MK-AMI: efficient Multi-group Key management scheme for secure communications in AMI systems

Mourad Benmalek, Yacine Challal

MK-AMI: efficient Multi-group Key management scheme for secure communications in AMI systems

Mourad Benmalek
Laboratoire de Méthodes de Conception de Systèmes
Ecole nationale Supérieure d'Informatique, ESI
Algiers, Algeria
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, Algiers, Algeria
Email: y_challal@esi.dz

***Abstract*—The Smart Grid (SG) is widely considered to be the informationization of the power grid. Advanced Metering Infrastructure (AMI) has been regarded as a key component of the SG. The critical role of AMI in the SG has made this system a privileged target of cyber attacks. Consequently, AMI security is of very high importance for the security of the SG. For this reason, Key Management has been identified as one of the most challenging topics in AMI development because of the great scale of SG and dynamism of connected clients with respect to tariff programs. This paper proposes a new efficient Multi-group Key management for AMI (MK-AMI) to secure data communications in the smart grid. It is a novel key management scheme that can support unicast, multicast and broadcast communications. An analysis of security and performance, and a comparison of our scheme with recently proposed schemes illustrate that MK-AMI achieves efficient key management and induces low storage and communications overheads compared to existing solutions.**

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

I. Introduction

Smart Grid (SG), also called intelligent grid, intelligrid or futuregrid refers to the next generation power grid in which the electricity distribution and management is upgraded by incorporating advanced two-way flows of electricity and information and pervasive computing capabilities for improved control, agility, efficiency, reliability, economy 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 [2].

Advanced Metering Infrastructure (AMI) has been regarded as a key component 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 consumer's time-based data and transmitting them to the AMI host system, and it is also responsible for implementing price signals and control commands to perform necessary control actions [3]. AMI is a privileged target for security attacks with potentially great damage against infrastructures and privacy. Consequently, security is one of the most challenging topics in AMI development

Some of the key AMI security requirements are described below [3]:

- **Confidentiality**- Requirement that customer's sensitive data is accessible only to authorized systems, 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**- Requirement that the transmitted data from smart meters to the utility as well as control commands (such as Demand Response DR mechanisms that enable customers to cut down energy usage at peak times for example) are authentic, complete, and without unauthorized deletions, modifications, or additions.
- **Availability**- Requirement that data (informations and control commands) is accessible by the smart meters and the utility whenever they need it.
- **Accountability (non-repudiation)**- Requirement that the entities receiving the data will not subsequently deny receiving and vice versa.

To meet these security requirements, cryptographic countermeasures must be deployed. 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 [4]–[12], but none of them can completely satisfy the security requirements mentioned previously. Hence, we propose a new key management scheme based on an efficient and scalable multi-group key graph technique to secure unicast, multicast, and broadcast communications in a SG 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 present our multi-group key management scheme. A security and performance analysis is performed in Section IV. Finally, we draw our conclusions in Section V.

II. Related Works

In recent years, several schemes have been proposed to secure communications for AMI in smart grid. According to [12], key management has been identified as a fundamental security challenge in an AMI.

A key distribution and management scheme for large customer networks to achieve authentication, privacy and data confidentiality in AMI is proposed by Kamto *et al.* in [4].

Yan *et al.* [5] proposed an integrated approach in which trust services, integrity and data privacy could be provided by mutual authentications. In [6], 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.* [7] 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 [14]. Although the proposed key management protocol aims to reduce the computation overheads, the synchronization process still demands considerable computation efforts. Wu and Zhou [8] combines symmetric key technique based on the Needham-Schroeder authentication protocol [15] and elliptic curve public key technique [16] 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 [9], the authors showed that Wu and Zhou scheme [8] 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.

A key management scheme is proposed by Liu *et al.* [10] to secure unicast, multicast, and broadcast communications in AMI. Authors pretend that this scheme was based on the key graph management approach [17] but it 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.* [11] proposed an improvement for Liu's *et al.* scheme (called SKM) that combines an adapted identity-based cryptosystem [18] and One-way Function Trees (OFT) approach [19] 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.

