

Identity Based Key Distribution Framework for Link Layer Security of AMI Networks

Vahe Seferian, *Student Member, IEEE*, Rouwaida Kanj, *Senior Member, IEEE*,
Ali Chehab, *Senior Member, IEEE*, and Ayman Kayssi, *Senior Member, IEEE*

Abstract—Advanced metering infrastructures (AMIs) enable two-way communications between the utility and customers opening up opportunities for an efficient and reliable smart grid. This, however, opens up the grid to a vast amount of attack vectors and intruders. Therefore, cyber security is an indispensable part of the smart grid communications’ infrastructure and must be considered in the early phases of planning. The achievement of a secure grid infrastructure requires cryptographic primitives whose strength is dependent on the utilized keys, raising the need for secure key generation/distribution mechanisms. Furthermore, the underlying security mechanisms should be scalable due to the large AMI network sizes. In this paper, we propose the usage of identity-based-cryptography as the basic block of the key distribution mechanism to generate secret keys between neighboring smart meters in a non-interactive manner. Physical unclonable functions are proposed on the hardware platform to eliminate key compromise at the hardware level. We also propose a secure mechanism for updating private keys that are utilized as a seed within the identity-based cryptosystem. A lightweight key delivery mechanism exploiting multicast network features is presented along with a new authentication algorithm for the updated keys. Hardware tests are performed on a Cortex M3-based microcontroller to mimic smart meters’ processors and assess the feasibility of the proposed key authentication. Experiments conducted on a network simulator validate the feasibility of the proposed methodology indicating key delivery latency reduction up to 80% and network traffic reduction up to 27%.

Index Terms—Security, advanced metering infrastructure, smart meter, key distribution, identity based security.

I. INTRODUCTION

THE ESTABLISHMENT of communication channels between the different components of the smart grid has a significant role in increasing the reliability of power delivery networks. The Advanced Metering Infrastructure (AMI) is often referred to as the last mile of the smart grid that connects power consumers to the utility. By reporting real time data, smart meters have a critical role in aiding the utility in demand

management and detection of grid failure problems, such as early blackout detection, prior to their occurrence [1], [2]. However, security mechanisms intended for the AMI should ensure resiliency against a vector of cyber-attacks and ensure scalability to accommodate for a vast amount of users and data exchange. Data exchanged between smart meters and the utility traverse several hops before reaching the final destination. If an adversary captures unencrypted data, she can infer users’ activities thereby causing a major privacy breach. An adversary may also flood the network with bogus packets to create a Denial of Service attack (DoS) preventing meters or utility packets from reaching the other end. She may also craft packets to exploit certain functionalities such as the ‘remote disconnect’ functionality [3]. For such reasons, end-to-end data between meters and the utility should be (a) encrypted to ensure confidentiality and (b) authenticated to ensure integrity and to verify the source of data.

Several end-to-end security mechanisms such as [4]–[6] have been proposed. Nabeel *et al.* [4] rely on end-to-end encryption and authentication mechanisms by using physical unclonable function devices (PUFs) to generate secret keys. So *et al.* [5] propose a generic and scalable ID-based “*sign-cryption*” scheme to enable end-to-end secure communications in the smart grids. In [7], a Merkle tree based authentication scheme is proposed to secure the communication of data reports between smart meters and the neighborhood gateways of the AMI network. The design relies on keys generated via Diffie Hellman key exchange between the meter and the gateway while taking into consideration the computational constraints of the smart meters. There is also a need to guard against attacks targeting layers below the application layer. This includes spoofing, tampering and Denial of Service (DoS) attacks along with the different types of Layer 2 routing attacks [8]. Several security frameworks have been established to ensure link layer security [9], [10], and the strength of the cryptographic primitives is dependent on the strength of the utilized keys. This raises the need for a strong, simple and scalable key exchange system due to the large size of AMI networks. Particularly, the usage of public key infrastructures (PKIs) along with certificates and Certificate Authorities (CAs) is discouraged in the context of smart grids due to challenges in scalability and complexity [11], [12].

In [8], we propose a secure and scalable identity based key distribution framework for the link layer security of wireless mesh-based AMI. It employs Identity based Non-Interactive Key Distribution (ID-NIKD) [13] which can be

Manuscript received December 22, 2015; revised May 6, 2016 and September 4, 2016; accepted October 23, 2016. Date of publication November 11, 2016; date of current version June 19, 2018. This work was supported by the Lebanese National Council for Scientific Research. Paper no. TSG-01622-2015.

The authors are with the Department of Electrical and Computer Engineering, American University of Beirut, Beirut 1107 2020, Lebanon (e-mail: vss01@aub.edu.lb; rouwaida.kanj@aub.edu.lb; chehab@aub.edu.lb; ayman@aub.edu.lb).

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.2016.2628090

1949-3053 © 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.

See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

seen as a prominent solution to the problems faced. The proposed methodology enables the following key features.

- 1) Secure and scalable identity-based non-interactive key distribution framework between neighboring smart meters. The proposed framework provides secure symmetric keys to neighboring smart meters without having the need to exchange extra packets for key negotiation between smart meters. The derived keys between the neighboring meters are utilized for the security of the link layer.
- 2) Secure and leakage-free storage of the cryptographic keys. These include private keys provided by the utility to the smart meters and derived symmetric keys between neighboring meters. Storing the symmetric keys helps avoid their recomputation overhead. Cryptographic keys are protected by encrypting them with volatile keys generated from physical unclonable functions (PUFs). Such framework requires periodic private key updates to provide fresh keys for the meters and render cryptographic attacks even more difficult. For this purpose, we propose a novel lightweight update and delivery of the private keys utilized by the identity-based non-interactive key distribution framework. Our contributions are as follows.
- 3) We propose an efficient algorithm that relies on bit shuffling to ensure the authenticity of the updated keys. We then analyze the security of the proposed algorithm.
- 4) We propose a delivery process that exploits multicasting capabilities of the network and properties of the novel authentication mechanism and sends packets containing aggregated list of keys. This allows us to reduce the size of the key update packets as they traverse the network, and therefore reduce the network latency and traffic overhead.
- 5) We evaluate the key authentication algorithm on an ARM cortex M3-based microcontroller and perform network simulations on NS3, a discrete event simulator, to assess the proposed methodology.

We revise our contributions in [8] in Sections II–IV. We present the novel key update methodology along with relevant simulations and analysis in Sections IV-B, V, and VI. The rest of the paper is organized as follows. Section II presents a background review. Sections III and IV describe the proposed identity-based framework. Section V presents the authenticated key delivery and update mechanisms. Section VI presents simulations of the key delivery mechanism along with hardware testing of the authentication mechanism. Finally, Section VII concludes the paper.

II. PRECURSOR

A. Smart Grid Secure Communications and Scalable Key Generation Frameworks

The Modern Grid Initiative has identified integrated communications as one of the top five emerging technology areas for smart grids [14]. Two-way and real-time communications between the consumers and providers enables a reliable and

an efficient smart grid system. Furthermore, real-time communications allows for faster grid self-recovery and enables coordination and control among the different sources of energy.

However, not only does the integration of the smart grid elements require a proper communications infrastructure [15], it also requires that security be embedded into the infrastructure from the beginning [16]. According to [17], the power sector is subject to an increasing potential for cyber-attacks as it becomes more interconnected. In fact, “cyber systems are the weakest link” in the electric grid and threats are growing at the same pace as the technological advances [16]. If we add to this the human factor of error, the challenges of the cyber physical are becoming more complex. Cyber security must address both deliberate attacks as well as accidental attacks such as user errors, equipment failures, and natural disasters. It is possible for the attacks to “destabilize the grid in unpredictable ways” [17], [18].

Compared to the Internet, the smart grid has tighter security requirements [19]. Standards bodies such as NIST are working to ensure the reliability, integrity and confidentiality of the smart grid system as sensitive and private user data is being transmitted over the grid and as the need for real-time control grows. Metke and Ekl [18] state that grid security highly depends on the underlying authentication, authorization, and privacy technologies. The large number of meters in the network further leads to scalability issues, and Metke and Ekl [18] focus on the need for scalable security key generation frameworks such as Public Key Infrastructures. However, PKIs can be 1) *complex to manage* [11] and 2) *cryptographic keys stored in hardware require management to prevent their discovery*. In this work, we rely on identity-based cryptography whereby public information is used as public keys for the crypto-system. This reduces the complexity and the burden of installing and managing Public-Key Infrastructures (PKIs) as is the case in certificate-based crypto-systems. We also rely on PUFs for secure key storage and reduced computational overhead.

B. Wireless Mesh-Based AMI

AMI enable two-way communications between the utility and the customers, and smart meters are at the core of the AMI. Figure 1 illustrates a snapshot of the organization of an AMI [8]. It encompasses three main networks: The Home Area Network (HAN), the Neighborhood Area Network (NAN) and the Wide Area Network (WAN). The HAN covers the communication between smart appliances at home and the smart meter. Each meter acts as a relay and forwards its own data as well as data from neighboring meters. The set of neighboring meters form a Neighborhood Area Network (NAN) and they forward the data to the aggregator that connects the NAN to the WAN and the utility. Using wireless mesh-based networks at the NAN level enables the deployment of large-scale and reliable AMI networks [1] without the need for an expensive communication infrastructure.

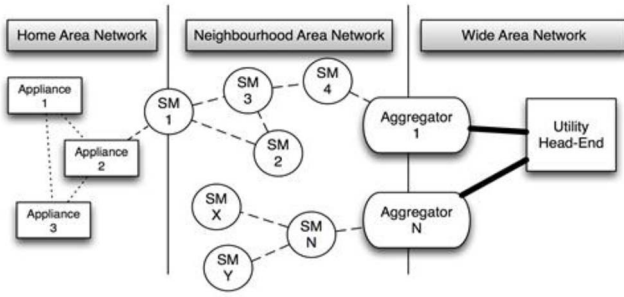


Fig. 1. Simplified organization of an AMI [8].

