IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 3, 2022, accepted February 14, 2022, date of publication February 18, 2022, date of current version March 3, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3152895

Empirical Studies of TESLA Protocol: Properties, Implementations, and Replacement of Public Cryptography Using Biometric Authentication

KHOULOUD ELEDLEBI[1], CHAN YEOB YEUN[1,2], (Senior Member, IEEE), ERNESTO DAMIANI[1,2], (Senior Member, IEEE), AND YOUSOF AL-HAMMADI[1,2]

[1]Center for Cyber-Physical Systems, Khalifa University, Abu Dhabi, United Arab Emirates
[2]Department of Electrical Engineering and Computer Science, Khalifa University, Abu Dhabi, United Arab Emirates

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 This study discusses the general overview of Timed Efficient Stream Loss-tolerant Authentication (TESLA) protocol, including its properties, key setups, and improvement protocols. The discussion includes a new proposed two-level infinite μTESLA (TLI μTESLA) protocol that solves the authentication delay and synchronization issues. We theoretically compared TLI μTESLA with the previously proposed protocols in terms of security services and showed that the new protocol prevents excessive use of the buffer in the sensor node and reduces the DoS attacks on the network. In addition, it accelerates the authentication process of the broadcasted message with less delay and assures continuous receipt of packets compared to previous TESLA Protocols. We also addressed the challenges faced during the implementation of TESLA protocol and presented the recent solutions and parameter choices for improving the efficiency of the TESLA protocol. Moreover, we focused on utilizing biometric authentication as a promising approach to replace public cryptography in the authentication process.

INDEX TERMS Biometric authentication, lightweight cryptography, machine learning, TESLA protocol.

I. INTRODUCTION TO LIGHTWEIGHT CRYPTOGRAPHY

Currently, the internet of things (IoT) is rapidly expanding and being applied to several fields, such as in healthcare monitoring, environmental monitoring, smart censoring, and vital decision-making in different professional careers. However, the challenging features of IoT include their involvement in constrained devices such as RFIDs, sensor devices, and mobile phones, which have limited energy resources, communication bandwidth, and memory storage. With the increase in the application of these IoT devices, they become vulnerable to malicious attacks, and thus, the implementation of efficient yet lightweight security protocols is urgently needed [1].

Lightweight cryptography involves simplified encryption protocols and schemes with low computational complexity that can be processed on such constrained devices to provide adequate security, considering the limited energy, bandwidth, and memory storage [1], [2]. It implements appropriate cryptographic functions/properties without expensing the power of their constrained devices and occupies less RAM for the applications to enable the network to secure their members and the data [3]–[5].

In context, confidentiality is an essential aspect for maintaining the security services in cryptographic protocols, where only the authorized users in a certain organization or system should be allowed to communicate and transfer information to one another. In addition to authenticating the user or the device, the integrity of the message should not be manipulated by an attacker during transmission. Moreover, the authentication process between two parties should be completed within a short time interval to avoid the occurrence of a DoS attack during the process. Furthermore, the availability of the network members is vital for ensuring the connection and communication with the authorized parties to prevent the connection of a malicious node pretending as a system component. Finally, the entire authentication process should not expose the computational demands and

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

communication bandwidth to avoid a high communication and computation overhead [4].

Therefore, the maintenance of all the security services is becoming a challenge to researchers in the design of cryptographic protocols, and the services are required to be prioritized by focusing on the confidentiality and authentication of users along with providing multiple layers of authorization [5]. However, the integrity of the message, especially for constrained devices, is still a weak property that needs to be maintained during the implementation of simple lightweight cryptographic schemes, where users should be allowed to verify whether the received data is transmitted from a legitimate claimed source and is not being manipulated during the transmission process [5]. All the previous challenges have motivated us to focus on developing a lightweight cryptographic protocol feasible for constrained devices, aiming to achieve user/device authentication and integrity properties, while considering their limited power resources, limited memory space and limited computational capabilities.

In this study, we focused our analysis on the Timed Efficient Stream Loss-tolerant Authentication (TESLA) protocol, which is a lightweight cryptography capable of providing the existing security services with low cost [6]. Additionally, the protocol has the following specific requirements:

1- Simple functions that are understandable and adaptable to several types of IoT devices are implemented to enable appropriate cryptographic properties.
2- The power of the constrained IoT devices is not expensed.
3- A smaller RAM size is occupied during its implementation in IoT devices.

Although the TESLA protocol provides important functionalities, it relies heavily on public key infrastructure (PKI) for initiating the authentication channel between the network members, which increases its vulnerability toward quantum attacks [7].

Our contribution toward the enhancement of the TESLA protocol initiated with the design of a new hybrid TESLA protocol called *two-level infinite μTESLA* (TLI μTESLA), where we theoretically established its ability to provide security services within the acceptable levels of computation and communication demands as compared to previous TESLA protocols [8]. This study aims to further improve and provide simulation analysis to the proposed TLI μTESLA, considering the suitable implementation environments for TESLA protocol, selecting parameters that provide optimum performance, and introducing an alternative to PKI using biometric authentication methods to establish the first line of authentication among the IoT members. We therefore listed our contribution as follows:

1. Establishing security analysis of TLI μTESLA protocol and time complexity comparison with variant TESLA protocols.
2. Performing theoretical analysis on the selection of parameters that help in achieving best performance for TLI μTESLA protocol.
3. Introducing an alternative to PKI for Initiating the authentication channel between the network members using biometric authentication for generating the Initial authentication parameters in TLI μTESLA μTESLA protocol.

The remainder of this paper is organized as follows. The fundamental properties of the TESLA protocol along with its general functionality are presented in Section II. In addition, the list of updates of TESLA protocol is introduced in Section III, wherein the compatibility aspects of the previously proposed hybrid TESLA protocols are discussed in terms of the scalability of IoT. In Section IV, the TESLA protocols are compared in terms of the security services they provide, and the possible implementations of TESLA protocol in IoT systems are summarized. Moreover, the recent challenges faced during the implementation of TESLA protocol along with the proposed solutions and selection of parameters are discussed in Section V. Subsequently, the importance of establishing Root of Trust among IoT members to implement authentication protocols is highlighted in Section VI. Thereafter, in Section VII, the biometric authentication is introduced as a replacement to the public cryptography used for sharing the commitment key and initial security parameters among the IoT members. Finally, a conceptual summary of the proposed methods to secure the biometric data during the authentication process is provided in Section VIII, and the overall discussion along with the conclusions of the current research are presented in Section IX.

II. TESLA PROTOCOL: GENERAL OVERVIEW AND IMPORTANT PROPERTIES

TESLA is a broadcast authentication protocol used in wireless sensor networks (WSNs)/IoT with a single source of trust. In addition, it uses lightweight primitives to realize important properties for implementing the constrained IoT devices [6]. First, it relies on symmetric cryptography with a symmetric key shared between two parties (e.g., sender and receiver). It relies on the message authentication control (MAC) function, which is a pseudorandom function that uses the symmetric key with the original message as an input to generate a MAC value as an output to be used with the original message for transmission to the receiver. Subsequently, the receiver side uses the symmetric key with the original message received as input to calculate its own MAC value from the MAC function that has already been established between the sender and receiver. Therefore, the receiver can review if the calculated number corresponds to the received number for authenticating the sender and the message.

The second vital property of the TESLA protocol is the presence of a delay interval to disclose the symmetric key between the sender and receiver. Thus, the symmetric key will

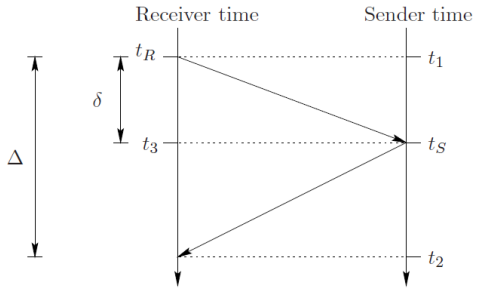

FIGURE 1. Establishment of loose synchronization between sender and receiver in the TESLA protocol.

not be disclosed during the transmission period, but a certain delay is present during which the receiver is required to wait until the sender reveals the key to authenticate the previous message [6]. The delay aids in providing data authentication and integrity review as the attacker will be unable to accurately predict the period until the key is revealed, and consequently, the receiver side would be secured by the time the key is disclosed. This process reduces the probability of the attacker sniffing the key to manipulate and force malicious messages.

