Fault-Tolerant and Scalable Key Management for Smart Grid

Dapeng Wu and Chi Zhou

***Abstract*—In this paper, we study the problem of secure key management for smart grid. Since existing key management schemes are not suitable for deployment in smart grid, in this paper, we propose a novel key management scheme which combines symmetric key technique and elliptic curve public key technique. The symmetric key scheme is based on the Needham-Schroeder authentication protocol. We show that the known threats including the man-in-the-middle attack and the replay attack can be effectively eliminated under the proposed scheme. The advantages of the new key management scheme include strong security, scalability, fault-tolerance, accessibility and efficiency.**

***Index Terms*—Security, key management, fault tolerance, scalability.**

I. Introduction

The smart power grid is the convergence of information technology, communications and power system engineering to provide a more robust, efficient and flexible electrical power system [1]. The smart concept specifies the addition of bi-directional communication and intelligence to the power grid to facilitate real-time metering of customers, enable remote control of residential appliances via smart meters, and facilitate the wide-spread use of demand response programs permitting the utility to control consumer loads in order to reduce power system load [2]. They also transform the power grid into a bi-directional power system in which customers can supply as well as receive power from the grid, converting the grid into a distributed power generation system [3].

Smart power grids consist of sensing, communication, control and actuation systems which enable pervasive monitoring and control of the power grid [4]. Millions of smart metering devices are being deployed around the country to enable utilities to interact with consumers. These smart meters can monitor energy consumption in real-time, provide customers with real-time power pricing information, and perform automatic control on smart appliances to conserve energy [5]. While these devices promise to transform the electricity grid, they bring a plethora of security related problems which must be addressed in order to guarantee safe and secure grid operation [6], [7]. For example, the user data or occupant profiling may be maliciously collected via means of the electricity usage for misuse [7], [8], the communication system may be vulnerable to denial of service attacks [9], and security holes may exist which permit hacking into the smart meters to manipulate usage data.

This work was supported in part by the US Department of Energy under grant DE-FC26-08NT02875. Prof. Dapeng Wu is with Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611. Prof. Chi Zhou is with Department of Electrical and Computer Engineering, Illinois Institute of Technology, Chicago, IL; address: Siegel Hall 130, 3301 South Dearborn, Chicago, IL 60616. Correspondence author: Prof. Dapeng Wu, wu@ece.ufl.edu, http://www.wu.ece.ufl.edu.

Security is crucial to important infrastructures such as power grid. It is known that security breaches of power grid computer systems may have devastating effects and the likelihood of such breaches is rising as the power grid is increasingly relying on complex interconnected computer networks. This will pose a significant challenge on security system design for smart grid [10]. To the best of our knowledge, none of the existing authentication scheme meets the requirements of a smart grid, and existing authentication solutions are almost exclusively based on commercial-of-the-shelf (COTS) components or simply rely on the security infrastructure of Internet (c.f. [11]). However, known methods for securing computer networks or Internet may not be sufficient due to lack of necessary fail-safe mechanism or being prone to denial of service (DoS) attack, among others. For example, in the widely adopted Kerberos scheme [12], [13], for one client, the validity of a session key is only controlled by a timestamp; multiple sessions of the same client may share the same session key; a compromised session key will render a sequence of successive sessions vulnerable.

In this paper, we secure a smart power grid by forming interconnected trust realms. Four types of principals are needed in a realm as follows:

(1) Trust anchors, which manage key distribution in a realm; each trust anchor has one public key and one private key.
(2) Data aggregators, which are agents being able to perform complex data processing tasks; each data aggregator has a certified public key and a private key for data communications.
(3) Data collectors, which are data collecting and sensing agents; each data collector has a certified public key and a private key for communicating to other principals in a realm.
(4) Sensors, which are low-power devices for data gathering; each sensor has a smart card which contains two certificates for trust delegation issued by trust anchors, and these certificates facilitate efficient secure communications from sensors to other principals in a realm.

In this paper, a public-key and symmetric key combined approach is proposed for simplicity and scalability of key management as well as other desirable properties. The symmetric key scheme is based on the Needham-Schroeder authentication protocol, and the public key scheme is based on elliptic curve cryptography for high efficiency and strong security. The use of public keys also has a nice property that no

static symmetric key is needed between data aggregators and collectors; this eliminates the possibility that symmetric keys could be compromised, and it also avoids the overhead of managing symmetric keys.