C. ID-Based Cryptography & Bilinear Maps

Identity based cryptography is a type of asymmetric cryptography where publicly known information such as IP addresses or telephone numbers can be used as public keys. Identity based cryptosystems reduces the complexity of a system by removing the burden of managing and installing PKIs. Private keys are generated by a trusted entity known as the Private Key Generator (PKG).

The concept of ID-based cryptography was introduced by Shamir [20] in 1984. Recently, Boneh and Franklin [21] devised an identity-based encryption based on admissible bilinear pairings. A bilinear map is defined as follows.

$$\hat{e} : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T \quad (1)$$

\mathbb{G} is an additive subgroup of points on an elliptic-curve E/\mathbb{F}_n where \mathbb{F}_n is a finite field of size n , and \mathbb{G}_T is a multiplicative subgroup of a finite field. Both \mathbb{G} and \mathbb{G}_T are groups of prime order n . Pairing algorithms such as the Weil and the Tate pairings on elliptic curves can be used to realize such bilinear maps [21]. Admissible bilinear maps have 3 useful properties.

1. Bilinearity:

$$\begin{aligned} \forall P, Q \in \mathbb{G}, \forall a, b \in \mathbb{Z}_n^*, \\ \hat{e}(aP, bQ) &= \hat{e}(P, Q)^{ab} \\ \mathbb{Z}_n^* &= \mathbb{Z}_n \setminus \{0\}, O \text{ being the identity element} \end{aligned} \quad (2)$$

2. *Non-Degeneracy*: Map doesn't send all pairs $\mathbb{G} \times \mathbb{G}$ to the identity in \mathbb{G}_T :

$$\begin{aligned} \forall P \in \mathbb{G}_1, P \neq 0 \Rightarrow \hat{e}(P, Q) &= \mathbb{G}_T, \\ \text{Implies that } \hat{e}(P, P) &\text{ generates } \mathbb{G}_T \end{aligned} \quad (3)$$

3. *Computable*: There exists an efficient algorithm that computes $\hat{e}(P, Q)$ for $\forall P, Q \in \mathbb{G}$.

For more details on operations in mathematical cryptography we refer the reader to [22].

Pairings require several complex arithmetic operations and are computationally expensive [23]. Significant work has been done to speedup pairing computations on PCs [24], and more recently works aim at speeding up pairings on ARM processors for handheld devices [25], [26].

D. Physical Unclonable Functions

Physical Unclonable Functions (PUFs) can be used to increase physical security by exploiting the fact that different

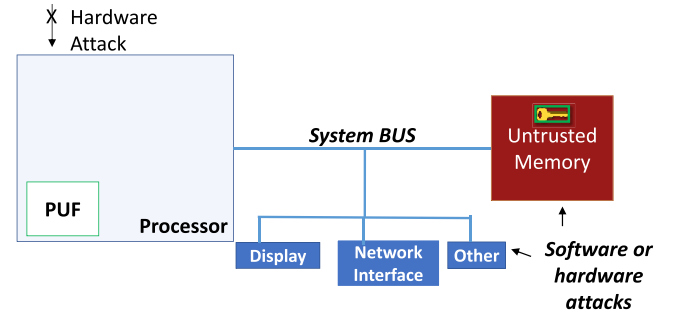


Fig. 2. The PUF is integrated within a secure processor.

instances of the same hardware result in different behavioural characteristics. This is due to the random and uncontrolled discrepancies that result at the circuit production stage [27]. PUFs are one-way functions where a challenge C_i produces a unique response R_i seen as the volatile secret key. Several PUF designs have been developed, and the ring oscillator PUF is popular due to its ease of implementation and reliability [27] at the expense of area and power. Since PUFs are determined by variations in the hardware, the volatile secret key is only derived upon circuit execution and cannot be derived based on assumptions of the function itself. According to [27], to carry a successful attack the adversary has to run the attack while the PUF circuitry is powered, creating a much more challenging scheme than the discovery of non-volatile secret keys. In addition, for the attack to be successful, she has to measure the PUF delay without altering the circuit delays and/or discover keys within registers without removing power from them. In our work, we adopt the secure processor model in [28]. The PUF is integrated within the trusted processor (Figure 2), and the processor's internal states cannot be observed directly by physical means. We assume that all the components outside the processor are insecure and prone to software or hardware attacks. We also assume that processor is equipped with methods to prevent side-channel attacks [28]. We rely on the PUF circuitry for the following two reasons.

- 1) The 1st is to provide volatile keys, R_i , generated from the PUF, which are not stored in any addressable memory. The generated keys will only depend on C_i . Although C_i are publicly accessible, the response R_i is not accessible across the bus, preventing the attacker from obtaining it.
- 2) The 2nd is to provide secure cryptographic storage to private and secret keys that are generated by the proposed methodology and stored in the meter's publicly accessible memory. This is achieved by encrypting the secret keys using the PUF response as will be explained later.

Since PUFs are implemented in hardware, their latency overhead is typically small compared to the execution of cryptographic software blocks. For purposes of our analysis, we do not account for the PUF overhead as it is in most cases common to the different experiments and/or very small.

III. PROPOSED FRAMEWORK

A. Attacker Model

The attacker is assumed to be present in the network after the installation of the smart meters in the field. She is assumed

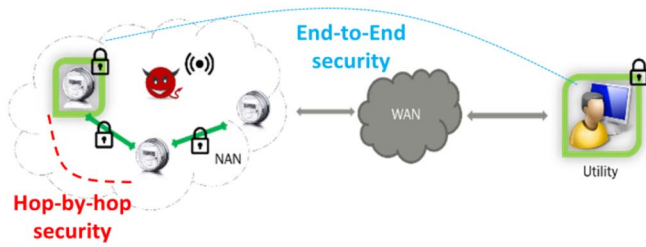


Fig. 3. End-to-end and hop-by-hop security.

to have a wireless device that has the same communication protocols used by regular smart meters. As stated in the introduction, the attacker can: Sniff packets passing through the wireless medium, inject new packets into the network to send information or commands to the meters or to the utility, inject new bogus packets into the network to flood the network and cause a DoS attack, and run various layer 2 attacks. Furthermore, the attacker may try to tamper with the hardware of the smart meter in order to retrieve secret keys used by different cryptographic functions.

B. Framework Overview

To guard against such an attacker model, the network relies on a hop-by-hop and an end-to-end encryption (authentication) mechanism as shown in Figure 3. The security roles for each entity can be described as follows.

- 1) *Utility*: It represents the communication end-point for all smart meters and is the Private Key Generator (PKG) of the ID-based cryptographic system. It is responsible for storing challenge/response pairs of the smart meters' PUF devices, and generating and transporting encrypted private keys for the smart meters during the setup/update phase.
- 2) *Smart meter*: It sends/receives messages to/from the utility while implementing end-to-end security. It acts as a relay to forward messages of its direct neighbors. It implements both application layer security and link layer authentication. The keys utilized in the application layer are based on keys derived from PUFs. The link layer authentication requires the generation of pairwise symmetric keys between the meter and its neighbors.

To generate the pairwise symmetric keys, we propose an ID-based non-interactive key distribution (ID-NIKD) scheme. By having an ID based cryptosystem and a non-interactive key distribution scheme, we ensure the scalability of the proposed scheme along with reducing the network overhead incurred by the key exchange. In contrast to PKI and certificate based cryptosystems, the proposed scheme eliminates the communication overhead associated with certificate exchange and reduces the system complexity by removing the burden of managing and installing a PKI. All generated and stored keys inside the meters will be encrypted using the PUF responses of the meter to guard against key leakage at the hardware level [8]. Furthermore, the pair-wise symmetric key generation helps to circumvent total network compromise that may arise when a single shared key is used for the whole system.

TABLE I
SIMULATED SCENARIOS TO ANALYZE DoS ATTACK [8]

Scenario number	Attacker present	Authentication present
I	×	×
II	×	√
III	√	×
IV	√	√

TABLE II
PACKET DROP FOR SCENARIO III UNDER DoS ATTACK.
NO DROP RECORDED FOR SCENARIO IV [8]

Exp #1	Link Speed (Kbps)	100	200	300	400
	% Dropped Packets	6.52	6.72	5.48	6.02
Exp #2	Packet Size (Bytes)	200	300	400	500
	% Dropped Packets	0.36	6.72	16.84	23.22
Exp #3	# Hops	6	10	12	14
	% Dropped Packets	6.53	6.72	8.13	10.70

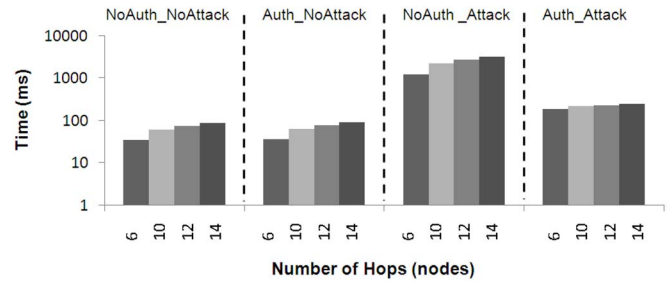


Fig. 4. Average latency for Experiment #3 [8].

C. A Wireless Mesh Network UnderAttack

As packets traverse the network, the link layer authentication mechanism mitigates the effects of DoS attacks that are caused by flooding the network with bogus packets [8]. This is achieved by discarding bogus packets at the point of detection and preventing them from further traversing the mesh network. Such flooding attacks would otherwise consume the energy and computational resources of all on-transit meters that carry packets to the utility. In addition, they consume network bandwidth and congest the queues of the on-transit meters. This can lead to increased packet latency and potential packet loss. In [8], we study the impact of a flood attack with and without authentication in the context of 802.15.4g, which is a protocol designed for Smart Utility Networks. We analyze four scenarios that include or exclude hop-by-hop symmetric authentication in the presence of an attack (Table I). We assume that all neighboring meters have pre-calculated the pairwise symmetric keys before running the simulations. It is found that, for the different network parameters, the lack of authentication can lead to packet drops as seen in Table II, and significant increase in latency. Packet drop happens when the smart meter queue is full due to flooding. No packet drop is encountered in authenticated approaches. This comes at a slight increase in latency (Fig. 4). The results of the simulations protrude the importance of link layer security to mitigate the effects of a DoS attack.

