Scalable Multi-group Key Management for Advanced Metering Infrastructure

Mourad Benmalek, Yacine Challal, Abdelmadjid Bouabdallah

Scalable multi-group key management for Advanced Metering Infrastructure

Mourad Benmalek*, Yacine Challal*§ and Abdelmadjid Bouabdallah‡
*Ecole nationale Supérieure d'Informatique ESI, Laboratoire de Méthodes de Conception de Systèmes, Algiers, Algeria
§Centre de Recherche sur l'Information Scientifique et Technique CERIST, Algiers, Algeria
Email: {m_benmalek, y_challal}@esi.dz
‡Université de Technologie de Compiègne, Laboratoire Heudiasyc, UMR CNRS 7253, Compiègne, France
Email: madjid.bouabdallah@hds.utc.fr

***Abstract*—Advanced Metering Infrastructure (AMI) is composed of systems and networks to incorporate changes for modernizing the electricity grid, reduce peak loads, and meet energy efficiency targets. AMI is a privileged target for security attacks with potentially great damage against infrastructures and privacy. For this reason, Key Management has been identified as one of the most challenging topics in AMI development. In this paper, we propose a new Scalable multi-group key management for AMI (SAMI) to secure data communications in an Advanced Metering Infrastructure. It is a key management scheme that can support unicast, multicast and broadcast communications based on an efficient multi-group key graph technique. An analysis of security and performance, and a comparison of our scheme with recently proposed schemes show that our solution induces low storage overhead (reduction reaches 83%) and low communication overhead (reduction reaches 99%) compared to existing solutions.**

***Keywords*—Advanced Metering Infrastructure (AMI); Smart Grid (SG); Security; Key Management.**

I. Introduction

SMART GRID (SG) refers to the next generation power grid in which the electricity distribution and management is upgraded by incorporating advanced two-way communications and pervasive computing capabilities for improved control, efficiency, reliability and safety [1]. A practical example of the benefits of introducing the smart grid includes the greater availability of electricity to homes at a lower cost, and the integration of distributed and renewable power generation such as local solar and wind generators [2].

Advanced Metering Infrastructure (AMI) is an essential part of the smart grid. It inherits the two-way communication capability of smart grid and introduces new opportunities for consumers and suppliers: it is responsible for collecting all the data and information from loads and consumers, and it is also responsible for implementing control signals and commands to perform necessary control actions [3]. The critical role of AMI in the smart grid has made this system a privileged target of cyber attacks. Consequently, AMI security is of very high importance for the security of the smart grid.

In general, the fundamental security requirements of AMI are: confidentiality, integrity, and availability [4]. Privacy of the customer's sensitive data like metering and energy consumption is the most important issue of confidentiality in AMI, customers do not want unauthorized people or marketing firms to know how much energy they are using, what their pattern of energy usage is, or other energy-related information. Integrity in AMI is very important for both meter reading stored in smart meters or transmitted over the communication channels and control commands such as Demand Response (DR) mechanisms that enable customers to cut down energy usage at peak times for example. Unlike traditional systems, availability of information and control commands generated and managed by AMI is compulsory for the operation of the whole smart grid which includes too many meter readings being exchanged between smart meters and utility system.

To meet these security requirements, cryptographic countermeasures must be deployed to protect data integrity and confidentiality for AMI. However, cryptographic mechanisms for AMI require also an efficient key management. Inadequate key management can result in possible key disclosure to attackers, and even jeopardizing the entire goal of secure communications in AMI. Therefore, key management is a critical process to ensure the secure operation of AMI.

Several key management schemes (KMS) have been proposed [5-12], but none of them can completely satisfy the security requirements mentioned previously. Hence, we propose a new key management scheme for AMI based on an efficient and scalable multi-group key graph technique to secure unicast, multicast, and broadcast communications in a smart grid network while achieving the security requirements of AMI.

The remainder of this paper is organized as follows. We discuss related works in Section II. In Section III we study the architecture and characteristics of messages of a AMI and the key management function requirements. In Section IV we present our key management scheme. We give a security and performance analysis of our KMS in Section V. Finally, we draw our conclusions in Section VI.

II. Related Works

In recent years, several schemes have been proposed to secure communications for AMI in smart grid. According to [13], key management has been identified as a fundamental security challenge in an AMI.

Kamto *et al.* [5] proposed a key distribution and management scheme for large customer networks to achieve authentication, privacy and data confidentiality in AMI. The proposed scheme is computationally expensive because of relying on Diffie-Hellman (DH) [14] key exchange and a group ID-based mechanism [15]. Furthermore, this scheme only secures communications between HAN (Home Area Network) devices and the gateway.

Yan *et al.* [6] proposed an integrated approach in which trust services, integrity and data privacy could be provided by mutual authentications. In [7], Li and Cao proposed a one-time signature scheme to address the problem of preventing message forgery attacks in multicast communications. The proposed scheme presents a significant reduction in the storage and communication overhead, but only focuses on communication integrity and do not address confidentiality.

