

A Lightweight Privacy-Preserving Scheme for Metering Data Collection in Smart Grid

Xiaoli Zeng*, Qin Liu[†], Hejiao Huang*, Xiaohua Jia*[§]

*Department of Computer Science, Harbin Institute of Technology Shenzhen Graduate School, China

[†]School of Computing, Wuhan University, China

[§]Department of Computer Science, City University of Hong Kong, Hong Kong

[†]Corresponding author: qinliu@whu.edu.cn

Abstract—In smart grids, smart meters are the metering devices that are responsible for collecting users' information about their electricity consumption. Smart meters provide an effective means for control center (CC) to monitor the power supply and consumption in a timely and accurate fashion. However, the collection of users' metering reports through an untrusted public network raises a serious privacy concern of users. Even though a lot of privacy-preserving approaches that have been proposed in the literature for secure data transmission, they are not directly applicable to the scenario of collecting metering reports in smart grids due to the limited computational resources of smart meters and the networking environment of smart grids. In this paper, we propose a lightweight privacy-preserving scheme for metering data collection in smart grids. There are three advantages of our proposed scheme: 1) **Lightweight**. There is no need for users' metering devices to carry out intensive encryption and decryption operations. 2) **No need of a trusted third party** for key generation and key management. 3) **Scalability**. Our scheme can cater for large scale of users connected via public networks. Experiment results show that the proposed scheme is efficiently and lightweight and is suitable for privacy-preserving data collection in smart grids.

Index Terms—Smart grid, privacy-preserving, metering data collection.

I. INTRODUCTION

With the rapid development of industrial and economic activities, the traditional power grid can no longer meet the new requirements of fine-grained and accurate monitoring of power generation and supply. A typical example of the failure of the traditional power grid is the incident of power blackout happened in the Midwest and Northeast of United States in 2003, which affected about 50 million people's lives and caused the losses of 6 billion dollars [1]. The main cause of the incident is the lack of real-time monitoring and diagnosis of the power system, which makes it difficult to distribute the electricity in a more stable and low-cost way. Meanwhile, it is also difficult for users to make better decisions on their power consumption to save electricity bills.

Recently, the introduction of smart grids attracts much attention due to many good features, such as self-healing, Hexitle power generation and storage, and more customer choices. The operation of smart grids heavily relies on accurate and timely data reports about the power consumption from consumers.

Smart meters are the metering devices that are installed in users' houses or apartments. These smart meters are connected to a Control Center (CC), which monitors the power consumption of the entire region and controls power supply and distributions. With the smart metering, a report about each user's power consumption and the quality of power supply, such as the voltage and frequency, can be transmitted to the CC in a real-time and periodical fashion, which facilitates the CC to efficiently control the power production and distribution, and to dynamically adjust incentive pricing schemes for power saving purpose. Users can also schedule the running time of their electric appliances, such as hot-water systems, electric car charging systems, and washing machines, to save their electricity bills. Although smart meters offer a number of benefits, they also pose severe security and privacy-preserving challenges. The detailed metering reports contain users' power consumption amount and patterns, which is private information that may reveal users' professions, life-style, and even social activities. There is a growing need to protect users' privacy in smart grids.

Consumer privacy related guidelines have also been actively debated in recent years [2]. In 2010, US Californias new smart meter privacy law, the first of its kind, shows strong demands for privacy protection for consumers energy consumption data [3]. Similarly, there are many privacy-preserving approaches that have been proposed in the literature in smart grids, such as the methods of data aggregation based on homomorphic encryption [4], [5], [6], transport protocols for establishing multiple sessions among smart meters and the CC [7], [8], and authentication schemes for data authentication between two entities [9], [10]. However, these methods often use a trusted third-party as the Certificate Authority and ignore the fact that the smart meters are resources-limited. In this paper, we aim at providing a privacy-preserving scheme for metering data collection when considering both the limited computational resources of the smart meters and the untrustable networking environment of smart grids.

It is not a trivial issue to achieve privacy-preserving in the collection of metering reports from users due to the following limitations of the current metering systems in smart grids: 1) The communication between users' smart meters and the CC is through untrusted public networks, which are vulnerable to malicious attacks. 2) The computing power and

memory capacity of a smart meter is very limited. Thus, it cannot run sophisticated encryption and decryption algorithms that require powerful computation resources. 3) There is no trusted third party (or authority) that can generate public keys and private keys for users and distribute keys to them. Therefore, conventional public key and private key encryption and decryption techniques cannot be directly applied to this scenario.

