Smart Grid Mesh Network Security Using Dynamic Key Distribution With Merkle Tree 4-Way Handshaking

Bin Hu, *Senior Member, IEEE*, and Hamid Gharavi, *Life Fellow, IEEE*

***Abstract*—Distributed mesh sensor networks provide cost-effective communications for deployment in various smart grid domains, such as home area networks (HAN), neighborhood area networks (NAN), and substation/plant-generation local area networks. This paper introduces a dynamically updating key distribution strategy to enhance mesh network security against cyber attack. The scheme has been applied to two security protocols known as simultaneous authentication of equals (SAE) and efficient mesh security association (EMSA). Since both protocols utilize 4-way handshaking, we propose a Merkle-tree based handshaking scheme, which is capable of improving the resiliency of the network in a situation where an intruder carries a denial of service attack. Finally, by developing a denial of service attack model, we can then evaluate the security of the proposed schemes against cyber attack, as well as network performance in terms of delay and overhead.**

***Index Terms*—IEEE 802.11s, EMSA, SAE, security attacks, security protocols, smart grid, wireless mesh networks.**

I. Introduction

WIRELESS local area networks (WLAN) can be deployed in various smart grid domains [1], [2] where a wire line infrastructure does not exist. These networks offer a cost effective solution when compared with other wired or wireless options. Fig. 1 shows a possible deployment of WLAN in various smart grid domains, which includes a home area network (NAN), neighborhood area network (NAN), and substation area network (SAN). To improve the coverage area these networks can extend to mesh networks to overcome their limited transmission range. Currently, mesh networks that are based on the IEEE 802.11s [3] and IEEE 802.15.4g smart utility network (SUN) [4], [5], have been extensively considered for smart grid systems. For neighborhood area networks (NAN), [6] proposes a multigate mesh network that is based on the IEEE 802.11s standard. In this approach, a combination of packet scheduling and multichannel frequency assignment is used. This combination is mainly utilized to solve the bottleneck problem under blackout conditions when a system expects to receive extensive power outage notifications and exchanges.

Manuscript received December 21, 2012; revised April 22, 2013, June 17, 2013; accepted July 18, 2013. Paper no. TSG-00876-2012.

The authors are with the Advanced Network Technologies, National Institute of Standards and Technology, Gaithersburg, MD USA (e-mail: bhu@nist.gov; Gharavi@nist.gov).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSG.2013.2277963

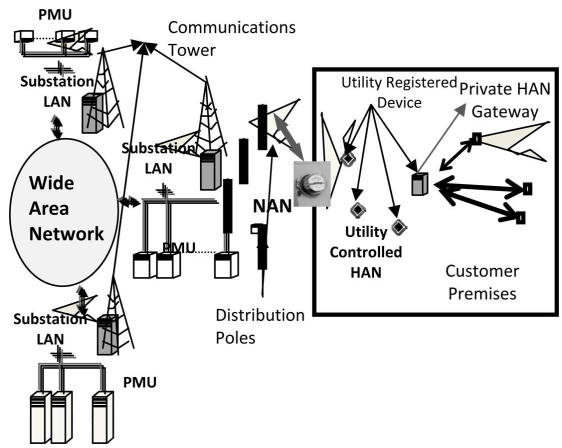


Fig. 1. Application of WLAN for deployment in various smart grid domains.

Fig. 1 also shows an example of a substation area network (SAN). SAN may consist of a number of phasor measurement units (PMUs) that are communicating with the phasor data collector (PDC) located at the gateway connected to the backbone network. A PMU is GPS synchronized to generate high-precision, common time, date packets. These packets need to be transmitted reliably with low delay to the final destination (e.g., super PDC) via the local PDC for archiving, monitoring, or control. While single hop WLAN technologies may be considered as a viable option in the absence of any wired or wireless infrastructure, their mesh extension would require thorough investigation with respect to latency and reliability. Nonetheless, mesh networks offer various unique features such as self-configuration, where the network can incorporate a new device (e.g., meter, PMU, etc.) into the existing structure. In addition, ease of installation, scalability, and self-healing are amongst other important features. Despite these advantages, a major drawback of multi-hop mesh networks is that they are more exposed to cyber attack as data packets have to be relayed on a hop-by-hop basis. For this reason the security of mesh/sensor networks has been a challenging issue in wireless communications. In particular, these networks, due to their lack of infrastructure, would require a distributed approach to authenticate the mesh points (MPs).

So far, there has been a significant amount of work on mesh network security protocols, namely network vulnerability against cyber attack [7]–[15]. For more information about existing security protocols [11] and [12] provide a survey of security requirements for mesh networks. For IEEE 802.11 WLAN networks, the newly adopted IEEE 802.11s standard

was recently released for mesh networks [3]. This standard supports simultaneous authentication of equals (SAE) as its default security protocol. SAE is based on a single password shared by all nodes in the network. Although an attacker may not be able to determine the password through eavesdropping, disclosure of the password would allow unauthorized nodes to join the network, hence compromising the confidentiality and integrity of the network. An alternative approach to SAE is a protocol known as efficient mesh security association (EMSA) [14]. Through the use of a mesh key hierarchy EMSA is capable of establishing link security between two MPs in a wireless mesh network. Since both protocols deploy a 4-way handshaking, the network can become vulnerable to a denial of service (DoS) attack. In particular, through eavesdropping an intruder can easily block the 4-way handshake by forging the unprotected Message-1 [15] or the defective Message-3 that an MP receives from the mesh authenticator (MA). To enhance network protection against such attacks, we had considered a periodic key refreshment and distribution strategy to further protect the network security against a denial of service attack [13]. While the periodic key updating approach can significantly improve the overall security of mesh networks [13], in the 4-way handshaking process Message-1 and Message-3 remain vulnerable to DoS attacks. Therefore, in this paper our main objective is to develop an efficient 4-way handshaking protection scheme. The proposed scheme is capable of improving the security of mesh networks for their deployment in various smart grid domains (see Fig. 1).

The paper is organized as follows: In Section II after a brief overview of SAE and EMSA, we describe the implementation of SAE and EMSA using a multigate mesh networks followed by introducing the key refreshment strategy, which is presented in Section III. In Section IV, we present a denial of service (DoS) attack model by an intruder during a 4-way handshaking process. This section also includes the application of a Merkle tree as well as a one-way hashing to construct a secure 4-way handshaking that can also protect the integrity of the key materials. Finally, in Section V we present the results in terms of delay and overhead based on the frequent updates of the key materials.

II. Mesh Security Systems

The security of a mesh network relies on its ability to protect the message integrity against malicious attacks. This requires guaranteeing the confidentiality and authenticity of the data packet exchanges, which can be achieved by designing a highly reliable association and authentication processes to prevent an attacker (the adversary) accessing the network by originating fake messages to interrupt the network. An example of the latter is a black hole attack where a node can tamper with the routing and prevent packets reaching their intended destinations by sending fake messages (also causing DoS) [11], or making all packets to be routed to itself. To securely maintain operation of the network over the long haul, we developed a strategy that is capable of dynamically changing the key information periodically and/or in situations where an active attack has been detected. Before describing the key refreshment strategy, the following provides a brief description of the mesh security protocols namely, EMSA and SAE.

