrow \text{msg}_i, \text{MAC}(\text{msg}_i, K^m_{i-1}), K^m_{i-1}, e_{K_m}(K^{m+1}_0) \quad (1)$$

$$P_{i+1} \rightarrow \text{msg}_{i+1}, \text{MAC}(\text{msg}_{i+1}, K^m_{i+1}), K^m_{i+1-d}, e_{K_m}(K^{m+2}_0) \quad (2)$$

where $\text{msg}_i$ is the message sent during the $i$th interval, $\text{msg}_{i+1}$ is the message sent during the $(i + 1)$th interval, $d$ is the disclosure delay assigned by the protocol, $K^m_{i-1}$ is the key used to generate the MAC value of the message at the $i$th interval, $K^m_{i+1}$ is the key used to generate the MAC value of the message at $(i + 1)$th interval, $K^m_{i+1-d}$ is the key to be used to authenticate the message sent at $(i + d)$th interval, and $K^m_{i+1-d}$ is the key to be used to authenticate the message sent at $(i + 1 - d)$th interval. One needs to note that $P_i$ and $P_{i+1}$ are each sent using alternating keychains. Fig.2 shows the last packets of the keychains that contain the hidden commitment keys encrypted using the master key.

FIGURE 1. Discontinuity problem associated with the regeneration of new keychains in the Inf-TESLA protocol.

FIGURE 2. Overview of the insertion process of the new commitment keys in the last packets of the previous keychains in the enhanced Inf-TESLA protocol.
During packet reception, the receiver checks the number of elements sent in the packets to distinguish the upcoming encrypted commitment keys from those of the normal packets sent during the authentication process. The receiver also checks the time at which the packet is received; if the packet is received during the disclosure delay, it is considered as a forged packet and is dropped, otherwise it is saved in the buffer. The master key received in the synchronization packet is then used to decrypt the commitment keys needed to authenticate incoming packets.

To avoid sniffing attacks by malicious users and revealing the original master key value during the broadcasting process, every time the sender needs to use the master key, it is first hashed using a one-way hash function to generate a new master key for encrypting the commitment key. The receiver side performs the same procedure by hashing its master key to generate a new key whenever the master key is required to decrypt the commitment key. This process helps to secure the original value of the master key from being sniffed, as it is protected by the irreversible hashing process and avoids breaking the encryption process of the hidden commitment keys.

Fig. 3 illustrates the hashing process performed at the sender and receiver sides when the last interval of the keychain is reached. At the end of keychain 1, the sender hashes the master key \( K_m \) using a one-way hash function to generate a new master key \( h_1 K_m \) that will act as the secret key of the symmetric encryption. After encrypting the new commitment key, the ciphertext will be sent within the last packet to the receiver. Once the packet is received by the receiver, the number of elements will be counted to distinguish the last packet from a normal broadcasted packet.

The fourth element will then be extracted for the decryption process. Subsequently, the receiver will apply the hashing process to its master key to generate the same hashed master key \( h_1 K_m \) used for the encryption process. This way the sender and the receiver are in explicit synchronization with each other regarding the generation of the hashed master keys used in the encryption and decryption processes.

V. AUTHENTICATION ISSUE OF THE LAST D PACKETS IN THE VARIANT TESLA PROTOCOLS

An unaddressed issue exists in the previously discussed variant TESLA protocols related to the last \( d \) packets (recall that “\( d \)” is the disclosure delay). During the broadcasting process, the last \( d \) packets received by the receiver are not authenticated, as the chain from the sender side ends before disclosing its corresponding keys.