Nicanfar *et al.* [8] developed a key management protocol for data communication between the utility server and customers' smart meters based on the concept of ID-Based public/private key pair model [15]. Although the proposed key management protocol aims to reduce the computation overheads, the synchronization process still demands considerable computation efforts. Wu and Zhou [9] combines symmetric key technique based on the Needham-Schroeder authentication protocol [16] and elliptic curve public key technique [17] to provide a novel key management scheme for smart grid assuring strong security, fault-tolerance, efficiency and scalability. In the work of Xia and Wang [11], the authors showed that Wu and Zhous scheme is vulnerable to the man-in-the-middle attack and proposed an improvement for this scheme based on a trusted third party. However, these two schemes do not support secure multicast communications that play an important role and have wide applications in SG.

Recently, a key management scheme is proposed by Liu *et al.* [11] to secure unicast, multicast, and broadcast communications in AMI. This scheme based on the key graph management approach [18] suffers from a lack of scalability due to inefficient key management that results in non-negligible communication overhead for such a large-scale system. Moreover, we found that Liu's *et al.* scheme is not tolerant to packet loss. Wan *et al.* [12] proposed an improvement for Liu's *et al.* scheme that combines an adapted identity-based cryptosystem [19] and one-way function tree (OFT) approach [20] for multicast key management. The use of an OFT separately for each DR project (DR projects are programs designed to decrease electricity consumption or shift it from on-peak to off-peak periods depending on consumers' preferences) results in non-negligible overhead for key storage.

None of the proposed KMS can completely satisfy the aforementioned requirements (confidentiality, integrity, and availability). Some schemes are not designed to secure multicast or broadcast communications that are frequently used in AMI, and some others result in non-negligible storage or communication overhead. Hence, we propose a new efficient and scalable key management scheme that meet the security requirements of AMI.

III. AMI Architecture and Security Requirements

In this section we analyze AMI system structure and interactive messages exchanged via AMI communication networks, to identify the basic requirements that are relevant to key management.

A. AMI Architecture

An AMI is composed of (Fig. 1):

1) Smart Meters (SMs): Which are electrical meters providing two-way communications, automated meter data collection and outage management. They also allow dynamic pricing, and joining/leaving Demand Response pricing projects for load control.

2) Distributed Energy Resources (DERs): Are small scale renewable electricity generation systems for family use and energy storage.

3) Gateways (GWs): Implement protocol conversion and communications between two heterogeneous networks, like the in-home network and wide area network.

4) Wide Area Communication Infrastructure: It supports bidirectional communication between costumers domain and the utility system. Different architectures and medias can be used like power line communication system, cellular networks, or IP-based networks [21].

5) Meter Data Management Systems (MDMS): Acts as a database system for storing, managing, and further analyzing metering data in order to propose dynamic pricing, better customer service, DR and energy consumption management purposes.

B. AMI Interactive message characteristics

Interactive messages in AMI include metering data, release of DR projects, joining/leaving a DR project, electricity pricing, remote load control and notifications (loss or restoration of power).

Messages can be categorized into three classes according to their transmission mode: unicast, broadcast, and multicast.

- *Unicast communication:* is used when messages are transmitted from one point to the other one, for instance when a SM reports its power consumption statistics and estimated future energy demand to the utility system.
- *Broadcast communication:* is used when messages are transmitted from one point to all the other points. A typical example of a broadcast notification message includes the real-time electricity pricing information sent from utility system to all SMs.
- *Multicast communication:* is used when messages are transmitted from one point to a subset of the other points, e.g., a remote load control message from utility system to SMs which subscribed to the same DR project.

C. KMS Function Requirements

As KMS is a critical subsystem of the whole AMI security architecture, and given the above characteristics of AMI interactive messages, we summarize in what follows the basic requirements for an effective KMS for AMI security:

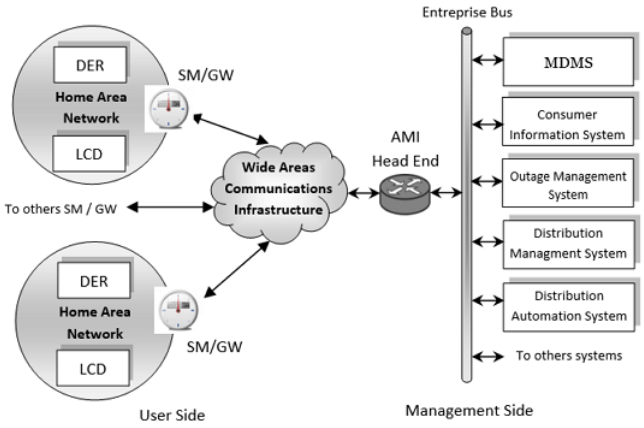


Fig. 1: System structure of AMI

1) Hybride Transmission Modes: The key management framework should support the three tansmission modes in AMI: unicast, multicast and broadcast. For each mode, methods of key generating, refreshing, and distribution policies must be designed clearly.

2) Scalability: It represents a major issue for such a large-scale system consisting of millions of SMs.

