The Journal of The Institute of Internet, Broadcasting and Communication (IIBC)
Vol. 18, No. 3, pp.49-59, Jun. 30, 2018. pISSN 2289-0238, eISSN 2289-0246

https://doi.org/10.7236/JIIBC.2018.18.3.49
JIIBC 2018-3-8

스마트 미터 프라이버시 시스템을 위한 잡음 가중치 데이터 집계

Noisy Weighted Data Aggregation for Smart Meter Privacy System

김용길*, 문경일**

Yong-Gil Kim*, Kyung-Il Moon**

요 약 현재 여러 국가들은 스마트 그리드 시스템의 법적, 기술적, 비즈니스 측면에서 여러 문제점들이 발견되고 있음에도 불구하고 스마트 그리드 배치를 서두르고 있다. 스마트 그리드와 관련하여 중요한 문제는 스마트 미터기의 주된 성능을 그대로 유지하면서 미터 측정값들이 믿을 수 없는 이해집단들로부터 공격을 당하지 않도록 하는 것이다. 이러한 프라이버시 보호 문제는 하드웨어 제약사항, 보안 암호화 시스템과 보안 신호 처리와 같은 몇 가지 해결책들을 요구하고 있다. 본 연구에서는 이와 관련하여 현재 스마트 미터 프라이버시 보호 영역에서 주된 도전 문제가 되고 있는 미터 사용량 집계 암호화에 관한 하나의 접근방식을 제공한다. 개별 에너지 총 사용량에 관한 프라이버시 보호를 위해 개별 사용자 집계 함수에 잡음 가중치를 부여하는 방식을 나타낸다. 접근방식에서 준동형 암호화 성질을 충족하기 위해 잡음 가중치의 곱은 1이 된다. 단적으로 개별 에너지 사용자 집계를 알 수 없도록 하는데 있다. Diffie-Hellman 생성기를 적용하는 경우에 잡음 가중치 곱은 잡음 가중치 합으로 전환되고 가중치 합은 0이 된다. Diffie-Hellman 키 교환은 보통 512비트를 사용하기 때문에 아주 큰 키들을 사용하는 다른 Paillier 계통 암호화 방법들에 비해 보다 우수한 성능을 가진다.

Abstract Smart grid system has been deployed fast despite of legal, business and technology problems in many countries. One important problem in deploying the smart grid system is to protect private smart meter readings from the unbelievable parties while the major smart meter functions are untouched. Privacy-preserving involves some challenges such as hardware limitations, secure cryptographic schemes and secure signal processing. In this paper, we focused particularly on the smart meter reading aggregation, which is the major research field in the smart meter privacy-preserving. We suggest a noisy weighted aggregation scheme to guarantee differential privacy. The noisy weighted values are generated in such a way that their product is one and are used for making the veiled measurements. In case that a Diffie-Hellman generator is applied to obtain the noisy weighted values, the noisy values are transformed in such a way that their sum is zero. The advantage of Diffie and Hellman group is usually to use 512 bits. Thus, compared to Paillier cryptosystem series which relies on very large key sizes, a significant performance can be obtained.

Key Words : Privacy security, Smart meter, Measurement aggregation, Diffie-Hellman key exchange

*정회원, 조선이공대학교 컴퓨터보안과
**정회원, 호남대학교 공과대학 컴퓨터공학과(교신저자)
접수일자: 2018년 3월 19일, 수정완료: 2018년 4월 19일
게재확정일자: 2018년 6월 8일

Received: 19 March, 2018 / Revised: 19 April, 2018
Accepted: 8 June, 2018
**Corresponding Author: kimoon@honam.ac.kr.
department of computer engineering, Honam university. korea

Ⅰ. Introduction

Smart grid is usually divided by three parts such as power generation, transmission-distribution network, and smart meters. Power generation is not predictable because of environmental parameters. But, transmission-distribution network[11] has efficiency features in view of two way energy transmission and distribution. Smart meters provide various opportunities for new market, even though there are several challenges in measuring, analyzing, and communicating. Smart meters are designed to be read periodically in very shorter time interval units remotely. Therefore, utility providers can have various new business opportunities such as effective consumption advices and profiling that prevent power shortages and obtain load balance. In view of energy consumers, the smart metering in detail can obtain more economic energy use and choose an economic energy tariff, etc. However, smart metering has several security threats such as safety, fraud, and privacy protection[1]. Remote control can be a major target for smart meter readings, and it can give severe economic risks. In particular, the research on privacy prevention is not sufficiently addressed. The energy actions of the consumers are easily analyzed through the smart meter readings[3]. Thus, smart meter readings are very serious privacy threat for the energy consumers. Secure signal processing is very useful mechanism to prevent the untrustworthy party from the smart meter readings, and to process the tasks such as energy billing and data analysis[6]. It protects the privacy-specific data from accessing the smart meter readings, and enables the utility provider to do data analysis for the energy management. In the secure signal processing, encryption method and secure computing tools have been used. The utility company takes the encrypted data from the smart meters instead of plaint text data[5]. If the decryption key is not used, the utility company cannot access encryption data, and it guarantees the privacy protection. To do the typical smart meter operations such as energy billing, the utility company must use a suggested protocol[4]. However, before discussing for this secure signal processing, there is a special interdependency between trust and privacy. The adoption of smart meters affects the trust model according to the management styles. A centralized management gives a common trust to the grid operator. The grid operator would play the role of the collector, concentrating also the authentication and storage sides, and having access to all the detail readings. Thus, the energy consumers are concerned not only with privacy protection, but also with the validity of meter usage and tariff. It can be appeared as privacy invasion that breaks the data protection. A decentralization management involving with the collaborative computing among the meters can provide some effective solutions for actual privacy protection and tariff validity. Consequently, a certain level of decentralization management is recommended in the smart metering field, in such a way that there is user trust among other users of the same group. Of course, as the trust about the suppliers or operators, there must be trustworthy for the meter elements such as the sensors, timing devices, secure storage and secure cryptographic system. Here, very important problem is the computation and encryption of individual aggregated energy consumption. The individual aggregates of smart meter reading can be best defense regarding to the basics of privacy protection for the smart meters. Doing cryptographic operations for individual aggregated measurements is required in the smart meter system with limited computational power and memory. Though such operation types are somewhat different, but hash functions, pseudo-random number generators, symmetric and asymmetric encryptions have been generally used. In this paper, we focus on the smart meter reading aggregation, which is the major research field in the smart meter privacy-preserving. Privacy-preserving aggregation requires some assumptions such as network availability, valid certificate per smart meter

