Secure Authenticated Key Exchange with Revocation for Smart Grid

Fangming Zhao, Yoshikazu Hanatani, Yuichi Komano (*Member* IEEE),
Ben Smyth, Satoshi Ito, Toru Kambayashi
Corporate Research & Development Center, Toshiba Corporation, Saiwai-ku, Kawasaki, Japan
Email: {fangming.zhao, yoshikazu.hanatani, yuichi1.komano, satoshi.ito, tooru.kamibayashi}@toshiba.co.jp,
research@bensmyth.com

***Abstract*—Using cryptographic technologies to provide security solutions in smart grid is extensively discussed in NISTIR 7628 [1] and IEC 62351 standards series [2]. Both series identify cryptographic key management for Intelligent Electronic Devices (IEDs) communication as one of the most important issues. In this paper, considering the system constraints and the security requirements in the smart grid, we propose an authenticated key exchange scheme with revocation by exploiting a well-known cryptographic protocol: Broadcast encryption [3], [11], [12] using a media key block(MKB) [15]. Furthermore, we show that our scheme is efficient in comparison with the PKI-signature based Internet Key Exchange(IKE) protocol [4], [8] in terms of the following points of view: (1) communication cost; (2) compuation cost; (3) device revocation cost. The comparison results show that our scheme is efficient and cost-effective in most cases for devices and systems in smart grid.**

I. Introduction

A. Background and Motivation

Efficient and secure use and management of energy resources is becoming increasingly important in power systems. A smart grid, which uses intelligent transmission and distribution networks to efficiently deliver electricity and manage energy usage, is attracting attention from both the IT industry and electricity providers. Smart grid aims to improve the electric system's reliability, security, and efficiency through two-way communication of consumption data and dynamic optimization of power system operations, maintenance, and planning. In addition to the closed proprietary networks used by conventional power control systems, smart grid systems use the Internet to transmit commands and gather information from remote units. However, this shift also introduces a vulnerability: attackers can read and manipulate control signals sent on the Internet and potentially have the ability to cause blackouts. Moreover, private information from the smart devices in the power grid may also be leaked by crackers.

To overcome this limitation, secure network communication and secure data transmission must be achieved between any two electrical devices in the power control system. Using cryptographic technologies to provide security solutions for these problems in smart grid is widely discussed in NISTIR 7628 [1](the foundation document for the architecture of the US smart grid) and IEC 62351 standards series [2]. Both series identify cryptographic key management for substation communication as an important research topic. This requires the distribution and management of keys to intelligent electronic devices(IEDs). In addition, we must develop secure protocols for authentication, communication, and key revocation.

B. Challenging Issues

As discussed in both NISTIR 7628 [1] and IEC 62351 standards series [2], cryptographic key management should considers asymmetric keys (e.g. PKI: Public key infrastructure) or on symmetric keys (e.g. Pre-shared-key/secret-key). Asymmetric key management mechanisms typically use X.509 certificates [5], [6]. However, this solution has a high computation and communication cost for the client(electric device, e.g. a smart meter). For example, in the Internet Key Exchange (IKE) protocol [4], [8], the participant must verify the digital signature in the public key certificate to ensure their interlocutor's public key, check the certificate's validity period and ensure the key is not on a Certificate Revocation List (CRL) each time from a trusted Certificate Authorities (CA) to verify the interlocutor's identification.

By comparison, symmetric key mechanisms are more efficient, but may assume the existence of a pre-shared key. In these schemes care should be taken to protect against "break-once break-everywhere", due to the use of one secret key or a common credential across the entire infrastructure [1]. In this symmetric key based system, after any device is compromised, revoking that device and reconstructing a new secret key for other valid devices is the first to be undertaken by electric power utilities. However, key revocation is not considered by NISTIR 7628 [1] and IEC 62351 standards series [2]. It follows immediately that developing an efficient cryptographic key management scheme with key revocation is one of the challenges for smart grid.

C. Our contributions

In this paper we introduce a cryptographic key management mechanism for secure and efficient key revocation and key exchange. Our scheme is based upon a well-known cryptographic protocol: Broadcast encryption [3], [11], [12] using a media key block(MKB) [15].

In comparison with the individual identification process in the PKI-signature based IKE protocol [4], [8], our scheme is more efficient. In the IKE scheme, each device need to verify signatures in the certificate and CRL for each device before

978-1-4577-2159-5/12/$31.00 ©2011 IEEE

the each communication starts, whereas our proposed scheme only require each device to perform an MKB process to extract the media key(or, shared key) which contains one signature verification at the system's setup step.