3) Efficiency: We consider three aspects: computation, storage, and communication because of their impact on the overal system performance. The KMS processes should be computationally efficient as well as memory-usage efficient meeting the scarcity of computation and storage capacities in SMs. The processes of key generation, distribution, usage, and refreshment should also induce low communication overhead, which is important to time-critical scenarios in AMI.

4) Backward and forward secrecy: Users participating in DR projects are not fixed. Any user can join or leave any DR project at any time. For this reason, it is obvious that the forward and the backward sercery [18] should be guaranteed. The forward sercery implies that previously used secret keys and messages must be inaccessible by the new users who participate in a DR project, and the backward secrecy means that the future secret keys and messages must be inaccessible by users who leave a DR project.

5) Collusion freedom: Any set of users that unsubscribe a DR project should not be able to deduce the current used group key.

IV. Scalable multi-group key management for Advanced Metering Infrastructure: SAMI

We introduce a new scalable and efficient key management scheme that we call Scalable multi-group key management for Advanced Metering Infrastructure (SAMI). It is based on a multi-group key graph structure that supports the management of multiple Demand Response projects simultaneously for each customer. We will demonstrate later that this new structure scales to large smart grids with dynamic Demand Response projects membership while meeting smart meters' constraints in terms of memory and bandwidth capacities.

A. Assumptions

1) The Advanced Metering Infrastructure complies with the architecture illustrated in Fig. 1. The MDMS denotes the management side and it is responsible for key generation and rekeying, and it is well protected from attacks.

2) A specific default DR project is mandatory for all users of the SG, i.e. all users are subscribed to this default DR project. This default DR project will be used by MDMS to broadcast control messages or information to all customers of the SG.

3) Except the mandatory DR project, any user can join or leave any DR project at any time.

B. Initialization of the KMS

Let us consider a set of n smart meters. Initially, a specific method of securely exchanging cryptographic keys over a public channel is used to establish individual keys between the MDMS and smart meters (For example, we can use the Elliptic Curve Diffie-Hellman ECDH key agreement [22] that is known to induce less overhead compared to many exiting end-to-end key establishment using standard Diffie-Hellman protocol). Theses individual keys $\{k_1, ..., k_n\}$ will be refreshed periodically and will be used in two ways. The first one is to secure unicast communications between MDMS and the SMs, and in the other one they are used for generating the multi-group key graph for secure multicast communications.

Moreover, The MDMS must generate a group key GK_0 (refreshed periodically) for the default DR project. This key will be generated and transmitted through secure channels for each SM, and will be used to secure messages transmitted in broadcast mode. In Table I, we summarize the terminology that we will use throughout the remaining of this paper.

TABLE I: Notation Table

Notation	Description
$H(.)$	A One-way hash function
n	Number of SMs
m_i	Number of the i^{th} DR project members
h_i	Height of i^{th} OFT $h_i = log_2(m_i)$
N_{pr}	Number of DR projects
$Nsub(u_i)$	Number of DR projects to which subscribes user u_i
$Home\_DR(u_i)$	First DR project to which subscribes user u_i
$set(u_i)$	Set of DR projects to which subscribes user u_i
DR_i	The ith DR project
GK_i	Group key of DR_i
$Path(u_i)$	All keys corresponding to the nodes in the path from u_i's individual key to $Home\_DR(u_i)$ Group key
$right(k_i)$	Right children of node k_i in the tree
$left(k_i)$	Left children of node k_i in the tree
$a\|\|b$	A concatenation between a and b
$Enc(M, k)$	Message M encrypted with key k
$HMAC_k(c)$	Keyed-hash using k as the key
$\oplus$	Mixing function such as bitwise exclusive-or (XOR)
$A \rightarrow B : M$	A sends a message M to B

C. Group Key Management

In our solution, we propose a secure, efficient and scalable management of group keys in the AMI system. To address the scalability issue, key graph techniques are mostly used in the literature. Specifically, we adopt OFT [20] (One-Way Function Trees) which is an improvement of LKH protocol (Logical Key Hierarchy) proposed in [18] that allows to reduce the number of rekey messages.

In OFT, the MDMS and all users individually compute the group key GK. The keys of interior nodes are recursively computed from the keys of their children rather than attributed by the MDMS using the formula:

$$k_i = f(k_{right(k_i)}) \oplus f(k_{left(k_i)}) \quad (1)$$

The result of applying the one-way function f to a key k: $f(k)$, is called the *blinded key* version of k. A user only knows his individual key and the *blinded keys* of the *sibling nodes* of the nodes on his path to the root node, and these keys allow him to compute its *ancestors*. An example of OFT is illustrated in Fig. 2.

However, as the users can subscribe to multiple DR projects at the same time, an intuitive solution is to use a key tree for each DR project. Hence, if a user u_i subscribes to two or more DR projects simultaneously (e.g. DR_j and DR_k), he needs to manage two sets of keys. As a result, this naive application of OFT may be costly and induces a non-negligible key storage overhead.

