eSKAMI: Efficient and Scalable Multi-group Key Management for Advanced Metering Infrastructure in Smart Grid

Mourad Benmalek, Yacine Challal

eSKAMI: Efficient and Scalable multi-group Key management for Advanced Metering Infrastructure in Smart Grid

Mourad Benmalek
Laboratoire de Méthodes de Conception de Systèmes
Ecole nationale Supérieure d'Informatique, ESI
Alger, Algérie
Email: m_benmalek@esi.dz

Yacine Challal
Laboratoire de Méthodes de Conception de Systèmes
Ecole nationale Supérieure d'Informatique, ESI
Centre de Recherche sur l'Information Scientifique et Technique, CERIST, Alger, Algérie
Email: y_challal@esi.dz

***Abstract*—Advanced Metering Infrastructure (AMI) is composed of systems and networks for measuring, collecting, storing, analyzing, and exploiting energy usage related data. AMI is an enabling technology for Smart Grid (SG) and hence represents a privileged target for security attacks with potentially great damage against infrastructures and privacy. For this reason, security has been identified as one of the most challenging topics in AMI development, and designing an efficient Key Management Scheme (KMS) is one of first important steps. In this paper, we propose a new scalable and efficient key management scheme that we call Efficient and Scalable multi-group Key Management for AMI (eSKAMI) 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 comparision of our scheme with recently proposed schemes show that our KMS induces low storage overhead compared to existing solutions (reduction reaches 83%) without increasing the communication overhead.**

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

I. Introduction

A Smart Grid (SG) is an electrical grid that is enhanced with communications and networking, computing, and signal processing technologies [1]. Two-way flows of electricity and real-time energy related information (production, transport, distribution and consumption) issued by smart devices and smart meters, brings new perspectives to energy management and optimization in the SG. 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]. To achieve an intelligent grid, a succession of sub-systems should be realized [3] : Advanced Metering Infrustructure (AMI), Advanced Distrubtion Operations (ADO), Advanced Transmission Operations (ATO) and Advanced Assent Management (AAM).

Advanced Metering Infrastructure (AMI) is a key element in the smart grid. It is responsible for collecting all the data and information from loads and consumers. AMI is also responsible for implementing control signals and commands to perform necessary control actions [3]. A typical AMI involves Smart Meters (SMs), Home Area Networks (HANs), wide area communications infrastructure, and Meter Data Management Systems (MDMS). 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 customers 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 traditionnal systems, availability of information and control commands generated and managed by AMI is compulsory for the operation of the whole smart grid. which contains much more 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 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 meeting the security requirements of AMI.

The remainder of this paper is organized as follows. We discuss related work in Section II. In Section III we study the architecture of an AMI and the key management function requirements. In Section IV we present our KMS which is an efficient key management scheme that can support unicast, multicast and broadcast communications. We give a security and performance analysis of our KMS in Section VI. Finally, we draw our conclusions and future works in Section VII.

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.

AMI devices authentication, and confidentiality for user privacy and user behavior is an issue that still lacks a complete solution. 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 prsotocol 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 srid 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 the 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 *et al.*'s scheme is not tolerant to packet loss. Wan *et al.* [12] proposed an improvement for Liu *et al.*'s 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 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.

III. AMI System Structure and KMS Security Requirements

In this section we analyze AMI system structure to identify the basic requirements that are relevant to key management.

A. AMI System Structure

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.

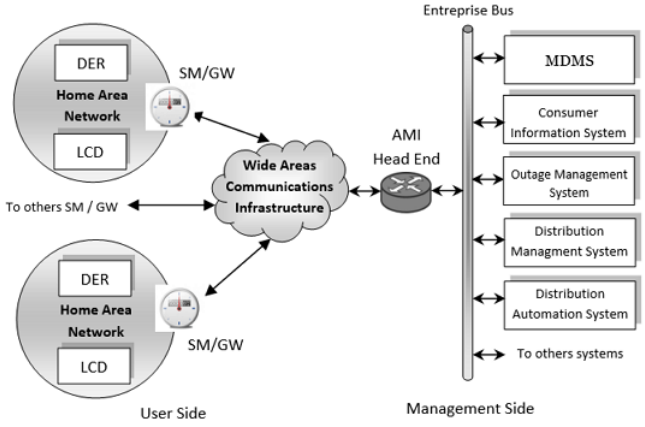


Fig. 1: System structure of AMI

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. 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 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. Proposed Solution

We introduce a new scalable and efficient key management scheme that we call Efficient and Scalable multi-group Key management for secure data communications in an Advanced Metering Infrastructure (eSKAMI). 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.