Especially from the viewpoints of compliance by electric power utilities, the distribution of a common key material: MKB to all devices in the same management area as we proposed is more efficient than the PKI based scheme which need to distribute individual certificate for each device. In addition, MKB has been shown to be practical in Blu-ray Disc players [15], [16]. Finally, we give a detailed comparison of our scheme with the PKI-signature based IKE protocol [4], [8] mainly form the following points of view: (1) communication cost; (2) computation cost; (3) device revocation cost. The comparison results show that in most cases our scheme is efficient and cost-effective for power devices and systems in smart grid.

OVERVIEW. This paper is structured as follows: Section II introduces the system models, which include system components, system constrains and security requirements. Section III introduces the technique preliminaries used in this paper. Section IV presents the details of our proposals. The performance analysis and a comparison between our approach and an existing protocol are provided in section V. Section VI concludes the paper and refers to future work.

II. System Model

We describe the system models in this section which include system components, system constraints and security requirements.

A. System Components

In this paper, we consider a hierarchical power & information control system (Fig. 1) that includes five entities: Master Station (MS), Key Distribution Center (KDC), Phasor Data Concentrator (PDC), Phasor Measurement Unit (PMU), and Intelligent Electronic Device (IED).

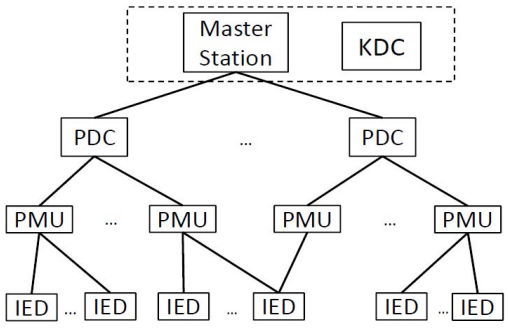

Fig. 1. System model

Master Station (MS). The master station is the brain(root) of the entire power & information control system. It controls and oversees the various substations, such as PDCs/PMUs. The MS must have reasonable computational resources and must be deployed in the independent security area.

Key Distribution Center (KDC). The key distribution center manages cryptographic keys[1] of all devices in the system. The KDC is typically deployed in the independent security areas and may be located with the MS. The KDC is responsible for the following tasks in our scheme: key generation, key distribution, key update, and key revocation.

Phasor Measurement Unit (PMU). The phasor measurement unit measures the electrical waves of the power distribution infrastructure to evaluate the system status and diagnose faults. The data measured by PMUs are collected by an intermediate element, called Phasor Data Concentrator, which in turn communicates with the supervisor system-MS. A PMU can be a dedicated device, or the PMU function can be incorporated into a protective relay or other device.

Phasor Data Concentrator (PDC). The phasor data concentrator collects, aligns, selects, and possibly decimates data from several PMUs or other PDCs. In addition, the PDC monitors the overall measurement system and provides a display and record of performance. It can provide a number of specialized outputs, such as a direct interface to a SCADA or EMS system.

Intelligent Electronic Device (IED). An IED is a device that performs electrical protection functions, advanced local control intelligence, and process monitoring. The IED can communicate directly with a PMU. An IED will be deployed in a public place, for example, on top of a pole. Unless there are very strong measures to protect the data within the IED from probing, one has to assume that any IED can be compromised. IEDs may have limited memory and processing power, e.g. a smart meter.

In this system model (Figure 1), communication occurs between MS and PDC, PDC and PMU, PMU and IED, IED and IED. A KDC can communicate with any party in the system for key management service. The MS can communicate with many PDCs, and a PDC communicates with at least one MS or another PDC. A PDC communicates with one or more PMUs by sending requests for information. PMUs communicate with IEDs in order to send control commands or collecting usage data.

B. System Constraints and Security Requirements

In this section, we recall system constraints and security requirements for the power & information control system depicted in Fig.1. Discussions are mainly based on the standards and the reports as mentioned in the introduction that are tightly related to our proposed scheme.

The devices and network of smart grid are different from those of a general PC/network environment. Some constraints of power system's environment make it difficult to apply various existing security technology to this system. In particular, we must consider the following system constraints:

[1]Devices can generate session keys by using these cryptographic keys. Session keys are not known to the KDC.

- Computational capability: Some power utility devices, e.g. PMUs or IEDs, have limited processing power processors. Therefore, it is impractical to perform intense computation such as PKI-based Encryption/Decryption.
- Communication limitation: Limited bandwidth is a key challenge in current power networks and similar bandwidth issues are expected in smart grid. Moreover, some communications links also charge costs per octet transmitted. Therefore, the authentication mechanism must not add too much overhead to the affected protocols.
- Off-line processing: Since the device management center(e.g. KDC) may not always be reachable, it should be possible for secure communication to occur between IEDs when the management center is unavailable.
- Storage capability: Memory resources of most IEDs are limited.
- Management cost: There may exist thousands of IEDs (or even more) in each management domain. From the viewpoint of compliance, it is important to consider the management cost.