The third essential property of the TESLA protocol is the loose synchronization established between the sender and receiver to reduce the computational demands and the energy drain of the constrained devices. The synchronization between the sender and receiver is established to initiate a communication channel, as presented in Fig.1. Generally, the synchronization and sharing of important security properties rely on asymmetric cryptography [9]. The receiver initiates a request message, including the receiver time t_R, and generates a nonce—a number used only once to avoid replay back attacks. Thereafter, the sender receives the message at time t_S and replays back with t_S, and the received nonce is encrypted with the sender private key. At the receiver side, the receiver will authenticate the message by decrypting it using the sender's public key and inspect the nonce in the message. Upon authenticating the message, the receiver records t_S, t_R, and the current time t to calculate the upper bound time expressed as $t - t_R + t_S$. This represents the maximum synchronization error for the receiver to wait until the message is received by the sender and respond back [10].

The security of TESLA protocol relies on a one-way hash chain, which is a chain containing a sequence of keys generated using a one-way hash function [6]. Upon deciding the channel between the sender and receiver, the sender will divide it into sub-time intervals of the same duration. The time-window duration is agreed between the sender and receiver. Each time interval will be protected by a symmetric key from the corresponding key chain. The sender will randomly select a value representing the last key element in the chain and apply it to the one-way hash function for generating the previous key element in the chain. This process continues until the first key element is generated in the chain, which is called the commitment key, K_0 This keychain

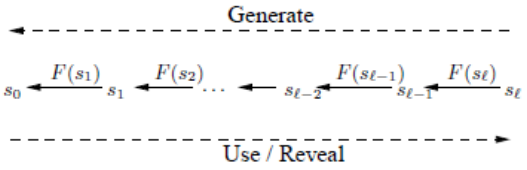

FIGURE 2. Generation of keychain in the TESLA protocol.

exhibits important properties: first, the commitment key can generate and verify any key element in the chain; second, we can verify and generate key K_j from the chain using another key K_i from the chain for any i^{th} value less than the j^{th} value. This is because the lower key elements can be used to generate and verify higher key elements in case one of the keys is lost. During the authentication between the sender and receiver, the disclosure of the keys will be in reverse order—initiating by disclosing the first key element, and thereafter, the second key element, and so on, as presented in Fig.2.

III. UPDATED TESLA PROTOCOLS

Although the TESLA protocol exhibits symmetric properties, it does not support the scalability of new IoT devices joining a system or the loss of the predefined keychain packets owing to weak communication [11]. Therefore, improvements and updates are proposed to the original TESLA protocol to achieve more security services and scalability.

A. TESLA ++

TESLA ++ was developed to simplify the messages transmitting between the sender and receiver to reduce the computation overhead and the loss of packets [12]. In the original TESLA, the calculated MAC value and the original message are sent to the receiver, and after a certain delay, the key is disclosed to be used by the receiver to generate its own MAC value and verify the sender's message. However, once the sender calculates the MAC value in TESLA ++, it will be transmitted only with the index of the time interval that the sender is talking to the receiver, and after a certain delay, the key and original message will be disclosed to the receiver for generating the MAC value and verifying the message. The advantage of this protocol is that if the packet containing the key and message is lost, the attacker will not have prior knowledge of the message before disclosing the key, and therefore, the message cannot be manipulated. Moreover, this reduces the buffering size of the messages waiting until key disclosure.

B. STAGGERED TESLA

Staggered TESLA is proposed to reduce the time required to filter the packets being received by the receiver side and reduce the probability of buffering overflow while waiting for key disclosure [13]. This protocol aims to include several MAC values within the transmitted packet, and these MAC values are related to the time intervals corresponding to the

undisclosed keys to ensure that an attacker cannot manipulate the packet. The number of MAC values included in the message depends on the type of application and the level of security it can manage. This protocol is advantageous because the inclusion of the MAC values in the message can partially authenticate the packet before disclosing the key. For instance, once the receiver can detect a pattern from the MAC values being received from prior authenticated packets, the receiver can authenticate the packet arriving from a legitimate source. In case unusual MAC numbers are received, the receiver will immediately drop the packet without buffering it until key disclosure, which reduces the buffer overflow in the system.

C. μTESLA PROTOCOL

μTESLA protocol aims to simplify the functionality of the TESLA protocol from a broadcast authentication into a unicast authentication, where the sender (base station) authenticates the receivers individually [1], [11], [14]. The protocol relies on the condition that the receiver should review a value related to the time interval of the transmitting base station, to ensure that the key is not disclosed yet. Otherwise, an outside attacker can manipulate the message. This process reduces the computational power and communication bandwidth usage of the receiver receiving unnecessary authentication packets that do not belong to the receiver and can aid in limiting the authenticated users.

D. UPDATED μTESLA PROTOCOLS

To overcome the scalability issue in the μTESLA protocol, researchers improved the scheme through the inclusion of a third trusted party between the base station and receiver [15]. Instead of a single party (base station) sending the message and symmetric key to the receiver, a third trusted party called the key server, responsible for sending the symmetric key, is included, whereas the base station is only required to send the authentication message. This protocol is advantageous in that it includes two parties transmitting key information that cannot be easily forged by the attacker.

An additional advantage of this protocol is considered through the following example: an attacker succeeds in forging its key to the receiver, and any message or key sent for authentication suffers from that single point of failure. In the protocol, the receiver will initiate a threshold value for the maximum error failures of authentication messages arriving from the base station. Moreover, on every instance of an authentication failure, an encounter will start adding these failures until the threshold value is reached. Upon reaching the threshold value, the receiver will initiate a request to the key server to update the key. Thereafter, the key server will review the time interval at which the base station is communicating to that receiver and will transmit the key corresponding to that interval. Subsequently, the receiver will use the received key to authenticate the message transmitted from the base station. In such cases, the successful authentication of the message indicates that the already saved key is malicious, and the protocol will replace it with a new key.

An important stage is securing the communication link between the receiver and key server. As the receiver initiates a request to the key server, the latter will notify the base station regarding the request for updating the key. Thereafter, the base station will broadcast a message containing a new symmetric key used to communicate the key server with the receiver, but this message will be encrypted with a symmetric key that will be disclosed by the key server at a later stage. After a certain delay, the key server will reveal the key to allow the receiver to authenticate both parties and extract the new key for communicating the receiver with the key server.

Furthermore, an additional improvement to the μTESLA protocol is called multilevel μTESLA that provides the advantages of authenticating the base station and reducing the authentication delay between the sender and receiver to reduce the probability of DoS attack [16]. This protocol introduces two keychain levels: a high-level keychain directly connected to the base station, and a low-level keychain responsible for authenticating the messages transferred between the sender and receiver. In particular, the high-level keychain exhibits a long-time interval to cover the entire lifetime of the receiver without requiring an additional establishment of a new keychain, which reduces the computational complexity and demands of the process. Moreover, each time interval in the high-level key chain will be further divided into short time intervals corresponding to the low-level key chain. The use of short time intervals reduces the time required to receive the message from the receiver and to authenticate the message, so that the delay can be within tolerable range to diminish the probability of a DoS attack.

A vital property of this protocol is that the high-level keychain is connected to the low-level keychain such that the low-level keys can be generated from the high-level keys using the one-way hash function in case several low-level packets are lost. The authentication message transmitted from the base station to the receiver is called the commitment distribution message (CDM), which contains the time interval of communication between the receiver and base station, the commitment key of the low-level keychain, the MAC value for the receiver for verification, and the high-level key for authenticating the previous message from the prior time interval. In addition, the CDM packet is periodically transmitted by the base station to reduce the probability of loss, as high-level key packets require a long time to re-establish synchronization between the sender and receiver. Contrarily, this causes buffer overflow on the receiver, including communication and computational overhead.

Owing to the problems discussed for multilevel μTESLA, an improvement protocol called efficient fault-tolerant multilevel μTESLA protocol contributes toward shortening the recovery period of lost high-level packets by acting on a single high-level time interval, which reduces the buffering time and the risk of experiencing memory-based DoS attacks [17]. In context, another improvement to the multilevel μTESLA

is called enhanced DoS-resistant protocol that contributes to tolerating packet loss by reducing the authentication time of CDM packets through adding an image value to these packets and maintaining continuity in occurrence of a packet loss [17]. For instance, if the receiver is receiving the CDM_i at i^{th} time interval, it will contain an image value of the CDM_{i+1} packet. Upon receiving the second packet, the image value will be calculated and compared with the value transmitted in the previous packet for authentication. In case the CDM_{i+1} packet is lost, the receiver will wait for CDM_{i+2} and use the high-level key of the CDM_i packet to verify the key in the CDM_{i+2} packet, as the lower keys from the keychain can verify the higher keys in the chain. In case the verification is achieved, the receiver can utilize the image of the lost CDM_{i+1} packet that is available in the CDM_i packet to provide continuous authentication of the packets.