To reduce storage and communication costs in key management, we propose a novel multi-group key graph structure. The idea of our new key graph technique is to allow multiple DR projects to share a new set of keys.

1) Multi-group Key Graph Structure: Our multi-group key graph structure can be modeled as shown in Fig. 3: in the *lower level*, each OFT represents a set of users with the same first DR project subscription, the leaf node of the tree is a user's individual key and tree's root is the DR project's group key. The graph in the *upper level* represents combinations of root keys for users subscribing to multiple DR projects at the same time. Our multi-group key graph has the following properties: (a) a user only belongs to one OFT in the multi-group key graph corresponding to his Home DR project (first DR project subscription). He holds a copy of his leaf secret key and all keys corresponding to the nodes in the path from his leaf to the root in this tree; (b) a user has all group keys of the other DR projects to which he is subscribed; (c) if a user leaves his Home DR project and remains subscribed to one or more DR projects, he will shift to a new OFT (this OFT will be the tree corresponding to his new Home DR). These features ensure that a user will not subscribe and pay for the same DR project multiple times.

An example of the key graph is given according to Fig. 3, the MDMS provides 4 DR projects. Some users subscribe to only one DR project (e.g. u_2 subscribes only to DR_1), while other users may subscribe to multiple DR projects simultaneously (e.g. u_1 subscribes to DR_1, DR_2, and DR_3).

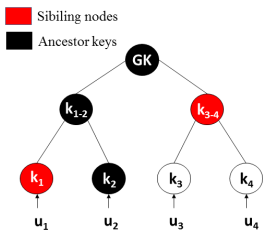


Fig. 2: Example of OFT key tree illustrating the ancestors and their corresponding sibling nodes of member u_2

In this figure, no user subscribes to both DR_2 and DR_3 at the same time. We next illustrate both member join and leave procedures executed by the MDMS when receiving a member join or leave request.

2) Rekeying operations: In our solution, when a user subscribes or unsubscribes to/from a DR project, rekeying consists of 3 operations: joining/leaving a OFT, shifting among trees, and receiving new keys for new subscriptions.

a) Leave procedure: The leave procedure deals with the case when a user unsubscribes from a DR project (u_i leaves DR_j).
Let $\phi_j = \{u_l / u_l$ subscribed to $DR_j\}$,
Let $\mathcal{X}_{jk} = \{u_l / u_l \in \phi_j$ and $Home\_DR\ (u_l) = DR_k\}$,
Let $\omega_k = \{u_l / u_l \in \mathcal{X}_{jk}$ and $DR_k \in set(u_i)\}$,
Let $\delta_k = \{u_l / u_l \notin \mathcal{X}_{jk}$ and $DR_k \in set(u_i)\}$.

- **Case 1:** We consider a user who subscribed to one or multiple DR projects and leaves his Home DR project: The MDMS updates and renews keys according to Algorithm 1.

Algorithm 1 : Update keys when user leaves Home DR

Function leaveHomeDR (u_i, DR_j) ;

1: Apply standard OFT approach in DR_j tree to update GK_j (GK'_j represents the new group key);

2: **If** $Nsub(u_i) = 1$:

3: MDMS $\rightarrow \mathcal{X}_{jk}$:

$$\bigcup Enc(Enc(GK'_j, GK_k), GK_j)$$

4: **Else** :

5: MDMS $\rightarrow \omega_k$:

$$Enc(Enc(GK'_j, k_{right(GK_k)}), GK_j)$$
$$Enc(Enc(GK'_j, k_{left(GK_k)}), GK_j)$$

6: MDMS $\rightarrow \delta_k$:

$$\bigcup Enc(Enc(GK'_j, GK_k), GK_j)$$

7: Shift user u_i to OFT tree corresponding to his second subscription DR_x using standard OFT apporach (without updating key GK_x that u_i already has)

Example: Let us consider the key graph in Fig. 3. When u_5 (user who subscribed to DR_1, DR_2 and DR_3) leaves DR_1: (a) standard OFT approach is used to replace keys

in the key tree corresponding to DR_1: the two leaf nodes (k_5 and k_{4-5}) will be removed from the key tree and new individual key k'_4 is established between the MDMS and u_4:

$$\text{MDMS} \rightarrow \{u_6, u_7\} : Enc(f(k'_4), k_{6-7}) \quad (2)$$

$$\text{MDMS} \rightarrow \{u_1, u_2, u_3\} : Enc(f(k_{4-7}), k_{1-3}) \quad (3)$$

(b) update GK_1 for users in ω_k:

$$\text{MDMS} \rightarrow \omega_2 : Enc(Enc(GK'_1, k_{8-10}), GK_2) \quad (4)$$

$$Enc(Enc(GK'_1, k_{11-13}), GK_2) \quad (5)$$

$$\text{MDMS} \rightarrow \omega_3 : Enc(Enc(GK'_1, k_{14-15}), GK_3) \quad (6)$$

$$Enc(Enc(GK'_1, k_{16-17}), GK_3) \quad (7)$$

(c) update GK_1 for users in δ_k:

$$\text{MDMS} \rightarrow \delta_4 : Enc(Enc(GK'_1, GK_4), GK_1) \quad (8)$$

(d) shift u_1 the OFT key tree corresponding to DR_2 which becomes his new home DR project using standard OFT approach (without updating GK_2 that u_1 already has).