Recently, Nabeel et *et al.* [12] proposed a PUF-based key management scheme for advanced metering infrastructures. Although, their scheme supports decentralized key management, the scheme requires a PUF hardware device.

III. MK-AMI: Efficient Multi-group Key Management for AMI

We introduce a new scalable and efficient key management scheme that we call efficient Multi-group Key management scheme for secure communications in AMI systems (MK-AMI). It is based on a novel multi-group key graph structure that supports the management of multiple Demand Response projects simultaneously for each customer.

A. Assumptions

1) The AMI complies with the architecture illustrated in Fig. 1 and involves Smart Meters (SMs) and Distributed Energy Resources at the user end, communication networks to connect two ends, Meter Data Management Systems (MDMS) and the means to integrate the collected data into software application systems at the utility end.

2) A specific default DR project is mandatory for all users of the SG. This default DR project will be used by MDMS to broadcast control messages or informations to all customers.

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 $\{k_1, ..., k_n\}$ between the MDMS and SMs. Theses individual keys (refreshed periodically) will be used to secure unicast communications, and to generate the multi-group key graph for secure multicast communications.
- 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 AMI. To address the scalability issue, key graph techniques can be used. Specifically, we adopt a variation of OFC (One-way Function Chain) [20]. We consider that the MDMS and all users individually compute all the keys of interior nodes using a pseudo-random function. However, the group keys are always chosen by the MDMS.

Moreover, 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. But, this naive application of OFC may be costly and induces a non-negligible key storage overhead.

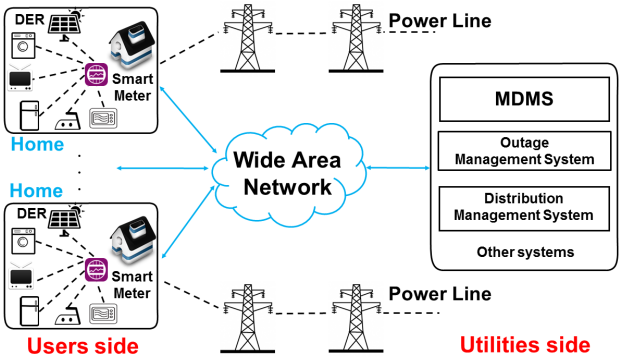


Fig. 1: Basic Components of an AMI System

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} OFC tree $h_i = log_2(m_i)$
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
$Enc(M, k)$	Message M encrypted with key k
$A \rightarrow B : M$	A sends a message M to B

To reduce storage and communication costs in key management, we propose a novel multi-group key graph structure.

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 OFC 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 combinations of root keys for users subscribing to multiple DR projects at the same time. Our key structure has the following properties:

- A user only belongs to one OFC tree in the multi-group key graph corresponding to his Home DR project. 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,
- A user has all group keys of the other DR projects to which he is subscribed,
- if a user leaves his Home DR project and remains subscribed to one or more DR projects, he will shift to a new OFC tree. These features ensure that a user will not subscribe and pay for the same project multiple times.

2) Rekeying operations: In our solution, when a user subscribes or leaves a DR project, rekeying consists of 3 operations: joining/leaving an OFC tree, shifting among trees.

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 OFC 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 OFC tree corresponding to his second subscription DR_x using standard OFC apporach (without updating key GK_x that u_i 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 $\psi_k = \{u_l / u_l \in \omega_k$ and $DR_k \neq DR_x\}$,
Let $\pi_i = \{k_l / k_l = right(k_c)$ 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: Let us consider the key graph in Fig. 2. When u_1 (subscribed to DR_1, DR_2 and DR_3) leaves DR_2 which