Since in Needham-Schroeder authentication protocol [14], the only-once semantics on involved messages is crucial to countermeasure the replay attacks [13], message timestamping or the use of nonce is normally considered sufficient against replay attack [15]. However, on one hand, real-time clocks in most low-power sensors have some intrinsic drift which cannot be synchronized within sufficient accuracy for the authentication protocol; on the other hand, as network bandwidth increases, messages in these protocols and schemes take less time to transmit. These two factors make opportunistic replay attacks highly possible. In this paper, methods combining both timestamping and once-only nonce are developed to countermeasure replay attacks on ZigBee wireless sensors. Harware-assisted implementation is proposed to ensure the once-only semantics of messages. It countermeasures this opportunistic replay attack under real-time constraints in a practical environment without trading off the performance.

Clock drift between a transmitting wireless device and a receiving device is a common problem in wireless networks. For low-power wireless devices such as the ZigBee sensors used in smart grid, this problem is further magnified by the aging of crystal (which is the essential part of a real-time clock (RTC) circuit), ambient temperature variation, stray load capacitance, etc. For example, in a typical 32.768 kHz RTC oscillator, this clock drift can be more than 20 parts-per-million (ppm), which amounts to at least 1.7 seconds per day. On one hand, when the end-to-end transmission time of a message becomes shorter due to new wireless technologies, network synchronization accuracy needs to be stringent since timestamping would be much less effective when the synchronization accuracy is close to the message end-to-end delay. On the other hand, due to clock drift variation, network synchronization needs to be performed periodically, but under periodic network synchronization, the mechanism, which uses timestamps as nonces, will violate the once-only semantics; furthermore, it may not be able to provide sufficient guarantee on message freshness. These are caused by the fact that at a message originator, the time itself is not a monotonic function.

Figure 1 shows an example that an adversary's replay message is correctly identified as stale when the collector and a sensor is perfectly synchronized. In Figure 1, **S** represents a sensor, **A** represents an adversary, and **C** represents a collector. In this example, it is assumed that the end-to-end delay of a message from **S** to **C** is 50 μs, and **C** knows that the end-to-end delay is 50 μs; the delay of 50 μs is due to transceivers' circuitry latency, buffering latency in the nodes on the transmission path, among others. As shown in Figure 1, a message is transmitted by **S** at epoch 0 μs and has a timestamp of 0 μs; the message arrives at **C** at epoch 50 μs; since the timestamp of 0 μs plus the delay of 50 μs is not less than the current time at **C**, **C** declares the message from **S** as 'fresh'. At epoch 25 μs, **A** launches a replay attack by sending the eavesdropped message from **S**, which has a timestamp of 0 μs. Assume that the end-to-end delay of a message from **A** to **C** is also 50 μs since **A** needs to be close to **S** in order to eavesdrop messages from **S**. Then the replay message arrives at **C** at epoch 75 μs. Since the timestamp of 0 μs plus the delay of 50 μs is less than the current time at **C**, i.e., the replay message is expected to arrive at epoch 50 μs but the current time is 75 μs, hence **C** declares the replay message as 'stale'. So the replay attack fails.

Figure 2 shows an example that an adversary can exploit the clock drift and launch an effective replay attack when the clock of a sensor **S** leads the clock of a collector **C** by 50 μs. As shown in Figure 2, a message is transmitted by **S** at epoch 50 μs (with respect to the clock of **S**) and has a timestamp of 50 μs; the message arrives at **C** at epoch 50 μs (with respect to the clock of **C**); since the timestamp of 50 μs plus the delay of 50 μs is not less than the current time at **C**, **C** declares the message from **S** as 'fresh'. At epoch 75 μs (with respect to the clock of **S**), **A** launches a replay attack by sending the eavesdropped message from **S**, which has a timestamp of 50 μs. Then the replay message arrives at **C** at epoch 75 μs (with respect to the clock of **C**). Since the timestamp of 50 μs plus the delay of 50 μs is not less than the current time at **C**, hence **C** declares the replay message as 'fresh'. So the replay attack succeeds.

It is known that the ideal case to ensure this once-only semantics is to use a purely random bit sequence as a nonce. The practical compromise is to use a pseudo-random bit sequence for a nonce. However, it is virtually impossible to require a low-power receiver to verify a nonce for the fulfillment of this semantics against all used nonces in real-time in a large smart grid network. In this paper, we will demonstrate that it is feasible to ensure this once-only semantics at a receiver by combining timestamp and nonce using a pseudo-random bit-sequence at a message originator. Since the message originator can simply generate a pseudo-random number with a sufficient number of bits, a nonce collision between two instances is highly unlikely at the message source; for this reason, we focus on ensuring the once-only semantics on the message receiver's side. Next, we present our proposed key management for smart grid.