TABLE I: Notation Table

Notation	Description
$H(.)$	A One-way hash functions
n	Number of SMs
m_i	Number of the i^{th} DR project members
d	LKH trees degree
h_i	Height of i^{th} LKH tree $h_i = log_d(m_i)$
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
$Child_i(GK_j)$	The i^{th} child of (GK_j) in LKH 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

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 establishement 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.

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, LKH (Logical Key Hierarchy) [18] is mostly used in the literature. 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 as shown in Fig. 2. In LKH, each member holds a copy of its leaf secret key and all the keys corresponding to the nodes in the path from its leaf to the root. Hence, if a user u_i subscribes to two or more DR projects simultaneously (e.g. DR_j and DR_k), she/he needs to manage two sets of keys. As a result, directly applying LKH may be costly and has a non-negligible overhead for key storage.

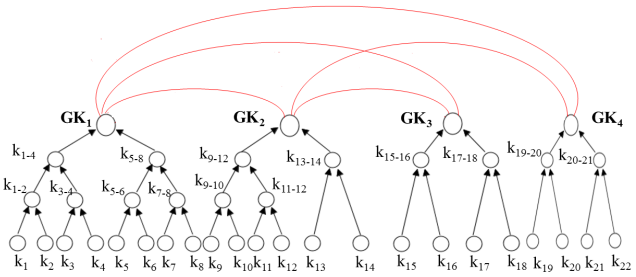


Fig. 2: Example of our multi-group key graph structure

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. For instance (as illustrated in Fig. 2), a user u_1 in $DR_1 \cap DR_2 \cap DR_3$ does not need to manage three set of keys to handle the three DR projects. Indeed, in our solution u_1 will hold only keys on its path to group key corresponding to her/his last subscription. Moreover, when a user joins or leaves a DR project, the communication cost for rekeying operations will not increase significantly compared to the cost induced by using separate LKH tree inside each DR project.

1) Multi-group Key Graph Structure: Our multi-group key graph structure can be modeled as shown in Fig. 2: in the *lower level*, each LKH tree 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 combinaisons of root keys for users subscribing to multiple DR projects at the same time. Our multi-group key graph has the following proprieties: (a) a user only belongs to one LKH tree in the multi-group key graph corresponding to her/his Home DR (first DR project subscription). She/He holds a copy of its leaf secret key and all keys corresponding to the nodes in the path from its leaf to the root in this tree; (b) a user has all group keys of the other DR projects to which she/he is subscribed; (c) if a user leaves her/his first DR project and remains subscribed to one or more DR projects, she/he will shift to a new LKH tree (this LKH tree will be the tree corresponding to her/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. 2., the MDMS provides 4 DR projects. Some users subscribe to only one of DR projects (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). 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 leaves a DR project, rekeying consists of 3 operations: joining/leaving an LKH tree, shifting among LKH trees, and receiving new keys for new subscriptions. Table II. lists the rekey operations and their corresponding user events.

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 $\omega_{1k} = \{u_l / u_l \in \chi_{jk}$ and $DR_k \in set(u_i)\}$,
Let $\omega_{2k} = \{u_l / u_l \notin \chi_{jk}$ and $DR_k \in set(u_i)\}$.

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

Algorithm 1 : Update keys when user leaves Home DR

Function leaveHomeDR (u_i, DR_j) ;
Update GK_j (GK'_j is the new group key);
Apply standard LKH approach in DR_j tree;
If $Nsub(u_i) = 1$:
  MDMS $\rightarrow \mathcal{X}_{jk}$:

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

Else :
  MDMS $\rightarrow \omega_{1k}$:

$$\bigcup_{1 \le h \le d} Enc(Enc(GK'_j, Child_h(GK_k)), GK_j)$$

  MDMS $\rightarrow \omega_{2k}$:

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

  Shift user u_i to LKH tree corresponding to his/her second subscription DR_x using standard LKH apporach (without updating key GK_x that u_i already has)