From the example in Fig.4, the last packet received by the receiver contains \( msg_n \), \( MAC_n \), which are generated using the key, and the disclosed key \( K_{n-d} \) is used to authenticate. This indicates that any message sent after the \((n-d)\)th interval will be buffered at the receiver side without being authenticated. To overcome this issue in the enhanced Inf-TESLA protocol, we propose adding additional \( d \) intervals to the sender’s keychain to send the last \( d \) keys to authenticate the remaining buffered packets at the receiver side before starting a new keychain. Fig.5 shows the authentication process of the last \( d \) packets on the receiver side. The last \( n + d \) interval packet contains the disclosed key, which is the last key element used to protect the \( n \)th packet sent to the receiver for buffering. Hence, all the \( n \) packets sent by the sender are verified by the receiver with the inclusion of additional \( d \) packets to the existing keychain. Moreover, the overall computational overhead will not change at the sender or the receiver side, as the number of \( d \) packets added to the chevalier ain is much smaller compared to the size of the keychain. This solution is also implemented in the original TESLA protocol for further comparative analysis.

VI. COMPARATIVE ANALYSES BETWEEN THE ENHANCED INF-TESLA AND ORIGINAL TESLA PROTOCOLS

We performed a simulative comparative analysis between our enhanced Inf-TESLA and the original TESLA protocols by discussing the time complexity analyses of the protocols for
FIGURE 4. Schematic view of the non-authenticated d packets in variant TESLA protocols.

FIGURE 5. Schematic view of adding d extra packets to overcome the authentication issue in the variant TESLA protocols.

the first time and the computational overhead at the sender and receiver sides.

A. TIME COMPLEXITY ANALYSIS

In this section, we analyze the time complexity of the original and enhanced Inf-TESLA protocols from a mathematical and simulation point of view. We used the Big O notation to indicate the time complexity of an algorithm, as it quantifies the amount of time taken by an algorithm to execute as a function of the length of the string representing the input.

1) SIMULATION ANALYSIS

In the time complexity analysis, the most time-consuming part is a repetitive operation that is controlled by the input for which its size can grow/change according to the application specifications. Therefore, we investigated a part of the TESLA algorithm containing a loop, which is related to the keychain size, and studied the effect of changing the keychain length on the overall generation time. We varied the keychain between 100–1000 keys and recorded its generation time. An average simulation data of 30 iterations were recorded for each keychain length, and curve fitting was performed to best describe the resulting data.

From the simulation data and the curve fitting results shown in Fig.6(a) and Fig. 6(b) for both algorithms, we observed that the generation time of the keychain increases linearly with an increase in the number of keys required in the chain. Consequently, the linear relationship in the Big O notation is described by O(n), where n is the number of keys in the chain. The linear relation represents an acceptable level of computation demands for constraint devices.

2) MATHEMATICAL ANALYSIS

We initially considered the original TESLA protocol and discussed earlier that one keychain must be generated for each connection between the sender and the receiver. Therefore, the Big O notation O(n) describes the original TESLA protocol. According to the enhanced Inf-TESLA protocol, two keychains must be present per connection. These two keychains operate in parallel; therefore, the mathematical
representation of each keychain generation is as follows:

\[ O(n) + O(N) = 2O(n) \]  

where \( n \) is the number of keys in the chain. In the time complexity analysis, the constants are neglected; hence, the final Big O notation is represented as \( O(n) \). The representation is also applicable to the Inf-TESLA protocol, as we did not change the number of keychains required in the authentication process.

**B. COMPUTATIONAL OVERHEAD ANALYSIS**

We compared the computational overhead of the enhanced Inf-TESLA algorithms with that of the original TESLA from the sender and receiver perspectives. We studied the time complexity at the sender side by varying the keychain length. We also checked the effect of changing the authentication key size on the generation time of the keychain on the sender side. The key length was varied between 80 and 200 bits, and the simulation data resulted from an average of 50 iterations. The result of the comparison is presented Fig. 7.

According to the results, we observed that the generation time of the keychain in the enhanced Inf-TESLA protocol was higher than that of the original TESLA protocol. This result was expected because the enhanced Inf-TESLA protocol requires two keychains to be generated, whereas the original TESLA protocol requires only one keychain for the authentication process to occur. We also observed that as the key length size increased, the enhanced Inf-TESLA generation time of the keychain approached the original TESLA generation time until they became equal at a key length size of 200. This observation significantly helps in choosing a sufficiently large key length size to help increase the immunity against brute-force attacks and in reducing the probability of the keychain being broken by attackers, without increasing the computational overhead of the system [22].

