
Article:
Gope, P. (2020) PMAKE : Privacy-aware multi-factor authenticated key establishment scheme for advance metering infrastructure in smart grid. Computer Communications, 152. pp. 338-344. ISSN: 0140-3664

https://doi.org/10.1016/j.comcom.2019.12.042

PMAKE: Privacy-Aware Multi-Factor Authenticated Key Establishment Scheme for Advance Metering Infrastructure in Smart Grid

Prosanta Gope

Department of Computer Science

University of Sheffield, Regent Court, 211 Portobello Sheffield S1 4DP, United Kingdom

Email: p.gope@sheffield.ac.uk

Abstract

Advanced Metering Infrastructure (AMI) ensures bidirectional communication for analysis and improvement of energy usage between service providers and customers. Establishing secure and efficient key exchange has become an important issue for the Internet-based smart grid environment. To ensure a secure, reliable and efficient operation in the smart grid environment, AMI should be protected from security attacks. On the other hand, since smart meters are installed in customer's homes and businesses without hardware protection mechanisms that could be susceptible to physical attacks, where a dishonest customer may attempt to temper the metering data in order to make differences in billing. In this paper, an efficient privacy-aware multi-factor authenticaticated key establishment scheme (PMAKE) is proposed. One of the notable properties of the proposed scheme is that it can ensure physical security of the smart meters. From the performance analyses, we can claim that our scheme is secure and suitable for the resource limited smart meters.

Keywords: Multi-Factor authentication, Advanced metering infrastructure, Smart grids.

Table 1: Symbols and cryptographic function

Symbol	Definition
SID	Shadow identity of SM
$CRP(C, R)$	Challenge-Response pair
sk	Session key (SM -service provider)
P_{SM}	Physically uncloneable functions of SM
$h(\cdot)$	One-way hash function
$\oplus$	Exclusive-OR operation
FE	Fuzzy extractor
$\|\|$	Concatenation operation

1 Introduction

Smart grid is envisioned to be the next-generation power grid that comprises of integrated Information and Communications Technologies (ICT), cyber-physical systems and power system technologies [1]. By integrating Internet technologies, smart grid enables seamless interaction amongst all energy systems and networks in order to achieve increase in energy efficiencies [2]. Energy domains such as generation, transmission, distribution, service provider, operations, markets, and customers will benefit from secure and efficient communication on decisions about energy and information flow. All components in the smart grid are integrated to ensure that data are delivered in real-time. Advanced Metering Infrastructure (AMI) in smart grid is an integration of technologies to ensure seamless and intelligent link between the utility service providers and consumers. As shown in Fig. 1, AMI architecture consists of five major components such as a service provider, a set of smart meters (SMs), numerous home area networks (HANs), meter data management system (MDMS), and wide area network (WAN). Fig. 1 shows that the smart meters communicate with the service provider via the WAN. The service provider maintains the MDMS for storing all the credentials related to each SM.

Since, smart grid has many connections with people's daily lives [3]. Hence, it has now attracted many researchers' attention for its ability to enable power utilities to allocate the power resources more reasonably and efficiently, and its important role in establishing smart cities. Although the benefits brought by smart grid to utility companies, service providers and electricity consumers are obvious, information security along with it is an important factor restricting the development of the smart gird applications. Especially, ensuring the security of communication between a user and a smart meter needs more attention because of that this communication channel is usually open. In view of this, numerous solutions have been put forward to ensure security and privacy.

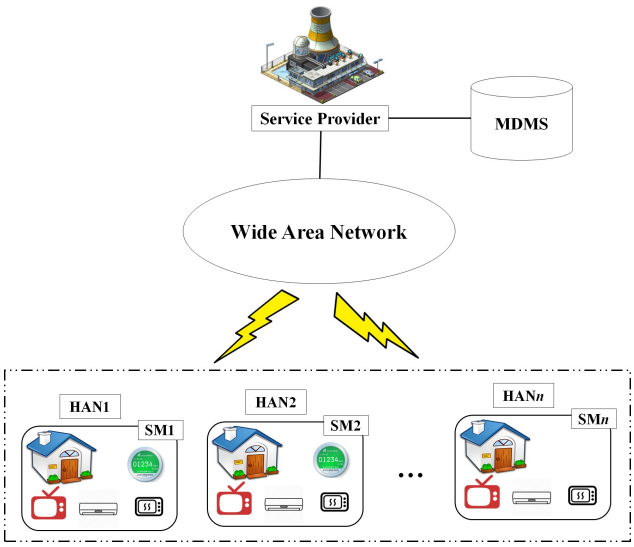

Figure 1: Advanced Metering Infrastructure (AMI) Architecture

1.1 Related Work

Secure authentication and key exchange concerns usually arise in many applications. Many researchers around the world have proposed several authentication and key exchange schemes suitable for the secure smart grid communication with considerations of security goals. Mohammadali et al. [4] proposed two identity-based key establishment protocols, based on ECC. However, they are vulnerable to side channel attacks, probing attack and false data injection attack, and they incur high computational cost during key establishment. Wu and Zhou [5] combined symmetric key and public key techniques to present a fault-tolerant and scalable key management scheme that can eliminate man-in-the-middle attack and replay attack. Xia and Wang [6] later showed that [5] is vulnerable to man-the-in-middle attack. Furthermore, Park et al. [7] found the scheme of [6] vulnerable to impersonation attack and unknown key share attack. Tsai et al. [8] combined identity-based signature scheme and identity-based encryption scheme to achieve security and efficiency for key distribution in the smart grid. They proposed a tamper-proof module (for storing key data), which can mitigate probing attack. However, Odelu et al. [9] reported that the Tsai et al.'s scheme provides weak security to the session key and that leads to many other security attacks. Consequently, most of the proposed schemes are either vulnerable to security attacks or require high computation costs at resource-constrained SMs. Furthermore, none of the existing scheme can ensure the physical security of the smart meters, which is imperative to resist an inside attacker (home user) to compromise and control of the smart meter for their own profit. In this paper, we propose an effective solution, which can address all the above issues.

The paper is organized as follows. Section 2 deals with preliminaries. In Section 3, the proposed scheme is described. In Section 4, we give a formal proof of the security and Privacy. The computational and the communication cost is explained in Section 5. Finally, conclusions

are drawn in Section 6. The symbols and cryptographic functions of the proposed scheme are defined in Table I.