A. EMSA for Multigate Networks

EMSA services are based on providing an efficient establishment of link security between two MPs in a wireless mesh network through the use of a mesh key hierarchy [14]. As an example, we use a multigate mesh network that was previously developed for NAN [6]. As shown in Fig. 10, this network consists of multiple gateways where every mesh node (e.g., meter) can access each gateway through a separate route. A tree-based routing scheme, which is an extension of the Hybrid Wireless Mesh Protocol (HWMP) of the IEEE 802.11s [3], is used to implement this network. As described in [6], each gateway (GW-1, GW-2, . . .) at the root of a tree periodically broadcasts root announcements to set up its tree. All the gateways are wirelessly connected to the backbone network through the master gateway.

We assume that the master gateway (see Fig. 10) will act as the mesh authenticator (MA), as well as the mesh key distributor (MKD). Within the MKD domain there are a number of gateways and meters (mesh points) [6]. The MKD derives keys to create a mesh key hierarchy. In our network the master gateway is responsible for creating and distributing a mesh key hierarchy to its local gateways and subsequently to all the mesh points after each stage of the authentication process. In other words, the master gateway stores all MP's authentication information. The EMSA operation consists of peer link establishment, followed by EAP (Extensible Authentication Protocol) [16] and 4-way handshaking for the key derivation between every pair of mesh nodes in the network. After Mesh Key Holder Security Handshake (MKHSH) of the EMSA, the authenticated supplicant becomes a mesh authenticator.

At the initial stage the EMSA capability is advertised through beacon and probe response frames using the MKD domain identifier (MKDD-ID) value. This value is received from the MKD during the mesh key holder security handshake. In initial EMSA authentication, an MP carries out its first security association with an MA and establishes mesh key hierarchy for securing future links. This contains communication exchanged between an MP and an MA where a supplicant MP issues an association request frame containing a Peer Link Open IE and the MKDD-IE requests to establish a mesh key hierarchy. The supplication MP is expected to receive an association response frame containing a Peer Link Confirm IE and the information to perform key derivations for establishing link security. If required, the 802.1X authentication [16] occurs next and is followed by an EMSA 4-way handshake.

Prior to the EMSA authentication each gateway (as a supplicant) initiates the link establishment with the master gateway through the Association Request and Association Response frames. This consists of exchanging Peer Link Open and Peer Link Confirm information elements. As soon as the link establishment succeeds, the master gateway begins the authentication process. Under IEEE 802.1X, which also defines EAP over LANs (EAPOL), EAP messages are exchanged between the supplicant (e.g., gateway) and authenticator (e.g., Master gateway). EAP messages from the supplicant are relayed to the

authentication server. In our model we assume that the authentication server and master gateway are co-located (otherwise, EMSA should provide a mechanism for secure communications between the master gateway and mesh key holders). This process in 802.11s is referred to as initial authentication. Upon successful authentication, the master gateway and a supplicant gateway will initiate a 4-way handshake that results in deriving PTK (Pairwise Transient Key) for unicast communications and GTK (Group Transient Key) for multicast communications. After 4-way handshaking, the supplicant MP is now able to receive the route announcement from the mesh authenticator and then has the route to the mesh key distributor (e.g., the master gateway). Before a supplicant MP (e.g., gateway) becomes an authenticator itself, another set of hierarchical key needs to be established via the Mesh Key Holder Security Handshake (MKHSH). This key, referred to as PTK-KD (PTK for Key Distribution), is derived from the KDK (Key Distribution Key) for communication between the supplicant node (e.g., gateway) and the Master gateway. It is used for all communications between the mesh authenticator (see Fig. 2) and mesh key distributor (e.g., Master gateway) when the supplicant becomes a mesh authenticator.

The newly authenticated supplicant gateway then begins to initiate the authentication process for one of its children selected in the routing tree. If the child MP has already been authenticated previously by another neighbor MP (or gateway), the authentication process may consist of only a peer link establishment with 4-way handshaking, but without the need of EAPOL authentication. This is referred to as the "Subsequent Authentication" in [14]. Specifically, in Subsequent Authentication the supplicant MP includes a value of PMK-MKDName in the peer link open message when associating with other MA's. This value is used to identify the PMK-MKD that the supplicant MP generated in its initial EMSA authentication. MA's derive the PMK-MAName based on the received PMK-MKDName and check if they have the corresponding PMK-MA key. If not, they will retrieve that key from the MKD (e.g., Master gateway). After obtaining the desired PMK-MA, the MA's initiate a 4-way handshake without EAPOL authentication.

The process of link establishment and authentication will continue until every node possesses PTK, GTK and PTK-KD throughout the routing tree. We should point out that the multigate network structure routing tree is constructed according to [6].

B. SAE for Multigate Networks

In SAE, a single shared password is used by all MPs to authenticate each other in the absence of knowledge proof [3]. Unlike EMSA, there is no authentication server involved in SAE. In SAE the participating pair of MPs can equally initiate the protocol. Indeed, either side may initiate the protocol simultaneously as their messages are independent of each other [3]. In this paper, the parties involved are defined as MP-A and MP-B and identified by their MAC addresses. After discovering a peer through passively monitoring beacons or active probing, MPs initiate the SAE protocol. Prior to the message exchanges, the involved parties will generate the PWE based on the shared password and their MAC addresses. After generation of the PWE, two random numbers, namely rand and mask, are produced and used with PWE to complete the SAE authentication. Upon successful SAE authentication, both MP-A and MP-B generate a PMK (Pairwise Master Key), which is used in the 4-way handshake to produce PTK and GTK.

In the case of a multigate network, every pair of MPs will perform SAE authentication after discovering each other and generate PMK keys. Security policy is then negotiated in the following association procedure [3]. Based on the PMK keys, a 4-way handshaking is then carried out to generate PTK and GTK.

III. Periodic Key Refreshment Strategy

In this strategy all the key materials will be updated at regular intervals. This is achieved by initiating EAP or SAE authentication and 4-way handshaking to derive a new set of keys before expiration of the existing key materials.

In EMSA, for instance, the lifetimes of the PMK-MKD (Mesh Key Distributor PMK) and KDK should not be more than the lifetime of the MSK (Master Session Key). Also, the lifetime of the PTK and PMK-MA (Mesh Authenticator PMK) should remain the same as that of the PMK-MKD. Similarly, the lifetime of the PTK-KD should be the same as that of the KDK [14]. As soon as the key lifetime expires, each key holder deletes their respective derived keys.

A similar situation occurs in the SAE case where the lifetime of the PMK-R0, PMK-R1, and PTK are bound to the lifetime of the Master PMK (MPMK) from which they are derived. In both cases, upon expiration of the keys' lifetime the corresponding MP's operation will come to an end and will resume only after a successful security process. This can consequently disrupt the operation of the network if the life cycle of the key materials is short. At the same time, if keys remain unchanged over a long period of time (until they expire), the network becomes more vulnerable to cyber attack.