E. INF-TESLA PROTOCOL

An additional improvement to TESLA Protocol is called the infinite-TESLA, which considers providing continuous resynchronization between the sender and receiver in case the keychain level is terminated [11]. In the original TESLA protocol, when the key level attains the last key element, the system needs to re-establish a new synchronization between the same sender and receiver, such as they are new to the connection. Those unnecessary establishments increase the computational demands and energy wastage. Thus, the Infinite-TESLA introduced two key chains in offset alignment between each other, which maintains the functioning of a chain and the synchronization between the sender and receiver in case a key chain has been terminated. The way these two keys are included in the CDM packet can follow either the two-key mode, where both keys are transmitted in the CDM packet, or they can follow an alternating mode, where a key from either of the chains is presented alternatingly as if one key chain is corresponding to the odd intervals and the other chain is corresponding to the even intervals.

F. TWO-LEVEL INFINITE μTESLA (TLI μTESLA)

We proposed a hybrid TESLA protocol called two-level infinite μTESLA (TLI μTESLA), which combines both the multilevel μTesla and the infinite-Tesla to combine the benefits of reducing the authentication delay as well as providing continuous synchronization between the sender and receiver [8]. The theoretical process of this protocol relies on the hash function and the establishment of loose synchronization between the sender and receiver. Similar to the multilevel-μTESLA, two keychain levels are introduced, where the high-level keychain has a long-time interval to cover the lifetime of the receiver. This keychain will be further divided into sub-intervals to represent the low-level keychain, where the infinite-TESLA protocol is implemented. Additionally, the low-level keychain will contain two keychains in offset alignment to each other; the CDM packet will contain two commitment keys for the low-level keychain with their MAC numbers for verification, including the high-level key related to the previous CDM packet. Similar to the multilevel-μTESLA, the low-level commitment keys in TLI-μTESLA can be derived from the high-level commitment key through a special one-way hash function F_{01}.

IV. SECURITY ANALYSIS AND SERVICES DISCUSSION

Evaluating the computational security of TESLA Protocols relies on the security capability of their respected hash functions: one-way hash function used to generate the keys in the keychain and MAC function used to encrypt the message with Its corresponding key. The design goals of one-way hash function Is to possess preimage resistance (Inability to reverse the output to extract the Input) and collision resistance (considering a low probability of generating the same output from two different Inputs).

Therefore, the best guidance toward ensuring the security of hash function is analyzing the complexity of attacking the previous goals. For an n-bit hash function, an adversary would require 2^n number of operations to produce preimage and $2^{n/2}$ number of operations to produce a collision [18]. By the time the adversary breaks the hash function, the key would be authenticated at the receiver side and the message is received successfully. Regarding MAC function, two Important security properties need to be obtained: key non-recovery and computation resistance of the MAC value. For an adversary to determine the MAC key, exhaustive research is required by checking all possible t number of keys to find a value that agrees with the sent one, which requires a 2^t number of operations. As for guessing the MAC value of a preimage of a given MAC value requires about 2^{-n} number of operations for n-bit MAC algorithm [18]. however, this guessed value cannot be verified without a prior knowledge of either the text message or the key, which makes the probability of forging a malicious MAC value nearly Impossible within the given short authentication time in variant TESLA protocols.

Let us now consider proving the position and integrity properties of the packets delivered by the TESLA protocol. Such discussion applies to all versions of the TESLA protocol, including the one put forward in this paper (TLI-μTESLA protocol discussed in [8] and In section III-F) as they all share the same key-checking provisions. In principle, the properties can be proven by following the hash-chain to verify the relation between the disclosed key and the commitment key. If the relation holds, the received packet occupies in the receiving order the same position it had in the sending order. Also, the disclosed key is the one originally used to encrypt the packet; as a consequence, the packet delivered was not modified after its encryption, and integrity is proven.[1] This proof can be formalized by modeling the TESLA protocol as a finite state automaton where each

[1]For the authenticity property, the disclosed key must be signed. Upon verification of the signature, the receiver can link the holder of the disclosed key to an identity.

TABLE 1. Comparison between TESLA protocols.

Protocol Security Properties	TESLA	TESLA ++	Staggered TESLA	µTESLA	Improved µTESLA	Multilevel µTESLA	Enhanced Multi-level µTESLA	Inf-µTESLA	Proposed TLI µTESLA
Data Integrity	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Immediate Authentication	No	No	Yes	No	No	Yes	Yes	Yes	Yes
DoS Resistance	No	Yes	No	No	Yes	No	Yes	Yes	Yes
Communication overhead	Low	Low	Low	Low	Medium	Low	Low	Low	Low
Computation overhead	Low	Low	Low	Low	Low	Medium	Low	Low	Low
Scalability	No	No	Yes	No	No	Yes	Yes	No	Yes
Continuity of Authentication Process	No	No	No	No	No	No	No	Yes	Yes
Time Complexity	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(n^2)$	$O(n^2)$	$O(n)$	$O(n^2)$

step along the hash chain corresponds to a transition. The properties can thus be proven for any fixed hash-chain, i.e., for any fixed distance between the delivered packet and the initial one. In the general case, however, an infinite state system would be needed to represent the inductive relationship between an arbitrary *i*-th packet and the initial packet. In timed automata, transitions may be put local timing constraints called *invariants*. An automaton can pass through an invariant transition an arbitrary number of times. For such reasons, TAME, a proof engine for timed finite state models, was used in [19] to model TESLA protocol as timed automaton with an invariant, the transition modeling a step along the hash-chain. TAME invariant analysis proves that the TESLA protocol can guarantee the order and data integrity of packets coming at an arbitrary distance from the initial one. The above-mentioned proofs of correctness apply also to our TLI- μTESLA since the core of the authentication process the same and our modifications to the mechanisms did not affect the correctness of the protocol.

Regarding the security of the disclosed key, guessing attacks are not feasible [19], [20] as there is no a strategy that an attacker can use to guess the disclosed key that is better than random guessing. Moreover, the generation of the keys is done using one-way hash function, which is impossible to be inverted, likewise the MAC function, which is designed to be non-invertible. Therefore, choosing a relatively large key size, will decrease the probability of brute force attack to disclose the key and break the keychain to a significate low value [21]. So, by expanding the key space, the protocol can achieve a low-key guessing probability. This proof is also applicable to our TLI- μTESLA which has the same key-checking provisions as the original TESLA protocol proven In [19].

The services properties of the proposed scheme were analyzed by discussing the essential security services and comparing them with the limitations of the previous TESLA protocols. The limitations of TESLA protocol include its inability to support the scalability of IoT devices, as the one-way key chain should be predefined. However, this poses communication and computational demands and can cause loss of packets. Upon the termination of the key chain, a new synchronization process is required to be established between the sender and receiver, which does not support immediate and continuous authentication, and thus, results in vulnerability toward DoS attacks.

The improvement of TESLA++ over TESLA is in terms of buffering of MAC and its index to occupy less memory as compared to the buffering of MAC and message in the TESLA protocol, which aims to reduce the DoS attacks. However, the protocol does not support the scalability of IoT network and follows the synchronization establishment between the network members upon the termination of the key chain, which lacks immediate and continuous authentication.

Although the staggered TESLA improves the authentication process by including the MAC numbers and enhances the scalability of the IoT network, it augments the buffering issues and packet loss if an attacker floods the buffer with replicas of MAC numbers. In addition, it does not support continuous authentication between the network members as the key chain terminates.

The properties of μTESLA are beneficial in saving computation power, communication bandwidth, and memory requirements by reducing the size of the transmitted packets. However, unicasting the initial key and security parameters will delay the joining of new members to the network, which

does not support scalability. Moreover, it does not resolve the problems of the original TESLA protocol, such as the lack of immediate and continuous authentication and the vulnerability toward DoS attacks.