2 Preliminaries

2.1 Fuzzy Extractor

A fuzzy extractor (FE) (d, λ, ϵ) [11-14], [17]is a is defined as a pair of functions FE.Gen(.) and FE.Rec(.), corresponding to the key generation and reproduction procedures respectively. FE.Gen(.) is basically a probabilistic algorithm that generates a key K and helper data hd, i.e., (K, hd) = FE.Gen(R) on a given input bit string R. FE.Rec(.) is a deterministic function, which takes a noisy input R' and the helper data hd and then it outputs the key K i.e., K = FE.Rec ($R^{'}$, hd,) i.e., K = FE.Rec ($R^{'}$, hd) when the hamming distance between R' and R is at most d.

2.2 Reverse Fuzzy Extractor

The concept of reverse fuzzy extractor ensures fast implementation of secure sketch and fuzzy extractors. In this context, the PUF-enabled smart meters do not require to perform the computationally intensive reconstruction algorithm FE.Rec. Instead of that, the smart meters are required to execute the helper data generation algorithm FE.Gen. Therefore, each time when the PUF is queried, new helper data hd is generated. Then, the verifier corrects the reference value R of its database to the noisy PUF response R', which is different each time the PUF is evaluated.

2.3 Physically Uncloneable Function

We can define PUF as a challenge-response pair (CRP). For a given input challenge C, the PUF outputs a random string R i.e., $R = P(C)$. A PUF P is said to be $(d, n, l, \lambda, \epsilon)$-secure if the following conditions hold:

1. Conceive, there are two PUFs $P_1(\cdot)$ and $P_2(\cdot)$, and for any given input $C_1 \in \{0,1\}^k$, $\Pr[\mathrm{HD}(P_1(C_1), P_2(C_1)) > d] \geq 1 - \varepsilon$.
2. For any PUF $P_i(\cdot)$ and for any input $C_1, \cdots, C_n \in \{0,1\}^k$, $\Pr[\mathrm{HD}(P_i(C_1), P_i(C_2)) > d] \geq 1 - \varepsilon$.
3. Conceive, there are two PUFs $P_i(\cdot)$ and $P_{i*}(\cdot)$, and for given any inputs $C_1, \cdots, C_n \in \{0,1\}^k$, $\Pr[\hat{H_\infty}(P_i(C_k), P_i*(C_j))_{1 \leq j,k \leq n, i \neq i*, j \neq k} > \lambda] \geq 1 - \varepsilon$. Therefore, we can say that in order to evaluate different PUFs using multiple inputs, the min-entropy of the PUF outputs $\geq \lambda$ and the intra-distance $\leq d$, and the inter-distance $\geq d$.

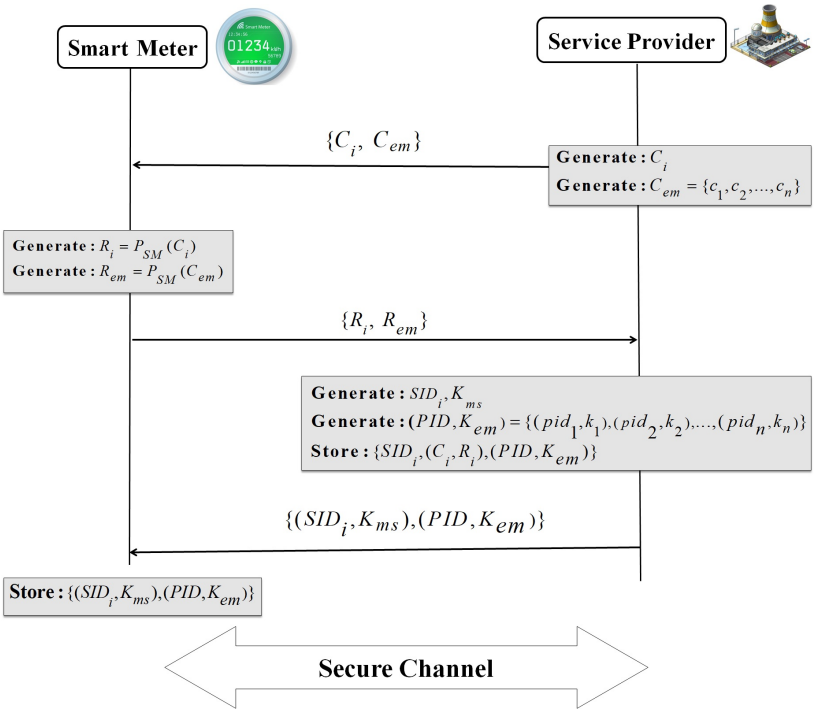

Figure 2: Enrollment phase of the proposed PMAKE scheme

2.4 Pseudorandom Functions

A pseudorandom function PRF:$\{0,1\}^k \times \{0,1\}^* \rightarrow \{0,1\}^{k'}$ which takes a secret security parameter $K \epsilon \{0,1\}^k$ and a message $M \epsilon \{0,1\}^*$ as input and provides an arbitrary string PRF(K, M) which is indistinguishable from random string. Now, assuming that h be a polynomial-time computable pseudorandom function. For distinguishing h, a probabilistic polynomial-time (PPT) adversary $\mathcal{A}$ may request polynomial bounded queries with its selected inputs and obtain the outputs computed by h for training. After the training phase, $\mathcal{A}$ is given a function, which is either h or a truly random function. We say that h is a pseudo-random function, if it is indistinguishable from a truly random function under $\mathcal{A}$. Namely, $\mathcal{A}$ is given either h or a truly random function according to a random bit $\{0, 1\}$ and it has only the probability $\frac{1}{2} + \varepsilon$, to distinguish h.

3 Proposed Scheme

An efficient privacy-preserving multi-factor authenticated key establishment (PMAKE) scheme based on the PUF, reverse fuzzy extractor, and cryptographic one-way hash function is proposed here for achieving secure smart grid communication. The main objective of this novel scheme is of two aspects: (1) The authentication phase is provided to achieve mutual authentication and untraceability property and (2) it can withstand all known passive and active attacks (including the physical security of the smart meter). The proposed scheme consists of two phases: *Enrollment*, and *Authentication*.