Therefore, to securely maintain operation of the network over the long haul, we developed a strategy that is capable of dynamically changing the key information periodically and/or in situations where an active attack has been detected, as will be further discussed in Section IV. In the absence of any reliable detection scheme, the system can update the key materials seamlessly, hence eliminating network disruption.

Under these conditions, all the key materials, together with MSK, will be updated periodically. For EMSA, MAs refresh the MSK with MKD through EAP authentication, as shown in Fig. 2. Such updates may take place at regular intervals. Therefore, during each MSK lifetime, also referred to as a MSK session, multiple PTK/GTK updates can be performed before expiration of the MSK. This would consequently result in generating a new PTK/GTK through 4-way handshaking. A similar process is applied to the SAE protocol, as seen in Fig. 3. It is important to point out that updating the key materials before expiration will result in maintaining the existing routes in the network; otherwise it would become necessary to carry out a fresh routing and association process of the involved MPs. This

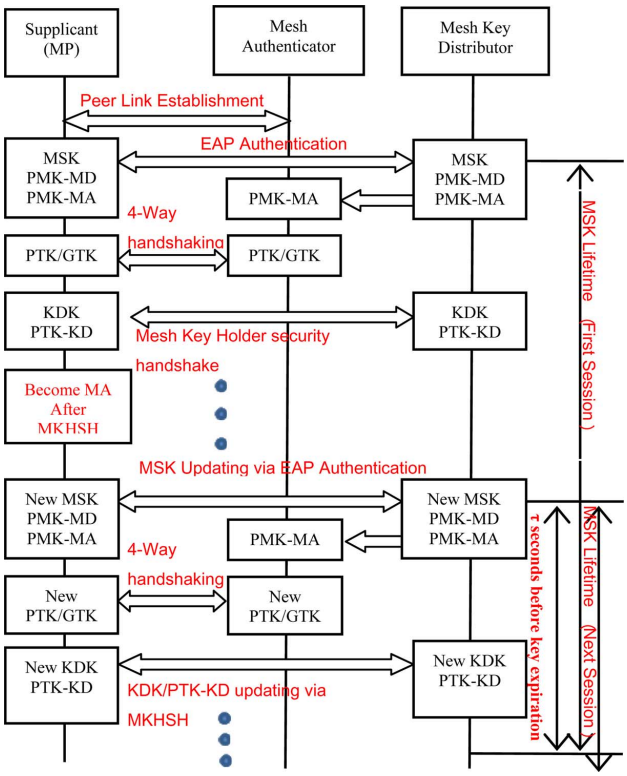


Fig. 2. Periodic-key-updating scheme for EMSA.

would consequently cause a significant delay in re-establishing the network.

IV. Security-Improved 4-Way Handshaking

Protecting the confidentiality and integrity of data packet exchanges would require designing a highly reliable association and authentication processes in order prevent an adversary to originate fake messages that can interrupt the network during the 4-way handshaking process. For example, as shown in Figs. 2 and 3, after acquiring PMK-MA (in EMSA) or PMK-R0 and PMK-R1 (in SAE), the MA and supplicant will begin a 4-Way handshake. It is reasonable to assume that the PMK key (derived after EAP authentication or SAE authentication) is known only to the authenticator and the supplicant.

As stated in [15], attacks are expected to occur only before the generation of the first PTK because of the Link Layer Data Encryption. Therefore, protecting PTK at all times is vitally important as it is nearly impossible to break the cryptographic functions, unless the integrity of the PTK is compromised.

To assess this situation, in our model we assume an intruder is carrying out a DoS attack during the 4-way handshake, to deny the authenticator and supplicant from deriving PTK keys. The intruder is assumed to be able to forge other MPs' MAC addresses, eavesdrop, and forge received messages. Fig. 4 shows the abstract messages that are exchanged in a 4-way handshake, In this figure SPA and AA, SNonce and ANonce, represent the MAC address and Nonces of the supplicant and authenticator, respectively; sn is the sequence number; msg1, 2, 3, 4 are indicators of different message types; and MIC$_{PTK}\{\}$ represents the Message Integrity Code (MIC) calculated for the contents

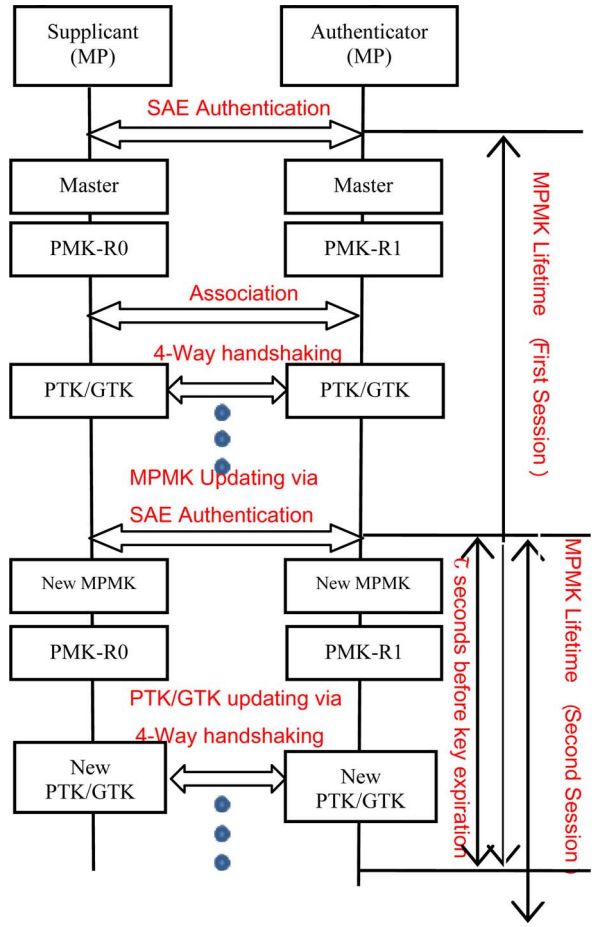


Fig. 3. Periodic-key-updating scheme for SAE.

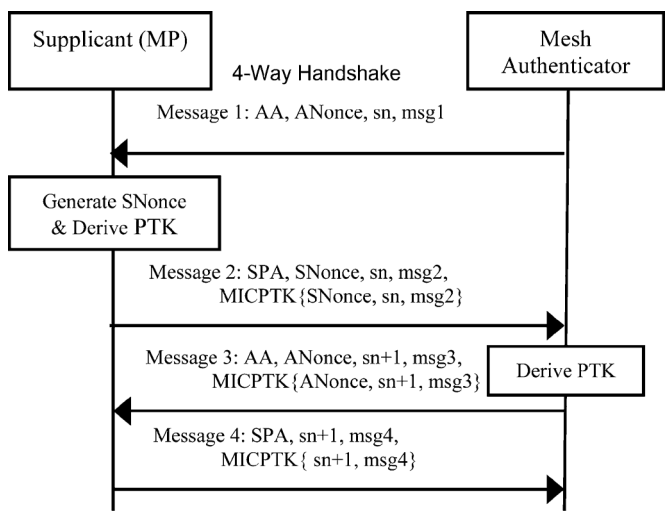


Fig. 4. The 4-way handshaking procedure.

inside the bracket with the fresh PTK [15]. MIC is used to prevent attackers from tampering the message without detection. Bear in mind that in the case of the robust security networks (RSNs) of the IEEE 802.11i, which incorporate the 4-way handshake, the security capabilities, authentication, and cipher key selection are advertised through an RSN information element (RSNE).