In this paper, we propose a lightweight privacy-preserving scheme for metering data collection in smart grids, which ensures secure data collection through the untrusted public networks in smart grids. There are three advantages of our proposed scheme: 1) Lightweight. There is no need for users' metering devices to carry out intensive encryption and decryption operations. 2) No need of a trusted third party for key generation and key management. Our proposed scheme only uses exchange of key materials between the user and the CC to ensure the secure data transmission of metering reports. 3) Scalability. Our scheme can cater for large scale of users connected via public networks. With our scheme, there is no need for CC to keep and manage keys for users. All keys are dynamically generated and disposed after use.

The rest of this paper is organized as follows. Section II discusses related work. Section III discusses the system model and our design goals. Section IV presents our scheme. Section V discusses how the proposed scheme meets our design goals, and Section VI shows the experiment results. Finally, concluding the paper in section VII.

II. RELATED WORK

Data privacy protection in smart grids is an important issue that attracts great attention from research communities, because there are all various kinds of attacks in the smart grids and not all the devices in smart grids are trusted. The existing works on protecting privacy in smart grids can be classified in three categories: 1) Privacy-preserving data aggregation schemes; 2) Authenticated communication schemes; 3) Anonymization of the metering reports.

The typical examples of the privacy-preserving aggregation schemes can be found in the literature [11], [12], [13], [14]. These schemes allow the control center (CC) to compute over the encrypted data without the decryption key. Smart meters encrypt the metering reports in a way that the CC can decrypt their aggregation with the corresponding key, but cannot decrypt the individual metering reports without the key owned by each smart meter. To some extent, these schemes might protect the single user's data from leaking out. But the CC can only receive the aggregation data, which is a limitation for it to do data analysis for a single user.

The authenticated communication schemes aim to guarantee the security of data communication between users and the CC [15], [16]. But in this way, the storage cost in the users' metering devices is high. The authors of the [15], [17] assume that every 15 minutes the smart meters collect the data and send 96 metering reports to the CC every day. Thus, smart meters need to construct the 7-level Merkle hash tree with 128

leaf nodes to authenticate the corresponding path messages. Although only 96 path messages are to be appended with the encrypted form of 96 electricity metering reports, all the 128 path messages are stored in the smart meters. The author of the [18] calculates the storage cost of a smart meter in this way is 14KB each day. It is a huge storage burden for smart meters, because the smart meters own limited storage resources, such as the 8 KB RAM, mentioned in [15].

Anonymization schemes are the methods that insert certain noise into the metering reports to prevent illegal users from interpreting the reports, but allow legal users to remove the noise and extract the correct information. The typical schemes can be found in literature [19], [20]. In these approaches, the trusted third party (TTP) is always used to act as an information relay and just passes data from smart meters to the CC. The measurements are signed and encrypted by using certificates and a PKI. TTP checks the validity of signatures from the smart meters, then removes identifying information about the smart meters. The remaining information is signed with the certificate of the TTP, and sent to the CC in encrypted form. These simple schemes achieve privacy for the users, but are not suitable for our scenario which has no trusted third party.

The methods presented above are not directly applicable to our scenario of collecting metering reports in smart grids, which has the limited computational resources of smart meters and the untrusted networking environment. In this paper, we aim at providing a lightweight privacy-preserving scheme for metering data collection in smart grids through the untrusted public networks.

III. SYSTEM MODEL

In this section, we introduce the metering data collection model. And then discuss the adversary model and design goals.

A. Metering data collection model

As shown in Fig. 1, the system consists of three components: users, the control center (CC) and a public network that connects all users to the CC. Each user is represented by a metering device that can execute some simple tasks, send metering reports to the CC and receive instructions or information from the CC. The CC is responsible for collecting metering reports from users and making decisions on power supply, distribution and pricing. Each user (so does the CC) has a unique ID. The CC knows the ID of every user, which can be done at the installation or registration time of users. The CC also has a secret key, which is kept by itself. There is no need of a trusted third party for key generation and key management in our system. Let ID_u denote the ID of user u , and ID_c and K_c denote respectively the ID and secret key of the CC.