Example 1: When u_2 (user who subscribed only to DR_1) leaves DR_1 : (a) standard LKH approach is used to replace keys in the key tree corresponding to DR_1:

$$\text{MDMS} \rightarrow \{u_1\} : Enc(GK'_1, k_1) \quad (1)$$
$$Enc(GK'_{1-4}, k_1) \quad (2)$$

$$\text{MDMS} \rightarrow \{u_3, u_4\} : Enc(GK'_1, k_{3-4}) \quad (3)$$
$$Enc(k'_{1-4}, k_{3-4}) \quad (4)$$

$$\text{MDMS} \rightarrow \{u_5, u_6, u_7, u_8\} : Enc(GK'_1, k_{5-8}) \quad (5)$$

(b) update GK'_1 for users in $\mathcal{X}_{1k}$: first, we encrypt the new key GK'_1 with keys GK_k to ensure that only users belonging to DR_k tree can obtain the relevant key, and the second encryption with GK_1 to ensure that only users subscribing to DR1 can obtain the key.

$$\text{MDMS} \rightarrow \mathcal{X}_{12} : Enc(Enc(GK'_1, GK_2)), GK_1) \quad (6)$$
$$\text{MDMS} \rightarrow \mathcal{X}_{14} : Enc(Enc(GK'_1, GK_4)), GK_1) \quad (7)$$

Example 2: When u_1 (user who subscribed to DR_1, DR_2 and DR_3) leaves DR_1 : (a) standard LKH approach is used to replace keys in the key tree corresponding to DR_1:

$$\text{MDMS} \rightarrow \{u_2\} : Enc(GK'_1, k_2) \quad (8)$$
$$Enc(GK'_{1-4}, k_2) \quad (9)$$

$$\text{MDMS} \rightarrow \{u_3, u_4\} : Enc(GK'_1, k_{3-4}) \quad (10)$$
$$Enc(k'_{1-4}, k_{3-4}) \quad (11)$$

$$\text{MDMS} \rightarrow \{u_5, u_6, u_7, u_8\} : Enc(GK'_1, k_{5-8}) \quad (12)$$

(b) update GK_1 for users in ω_{1k}:

$$\text{MDMS} \rightarrow \omega_{12} : Enc(Enc(GK'_1, k_{9-12}), GK_2) \quad (13)$$
$$Enc(Enc(GK'_1, k_{13-14}), GK_2) \quad (14)$$
$$\text{MDMS} \rightarrow \omega_{13} : Enc(Enc(GK'_1, k_{15-16}), GK_3) \quad (15)$$
$$Enc(Enc(GK'_1, k_{17-18}), GK_3) \quad (16)$$

(c) update GK_1 for users in ω_{2k}:

$$\text{MDMS} \rightarrow \omega_{24} : Enc(Enc(GK'_1, GK_4), GK_1) \quad (17)$$

(d) shift u_1 the LKH tree corresponding to DR_2 which becomes her/his new home DR project using standard LKH 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 her/his Home DR project: The MDMS updates and renews keys according to Algorithm 2.
Let $DR_x = Home\_DR(u_i)$,
Let $\omega_{3k} = \{u_l / u_l \in \omega_{1k} \text{ and } DR_k \neq DR_x\}$.

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 $\rightarrow \phi_j$:

$$\bigcup_{1 \leq h \leq d} Enc(GK'_j, Child_h(GK'_j))$$

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

$$\bigcup_{\substack{k_{\alpha x} \text{shared keys} \\ \text{in } DR_k \text{ tree}}} Enc(Enc(GK'_j, k_{\alpha x}), GK_j)$$

4 MDMS $\rightarrow \omega_{3k}$:

$$\bigcup_{1 \leq h \leq d} Enc(Enc(GK'_j, Child_h(GK_k)), GK_j)$$

5 MDMS $\rightarrow \omega_{2k}$:

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

Example 3: When u_1 (user who subscribed to DR_1, DR_2 and DR_3) leaves DR_2 : (a) update GK'_2 for users in ϕ_2:

$$\text{MDMS} \rightarrow \{u_9, u_{10}, u_{11}, u_{12}\} : Enc(GK'_2, k_{9-12}) \quad (18)$$
$$\text{MDMS} \rightarrow \{u_{13}, u_{14}\} : Enc(GK'_2, k_{13-14}) \quad (19)$$

(b) update GK'_2 for users in ϕ_1 using a double encryption to ensure that only users susbscribing to DR_2 can obtain the new key (suppose u_5 and u_7 subscribed to DR_2) :

$$\text{MDMS} \rightarrow \{u_5, u_7\} : Enc(Enc(GK'_2, k_{5.8}), GK_2) \quad (20)$$