Considering that the first message is not encrypted, tampering with it would be an easy task. For example, by taking another

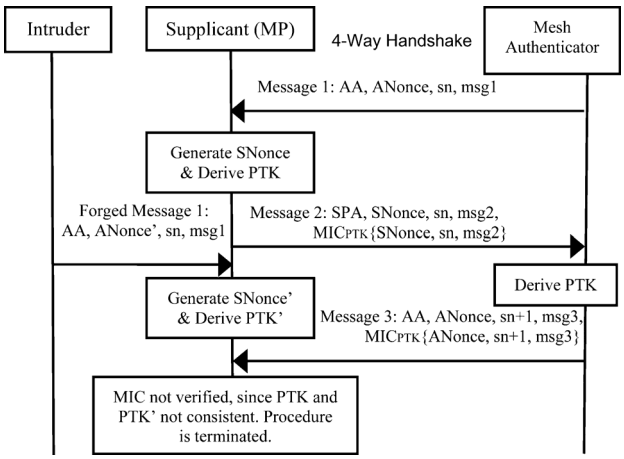


Fig. 5. The DoS attack on Message-1 the 4-way handshaking.

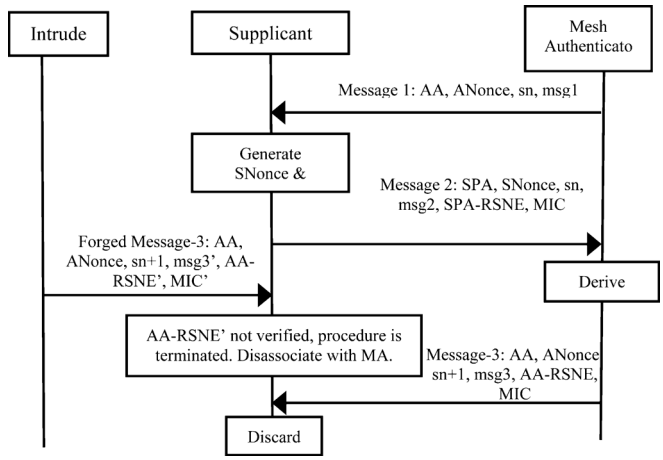


Fig. 6. The DoS attack on Message-3 of the 4-way handshaking.

look at the 4-way handshaking process, as soon as the supplicant has received Message-1, it will have the necessary information (as shown in Fig. 4) to construct its reply message. Subsequently, the supplicant will encrypt Message-2 by computing the MIC over the entire Message-2. This would permit the MA to detect whether the reply message (Message-2) has been tampered with. Since Message-1 is not protected by the MIC field, an intruder would be able to disrupt the 4-way handshake by forging it. For the sake of clarity, this form of DoS attack is depicted in Fig. 5, where an intruder eavesdrops on Message-1 from the authenticator and sends a forged Message-1 with a new ANonce to the supplicant after Message-2. Consequently, the supplicant has to generate a new PTK after receiving the forged Message-1. Obviously, this PTK would be inconsistent with the one in the authenticator, hence causing a termination of the 4-way handshaking process.

One solution to this one-message DoS attack is to store two temporary PTKs (TPTKs) and one PTK in supplicant [15], where TPTK is updated when receiving Message-1, while PTK is updated only upon receiving Message-3 with a valid MIC. This way it would be possible to defeat the DoS attack once the MIC in Message-3 is verified by the two TPTKs or PTK.

Nonetheless, the intruder can still attack the supplicant by employing a multiple-message DoS attack, where forged messages with different Nonces are sent to the supplicant. In this case, the supplicant has to store all the received Nonces, TPTKs, and PTKs, in order to complete the 4-way handshaking with a legitimate authenticator. Unfortunately, this multiple-message DoS attack can exhaust the supplicant's memory and, more importantly, cause a significant delay if the intruder floods huge numbers of forged Messages-1 to the supplicant.

We should point out that in addition to Message-1, a DoS attack can also be carried on Message-3. For instance, after receiving Message-3, the supplicant verifies the Robust Security Network Element (RSNE) by comparing it with the RSNE previously received (either in the Beacon or Probe Response Frame). If the two RSNEs are not identical, the supplicant terminates the 4-way handshake. As indicated in Fig. 6, an intruder can also carry out an attack on Message-3 by forging a Message-3 with a fake AA RSNE', msg3', and MIC'. This clearly indicates that it is not difficult for the intruder to extract and derive the correct AA, Anonce and $\mathrm{sn}+1$ information from the eavesdropped Message-1. In fact, it is quite possible that the intruder can construct and send a faked Message-3 earlier than the MA without requiring any MIC computation. When the supplicant receives the forged Message-3 with correct AA, Anonce and $\mathrm{sn}+1$, it will check the $\mathrm{sn}+1$ and then verify the AA RSNE' [17]. Since the fake RSNE' does not match with what it received before, the supplicant will abort the 4-way handshake and disassociate itself from the MA. While the protection of message-3 is also considered in this paper, we will begin with Message-1.

A form of Message-1 authentication has been suggested in [15]. This method is based on generating a trivial PTK, which is derived from the PMK and is known by both MA and the supplicant. An MIC is then calculated with this trivial PTK. In this way, the intruder cannot forge Message-1. However, if the PMK remains unchanged for a relatively long period, the authenticated Message-1 is still vulnerable to replay attacks [15], [18]. A pair of synchronized counters has been suggested in [18] to avoid replay attacks. Although no details are provided in [18], unfortunately, the design and implementation of the synchronized counters, at the expense of increasing overhead, is problematic especially in wireless environments.

In this paper, as the most reliable solutions, we propose a Merkle-Tree based hashing, as well as a single-hash function scheme. We then apply both to protect Message-1 and Message-3. For this purpose, the MA can use one-way hash functions, such as SHA-1 [19] and SHA-2 [20]. However, in our implementation we apply SHA-1 to construct secure authentication. The following provides further details of the proposed hashing schemes.

As shown in Fig. 7(a), a Merkle Tree [21]–[23] is a binary tree consisting of a set of leaf tokens and internal nodes, each of which is the hash of the concatenation of its left and right children nodes. For instance, we have: $\mathrm{U}_{12} = \mathrm{hash}(\mathrm{U}_1 \,\|\, \mathrm{U}_2)$ and $\mathrm{U}_1 = \mathrm{hash}(\mathrm{V}_1)$, where $\|$ represents the concatenation of two strings. A Merkle tree with a height of H has a set of $m = 2^H$ leaf tokens. Since the used hash function is a one-way function, it is computationally impossible to derive the leaf tokens ($\mathrm{V}_1, \mathrm{V}_2, \mathrm{V}_3$ and V_4) from the root of the Merkle tree: U_{1234} in this case.