B. Adversary model

We assume that the public network is vulnerable to malicious attacks. It is honest but curious. By honest, it will strictly follow the protocols of data transmission. By curious, it may

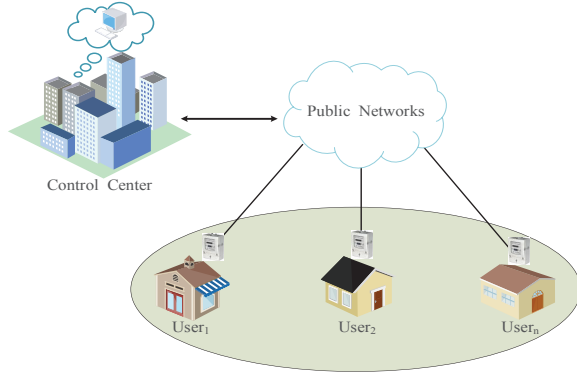


Fig. 1. Metering data collection model in smart grids.

steal metering reports for its own consumption or selling to other illegal users.

Our purpose is to protect user's privacy during the data transmission through the untrusted public networks.

C. Design goals

In this paper, we focus on guaranteeing the secure metering data collection in smart grids through the untrusted public networks. There are three goals in our design, privacy-preserving, lightweight and scalability, as defined below:

Privacy-preserving: The metering reports between the users and the CC cannot be revealed to the public networks.

Lightweight: The computing power and memory capacity of a smart meter is very limited. The encryption and decryption algorithm at the user side (i.e., running on the smart meters) should be lightweight. It cannot support complicated encryption and decryption operations.

Scalability: The data collection scheme should be scalable, because the number of users connected to the CC could be very large. It is not practical to let the CC provide key management operations for all users.

IV. THE LIGHTWEIGHT PRIVACY-PRESERVING SCHEME

For the metering data collection model in smart grids, each user has a unique ID (ID_u). The CC knows the ID of every user when the user is registered with it. All the metering reports are transmitted through the untrusted public networks. For each round of report collection, the scheme works in the following steps:

- 1) The CC sends an initial request of collecting a metering report to a user.
- 2) The user encrypts the metering report with one-time session key and sends the encrypted report to the CC periodically, until receiving further instruction from the CC.
- 3) The CC receives the metering report and decrypts it.

In the proposed scheme, we aim at providing a lightweight privacy-preserving method for metering data collection in smart grids through the untrusted public networks. We will explain each step in our scheme in detail in the following subsections.

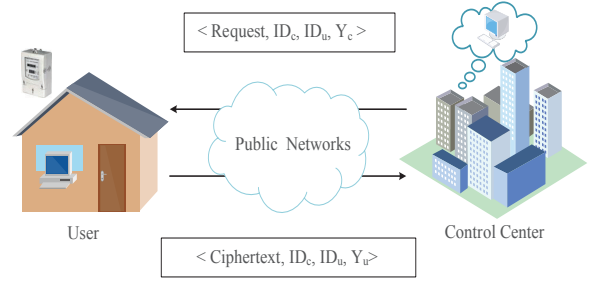


Fig. 2. The message during the metering data collection.

A. Initial request for metering report by CC

When the CC wants to collect the metering reports from a specific user, it firstly sends the initial request to the user. As shown in Fig. 2, the request ($Request, ID_c, ID_u, Y_c$) includes the request, ID of the CC and ID of the user and a key material Y_c . The CC knows the ID of every user after user's installation and registration. The CC generates a unique key material Y_c for each user, discussed in following.

In the metering data collection model, the CC has a secret key, denoted as K_c , which is kept by itself. The CC first generates a random number R_c based on its secret key K_c and the user's ID ID_u by using a one time Pseudo Random Number Generator (PRNG):

$$R_c = PRNG(K_c || ID_u). \quad (1)$$

This is with the similar concept of the Identity Based Encryption (IBE). It takes the two unique parameters K_c and ID_u as the input, and then produces an unique output R_c . Since the user's ID is different from each other, the R_c is always an unique number corresponding to the user. Therefore, the CC can generate this R_c to produce the unique decryption key for decrypting the metering reports from this user.