(c) update GK'_2 for users in ω_{3k}:

$$\text{MDMS} \rightarrow \omega_{33} : Enc(Enc(GK'_2, k_{15-16}), GK_2) \quad (21)$$
$$\text{MDMS} \rightarrow \omega_{33} : Enc(Enc(GK'_2, k_{17-18}), GK_2) \quad (22)$$

(d) update GK'_2 for users in ω_{2k}:

$$\text{MDMS} \rightarrow \omega_{24} : Enc(Enc(GK'_2, GK_4), GK_2) \quad (23)$$

TABLE II: Rekey Operations and User Events

Operations	User events
Join an LKH tree	A user has subscribed to only default DR project. Then, he subscribes to a new DR project.
Leave an LKH tree	A user has subscribed to only one DR project. Then, he leaves this DR project.
Shift among LKH trees	A user has subscribed to multiple DR projects. Then, she/he leaves her/his Home DR project.

b) Join procedure: The join procedure deals with the case that a user subscribes to a new DR project (u_i joins DR_j).

- **Case 1:** We consider a user who joins her/his first DR project: The MDMS apply the join rekeyin Algorithm 3.

Algorithm 3 : Update keys when user joins Home_ DR

Function joinHomeDR (u_i, DR_j) ;
1 $GK'_j = H(GK_j)$;
2 Send a notification to all users in $\mathcal{X}_{jk}$ about the application of the one-way function;
3 Apply standard LKH approach in DR_j tree.

- **Case 2:** We consider a user who joins one DR project which is not her/his Home DR project: The MDMS updates and renews keys according to Algorithm 4.

Algorithm 4: Update keys when user joins DR project

Function joinHomeDR (u_i, DR_j) ;
1 $GK'_j = H(GK_j)$;
2 Send the new group key GK'_j to u_i ;
3 Send a notification to all users in ϕ_j about the application of the one-way function.

D. Secure Unicast, Multicast, and Broadcast Communications

In the unicast, multicast, and broadcast communication process, the confidentiality and integrity of the messages should be provided. For this purpose, individual keys $\{k_1, ..., k_n\}$, broadcast key $\{GK_0\}$, and group keys $\{GK_1, ..., GK_m\}$ are used to secure interactive messages in AMI exchanged between the MDMS and SMs. Our scheme adopts the following message transmission methods:

1) Secure Unicast Communications: Suppose the individual key k_i is established between the MDMS and a smart meter SM_i. When the MDMS (resp. SM_i) wants to send a message M to SM_i (resp. MDMS), the MDMS (resp. SM_i): (a) generates a session key sk_i from individual user key k_i; (b) encrypts, authenticates, and sends the following message:

$$\text{MDMS} \rightarrow SM_i : (Enc(M, sk_i) \parallel HMAC_{sk_i}(M)) \quad (24)$$

On receiving the message, SM_i (resp. MDMS) generates the session key sk_i and then *verifies* and *decrypts* the message M.

2) Secure Multicast, and Broadcast Communications: The same transmission method used to secure unicast communications is adopted to send a secure message M from MDMS to all users or subscribed users to a specific DR project. The MDMS: (a) generates a session key sgk_j (resp. sgk_0) from j^th DR project group key GK_j (resp. broadcasts key GK_0) (b) encrypts, authenticates, and sends the following message: MDMS $\rightarrow \{SM_i\}$:

$$(Enc(M, sgk_j) \| HMAC_{skg_j}(M)) \quad (25)$$

MDMS $\rightarrow \{SM_i, ..., SM_n\}$:

$$(Enc(M, sgk_0) \| HMAC_{skg_0}(M)) \quad (26)$$

V. Performance Evaluation

In this section we present a security and performance analysis of our solution and prove its safety and efficiency. In Table III we compare our scheme with two recently proposed schemes in [11] and [12].

TABLE III: Comparison of Key Management Schemes

		Liu's *et al.*, 2013 [11]	**SKM+, 2014 [12]**	**eSKAMI**
Key graph techniques		No	Yes	Yes
Hybride Transmission		Yes	Yes	Yes
Backward and forward secrecy		Yes	Yes	Yes
Collusion freedom		Yes	Yes	Yes
Overhead	Commun.	High	Very Low	Low
	storage	Low	High	Low

A. Security Analysis

1) Confidentiality and Integrity: As mentioned above, the session keys used to ensure the communciations are generated before every session. These session keys are held only by the two communication ends which garantees message confidentiality. Moreover, receiver verifies the MAC code of encrypted message using these session keys to ensure message integrity.