The next analysis monitors the computational overhead at the receiver side, where two critical parameters are computed:

**FIGURE 6.** Time complexity results of (a) the original TESLA protocol and (b) enhanced Inf-TESLA protocol both following linear relation of \( O(n) \).

**FIGURE 7.** Simulation results of the original TESLA protocol (blue line) and enhanced Inf-TESLA protocol (red line) demonstrating the computational overhead at the sender side during the generation of a keychain as a change in the key size.

**FIGURE 8.** Simulation results of the original TESLA protocol (blue line) and enhanced Inf-TESLA protocol (red line) demonstrating the computational overhead at the receiver side during the generation of the MAC value as a change in the key size to authenticate the received packet.
the generation time of the MAC value and the authentication time of the message. Fig.8 illustrates a comparison between the original and enhanced Inf-TESLA protocols in terms of the generation time of the MAC value. The results revealed that the generation time of the MAC value for the enhanced Inf-TESLA is lower than that of the original TESLA, which proves that the proposed Inf-TESLA reduces the computation complexity of the authentication process [9]. We further observed that the overall increase in the generation time of the MAC values for both protocols was within a small range (∼10 ms), which indicates a systematic behavior among the protocols. Another important observation is that the generation time of the MAC values in the enhanced Inf-TESLA protocol stabilizes as the key length size increases, which is another supporting evidence of the applicability of increasing the key length size for better immunity against brute-force attacks without increasing the computational overhead on the network.

The last analysis is related to the authentication time of the packets at the receiver side and compares the performance of the original and enhanced Inf-TESLA protocols. Fig.9 shows the simulation results related to the change in the authentication time of the packets with respect to the change in the key length size. The first observation is that the authentication time behavior in the enhanced Inf-TESLA protocol is similar to that in the original TESLA protocol, which is another indication for the systematic behavior between the protocols. Furthermore, the authentication time in the enhanced Inf-TESLA protocol is lower than that in the original TESLA by 0.5 ms. Moreover, as the key length increases, the authentication time of the packets increases until it reaches a turning point (around 150 bits key size), after which the authentication time decreases until it reaches its lowest value of approximately 1.6 ms. These results ensure that by using a relatively large key size, the authentication time of the packets would be low enough for a better lifetime of the receivers.

### TABLE 1. Comparison between variant TESLA protocols.

| Protocol  | Advantages                                      | Disadvantages                  |
|-----------|------------------------------------------------|--------------------------------|
| TESLA     | Lightweight cryptography Simple implementation | DOS attack Packet loss Non scalability Non continuity in synchronization |
| TESLA ++  | Reduces buffering                               | No scalability Non continuity in synchronization |
| Staggered TESLA | Reduces authentication delay                     | Increases buffering Non scalability |
| μTESLA   | Saves computation power and memory requirements | Increases buffering Non scalability |
| Improved μTESLA | Reduces the size of transmitted packets | Non continuity in synchronization |
| Multilevel μTESLA | Supports scalability | Increases buffering Increases computation overhead Non continuity in synchronization |
| Improved Multilevel μTESLA | Reduces the recovery time of lost packets | Non continuity in synchronization |
| Infinite TESLA | Supports scalability | No scalability Non continuity in synchronization |
| Enhanced Infinite TESLA | Continuous synchronization | Moderate Computation overhead |
| TESLA     | Continuous synchronization                      | Saves computation power       |

### C. CYBERSECURITY ANALYSIS

Our final analysis considered cybersecurity attacks and studied the resistance of the protocol and its ability to mitigate authentication risks. We also performed a theoretical comparative analysis between our enhanced Inf-TESLA and the previous variant TESLA protocols in Table 1 discussing their advantages and disadvantages to IoT constrained devices.