In addition to our system constraints we consider the following security requirements:

- Secure key exchange: Cryptographic key is the root of authentication protocols. Therefore the key exchange between two devices is extremely important. Since the overhead of PKI-based solutions is impractical for the power devices due to the computational cost, we focus on symmetric key exchange mechanisms.
- Key revocation: When a device is compromised by attackers, the system administrator need to revoke the pre-shared key of that device as soon as possible for the security of the whole system.
- Efficient authentication: Entity authentication shall be implemented and given the aforementioned system constraints, the authentication protocol must be efficient in terms of both communication cost and computation cost.

Other security requirements such as data integrity and non-reputation are also important for power systems and other protocols, but we restrict our focus to the above security properties in this paper.

III. Technical Preliminaries

Broadcast encryption [3], [11], [12] is widely used in copyright protection system such as Advanced Access Content System(AACS) [15], [16]. Broadcast encryption is one of the cryptographic primitives that enables the key distribution center (KDC) to securely and efficiently distribute the group key, called media key K_M hereafter, to the compliant devices. The copyrighted content, such as TV programs and movies, is encrypted with K_M and its ciphertext is distributed via a broadcasting system or media such as Blu-ray Disc. Since only a compliant device can decrypt the ciphertext with K_M and play the content, its copyright should be well protected. Broadcast encryption has a special property that revokes maliciously cracked devices at any time without updating the initial parameters for each device.

Let us review the basic idea of broadcast encryption[2] from [12]. In broadcast encryption, devices are placed as leaves of a binary tree T with depth d. For simplicity, let us consider the case of $d = 3$. We denote the nodes of tree by n_i where $i \in [0, 14]$ as shown in Figure 2. n_0 is the root node and eight (= 2^3) devices $D_1, \cdots, D_8$ are assigned to $n_7, \cdots, n_{14}$, respectively.

At the initialization, KDC randomly chooses ℓ-bit secret keys $k_0, \cdots, k_{14}$ and assigns them to each node, respectively. KDC then securely writes the device key KD_i into device D_i, where KD_i is the set of secret keys corresponding to the nodes from the leaf to root. For example, $KD_1 = \{k_0, k_1, k_3, k_7\}$ and $KD_6 = \{k_0, k_2, k_5, k_{12}\}$.

In broadcasting, KDC broadcasts a ciphertext from which only the compliant devices can recover K_M. KDC first divides T into subtrees that cover all the compliant devices but none of the revoked one. KDC then computes the ciphertext(s) by encrypting K_M with the secret key of root node in each subtree. Let us consider two scenarios: (i) no device is revoked, and (ii) three devices are revoked.

(i) When all devices are compliant, the subtree is T itself. KDC randomly generates the media key K_M and encrypts it with k_0 (corresponding to root node in T) into the ciphertext $\mathsf{Enc}(k_0, K_M)$ where $\mathsf{Enc}(key, msg)$ represents the ciphertext of message msg with key key. KDF finally broadcasts $\mathsf{Enc}(k_0, K_M)$ (and possibly $\mathsf{Enc}(K_M, contents)$) to all devices. Since all D_i have the secret key k_0, it can decrypt K_M from $\mathsf{Enc}(k_0, K_M)$ with k_0 (and also $contents$ from $\mathsf{Enc}(K_M, contents)$ with K_M).

(ii) Assume that D_1, D_5 and D_6 are revoked(Figure 3). KDC divides T into three subtrees S_1, S_2 and S_3 whose root nodes are n_7, n_4 and n_6, respectively. KDC then computes a tuple of ciphertexts $\{\mathsf{Enc}(k_7, K_M), \mathsf{Enc}(k_4, K_M), \mathsf{Enc}(k_6, K_M)\}$ and distributes the tuple to all devices. The compliant devices have one of k_7, k_4 and k_6 but the revoked ones have none of them, and therefore, only the compliant devices can recover K_M from the tuple.

Using the broadcast encryption, KDC can distribute K_M efficiently. Without it, KDC should compute $n-m$ ciphertexts for each compliant device individually, where n and m are numbers of total and revoked devices, respectively. On the other hand, with Naor et al. 's approach [12], the number of ciphertexts that KDC computes may be one as in (i) and is at most $m \log_2(n/m)$.

IV. Our Proposal: Secure Authenticated Key Exchange

In this section, we propose an authenticated key exchange scheme with revocation by implementing and exploiting a well-known cryptographic protocol: broadcast encryption [3], [11], [12] using an MKB [15]. Our scheme contains two sub-protocols: The first sub-protocol contributes to key-sharing

[2]Naor et al. [12] proposed two kinds of broadcast encryption in the subset-cover framework; *the complete subtree method* and *the subset difference method*. This section deals with a simple case of the former method.