3.1 Enrollment Phase

As displayed in Fig. 1, the new consumer registers his/her smart meter to the server as to becoming a valid consumer of the service provider. In this context, a smart meter SM needs to send its enrollment request to the service provider. After receiving the request, the service provider first randomly generates a challenge C_i for the i-th round interaction with the SM. Then the service provider also generates a set of new challenges C_{em}= $\{c_1, ...,c_n\}$. These challenges (C_{em}) will be used if there is any desynchronization occurs between the service provider and the smart meter. Next, the service provider sends $\{C_i, C_{em}\}$ to the SM. After receiving the challenges $\{C_i, C_{em}\}$, the smart meter SM extracts the PUF outputs R_i= $P_{SM}(C_i)$, R_{em}= $P_{SM}(C_{em})$ and sends $\{R_i, R_{em}\}$ to the service provider. Then, the service provider first randomly generates a unique shadow identity SID , and a secret key K_{ms}. Next, the service provider also generates a set of unique pseudo identity and synchronization key pair (PID, K_{em}) =$\{(pid_1,k_{em1}),..., (pid_n, k_{emn})\}$ and sends $\{(SID, K_{ms}), (PID, K_{em})\}$ to the smart meter. Finally, for each enrolled smart meter SM, the service provider will store $\{(SID, K_{ms}), (C_i, R_i),(C_{em}, R_{em}),(PID, K_{em})\}$ in its database and the device stores $\{(SID, K_{ms}), (PID, K_{em})\}$.

3.2 Authentication Phase

Our authentication phase consists of the following steps:

Step 1: When the smart meter SM intends to communicate with the service provider, then the service provider first selects the shadow identity SID and then generates a nonce n_m and computes n_m^*=$n_m \oplus K_{ms}$, v_0=$h(SID||K_{ms}||n_m^*)$. Finally, the smart meter composes a request message M_1: $\{SID, n_m^*,v_0\}$ and sends it to the service provider.

Step 2: After receiving the request message M_1, the service provider first locates SID in its database and subsequently reads and loads $\{(C_i, R_i),K_{ms}\}$ into its memory. Hereafter, the service provider generates a nonce n_s and computes n_s^*= $K_{ms} \oplus n_s$,v_1= $h(n_m// K_{ms}//n_s^*)$ and then composes a message M_2 :$\{C_i, n_s^*, v_1\}$ and sends to the service provider.

Step 3: Next, upon receiving the message M_2, the smart meter extracts the PUF output $R_i^{'}$=$P_{SM}(C_i)$ and checks the key-hash response v_1. If it is valid, the smart meter calculates the following:$n_s = K_{ms}\oplus n_s^*$, $(k_i,hd_i) = \text{FE.Gen}(R_i^{'})$, hd^*=$h(K_{ms}||n_s)\oplus hd_i$, C_{i+1}=$h(C_i||k_i)$,$R_{i+1}^{'}$=$P_{SM}(C_{i+1})$,R_{i+1}^*= $k_i\oplus R_{i+1}^{'}$, v_2= $h(n_s||k_i||R_{i+1}^*||hd^*)$,$SID_{new}$=$h(SID\ ||k_i)$,$K_{ms}$=$h(K_{ms}//k_i)$, and the session key sk =$h(K_{ms}||k||n_m)$. Then, the device forms a message M_3:$\{R_{i+1}^*, v_2,hd^*\}$ and sends it to the service provider.

Step 4: After receiving the message M_3, the service provider first computes hd_i = $h(K_{ms}$ $||n_s)\oplus hd^*$, k_i=FE.Rec(R_i, hd_i)and then verifies the key-hash response v_2. If the verification is successful, the service provider calculates C_{i+1}= $h(C_i||k_i)$, $R_{i+1}^{'}$= $k_i\oplus R_{i+1}^*$, SID_{new}=$h(SID$