2) Forward and Backward Sercery: The proposed key management scheme supports both backward secrecy and forward secrecy. Aftze a new node joins a DR project, all users applies a one-way hash function to the affected keys. That ensures that none of the old key can be recovered by the new coming user which garantees backward secrecy. When a user leaves a DR project, all affected keys (those known be the departing user in both lower and upper level) will be changed and redistributed securely which prevents the departing customer from having acces to the new keys and hence forward secrecy is preserved.

3) Collusion freedom: Any set of users unsubscribe a set of DR projects can not be able to deduce the current used DR projects keys, because all affected keys when any user leaves a DR project will be update and new keys are independents.

B. Performance Analysis

1) Storage Cost: On the aspect of the storage cost, we mainly focus on 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 these proposed in [11] and [12] as shown in Table IV (we use balanced binary LKH trees).

TABLE IV: Storage Cost

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

2) Communication Cost: The solution we proposed uses 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 her/his Home DR project DR_j (user subscribed only to one DR project) :

$$comCost = (2h_j + N_{pr} - 1)|K| \quad (27)$$

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

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

$$comCost = (2h_j + d.A + B + 2h_k)|K| + c \quad (28)$$

  h_k : the hight of the new Home DR project.
  $A = Nsub(u_i)$
  $B = N_{pr} - Nsub(u_i)$
  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 her/his Home DR project DR_l:

$$comCost = (d + 2h_l + d.A + B)|K| \quad (29)$$

b) Join procedure: According to Algorithm 3 and Algorithm 4, the communication cost will be as follows:

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

$$comCost = 2h_j|K| + c \tag{30}$$

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

$$comCost = |K| + c \tag{31}$$

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 arrivals are 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 law with parameter μ. Our assumption stays very close to reality.

  A typical user session starts by a *join* event, which can be followed by one or more *join/leave* to/from other DR projects events. At the end of a membership in a DR project, a user leaves this DR project.

  We will consider a session of 24 months. Interarrival average λ is of 1000 users/month, and average membership duration μ is 4 months. We will use a 128b long symmetric keys, and balanced binary LKH trees ($d = 2$). Storage and Communication costs of Lius *et al.* and SKM+ KMS are readily obtained from [11] and [12].

- **Simulation Results :**

  Storage Cost:

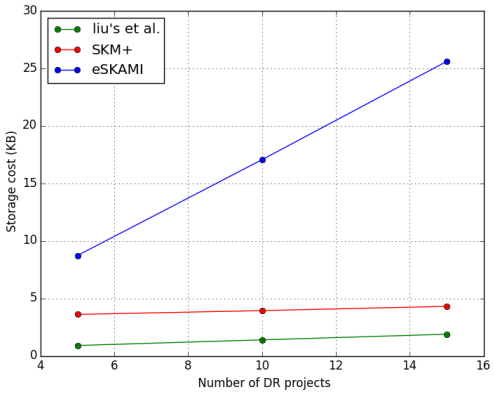
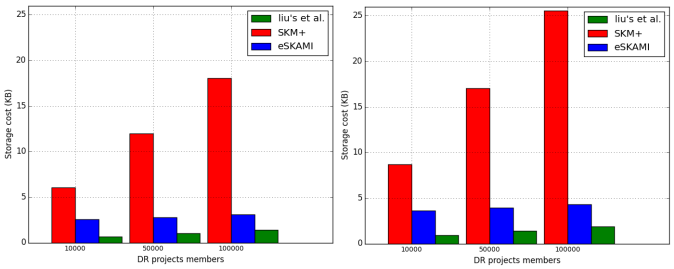


Fig. 3: Average storage cost in SMs according to number of subscribed DR projects

For MDMS, the storage cost is not a problem, we can use special key servers as storage. In constract, the storage ability of SMs is limited to 4-12 KB [23]. Fig. 3 shows a comparison of average storage cost in SMs between the three schemes according to the number of subscribed DR projects at the same time and fixing the members of DR projects members to around 100000. We can see that in our scheme, a SM stores much 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.* do not adopt a key graph technique, a SM stores one key for each subscribed DR project which represents an inefficient key management that results in non-negligible communication overhead. In SKM+, authors used a One-way Function Tree OFT for each DR project, the number of keys stored will increase significantly when a user subscribes to new DR projects. Whereas, in eSKAMI we see that the number of subscribed DR project do not affect significantly the storage cost.