II. Proposed Key Management for Smart Grid

In this section, we propose an authentication scheme, which applies an elliptic curve public key cryptography to the Needham-Schroeder protocol. Via a trust anchor, the public key method is employed to establish symmetric keys for agents to communicate with each other. It uses a trust delegation mechanism for sensors to access the local grid via agents. The public key based delegation key can be fast verified by a collector so that erroneous requests from DoS attackers can be filtered out at the grid entry points. Sensors and all agents are issued with a private key and a certificate of a public key by the trust anchors during the initial security setup. This significantly simplifies the key management even in a large smart grid. For cross realm secure access, data aggregator sends a request to a local trust anchor for a session key for communicating to a remote data aggregator, and the trust anchor in the remote realm will instead issue the actual session key. All session

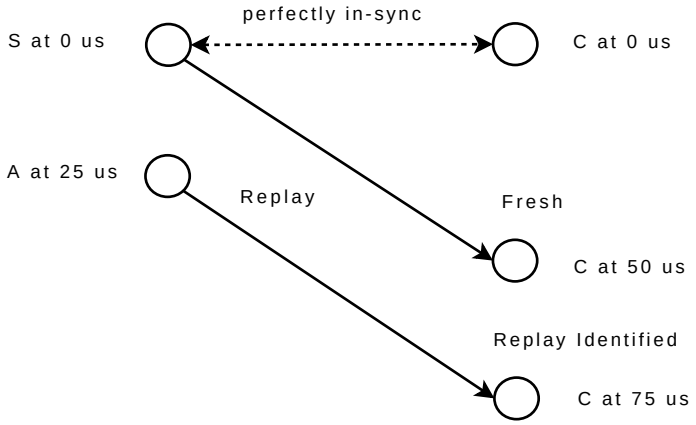


Fig. 1. Unsuccessful replay attack in a perfectly synchronized network

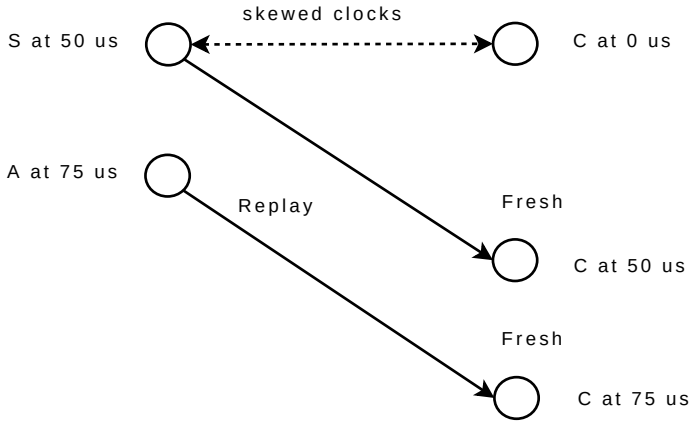


Fig. 2. Successful replay attack in a out-of-sync network

TABLE I
NOTATION AND ACRONYMS

E/F	: additive group derived from E and F with respect to T for a cryptographic use
p	: the largest prime factor of the order of T, non-smooth and of length at least 163 bits
Z_x^*	: a cyclic group of order $x-1$ for prime number x
$\uplus$	: a point addition operator in E/F
xT	: a scalar point multiplication of $x \in Z_p^*$ to T in E/F
$h(\cdot)$	: a collision resistant one-way hash function from Z_p^* to Z_p^*
m_w	: a warrant containing its generator's restrictions imposed on the delegation holder
$\mid$ (or ',')	: concatenation operators of two bit strings whenever the context is clear
TA	: trust anchor (local or cross-realm)
IDTA	: Identity (a number in Z_p^*) of a trust anchor
C	: Data Collector
IDC	Identity (a number in Z_p^*) of a data collector
S	: Smart sensor
IDS	: Identity (a number in Z_p^*) of a smart sensor
$K_{(C,TA)}$	: a session key between **C** (with IDC) and **TA** (with IDTA)
$\{x\}$	: a message labelled by x
ts	: timestamp
ck	: a symmetric communication key used for message encryption and decryption
$\mathcal{N}$	: nonce, which is a random number used at most once
T_{exp}	: expiration time of a session key
$[m]_K$	: a message 'm' enciphered under a symmetric key K
$\Pi(\cdot)$	: a point representation function: $E/F \mapsto Z_p^*$
$[x \mapsto y, \{z\}]$	: x sends y Message {z}

keys for agents are only for per-session use in the proposed scheme. (In this paper, a session is meant to be the message flow resulted from a single triggering event, e.g., a change of bus load, or a data refresh timeout.) Hence, this scheme scales well, and its key management is efficient since the number of keys to be managed is linearly related to the number of principals.