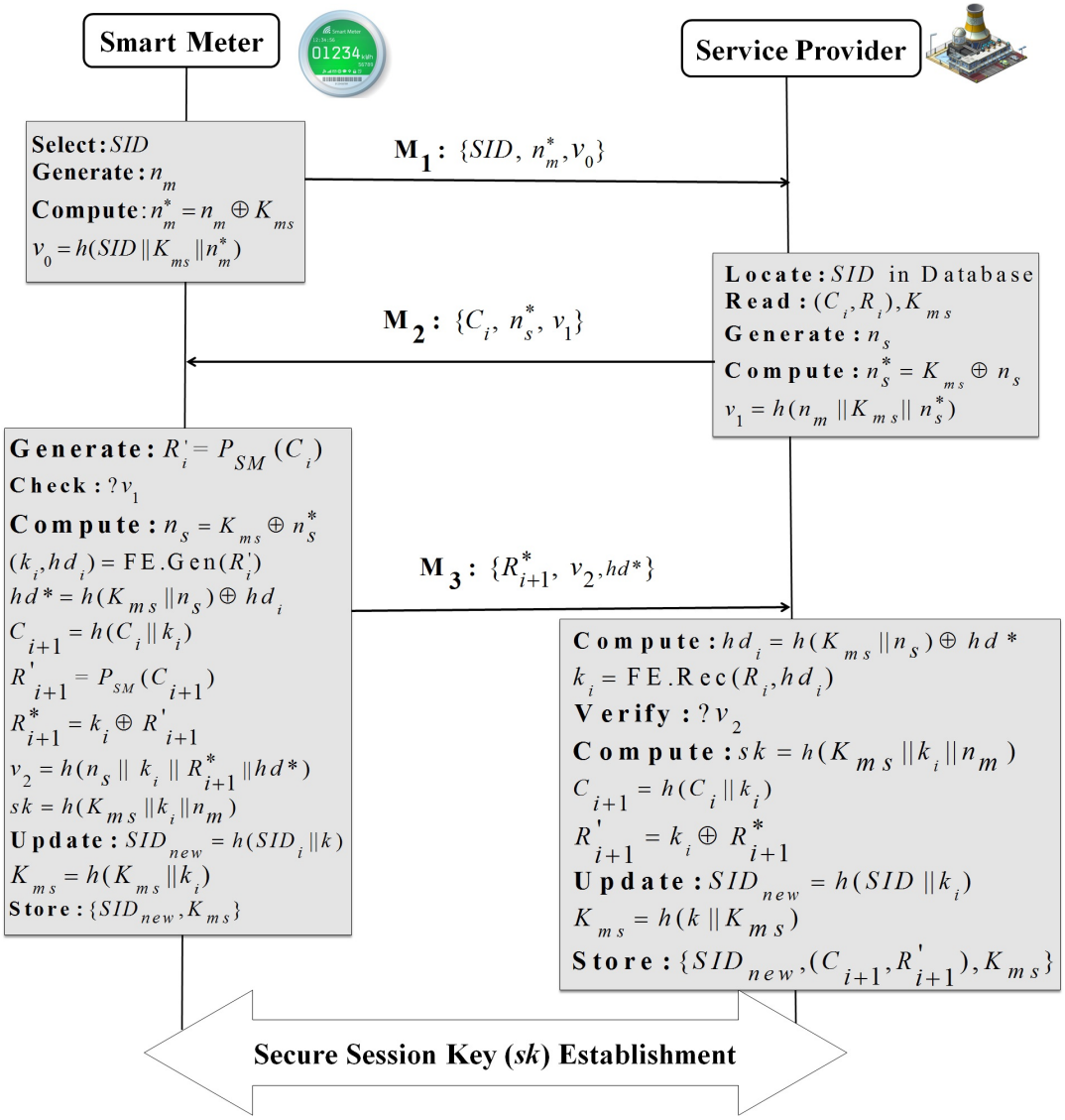

Figure 3: Efficient multi-factor authentication scheme (PMAKE) for smart grid communication.

$||k_i)$, K_{ms}=$h(K_{ms}//k_i)$, and the session key sk =$h(K_{ms}||k_i||n_m)$. Finally, the server stores $\{SID_{new}, (C_{i+1}, R'_{i+1}), K_{ms}\}$ for the next $(i+1)$-th interaction with the smart meter.

Now in Step 2, if the service provider cannot recognize the smart meter, then the service provider asks the smart meter to try again by using one of the unused pairs of $(pid_j,k_j)\in(PID, K_{em})$. Once a pair is used up, it must be deleted from the both ends. In this case, the service provider will select one of the unused CRP from (C_{em}, R_{em}) and a new shadow identity will be provided to the smart meter. Finally, the used pair of emergency CRP is also needed to be deleted from (C_{em}, R_{em}). In this way, we can address the desynchronization problem without compromising anonymity support. Details of this phase is depicted in Fig. 3.

4 Security Model

In this section we define a security model for our proposed scheme, adapting the model presented in [10].

Consider a set of smart meters $\mathcal{M} = \{M_1, M_2, \cdots, M_n\}$ that interact with the service provider S. S initially executes Setup(1^k) and produces a public parameter pp and a shared secret parameter sp. Here, pp denotes all the available public parameters (crypto suites) of the environment (e.g., PUF output length, coding mode, pseudo-random function (PRF) algorithm name, etc.) and sp represents the secret PUF responses. In this setup phase, S communicates with the smart meters in a secure environment and transfers the security credentials to start the authentication process. During the execution of the authentication phase, these parties interact through an insecure network and mutually authenticate each other. At the end, the parties output 1 (acceptance) or 0 (rejection) as the authentication result. The communication sequence between the parties is called a session and each session is distinguished by its session identifier, denoted by xid. We say that a session has a *matching session* if the messages exchanged between S and members of $\mathcal{M}$ are honestly transferred.

In this section, we consider security against the *man-in-the-middle attack*, which is the canonical security level for any authentication protocol. Now we consider a security game, denoted by $\text{Exp}^{\text{Sec}}_{\Pi,\mathcal{A}}(k)$, between a challenger $\mathcal{C}$ and adversary $\mathcal{A}$ against an authentication protocol Π.

$\text{Exp}^{\text{Sec}}_{\Pi,\mathcal{A}}(k)$:

1. $(pp, sp) \xleftarrow{Random} \text{Setup}(1^k)$;
2. $(\text{xid}^*, M_j) \xleftarrow{Random} \mathcal{A}_1^{\text{Launch,Send}S,\text{Send}\mathcal{M},\text{Result,Reveal}}(pp, S, \mathcal{M})$;
3. $b := \text{Result}(\text{xid}^*, M_j)$;
4. Output: b.

At the end of the setup phase, $\mathcal{A}$ can interact with the smart meter and the service provider and obtain various information by issuing the following oracle queries:

- Launch(1^k): Launch a service provider unit S to begin a new session with security parameter k.
- Send$S$: Send a random message m to S.
- Send$\mathcal{M}(M_j, m)$: Send arbitrary message m to the meter $M_j \in \mathcal{M}$.
- Result$(\mathcal{P}, \text{xid})$: Output whether the session sid of $\mathcal{P}$ is accepted or not, where $\mathcal{P} \in \{S, \mathcal{M}\}$.
- Reveal(M_j): Output the entire information contained in the memory of the meter M_j.

The advantage of the adversary $\mathcal{A}$ against the protocol Π, denoted by $\mathrm{Adv}^{\mathrm{Sec}}_{\Pi,\mathcal{A}}(k)$, is defined as the probability that the game $\mathrm{Exp}^{\mathrm{Sec}}_{\Pi,\mathcal{A}}(k)$ outputs 1 when xid* of $\mathcal{P}$ has no matching session.

Definition 1: *An authentication protocol* Π *is said to be secure against man-in-the-middle attacks with key compromise if for any probabilistic polynomial time adversary* $\mathcal{A}$, $\mathrm{Adv}^{\mathrm{Sec}}_{\Pi,\mathcal{A}}(k)$ *is negligible, i.e.,* $\mathrm{Adv}^{\mathrm{Sec}}_{\Pi,\mathcal{A}}(k) \leq \epsilon$ *in* k *(for large enough* k*).*

4.1 Privacy Model

Now we consider indistinguishability-based privacy. In this case, the adversary randomly picks two smart meters and tries to distinguish the communication derived from any one of the two meters. The privacy experiment between the challenger and the adversary $\mathcal{A} := (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3)$ is then described as follows:

$\mathrm{Exp}^{\mathrm{IND}*-\mathrm{b}}_{\Pi,\mathcal{A}}(k)$:

1. $(M_0^*, M_1^*, st_1) \underleftarrow{Random} \mathcal{A}_1^{\mathrm{Launch,Send}\mathcal{S},\mathrm{Send}\mathcal{M},\mathrm{Result,Reveal}}(pp, S, \mathcal{M})$;
2. $b \underleftarrow{\mathrm{U}} \{0,1\}$, $\mathcal{M}' := \mathcal{M}\{M_0^*, M_1^*\}$;
3. $\Pi_0 \underleftarrow{Random} \mathrm{Execute}(S, M_0^*)$, $\Pi_1 \underleftarrow{Random} \mathrm{Execute}(S, M_1^*)$, $st_2 \underleftarrow{Random} \mathcal{A}_2^{\mathrm{Launch,Send}\mathcal{S},\mathrm{Send}\mathcal{M},\mathrm{Result,Reveal}}(S, \mathcal{M}', \mathcal{I}(M_b^*), \Pi_0, \Pi_1, st_1)$;
4. $\Pi_0' \underleftarrow{Random} \mathrm{Execute}(S, M_0^*)$, $\Pi_1' \underleftarrow{Random} \mathrm{Execute}(S, M_1^*)$;
5. $b' \underleftarrow{Random} \mathcal{A}_3^{\mathrm{Launch,Send}\mathcal{S},\mathrm{Send}\mathcal{M},\mathrm{Result,Reveal}}(S, \mathcal{M}, \Pi_0', \Pi_1', st_1)$
6. Output b'.