The improved μTESLA protocol improves the resistance against DoS attacks but increases the communication overhead by requiring several exchanges of messages between the key server and base station. Moreover, it does not support immediate and continuous authentication as it requires resynchronization after the termination of the key chain.

Relatively, multilevel μTESLA introduces several improvements including supporting scalability of IoT devices and fault-tolerance toward the loss of packets, as the low-level key chains can be derived from the high-level key chains. Additionally, multilevel μTESLA provides immediate authentication to the CDM message, as several copies of CDM packets are frequently transmitted to reduce the risk of losing high-level packets. However, the copy of the subsequent CDM included in the current CDM increases the size of the CDM as well as the buffering on the sensor nodes, because the copy of the subsequent CDM might be of similar length to the current CDM, which is buffer consuming. Moreover, the inclusion of two-level key chains increases the computation overhead in comparison to the original μTESLA. In addition, multilevel μTESLA does not support continuous authentication between the network members.

In context, enhanced multilevel μTESLA aims to reduce the computation overhead of the multilevel μTESLA by shortening the recovery period of lost high-level packets using a single high-level time interval. Additionally, it tolerates packet loss by reducing the authentication period of CDM packets via adding an image value to these packets and maintaining continuity in the occurrence of packet loss. However, this continuity assumption was not evaluated and analyzed to avoid any high demand of memory resource for the long key chains.

Inf-TESLA provides continuous authentication between the network members, as it reduces the resynchronization process by including dual offset keychains. This reduces the risks of man-in-the-middle attacks in case an attacker attempts to inject the attacker key over the network key chain, wherein the algorithm will notify the receiver regarding the violation of the key-chain exchange procedure. However, Inf-TESLA does not support the scalability of the network members owing to the number of keychains required to be specified prior to the synchronization packets.

In comparison, the proposed TLI-μTESLA protocol enhances the original TESLA with two commitment keys in the CDM message and two low-level key chains and using image value of upcoming CDM instead of using the copy of the subsequent CDM in the current CDM. This allows the protocol to avoid increasing the size of the buffer in the sensor node and reduce the DoS attacks on the network. The low-level key chain exhibits short time intervals to accelerate the authentication process of the broadcasted message with less delay. Additionally, the dual-offset key-chain mechanism is used in the low-level key chains to assure continuous receipt of packets from the high-level key chain. All the services are discussed in detail as follows:

Immediate Authentication: In addition to the symmetric property in TESLA protocol, the proposed protocol relies on the two commitment keys in the low-level keychain for authentication instead of sending a copy of the CDM packet on every instance of transmission between the sender and receiver, which reduces the authentication delay to a tolerable value.

Data Integrity: The originality of the message is maintained by ensuring that it is not altered during transmission, and a higher security level is achieved with the implementation of two keychain layers and offset alignment keychains as compared to alternative TESLA protocols.

Communication and computation overhead: The implementation of two offset alignment keychains realizes the continuous authentication instead of sending copies of CDM packets during transmission, which considerably reduces the communication overhead and computation complexity in comparison to previous TESLA protocols.

Scalability: The successful application of IoT technology to daily-life scenarios involves security schemes that are required to display their ability for adapting to the variations in the environment and the inevitable growth in the amount of work and the number of network members [22]. The implementation of two-level keychains in TLI-μTESLA enhances the broadcasting of the messages to a scalable number of devices and increases the number of messages broadcasted between the members.

Resistance to DoS attacks: The authentication protocols implemented on constrained devices are highly targeted at increasing their immunity against various forms of DoS attacks, including buffer overflow attacks and lack of continuity in the authentication process [23]. In TESLA protocols, a buffering process occurs in the CDM packets until the subsequent packet is received to authenticate the previous message. In particular, the authentication will not occur if the receiver does not have adequate buffer space to wait until key disclosure. This can create network traffic that forces the receiver to drop the packets, thereby increasing the vulnerability of the receiver to DoS attacks. Moreover, a high probability of experiencing communication overhead exists in a constrained network that can result in lost keys and lack of continuity in the authentication process. In the proposed TLI μTESLA protocol, two commitment keys in the low-level keychain are presented to authenticate the message after the disclosure of the high-level key instead of sending a copy of the CDM packet, which reduces the excessive usage of the buffer, and consequently, reduces the vulnerability toward DoS attacks. The short interval in the low-level keychains allows the key to be authenticated immediately without buffering. In addition, the offset alignment of the commitment keys in the low-level keychain allows continuous authentication of the packets received from the

high-level keychain, as the low-level keys are used in an alternate manner. The first keychain index covers the period of the high-level interval, while the second keychain index covers half between the first high-level interval and the next high-level interval, where both commitment keys of the low-level keychains can be derived from the high-level commitment key.

Let us consider an example where both the authentication delay and continuous authentication are solved in TLI μTESLA protocol. At i^{th} time interval, the receiver receives CDM_i packet containing the high-level key K_{i-1}. To authenticate the CDM_i packet, the receiver needs to buffer it until receiving the CDM_{i+1} to use the key K_i disclosed in it. The receiver needs to authenticate K_i by applying the one-way hash function $K_{i-1} = F_0(K_i)$. if the first condition is satisfied, the receiver needs to authenticate the MAC number of the CDM_i packet to authenticate the commitment keys of the low-level keychain. If the first condition is not met, the receiver will drop the packet. On the other side, if the CDM_{i+1} packet is lost, the receiver will wait until CDM_{i+2} is received to use the one-way hash function F_0 to authenticate the high-level key. Consequently, the low-level keychains will be derived from the authenticated high-level key using the one-way function F_{01}. Using the short time intervals in the low-level keychains, the authentication process can be accelerated with less delay, allowing the packets and their keys to be immediately authenticated without oversizing the buffering. Moreover, the presence of the two offset low-level keychains instead of one keychain allow a continuous initialization and authentication of the sensor nodes. Once the first low-level index chain Is expired, the second low-level index chain will continue covering half of the next high-level index chain.

The security services offered by the proposed protocol and the previous improvements to TESLA Protocol in addition to the time complexity of each protocol are comparatively presented in Table 1. Based on a theoretical perspective, we can observe that the core of the TLI-μTESLA protocol is not changed compared to the original TESLA protocol, considering the exchange of the commitment key and other essential security parameters between the server and its clients, to the usage of the one-way hash function and the MAC function to process the security computations during the authentication process. Furthermore, the authenticity of the coming packets in TESLA protocols depend on the previous packets being legitimate as discussed in [19], [21] which indicates a recursive authentication. Therefore, the whole authentication scheme in TESLA must be bootstrapped by guaranteeing that the initial packet is authentic. This is assumed to be done by the sender using the more expensive method of digitally signing the first packet [6].

the additional Improvements proposed In TLI μTESLA protocol can achieve the required services within acceptable computation and communication overhead and with similar time complexity as compared to the existing protocols. Thus, our future step is to verify and prove that the proposed protocol can achieve the security services by performing simulation and numerical analysis. Our first step in this paper is investigating the most suitable environments for implementing the proposed TESLA protocol.

V. CHALLENGES IN THE TESLA PROTOCOL AND PROPOSED SOLUTIONS

Throughout the implementation of TESLA protocol in GPS navigation messages and VANET networks, researchers were concerned about two critical weaknesses: the disclosure delay of the key and the loose time synchronization between the sender and receiver. As discussed in Section II, the disclosure delay is used to introduce the asymmetric property in TESLA Protocol to protect the keys used in authenticating the communication between the network members, whereas the loose synchronization provides simplicity and light-weighted functionality to the protocol. Nevertheless, a long disclosure delay and loose synchronization time error can introduce vulnerability to the protocol by allowing attackers to use the time gap for spoofing the messages with the previously disclosed keys [21]–[25].

The issue of loose synchronization is a critical weakness of the VANET network in implementing the TESLA protocol. Therefore, researchers suggested increasing the awareness of the loose synchronization delay at the sender side to limit the option of sending messages to necessary neighboring vehicles as well as prevent a probable attack [25]. Moreover, the risks of the previous challenges can be reduced and the most suitable performance can be achieved from the TESLA protocol by analyzing the decisions based on certain parametric selections [21]. For instance, the suitable hash function (e.g., SHA-256) must be selected to provide pre-image resistance for reducing the ability of reversing the output inside the hash function and generating the input. In addition, the hash function should permit collision resistance to reduce the probability of generating the same output from two distinct inputs.