As shown in Figure 3, our proposed scheme consists of the following three components:

(A) Mutual authentication between a collector and an aggregator.
The message A sequence (i.e., Message A1 to Message A6) is used for mutual authentication between a collector and an aggregator. As shown in Figure 3, a collector initiates the authentication process by sending Message A1 to a trust anchor. Then, the trust anchor sends Message A2 (containing a symmetric key) to the collector. To obtain a session key, the same procedure as that in the Needham-Schroeder protocol is used: first, the collector sends Message A3 to the trust anchor; then, the trust anchor replies the collector by Message A4; the collector sends Message A5 to an aggregator; the aggregator replies the collector by Message A6.

(B) Mutual authentication between aggregators across realms.
The message B sequence (i.e., Message B1 to Message B6) is used for mutual authentication between aggregators across realms. As shown in Figure 3, an aggregator initiates the authentication process by sending Message B1 to a trust anchor in the same realm. Then, the trust anchor sends Message B2 (containing a symmetric key) to the aggregator. To obtain a session key, the same procedure as that in the Needham-Schroeder protocol is used: the aggregator sends Message B3 to a trust anchor in another realm; the trust anchor replies the aggregator by Message B4; the aggregator sends Message B5 to an aggregator in another realm; the aggregator in another realm replies the originating aggregator by Message B6.

(C) Mutual authentication between a sensor and a collector.
The message C sequence (i.e., Message C1 to Message C6) is used for mutual authentication between a sensor and a collector. Messages C1 and C6 are used for trust delegation request and verification. Messages C2 and C3 are used to acquire a symmetric key. Messages C4 and C5 are used for the session key delivery under Protocol 2 in Appendix B [16].

In all the three aforementioned components, messages (A1 and A2, B1 and B2, C2 and C3) are designed to generate a symmetric key using the public key cryptography protocol [17]. Then the symmetric key, which is generated on-the-fly, will be used in the successive Needham-Schroeder authentication protocol. Such a mechanism has a few advantages as follows:

(i) The key management at trust anchors is significantly simplified since there is no need to maintain shared symmetric keys.
(ii) A fast response on key request can be assured since a trust anchor can assign another lightly loaded trust anchor to issue a session key.
(iii) A high level of fault tolerance can be achieved since another trust anchor can be designated, should one of them fail.

Messages A3 to A6, and Messages B3 to B6 are generated by the Needham-Schroeder protocol for obtaining a session key. In addition, to support scalability and robustness, two messages are added to the original Needham-Schroeder protocol, which is similar to what is performed in the well-known Kerberos protocol. The only difference is that these two messages employ the public key infrastructure to generate the session key between a trust anchor and another principal (e.g., collector, aggregator) rather than depending on a preselected long-term shared symmetric key as in the Kerberos protocol. For a principal (denoted by P) such as a collector or an aggregator, the following two steps are involved in the public key based initialization:

- P sends a message, which contains 1) a public key certificate of P, 2) a token (denoted by T1) encrypted by the private key of P, and 3) a nonce n_1. This message is time-stamped by P. In addition, the token T1 contains a timestamp, another nonce n_2, and P's identity.
- Upon receiving the message from P, TA generates a new session key, and a secret key k, and sends a message, which contains 1) TA's public key certificate, 2) a token (denoted by T2) encrypted by P's public key, and 3) another token (denoted by T3) encrypted by k. The token T2 contains TA's public key certificate, and a message encrypted by TA's private key; the encrypted message contains k, n_2, TA's identity, and a timestamp. The token T3 contains the session key, n_1, a timestamp, session key expiration time.

Once a session key is obtained, the principal can further request a communication key from a trust anchor using the standard Needham-Schroeder protocol. Messages A3 to A6 and Messages B3 to B6 in Figure 3 follow the standard Needham-Schroeder protocol. Note that Token T1 and T2 have an identity attached; then by following the arguments in Section III of Ref. [18], this protocol belongs to the category of a name-stamp protocol; hence this protocol is secure against the man-in-the-middle attack.

We next give a brief review on trust delegation on smart sensors [16], and refer to [19], [20], [21], [22] for additional details on trust proxy, trust delegation for the purpose of authentication. Let Y be the certified public key of a trust anchor whose private key is $x \in Z_p^*$ and $Y = xT \in E/F$, and denote the identity of a sensor by IDS, and that of a collector by IDC, and that of a trust anchor by IDTA. In the rest of the paper, we will use sensor or its identity interchangeably; the same is true for a collector or a trust anchor. All notations and acronyms are listed in Table I. The additional public information Γ and the shared secret σ is generated and verified by Protocol 1 (see Appendix A).