At the end of the setup phase, the adversary $\mathcal{A}_1$ issues the oracle queries and sends the queries containing (M_0^*, M_1^*) to the challenger $\mathcal{C}$. After that, $\mathcal{C}$ flips a random coin $b \underleftarrow{\mathrm{U}} \{0,1\}$ and permits the adversary to anonymously interact with M_b^*. For the accomplishment of anonymous access, $\mathcal{A}_2$ invokes the Send$\mathcal{M}$ query with intermediate algorithm $\mathcal{I}$ as the input to honestly transfer the communication message between $\mathcal{A}_2$ and M_b^*. After the challenge phase, $\mathcal{A}_3$ can continuously interact with all meters including (M_0^*, M_1^*) as $\mathcal{A}_1$. Next, M_0^* and M_1^* call the Execute query to avoid trivial attacks (such as man-in-the-middle attaks) in the symmetric key based construction, and after that, they send their transcripts (Π_0, Π_1) and (Π_0', Π_1') of the protocol Π to the adversary. Therefore, the advantage of the adversary in guessing the correct bit can be defined as follows:

$$\mathrm{Adv}^{\mathrm{IND}*}_{\Pi,\mathcal{A}}(k) := \left| \Pr[\mathrm{Exp}^{\mathrm{IND}*-0}_{\Pi,\mathcal{A}}(k) \rightarrow 1] - \Pr[\mathrm{Exp}^{\mathrm{IND}*-1}_{\Pi,\mathcal{A}}(k) \rightarrow 1] \right|.$$

4.2 Security Considerations of the Proposed Authentication Scheme

Now we analyze the security of the proposed authentication protocol by using the above models.

Theorem 1 (Security). *Consider a (d, n, l, λ, ϵ_1)-secure PUF, and let* FE *be a (d, λ, ϵ_2)-secure fuzzy extractor, and h be a ϵ_3-secure pseudorandom function. Then the proposed protocol is secure against man-in-the-middle attacks with memory compromise. In particular, we have* $\mathrm{Adv}^{\mathrm{Sec}}_{\Pi,\mathcal{A}} \leq l.n(\epsilon_1 + \epsilon_2 + \epsilon_3)$.

Proof. The objective of adversary $\mathcal{A}$ is to violate the security experiment. In this context, the goal of $\mathcal{A}$ is to convince the smart meter or the service provider to accept the session without any matching session, especially when the communication is altered by the adversary. Now the following game transformations is considered. Let X_i be the advantage of the adversary at winning the game in Game i.

Game 0. It specifies the original game between the challenger $\mathcal{C}$ and the adversary.

Game 1. $\mathcal{C}$ randomly guesses the meter $M^* \underset{\leftarrow}{\mathrm{U}} \{M_1, \cdots, M_n\}$. $\mathcal{C}$ aborts the game if the adversary has a different xid* and/or the adversary does not impersonate M^*.

Game 2. Let l be the maximum number of sessions that the adversary can establish in the game. For $1 \leq j \leq l$, we verify or alter the related parameters of the session between the service provider and M^* up to the l-th session as per the following games:

- **Game $2-j-1$.** At the j-th session, $\mathcal{C}$ evaluates the output of the PUF implemented in M^*. $\mathcal{C}$ aborts the game if the output does not have enough entropy or if it is correlated to the other outputs derived from the inputs to the PUF.
- **Game $2-j-2$.** The output from the fuzzy extractor (k, hd) is turned into a random variable.
- **Game $2-j-3$.** In this game the output from the pseudorandom functions (PRF) $h(k, \cdot)$ and $h(K_{ms}, \cdot)$ is derived from a truly random function.
- **Game $2-j-4$.** In this game the resultant output from the PRF $h(K_{em}, \cdot)$ is obtained from a truly random function.
- **Game $2-j-5$.** In this game, we alter the XORed output $R^*_{i+1} = k \oplus R'_{i+1}$ and $hd^* = h(K_{ms}||n_m) \oplus hd$ to arbitrarily chosen $R^*_{i+1}, hd^* \underset{\leftarrow}{\mathrm{U}} \{0,1\}^{|R^*_{i+1}, hd^*|}$.

The main idea of the security proof is to modify the messages corresponding to the target smart meter M^* to arbitrary strings. The attacker wins the game and breaks the security of the proposed scheme if he/she can distinguish the random strings from real messages/outputs and/or convince the smart meter or service provider to accept the session while the communication is modified. We proceed with the game transformation starting with the first call of the smart meter M^*. After that, we gradually change the communication message from Game 2-j-1 to

Game 2-j-5. We move to the next section, once these transformations are finished. Here, we recursively apply this strategy up to the upper bound l on the number of sessions that the attacker can establish. Through these game transformations, we show that the advantage of the adversary against the authentication protocol can be limited to negligible values as shown in the results of Lemma 1 through 5.

Lemma 1 (Random Guessing): *If there are n smart meters, then $X_0 = nX_1$.*

Sub-Proof: We say that the adversary wins the game when there is a session which the service provider or smart meter accepts, while communication is modified by the adversary. Since we assume that there are at most n smart meters, therefore the probability that the challenger $\mathcal{C}$ can correctly guess the related session is $1/n$.