Finally, with this random number R_c , the CC generates the key material Y_c with the Diffie-Hellman key exchange method. This method is used to generate the key materials at both sides of the CC and the users:

$$Y_c = g^{R_c} \bmod p, \quad (2)$$

where g and p is the element of the Diffie-Hellman key exchange method.

B. Encryption of metering report by the user

When the user receives the request for metering reports from the CC, in the format of ($Request, ID_c, ID_u, Y_c$), it uses the symmetric encryption to encrypt the metering report and then sends it to the CC. It generates the encryption key Key_u to encrypt the one-time metering report into ciphertext as follows.

The user first generates a timestamp t_u , and then generates a pseudo-random number R_u from t_u and the ID_c by using the PRNG function:

$$R_u = PRNG(t_u || ID_c). \quad (3)$$

Note that t_u is a timestamp, which will be changed each time when the user sends a report.

Then it generates the encryption key Key_u :

$$Key_u = Y_c^{R_u} \bmod p, \quad (4)$$

where Y_c is the key material that CC sends to the user.

The user encrypts the metering report M by the encryption key Key_u :

$$Ciphertext = E_{Key_u}(M), \quad (5)$$

where M is the plain text of a metering report.

The user also needs to produce the key material Y_u for the CC to generate the key to decrypt the metering reports. The key material Y_u of user is generated following the Diffie-Hellman key exchange method:

$$Y_u = g^{R_u} \bmod p. \quad (6)$$

As shown in Fig. 2, the user sends the report in the format of $(Ciphertext, ID_c, ID_u, Y_u)$ to the CC. Note that the user always sends the ciphertext to the CC with the key material Y_u , which enables the CC to generate the decryption key based on it to decrypt the metering reports.

Each time when the user sends a metering report to the CC, it always generates a new random number R_u based on the timestamp t_u , then produces the one-time encryption key. This makes the data communication between the user and the CC very secure, because each time it uses a new session key for report encryption. Even a session key could be compromised at a time point, it will not affect the secure communication afterwards. This scheme guarantees the data security of the metering data collection model in smart grids.

C. Decryption of metering report by CC

When the CC receives the metering report in the format of $(Ciphertext, ID_c, ID_u, Y_u)$, it needs to generate the decryption key Key_c to decrypt the ciphertext. The generation of the decryption key is as follows, which has similar process of the encryption key Key_u .

The CC first recovers the random number R_c based on the user's ID ID_u , as in Eq. (1). Then it generates the Key_c based on the R_c and Y_u :

$$Key_c = Y_u^{R_c} \bmod p, \quad (7)$$

where Y_u is the key material sent from the user.

We can easily get Key_c is equal to Key_u :

$$\begin{aligned} Key_c &= Y_u^{R_c} \bmod p \\ &= (g^{R_u} \bmod p)^{R_c} \bmod p \\ &= g^{R_u * R_c} \bmod p. \end{aligned}$$

Recall the generation of encryption key Key_u at the user side in the last subsection:

$$\begin{aligned} Key_u &= Y_c^{R_u} \bmod p \\ &= (g^{R_c} \bmod p)^{R_u} \bmod p \\ &= g^{R_c * R_u} \bmod p. \end{aligned}$$

We can see:

$$Key_c = Key_u. \quad (8)$$

Thus, the CC can use the Key_c to decrypt the metering report and get the plain text M :

$$M = E_{Key_c}(Ciphertext). \quad (9)$$

Note that the CC generates the decryption key by the recovered number R_c and the key material Y_u from the user. It is not necessary for CC to keep and manage keys for users. All the keys are dynamically generated and disposed after use. This is scalable for the CC to support a large number of users connected via public networks.

Here gives a summary of our scheme. To collect the metering reports, the CC sends a initial request $(Request, ID_c, ID_u, Y_c)$ to a specific user. When receiving the request, user always encrypts the metering reports to ciphertext and sends the $(Ciphertext, ID_c, ID_u, Y_u)$ to the CC. The key material in our scheme is used to exchange with the other side to generate the encryption and decryption key. With the symmetric encryption, the encryption key generation is the same as the decryption key. Besides, the key varies from time to time, which makes it impossible for the public networks to decrypt the metering reports and protects the privacy of users.

V. SECURITY ANALYSIS

In this section, we will explain how our scheme achieves the design goals of the privacy-preserving, lightweight and scalability.