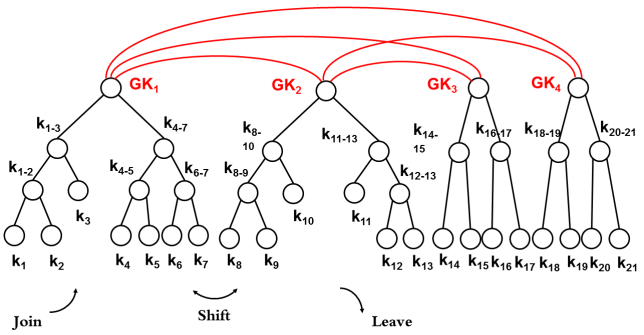


Fig. 2: Example of our multi-group key graph structure

is not his Home_DR: (a) update GK'_2 for users in $\mathcal{X}_{22}$:

$$\text{MDMS} \rightarrow \{u_8, u_9, u_{10}\} : Enc(GK'_2, k_{8-10}) \quad (1)$$

$$\text{MDMS} \rightarrow \{u_{11}, u_{12}, u_{13}\} : Enc(GK'_2, k_{11-13}) \quad (2)$$

(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) :

$$\text{MDMS} \rightarrow \{u_6, u_7\} : Enc(Enc(GK'_2, k_{6-7}), GK_2) \quad (3)$$

(c) update GK'_2 for users in ψ_k:

$$\text{MDMS} \rightarrow \psi_3 : Enc(Enc(GK'_2, k_{14-15}), GK_2) \quad (4)$$

$$\text{MDMS} \rightarrow \psi_3 : Enc(Enc(GK'_2, k_{16-17}), GK_2) \quad (5)$$

(d) update GK'_2 for users in δ_k:

$$\text{MDMS} \rightarrow \delta_4 : Enc(Enc(GK'_2, GK_4), GK_2) \quad (6)$$

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

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 OFC approach in DR_j tree whithout updating GK_j.

IV. Performance Evaluation

A. Security Analysis

1) Backward and forward Sercery: When a new user joins a DR project, he cannot learn previous group keys because he does not have access to previous group key, and hence backward secrecy is preserved. Moreover, when a user leaves a DR project, all affected keys 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 keys stored in the MDMS/SMs (Table II).

2) Communication Cost: The number of keys to be updated varies according to the position of the joining/leaving member.

a) Leave procedure

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

$$comCost = (h_j + N_{pr} - 1)|K| \quad (7)$$

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

- **Case 2:** When u_i leaves his Home DR project DR_j (DR_k is the new Home DR project):

$$comCost = (h_j + 2.A + B + h_k)|K| + c \quad (8)$$

$A = Nsub(u_i)$
$B = N_{pr} - Nsub(u_i)$
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 his Home DR project DR_l:

$$comCost = (2 + h_l + 2.A + B)|K| \quad (9)$$

b) Join procedure

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

$$comCost = h_j|K| + c \quad (10)$$

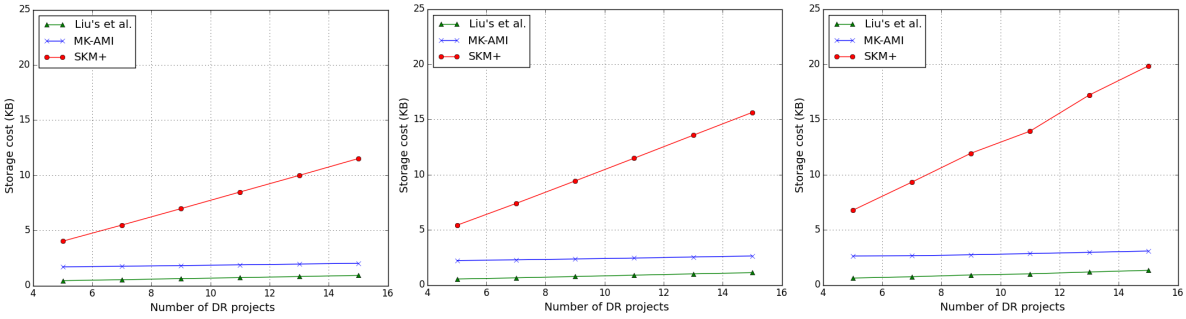
- **Case 2:** When u_i joins a new DR project DR_j which is not his Home DR project DR_l:

$$comCost = |K| + c \quad (11)$$

TABLE II: Storage Cost

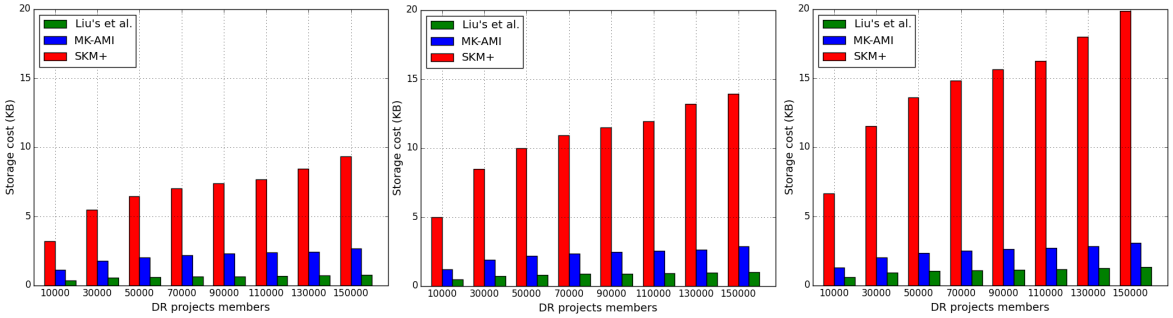
Scheme	Storage Overhead	
	MDMS	SM_i
Liu's *et al.*, 2013 [10]	$n + N_{pr} + 1$	$Nsub(u_i) + 2$
SKM+, 2014 [11]	$2\sum_{j=1}^{N_{pr}}(m_j - 1) + 1$	$\sum_{j=1}^{Nsub(u_i)}(log_2\ m_j + 1) + 1$
MK-AMI	$2\sum_{j=1}^{N_{pr}}(m_j - 1) + 1$	$log_2\ (\|Home\_DR(u_i)\|) + Nsub(u_i) + 1$

* 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.



(a) Subscribers = 30000 (b) Subscribers = 90000 (c) Subscribers = 150000

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



(a) DR projects = 7 (b) DR projects = 11 (c) DR projects = 15

Fig. 4: Average storage cost in SMs with respect to DR projects' size

3) Simulation:

- **Simulation Model:** We consider a SG with 1 million users. The utility provides 15 DR projects to users. 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 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. 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.

- **Simulation Results**

  Storage Cost

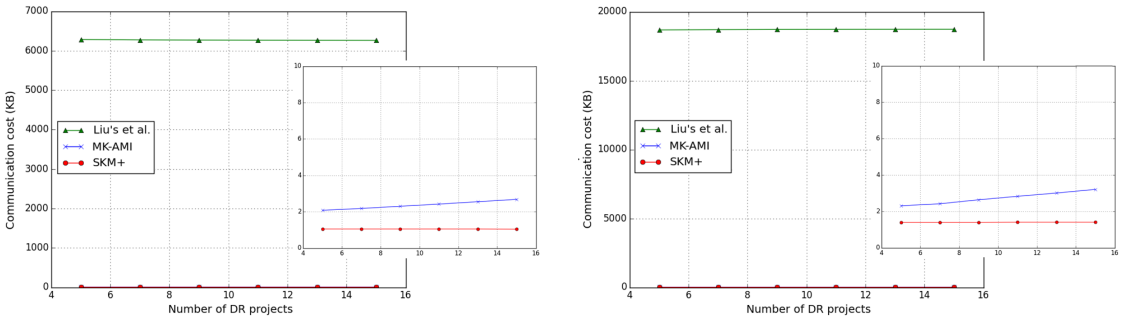
For MDMS, the storage cost can be afforded using special key servers. In contrast, the storage capacity of SMs is limited to 4-12 KB [21]. Fig. 3 (a), (b) and (c) show the average storage cost induced at SMs with respect to the number of subscribed DR projects. We can see that in our scheme, a SM stores fewer keys than that in SKM+ (reduction reaches 93% while number of subscribers is about 150000 and a user can subscribe to 15 DR projects at the same time) and little more keys than that in Liu's *et al.* scheme. This can be explained as follows: in Liu's *et al.* scheme, a SM stores one key for each subscribed DR project. In SKM+, authors used an OFT for each DR project, the number of keys stored will increase significantly when a user subscribes to new DR projects. Whereas, in MK-AMI the number of subscribed DR projects does not affect significantly the storage cost.