The most common attacks include the man-in-the-middle and masquerading attacks, where an attacker listens to the broadcasted packets and tries to forge its key while pretending to be a legitimate user to have complete control of the key-chain. To overcome these issues, two keychains are used with
their keys alternating during the broadcast of the packets in Inf-TESLA and the enhanced Inf-TESLA protocols. Hence, the probability of an attacker breaking two keychains in a short time is extremely low.

An alternative way to forge an attack is for the attacker to listen to the packets sent at the end of the keychain, where it is possible to substitute the message and MACs in the packets, causing a modification attack. However, any attempt to forge a key by an attacker during the disclosure delay time will be a violation of the protocol because the protocol does not allow any alternation between keychains during the disclosure delay time [9]. This is because the receiver checks the forged key against the commitment keys related to the new keychains secretly sent in the enhanced Inf-TESLA. If the forged key is not authenticated by any of the commitment keys, the packets are dropped. Another attempt is to flood the receiver using replay packets. Such an attack is solved by including a time stamp for each packet; once the disclosed key of a packet is authenticated, any replays of the same packet containing the same key will be rejected to avoid replay attacks.

VII. CONCLUSION

In this study, we developed a continuous and low-overhead broadcast authentication protocol, named the enhanced Inf-TESLA protocol, to provide continuous authentication between the sender and the receiver while regenerating the necessary offset keychains without re-establishing a new synchronization process that drains the energy and buffer size of the receiver. Symmetric encryption and a hashing function are the only additional processes required to ensure the continuous generation of keychains in the enhanced Inf-TESLA protocol. They have relatively low computational demands, compared to the generation of the synchronization packet. Therefore, the enhanced Inf-TESLA protocol overcomes the discontinuity issue and enhances the lifetime of the network by avoiding unnecessary computations and synchronizations. We also revealed an unaddressed issue in authenticating the last $d$ packets in the previously existing TESLA protocols and proposed a considerable solution by adding $d$ extra packets that send only the required authentication keys, which does not increase the computational demand on the protocol.

The simulation analysis performed using JAVA ensured that the enhanced Inf-TESLA protocol can authenticate packets continuously for the entire lifetime of the receiver, in contrast to the Inf-TESLA protocol. Furthermore, we demonstrated a comparative analysis between the enhanced Inf-TESLA and the original TESLA protocols in terms of the time complexity and critical authentication processes at the sender and receiver sides. The results revealed that both the original TESLA and enhanced Inf-TESLA protocols follow the linear time complexity behavior of $O(n)$, which represents an acceptable level of computational demands to constrained devices. We also showed that the generation time of the MAC values and the authentication time of the packets at the receiver side are lower in the enhanced Inf-TESLA protocol compared with that in the original TESLA protocol.

Finally, we studied the resistance of the enhanced Inf-TESLA protocol against cybersecurity attacks including masquerading, man-in-the-middle attacks, modifications, and replay attacks, which proves the superiority of the enhanced Inf-TESLA in improving the lifetime and performance of the constrained devices. Our future plan is to perform packet loss analysis on the original TESLA, Inf-TESLA, and enhanced Inf-TESLA protocols to study the superiority of the proposed protocol over the existing protocols.