- **Case 2:** We consider a user who is subscribed to multiple DR projects and leaves one DR project which is not his Home DR project (u_i leaves DR_j) The MDMS updates and renews keys according to Algorithm 2.
Let $DR_x = Home\_DR(u_i)$,
Let $\psi_k = \{u_l / u_l \in \omega_k \text{ and } DR_k \neq DR_x\}$,
Let $\pi_i = \{k_l / k_l = right(k_c) \text{ or } k_l = left(k_c),\ k_c \in Path(u_i)\}$.

Algorithm 2: Update keys when user leaves DR project

Function leaveDR (u_i, DR_j) ;

1: Update GK_j (GK'_j is the new group key);

2: MDMS $\rightarrow \mathcal{X}_{jj}$:

$$Enc(GK'_j, k_{right(GK_j)})$$

$$Enc(GK'_j, k_{left(GK_j)})$$

3: MDMS $\rightarrow \mathcal{X}_{jx}$:

$$\bigcup_{k_\alpha \in \pi_i} Enc(Enc(GK'_j, k_\alpha), GK_j)$$

4: MDMS $\rightarrow \psi_k$:

$$Enc(Enc(GK'_j, k_{right(GK_k)}), GK_j)$$

$$Enc(Enc(GK'_j, k_{left(GK_k)}), GK_j)$$

5: MDMS $\rightarrow \delta_k$:

$$\bigcup Enc(Enc(GK'_j, GK_k), GK_j)$$

Example: When u_1 (user who subscribed to DR_1, DR_2 and DR_3) leaves DR_2 : (a) update GK'_2 for users in $\mathcal{X}_{22}$:

$$\text{MDMS} \rightarrow \{u_8, u_9, u_{10}\} : Enc(GK'_2, k_{8-10}) \quad (9)$$

$$\text{MDMS} \rightarrow \{u_{11}, u_{12}, u_{13}\} : Enc(GK'_2, k_{11-13}) \quad (10)$$

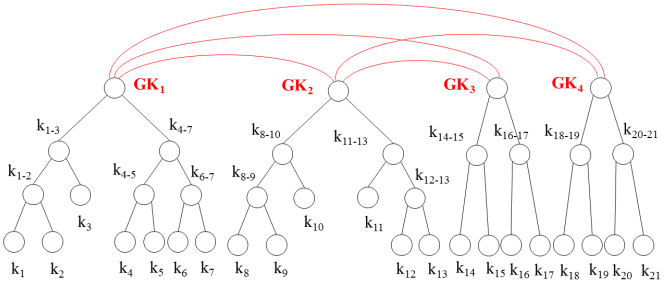


Fig. 3: Example of our multi-group key graph structure

(b) update GK'_2 for users in $\mathcal{X}_{21}$ using a double encryption to ensure that only users susbscribing to DR_2 can obtain the new key (suppose u_6 and u_7 subscribed to DR_2) :

$$\text{MDMS} \rightarrow \{u_6, u_7\} : Enc(Enc(GK'_2, k_{6-7}), GK_2) \quad (11)$$

(c) update GK'_2 for users in ψ_k:

$$\text{MDMS} \rightarrow \psi_3 : Enc(Enc(GK'_2, k_{14-15}), GK_2) \quad (12)$$

$$\text{MDMS} \rightarrow \psi_3 : Enc(Enc(GK'_2, k_{16-17}), GK_2) \quad (13)$$

(d) update GK'_2 for users in δ_k:

$$\text{MDMS} \rightarrow \delta_4 : Enc(Enc(GK'_2, GK_4), GK_2) \quad (14)$$

b) Join procedure: The join procedure deals with the case when a user subscribes to a new DR project (u_i joins DR_j). The MDMS apply the join rekeyin Algorithm 3.

Algorithm 3 : Update keys when user joins a DR project

Function joinDR (u_i, DR_j) ;

1: $GK'_j = H(GK_j)$;

2: **If** $Nsub(u_i) \geq 1$:

3: Send the new group key GK'_j to u_i ;

4: Send a notification to all users in ϕ_j about the application of the one-way function;

5: **Else** :

6: Send a notification to all users in $\mathcal{X}_{jk}$ about the application of the one-way function;

7: Apply standard OFT approach in DR_j tree whithout updating GK_j.

V. Performance Evaluation

In this section we present a security and performance analysis of our solution and prove its safety and efficiency.

A. Security Analysis

1) Forward and Backward Sercery: The proposed key management scheme supports both backward secrecy and forward secrecy. When a new user joins a DR project, he cannot learn previous group keys because he does not have access to previous node keys that were used to compute these group. When a user leaves a DR project, all affected keys (those known by the departing user in both lower and upper level) will be changed and redistributed securely which prevents the departing user from having access to the new keys and hence forward secrecy is preserved.