A. Privacy-preserving

The privacy-preserving in our scheme requires that there would be a privacy-preserving way to transfer the metering reports between the users and the CC through the untrusted public networks. In our scheme, the encryption key is generated by random number R_u and key material Y_c . It is because that the random number R_u is generated based on the timestamp t_u , the public networks cannot get the timestamp to recover R_u to produce the encryption key. Similarly, the decryption key is generated by using the random number R_c and key material Y_u . R_c is generated with the ID of users and the secret key K_c of the CC. Since the secret key K_c is confidential to the CC, public networks cannot recover the R_c to generate the decryption key to decrypt the metering reports. Thus, the privacy-preserving is achieved in our scheme.

Furthermore, the encryption key is only one-time and disposed after use. Each time before a metering report is encrypted, a new random number R_u is generated by the timestamp t_u at user side. Then the encryption key is produced. Since the random number R_u changes from time to time, the encryption key will be different for each metering report. Even once the encryption key is compromised, it will not affect the secure communication afterwards. Our scheme guarantees the security of the metering data collection model.

B. Lightweight

The lightweight in our scheme means that there is no need for users' metering devices to carry out intensive encryption and decryption operations in our scheme.

When generating the encryption key, the user uses timestamp t_u to generate the one time random number R_u . Encryption key Key_u is generated by the one-time random number R_u and key material Y_c . There is nothing previous used to generate the encryption key. It is obvious that users do not need to carry out intensive operations of encryption in our scheme and will finished every encryption efficiently. This is lightweight for the users' metering devices even though they have limited computation power and memory capacity.

C. Scalability

The scalability in our scheme means that there is no need for the CC to keep and manage keys for users, which make it cater for large scale of users in smart grids. When decryption, the R_c is recovered only based on user's ID and the secret key K_c of CC. It is no necessary for CC to generate the decryption key using the priori information. All the keys are dynamically generated and disposed after use. Thus, it is scalable for the CC to support a large number of users in our scheme.

VI. EXPERIMENT RESULT

To further understand the computational performance of our scheme, we present the detail implementation and evaluate it in this section. We get the time of encryption and decryption respectively through the simulation experiments.

A. Encryption Analysis at user side

Experiments have been conducted to study the efficiency of our proposed algorithm. The user side executes the encryption algorithm that has shown in Algorithm 1. The simulation is carried out on an ARM single chip microcomputer with 100MHZ frequency. This is a close emulation of the CPU embedded in metering devices.

Algorithm 1 Encryption algorithm for users

- 1: receive a request from the control center;
 - 2: generate the random number R_u : $R_u = PRNG(t_u || ID_c)$;
 - 3: generate the encryption key: $Key_u = Y_c^{R_u} \bmod p$;
 - 4: encrypt the metering report M : $Ciphertext = E_{Key_u}(M)$;
 - 5: send the metering report ($Ciphertext, ID_c, ID_u, Y_u$) to the CC.
-

Table I shows the computation time of the encryption of different sizes of reports with different sizes of keys. First, we focus on the encryption time of different sizes of keys in the Table I. It can be easily seen that as the sizes of keys increase, the time for encryption increases dramatically and the larger the report size is, the faster the encryption time increases when changing the sizes of keys. For instance, when the size of the metering report is 1024 bytes, the difference in computational

TABLE I
TIME OF ENCRYPTION.

Report Size	Time for encryption		
	16 bits key	32 bits key	64 bits key
64 bytes	16 ms	21ms	31ms
128 bytes	21 ms	31ms	95ms
256 bytes	31 ms	95ms	335ms
512 bytes	53 ms	175ms	655ms
1024 bytes	95 ms	655ms	1295ms

time between using the 16 bits key and 64 bits key is 1200ms while the 64 bytes report is only 15ms. Thus, we can conclude that using the 16 bits key is more efficient when the size of the metering reports is large.

Note that the encryption key is changed each time of a report transmission in our scheme. Even it is sometimes broken, it will not affect the secure communication afterwards. Thus, using the 16 bits key is sufficiently strong for our privacy-preserving goal. Therefore, we conclude that using the 16 bits key can better meet the efficiency, lightweight goals of user side in smart grids.

Then we focus on the computation time cost in different sizes of reports when using the same size of key in TABLE I. In order to better study it, we take the encryption time of the 16 bits key as an example to draw line graph in Fig. 3.