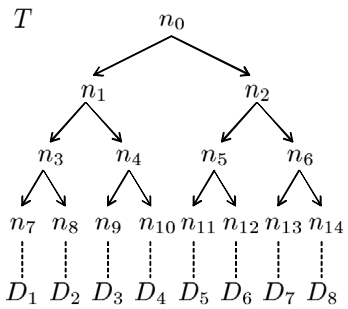

Fig. 2. Broadcast Encryption

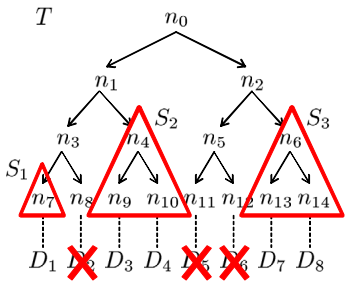

Fig. 3. Devices Revocation

between un-revoked devices. It allows a pair of devices to share a key(called media key) for authentication, where a revoked device is unable to share the key. The key-sharing process contains broadcast encryption where the shared media key can be extracted by each device with its device key. The second sub-protocol is devoted to authentication. It uses random numbers generated during the first sub-protocol and thus should be preceded by the first sub-protocol. In the second sub-protocol, a device is able to identify another device when their communication begins. The first sub-protocol of key-sharing is mandatory while the second sub-protocol of authentication is optional though this is strongly recommended whenever identification of a device is necessary for some systems with higher security requirement.

A. Notations and Definitions

The following notations are used throughout the remainder of this paper.

- P: a base point on the elliptic curve. P is a public value for the whole system. We consider the 256 bits elliptic curve whose base point is 512 bits;
- $[r]P$: the open ID of a device. It is called the elliptic scalar multiplication of the elliptic curve point P whose result is also 512 bits. r $(1 \leqq r < ord(P))$ is a positive integer[3]. $ord(P)$ means the order of P;
- K_M: the media key (group key), we recommend the size of a key is 128bits;
- $H()$: a one-way hash function;
- $h(key, message)$: an AES based one-way hash function. $message$ is hashed by the secret key key whose size is also recommended as 128 bits;
- SID_i: the secret ID for device(node) i;
- l_i: the ith device's leaf number in the binary tree of keys. The maximum size of a leaf number can be 64 bits;
- k_i: the secret key of each node in the binary tree T. We recommend the size of a key is 128bit;
- KD_i: the device key for a device D_i, whose corresponding secret ID is SID_i. KD_i is a key set which is composed of the set of secret keys corresponding to the nodes from the T's leaf to root;
- *I, R*: the system public value, which indicate the initiator and responder during the communication. We assume $I \neq R$.
- $Enc(key, message)$: an encryption algorithm of a symmetric cipher(such as AES [13]) using a secret key key;
- $Dec(key, message)$: a decryption algorithm using the secret key key corresponding with $Enc(key, message)$;
- $MAC(key, message)$: a tag generation algorithm of a message authentication code (such as OMAC [14]) using a secret key key.

[3]From the Elliptic Curve Discrete Logarithm Problem(ECDLP) [17], it is infeasible to recover r with P and $[r]P$

B. Setup

The KDC generates a key pair (K_p, K_s), where K_p and K_s are its public key and the private key respectively. Each device (node) shall get a secret ID, SID_i, from the KDC most probably at the start of either the factory acceptance test or the site acceptance test. The secret ID is confidential so that a device must keep it in a secure manner. The result of $[SID_i]P$ acts as the open ID of the device with secret ID SID_i. As described in Section III, each device shall be assigned a unique leaf number l_i that corresponds to the binary tree of keys T as described. Also at the initialization, KDC chooses a set of secret keys k_i and assigns it to each node respectively. KDC then securely writes the device key KD_i into each device D_i, where KD_i is the set of secret keys corresponding to the nodes from the leaf to root in the binary tree T. E.g. in Figure 1, $KD_1 = \{k_0, k_1, k_3, k_7\}$.

In this paper, we still call the ciphertext containing the media key(or, group key) K_M broadcasted from the KDC, the MKB, like in [15]. Any compliant device can extract the media key K_M by using its corresponding device key (Figure 4). If a set of device keys is compromised in a way that threatens the integrity of the system, an updated MKB can be released that causes a device with the compromised set of device keys to be unable to calculate the correct K_M. In this way, the compromised device keys are "revoked" by the new MKB. Please refer to Section III for details of the revocation process. For security considerations, we make the following modification to the original MKB [11], [15]:

- Adding version information to the MKB in our scheme that is managed by the KDC when it issues the MKB to each device. This information can ensure devices the usage of the latest MKB from KDC.
- Adding expiration information to the MKB in our scheme following the version field. This information tells a device not to use a K_M derived from an expired MKB.
- Adding a signature to the MKB for integrity verification. The calculation of the MKB signature includes the two new fields described above.