2) Collusion freedom: Any set of users unsubscribed from a set of DR projects cannot deduce the current used DR projects keys, because all affected keys when any user leaves a DR project will be updated and new keys are independent.

B. Performance Analysis

1) Storage Cost: We approximate the storage cost with the number of symmetric keys stored in the MDMS/SMs, and used for unicast, broadcast and multicast transmissions (individual keys, group keys and broadcast key). We compare our scheme with the schemes proposed in [11] and [12] (Table II).

2) Communication Cost: Our solution relies on an efficient multi-group key graph structure. Rekey operations (join, leave, and shift) introduce extra rekey cost. In the joining/leaving scenario, even though the number of group members and subscribed DR projects are the same, the number of keys to be updated varies according to the positions of the joining/leaving member in the multi-group key graph.

a) Leave procedure: According to Algorithm 1 and Algorithm 2, the communication cost in the worse cases will be as follows:

- **Case 1:** When u_i leaves his Home DR project DR_j (user subscribed only to one DR project) :

$$comCost = (h_j + N_{pr} - 1)|K| \tag{15}$$

$|K|$: the size of the key in bit.

- **Case 2:** When u_i leaves his Home DR project DR_j (user subscribed to multiple DR projects at the same time):

$$comCost = (h_j + 2.A + B + h_k)|K| + c \tag{16}$$

$A = Nsub(u_i)$
$B = N_{pr} - Nsub(u_i)$
h_k : the hight of the new Home DR project.
The " $+ c$" term is to specify on which group key we must apply the one-way function $c = log_2\ (N_{pr})$.

- **Case 3:** When u_i leaves one DR project DR_j which is not his Home DR project DR_l:

$$comCost = (2 + h_l + 2.A + B)|K| \tag{17}$$

b) Join procedure: According to Algorithm 3, the communication cost will be as follows:

- **Case 1:** When u_i joins his Home DR project DR_j :

$$comCost = h_j|K| + c \tag{18}$$

- **Case 2:** When u_i joins a new DR project DR_j which is not his Home DR project DR_l:

$$comCost = |K| + c \tag{19}$$

TABLE II: Storage Cost

Scheme	Storage Overhead	
	MDMS	SM_i
Liu's *et al.*, 2013 [11]	$n + N_{pr} + 1$	$Nsub(u_i) + 2$
SKM+, 2014 [12]	$2\sum_{j=1}^{N_{pr}}(m_j - 1) + 1$	$\sum_{j=1}^{Nsub(u_i)}(log_2\ m_j + 1) + 1$
SAMI	$2\sum_{j=1}^{N_{pr}}(m_j - 1) + 1$	$log_2\ (\|Home\_DR(u_i)\|) + Nsub(u_i) + 1$

* n is the number of SMs, N_{pr} is the number of DR projects, m_j is the number of j^{th} DR project members, $Nsub(u_i)$ is the number of DR projects to which subscribes user u_i.

3) Simulation:

- **Simulation Model:** We consider a smart grid with 1 million users. The utility provides 15 DR projects to users (for example, Real Time Pricing program, Time Of Use Pricing program, Critical Pick Pricing program, ... etc). We assume that users arrival is modeled as a Poisson process with parameter λ (users/months), and given that there are no statistical studies of DR projects membership behavior for the moment, we assume that membership duration in each DR projects follows an Exponential law with parameter μ.

  A typical user session starts by a *join* event, which can be followed by one or more *join/leave* events to/from other DR projects. At the end of a membership in a DR project, a user leaves this DR project. We will consider a session of 24 months. Average arrival rate λ is of 10000 users/month, and average membership duration μ is 4 months. We will use a 128b long symmetric keys, and balanced OFT trees. Storage and communication costs of Liu's *et al.* and SKM+ KMS are readily obtained from [11] and [12].

- **Simulation Results :**

  Storage Cost:

For MDMS, the storage cost can be afforded using special key servers. In contrast, the storage capacity of SMs is limited to 4-12 KB [23]. Fig. 4 shows a comparison of average storage cost induced at SMs between the three schemes with respect to the number of subscribed DR projects. We considered a 100.000 fixed size of DR projects. We can see that in our scheme, a SM stores fewer keys than that in [12] (reduction reaches 83% while a user can subscribe to 15 DR projects at the same time) and little more keys than that in [11]. This can be explained as follows: the scheme proposed by Liu *et al.* does not adopt a key graph technique, a SM stores one key for each subscribed DR project which induces a high communication overhead even the storage overhead remains relatively low. In SKM+, authors used a OFT tree for each DR project, the number of keys stored will increase significantly when a user subscribes to new DR projects. Whereas, in SAMI we see that the number of subscribed DR projects does not affect significantly the storage cost.