From the Fig. 3, we can see that in the proposed scheme, with increasing size of the metering reports, the total computation time increases nonlinearly. But the computation cost is constantly low. For instance, even when the size of the metering report is 1024 bytes, the cost is still a low value (only 95ms). Therefore, the proposed scheme enables the resource-limited smart meters to encrypt more efficiently.

B. Decryption Analysis at the CC side

For the decryption, we implement the proposed scheme in C Language and run it in the computer in Windows 7 OS. The CC side executes the decryption algorithm as shown in Algorithm 2. Similarly, we can see from the Table II that the time spent in decrypting the reports is little. And the difference in computational time between data sizes of 64 bytes and 1024 bytes is always less than 20ms no matter which size of keys are used. That can be concluded that our scheme scales well for typical amounts of data in the considered smart grids context.

Algorithm 2 Decryption algorithm for control center

- 1: send user a request of collecting the metering reports;
 - 2: receive a metering report from a specific user;
 - 3: generate the random number R_c : $R_c = PRNG(K_c || ID_u)$;
 - 4: generate the decryption key Key_c : $Key_c = Y_u^{R_c} \bmod p$;
 - 5: decrypt the metering report: $M = E_{Key_c}(Ciphertext)$.
-

From the aforementioned analysis, We thus conclude that: 1) The encryption key of 16 bits is sufficiently strong, and efficient for a less powerful CPU. 2) As the metering reports' size increases, time for encryption at user side increases

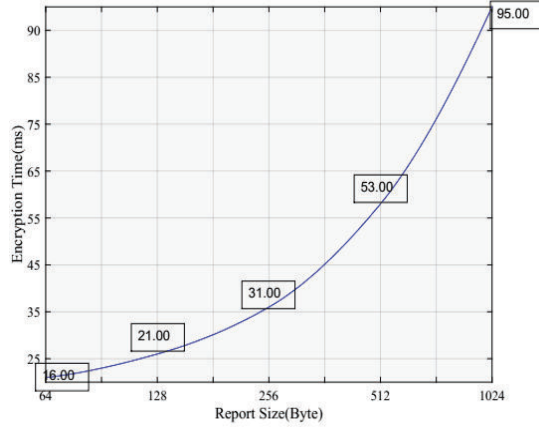


Fig. 3. Time of encryption for reports in different size using 16 bits key.

TABLE II
TIME OF DECRYPTION.

Report Size	Time for decryption		
	16 bits key	32 bits key	64 bits key
64 bytes	3 ms	6ms	10ms
128 bytes	5 ms	9ms	14ms
256 bytes	7 ms	12ms	18ms
512 bytes	10 ms	17ms	24ms
1024 bytes	16 ms	26ms	36ms

nonlinearly. 3) The decryption time in CC always remains at a low level even though the amount of data increases. Therefore, our scheme fulfills the goal of being scalable, lightweight and efficient, which is suitable for real-time high-frequency privacy-preserving data collection in the smart grids.

VII. CONCLUSION

In this paper, we propose a lightweight privacy-preserving scheme for metering data collection in smart grids, which ensures secure data collection through the untrusted public networks. There are three advantages of our proposed scheme: 1) Lightweight. There is no need for users' metering devices to carry out intensive encryption and decryption operations. 2) No need of a trusted third party for key generation and key management. 3) Scalability. Our scheme can cater for large scale of users connected via public networks. Through the experiments, we found that: 1) The encryption key of 16 bits is sufficiently strong, and efficient for a less powerful CPU. 2) As the metering reports' size increases, time for encryption at user side increase nonlinearly. 3) The decryption time in CC always remains at a low level even though the amount of data increases. Therefore, our scheme is lightweight and is suitable for privacy-preserving data collection in smart grids.

ACKNOWLEDGEMENT

This work was financially supported by National Natural Science Foundation of China with Grant (No.11371004) and (No.61672195) and (No.61572370), National Key Research and Development Program of China with Grant

(No.2016YFB0800804), and Shenzhen Science and Technology Plan with Grant (No.JCYJ20160318094336513, J-CYJ20160318094101317, KQCX20150326141251370).