Regarding the selection of the hash function, the brute-force attack should be identified; this is a scenario where the attackers perform hash-chain computations to break the keychain by matching their key with the latest released key in the chain. A proposed suggestion to avoid this precomputation and breaking of the keychain is to introduce a type of cryptographic randomness called salt, which is added to the key before it is hashed to generate the previous key in the chain [21]. The salt value can be added to the key following two major approaches: using a timestamp of the key release, which requires a time-varying hash function to be used in a deterministic agreement between the network members and adding a fixed random number to the key before being hashed. The addition of the salt value is required to be the same for all the keys belonging to the same keychain but is required to be altered in case the sender and receiver initiate an additional keychain between each other.

Apart from the addition of the salt value, certain parameters can be controlled in TESLA to reduce the brute-force attack and the probability of success in breaking the keychain.

In context, the key length and keychain length are the most important parameters that strongly influence the reduction in the probability of predicting the key in the chain and the probability of calculating the number of hash functions that the attacker needs to perform to break the key chain. Researchers in [21] studied the influence of various key and keychain sizes on the probability of brute-force attack and determined that the linear increase in the key length is exponentially related to the increase in the immunity toward the brute-force attack. Therefore, they deduced that the keychain size does not need to be quite long if the key length is adequately large. In particular, [26] proposed that a minimum of 128 bits is necessary for maintaining a secure chain. Another study in [21] reviewed the variations in the authentication delay and computation speed upon increasing the key size to achieve a certain level of immunity against brute-force attacks. The results revealed that a shorter authentication time delay allows the algorithm to use smaller key lengths and key sizes. However, the large variations in the authentication delay and computation speed resulted in only small variations in the required key lengths, which maintained the security level of the algorithm even with a long authentication delay.

Regarding the key length size, [24] analyzed the computational load required by the user to apply a TESLA-based navigation-message authentication scheme. TESLA protocol was implemented in four mobile devices with varying processing power and capability to study the effect of the processor on the performance of the TESLA protocol and its energy expenditure. The analysis was related to monitoring the time required for verifying the commitment key, the time required to process the MAC number and message, and the time required to authenticate the last key element in the chain using the commitment key by altering the number of subintervals in the communication channel. The results revealed that the time required for verifying the commitment key or the MAC number was not significantly influenced by the devices as compared to that resulting from variations in the keychain length (time distance between a certain key and the commitment key). The processing required for verifying a key using the commitment key increases with the time distance, which further increases the battery drainage in the network. This indicates that there exists a tradeoff between increasing the key length to achieve higher security levels against brute-force attacks and increasing the computation complexity in the network that affects the power consumption and the lifetime. Therefore, a compromise value must be selected for the key length size to balance the security and energy expenditure in the network. The selection of the parameter values that pose the most influence on TESLA protocol and its performance are summarized in Table 2.

Recent implementation of TESLA protocols involved the authentication of GPS navigation messages and event-driven traffic between the VANET network members [25]–[28]. TESLA protocol is proposed to be used during the real-time nature of VANETs as it uses symmetric key encryption

TABLE 2. TESLA parameter selection for better performance.

Parameter	Value
Hash Function	SHA-2
Key Length	>128 bits
Disclosure time delay	880 ms
Time channel window	10 ms

schemes, which are verified by the receiver in a shorter time as compared to using asymmetric digital signatures [25], [27]. In addition, the TESLA protocol was considered as a favorable option to authenticate the one-way navigation messages owing to its hybrid properties (symmetric/asymmetric functionalities), reduced authentication message size, and the simplicity of symmetric key transfer [28], [29].

With reference to GPS navigation system, TESLA protocol can also be Implemented In location-based services (LBS) to offer an unconditional privacy to the user's query and protects the services offered by the service provider [30]–[32] without revealing the location of the service provider or the user. LBS can be found In VANET where privacy-preserving mechanisms are essential to avoid having a malicious vehicle among the members causing Intentional accidents [33]. therefore, TESLA protocol allows the vehicle to request for services from the location server without revealing the query content to the location server.

TESLA protocol can also be used in urban aircraft mobility (UAM) systems, which have been developed from unmanned aircraft vehicles and have provided the opportunity of highly automated aircrafts operating and transporting passengers or cargo at lower altitudes within urban and suburban areas [28], [29]. Unlike conventional drones flying over unoccupied areas, UAM members are designed to operate over metropolitan areas with high density of population and property. Consequently, an aircraft failure will certainly result in substantial damage. Moreover, the design of such network architecture, including the sensors and the autopilot systems, are more complicated than that in drones. Thus, the UAMs are more exposed to attacks that can target specific data and affect the integrity and availability of the services [29]. Such security requirements are certainly achieved with the TESLA protocols that assure its lightweight property and flexibility between the network members. The implementation of the TESLA Protocol to secure the authentication of the network members will aid in protecting critical navigation data along with providing command and control components with sensor information.

VI. ROOT OF TRUST

During the discussion of existing TESLA protocols, researchers assumed that the initial security parameters, e.g., the hash function, commitment key, and disclosure delay, were already shared between the two parties. However, to simulate the proposed TESLA protocol, we need to

understand the initialization process and transmission of the initial security parameters and the initial symmetric key between the sender and the receiver before establishing the TESLA protocol process. Thus, the concept of the Root of Trust (RoT) is important as it provides the foundational security component of a connected device and is a set of implicitly trusted functions that the remainder of the system or device can use to ensure security [34]–[36]. As IoT is more concerned with wireless sensor network (WSN), we need to understand that WSN is a distributed infrastructure that establishes a trust routine between the members to ensure the security of the communication and integrity of the messages. Typically, RoT exhibits multiple forms depending on the type of the implementation network [35]. For instance, there is a centralized node distributing various hierarchical trust values among the members in the centralized network. Nonetheless, this form can be affected by the central point of failure, e.g., if an attacker manages to attack the central node the entire system will become dysfunctional. An alternative form of trust is in the distributed network, where each node monitors the other nodes in the system and evaluates their trust based on the performance and behavior of the network. However, this addressed value must be frequently updated, which increases the computational demands and depletes the energy of the network.

Another form of trust that seamed feasible to most networks and systems is the certificate-based trust model, wherein a trust party generates the certificates to the users signed by the private key of this trusted party and each node can verify the others' certificates in the system using the public key of the trusted party. This concept forms the basics of PKI that creates the digital certificates to authenticate the members in the network [37]. The types of PKI include the RSA and elliptic curve cryptography, where the latter demonstrated the ability to provide the same security performance but with a shorter key size as compared to the RSA, to enhance its feasibility in application in constrained devices [36].

Although PKIs appear to be highly secured as they rely on three hard mathematical problems (integer factorization problem, discrete logarithm problem, and elliptic-curve discrete logarithm problem), they are vulnerable toward quantum attacks as the evolution of quantum computing improves the processing speed to alleviate the previous problems [37]. Therefore, the primary objective is to replace the PKI that is used for transmitting the initial security parameters and reduce the risk of quantum attacks. For instance, in the implementation of the TESLA protocol on mobile applications, PKI can be replaced by using the SIM platform as the trusted party for transmitting the symmetric key and initial security parameters. However, in sensor devices such as RFID or wireless sensor nodes, we can replace the PKI with biometric tools and biometric authentication schemes that will aid in sending the initial security parameters between the two parties. The following section contains a thorough explanation about biometric authentication and its securing methods that are helpful to generate the root of trust for TLI-μTESLA protocol.

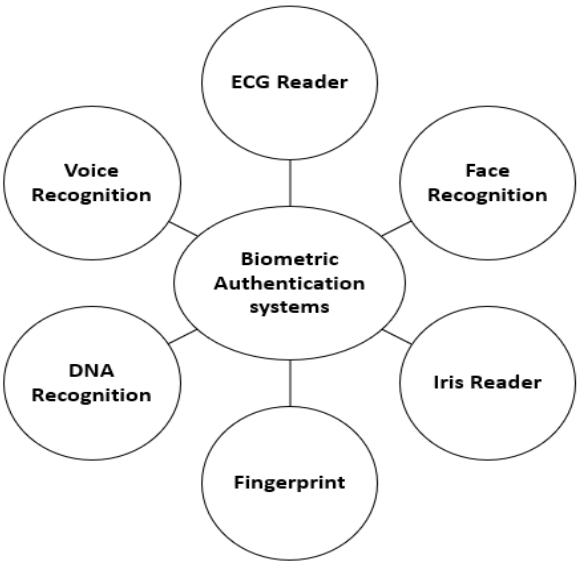

FIGURE 3. Biometric authentication systems.