REFERENCES

[1] C. Li, “Security of wireless sensor networks: Current status and key issues,” in Smart Wireless Sensor Networks, Rijeka, Croatia: InTech, 2010.
[2] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler, “SPINS: Security protocols for sensor networks,” Wireless Netw., vol. 8, no. 5, pp. 521–534, Sep. 2002, doi: 10.1023/A:1016598314198.
[3] W. J. Buchanan, S. Li, and R. Asif, “Lightweight cryptography methods,” J. Cyber Secur. Technol., vol. 1, nos. 3–4, pp. 187–201, Sep. 2017, doi:10.1080/23742917.2017.1384917.
[4] S. Kim and R. Shrestha, “Introduction to automotive cybersecurity,” in Automotive Cyber Security. Singapore: Springer, 2020, pp. 1–13.
[5] K. Grover and A. Lim, “A survey of broadcast authentication schemes for wireless networks,” Ad Hoc Netw., vol. 24, pp. 288–316, Jan. 2015, doi:10.1016/j.adhoc.2014.06.008.
[6] A. Perrig, The TESLA Broadcast Authentication Protocol. Boston, MA, USA: Springer, 2003.
[7] A. Perrig, D. Song, R. Canetti, J. D. Tygar, and B. Briscoe, “Timed efficient stream loss-tolerant authentication (TESLA): Multicast source authentication transform introduction,” Netw. Work Gr., vol. 1, no. 1, pp. 1–22, 2005.
[8] S. Krishnakumar and R. Srinivasan, “Securing TESLA broadcast protocol with Diffie–Hellman key exchange,” Int. J. Comput. Aided Netw. Archit. Technol., vol. 4, pp. 152–170, Feb. 2013. [Online]. Available: https://www.academia.edu/2539334/SECURING_TESLA_BROADCAST_PROTOCOL_WITH_DIFFIEHELLMAN_KEY_EXCHANGE
[9] S. Cabrera, D. Anand, V. Pillitteri, and L. Carmo, “Multicast delayed authentication for streaming synchrophasor data in the smart grid,” IFIP Adv. Inf. Commun. Technol., vol. 471, pp. 32–46, May 2016, doi:10.1007/978-3-319-33630-5_3.
[10] S. Bao, W. Hathal, H. Cruickshank, Z. Sun, P. Asuquo, and A. Lei, “Multicast authentication and privacy-preserving scheme for VANETs using TESLA and Bloom filters,” ICT Exp., vol. 4, no. 4, pp. 221–227, Dec. 2018, doi:10.1016/j.ictexp.2017.12.001.
[11] B. Siwenni, M. Mangwala, and J. Chuma, “Modified timed efficient stream loss-tolerant authentication to secure power line communication,” Int. J. Electr. Comput. Eng., vol. 9, no. 4, pp. 2281–2295, Aug. 2019, doi:10.11591/IJECE.V9I4.PP2281-2295.
[12] S. K. A. Theodore, K. R. Gandhi, and V. Palanisamy, “A novel lightweight authentication and privacy-preserving protocol for vehicular ad hoc networks,” Complex Intell. Syst., vol. 2021, pp. 1–11, Oct. 2021, doi:10.1007/s40747-021-00562-Z.
[13] J. Branquinho, C. Senna, and A. Zúquete, “An efficient and secure alert system for VANETs to improve crosswalks’ security in smart cities,” Sensors, vol. 20, no. 9, p. 2473, Apr. 2020, doi: 10.3390/S20092473.
[14] A. Studer, F. Bai, B. Bellur, and A. Perrig, “Flexible, extensible, and efficient VANET authentication,” J. Commun. Netw., vol. 11, no. 6, pp. 574–588, Dec. 2009, doi:10.1109/JCN.2009.608411.
[15] Q. Li and W. Trappe, “Staggered TESLA: A multicast authentication scheme resistant to DoS attacks,” in Proc. Global Telecom. Conf., Dec. 2005, pp. 1670–1675, doi:10.1109/GLOCOM.2005.1577934.
[16] Y. Fan, L.-R. Chen, and M. Eltoweissy, “On optimal key disclosure interval for μTESLA: Analysis of authentication delay versus network cost,” in Proc. Int. Conf. Wireless Netw., Commun. Mobile Comput., Jun. 2005, pp. 304–309, doi: 10.1109/WIRELESS.2005.1549427.
KHOULOUD ELEDLEBI received the B.Sc. degree in communication engineering and the M.Sc. and Ph.D. degrees in electrical and computer engineering from KUST, in 2013, 2015, and 2019, respectively. She is currently a Postdoctoral Fellow at the Cyber Security and Physical Systems (C2PS), Khalifa University. Her research interests include cybersecurity in IoT systems, cognitive radio networking, nanotechnology, and low power semiconductor devices as she had training in modeling of nanoscale device and wireless sensor network optimization and has expertise in several evolutionary computing methods.