Lemma 2 (PUF Response): $X_1 = X_{2-j-1}$ *and* $X_{2-(j-1)-5} = X_{2-j-1}$ *for any* $2 \leq j \leq l$, *if the PUF used in the smart meters is a* $(d, n, l, \lambda, \epsilon_1)$*-secure PUF.*

Sub-Proof: Since the PUF used in the proposed protocol is $(d, n, l, \lambda, \epsilon_1)$-secure, it implies that its intra-distance is less than d, the inter-distance is larger than d, and the min-entropy of the PUF is lager than λ. Besides, the PUF also has the desirable property that even if the input to the PUF is exposed, the output derived from the PUF satisfies the sufficient min-entropy property and that makes each output uncorrelated. Here, the challenger does not check the entropy of the output in this game. Now, consider a scenario where an adversary issues the *reveal query* and obtains the stored information from the PUF's memory. In this regard, since X_1, X_{2-j-1} and $X_{2-(j-1)-5}$ use the $(d, n, l, \lambda, \epsilon_1)$-secure PUF, the distance between them is bounded by ϵ_1. Therefore, we can write $|X_1 - X_{2-j-1}| \leq \epsilon_1$ and $\left|X_{2-(j-1)-5} - X_{2-j-1}\right| \leq \epsilon_1$. This means there is no effect on the game transformation.

Lemma 3 (FE Output): *If the* FE *is a* (d, λ, ϵ_2)*-secure fuzzy extractor, then* $X_{2-j-1} = X_{2-j-2}$ *for any* $1 \leq j \leq l$.

Sub-Proof: As discussed, the fuzzy extractor is secure if the min-entropy of the PUF input R in the FE.Gen$(R) = (K, hd)$, is at least λ and K is statistically ϵ_2-close to a uniformly random variable in $\{0,1\}^k$, even if the helper data hd is disclosed. Now, since the PUF provides enought min-entropy λ, the property of the (d, λ, ϵ_2)-fuzzy extractor ensures that the output of the fuzzy extractor is close to a random string. Therefore, no adversary can distinguish the difference between the games X_{2-j-1} and X_{2-j-2}. Therefore, the advantage of the adversary in distinguishing the two games can be represented as $|X_{2-j-2} - X_{2-j-1}| \leq \epsilon_2$.

Lemma 4 (Authentication with Secure PRF): $\forall\ 1 \leq j \leq l$, $|X_{2-j-2} - X_{2-j-3}| \leq \mathrm{Adv}^{\mathrm{PRF}}_{h(.),\beta}(k)$, *where* $\mathrm{Adv}^{\mathrm{PRF}}_{h(.),\beta}(k)$ *denotes the advantage of* β *to break the security of the* PRF $h(\cdot)$.

Sub-Proof: If there is a difference between these games, then we can construct an algorithm β which breaks the security of the PRF $h(\cdot)$. β sets up all the security credentials and simulates

our protocol except the i-th session. β can access the real PRF $h(K,\cdot)$ or a truly random function. When the adversary invokes the i-th session, β sends $\{n_s^* \xleftarrow{U} \{0,1\}^k\}$ as the output of the service provider. When $\mathcal{A}$ sends $n_s^{\#}$ to the service provider, β continues the computations as per the protocol specifications and issues $n_s^{\#}$ to the oracle instead of the normal computation of $h(\cdot)$. Upon receiving V_1, β outputs $\{R_{i+1}^*, V_1\}$ as the response of the smart meter. When the adversary sends $\{R_{i+1}^{\#}, V_1^{\#}\}$, β issues $n_s^{\#}$ to the oracle and obtains V_1, which is used to authenticate the smart meter.

If β accesses the real PRF, then this simulation is similar to Game $2-j-2$. Otherwise, it can be argued that the oracle query invoked by β is completely random, where the distribution is equivalent to Game $2-j-3$. Therefore, we can write $|X_{2-j-2} - X_{2-j-3}| \leq \text{Adv}^{\text{PRF}}_{h(\cdot),\beta}(k)$.

Lemma 5 (Secure PRF): $\forall\, 1 \leq j \leq l$, $|X_{2-j-3} - X_{2-j-4}| \leq \text{Adv}^{\text{PRF}}_{h(\cdot),\beta}(k)$.

Sub-Proof: This lemma can be proved in a way similar to the proof for Lemma 4.

Lemma 6 (Random String): $\forall\ 1 \leq j \leq l$, $X_{2-j-2} = X_{2-j-4} = X_{2-j-5}$.

Sub-Proof: The fuzzy extractor FE and the PRF $h(\cdot)$ are already changed to the truly random function in the above games. Therefore, K and $h(K||n_s)$ are used as an effective one-time pad to encode $R_{i+1}^{'}$ and hd_i, respectively. Therefore, no adversary can differentiate $R_{i+1}^* = K \oplus R_{i+1}^{'}$ from a randomly chosen string.

Theorem 2 (Privacy): *Consider a $(d, n, l, \lambda, \epsilon_1)$-secure physically uncloneable function,* and let FE *be a (d, λ, ϵ_2)* fuzzy extractor, and let h be a ϵ_3-secure pseudorandom function. Then our protocol satisfies the indistinguishability-based privacy property.

Proof: The proof of this theorem is similar to that for Theorem 1. In Theorem 1, we have shown that that the proposed authentication protocol is secure against any forgery attacks. According to the game transformation described in the proof of Theorem 1, if we repeatedly modify the messages communicated for the smart meters M_0^* and M_1^*, then the entire transcript will be identical to random strings. Thus, no information that identifies the challenger's coin will be leaked. Also, all the parameters stored in the smart meter such as $\{\text{SID}, \text{PID}, (C, K), (C_{em}, K_{em})\}$ are randomly generated and each pair can only be used once. Hence, these parameters do not provide any information about the smart meter. The probability that the challenger can identify M_0^* and M_1^* so that the game transformation is finished within a polynomial time is $1/n^2$. Therefore, we can argue that our proposed scheme satisfies indistinguishability-based privacy.

5 Performance Analysis and Comparison

In order to demonstrate the performance of the proposed scheme, this section compares our proposed scheme with respect to the existing authentication schemes (Mohammadali [4], Wu and Zhou [5], Xia and Wang [6], Tsai and Lo [8], and Odelu et al. [9]) for smart grid communication.