As shown in Figure 4, when the session key $K_{(C,TA)}$ is created in advance before the authentication process, the proposed protocol generates four messages, denoted by $\{$C1$\}$,

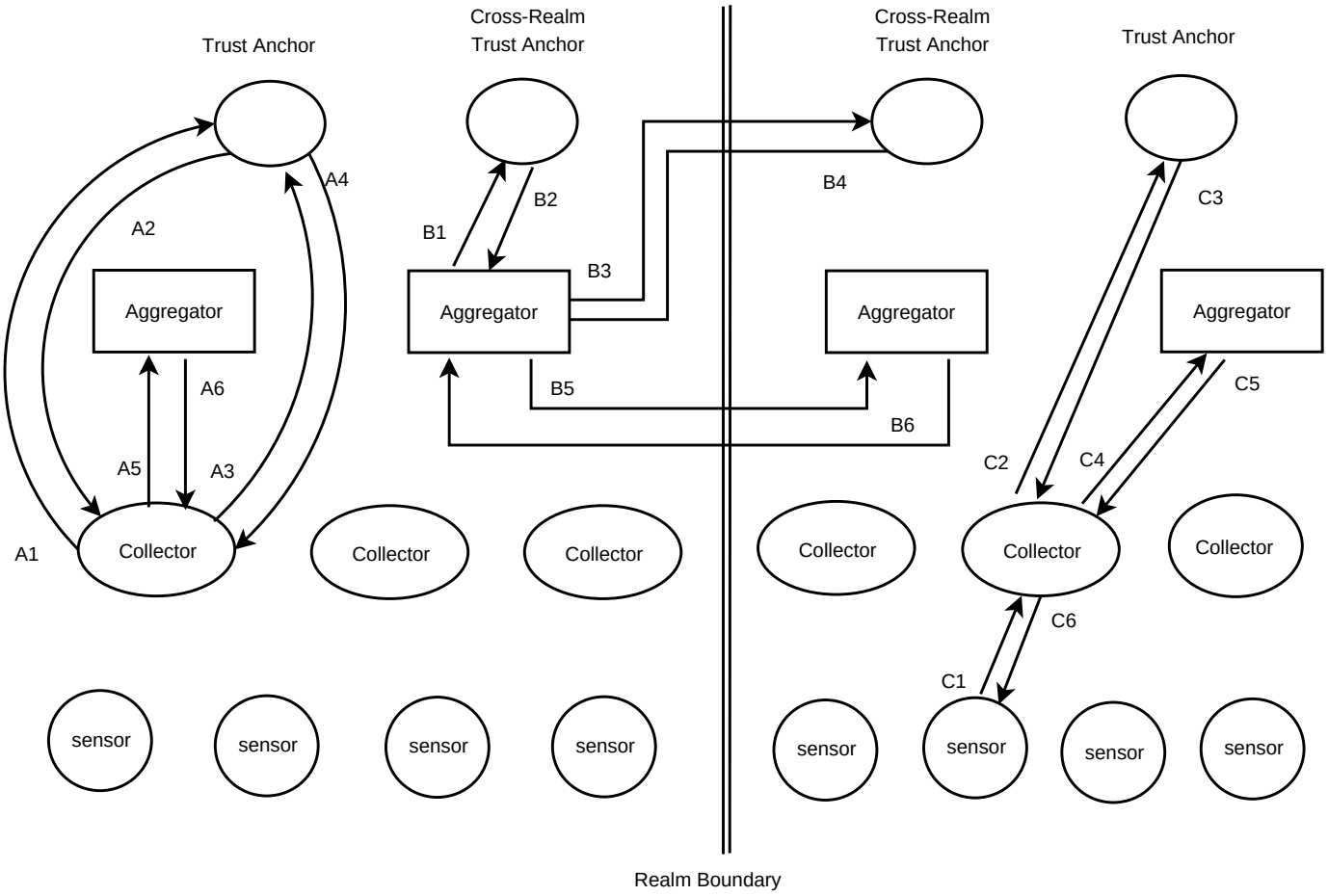
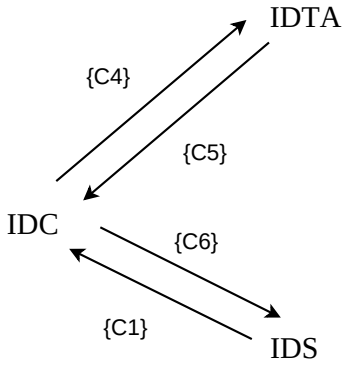