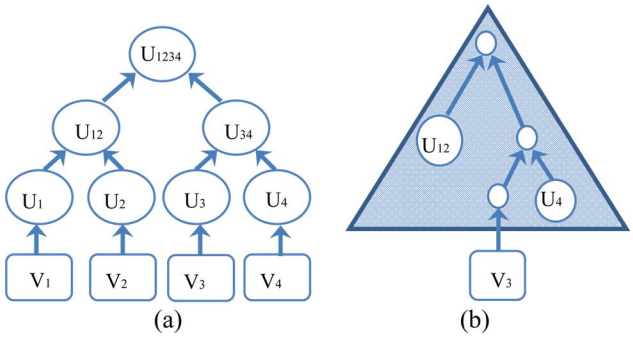


Fig. 7. Merkle Tree Construction. (a) A Merkle tree with 4 leaves and 4 leaf pre-images (V_1, V_2,... V_4), Where $U_{ij} = \text{hash}(U_i \| U_j)$ and $U_i = \text{hash}(V_i)$. (b) The authentication path for the leaf token V_3, consisting of a set of white circle nodes (U_4 and U_{12} in this case).

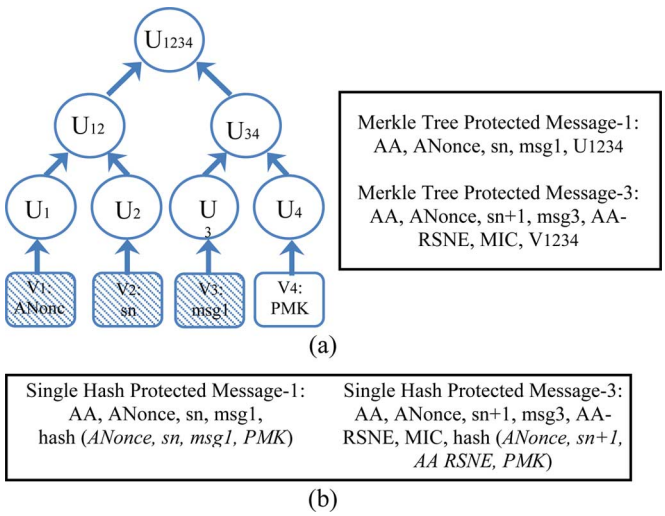


Fig. 8. (a) Merkle Tree Assisted Security-Improvement for Message-1 and Message-3 in 4-way handshaking. (b) Single Hash Protected Message-1 and Message-3.

Specifically, in our Merkle tree-secured 4-way handshaking for Message-1, for instance, two Merkle trees are considered. In the first one, the MA uses ANonce, sn, msg1, and PMK as leaf tokens to derive the root U_{1234}, as shown in Fig. 8. It then includes the encrypted U_{1234} in Message-1. We should point out that, in this approach, PMK information is not included in Message-1. Once the supplicant receives the Merkle Tree secured Message-1, it then uses the ANonce, sn, msg1 from the received Message-1 and its own PMK to compute U_{1234}. It then compares it with the U_{1234} in Message-1 to verify the authenticity of Message-1. Indeed, without the PMK information, the intruder is unable to derive the correct Merkle tree root U_{1234} by using a new ANonce. Thanks to the one-way hash function, the Merkle tree makes it impossible to derive the PMK.

More importantly, to prevent any further replay attacks this Merkle tree will not be used again. For instance, in situations where 4-way handshaking may have to be executed again, the MA constructs the second Merkle tree with $m = 2^H$ random authentication tokens by recursively computing the root, which is referred to as φ in this case. With the help of the second Merkle tree the MA then encrypts root φ with the PMK information and sends it to the supplicant via Message-3 of the first 4-way handshaking. Under these conditions, the MA will identify itself to supplicants by releasing one of the authentication tokens and the corresponding authentication path [shown in Fig. 7(b)] in Message-1. The supplicant will compute a root that is based on the received authentication token and the corresponding authentication path. It then compares the computed root with the stored root φ (achieved during the first 4-way handshaking) to verify the MA's authenticity. Subsequently, the released authentication token is discarded to prevent any replay attacks. Because of the deployment of the one-way hash function, other authentication tokens cannot be derived from the disclosed authentication path, hence they can be used in future 4-way handshaking. As shown in Fig. 7(a), Merkle trees are able to efficiently store and provide multiple one-time authentication tokens to a single root. This will effectively prevent any potential replay attacks. As shown in [24], a Merkle Tree with m authentication tokens requires $O(\log m)$ space and $O(\log m)$ computational effort. Therefore, if the PMK is static for a longer period of time, a Merkle tree with more authentication tokens would be required, hence increasing the complexity.

For Message-3, AA RSNE can be protected by using the first Merkle tree in a similar way. Specifically, the MA uses ANonce, $\text{sn} + 1$, AA RSNE and PMK as leaf tokens to derive a root V_1234, which is then included in Message-3 [see Fig. 8(a)]. Once the supplicant receives the Merkle Tree secured Message-3, it uses the ANonce, $\text{sn} + 1$, AA RSNE from the received Message-3 and its own PMK to compute V_{1234}. It then compares it with the V_{1234} in Message-3 to verify the authenticity of Message-3. Obviously, without the PMK information the intruder won't be able to derive the correct Merkle tree root V_{1234} by using a different AA RSNE. To prevent replay attacks (similar to the Message-1 protection), this Merkle tree shall not be used in any future 4-way handshaking. Again, one of the authentication tokens of the second Merkle tree is used to protect Message-3.

In this paper we have also considered replacing the first Merkle tree by a single hash of Message-1/Message-3. Specifically, for Message-1, the MA uses $ANonce, sn, msg1$ and PMK as input of the one-way hash function to derive a hashed value hash $(ANonce, sn, msg1, PMK)$ and insert it in Message-1. Fig. 8(b) shows the structure of Message-1. Note that, since a one-way function is used to encrypt the PMK information, it is computationally impossible to derive the PMK from message-1. In other words, once the supplicant receives the one way hashing-secured Message-1, it will use the $ANonce, sn, msg1$ from the received Message-1, together with its own PMK, to compute the hashed value. It then compares it with that included in Message-1 for verification. Indeed, without the PMK information, the intruder is unable to derive the correct hashed value by using a new ANonce.

As mentioned earlier, we have also considered a one-way hashing scheme for Message-3 to avoid DoS attacks. Similarly, the MA uses $ANonce, sn + 1, AARSNE$ and PMK as input to derive a hashed value: hash $(ANonce, sn + 1, AA, RSNE, PMK)$ and insert it in Message-3 [Fig. 8(b)]. As soon as the supplicant receives Message-3, it will first check and compare the hashed value before verifying AA RSNE. Again, the intruder is unable to construct a correct hashed value: hash($ANonce, sn + 1, AARSNE'$, PMK) by using a different AA RSNE$'$ within a relatively short time frame.