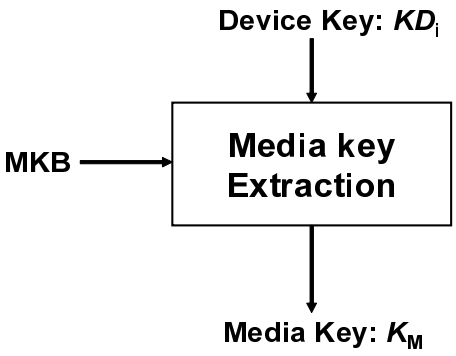

Fig. 4. Media Key Extraction Process

After receiving the MKB, the media key K_M extraction process executed on the device side can be summarized as follows (the process terminates if a check fails):

1) Verify MKB Signature.
2) Check the expiration information of the MKB data.
3) Check the version information of the MKB data.
4) Perform a media key extraction process described above.
5) Check if the device is revoked.
6) Verify the signature of the MKB data.
7) Store the media key K_M, the expiration time and the version of MKB.

C. Secure Authenticated Key Exchange: Key Sharing Scheme

In this subsection, we introduce the key sharing scheme between two devices after both have received MKB from the KDC respectively.

There are two phases in the key sharing scheme(Figure 5). The first is the key sharing phase, and the second is called the conformation phase. Let's assume there are two devices, device 1 and device 2, that want to share a key for future secure data transmission. A description of the first phase is as follows:

1) **Device 1→Device 2**: Device 1 first obtains K_M from the media key extraction process by its device key. Then it generates a random number, r_1. Using the public value P, device 1 calculates the $[r_1]P$ and sends it to device 2.
2) **Device 2→Device 1**: Having obtained $[r_1]P$ from device 1, device 2 also obtains the media key K_M from the key extraction process by its device key. Then device 2 also generates a random number r_2 and calculates $[r_2]P$ and then sends $[r_2]P$ back to device 1.
3) **Device 1**: Device 1 then calculates a common key $K_{C1} = h(K_M, [r_1]([r_2]P))$ as the shared key for secure communication with device 2.
4) **Device 2**: Device 2 calculates $K_{C2} = h(K_M, [r_2]([r_1]P))$ as its shared key with device 1.

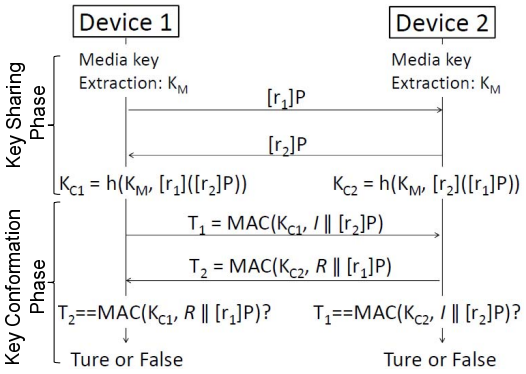

Fig. 5. Secure Authenticated Key Exchange: Key Sharing Scheme

The shared key between two devices may be verified in the conformation phase. This phase is optional and can be performed when additional security is required.

1) **Device 1→Device 2**: Device 1 calculates a token T_1 for key verification: $T_1 = MAC(K_{C1}, I \parallel [r_2]P)$. Device 1 sends the token to device 2.
2) **Device 2→Device 1**: Device 2 also calculates and sends a token $T_2 = MAC(K_{C2}, R \parallel [r_1]P)$ to device 1.
3) **Device 1**: Device 1 checks whether $MAC(K_{C1}, R \parallel ([r_1]P))$ equals to T_2. If true, device 1 can confirm that K_{C1} is shared with device 2.
4) **Device 2**: In the same way, device 2 can also confirm K_{C2} is shared with device 1.

Note: Prepending a marker (also called public value, I and R in the Fig.5) to the plaintexts of T_1 and T_2 generated by the initiator and the responder can prevent replay attacks from adversaries that exist in the communication channel.

D. Secure Authenticated Key Exchange: Authentication Scheme

If two devices want to confirm the identity of the device with open ID as he announced, they can perform the mutual authentication scheme after the key sharing phase. During the *setup* step, using K_s, SID_i, l_i and P, the KDC assigns a signature Sig_i to each device: $Sig_i = \text{sign}(K_s, ([SID_i]P)||l_i)$, where $([SID_i]P)||l_i$ is the concatenation of $[SID_i]P$ and l_i. Sig_i is the ECDSA signature [17], [7] by the private key K_s. The device keeps the signature for future's identification. Figure 6 describes the whole process. A description of this scheme is as follows:

1) **Device 1→Device 2**: Using the r_1 and $[r_1]P$ generated in the key sharing scheme, device 1 calculates