Fig. 3. Messages in the proposed scheme



$\mathcal{T}_{C,S} =$ IDC, nonce
{C1}:$m_w, R, s,$ IDTA, $[ck, ts, T_{exp},$ nonce$]_\sigma$, nonce
{C4}:IDS, $[ck, ts, T_{exp},$ nonce$]_\sigma$
{C5}:[IDS, $T_{exp}, ts, ck,$ nonce$]_{K_{(C,TA)}}, [\mathcal{T}_{C,S}]_\sigma$
{C6}:IDTA, [nonce, IDC, $[\mathcal{T}_{C,S}]_\sigma]_{ck}$

Fig. 4. Messages in sensor trust delegation

{C4}, {C5}, {C6} in Figure 4. Message {C1} is used for 1) the request for communicating with IDC, and 2) the IDS's authentication to IDC via trust delegation. Message {C4} is a request to IDTA for the communication key with IDS. Message {C5} is used to deliver the communication key back to IDC. Message {C6} authenticates IDC to IDS. In this way, IDS and IDC achieve mutual authentication between them. The authentication part of trust delegation is called ESA and presented in Appendix B as Protocol 2. Mutual authentication between IDS and IDTA is provided in the sensor trust delegation scheme shown in Figure 4, provided that IDC and IDTA are mutually authenticated in advance (cf. Proposition 2 of Ref. [16]). There are one transmission and one reception needed on IDS in ESA and each message length is in $O(\log(p))$. IDS needs to perform only one point scalar multiplication in actual authentication process and this is desirable for a low-power sensor.

For a timestamp based scheme, an opportunistic replay attacker can exploit the following simple scheme and be successful with non-negligible probability: whenever it sees a new message a sensor sends, it replaces the sender and the nonce in the message by the adversary's address and a new random nonce and re-sends the modified message as if it were the session initiator. Given the clock drift and periodical clock readjustment at the sensor, a perfect condition

could occur when the compromised message is taken as a refresh as illustrated in Figure 2. In the proposed scheme, both timestamps and nonces are used for key validity and nonce checks. Upon reception of a message, a collector performs the following procedures:

(1) It extracts the newly arrived nonce, time-stamps it, and saves it into the used nonce list.
(2) It checks the nonce list, and any nonce whose timestamp expires is removed from the nonce list.
(3) It checks the timestamp on the newly arrived message, if it expires, this message is a replay.
(4) It checks the nonce against the nonce list, if there is a collision, the message is a replay.

As for possible performance degradation, field programmable gate array (FPGA) based look-up table mechanism is used to speed-up the nonce check. This nonce check is against a much shorter list than otherwise would be needed. Taking into account the shortened list, this speed-up mechanism should be sufficient in practice.

The security of the proposed scheme is built upon the foundation of a public key infrastructure and the secure Needham-Schroeder authentication protocol. The known threats to the scheme including the man-in-the-middle attack and the replay attack have been shown to be effectively eliminated. Additional vulnerabilities on session keys and communication keys are addressed via techniques including a strict one-time use rule and on-the-fly key generation.

Since a trust anchor can designate any other legitimate trust anchor for the actual authentication on agents, a greater accessibility and scalability is ensured. To further improve fault tolerance, a primary delegation key and a secondary delegation key are preloaded into the smart card of a sensor. Should the primary trust anchor fail, the sensor will use the secondary trust anchor. By these mechanisms, a higher degree of fault tolerance is ensured.

The proposed scheme is also highly efficient and scalable due to the following properties:

1) One transmission and one reception on the low-power sensors for a mutual authentication between a sensor and an aggregator. Message size on these exchanges are around 100 bytes using a key representing an elliptic curve point under elliptic curve cryptography.
2) Messages on Needham-Schroeder authentication protocol can be redirected to any other trust anchor within a realm or cross realm; this will yield denial-of-service attack much less effective.
3) Any principals except trust anchors need to have an independent public/private keys or certificates, which enables the key management highly scalable and simplifies the key management. For example, to add an aggregator to the system, from the security perspective, one certificate and one private key are all what is needed; and to remove a fault aggregator from the system, what is only needed is to disqualify that certificate.
4) Securing a whole smart grid is equivalent to securing interconnected trust anchors, which are much smaller in number and are easier to fortify.

III. CONCLUSIONS

In this paper, we have studied the requirements on key management for smart grid. A key management scheme is proposed for its use in smart grid and it meets these requirements. The security of the proposed scheme is built on the foundation of a public key infrastructure and the secure Needham-Schroeder authentication protocol. We show that the known threats including the man-in-the-middle attack and the replay attack can be effectively eliminated under the proposed scheme. We also address the issue of additional vulnerabilities on session keys and communication keys via techniques including a strict one-time use rule and on-the fly key generation. The advantages of the new key management scheme include strong security, scalability, fault-tolerance, accessibility and efficiency.