TABLE III
SETUP ALGORITHM OF PRE-DEPLOYMENT PHASE

Setup	
1.	Execute $\mathcal{G}(1^k)$ to obtain $\{q, \langle \hat{e} \rangle, \langle \mathbb{G} \rangle, \langle \mathbb{G}_T \rangle\}$.
2.	Let $P \in \mathbb{G}_1$ be a random generator. Randomly select $s \in \mathbb{Z}_q^*$ the PKG's secret key
3.	Choose a cryptographic hash function $H: \{0,1\}^* \rightarrow \mathbb{G}$
4.	Set $\{P, H, \langle \hat{e} \rangle, \langle \mathbb{G} \rangle, \langle \mathbb{G}_T \rangle\}$ as the public parameters of the cryptosystems.

IV. PROPOSED FRAMEWORK IMPLEMENTATION

In this section, we discuss the pre-deployment phase, private key generation, and private key update. We also discuss the non-interactive pairwise key generation between two neighboring meters, which relies on pairings involving the private keys and public ID. We present the methodology block diagram in Section IV-D, and highlight packet encryption/authentication at the application and data link layers in Section IV-E.

A. Pre-Deployment Phase

The pre-deployment phase is when the meters are still available at the utility's vicinity prior to deploying them in the field. At this stage, it is assumed that the utility has physical access to the meters. The first step is to input a challenge C_i (i is device index) to the PUF of the smart meter and obtain a response R_i . The utility then securely stores the value of the response R_i in its database. This operation is repeated for all meters that will be deployed in the fields. The second step involves setting up the ID-based cryptosystem and generating the private keys for the meters, referred to as the **Setup** and the **Keygen** algorithms, respectively. Let $k \in \mathbb{Z}^+$ be the security parameter given to the setup algorithm. Let $\mathcal{G}(1^k)$ be a BDH parameter generator having a polynomial runtime in k and satisfying the BDH assumption [21], [22].

Setup Process: The utility acts as a private key generator (PKG). Utility runs the function $\mathcal{G}(1^k)$; takes k and outputs a random k -bit prime number q , description of the groups \mathbb{G} and \mathbb{G}_T and the description of a bilinear map $\hat{e}: \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$. The descriptions $\langle \mathbb{G} \rangle$ and $\langle \mathbb{G}_T \rangle$ contain algorithms for computing the group action in \mathbb{G}/\mathbb{G}_T and contain a generator of \mathbb{G}/\mathbb{G}_T . On the other hand, the description $\langle \hat{e} \rangle$ contains a polynomial time algorithm for computing \hat{e} . It also defines the hash function $H: \{0,1\}^* \rightarrow \mathbb{G}$. The pairing \hat{e} is bilinear, non-degenerate and efficiently computable. The PKG then selects a master key $s \in \mathbb{Z}_q^*$ [29]. The steps of the algorithm are summarized in Table III.

Keygen Process: After generating its master key s , the PKG generates the meter's public and private keys defined by P_x and pvt_x . The steps of the algorithm are summarized in Table IV. ID_x is an identity related to the meter; it can be any publicly known information about the meter. In our case, it is the network interface's hardware address (known as MAC address). We assume that there is a neighbor discovery protocol utilized by the meters from which the meters would acquire the MAC addresses of their neighboring meters. After executing

TABLE IV
KEYGEN ALGORITHM OF PRE-DEPLOYMENT PHASE

Keygen	
1.	For every meter x having an identity denoted by ID_x , calculate $P_x = H(ID_x)$
2.	Generate smart meter x 's private key $pvt_x = sP_x$ where s is the PKG's secret key

the **Setup** and **Keygen** algorithms, the utility stores $e_{R_i}(pvt_x)$, the encrypted version of pvt_x in the meter's memory using the response obtained from the PUF as the symmetric key for the encryption. Other than storing pvt_x , the utility stores the challenge C_i and the public parameters $\{P, H, \langle \hat{e} \rangle, \langle \mathbb{G} \rangle, \langle \mathbb{G}_T \rangle\}$ of the cryptosystem generated by \mathcal{G} during the **Setup** phase. At a later stage, when the meter wants to retrieve pvt_x , it executes $PUF(C_i) \rightarrow R_i$ and uses its outcome to decrypt the encrypted private key stored within its memory in the following manner.

$$d_{R_i}(e_{R_i}(Pvt_x)) = pvt_x. \quad (4)$$

B. Private Key Update Process

The key update process can be seen as a variant of the **Keygen** process. By initiating a key update, the utility generates a new set of private keys for the concerned smart meters. In addition to updating keys, the proposed key update process is intended to achieve identity revocation. This revocation is achieved by merely excluding identities from the rest of the group that the utility wishes to grant new keys. The proposed time-correlated key update process is motivated by the work in [21]. The key update procedure relies on the Real Time Clocks (RTCs) embedded within meters' microcontrollers providing precise timing information. This assumption is realistic since all vendors that produce microcontrollers used in smart meter applications embed RTCs. The proposed key update mechanism differs from the **Keygen** process in the definition of P_x , such that:

$$P'_x = H(ID_x || \tau) \quad (5)$$

The function $H()$ and ID_x were previously defined. The symbol $||$ denotes the concatenation operation. τ is an argument that represent timing information in epoch format. It denotes the beginning of a time frame such of a day, week, month, etc. It is generated based on 2 factors: 1) Wall clock time and 2) φ , an argument that represents a timing information indicating how frequent the utility wants to update the keys (daily, weekly, etc.). A higher frequency of key update indicates better security and reduces the effects of a compromised key. The update process at the utility's side is summarized in Table V. For this scheme to work, smart meters will have to use $ID_x || \tau$ as the public key of the neighboring meter instead of only using ID_x . τ is generated inside smart meters by relying on 2 values; the first is the time information obtained from the RTC that denotes wall clock time. The second is based on the frequency of the key update dictated by the utility and that can be stored inside

TABLE V
UPDATE ALGORITHM OF PRE-DEPLOYMENT PHASE

Update	
1.	For every meter x having an identity denoted by ID_x , calculate $P_x = H(ID_x \tau)$
2.	Generate smart meter x 's private key $pvt_x = sP_x$ where s is the PKG's secret key

the meter during the **Setup** phase. The utility has to store $\{\varphi, P, H, \langle \hat{e} \rangle, \langle G \rangle, \langle G_T \rangle\}$. By following such a scheme, smart meters that haven't obtained new keys for a certain date span will not be able to communicate with the neighboring meters, thus automatically having their identities revoked.

C. ID-NIKD and Pairwise Symmetric Key Generation

Consider two neighboring smart meters A and B with unique IDs, ID_A and ID_B . The meters, having private keys pvt_A and pvt_B and by knowing the ID of the other meter, are able to derive a pairwise symmetric key without exchanging any data. This is done using ID-based non-interactive key distribution scheme [12]. Meter A derives the symmetric key $SK_{A,B}$ using pvt_A and ID_B as follows.

$$\begin{aligned} \hat{e}(pvt_A, H(ID_B)) &= \hat{e}(sH(ID_A), H(ID_B)) \\ &= \hat{e}(H(ID_A), H(ID_B))^s = SK_{A,B} \end{aligned} \quad (6)$$

Meter B derives the symmetric key $SK_{B,A}$ using pvt_B and ID_A as follows.

$$\begin{aligned} \hat{e}(pvt_B, H(ID_A)) &= \hat{e}(sH(ID_B), H(ID_A)) \\ &= \hat{e}(H(ID_B), H(ID_A))^s = SK_{B,A} \end{aligned} \quad (7)$$

Note that \hat{e} is the pairing function described in the **Setup** process. It is clear from equations (3) and (4) that $SK_{A,B} = SK_{B,A}$. The resultant symmetric key is then post-processed with a key derivation function to make it compatible with the used cryptographic functions [29]. Although pairings are computationally expensive and can take few milliseconds [23]–[26], this step is only performed when a new neighbor is discovered or a new key is obtained. Once the pairwise symmetric key is derived, it is stored securely in the device memory. Hence, it is encrypted using the PUF response as the key (see Figure 5) and is not stored in clear text. If the two meters need to communicate again, they only need to decrypt the key that is securely stored locally. We would like to note that the ID-based key exchange was chosen because of the following properties.

- 1) It has the minimum overhead in terms of key exchange between meters. Because of the non-interactive property of the proposed mechanism, meters do not explicitly exchange packets for purpose of the key exchange.
- 2) By relying on identity-based cryptosystem, the mechanism is highly scalable in terms of key management and exchange.

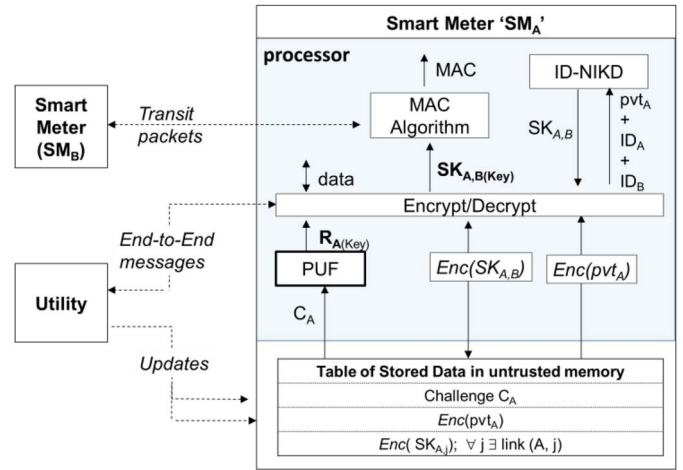


Fig. 5. Methodology block diagram [8]. PUF block integrated within the secure processor as illustrated in Figure 1. Cryptographic blocks are implemented in software. Data stored in external memory is encrypted.