Table 2: Performance Benchmarking based on Security Features

Schemes	**SF1**	**SF2**	**SF3**	**SF4**	**SF5**	**SF6**	**SF7**
Mohammadali et al. [4]	Yes	Yes	No	No	Yes	No	No
Wu and Zhou [5]	No	No	No	No	No	No	No
Xia and Wang [6]	No	No	Yes	No	No	No	No
Tsai and Lo [8]	Yes	Yes	Yes	Yes	No	No	No
Odelu et al. [9]	Yes	Yes	Yes	Yes	Yes	No	No
Proposed Scheme	Yes	Yes	Yes	Yes	Yes	Yes	Yes
SF1: Smart meter's privacy ; **SF2:** Protection against eavesdropper; **SF3:** Protection against MIM attacks;							
SF4: Forward secrecy; **SF5:** Session key security; **SF6:**Resilient against side-channel attacks**;**							
SF7: Physical security of the smart meter;							

Table 3: Performance Benchmarking based on Computation Cost

Schemes	**Smart Meter**	**Service Provider**
Mohammadali et al. [4]	$2T_{mp}+T_m+T_{cert_{gen}}+3T_h$	$3T_{mp}+T_m+T_{cert_{ver}}+4T_h$
Wu and Zhou [5]	$3T_{mp}+T_m+T_{cert_{gen}}+T_h$	$4T_{mp}+T_m+T_{cert_{ver}}+4T_h+T_s$
Xia and Wang [6]	T_s+4T_h	T_s+4t_h
Tsai and Lo [8]	$4T_{mp}+T_e+5T_h$	$3T_{mp}+T_e+2T_b+5T_h$
Odelu et al. [9]	$3T_{mp}+T_e+6T_h$	$2T_{mp}+T_e+2T_b+6T_h$
Proposed Scheme	FE.Gen + $6T_h+2T_{PUF}$	FE.Rec + $6T_h$
T_{mp}: Time for multiplication point operation; T_m: Time for multiplication operation;		
T_e:Time of a modular exponential operation; T_s:Time for symmetric encryption/decryption;		
T_b:Time for bilinear pairing;T_h: Time of a hash operation;		
T_{PUF}: Time for PUF operation;$T_{cert_{gen/ver}}$: Time for certificate generation/verification operation		

In this context, we first compare the performance our proposed scheme with respect to [4], [5], [6], [8] and [9] on the security front. In this context, we consider several security features such as the "privacy of the smart meter", protection against the "man-in-the-middle attacks", "physical security of the smart meters", etc. Table 2 shows that the authentication protocols presented in [4], [5], [6] and [8] fails to provide some of the imperative security features such as the session key security, privacy against eavesdropper, etc (as discussed in Section I). Although, the authentication scheme presented in [9] can ensure most of the security features. However, similar to the other existing schemes, it cannot guarantee the physical security of the smart meter. Therefore, these schemes ([4], [5], [6], [8] and [9]) allow the inside attackers (e.g. home users) to exploit this weakness and control the smart meter for their own profit. On the contrary, our proposed scheme can ensure all the imperative security features including the physical security of the smart meters. In our proposed scheme if a consumer tries to perform any physical tampering of the smart meter, then this will affect the PUF's behavior and during authentication process the service provider will be able to detect such issue.

Now, we compare our proposed scheme in terms of the computation cost. In general, for smart grid low cost devices are applied which have limited computation power and limited storage space. Thus, due to resource constraints in low cost devices, the authentication protocol must give priority to the efficiency. Table 3 shows the performance of the proposed scheme with related

Table 4: Execution Time of Various Cryptographic Operations

Operations	**Smart Meter**	**Service Provider**
T_{mp}	5.9 ms	2.6 ms
T_m	22.93 ms	14.5 ms
T_b	9.23 ms	3.78 ms
$T_{cert_{gen}}$	57.63 ms	-
$T_{cert_{ver}}$	-	17.24 ms
T_h	0.026 ms	0.011 ms
T_e	7.86 ms	2.34 ms
T_s	0.079 ms	0.041 ms
T_{PUF}	0. 12 ms	-
FE.Gen (.)	1.67 ms	-
FE.Rec (.)	-	2.85 ms

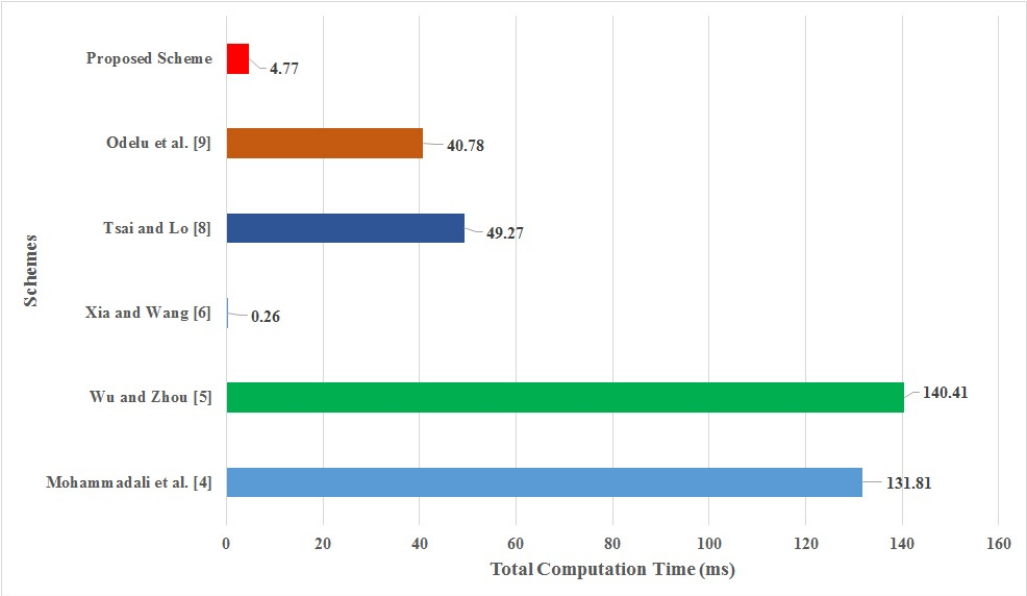

Figure 4: Performance Comparison based on Execution Time.