REFERENCES

- [1] M. Yazdani and A. Mehrizi-Sani. Distributed control techniques in microgrids. *IEEE Transactions on Smart Grid*, 5(6):2901–2909, 2014.
- [2] Jack I Lerner and Deirdre K Mulligan. Taking the 'long view' on the fourth amendment: Stored records and the sanctity of the home. *Social Science Electronic Publishing*, (2), 2008.
- [3] Younghun Kim, C. H. Ngai, and Mani B. Srivastava. Cooperative state estimation for preserving privacy of user behaviors in smart grid. In *IEEE International Conference on Smart Grid Communications*, pages 178–183, 2011.
- [4] Nico Saputro and Kemal Akkaya. Performance evaluation of smart grid data aggregation via homomorphic encryption. In *IEEE Wireless Communications and Networking Conference*, pages 2945–2950, 2012.
- [5] Fenjun Li, Bo Luo, and Peng Liu. Secure information aggregation for smart grids using homomorphic encryption. In *IEEE International Conference on Smart Grid Communications*, pages 327–332, 2010.
- [6] Dongwon Seo, Heejo Lee, and Adrian Perrig. Secure and efficient capability-based power management in the smart grid. In *IEEE Ninth International Symposium on Parallel and Distributed Processing with Applications Workshops*, pages 119–126, 2011.
- [7] T. Khalifa, K. Naik, M. Alsabaan, and A. Nayak. Transport protocol for smart grid infrastructure. In *International Conference on Ubiquitous and Future Networks*, pages 320–325, 2010.
- [8] Young Jin Kim, V. Kolesnikov, Hongseok Kim, and M. Thottan. Sstp: a scalable and secure transport protocol for smart grid data collection. In *IEEE International Conference on Smart Grid Communications*, pages 161–166, 2011.
- [9] Mostafa M. Fouda, Zubair Md Fadlullah, Nei Kato, Rongxing Lu, and Xuemin Shen. A lightweight message authentication scheme for smart grid communications. *IEEE Transactions on Smart Grid*, 2(4):675–685, 2011.
- [10] C. Bekara, T. Luckenbach, and K. Bekara. A privacy preserving and secure authentication protocol for the advanced metering infrastructure with non-repudiation service. pages 60–68, 2012.
- [11] Rongxing Lu, Xiaohui Liang, Xu Li, Xiaodong Lin, and Xuemin Shen. Eppa: An efficient and privacy-preserving aggregation scheme for secure smart grid communications. *Parallel and Distributed Systems IEEE Transactions on*, 23(9):1621–1631, 2012.
- [12] Hongwei Li, Xiaodong Lin, Haomiao Yang, and Xiaohui Liang. Eppdr: An efficient privacy-preserving demand response scheme with adaptive key evolution in smart grid. *IEEE Transactions on Parallel and Distributed Systems*, 25(8):2053–2064, 2014.
- [13] Mohamad Badra and Sherali Zeadally. Design and performance analysis of a virtual ring architecture for smart grid privacy. *IEEE Transactions on Information Forensics and Security*, 9(2):321–329, 2014.
- [14] Zongqing Lu and Yonggang Wen. Distributed algorithm for tree-structured data aggregation service placement in smart grid. *IEEE Systems Journal*, 8(2):1–9, 2014.
- [15] Hongwei Li, Rongxing Lu, Liang Zhou, Bo Yang, and Xuemin Shen. An efficient merkle-tree-based authentication scheme for smart grid. *IEEE Systems Journal*, 8(2):655–663, 2014.
- [16] A. Hamlyn, H. Cheung, T. Mander, Lin Wang, Cungang Yang, and R. Cheung. Network security management and authentication of actions for smart grids operations. In *Electrical Power Conference, 2007. EPC 2007. IEEE Canada*, pages 31 – 36, 2007.
- [17] Ralph C. Merkle. Protocols for public key cryptosystems. In *IEEE Symposium on Security and Privacy*, page 122, 1980.
- [18] Yining Liu, Chi Cheng, Tianlong Gu, and Tao Jiang. A lightweight authenticated communication scheme for smart grid. *IEEE Sensors Journal*, 16(3):1–1, 2015.
- [19] Ronald Petric. A privacy-preserving concept for smart grids. In *Sicherheit in vernetzten Systemen: 18. DFN Workshop*, pages B1–B14, 2010.
- [20] Es, , Prashant Shenoy, Kevin Fu, Emmanuel Cecchet, and David Irwin. Private memoirs of a smart meter. 2010.