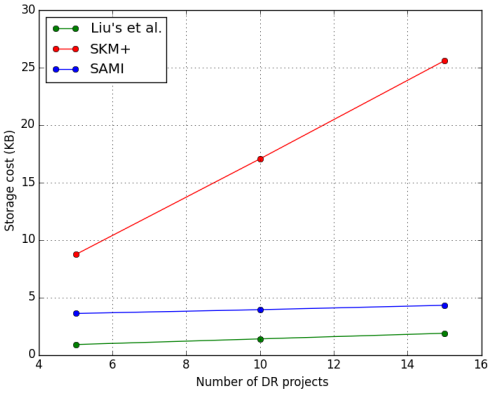
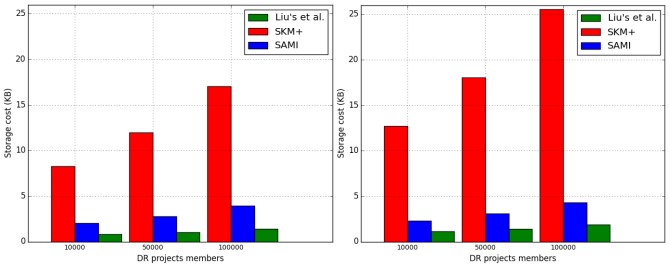


Fig. 4: Average storage cost in SMs according to number of subscribed DR projects

Fig. 5 (a) and (b) shows a comparison of average storage cost in SMs between the three schemes with respect to DR projects' size. A user can subscribe respectively to 10 and 15 DR projects at the same time. In Liu's *et al.* scheme, the projects's size does not affect significantly the storage cost, SMs store only the group keys. Whereas in SKM+ and SAMI the DR projects' size affects the storage cost, as the number of users increases, the storage cost increases due to the rise of the height of the used key trees, but we can see that SMs store much fewer keys in SAMI with respect to SKM+.

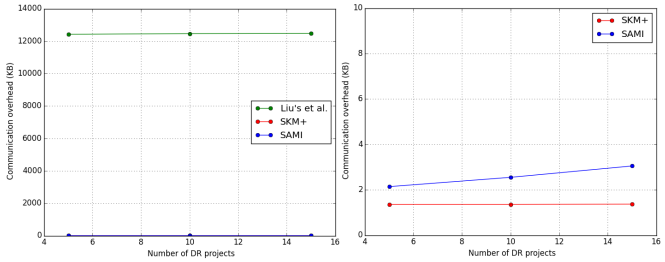
Communication Cost:

Fig. 6 (a) and (b) shows a comparison of average communication cost per event (join/leave) with respect to the number of subscribed DR projects at the same time. We assume that there are 100.000 users (in average) subscribing to each DR project. We notice that the bandwidth overhead due to a join is the same as a leave for the Liu's *et al.* scheme and it is remarkably higher than that of scheme because of the inefficient multicast key management. In SAMI, bandwith overhead reduction reaches 99% with respect to Liu's *et al.* scheme while a user can subscribe to 15 DR projects simultaneously. Note that although SKM+ has less communication overhead than SAMI for join/leave event, the difference is not significant and it is too little to be seen in the Fig. 6 (a).



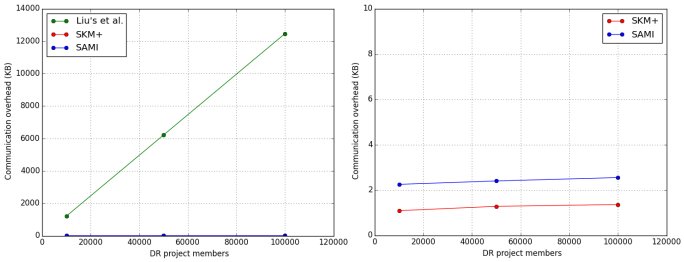
(a) Number of DR projects = 10 (b) Number of DR projects = 15

Fig. 5: Average storage cost with respect to DR projects' size



(a) (b) Zoom of (a)

Fig. 6: Average communication cost by event with respect to number of DR projects



(a) (b) Zoom of (a)

Fig. 7: Average communication cost by event with respect to DR projects' size

Fig. 7 (a) and (b) shows a comparison of average communication cost per event for the three schemes with respect to the number of subscribers in DR projects while fixing the number of DR projects to 10 DR projects. Fig. 7(a) shows that in Liu's *et al.* scheme the bandwidth overhead increases proportionally with the increase of the number of subscribers in DR projects. Whereas, the bandwidth overhead remains much lower in SKM+ and SAMI as shown in Fig. 7(b) (the bandwidth overhead of SKM+ and SAMI is too little to be seen in Fig. 7(a)). Certainly, our scheme introduces extra communication cost compared to SKM+, but this overhead is minor regarding the overall advantages of the proposed scheme, and mainly when considering the storage cost as shown above.

VI. Conclusion

In this paper, we proposed a new key management scheme for AMI in smart grid. It is an efficient and scalable key management scheme supporting unicast, multicast, as well as broadcast communications. The proposed scheme use a novel multi-group key graph technique that supports the management of multiple Demand Response projects simultaneously for each customer and induces low storage overhead compared to existing solutions without increasing the communication overhead. Moreover, the proposed KMS guarantees both forward and backward secrecy.