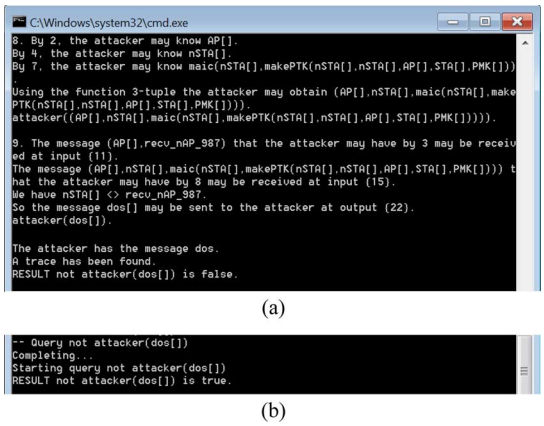


Fig. 9. (a) The result of DoS attacks on the standard 4-way handshake. (b) The result of DoS attacks on the Merkle Tree based 4-way handshake.

We should point out that a single hash is not computationally as efficient as the Merkle tree to verify the authenticity of Message-1/Message-3. For example, in Message-1 Merkle-tree based authentication, the intruder uses a new Anonce to forge an encrypted root U_{1234} while keeping the same sn and $msg1$. Consequently, the MA has to derive only $U_1 = hash(Anonce)$ and then calculates $U_{12} = hash(U_1 \,||\, U_2), U_{1234} = hash(U_{12} \,||\, U_{34})$. In other words, since sn and $msg1$ in the forged Message-1 is same as those in the previous received authentic Message-1, the MA no longer needs to re-calculate U_2, U_3, U_4, and U_{34}. Instead, when using a single hash function, the MA has to use all the elements ($ANonce, sn, msg1$ and PMK) as the input of the single hash function to verify the authenticity of Message-1. More importantly, the Merkle has the flexibility to construct the second Merkle tree, which we have also considered to further enhance the reliability against reply attacks. This is an important feature that cannot be offered by a single hash function.

Protocol Verification

To analyze the flaw of the four-way handshaking process and verify the resistance of the proposed Merkle tree we use ProVerif [25]. The ProVerif is used to reconstruct attacks during the 4-way handshaking process. An execution trace file in Fig. 9 shows that Message-1 and Message-3 received by the supplicant share the same information: unprotected Anonce [nAP in Fig. 9(a)]. In this scenario, the attacker uses a fake Anonce ($nSTA$ in this case) to forge Message-1 and sends it to the supplicant before the authentic Message-3. Since different Anonces generate inconsistent PTKs, the supplicant then fails and discards the authentic Message-3. However, as shown in Fig. 9(b), after applying the Merkle Tree based scheme, there is no further DoS attack on Message-1.

Thanks to the key-refreshment strategy proposed in Section III, the PMK, PTK, and GTK are periodically updated. We can therefore utilize the updated key materials in the proposed one-way hashing schemes in order to better protect Message-1 and Message-3. Bear in mind that more frequent updates of the key materials means a less complex Merkle tree with fewer authentication tokens, but this would be at the expense of more overhead and delay.

V. Simulation Results

In this section, the proposed auto-key-upgrade EMSA and SAE security protocols are investigated using a mesh network, shown in Fig. 10. This network consists of three gateways (GW-1, GW-2, and GW-3) and 36 meters [6].

In the simulation the input data generated at a variable bit rate (VBR), is encapsulated into fixed 512 bytes User Datagram Protocol (UDP) packets. IEEE 802.11b [17] is used in the physical layer and the data-rate is 2 Mbps, while the gateways are assumed to have an unlimited bandwidth. The noise factor is 10.0, as recommended for testing IEEE 802.11b. The path loss factor used in this paper is 2 and the retransmission limit is 7. In the simulations, a set of MSK/MPMK lifetime values, namely 20 seconds, 100 seconds, and 200 seconds, is used to study the impact of the overhead caused by periodical key-refreshment schemes. Furthermore, in each MSK/MPMK session, multiple PTK/GTK updates (e.g., 1, 2 and 5 updates) will be performed to mitigate vulnerability.

In Figs. 11 and 12, we assess the security performance for EMSA and SAE with and without a periodical key-refreshment scheme. When the EMSA and SAE schemes refrain from periodical updates, mesh nodes stop communication with each other as soon as the PTK keys expire and this will result in re-initiating EMSA or SAE authentications. It can be seen from Figs. 11 and 12 that the EMSA and SAE schemes achieve a slightly worse performance than the non-security system when periodic key refreshment is applied. The EMSA and SAE schemes without periodical updating, obtain the worst performance. Obviously, re-initiation of the EMSA or the SAE authentication after the keys' expiration will halt the data transmission temporarily and cause more overhead. Furthermore, SAE schemes outperform EMSA schemes because of less overhead. Fig. 13 shows that at the selected MSK/MPMK lifetime value, the system performance does not degrade significantly with more frequent key refreshment. This is a price that may be worth paying in order to improve system security. Furthermore, it can be seen from Fig. 13 that there is only slight differences when carrying out different numbers of PTK/GTK updates in each MSK/MPMK session. It is reasonable to draw the conclusion that in our schemes, the overhead resulting from 4-way headshaking is negligible.

In Figs. 14 and 15, we construct a DoS attack scenario where an intruder eavesdrops and spoofs neighbors' messages. The simulation results in Fig. 14 demonstrate the extent of the damage caused by the DoS attacks. However, after employing Merkle tree based Message-1 authentication, the EMSA and SAE systems' performances remain unaffected. In this simulation, PMK, PTK and GTK are auto-updated every 200 seconds. A second Merkle tree with a higher number of authentication tokens (i.e., 32) is used to prevent replay attacks. In Fig. 15, we assume that an intruder carries out a black hole attack after it passes EMSA authentication or SAE authentication. In SAE, once the intruder cracks the password or receives it from a legitimate MP, it cannot be excluded from the network without changing this password in all MPs and restarting the network [26]. By contrast, the EMSA is capable of removing the intruder from the network with the involvement of authentication

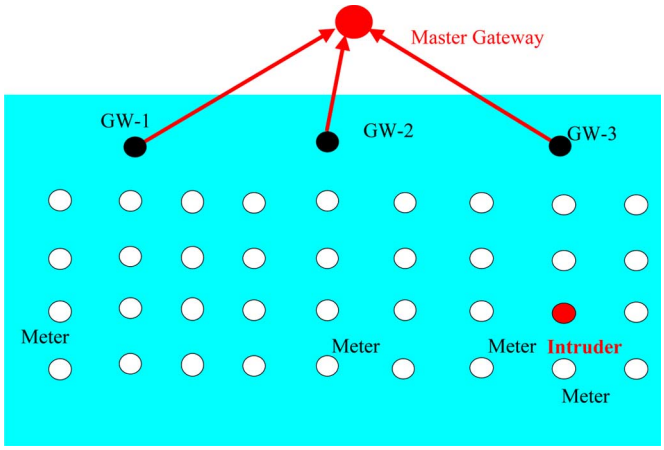


Fig. 10. A multi gateway (GW) network scenario consisting of 3-Gateway and 36 meters.