D. Methodology in a Nutshell

Figure 5 presents the proposed methodology block diagram along with the different key generation, authentication and encryption blocks. Bolded variables represent keys used for the cryptographic blocks. Each meter stores 1) the challenge for its PUF, 2) encrypted private key, which is an input for the ID-NIKD scheme, and 3) the encryption of the pair-wise symmetric keys it shares with its direct neighboring meters. The response of the PUF is used to encrypt/decrypt private and pairwise symmetric keys. The encryption can be based on AES-128, and the message authentication code (MAC) algorithm can be based on any MAC generating algorithm such as HMAC-SHA or CBC-MAC.

E. Field Phase and Packet Overview

Figure 6 illustrates the general packet format that traverses the network of Figure 3. At the application layer, meter-to-utility communication is based on the ANSI C12.22 protocol [30] for meter data communication. The messages consist of various nested elements: Association Control Service Element (ACSE), user-information element and one or more Extended Protocol Specifications for Electric Metering (EPSEM) that carry data and ANSI C12.19 tables [30]. Data can be sent authenticated, encrypted or both. Optionally, the response R_x generated by feeding challenge C_x into the PUF can be used to provide a symmetric key for the authentication and encryption blocks used for meter to utility communication at the application layer.

At the data-link layer, where either 802.15.4g [31] or the alternative wireless mesh network protocol 802.11s [32] is used, the pairwise keys generated are supplied to the security mechanism in order to generate a MAC as illustrated in Figure 7. In what follows, we will assume HMAC-SHA-256 is used as the MAC generating algorithm. The authentication scheme utilizes the pairwise symmetric keys generated by the ID-NIKD scheme. A meter 'B' receiving a packet from 'A' will forward it to 'C' only if the computed message authentication

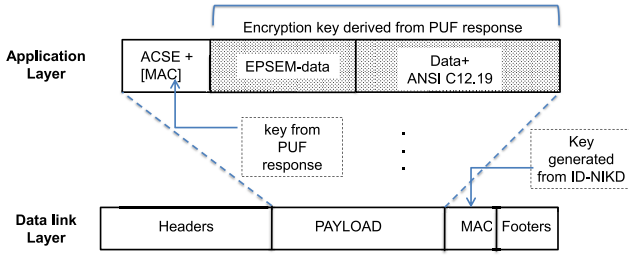


Fig. 6. Packet overview [8]. Keys generated from the PUF can be used for authentication and encryption for end-to-end communication. Keys generated from ID-NIKD are used for data link layer authentication.

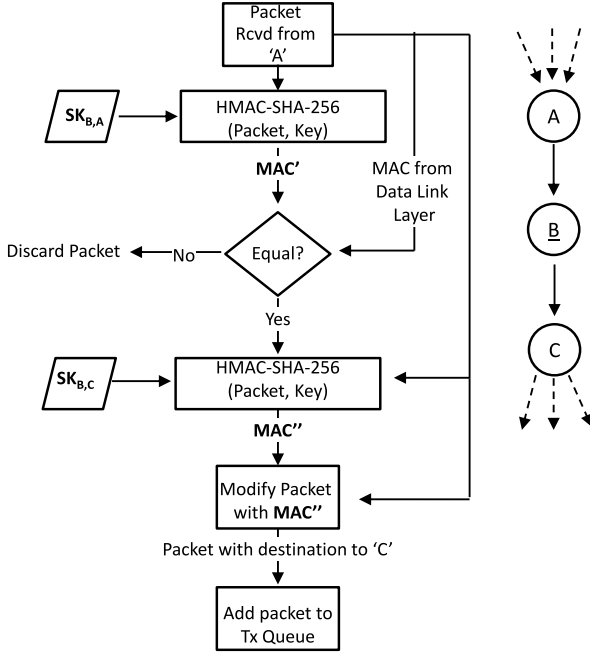


Fig. 7. Authentication mechanism at meter 'B' [8].

code, $MAC' = \text{HMAC-SHA-256}(\text{Packet}, SK_{B,A})$, matches the one received in the security header of the data link layer. If both MACs match, meter 'B' replaces the existing MAC with $MAC'' = \text{HMAC-SHA-256}(\text{Packet}, SK_{B,C})$ and forwards the updated packet to 'C'.

V. PRIVATE KEY DELIVERY MECHANISM AND SHUFFLED-ID BASED SECURITY SCHEME

The Key update and delivery mechanisms play crucial roles in the key distribution framework. Once smart meters are deployed and running, they need to obtain new keys for the different security schemes. For our application, the smart meters need to obtain new private keys for the ID-NIKD scheme. The frequency of the key update has an implication on security and on the effects of a compromised key. A higher frequency of key update indicates better security and reduces the effects of a compromised key.

For the key delivery mechanism, there are two security aspects that should be employed: encryption and authentication of the delivered private keys.

- Encryption ensures that the delivered keys are properly revealed only to the destined end parties. That is,

TABLE VI
NIST RECOMMENDED KEY SIZES AS LISTED IN [34]

Symmetric (bits)	RSA (bits)	ECC (bits)
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	521

no eavesdropper on the delivery path can obtain any data regarding the transmitted private key on route.

- Authentication allows the destination party to verify that the received key is truly generated by the utility. In the absence of authentication, an attacker might be able to generate a forged encrypted key (without necessarily knowing its content), forcing a smart meter to accept the key as a genuine key and thereby block its access to the network.

Typically, authenticated encryption is performed by having an encrypted block of data prepended next to a message authentication code (MAC). To achieve high level of security and to ensure very low collision probabilities, 16 bytes of MAC size is recommended. In the following subsections, we propose an efficient authenticated encryption methodology with lower packet overhead compared to typical authentication techniques. For purposes of our proposed private key delivery mechanism, we focus on the following aspects:

- 1) Security and robustness of the key delivery mechanism
- 2) Latency and network traffic reduction.

A. Precursor

The Advanced Encryption Standard (AES) will be used for encrypting the private keys and it will be used as a pseudo random permutation block (PRP) [33] for the authentication of the private keys. Since the ID-NIKD is a mechanism based on Elliptic Curve Crypto (ECC), we assume it uses one of the ECC key sizes recommended by NIST [34]. An equivalence table between the different cryptographic systems can be observed in Table VI.

B. Key Delivery and Authentication Mechanism

In order to reduce the latency and the overhead placed on the network during the delivery of the newly generated keys, the proposed mechanism exploits the multicasting feature (Figure 8) of the network and sends the keys in an aggregated fashion without degrading the security of the delivered keys.

The mechanism is designed in a manner that a smart meter is able to extract its own private key from the aggregated list of keys within the multicast packet and to forward the rest of the data without affecting the authenticity of the packet. This helps in the reduction of the size of the packet as it traverses the mesh network. The proposed delivery form reduces the traffic generated in the wireless mesh network compared to a regular unicast key delivery form. The headers used in the different OSI layers (link layer, network and transport) will be shared, thus reducing the overhead ratio. In order to be able to gain the capability of removing chunks of bytes from the packet as it traverses through the network, authentication has

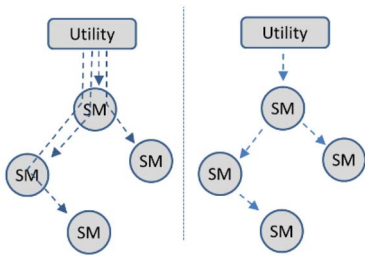


Fig. 8. Unicast key delivery (left) vs. Multicast (right).

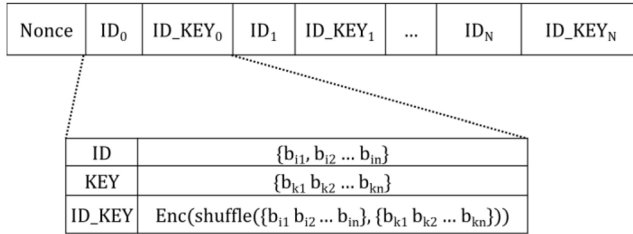


Fig. 9. Application layer key delivery packet.

to be performed on each individual key. A simple approach would be to place an encrypted private key and a message authentication code for each key. However, a secure MAC would have sizes not less than 16 bytes. This increases the size of the packet and decreases the space left for the private keys to be stored.

Figure 9 depicts how the proposed key delivery packet looks like at the application layer. The *Nonce* is a random number generated by the utility to add entropy into the system and is used to protect against replay attacks. This *Nonce* is employed in the encryption and authentication blocks. ID_N is a number that represents the identity of a smart meter. ID_Key_N is an encrypted version of both the private key and the shuffled bits of ID_N as illustrated in Figure 9. It is used also for the authentication of the private keys as will be explained in the following subsection. Figure 10 shows the average number of keys that can be stored in the application layer (y-axis) using the configuration specified in Section VI-A, versus the size of the keys used shown in Table VI. A *Nonce* of 4 bytes is assumed to be used along with an ID of size 4 bytes as it can cover ~ 4 billion smart meters.

C. Security of Key Distribution Mechanism

As mentioned earlier, the proposed algorithm adds 4 bytes and achieves an equivalent level of security as compared to adding 16 bytes of MAC next to each private key. The algorithm achieves this by embedding the randomly shuffled bits of ID_i at certain deterministic positions along with the private key within ID_KEY_i . The process relies on shared keys associated between a meter and the utility. This shared key can be generated by utilizing the PUF hardware as described in the previous sections or by using ID-NIKD as a block to generate a shared key between the utility and the smart meter.

Figure 11 presents the key authenticated encryption algorithm that runs at the utility.

- In step 1, the utility starts by generating a *Nonce* of size 4 bytes.

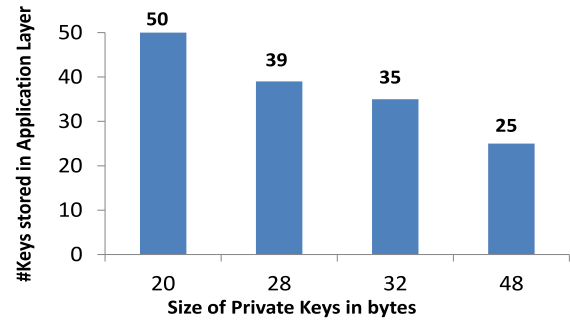


Fig. 10. Number of stored keys vs. key size (in bytes).