VII. BIOMETRIC AUTHENTICATION

Biometric authentication is rapidly replacing traditional authentication methods and is becoming a part of everyday life, including accessing banking and government services. It has shown significant advantages in the field of security since it is difficult to lose, forget, copy, forge, and break [38]. The main objective behind using biometric authentication is to try to generate the symmetric key between the two parties from biometrics samples or features for a secure message transmission without revealing sensitive information and without using public cryptography. Examples of biometric tools are electro-cardio diagram (ECG), electroencephalogram, fingerprint, face, iris, and voice-based recognition, as shown in Fig.3.

The most popular type used is the ECG, which allows the user to live monitor the body signals during authentication and is used for different purposes such as in hospitals, security checks, and in wearable devices [38]. Hospitals use ECG data to track patients' health history by registering the patients with their identities and the ECG signals, which need to be sufficiently monitored to perform subsequent identifications. Some security checkpoints are now using ECG authentication to increase their security level. Employees usually register their identities using their ECG that must be stabilized for subsequent recognitions within a short period. Wearable devices can continuously authenticate users; however, in this case, the wearable devices must be able to differentiate between different users' modes such as awake, anger, and sleep modes. All these modes have different signals and different energy demands in addition to the noise generated when monitoring the signal; these must be normalized when analyzing each user to help improve the quality of authentication.

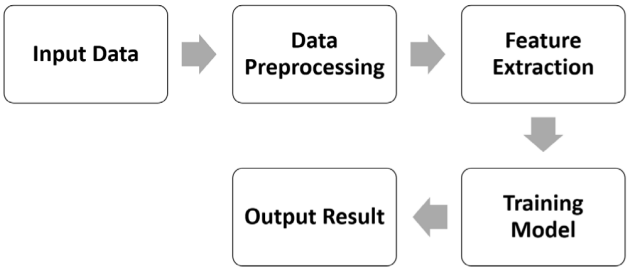

FIGURE 4. Traditional machine learning process.

Biometric authentication has been combined with machine learning techniques to train the models on biometric data, thereby improving the accuracy and efficiency of the authentication process [38]. Machine learning allows systems to perform tasks without being explicitly programmed to do so. Machine learning is therefore being widely used in areas including image processing and biometrics, as it can effectively analyze and interpret large datasets [39]. Machine learning models such as regression models are being used to predict the patterns in the data and generate output based on the identified patterns, or to make decisions using classifiers and pattern recognition models. Fig.4 shows a traditional machine learning process.

Biometric authentication has been discussed in [38], [39] in which ECG data from hospitals and security check points were analyzed for authentication purposes. The first stage involves feature extraction of the ECG signals to identify which case each data sample belongs to. The next stage involves cleaning the data before being imported to the training model, through checking and adjusting the drift between the different data samples, normalizing the different amplitudes of the signal, removing the noise generated during the monitoring process, and correcting flipped signals, if any. The next stage involves dividing the data into subintervals based on the peak-to-peak levels of the ECG signals with a time window determined based on the minimum heartbeat of a certain heart rate to ease the computational process. The following stage involves passing the adjusted data through the training model; in [40], [41], the decision tree was used because of its flexibility in dealing with data of different sizes and frequencies.

Fingerprint biometrics are also very commonly used for authentication and have been discussed in [42] as having two processing phases: user registration phase, which enables the user to use his fingerprint to generate his own private key for later use for authentication; and user authentication phase, which enables authentication between the user and server through the generation of a session key and a message authenticator. A brief explanation of the two phases is provided as follows.

A. USER REGISTRATION

This stage is responsible for registering the user by capturing his fingerprint using feature extraction and selecting minutiae points from the consistent region, which is mostly captured through feature extraction. These points are then applied through convolutional computations to generate the private key.

B. USER AUTHENTICATION

When authentication takes place between the user and server, the fingerprint is first captured and encrypted; it is then sent to the server for verification. The server uses another synthetic fingerprint from its own database to extract the minutiae points, add randomness, and generate security values to create the session key. These values will be sent to the user to generate a similar session from his side. To ensure that both sides generate the same session key, the server generates a certain value "B," encrypts it as "B"' with the session key and sends both B and B' to the user. The user then receives the values, encrypts B using his generated session key, and compares the result with the received B.' The authentication using fingerprint biometrics has shown an accuracy of approximately 95% [42].

A hybrid multimodal authentication protocol was presented in [43], wherein face recognition, fingerprint, and ECG data were used to authenticate the user and achieve gender reveal features. The proposed model uses feature extraction for each dataset, as each set can have distinctive characteristics and requires its own cleaning procedure. Specifically, a deep learning model was used instead of a machine learning one, to ensure that the analysis and classification processes are robust against the noises generated from the different and large biometric datasets. Since these three features (face recognition, fingerprint, and ECG) can be captured using a single device and can be used simultaneously, the model provides high security and immunity to attacks.

Our previous discussion showed the importance of biometric templates in declaring and authenticating the identity of the user during real-time monitoring process. Therefore, by extracting the minutiae points out of the fingerprints, or by generating the cleaned sampled ECG data, we can use them to represent the identity token of the user. The identity token will then be applied to a cryptographic function (e.g.: one-way hash function) to produce the commitment key, which is the essential parameter used for generating the keychain of TESLA protocol and for authenticating the communication channel between the network members, without relying on PKI to transfer the commitment key. The challenging process is protecting the biometric templates from being exposed and from revealing the identity of the user. We therefore discussed in the below section the proposed techniques used to secure the biometric data during the authentication process.

VIII. SECURING BIOMETRIC DATA DURING AUTHENTICATION

Biometrics authentication is widely used in mobile applications to allow access to several sensitive services including banking and government services; hence, it is important to

consider how the biometrical datasets (biometric samples and templates) can be protected from being spoofed by attackers and used to relate them back to the real identity of the user. As such, there were concerns regarding developing protocols to reduce exposing the biometrical identities/samples when performing authentication between the user and server. Among the proposed protocols was the zero-knowledge proof of knowledge protocol, which allows the user, called the ''prover,'' to prove to the other server, called the ''verifier,'' that he knows the value of ''x'' without revealing it but provides proof that he does. The method presented in [44] relies on a trusted party responsible for receiving the biometric identities and protecting them to protect the user identity and its sensitive information from being revealed and sniffed by an attacker during the process. The method consists of two phases to provide secure biometric authentication:

Enrolment phase: In this phase, the user receives an identity token from the identity provider (trusted party) containing three secrets related to the user; one secret is derived from his biometric identity, such as miniature points from his fingerprint or from his ECG signal or from face recognition; another secret is derived from the password; and the third secret is derived from the cryptographic salt value or artifact that will be used in case one of the previous secrets are lost. After establishing the identity token, the biometric templates will pass through the training classifier model to generate the classifier parameters that will be later used to authenticate the user with the server.

Authentication phase: During this phase, the server needs to check the originality of the identity token as well as the identity of the user. The identity token is authenticated by checking the signature of the identity provider by decrypting it using the identity provider public key. The server will then challenge the user by sending a challenge value to be used at the user side with its biometric templates extracted from the feature extraction, his password, and the classifier parameters to perform zero knowledge computations and generate proof values. The proof values will be sent to the server to perform another set of zero knowledge computations and generate results that will determine whether the user is legitimate or not. An additional verification step is then added from the server side to establish a session key to perform a handshake with the user to avoid man-in-the-middle attacks. Random numbers are generated from the server side and sent to the user to use them with his own secrets and establish a session key; the server uses the random numbers generated with the user identity token to generate the same session key, and so, they can initiate the handshake. The primary feature of this method is that it avoids saving the user's biometric templates in either the identity provider or the server. Moreover, the identity provider is not involved in the authentication process; this protects the sensitive information of the user. Furthermore, the addition of the handshake helps in reducing the possibility of a man-in-the-middle attack.