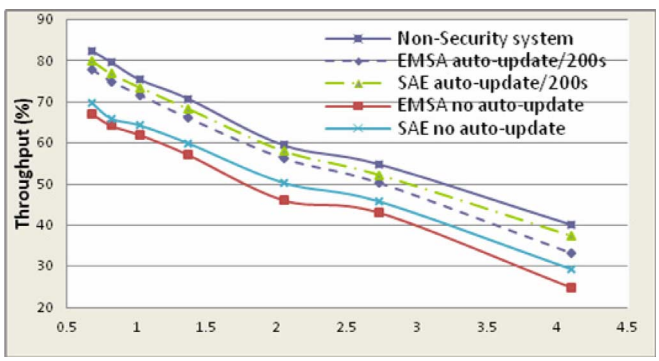


Fig. 11. Throughput performance of the proposed periodic-key-update EMSA and SAE schemes, where the beacon interval is 0.8 second and the MSK/MPMK lifetime is 200 seconds.

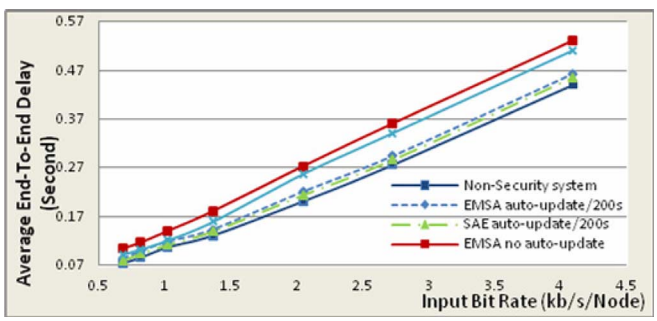


Fig. 12. Delay performance of periodic-key-update schemes.

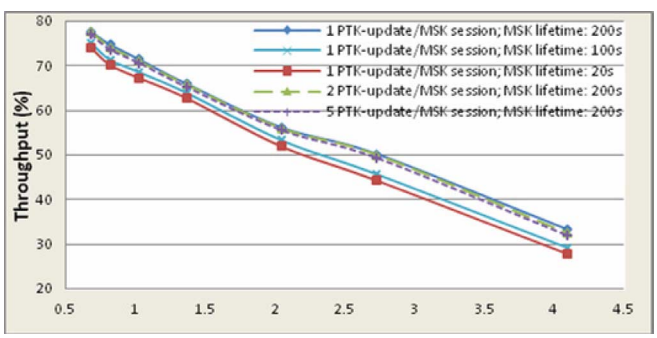


Fig. 13. Throughput performance of the proposed periodic-key-update EMSA with different MSK lifetime and multiple PTK updates per session.

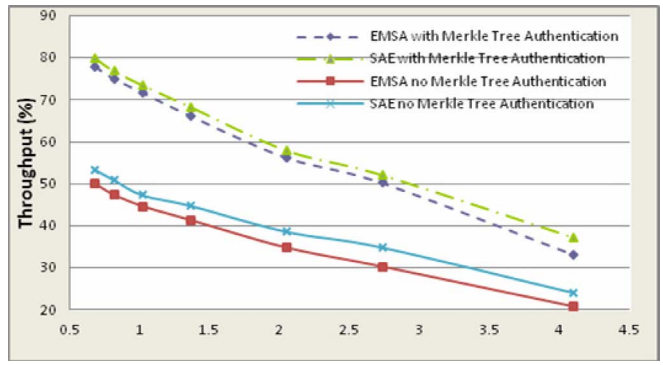


Fig. 14. Throughput performance of the proposed EMSA and SAE schemes when encountering DoS attacks.

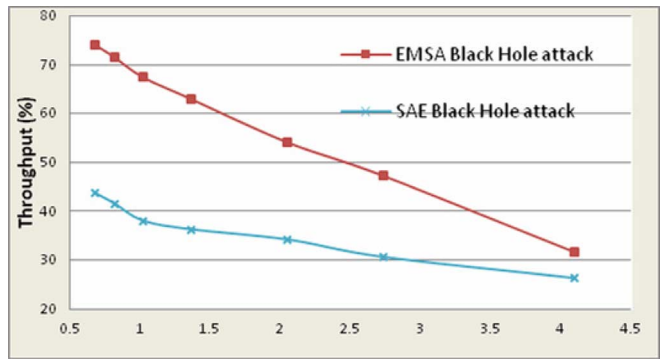


Fig. 15. Performance of the proposed EMSA and SAE schemes when trying to exclude the intruder from the network.

server. Fig. 15 demonstrates this advantage of the EMSA over the SAE.

VI. Conclusion

In this paper we first evaluate the performance of different authentication schemes for a multigate mesh network. We then adopt a strategy which is based on periodical refreshment of key materials and investigate its effect on improving network protection against cyber attack. This includes a denial of service (DoS) attack by an intruder during 4-way handshake message exchanges. To further protect the message exchanges we propose a Merkle tree based authentication scheme as well as a single-hash function scheme. The reliability of the Merkle tree schemes is verified by using Proverif. The simulation results signifies the advantage of EMSA over SAE, as well as demonstrates the effectiveness of the combined proposed key refreshment strategy and Merkle-tree based authentication scheme for both protocols.

References

[1] X. Feng, S. Misra, G. Xue, and D. Yang, "Smart grid—The new and improved power grid: A survey," *IEEE Commun. Surveys Tuts.*, no. 4, pp. 994–980, 2012.

[2] S. Z. Islam, N. Mariun, H. Hizam., M. L. Othman, M. A. M. Radzi, M. Hanif, and I. Z. Abidin, "Communication for distributed renewable generations (DRGs): A review on the penetration to smart grids (SGs)," in *Proc. IEEE Int. Conf. Power Energy (PECon)*, Dec. 2012, pp. 870–875.

[3] *Draft Amendment to Standard for Information Technology—Telecommunications and Information Exchange Between Systems—LAN/MAN Specific Requirements—Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Amendment: ESS Mesh Networking*, IEEE P802.11s/D1.0, IEEE 802.11s Task Group, Nov. 2006.

[4] *Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)—Amendment 4: Physical Layer Specifications for Low Data Rate Wireless Smart Metering Utility Networks*, IEEE Std. 802.15.4g-2012, Mar. 2012.

[5] ZigBee Alliance, ZigBee Specification: ZigBee Document 053474r172008.

[6] H. Gharavi and B. Hu, "Multigate communication network for smart grid," *Proc. IEEE*, vol. 99, no. 6, pp. 1028–1045, Jun. 2011.