- In step 2, it creates a unique private key to be delivered for each ID_i in the sub-network as described in the key update section. The utility allocates, for each key, T bits of memory in the ID_KEY buffer. T represents the number of bits allocated for one ID and one private key. This accounts for integrating the ID along the private key to be fed to the AES block used for authentication.
- In steps 4 and 5, the utility uses ID_i and the *Nonce* along with a shared secret key between the utility and the meter of identity ID_i to generate S_1 as a seed for the Pseudo Random Generator (PRG).
- The utility then proceeds according to step 7 and generates the index ID_bit_pos for each of the bits of ID_i and places those bits into the $ID_bit_pos^{th}$ position of ID_KEY buffer, respectively. After filling the ID_i bits in the designated positions, the utility fills the rest of the ID_KEY buffer with the unencrypted version of the private key KEY_i .
- The final step is to encrypt ID_KEY buffer with a shared key. ID_KEY_i will be the input plaintext to the AES block.

Finally, it is assumed that AES is being run in a mode that doesn't require padding (such as AES-OFB or AES-CTR) in order to keep the output size minimal.

By exploiting the fact that AES is a PRP block, the output from the AES block (Figure 12) will produce a pseudo random number of size 24 bytes, which cannot be distinguished from a totally random number by an adversary [33]. After this step, ID_KEY buffer contains the authenticated and encrypted version of KEY_i . When a smart meter carrying ID_i receives an update packet, it follows the same steps of generating ID_KEY_i but in reverse order. The smart meter accepts the key only if ID_i bits in ID_KEY_i have the correct values in their correct positions after decryption. Note that the meter is able to generate the same seed S_1 for the PRG locally, and hence the ID_bit_pos position values.

To assess the strength of the authentication mechanism, we calculate the probability of collision. We assume the same attacker model presented in Section III-A. The attacker might try to randomly generate 24 bytes of data with the goal of forcing the smart meter to accept a new fake private key. Although the attacker will not know the value of the key, however, she can prevent the smart meter from further communicating with other neighbors if she forges a bit string in a such a way that, after decryption, the ID bits and bit positions match the

```

0 Generate Nonce
1 ForEach IDi:
2   Generate KEYi
3   ID_KEY_BUFFER = allocate_ T_bits()
4   Generate S1 := HMACshared_KEY (IDi, Nonce)
5   PRG_seed = S1
6   for count:= 1 to T:
7     ID_bit_pos = PRG() mod T
8     ID_KEY_BUFFER [ID_bit_pos]= IDi
9     [count]
9 Fill rest of ID_KEY_BUFFER with KEYi
10 Encrypt ID_KEY_BUFFER with shared key

```

Fig. 11. ID_Key Generation algorithm at the utility.

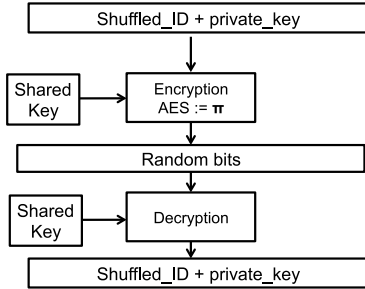


Fig. 12. AES encryption of private key and shuffled ID.

calculated ones. The probability of a successful attack is:

$$\begin{aligned}
 & Pr(\text{correct ID values}) \times Pr(\text{correct ID locations}) \\
 &= \frac{1}{2^N} \times \frac{(T-N)!}{T!} \times \frac{N!}{1} \quad (8)
 \end{aligned}$$

where T is the ID_Key size in bits and N is the ID size in bits. Figure 13 plots the probability of collision for different ID and Key sizes.

As the key size increases, T increases and reduces the collision rate. The size of the ID is typically a choice of the network provider. For purposes of our application, a 4-byte ID size is reasonable enough as it allows for 4 billion users as mentioned earlier. Twenty bytes is the smallest key size recommended by NIST, and for purposes of our proposed methodology, smaller keys are preferred as this allows more data to be packed into one packet. It can be seen from the graph that for a key size of 20 bytes and ID size of 4 bytes, the probability of collision is 2^{-153} ; hence, this combination is suitable from both security and compactness perspectives.

VI. SIMULATION ANALYSIS AND RESULTS

In this section, we demonstrate the low latency overhead of the key authentication algorithm on a fast wireless mesh network (802.11s). We also evaluate the effectiveness of the multicast key delivery mechanism. First, we perform hardware testing to evaluate the computational overhead of the proposed key authentication algorithm placed on the microcontroller of a smart meter.

For all our measurements and simulations, the ID_Key combination is based on a 20-byte key and a 4-byte ID. We assume an IPv4 network along with UDP at the transport layer having a maximum transfer unit (MTU) of 1500 bytes.

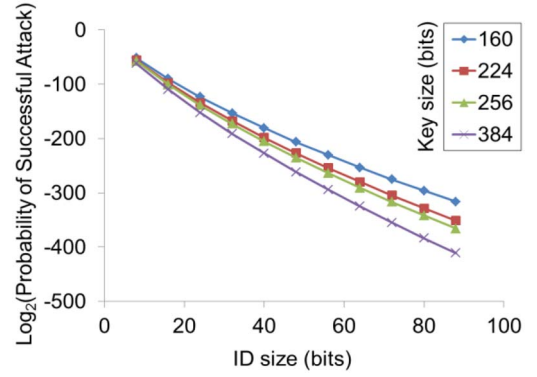


Fig. 13. Probability of a successful attack vs. key and ID sizes.

TABLE VII
REAL TIME DEBUGGER DATA

	AES128 Init.	AES128 Decrypt.	HMAC- SHA1 Init.	HMAC- SHA1 Gen.	DRNG	Total
Clock cycles	157	481	1252	1498	961	4349
Latency (us)	1.3	4.0	10.4	12.5	8.0	36.3

A. Hardware Testing

Many of the smart meter vendors use microcontrollers based on the Cortex M3 architecture [35]. These processors offer a low power solution and help cut down on the cost and number of smart meter parts. To assess the computational overhead of the key authentication algorithm described in Section IV, we rely on the STM32F217IGH microcontroller [36] (Figure 14); it is based on the ARM Cortex M3 architecture and is chosen in order to mimic smart meters. Additionally, many of the STM32F2x family of microcontrollers are equipped with cryptographic processors, hash processors and random number generators. The cryptographic processor implements AES, DES and 3-DES in ECB, CBC and CTR modes. The hash processor implements SHA-1, MD5 and HMAC-SHA-1-MD5. The random number generator (RNG) is a true random generator that generates random bits based on continuous noise signal.

We developed the code using the Keil MDK5 environment, compiled with time optimization. It is a software development solution to create, build, and debug embedded applications [37]. We also relied on the wolfSSL/CyaSSL [38] embedded library as the cryptographic library. To reduce the runtime, we relied on the built-in cryptographic and hash libraries to implement the cryptographic blocks. However, we did not use the true RNG hardware block because a deterministic RNG is needed to produce identical results on both ends (the utility and user). The input and key sizes used in the simulations are based on real scenarios. The HMAC-SHA1 called in step 4 of the pseudo-code was run using a key size of 20 bytes and an input of 4 bytes representing the ID. AES was run using a key size of 128 bits with an initialization vector (IV) of 4 bytes in order to encrypt the ID_key in step 10 of the pseudo code. The deterministic RNG was based on the RC4 stream cipher where the output was 32 bytes to represent the locations of the shuffled ID bits. Table VII presents

the algorithm runtime data along with individual clock cycles as obtained from the real time in circuit debugger. This indicates the total cost required by the meter to process its own ID_Key. It is clear that the added computational latency of the proposed algorithm is small and acceptable. This will be further emphasized in the network simulation results presented in the following section.

B. Network Simulations

In this section, we assess the efficiency of the authenticated key delivery mechanism. The goal is to evaluate the advantage of the reduced size authentication along with the multicast based approach, while accounting for the key authentication computational overhead.

1) *Roles and Scenarios*: To assess these advantages, we compare the following key delivery scenarios in our simulations. It is assumed that the utility is the private key generator and is responsible for generating the encrypted and authenticated packets that contain the keys and the relevant information needed to securely achieve the key update process. In all scenarios, the aggregator is the root of the NAN, and the rest of the nodes represent the smart meters. The smart meters will accept and forward key update packets after authenticating their respective keys and making sure they are generated by the utility.

- 1) *Unicast (SC1)*: Represents the key authentication mechanism with one key per packet delivery.
- 2) *Multicast noDrop (SC2)*: Represents the multicast key distribution without dropping any chunk of data of the packet en route. In this scenario, a node extracts its key and forwards the packet as is to its children without modifying the packet. *This highlights the advantage of multicast-based key update over unicast.*
- 3) *Multicast withDrop (proposed scenario, SC3)*: Represents the proposed multicast key distribution whereby a node extracts its key and then forwards to its children only the set of keys intended for their respective sub-trees. We assume that routing information is available from the routing protocols. *This highlights the advantage of packet drop in multicast-based key update.*
- 4) *Multicast withDrop Delay (proposed scenario, SC4)*: In order to improve the authenticity of the simulations, the results obtained from the hardware assessment are embedded in the latencies of the network simulator. This highlights the overhead of the proposed cryptographic computations on the latencies of scenario 3.
- 5) *Multicast NoDrop IMAC (SC5)*: This represents a multicast key distribution scenario with a single MAC being used to authenticate the whole packet as opposed to the proposed authentication mechanism. A node extracts its own data and forwards the packet without dropping any bytes. The IMAC occupies 16 bytes. A drawback of this approach is that it requires a shared key among all meters for the shared MAC, which can be seen as a drawback from a security perspective.



Fig. 14. Development board.