Upgrading the authentication process of mobile services is another matter, as several services based on a single authentication process must be accessed. This concept was introduced in [45], where mutual authentication and key agreement were performed using a single sign in to a trusted party called the token service provider. In this method, the user and the service providers are registered to the token provider; the user uses his biometric samples and password to generate zero knowledge proof values, which are then sent to the token provider to register and receive a token. The service providers also send their certificates and proof of identities for registration and to receive the token from the token provider. After establishing the tokens, the user and the service providers can mutually authenticate each other and communicate without performing an authentication process per service. The advantages of this method are as follows: reduction in the computation and communication overhead through the use of a single authentication process by the token provider; use of a centerless authentication process where the token provider is not included during communication with the service providers, thereby ensuring that sensitive information of the users are well protected, and avoiding the center point of failure on the token provider; and provision of a remote biometric-based authentication process between several services simultaneously, thereby increasing the scalability and usability of the system.

Finally, another method for protecting biometric identities and templates was proposed in [46] to provide blind authentication to both the user and the server side. The proposed method aims to protect the users' biometric identities from the servers and protects the servers' classifiers parameters from the users. A trusted party called the enrolment server will be responsible for establishing the blind authentication between the parties. The user will send the biometric templates from his feature extraction to the enrollment server to pass them through the training model to generate the classification parameters, which will then be sent to the server. During authentication between the user and server, the user will encrypt his biometric identity with his public key and send it to the server to compute the products of the encrypted biometrics and the encrypted classifier parameters and randomize the results for security purposes. The randomized products will then be sent to the user to unlock them and calculate the sum of the products. The resulting sum will be resent to the server to derandomize it and find the result to check it against a threshold value to determine whether to accept or reject that user. The advantage of this method relies on the ability of keeping the sensitive information (user's identity and server's classification parameters) hidden from both parties while still being able to authenticate each other. The method do**es** not involve the use of the enrollment server, which contains all the sensitive information, in the authentication process, thereby avoiding serious losses if the server or the client are compromised.

A conductive numerical proof of ZKP applicability is discussed deeply in [47], [48] to achieve confidential transactions and private smart contracts in blockchain technology.

Moreover, they emphasized on ZKP ability to provide a verifiable proof of the user's identity using remote biometric authentication, without leaking the biometric modalities to untrusted parties. The mentioned proofs can guarantee us that the usage of ZKP during the generation of the biometric commitment key in TLI-μTESLA can help in securing the identity of the user.

IX. CONCLUSION

In summary, we discussed an important lightweight cryptography protocol used in IoT-constrained devices—the TESLA protocol. In addition, the updates and improvements developed were presented, including our proposed TLI-μTESLA, and they were theoretically compared in terms of security services. We highlighted the important parameters of the TESLA protocol, for example, symmetric cryptography, presence of the disclosure delay, reduced message size, and loose synchronization between the network members. Moreover, we discussed the recent implementations of TESLA in the VANET network and GPS navigation message authentication and proposed a new implementation of TESLA in UAMs. The challenges faced during the implementation of the protocol were considered along with the suggested solutions and parameter selections, which will assist in the simulation stage of TLI μTESLA. Our study demonstrated that the determination of an adequately large key length strongly impacts the reduction of brute-force attack during the disclosure delay or the establishment of the loose synchronization between the network members. The addition of the salt value to the key chain aids in reducing the probability of attackers breaking the keychain. Furthermore, the challenges of reducing the involvement of public cryptography during the authentication process is required in the TESLA protocol to avoid quantum attacks through the utilization of biometric authentication to generate the session key. Finally, the authentication schemes using biometric templates revealed the importance of protecting the biometric templates during authentication of other parties in the network.

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. (2002). *SPINS: Security Protocols for Sensor Networks*. Accessed: Mar. 27, 2021. [Online]. Available: http://www.citris.berkeley.edu/

[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, R. Shrestha, 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, R. Canetti, J. D. Tygar, and D. Song, "The TESLA broadcast authentication protocol," Dept. IBM Res., Carnegie Mellon Univ., Pittsburgh, PA, USA, Tech. Rep., 2005. [Online]. Available: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.869.3259&rep=rep1&type=pdf

[7] X. Bogomolec, J. G. Underhill, and S. A. Kovac, "Towards post-quantum secure symmetric cryptography: A mathematical perspective," Cryptol. ePrint Arch., Tech. Rep., 2019.

[8] A. Al Dhaheri, C. Y. Yeun, and E. Damiani, "New two-level μTESLA protocol for IoT environments," in *Proc. IEEE World Congr. Services*, Jul. 2019, pp. 84–91, doi: 10.1109/SERVICES.2019.00029.

[9] S. Suwannarath, *The TESLA-Alpha Broadcast Authentication Protocol for Building Automation System*. Long Beach, CA, USA: California State Univ., 2016.

[10] K. S1 and S. R2. *Securing Tesla Broadcast Protocol With Diffie–Hellman Key Exchange*. Accessed: Mar. 28, 2021. [Online]. Available: https://iaeme.com/ijcet.asp

[11] S. Câmara, D. Anand, V. Pillitteri, and L. Carmo, "Multicast delayed authentication for streaming synchrophasor data in the smart grid," in *Proc. IFIP Adv. Inf. Commun. Technol.*, vol. 471, 2016, pp. 32–46, doi: 10.1007/978-3-319-33630-5_3.

[12] 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.2009.6388411.

[13] Q. Li and W. Trappe, "Staggered TESLA: A multicast authentication scheme resistant to DoS attacks," in *Proc. IEEE Global Telecommun. Conf.*, vol. 3, Dec. 2005, pp. 1670–1675, doi: 10.1109/GLOCOM.2005.1577934.

[14] Y. Fan, I.-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.*, vol. 1, 2005, pp. 304–309, doi: 10.1109/WIRLES.2005.1549427.

[15] D. Ruiying and W. Song, "An improved scheme of μTESLA authentication based trusted computing platform," in *Proc. 4th Int. Conf. Wireless Commun., Netw. Mobile Comput.*, 2008, pp. 1–4, doi: 10.1109/WiCom.2008.1127.

[16] D. Liu and P. Ning, "Multilevel μTESLA: Broadcast authentication for distributed sensor networks," *ACM Trans. Embedded Comput. Syst.*, vol. 3, no. 4, pp. 800–836, Nov. 2004, doi: 10.1145/1027794.1027800.

[17] X. Li, N. Ruan, F. Wu, J. Li, and M. Li, "Efficient and enhanced broadcast authentication protocols based on multilevel μTESLA," in *Proc. IEEE 33rd Int. Perform. Comput. Commun. Conf. (IPCCC)*, Dec. 2014, pp. 1–8, doi: 10.1109/PCCC.2014.7017109.

[18] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*. Boca Raton, FL, USA: CRC Press, Dec. 2018, doi: 10.1201/9781439821916.

[19] M. Archer. (Jan. 1, 2002). *Proving Correctness of the Basic TESLA Multicast Stream Authentication Protocol With TAME*. Accessed: Jan. 18, 2022. [Online]. Available: https://apps.dtic.mil/sti/citations/ADA464932

[20] L. Guo, C. Zhang, J. Sun, and Y. Fang, "A privacy-preserving attribute-based authentication system for mobile health networks," *IEEE Trans. Mobile Comput.*, vol. 13, no. 9, pp. 1927–1941, Sep. 2014, doi: 10.1109/TMC.2013.84.

[21] A. Neish, T. Walter, and P. Enge, "Parameter selection for the Tesla keychain," in *Proc. 31st Int. Tech. Meeting Satell. Division Inst. Navigat.*, Oct. 2018, pp. 2155–2171, doi: 10.33012/2018.15852.

[22] A. Gupta, R. Christie, and R. Manjula, "Scalability in Internet of Things: Features, techniques and research challenges," *Int. J. Comput. Intell. Res.*, vol. 13, no. 7, pp. 1617–1627, 2017, Accessed: Jul. 07, 2021. [Online]. Available: http://www.ripublication.com

[23] N. Ruan and Y. Hori, "DoS attack-tolerant TESLA-based broadcast authentication protocol in Internet of Things," in *Proc. Int. Conf. Sel. Topics Mobile Wireless Netw.*, Jul. 2012, pp. 60–65, doi: 10.1109/iCOST.2012.6271291.

[24] S. Cancela, J. D. Calle, and I. Fernández-Hernández, "CPU consumption analysis of TESLA-based navigation message authentication," in *Proc. Eur. Navigat. Conf.*, May 2019, pp. 1–6, doi: 10.1109/EURONAV.2019.8714171.

[25] M. H. Jahanian, F. Amin, and A. H. Jahangir, "Analysis of Tesla protocol in vehicular ad hoc networks using timed colored Petri nets," in *Proc. 6th Int. Conf. Inf. Commun. Syst. (ICICS)*, Apr. 2015, pp. 222–227, doi: 10.1109/IACS.2015.7103231.