Fig. 4 (a), (b) and (c) show the average storage cost in SMs with respect to DR projects' size. 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 MK-AMI 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 SMs store much fewer keys in MK-AMI with respect to SKM+.

Communication Cost

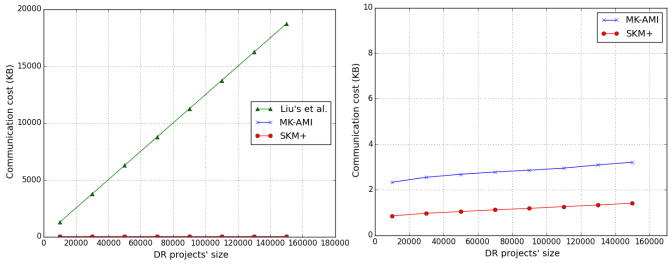
Fig. 5 (a) and (b) show a comparison of average communication cost per event (join/leave) with respect to the number of subscribed DR projects at the same time. The bandwidth overhead due for the Liu's *et al.* scheme is remarkably higher than that of our scheme because of the inefficient multicast key management. In MK-AMI, bandwith overhead reduction reaches 99% with respect to Liu's *et al.* scheme (Fig. 5 (b)). Note that although SKM+ has less communication overhead than MK-AMI, but the difference is not significant.

Fig. 6 (a) and (b) show a comparison of average communication cost per event for the three schemes with respect to the



(a) Subscribers = 50000 (b) Subscribers = 150000

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



(a) DR projects = 15 (b) Zoom of (a)

Fig. 6: Average communication cost by event with respect to DR projects' size

DR projects' size while fixing the number of DR projects to 15 DR projects. In Liu's *et al*. scheme the bandwidth overhead increases proportionally with the increase of the number of subscribers. Whereas, the overhead remains much lower in SKM+ and MK-AMI as shown in Fig. 6 (b) (the bandwidth overhead of SKM+ and MK-AMI is too little to be seen in Fig. 6 (a)). Certainly, MK-AMI introduces extra communication cost compared to SKM+, but this overhead is minor regarding the overall advantages of the proposed scheme, mainly when considering the storage cost as shown above.

V. Conclusion

In this paper, we proposed a new key management scheme for AMI in SG. MK-AMI is an efficient and scalable key management scheme supporting unicast, multicast, as well as broadcast communications. The proposed scheme uses a novel multi-group key graph technique that supports the management of multiple and dynamic Demand Response projects simultaneously for each customer. In addition, the proposed KMS guarantees both forward and backward secrecy. The detailed security analysis and performance evaluation show that MK-AMI is secure and efficient for AMI systems in smart grid.

References

[1] A. I. Sabbah, A. El-Mougy, and M. Ibnkahla, "A survey of networking challenges and routing protocols in smart grids," *IEEE Trans. Ind. Informat.*, vol. 10, no. 1, pp. 210-221, Feb. 2014.

[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] A. Anzalchi, A. Sarwat, "A survey on security assessment of metering infrastructure in Smart Grid systems," *Proceedings of the IEEE SoutheastCon 2015*, pp. 1-4, Apr. 2015.

[4] 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.

[5] 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.

[6] 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, 2011.

[7] 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.

[8] 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.

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

[10] N. Liu et *al.* "A Key Management Scheme for Secure Communications of Advanced Metering Infrastructure in Smart Grid," *IEEE Trans. Ind. Electron.*, vol. 60, no. 10, pp. 4746-4756, Oct. 2013.

[11] 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.

[12] M. Nabeel, X. Ding, S.H. Seo, and E. Bertino, "Scalable end-to-end security for advanced metering infrastructures," *Information Systems*, vol.53, pp. 213-223, Oct. 2015.

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

[14] 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.

[15] 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.

[16] 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.

[17] C. K. Wong, M. Gouda, and S. Lam, "Secure group communication using key graphs," *IEEE/ACM Trans. Netw.*, vol. 8, pp. 16-30, 2000.

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

[19] 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.

[20] R. Canetti *et al.*, "Multicast security: a taxonomy and some efficient constructions," *Proceedings of INFOCOM '99*, pp. 708-716, Mar 1999.

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