- 6) *Multicast withDrop MAC (SC6)*: This represents a multicast key distribution scenario with different MACs being used to authenticate the different keys. A node extracts its own data and then forwards to its children only the set of keys intended for their respective sub-trees. This enables an apple-to-apple comparison to our scenario 3 in terms of packet drop. However, each MAC occupies 16 bytes; hence the original packet is much larger than that of scenario 5.

2) *Simulation Setup*: To perform the simulations, we rely on the discrete event network simulator NS-3 [39]. The wireless mesh networks are interconnected using IEEE 802.11s, which is an IEEE 802.11 amendment for mesh networking. We set the link rate to 6 Mbps and the orthogonal frequency division multiplexing (OFDM) is used as the modulation scheme. The network layer is based on IPv4 and uses UDP as the transport layer. The newly updated keys are placed in the application layer. Note that, for purposes of maintaining the connectivity of the wireless mesh network many packets are exchanged between the nodes at the data link layer. Hence, many of the received and transmitted packets do not contain application layer data. Data generated from such packets will still be logged in our network simulations. The amount of data at the application layer is dependent on the key delivery mechanism scenario. For all the scenarios, the application layer contains a nonce used for the encryption process; the nonce has a size of 4 bytes, the ID has a size of 4 bytes and the private key has a size of 20 bytes. For the unicast scenario, the application layer contains a nonce, an ID, an encrypted ID_key and a MAC. Unlike the unicast, the multicast scenarios have a variable amount of data at the application layer depending on the number of nodes the packet is targeted to. Thus, the multicast packet contains one nonce and multiple IDs along with the corresponding encrypted key blocks. For purposes of our approach, the data of an encrypted key block corresponds to the private keys shuffled with the ID bits.

3) *Case Study 1 (Network Analysis)*: In our first set of experiments, we consider two network topologies. The first network is a chain shaped network and the second is a tree shaped network (Figure 15) such that each node can have a maximum of 2 children. The root (node 0) represents the aggregator and the remaining nodes represent the smart meters in the neighborhood area network. A proactive, tree-based routing protocol is used prior to running the key update to establish the routing tables. The cost of the establishment of the routing tables is excluded in the results of the simulations.

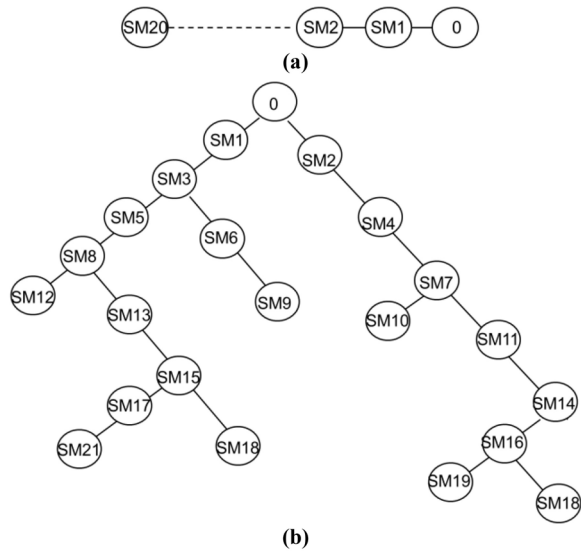


Fig. 15. Chain and Tree-based networks used in simulation.

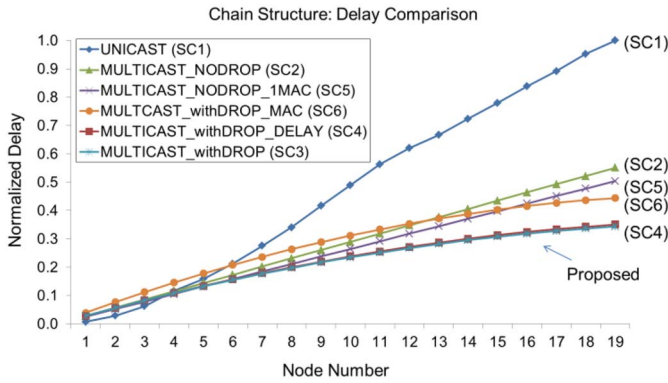


Fig. 16. Delay comparison for the chain based network.

Solid lines connecting two nodes indicate the presence of a wireless communication link between the two. For the tree network, some children are dropped to maintain tree configuration in a wireless mesh network based on distance requirements for establishing a link. The two main criteria used to assess the performance of the proposed key delivery mechanism are 1) the latencies incurred on the delivery of the keys and 2) the traffic generated during the key delivery process. Traffic assessment helps identify traffic footprints of the different methods. Reduction in the traffic aids in reducing the congestion seen in the wireless medium, thereby facilitating the wireless medium access to regular smart meter data. Figures 16 and 17 present the normalized latency data obtained at each node for the chain network and tree network, respectively.

We observe the following:

- 1) For the chain network, the proposed methodology reduces the latency by 70%, 30%, and 20% compared to the SC1, SC5, and SC6 scenarios, respectively.
- 2) For the tree network, the proposed *Multicast withDrop* scenario reduces the latency compared to SC5 since multiple keys are dropped when forwarding. On average, the proposed *Multicast withDrop* reduces the latency by

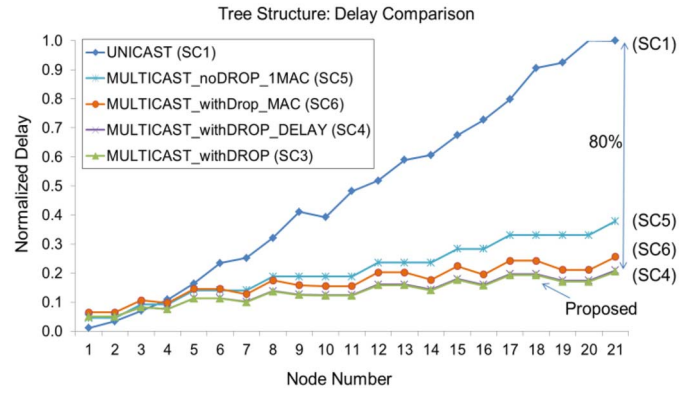


Fig. 17. Delay comparison for the tree based network.

TABLE VIII
AVERAGE TRAFFIC REDUCTION FOR PROPOSED METHODOLOGY

		Proposed	
		Multicast withDrop	
Chain	*Tx	Unicast	20%
		Multicast noDrop 1MAC	8%
		Multicast withDrop MAC	12%
	*Rx	Unicast	22%
		Multicast noDrop 1MAC	12%
		Multicast withDrop MAC	14%
Tree	*Tx	Unicast	27%
		Multicast noDrop 1MAC	22%
		Multicast withDrop MAC	14%
	*Rx	Unicast	31%
		Multicast noDrop 1MAC	23%
		Multicast withDrop MAC	20%

*Transmitted (Tx), Received (Rx)

80%, 44% and 20% compared to SC1, SC5, and SC6 scenarios, respectively.

The computational overhead introduced by the proposed authentication scheme are not significant and can be considered negligible. We observe a 2% overhead for both tree and chain structures.

On a side note, the *Multicast withDrop_MAC* (SC6) performs better than the *Multicast_NoDrop_1MAC* (SC5) for both structures; however, this advantage is seen towards the end of the chain network due to the initially large packet and the rate of one key drop per chain node. As for the traffic observed during the key update process, Table VIII shows the average reduction in the transmitted and received traffic packets for both networks.

- 1) For the chain network, the proposed *Multicast withDrop* achieves on average around 10%, 13% and 30% reduction in the amount of traffic compared to the SC5, SC6, and SC1 scenarios, respectively.
- 2) For the tree-based network, the proposed *Multicast withDrop* achieves on average around 17%, 20% and 30% reduction in the amount of traffic compared to SC6, SC5, and SC1 scenarios, respectively.

The reduction is mainly attributed to the smaller size of the shuffled_ID-based authentication mechanism, and the drop capability. Taking a closer look at the received bytes at each node of the tree network, Figure 18 shows that the *Unicast* based key delivery generates the most traffic and the *Multicast*

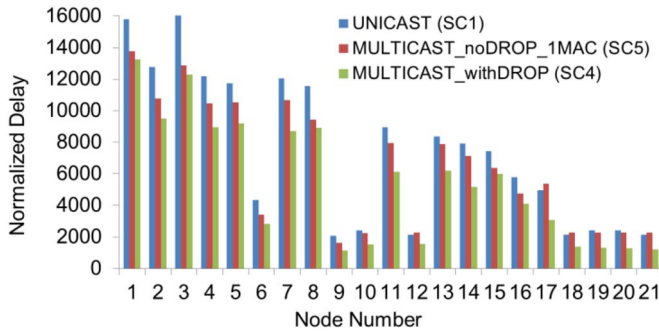


Fig. 18. Bytes received at each node for a tree based network.

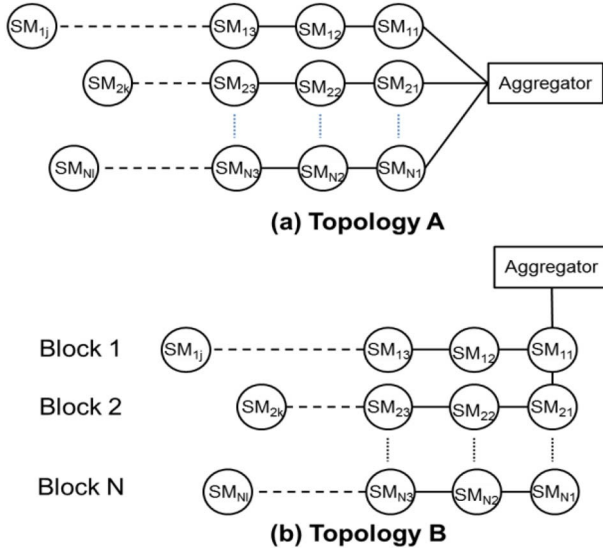


Fig. 19. The neighborhood consists of several blocks. Blocks may represent streets with multiple houses or a building with multiple houses. We present two topologies: (a) Topology A: all the blocks are connected directly to the aggregator. (b) Topology B: The blocks form a chain to connect to the aggregator.

withDrop generates the least traffic. Note that nodes such as 9 and 10 in the tree structure of Figure 15 are leaf nodes, and this explains the low received traffic. Finally, we note that, as discussed earlier, that network traffic is not solely due to the key update packets.