[7] X. Wang and P. Yi, "Security framework for wireless communications in smart distribution grid," *IEEE Trans. Smart Grid*, vol. 2, no. 4, pp. 809–818, Dec. 2011.

[8] K. Ren, S. Yu, W. Lou, and Y. Zhang, "PEACE: A novel privacy-enhanced yet accountable security framework for metropolitan wireless mesh networks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 21, no. 2, pp. 203–215, 2010.

[9] Y. Zhang, L. Wang, W. Sun, R. C. Green, and M. Alam, "Distributed intrusion detection system in a multi-layer network architecture of smart grids," *IEEE Trans. Smart Grid*, vol. 2, no. 4, pp. 796–808, Dec. 2011.

[10] J. Mišic and V. B. Mišic, "Wireless sensor networks for clinical information systems: A security perspective," in *Proc. IEEE Int. Conf. Distrib. Comput. Syst. Workshops (ICDCS)*, Jul. 4, 2006.

[11] A. Prathapani, L. Santhananr, and P. D. Agrawal, "Intelligent honeypot agent for blackhole attack detection in wireless mesh networks," in *Proc. IEEE 6th Int. Conf. Mobile Adhoc Sensor Syst. (MASS'09)*, pp. 753–758.

[12] B. He and S. D. P. Agrawal, "An identity-based authentication and key establishment scheme for multi-operator maintained wireless mesh networks," in *Proc. IEEE 7th Int. Conf. Mobile Adhoc Sensor Syst. (MASS)*, 2010, pp. 71–87.

[13] H. Gharavi and B. Hu, "Dynamic key refreshment for smart grid mesh network security," in *Proc. IEEE PES Innov. Smart Grid Technol. (ISGT)*, 2013.

[14] "Efficient mesh security and link establishment," doc.: IEEE 802.11-06/1470r3:, Nov. 2006.

[15] C. He and J. C. Mitchell, "Analysis of the 802.11i 4-way handshake," in *Proc. 2004 ACM Workshop Wirel. Security (WiSe'04)*, pp. 43–50.

[16] *Standard for Local and Metropolitan Area Networks: Port-Based Network Access Control*, IEEE Std. 802.1X-2004, Dec. 2004.

[17] *Standard for Information Technology—Telecommunications and Information Exchange Between Systems—LAN/MAN Specific Requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, ANSI/IEEE Std. 802.11, IEEE 802.11 Standard Working Group, 1999, 1st ed..

[18] Z. Bai and Y. Bai, "4-way handshake solutions to avoid denial of service attack in ultra wideband networks," in *Proc. 3rd Int. Symp. Intell. Inf. Technol. Appl.*, Nov. 2009, vol. 3, pp. 232–235.

[19] Secure Hash Standard, SHA-1, FIPS PUB 180-1 [Online]. Available: http://www.itl.nist.gov/fipspubs/fip180-1.htm

[20] Secure Hash Standard, SHA-2, FIPS PUB 180-2 [Online]. Available: http://csrc.nist.gov/groups/ST/toolkit/secure_hashing.html

[21] R. C. Merkle, G. Brassard, Ed., "A certified digital signature (subtitle: That antique paper from 1979)," in *Proc. CRYPTO 1989*, Santa Barbara, CA, USA, 1990, vol. 435, Lecture Notes on Computer Science, pp. 218–238, Springer.

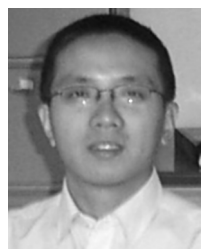
[22] M. S. Islam, Y. J. Yoon, M. A. Hamid, and C. S. Hong, "A secure hybrid wireless mesh protocol for 802.11s mesh network," in *Proc. Int. Conf. Comput. Sci. Its Appl., Part I (ICCSA'08)*, pp. 972–985.

[23] L. Santhanam, B. Xie, and D. P. Agrawal, "Secure and efficient authentication in wireless mesh network using merkle trees," in *Proc. Int. Conf. Comput. Sci. Its Appl., Part I (ICCSA'08)*, pp. 972–985.

[24] M. Szydlo, "Merkle tree traversal in log space and time," in *Proc. Eurocrypt 2004*, vol. 3027, Lecture Notes on Computer Science, pp. 541–554.

[25] B. Blanchet, "An automatic security protocol verifier based on resolution theorem proving (invited tutorial)," in *Proc. 20th Int. Conf. Automated Deduction (CADE'05)*.

[26] A. Egners and U. Meyer, "Wireless mesh network security: State of affairs," in *Proc. IEEE 35th Conf. Local Comput. Netw. (LCN)*, 2010, pp. 997–1004.



Bin Hu (SM'13) received his Ph.D. degree from the School of Electronics and Computer science at University of Southampton, Southampton, U.K., in 2006.

Since September 2006, he has been with the National Institute of Standards and Technology, U.S. Department of Commerce, Gaithersburg, MD, USA, where he is currently a Research Scientist in advanced network technologies division. His research interests include smart grid, security protocols, video/image transmission, mobile communications, and mobile ad hoc networks.

Hamid Gharavi (F'92) received the Ph.D. degree from Loughborough University, Loughborough, U.K., in 1980.

He joined AT&T Bell Laboratories, Holmdel, NJ, USA, in 1982. He was then transferred to Bell Communications Research (Bellcore) after the AT&T-Bell divestiture, where he became a Consultant on video technology and a Distinguished Member of Research Staff. In 1993, he joined Loughborough University as Professor and Chair of Communication Engineering. Since September 1998, he has been with the National Institute of Standards and Technology (NIST), U.S. Department of Commerce, Gaithersburg, MD, USA. His research interests include smart grid, wireless multimedia, mobile communications and wireless systems, mobile ad-hoc networks, and video/image transmission. He holds eight U.S. patents related to these topics.

Dr. Gharavi received the Charles Babbage Premium Award from the Institute of Electronics and Radio Engineering in 1986, and the IEEE CAS Society Darlington Best Paper Award in 1989. He served as a Distinguished Lecturer of the IEEE Communication Society. In 1992 Dr. Gharavi was elected a Fellow of IEEE for his contributions to low bit-rate video coding and research in subband coding for image and video applications. He has been a Guest Editor for a number of special issues, including "Smart Grid: The Electric Energy System of the Future," which has been published in June 2011 by the *Proceedings of the IEEE* and "The Future Smart Grid Signal Processing Challenges," which was published in September 2012 by the IEEE *Signal Processing Magazine*. He served as a member of the Editorial Board of the *Proceedings of the IEEE* from January, 2003 to December, 2008. He is currently a member of the Editorial board, IET *Image Processing*, IEEE ACCESS, and IEEE TRANSACTIONS ON SMART GRID. Dr. Gharavi served as an Associate Editor for the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (CSVT) from 1996 to 2006. He then became the Deputy Editor-in-Chief of this IEEE Transactions through December 31, 2009. Currently, he is serving as the Editor-in-Chief for CSVT. He was a core member of the Study Group XV (Specialist Group on Coding for Visual Telephony) of the International Communications Standardization Body CCITT (ITU-T).