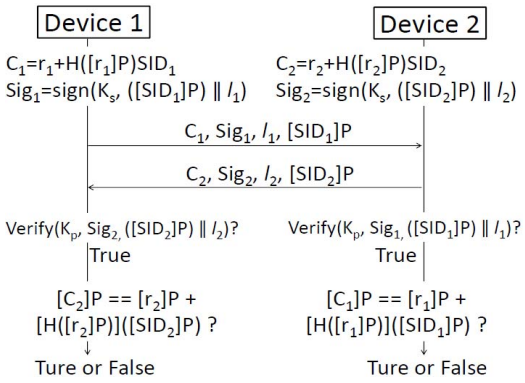

Fig. 6. Secure Authenticated Key Exchange: Authentication Protocol

$C_1 = r_1 + h(K_M, [r_1]P)SID_1$, and sends it together with l_1, $[SID_1]P$, Sig_1 = sign$(K_s, ([SID_1]P)||l_1)$ to device 2. (Note: the signature Sig_1 is issued by the KDC as described in the $setup$ step)

2) **Device 2→Device 1**: Device 2 also calculates and sends $C_2 = r_2 + h(K_M, [r_2]P)SID_2$, l_2, $[SID_2]P$, Sig_2 = sign$(K_s, ([SID_2]P)||l_2)$ to device 1. (The signature Sig_2 is issued by the KDC as described in the $setup$ step)
3) **Device 1**: With the public key K_p, device1 verifies the signature Sig_2. If the verification is passed, using the $[r_2]P$ received from device 2 in the key sharing phase, device 1 can calculate $[r_2]P + [h(K_M, [r_2]P)](SID_2P)$ and compare it with $[C_2]P$. If equal, device 1 can confirm that (SID_2P) is surely the open ID of device 2.
4) **Device 2**: By the same process, device 2 firstly verifies the signature Sig_1 with K_p then cryptographically identifies device 1 as the owner of open ID (SID_1P) by checking whether $[C_1]P$ is equal to $[r_1]P + [h(K_M, [r_1]P)](SID_1P)$.

V. Comparison and Discussion

In this section, we compare our proposed scheme with a typical PKI-based key exchange protocol, namely, a variant of the IKE protocol [4], [8] which is based on the elliptic curve digital signature algorithm (ECDSA) [7] scheme. We first give a simple review of the IKE scheme using ECDSA, then we discuss the result of the comparison between the IKE protocol using ECDSA and ours.

A. IKE Protocol using ECDSA

The IKE protocol is defined by RFC 2409 [4]. RFC 4754 [8] proposes a variant where the elliptic curve digital signature algorithm(ECDSA) is used as the authentication method within the original IKE. The ECDSA based IKE protocol is briefly summarized in Figure 7. In addition to the notation proposed in section IV, we introduce the following new notation:

- N_z: the nonce payload; z can be i or r for the initiator or responder respectively;
- $([x]P, x)$: The public/private-key pair for the ECDSA signature, where $[x]P$ is the public key for verification, x is the private key for signing.
- $Cert_z$: The certificate for the public key. z can be i or r for the initiator or responder respectively. This certificate is issued by a certification authority(CA);

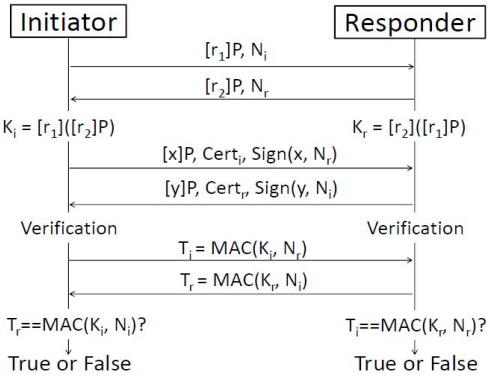

Fig. 7. IKE Protocol using ECDSA

The initiator and the responder first perform the elliptic scalar multiplication based on a public elliptic curve point P, then exchange $\{[r_1]P, N_i\}$ and $\{[r_2]P, N_r\}$. Both of them can calculate the $K_i = [r_1]([r_2]P)$ and $K_r = [r_2]([r_1]P)$ respectively. Using the private key x, the initiator calculate the signature $Sign(x, N_r)$, then he send $\{[x]P, Cert_i, Sign(x, N_r)\}$ to the responder. By the same calculation, the responder calculate and send $\{[y]P, Cert_r, Sign(y, N_i)\}$ the initiator. After receiving the public key, certificate and signature, each of them can first verify the public key certificate by checking the certificate revocation list (CRL) received from the CA(Note that the CRL carries a digital signature associated with the CA by which they are published), if the certificate is not revoked, he can verify the signature. After the signature verification, the initiator sends a token $T_i = MAC(K_i, N_r)$ to the responder, and also receives a token $T_r = MAC(K_r, N_i)$ from the responder. Finally, both of them can verify the exchanged keys K_i and K_r by the verification results of *MAC* computation.