REFERENCES

[1] Ye Yan, Yi Qian, H. Sharif, D. Tipper, "A Survey on Smart Grid Communication Infrastructures: Motivations, Requirements and Challenges," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 5-20, First Quarter 2013.

[2] Z.M. Fadlullah *et al.*, "Toward Intelligent Machine-to-Machine Communications in Smart Grid," *IEEE Communications Magazine*, vol. 49, no. 4, pp. 60-65, Apr. 2011.

[3] R.R. Mohassel, A. Fung, F. Mohammadi, and K. Raahemifar, "A survey on Advanced Metering Infrastructure," *International Journal of Electrical Power & Energy Systems*, vol. 63, pp. 473-484, Dec. 2014.

[4] F.M. Cleveland, "Cyber Security Issues for Advanced Metering Infrasttructure (AMI)," *IEEE Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century*, pp. 1-5, Jul. 2008.

[5] J. Kamto, L. Qian, J. Fuller, and J. Attia, "Light-weight key distribution and management for Advanced Metering Infrastructure", *IEEE GLOBECOM Workshops (GC Wkshps)*, pp. 1216-1220, 2011.

[6] Y. Yan, Y. Qian, and H. Sharif, "A secure and reliable in-network collaborative communication scheme for advanced metering infrastructure in smart grid," *IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 909-914, Mar. 2011.

[7] Q. Li, G. Cao, "Multicast authentication in the smart grid with one time signature," *IEEE Trans. Smart Grid*, vol. 2, no. 4, pp. 686-696, Dec. 2011.

[8] H. Nicanfar, P. Jokar, and V.C.M. Leung, "Smart grid authentication and key management for unicast and multicast communications," *IEEE PES Innovative Smart Grid Technologies Asia (ISGT)*, pp. 1-8, Nov. 2011.

[9] D. Wu, C. Zhou, "Fault-tolerant and scalable key management for smart grid," *IEEE Trans. Smart Grid*, vol. 2, no. 2, pp. 375-381, Jun. 2011.

[10] J. Xia, Y. Wang, "Secure Key Distribution for the Smart Grid," *IEEE Trans. Smart Grid*, vol. 3, no. 3, pp. 1437-1443, 2012.

[11] N. Liu, J. Chen, L. Zhu, J. Zhang, and Y. He , "A Key Management Scheme for Secure Communications of Advanced Metering Infrastructure in Smart Grid," *IEEE Trans. on Ind. Electron.*, vol. 60, no. 10, pp. 4746-4756, Oct. 2013.

[12] Z. Wan, G. Wang, Y. Yang, and S. Shi, "SKM: Scalable Key Management for Advanced Metering Infrastructure in Smart Grids," *IEEE Trans. Ind. Electron.*, vol. 61, no. 12, pp. 7055-7066, Dec. 2014.

[13] R. Shein, "Security Measures for Advanced Metering Infrastructure Components," *Power and Energy Engineering Conference (APPEEC), Asia-Pacific*, pp. 1-3, Mar. 2010.

[14] W. Diffie, M.E. Hellman, "New directions in cryptography, " *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644-654, Nov. 1976

[15] A. Shamir, "Identity-based cryptosystems and signature schemes," *Advances in Cryptology: Proceedings of CRYPTO 84, Lecture Notes in Computer Science*, vol 196, pp. 47-53, 1984.

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

[17] V.S. Miller, "Use of Elliptic Curves in Cryptography," *Advances in Cryptology : Proceedings of CRYPTO 85, Lecture Notes in Computer Science*, vol. 218, pp. 417-426, 1986.

[18] C. K. Wong, M. Gouda, and S. Lam, "Secure group communication using key graphs," *IEEE/ACM Trans. Netw.*, vol. 8, no. 1, pp. 16-30, Feb. 2000.

[19] L. Chen, C. Kudla, "Identity based authenticated key agreement protocols from pairings," *Proc. IEEE CSFW, Pacific Grove, CA, USA*, pp. 219-233, Jun. 2003.

[20] D. A. McGrew, A. T. Sherman, "Key establishment in large dynamic groups: Using one-way function trees," *IEEE Trans. Softw. Eng.*, vol. 29, no. 5, pp. 444-458, May 2003.

[21] T. Sauter, M. Lobashov, "End-to-End Communication Architecture for Smart Grids," *IEEE Trans. Ind. Electron.*, vol. 58, no. 4, pp. 1218-1228, Apr. 2011.

[22] NIST Special Publication 800-56A (2007, Mar. 8), *Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography (Revised)*, [Online]. Available: ⟨http://csrc.nist.gov/groups/ST/toolkit/documents/SP800-56Arev1_3-8-07.pdf⟩.

[23] W. Wang, Z. Lu, "Cyber security in the Smart Grid: Survey and challenges," *Comp. Networks*, vol. 57, no. 5, pp. 1344-1371, Apr. 2013.