APPENDIX A: TDI - TRUST DELEGATION INITIALIZATION FOR SENSORS

Denote the identity of a sensor by IDS; denote the identity of a collector by IDC; and denote the identity of a trust anchor by IDTA. In what follows, we will use sensor or its identity interchangeably; the same is true for a collector or a trust anchor. Our delegation initialization protocol is given as follows.

Protocol 1: TDI

1. [At a trust anchor with identity IDTA] The trust anchor performs the following steps:
   - Set key usage restrictions for the sensor with its identity IDS, and put the key usage restrictions in a warrant m_w.
   - Convert $(\text{IDS}|m_w)$ to an element in Z_p^*, and compute $h(\text{IDS}|m_w)$.
   - Select a random number $\kappa \in Z_p^*$, and produce (Γ, σ) (where $\Gamma \in E/F$ and $\sigma \in Z_p^*$) as follows:

$$\Gamma = (h(\text{IDS}|m_w)T) \uplus (\kappa T) \text{ (in } E/F) \quad (1)$$

$$\sigma = -xh(\Pi(\Gamma)) - \kappa \text{ (in } Z_p^*) \quad (2)$$

   where $h(\Pi(\Gamma))$ in (2) is performed in Z_p^* after the mapping on an appropriate point representation of Γ.
   - Put $(\Gamma, \text{IDS}, m_w)$ in public.
   - Deliver (σ, m_w) to the sensor securely.
2. [At a sensor with identity IDS] The sensor accepts the delegation key σ if (3) holds.

$$h(\text{IDS}|m_w)T = (\sigma T) \uplus (h(\Pi(\Gamma))Y) \uplus \Gamma \quad (3)$$

   where (3) is evaluated in E/F. ■

APPENDIX B: ESA - EFFICIENT SENSOR AUTHENTICATION

Under the same notations as in Protocol 1, we have the following authentication protocol for sensors.

Protocol 2: ESA

1. [At a sensor with identity IDS]: The sensor picks two random numbers $k \in Z_p^*$ and $\mathcal{N} \in Z_p^*$, and generates the communication key ck (upon one session use or timing

based invalidation), then computes R and s as in (4) and (5), respectively.

$$R = kT \qquad (\text{in } E/F) \qquad (4)$$

$$s = \sigma - kh(\Pi(R)|\mathcal{N}) \qquad (\text{in } Z_p^*) \qquad (5)$$

- The sensor generates a certificate $[ck, ts, T_{exp}, \mathcal{N}]_\sigma$ and then composes $\{\text{C1}\}$ as shown in Fig. 4.
- [IDS $\mapsto$ IDC, $\{\text{C1}\}$]: Sensor IDS initiates Protocol 2 by sending $\{\text{C1}\}$ to Collector IDC.
- [IDC $\mapsto$ IDS, $\{\text{C6}\}$]: Collector IDC sends $\{\text{C6}\}$ to Sensor IDS; Sensor IDS decodes $\{\text{C6}\}$ for IDC, $\mathcal{N}$, and checks if nonce is consistent.

2. [At a collector with identity IDC]: Upon receipt of message $\{\text{C1}\}$, Collector IDC checks warrant m_w for restrictions and verifies if (6) holds.

$$(sT) \uplus \Gamma \uplus (h(\Pi(\Gamma))Y) \uplus (h(\Pi(R)|\mathcal{N})R) = h(\text{IDS}|m_w)T \qquad (6)$$

- Collector IDC composes $\{\text{C4}\}$ on receipt of $\{\text{C1}\}$, and composes $\{\text{C6}\}$ on receipt of $\{\text{C5}\}$.
- [IDC $\mapsto$ IDTA, $\{\text{C4}\}$]: Collector IDC sends a request $\{\text{C4}\}$ to Trust Anchor IDTA for a communication key of Sensor IDS.
- [IDTA $\mapsto$ IDC, $\{\text{C5}\}$]: Trust Anchor IDTA sends $\{\text{C5}\}$ to Collector IDC; Collector IDC decodes $\{\text{C5}\}$ for ck, and checks expiration timestamp and consistence of nonce.
- [IDC $\mapsto$ IDS, $\{\text{C6}\}$]: Collector IDC authenticates Sensor IDS by sending $\{\text{C6}\}$ which is encrypted by the communication key ck and can be decrypted by Sensor IDS.

3. [At a trust anchor with identity IDTA]:
   - [IDC $\mapsto$ IDTA, $\{\text{C4}\}$]: Trust Anchor IDTA processes $\{\text{C4}\}$ using σ, then retrieves $K_{(C,TA)}$ and validates restrictions on m_w (saved copy at Trust Anchor IDTA for IDS during parameter generation phase) of IDS.
   - Trust Anchor IDTA composes $\{\text{C5}\}$ using σ and $K_{(C,TA)}$.
   - [IDTA $\mapsto$ IDC, $\{\text{C5}\}$]: Trust Anchor IDTA forwards the communication key contained in $\{\text{C5}\}$ to Collector IDC. ■

REFERENCES

[1] H. Farhangi, "The path of the smart grid," *IEEE Power and Energy Magazine*, vol. 8, pp. 18–28, Feb. 2010.