AHMED ADEL ALZUBAIDI received the Bachelor of Science degree in computer science from the University of Southampton, U.K., in 2020. He is currently pursuing the master’s degree in computer science with Khalifa University, United Arab Emirates. His master’s thesis topic explores the use of reinforcement learning in autonomous vehicles’ decision making.

CHAN YEOB YEUN (Senior Member, IEEE) received the M.Sc. and Ph.D. degrees in information security from the Royal Holloway, University of London, in 1996 and 2000, respectively. After his Ph.D. degree, he joined Toshiba TRL, Bristol, U.K., and later became the Vice President at the Mobile Handset Research and Development Center, LG Electronics, Seoul, South Korea, in 2005. He was responsible for developing mobile TV technologies and related security. He left LG Electronics, in 2007, and joined ICU (merged with KAIST), South Korea, until August 2008, and then the Khalifa University of Science and Technology, in September 2008. He is currently a Researcher in cybersecurity, including the IoT/USN security, cyber-physical system security, cloud/fog security, and cryptographic techniques, an Associate Professor with the Department of Electrical Engineering and Computer Science, and the Cybersecurity Leader of the Center for Cyber-Physical Systems (C2PS). He also enjoys lecturing for M.Sc. cyber security and Ph.D. engineering courses at Khalifa University. He has published more than 140 journal articles and conference papers, nine book chapters, and ten international patent applications. He also works on the editorial board of multiple international journals and on the steering committee of international conferences.

ERNESTO DAMIANI (Senior Member, IEEE) is currently a Full Professor with the Università degli Studi di Milano, Italy, the Senior Director of the Robotics and Intelligent Systems Institute, and the Director of the Center for Cyber Physical Systems (C2PS), Khalifa University, United Arab Emirates. He is also the Leader of the Big Data Area, Etisalat British Telecom Innovation Center (EBITIC) and the President of the Consortium of Italian Computer Science Universities (CINI). He is also part of the ENISA Ad-Hoc Working Group on Artificial Intelligence Cybersecurity. He has pioneered model-driven data analytics. He has authored more than 650 Scopus-indexed publications and several patents. His research interests include cyber-physical systems, big data analytics, edge/cloud security and performance, artificial intelligence, and machine learning. He was a recipient of the Research and Innovation Award from the IEEE Technical Committee on Homeland Security, the Stephen Yau Award from the Service Society, the Outstanding Contributions Award from IFIP TC2, the Chester-Sall Award from IEEE IES, the IEEE TCSS Research and Innovation Award, and a Doctorate Honoris Causa from INSA-Lyon, France, for his contribution to big data teaching and research.

VICTOR MATEU received the Ph.D. degree in computer science from the University of Lleida, Spain. He is currently working as a Principal Cryptography Researcher at the Technology Innovation Institute (TII). His research interests include post-quantum cryptography looking for new standards resistant to quantum computers and cryptographic protocols both from the design and implementation side.

YOUSOF AL-HAMMADI received the bachelor’s degree in computer engineering from the Khalifa University of Science and Technology (previously known as the Etisalat College of Engineering), Abu Dhabi, United Arab Emirates, in 2000, the M.Sc. degree in telecommunications engineering from The University of Melbourne, Australia, in 2003, and the Ph.D. degree in computer science and information technology from the University of Nottingham, U.K., in 2009. He is currently the Acting Dean of graduate studies and an Associate Professor with the Electrical and Computer Engineering Department, Khalifa University of Science and Technology. His main research interests include the area of information security which include intrusion detection, botnet/bots detection, viruses/worms detection, machine learning and artificial intelligence, RFID security, and mobile security.

* * *