protocols in terms of the computation cost. Since, both the proposed scheme and the scheme presented in [3] are based on the symmetric key crypto system. Accordingly, they will incur cost less computation cost on as compared to others. Next, in order to demonstrate the efficiency of the proposed scheme, here we conducted simulations of the cryptographic operations used in the proposed authentication scheme and the schemes presented in [4], [5], [6], [8] and [9] on an Ubuntu 12.04 virtual machine with an Intel Core i5-4300 dual-core 2.60 GHz CPU (operating as the Service Provider as per the scheme). On the other hand, for simulating a smart meter, we use one core 798 MHz CPU and 256 MB of RAM, which is not far away from a real SM [17]. For evaluating the execution time of different cryptographic operations used in the proposed scheme and [4], [5], [6], [8] and [9], the simulation utilizes the JPBC library Pbc-05.14 (shown in Table 3). Now, to evaluate the T_{PUF} operation, here we consider a a 128-bit arbiter PUF circuit on an MSP430 Microcontroller machine with 798 MHz CPU. Besides, for FE.Gen and FE.Rec operations, we adopt the code-offset mechanism using BCH code [20]. For symmetric-key based

Table 5: Performance Benchmarking based on Communication Cost

Scheme	Cost at SM	Cost at SP	Total Comm. Cost	No. of Rounds
Mohammadali et al. [4]	928-bits	480-bits	1408-bits	4
Wu and Zhou [5]	1248-bits	672-bits	1920-bits	3
Xia and Wang [6]	1152-bits	480-bits	1632-bits	3
Tsai and Lo [8]	1248-bits	672-bits	1920-bits	3
Odelu et al. [9]	1020-bits	840-bits	1860-bits	3
Proposed Scheme	652-bits	320-bits	908-bits	3

encryption/decryption T_e here we consider the 256-bit AES-CBC encryption mode

Now, from Fig. 4, it is clear that the overall computation cost of the scheme presented in [3] is less than others. However, the scheme is insecure against several security threats (as shown in Table 1). On the other hand, our proposed scheme takes significantly less computational cost than[4], [5], [6], [8] and [9]. In addition, the proposed scheme can ensure all the important security features (including physical security of the smart meter) and hence suitable for the secure computation in smart grid. Table 4 shows the communication cost of the proposed scheme and others. From that, we can see that, our proposed scheme takes less communication cost as compared to others.

6 Conclusion

Smart grid has many connections with people's daily lives. Therefore, it is very important to solve any security challenges in smart grid in a fast manner. In order to that, this article proposes a secure and anonymous authentication scheme using PUF. Here we argue that, our proposed scheme can ensure higher level of security and also efficiency, which are greatly imperative for realiable smart grid comminication.

References

[1] International Energy Outlook 2007, Energy Information Administration, U.S. Department of Energy, (http://www.eia.doe.gov/oiaf/ieo/highlights.html).

[2] V. C. Gungor et al., "A Survey on Smart Grid Potential Applications and Communication Requirements," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 1, pp. 28-42, 2013.

[3] P. Gope et al., "Lightweight and Privacy-Friendly Spatial Data Aggregation for Secure Power Supply and Demand Management in Smart-Grids," *IEEE Transactions on Information Forensics & Security*, vol. 14, no. 6, pp. 1554-1566, 2019.

[4] Mohammadali, M. Sayad Haghighi, M. H. Tadayon, and A. Mohammadi Nodooshan, "A Novel Identity-Based Key Establishment Method for Advanced Metering Infrastructure in Smart Grid," IEEE Transactions on Smart Grid, pp. 1-1, 2016.

[5] D. Wu and C. Zhou, "Fault-tolerant and scalable key management for smart grid," *IEEE Trans. Smart Grid*, vol. 2 no. 2 pp. 371-378 Jun. 2011.

[6] J. Xia and Y. Wang, "Secure key distribution for the smart grid," IEEE Trans. Smart Grid, vol. 3 no. 3 pp. 1437-1443 Aug. 2012.

[7] J. H. Park, M. Kim, and D. Kwon, "Security weakness in the smart grid key distribution proposed by Xia and Wang" *IEEE Trans. Smart Grid*, vol. 4 no. 3 pp. 1613-1614 Sep. 2013.

[8] J.-L. Tsai and N.-W. Lo, "Secure Anonymous Key Distribution Scheme for Smart Grid," *IEEE Transactions on Smart Grid*, vol. 7, no. 2, pp. 906–914, 2016.

[9] V. Odelu, A. K. Das, M. Wazid, and M. Conti, "Provably Secure Authenticated Key Agreement Scheme for Smart Grid," *IEEE Transactions on Smart Grid*, 2016, DOI: 10.1109/TSG.2016.2602282.

[10] P. Gope and B. Sikdar, "Privacy-Aware Authenticated Key Agreement Scheme for Secure Smart Grid Communication *IEEE Transactions on Smart Grid*, vol. 10, no. 4, pp. 3953–3962, 2019.

[11] Y. Dodis, J. Katz, L. Reyzin, A. Smith, "Robust fuzzy extractors and authenticated key agreement from close secrets," In: Advances in Cryptology (CRYPTO), LNCS, vol. 4117, pp. 232–250. Springer 2006.

[12] Y. Dodis, L. Reyzin, A. Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," In: Advances in Cryptology (EUROCRYPT). LNCS, vol. 3027, pp. 523–540, 2004.

[13] X. Boyen, "Reusable cryptographic fuzzy extractors," In: ACM Conference on Computer and Communications Security (ACM CCS). pp. 82–91. ACM 2004.

[14] J. Delvaux, D. Gu, I. Verbauwhede, M. Hiller, and M-D Yu, "Efficient Fuzzy Extraction of PUF-Induced Secrets: Theory and Applications, " In: Cryptographic Hardware and Embedded Systems (CHES). LNCS vol. 8913 pp. 412-430, Springer 2016.

[15] G. Suh, S. Devadas, Physical unclonable functions for device authentication and secret key generation, in: Design Automation Conference, 2007, DAC '07, 44th ACM/IEEE, 2007, pp. 9–14.

[16] S. Guilley, and R. Pacalet, "SoCs security: a war against side-channels," Annals of Telecommunications, vol. 11.pp. 998-1009, 2004.

[17] Atmel's family of smart power meters. http://www.atmel.com/products/smart-energy/power-metering/ (accessed on 28 May 2017).

[18] Pbc library. Tech. rep. http://crypto.standford.edu/pbc/(accessed on 16 April 2017).

[19] Oracle Technology Network. Java Cryptography Architecture (JCA). [Online]. Available: http://docs.oracle.com/javase/6/docs/technotes/ guides/crypto/CrypoSpec.html, accessed Apr. 20, 2017.

[20] Y. Dodis et al., "Fuzzy extractors: How to generate strong keys from from biometrics and other noise data. SIAM J. Compt. vol. 38, no. 1, pp. 97-139, 2008.