4) *Case Study 2 (Neighborhood Analysis)*: In this set of experiments, we emulate random neighborhoods and we extend the latency analysis to encompass two possible smart grid neighborhoods topologies. We assume that the neighborhood houses (meters) are organized in 2 topologies as illustrated in Figure 19 whereby in *Topology A*, all blocks are connected to the aggregator, while in *Topology B*, the roots of the blocks form a chain, which eventually connects to the aggregator. The difference between the two topologies is that for Topology B, a multicast packet has to travel first to the root of the block before shedding any keys. This adds extra penalty for larger packets like scenario 6 which has multiple MACs. We emulate 300 neighborhoods by randomly varying (1) the number of blocks per neighborhood, and (2) the number of meters per block. Both numbers are uniformly varied over the range [20, 50] allowing the aggregator to handle between 400

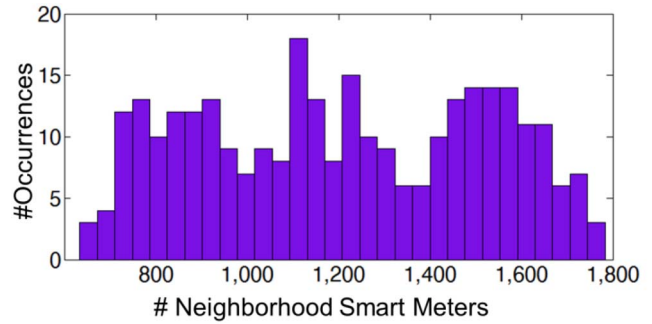


Fig. 20. Histogram of the number of smart meters in the neighborhood for the different topologies.

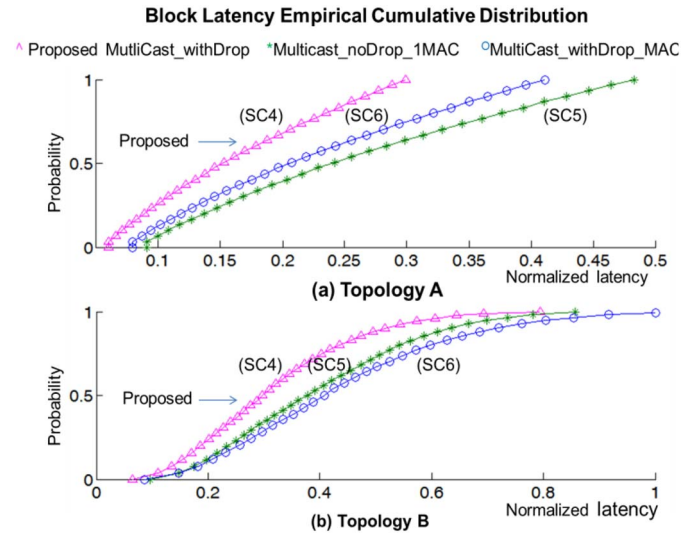


Fig. 21. Empirical Cumulative distribution for the individual block latencies collected from all neighborhood simulations. The results are shown for topologies A and B. Latency is normalized to maximum block latency observed in both topologies.

and 2500 meters for a given neighborhood; typically, an aggregator can handle few hundreds to few thousand meters over the NAN. Figure 20 presents the histogram of the number of smart meters per neighborhood for all the neighborhoods.

We perform latency analysis for the different neighborhoods for the proposed scenario SC4, and scenarios SC5 and SC6 (Section VI-B1). We define the block latency as the time required to deliver the keys to all the meters in a specific block. We also define the per-neighborhood average block latency, as the average of the block latencies in a given neighborhood. Figure 21 shows the empirical cumulative distribution function (cdf), also known as the Kaplan-Meier cdf estimate [40], of the individual block latencies for all the simulated neighborhoods. For both topologies, the latency of the proposed methodology is significantly reduced compared to the other two scenarios. We observe the following.

- 1) For Topology A, the block latencies are function of the block size, which is a uniformly distributed random variable; hence we observe close-to-linear latency cdf.
- 2) For Topology B, the latencies are function of two variables: the distance from the aggregator to the block, which is a function of the number of blocks per

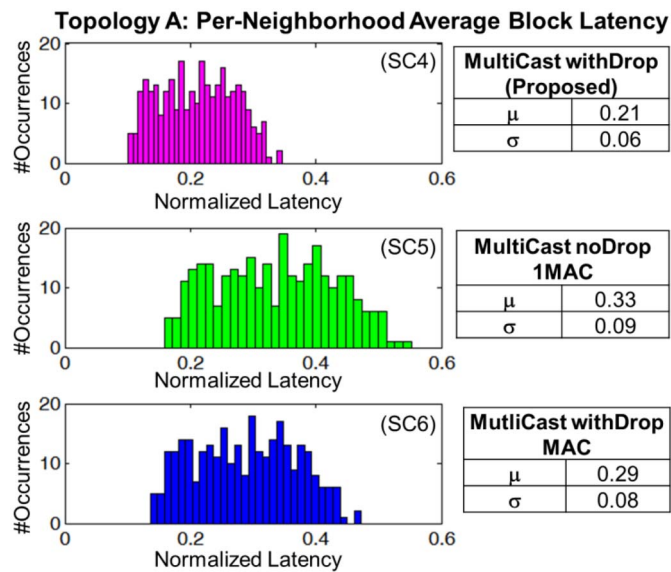


Fig. 22. Topology A: Histogram of the normalized per-neighborhood average block latency for the different neighborhoods. Latency is normalized to the maximum per-neighborhood average latency obtained for topology B.

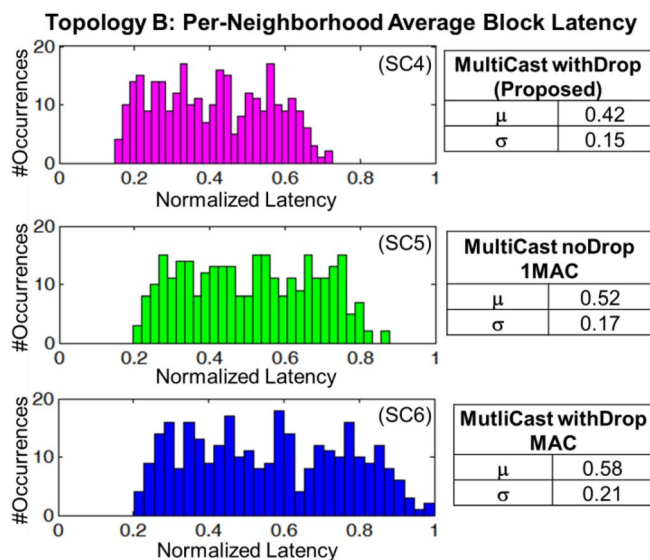


Fig. 23. Topology B: Histogram of the normalized per-neighborhood average block latency for the different neighborhoods. Latency is normalized to the maximum per-neighborhood average latency obtained for topology B.

neighborhood, and the block size. In fact, the skewed long tails for Topology B are attributed to large blocks that are far from the aggregator. Their respective large multicast packets have to travel long distances before shedding any keys; thus, they encounter larger overall latencies compared to near blocks and compared to Topology A scenario where this distance drops to zero. This offsets the gap between the different scenarios near tail regions for the extremely upper quantiles. It also penalizes the *MultiCast_withDrop_MAC* (SC6) technique further due to its much larger initial packet size (carrying multiple MACs) and renders it slower than the other two scenarios flipping the order versus Topology A.

3) Most importantly, due to the low authentication overhead along with drop capability, the proposed methodology portrays lower latencies for both topologies, (at all percentile levels). This trend is further portrayed when analyzing the per-neighborhood average block latency for the different neighborhoods as illustrated in Figures 22 and 23. For Topology A, the average latency reduction for the proposed methodology is 37% and 27% compared to SC5 and SC6, respectively. In the same order, the average latency reduction is 20% and 26% for Topology B.

VII. CONCLUSION

We proposed an ID-based non-interactive key distribution framework for secure AMIs. Neighboring meters rely on pairings of their private keys and public IDs to generate pair-wise symmetric keys without exchanging additional packets. The pairwise symmetric keys are generated once and are used to verify packet authenticity and validity at the link layer. The PUF response is used as the key to securely encrypt and store the pairwise symmetric keys and private keys, and to securely encrypt meter-to-utility end-to-end messages. We further propose a new lightweight key delivery mechanism that exploits the multicasting features of the network along with a new authentication algorithm that relies on pseudorandom properties of the output of AES to authenticate the updated keys. Simulations conducted on a network simulator along with hardware tests proved the feasibility and efficiency of the proposed key authentication and delivery mechanism.