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

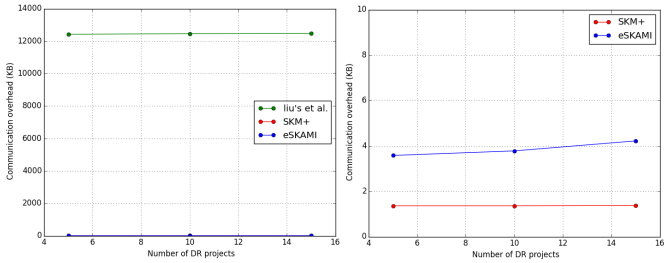
Fig. 4: Average storage cost according to DR projects members

Fig. 4 (a) and (b) shows a comparison of average storage cost in SMs between the three schemes according to the number of DR projects members. a user can subscribe respectively to 10 and 15 DR projects at the same time. In lius *et al.* scheme, the storage cost is constante, SMs stores only the group keys. Whereas in SKM+ and eSKAMI the number of DR projects members affects the storage cost, as the number of users increases, the storage cost increases due to the rise of the hight of the key trees used, but we can see that SMs store much fewer keys in eSKAMI.

Communication Cost:

Fig. 5 (a) and (b) shows a comparison of average communication cost par event (join/leave) according to the number of subscribed DR projects at the same time. We assume that there are 100000 users (on average) subscribing to each DR project. The bandwith overhead of a join is the same as a leave for the scheme of Liu *et al.* and it is a lot more than that of SKM+ and eSKAMI because of the inefficient multicast key management. Note that although SKM+ has less communication overhead than eSKAMI for join/leave event, the difference is not significant and it is too little to be seen in the Fig. 5 (a).

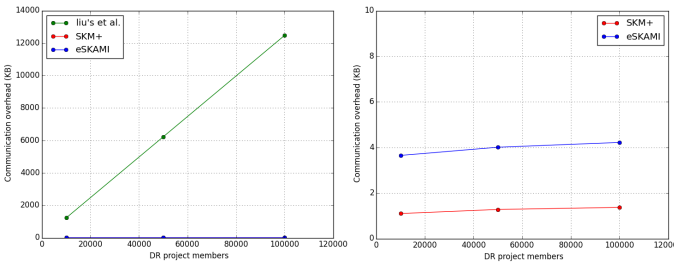
Fig. 6 (a) and (b) shows a comparison of average communication cost par event for the three schemes according to the number of subscribes in DR projects and fixing the number of



(a) (b)

Fig. 5: Average communication cost by event according to number of DR projects

DR projects to 10 DR projects. Fig. 6 (a) shows that in Liu's *et al*. scheme the bandwith overhead increases proportionally with the increase of number of subscribers in DR projects. Whereas, the bandwith overhead remains much less in SKM+ and eSKAMI as shown in Fig. 6 (b) (the bandwith overhead of SKM+ and eSKAMI is too little to be seen in Fig. 5(a)). Certainly, our scheme introduces extra communication cost compared to SKM+, but this overhead is minor regarding the overall advantages of the proposed multi-group key graph technique mainly in storage cost taking to account the storage ability of SMs which is limited to 4-12 KB [23] (reduction of storage cost comapred to SKM+ reaches 83%).



(a) (b)

Fig. 6: Average communication cost by event according to DR projects members

VI. Conclusions and Futur Works

In this paper, we proposed a new key management scheme for AMI in smart grid. It is an efficient and scalable key management scheme, capable of supporting unicast, broadcast, 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 whithout increasing the communiction overhead. In addition, the proposed KMS can achieve both forward and backward secrecy. An automatic verification of security with an automated validation tool like AVISPA is also in our perspectives.

References

[1] X. Fang, S. Misra, G. Xue, and D. Yang, "Smart Grid - The New and Improved Power Grid: A Survey," IEEE Communications Surveys & Tutorials, vol. 14, no. 4, pp. 944-980, Fourth Quarter 2012.

[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)," 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 Inter. Workshop on Smart Grid Communications and Networks, 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 Transactions on Industrial Electronics, 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. 993999, 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. 219233, 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. 444458, 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.