[26] A. J. Kerns, K. D. Wesson, and T. E. Humphreys, "A blueprint for civil GPS navigation message authentication," in *Proc. IEEE/ION Position, Location Navigat. Symp.*, May 2014, pp. 262–269, doi: 10.1109/PLANS.2014.6851385.

[27] S. Bao, W. Hathal, H. Cruickshank, Z. Sun, P. Asuquo, and A. Lei, "A lightweight 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.icte.2017.12.001.

[28] J. A. Maxa, R. Blaize, and S. Longuy, "Security challenges of vehicle recovery for urban air mobility contexts," in *Proc. IEEE/AIAA 38th Digit. Avionics Syst. Conf. (DASC)*, Sep. 2019, pp. 1–9, doi: 10.1109/DASC43569.2019.9081808.
[29] A. C. Tang, "A review on cybersecurity vulnerabilities for urban air mobility," in *Proc. AIAA Scitech Forum*, vol. 1, Jan. 2021, pp. 1–17, doi: 10.2514/6.2021-0773.
[30] V. K. Yadav, N. Andola, S. Verma, and S. Venkatesan, "P2LBS: Privacy provisioning in location-based services," *IEEE Trans. Services Comput.*, early access, Oct. 27, 2021, doi: 10.1109/TSC.2021.3123428.
[31] Y. Pu, J. Luo, Y. Wang, C. Hu, Y. Huo, and J. Zhang, "Privacy preserving scheme for location based services using cryptographic approach," in *Proc. IEEE Symp. Privacy-Aware Comput. (PAC)*, Sep. 2018, pp. 125–126, doi: 10.1109/PAC.2018.00022.
[32] V. K. Yadav, S. Verma, and S. Venkatesan, "Linkable privacy-preserving scheme for location-based services," *IEEE Trans. Intell. Transp. Syst.*, early access, May 5, 2021, doi: 10.1109/TITS.2021.3074974.
[33] V. K. Yadav, S. Verma, and S. Venkatesan, "Efficient and secure location-based services scheme in VANET," *IEEE Trans. Veh. Technol.*, vol. 69, no. 11, pp. 13567–13578, Nov. 2020, doi: 10.1109/TVT.2020.3031063.
[34] L. H. Adnan, H. Hashim, Y. M. Yussoff, and M. U. Kamaluddin, "Root of trust for trusted node based-on ARM11 platform," in *Proc. 17th Asia–Pacific Conf. Commun.*, Oct. 2011, pp. 812–815, doi: 10.1109/APCC.2011.6152919.
[35] M. Momani, "Trust models in wireless sensor networks: A survey," in *Recent Trends in Network Security and Applications* (Communications in Computer and Information Science), vol. 89. Berlin, Germany: Springer, 2010, pp. 37–46, doi: 10.1007/978-3-642-14478-3_4.
[36] Z. Chen, M. He, W. Liang, and K. Chen, "Trust-aware and low energy consumption security topology protocol of wireless sensor network," *J. Sensors*, vol. 2015, pp. 1–10, Jan. 2015, doi: 10.1155/2015/716468.
[37] S. Y. Yan, *Quantum Attacks on Public-Key Cryptosystems*, vol. 9781441977229. New York, NY, USA: Springer, 2013.
[38] S. K. Kim, C. Y. Yeun, E. Damiani, and N. W. Lo, "A machine learning framework for biometric authentication using electrocardiogram," *IEEE Access*, vol. 7, pp. 94858–94868, 2019, doi: 10.1109/ACCESS.2019.2927079.
[39] L. Chato and S. Latifi, "Application of machine learning to biometric systems—A survey," *J. Phys., Conf. Ser.*, vol. 1098, Sep. 2018, Art. no. 012017, doi: 10.1088/1742-6596/1098/1/012017.
[40] S.-K. Kim, C. Y. Yeun, and P. D. Yoo, "An enhanced machine learning-based biometric authentication system using RR-interval framed electrocardiograms," *IEEE Access*, vol. 7, pp. 168669–168674, 2019, doi: 10.1109/ACCESS.2019.2954576.
[41] E. Al-Alkeem, S.-K. Kim, C. Y. Yeun, M. J. Zemerly, K. Poon, and P. D. Yoo, "An enhanced electrocardiogram biometric authentication system using machine learning," *IEEE Access*, vol. 7, pp. 123069–123075, 2019, doi: 10.1109/ACCESS.2019.2937357.
[42] G. Panchal, D. Samanta, A. K. Das, N. Kumar, and K.-K.-R. Choo, "Designing secure and efficient biometric-based secure access mechanism for cloud services," *IEEE Trans. Cloud Comput.*, early access, Apr. 14, 2020, doi: 10.1109/tcc.2020.2987564.
[43] H.-K. Song, E. Alalkeem, J. Yun, T.-H. Kim, H. Yoo, D. Heo, M. Chae, and C. Y. Yeun, "Deep user identification model with multiple biometric data," *BMC Bioinf.*, vol. 21, no. 1, p. 315, Jul. 2020, doi: 10.1186/s12859-020-03613-3.
[44] H. Gunasinghe and E. Bertino, "PrivBioMTAuth: Privacy preserving biometrics-based and user centric protocol for user authentication from mobile phones," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 4, pp. 1042–1057, Apr. 2018, doi: 10.1109/TIFS.2017.2777787.
[45] W. Liu, X. Wang, W. Peng, and Q. Xing, "Center-less single sign-on with privacy-preserving remote biometric-based ID-MAKA scheme for mobile cloud computing services," *IEEE Access*, vol. 7, pp. 137770–137783, 2019, doi: 10.1109/ACCESS.2019.2942987.
[46] M. Upmanyu, A. M. Namboodiri, K. Srinathan, and C. V. Jawahar, "Blind authentication: A secure crypto-biometric verification protocol," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 2, pp. 255–268, Jun. 2010, doi: 10.1109/TIFS.2010.2043188.
[47] J. Partala, T. H. Nguyen, and S. Pirttikangas, "Non-interactive zero-knowledge for blockchain: A survey," *IEEE Access*, vol. 8, pp. 227945–227961, 2020, doi: 10.1109/ACCESS.2020.3046025.
[48] X. Sun, F. R. Yu, P. Zhang, Z. Sun, W. Xie, and X. Peng, "A survey on zero-knowledge proof in blockchain," *IEEE Netw.*, vol. 35, no. 4, pp. 198–205, Jul. 2021, doi: 10.1109/MNET.011.2000473.

KHOULOUD ELEDLEBI received the B.Sc. degree in communication engineering from KUST, in 2013, the M.Sc. degree in electrical and computer engineering, in 2015, and the Ph.D. degree in electrical and computer engineering, in 2019. She is currently a Postdoctoral Fellow at Khalifa University and an Active Member of Cyber Security and Physical Systems (C2PS). Her research interests include cyber-security, AI and ML for IoT devices, cognitive radio networking, nanotechnology, and low-power semiconductor devices as she is trained in the modeling of nanoscale device and wireless-sensor network optimization and possesses expertise in several evolutionary computing methods.

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., he joined Toshiba TRL, Bristol, U.K., and later became the Vice President at LG Electronics, Mobile Handset Research and Development Center, 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, as 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 serves on the editorial board of multiple international journals and on the steering committee of international conferences.

ERNESTO DAMIANI (Senior Member, IEEE) received the Honorary Doctorate degree from the Institut National des Sciences Appliquées de Lyon, France, in 2017, for his contributions toward the research and education of big data analytics. He is currently a full-time Professor with the Department of Computer Science, Universit à degli Studi di Milano, where he leads the Secure Service-Oriented Architectures Research (SESAR) Laboratory. In addition, he is also the Founding Director of the Center for Cyber-Physical Systems, Khalifa University, United Arab Emirates. He is also the Principal Investigator of the H2020 TOREADOR Project on big data as a service. He has published over 600 peer-reviewed articles and books. His research interests include cybersecurity, big data, and cloud/edge processing. He is a Distinguished Scientist of ACM and was a recipient of the 2017 Stephen Yau Award.

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 & Computer Engineering Department, Khalifa University of Science and Technology. His research interests include the area of information security—intrusion detection, botnet/bots detection, viruses/worms detection, machine learning and artificial intelligence, and RFID and mobile security.