REFERENCES

- [1] C. W. Gellings, *The Smart Grid: Enabling Energy Efficiency and Demand Response*. Boca Raton, FL, USA: CRC Press, Aug. 2009.
- [2] G. Neichin, and D. Cheng. (2010). *US Smart Grid Vendor Ecosystem: Report on the Companies and Market Dynamic Shaping the Current US Smart Grid Landscap.* [Online]. Available: <http://energy.gov/maps/recovery-act-smart-grid-projects>
- [3] R. Anderson and S. Fuloria, "Who controls the off switch?" in *Proc. IEEE Smart-GridComm*, Gaithersburg, MD, USA, 2010, pp. 96–101.
- [4] M. Nabeel, S. Kerr, X. Ding, and E. Bertino, "Authentication and key management for advanced metering infrastructures utilizing physically unclonable functions," in *Proc. IEEE Smart-GridComm*, Tainan, Taiwan, 2012, pp. 324–329.
- [5] H. K.-H. So, S. K. M. Kwok, E. Y. Lam, and K.-S. Lui, "Zero-configuration identity-based signcryption scheme for smart grid," in *Proc. IEEE Smart-GridComm*, Gaithersburg, MD, USA, 2010, pp. 321–326.
- [6] Y.-J. Kim, V. Kolesnikov, and M. Thottan, "Resilient end-to-end message protection for large-scale cyber-physical system communications," in *Proc. IEEE Smart-GridComm*, Tainan, Taiwan, Nov. 2012, pp. 193–198.
- [7] H. Li, R. Lu, L. Zhou, B. Yang, and X. Shen, "An efficient merkle-tree-based authentication scheme for smart grid," *IEEE Syst. J.*, vol. 8, no. 2, pp. 655–663, Jun. 2014.
- [8] V. Seferian, R. Kanj, A. Chehab, and A. Kayssi, "PUF and ID-based key distribution security framework for advanced metering infrastructures," in *Proc. IEEE Int. Conf. SmartGridComm*, Venice, Italy, 2014, pp. 933–938.
- [9] C. Karlof, N. Sastry, and D. Wagner, "TinySec: A link layer security architecture for wireless sensor networks," in *Proc. 2nd Int. Conf. Embedded Netw. Sens. Syst.*, Baltimore, MD, USA, 2004, pp. 162–175.
- [10] M. G. Gouda, E. N. Elnozahy, C.-T. Huang, and T. M. McGuires, "Hop integrity in computer networks," *IEEE/ACM Trans. Netw.*, vol. 10, no. 3, pp. 308–319, Jun. 2002.
- [11] S. W. Smith, "Cryptographic scalability challenges in the smart grid (extended abstract)," in *Proc. ISGT*, 2012, pp. 1–3.

- [12] H. Khurana, M. Hadley, N. Lu, and D. A. Frincke, "Smart-grid security issues," *IEEE Security Privacy*, vol. 8, no. 1, pp. 81–85, Jan./Feb. 2010.
- [13] R. Dupont and A. Enge, "Provably secure non-interactive key distribution based on pairings," *Discrete Appl. Math.*, vol. 154, no. 2, pp. 270–276, 2006.
- [14] N.E.T. Lab. (2007). *A Systems View of the Modern Grid*. [Online]. Available: https://www.smartgrid.gov/files/a_systems_view_of_the_modern_grid.pdf
- [15] R. C. Qui *et al.*, "Cognitive radio network for the smart grid: Experimental system architecture, control algorithms, security, and microgrid testbed," *IEEE Trans. Smart Grid*, vol. 2, no. 4, pp. 724–740, Dec. 2011.
- [16] S. M. Amin, "Securing the electricity grid," *Bridge Quart. Pub. U.S. Nat. Acad. Eng.*, vol. 40, no. 1, pp. 13–20, 2010.
- [17] *Report to NIST on Smart Grid Interoperability Standards Roadmap EPRI*, Accessed on Jun. 17, 2009. [Online]. Available: <http://www.nist.gov/smartgrid/InterimSmartGridRoadmapNISTRe-structure.pdf>
- [18] A. R. Metke and R. L. Ekl, "Security technology for smart grid networks," *IEEE Trans. Smart Grid*, vol. 1, no. 1, pp. 99–107, Jun. 2010.
- [19] W. Wang and Z. Lu, "Cyber security in the smart grid: Survey and challenges," *Comput. Netw.*, vol. 57, no. 5, pp. 1344–1371, 2013.
- [20] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Proc. Workshop Theory Appl. Cryptogr. Tech.*, Santa Barbara, CA, USA, 1984, pp. 47–53.
- [21] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," in *Proc. Annu. Int. Cryptol. Conf.*, Santa Barbara, CA, USA, 2001, pp. 213–229.
- [22] J. H. Silverman, J. Pipher, and J. Hoffstein, *An Introduction to Mathematical Cryptography*. vol. 1. New York, NY, USA: Springer, 2008.
- [23] S. D. Galbraith, K. G. Paterson, and N. P. Smart, "Pairings for cryptographers," *Discrete Appl. Math.*, vol. 156, no. 16, pp. 3113–3121, 2008.
- [24] D. F. Aranha, K. Karabina, P. Longa, C. H. Gebotys, and J. López, "Faster explicit formulas for computing pairings over ordinary curves," in *Proc. Annu. Int. Conf. Theory Appl. Cryptogr. Tech.*, Tallinn, Estonia, 2011, pp. 48–68.
- [25] T. Acar, K. Lauter, M. Naehrig, and D. Shumow, "Affine pairings on ARM," in *Pairing-Based Cryptography—Pairing 2012*. Heidelberg, Germany: Springer, 2012, pp. 203–209.
- [26] G. Grewal, R. Azarderakhsh, P. Longa, S. Hu, and D. Jao, "Efficient implementation of bilinear pairings on ARM processors," in *Selected Areas in Cryptography*. Heidelberg, Germany: Springer, 2012.
- [27] G. E. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," in *Proc. 44th ACM/IEEE Design Autom. Conf.*, San Diego, CA, USA, 2007, pp. 9–14.
- [28] G. E. Suh, C. W. O'Donnell, I. Sachdev, and S. Devadas, "Design and implementation of the AEGIS single-chip secure processor using physical random functions," in *Proc. 32nd Int. Symp. Comput. Archit. (ISCA)*, Madison, WI, USA, 2005, pp. 25–36.
- [29] K. G. Paterson, "Cryptography from pairings," in *Advances in Elliptic Curve Cryptography* (London Mathematical Society Lecture Notes), vol. 317. Cambridge, U.K.: Cambridge Univ. Press, 2005.
- [30] E. Beroaset, *ANSI C12.22 in Context*, American National Standard: Protocol Specification for Data Communication Networks, ANSI C12.22, 2008. [Online]. Available: <http://www.gridwiseac.org/>
- [31] K.-H. Chang and B. Mason, "The IEEE 802.15.4g standard for smart metering utility networks," in *Proc. IEEE Smart-GridComm*, Tainan, Taiwan, 2012, pp. 476–480.
- [32] G. R. Hiertz *et al.*, "IEEE 802.11s: The WLAN mesh standard," *IEEE Wireless Commun.*, vol. 17, no. 1, pp. 104–111, Feb. 2010.
- [33] M. Luby and C. Rackoff, "Pseudo-random permutation generators and cryptographic composition," in *Proc. ACM STC*, London, U.K., 1986, pp. 356–363.
- [34] E. Barker *et al.*, "Recommendation for key management-Part 1: General (revision 3)," NIST Special Publication: 800-57 Pt1 Rev 3, 2012.
- [35] (2016). *Empowering Users to Make Smarter Power Choices*. [Online]. Available: <https://www.arm.com/markets/embedded/smart-meter.php>
- [36] (2016). *STM32F217IGH Datasheet*. [Online]. Available: http://www.st.com/content/st_com/en/search.html#q=STM32F217IGH-t=keywords-page=1
- [37] (2016). *MDK Microcontroller Development Kit*. [Online]. Available: <http://www2.keil.com/mdk5>
- [38] (2016). *WolfSSL Embedded SSL Library*. [Online]. Available: <https://www.wolfssl.com/wolfSSL/Products-wolfssl.html>
- [39] (2015). *NS-3: Discrete Event Network Simulator for Internet Systems*. [Online]. Available: <https://www.nsnam.org/>
- [40] E. L. Kaplan and P. Meier, "Nonparametric estimation from incomplete observations," *J. Amer. Stat. Assoc.*, vol. 53, no. 282, pp. 457–481, 1958.



Vahe Seferian received the B.E. and M.E. degrees in electrical and computer engineering from the American University of Beirut in 2013 and 2016, respectively. During the graduate studies he focused on cryptography and the security of smart grid networks. He is currently pursuing a career in embedded system design where he is working on power and inverter applications.



Rouwaida Kanj received the M.S. and Ph.D. degrees in electrical engineering from the University of Illinois Urbana-Champaign in 2000 and 2004, respectively. She is currently an Assistant Professor with the American University of Beirut. From 2004 to 2012, she was a Research Staff Member with IBM Austin Research Laboratory. She worked on memory designs and statistical modeling and yield estimation methodologies. She has authored over 55 technical papers, 20 issued patents, and several pending patents. She was a recipient of three IBM Ph.D. Fellowships, the Outstanding Technical Achievement Award, six Invention Plateau Awards from IBM, the IEEE/ACM William J. McCalla ICCAD Best Paper Award in 2009, and ISQED Best Paper Award in 2006 and 2014.



Ali Chehab received the bachelor's degree in EE from American University of Beirut (AUB) in 1987, the master's degree in EE from Syracuse University in 1989, and the Ph.D. degree in ECE from the University of North Carolina at Charlotte, in 2002. From 1989 to 1998, he was a Lecturer with the ECE Department, AUB. He rejoined the ECE Department, AUB, as an Assistant Professor in 2002, became a Full Professor in 2014. He teaches courses in programming, electronics, digital systems design, computer organization, cryptography, and digital systems testing. He has about 200 publications. His research interests include wireless communications security, cloud computing security, multimedia security, trust in distributed computing, low energy very large scale integration (VLSI) design, and VLSI testing. He was a recipient of the AUB Teaching Excellence Award in 2007. He is a senior member of ACM.



Ayman Kayssi received the B.E. degree (with Distinction) in electrical engineering from the American University of Beirut (AUB), in 1987, and the M.S.E. and Ph.D. degrees from the University of Michigan, Ann Arbor, in 1989 and 1993, respectively. In 1993, he joined the Department of Electrical and Computer Engineering (ECE), AUB, where he is currently a Full Professor. From 2004 to 2007, he served as the Chairman of the ECE Department, AUB, and is currently an Associate Dean of the Faculty of Engineering and Architecture.

He teaches courses in electronics and in networking. He has published over 200 articles in the areas of security, networking, and very large scale integration. His research interests are in information security and networking, and in integrated circuit design and test. He was a recipient of AUB's Teaching Excellence Award in 2003. He is a member of ACM, ISOC, and the Beirut OEA.