B. Comparison and Discussion

In this subsection, Protocol 1 indicates our proposed mandatory key sharing scheme, Protocol 2 indicates our proposed optional key authentication scheme. We estimate computational cost and communication cost in the case of 128-bit security, i.e. IKE protocol using ECDSA and our scheme are constructed over a 256-bit elliptic curve group. Then the size of a point on the elliptic curve is 512 bits, and the size of an ECDSA signature is 512 bits. Let a tag size of a MAC be 256 bits respectively. Let a bits length of the leaf number be 64 bits.

TABLE I shows communication cost of Protocol 1, Protocol 1 + 2, and IKE protocol, respectively. The initiator and the responder of Protocol 1 respectively send one elliptic curve point and one encrypted elliptic curve point, in total, a 1024

TABLE I
COMMUNICATION COST

	Our Protocol 1	Our Protocol 1 + 2	IKE protocol
Initiator	768 bits	2112 bits	2432 bits
Responder	768 bits	2112 bits	2432 bits

bits string. In the case of executing both Protocol 1 and Protocol 2, the initiator and the responder respectively send two elliptic curve points ($|[r_i]P|$, $|[SID_i]P|$ = 512bits), a 128 bits tag (T_i), a 256 bits string (C_i), the leaf number (64 bits in max) and one ECDSA signature (512 bits), in total, 1984 bits string. On the other hand, the initiator and the responder of IKE respectively send two elliptic curve points and one tag, and two ECDSA signatures, in total, 2560 bits string. Therefore, the communication cost of our protocol is more efficient than the ECDSA based IKE protocol.

TABLE II
COMPUTATION COST

	Our Protocol 1	Our Protocol 1 + 2	IKE protocol
Initiator	$2M$	$6M$	$7M$
Responder	$2M$	$6M$	$7M$

Next, We compare the computation cost by estimating the numbers of the scalar multiplication which is a dominant computation. Let M be a computation cost of a scalar multiplication. The signing of ECDSA includes one scalar multiplication, and the verification of ECDSA includes two scalar multiplications. TABLE II shows the numbers of scalar multiplications. The initiator of Protocol 1 computes two times of scalar multiplication, i.e., $[r_1]P$ and $[r_1]([r_2]P)$. The initiator of Protocol 1 + 2 computes six times of scalar multiplication, i.e., $[r_1]P$, $[r_1]([r_2]P)$, $[C_2]P$, $[H([r_2]P)]([SID_2]P)$, and one verification of ECDSA. The initiator of IKE protocol computes seven times of scalar multiplication, i.e., $[r_1]P$, $[r_1]([r_2]P)$, one signing, and two ECDSA verification. The computation cost of the responder is also estimated by the same way. Therefore, our protocol achieves more efficient communication cost than the ECDSA based IKE protocol.

TABLE III
COMPUTATION COST FOR REVOCATION

	Our Protocol 1 / Protocol 1 + 2	IKE
KDC	$(m \log_2 n/m)Cost_{enc} + M$	M
Device	$Cost_{dec} + 2M$	$2M$

Finally, we compare the key revocation cost. Let $Cost_{enc}$ and $Cost_{dec}$ be computation cost of an encryption and a decryption of the symmetric key encryption, respectively. Let n be the number of all devices in the system, and let m be the number of revoked devices. TABLE III shows the computation cost for the key revocation in the worst case. In our protocol, the KDC computes $(m \log_2 n/m)$ times encryption of the symmetric key encryption and one signing of ECDSA signature, and each devise compute one decryption of the symmetric key encryption and one verification of ECDSA signature. On the other hand, in IKE protocol the CA generates CRL which contains the generation of one ECDSA signature, and each device verifies a CRL which contains the verification of one ECDSA signature. From the TABLE III, we can see that in our protocol, the computation at the KDC is a little heavier than the computation of CA in ECDSA based IKE. i.e. Some computation of symmetric key encryption and decryption occurs in our protocols. We think this will not become a problem for electrical systems in the smart grid because almost the same revocation has been practically adopted by the Blu-ray Disc players for a long time. This is just being an appropriate availability proof of the revocation mechanism.

VI. CONCLUSION

Considering the system constraints and security requirements in the smart grid, we proposed an authenticated key exchange scheme with revocation by implementing and exploiting a well-known cryptographic protocol: Broadcast encryption [3], [11], [12] using an MKB [15].