[2] United States Department of Energy, "The Smart Grid: An Introduction," in *Whitepaper*, 2008.

[3] S. Amin, "For the Good of the Grid," *IEEE Power and Energy Magazine*, vol. 6, pp. 48–59, 2006.

[4] E. Santacana, G. Rackliffe, Le Tang, and Xiaoming Feng, "Getting Smart," *IEEE Power and Energy Magazine*, vol. 8, pp. 41–48, 2010.

[5] Y. Strengers, "Smart metering demand management programs: challenging the comfort and cleanliness habitus of households," in *Proceedings of the 20th Australasian Conference on Computer-Human Interaction: Designing for Habitus and Habitat*, vol. 8, 2010, pp. 41–48.

[6] National Institute of Standards and Technology, "Guidelines for Smart Grid Cyber Security: Vol. 3, Supportive Analyses and References," *IEEE Power and Energy Magazine*, vol. , 2010.

[7] P. McDaniel and S. McLaughlin, "Security and Privacy Challenges in the Smart Grid," *IEEE Security and Privacy*, vol. 7, pp. 75–77, 2009.

[8] Z. Fan, G. Kalogridis, C. Efthymiou, M. Sooriyabandara, M. Serizawa, and J. McGeehan, "The new frontier of communications research: smart grid and smart metering," in *e-Energy '10: Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking*, New York, NY, USA, 2010, pp. 115–118.

[9] T. Flick, "Hacking the Smart Grid," in *Black Hat USA*, Las Vegas, NV, 2009.

[10] H. Khurana, M. Hadley, N. Liu, D. A. Frincke, "Smart-Grid Security Issues," *IEEE Security and Privacy*, vol. 8, no. 1, pp. 81–85, Jan. 2010.

[11] H. Khurana, R. Bobba, T. Yardley, P. Agarwal, and E. Heine, "Design Principles for Power Grid Cyber-Infrastructure Authentication Protocols," in *43rd Hawaii Intl. Conf. on System Sciences*, Jan. 2010.

[12] B. C. Neuman and T. Ts'o, "Kerberos: an authentication service for computer networks," *IEEE Communications Mag.*, vol. 32, no. 9, pp. 33–38, 1994.

[13] S. M. Bellovin and M. Merritt, "Limitations of the Kerberos Authentication System," *ACM Computer Communication Review*, vol. 20, no. 5, pp. 119–132, Oct. 1990.

[14] R. M. Needham and M. D. Schroeder, "Using encryption for authentication in large networks of computers," *Communications ACM*, vol. 21, no. 12, pp. 993–999, Dec. 1978.

[15] D. E. Denning and G. M. Sacco, "Timestamps in key distribution protocols," *Communications ACM*, vol. 24, no. 8, pp. 533–536, Aug. 1981.

[16] C. Tang and D. O. Wu, "An Efficient Mobile Authentication Scheme for Wireless Networks," *IEEE Trans. Wireless Comm.*, vol. 7, no. 4, pp. 1408–1416, April 2008.

[17] I. Cervesato, A.D. Jaggard, A. Scedrov, J.-K. Tsay, C. Walstad, "Breaking and Fixing Public-Key Kerberos," *Information and Computation*, vol. 206, no. 2-4, pp. 402–424, Feb.-April 2008.

[18] D. Dolev and A. C. Yao, "On the security of public key protocols," *IEEE Trans. Inform. Theory*, vol. 29, no. 2, pp. 198–208, Mar. 1983.

[19] M. Mambo and K. Usuda and E. Okamoto, "Proxy signatures for delegating signing operation," in *3rd ACM CCS*, 1996, pp. 48–57.

[20] R. Molva, D. Samfat and G. Tsudik, "Authentication of mobile users," *IEEE Network, Special Issue on Mobile Communications*, vol. 8, no. 2, pp. 26–34, 1994.

[21] W.-B. Lee and C.-K. Yeh, "A new delegation-based authentication protocol for use in portable communication systems," *IEEE Trans. Wireless Comm.*, vol. 4, no. 1, pp. 57–64, Jan. 2005.

[22] B. Lee, H. Kim and K. Kim, "Secure mobile agent using strong non-designated proxy signature," *LNCS 2119,Springer-Verlag*, vol. , pp. 474–486, 2001.