Our secure and efficient authenticated key exchange scheme only need each device to perform an MKB process to extract the media key(or, shared key) which contains one signature verification after receiving the MKB data at the system's setup step. Then, our proposed authenticated key sharing scheme can be efficiently performed among any valid devices by executing only a symmetric key(e.g. AES [13]) based process with the media key. Especially from the viewpoints of compliance by electric power utilities, the distribution of a common key material: MKB to all devices in the same management area as we proposed is more efficient than the PKI based scheme which need to distribute individual certificate for each device. Since the MKB based broadcast encryption key management method adopted in our scheme is practically and securely used in millions Blu-ray Disc players [15], [16], our protocol can also achieve a reasonable security in the smart grid. Moreover, the comparison results between our scheme and an ECDSA signature based IKE protocol [4], [8] show that our scheme is more efficient and cost-effective in most cases for devices and systems in smart grid.

As discussed by several existing works [9], [10], it may be very difficult to propose a perfect key management security solution only using mechanisms of a symmetric key or public key for all systems or devices in the smart grid. We hope our work inspires researchers and engineers to design security protocols for the smart grid which strike a good balance between high-efficiency and cost-effectiveness.

In our complimentary work, we have successfully analyzed authentication and secrecy properties of the Key Sharing Scheme using Abadi and Fournet's applied pi calculus [18], we hope to publish these results shortly. A natural direction for future work could extend this study to verify privacy and perfect forward secrecy properties of the Key Sharing Scheme and, moreover, consider security properties of the Authentication Protocol.

ACKNOWLEDGMENTS

Authors would like to thank Dr.Hirofumi Muratani, from TOSHIBA Corporation, for his valuable and suggestive discussion in the security verification of our protocols.

REFERENCES

[1] NIST. NISTIR 7628, Guidelines for Smart Grid Cyber Security, available at: http://csrc.nist.gov/publications/PubsNISTIRs.html#NIST-IR-7628

[2] IEC 62351. Power system management and associated information exchange - Data and communications security.

[3] RFC 2627: Key Management for Multicast: Issues and Architectures. http://www.ietf.org/rfc/rfc2627.txt

[4] RFC 2409: The Internet Key Exchange (IKE). http://www.ietf.org/rfc/rfc2409.txt

[5] RFC 3280: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. http://www.ietf.org/rfc/rfc3280.txt

[6] RFC 5280: Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. http://www.ietf.org/rfc/rfc5280.txt

[7] ANSI X 9.62, "American national standard for financial services – public key cryptography for the financial services industry: The elliptic curve digital signature algorithm (ECDSA)," American National Standard Institute, 1998.

[8] RFC 4754: IKE and IKEv2 Authentication Using the Elliptic Curve Digital Signature Algorithm (ECDSA) http://www.ietf.org/rfc/rfc4754

[9] S.Fuloria, R.Anderson, F.Alvarez and K.McGrath. Key Management for Substations: Symmetric Keys, Public Keys or No Keys? Proceedings of the IEEE Power Systems Conference & Exposition(PSCE 2011), Phoenix, Arizona, USA, 2011.

[10] Ludovic Pietre-Cambacedes and Pascal Sitbon. Cryptographic Key Management for SCADA Systems-Issues and Perspectives Proceedings of the 2008 International Conference on Information Security and Assurance, IEEE Computer Society, 2008.

[11] A. Fiat and M. Naor. Broadcast encryption. Proceedings of Crypto 1993, volume 773 of LNCS, pages 480-49191, Springer-Verlag, 1993.

[12] D. Naor, M. Naor, and J. Lotspiech. Revocation and tracing schemes for stateless receivers. Proceedings of Crypto 2001, volume 2139 of LNCS, pages 41-62, Springer-Verlag, 2001.

[13] FIPS PUB 197, advanced Encryption Standard, Federal Information Processing Standards Publication 197, Department of Commerce, National Institute of Standards and Technology (NIST), Information Technology Laboratory (ITL), 2001.

[14] OMAC: One-Key CBC MAC – Addendum. http://csrc.nist.gov/groups/ST/toolkit/BCM/documents/proposedmodes/omac/omac-ad.pdf

[15] AACS Specifications: Introduction and Common Cryptographic Elements Book Rev 0.951, 2010. http://www.aacsla.com/specifications/AACS_Spec_Common_FINAL_0951.pdf

[16] AACS Specifications: Blu-ray Disc Pre-recorded Book, AACS LA. 2006-07-27. p. 15. Retrieved 2007-11-01. http://www.aacsla.com/specifications/AACS_Spec_BD_Prerecorded_0.912.pdf

[17] Lan F.Blake, Gadiel Seroussi and Nigel P.Smart. Advances in Elliptic Curve Cryptography. CAMBRIDGE UNIVERSITY PRESS, 2005.

[18] Martín Abadi and Cédric Fournet. Mobile values, new names, and secure communication. In POPL'01: 28th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, pages 104-115. ACM Press, 2001.