and cryptographic system under hardware environment with limited computing power. Under these assumptions, we suggest a noisy weighted aggregation scheme to guarantee differential privacy in this paper. The noisy weighted values are generated in such a way that their product is one and are used for making the veiled measurements. A Diffie-Hellman generator can be used to obtain the noisy weighted values. In this case, the noisy values are transformed in such a way that their sum is zero. In particular, the computations on Diffie-Hellman group are usually 512 bits. Compared to the Paillier cryptosystem series which relies on very large key sizes, a significant performance can be obtained[8]. In section 2, we represent smart meter privacy basics, architecture and the related stakeholder group in view of smart meter readings. Section 3 presents a noisy weighted aggregation scheme to guarantee individual user privacy. This scheme tries to preserve the individual privacy for current smart meter readings, and to comply with data protection standards. Section 4 demonstrates Java implementation on Diffie and Hellman group. Finally, we discuss some problems in privacy-preserving aggregations and conclude the paper.

II. Smart Metering Architecture and Privacy Protection

Smart meters cannot be widely adopted until technological solutions hide the meter readings and protect individual privacy. In this section, we represent smart metering network having related all stakeholders and suitable trust models.

1. Smart Metering Network

Most of the works dealing with privacy in the smart metering system suppose only two or three parties, each of them playing several simultaneous roles (see Fig.1). Consumers are end-users that receive the power supply, either households or industrial users. The individual consumption patterns and the detail readings are sensitive information that must be protected for preserving the consumers' privacy. Typically, consumers have access to the metered data to get an appropriate and advantageous tariff and can control suitably their consumption habits and electric appliances. Smart metering devices are located at the consumer side of network and detect the consumed energy at every time slot, and send the outputs to the consumer and the aggregator. One meter must be present at each consumer, so it is commonly small and cheap device with limited computational power and transmission capabilities.

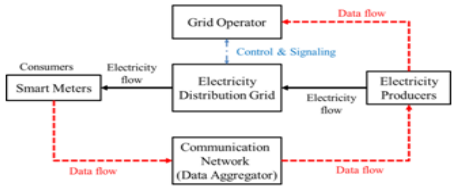

그림 1. 스마트 미터와 관계도
Fig. 1. Smart Meters and Stakeholders

Grid Operators manage the electricity distribution and transportation architecture. They evaluate electricity usage data and distribution in order to optimize the resources. In particular, load balancing is an important issue. Communication network is related to all the stakeholders in the smart metering. If individual consumption data comes from the meters, the communication channels must be secured. Data aggregator takes the meter readings and aggregates it, and produces the relevant information such as individual and average total power consumption, power demand estimation or average user profiling. It is typically done by the same provider that operates the grid. Electricity producers supply the electricity to customers through the provider's system. The price of the supplied electricity is agreed according to one or more tariffs. The provider must consider the demanded power in order to adjust the produced electricity, and also get the total individual consumptions for billing each consumer applying the contracted tariff. The

interrelations between the parties of the smart metering system are slight differences according to the implemented architecture which highly impacts the trust model. Two choices of smart metering architecture are namely centralized management and distributed one. In the centralized management, the smart meters fill the role of sensors and send the short period measurements to central data storage. The central data storage is used for consumption estimation, load balancing and billing. Users can access the stored data to get information related to their consumptions. The centralized management was the initial trend for smart metering, and all computations are done at the central aggregator that has a high computational power. In case of small grids such as the smart grids in rural areas, the meters play the role of aggregators, and all the outputs over the meter readings are distributed among the consumers that play together the role of grid operators. The meters compute a partial data aggregation and the total energy consumption in each tariff time, and send the results to the appropriate parties typically once per tariff time[9]. This management and data collection requires an absolute trust about the grid operators that play the role of the aggregator and have access to all the detail data. That is, the grid operators have access to update and remote control of the smart meters. Thus, the trustworthy agent will be required to users that are concerned with privacy protection and exactness of the meter usage and tariff computing. In the decentralized management that has possibly divided into groups and computes together with the collaborative data collection between the smart meters, some effective methods are necessarily needed to provide real privacy protection and to ensure correct tariff calculation. Therefore, some trust models are required in such a way that the trust of the users is appeared among other users of the same group. Data collection and aggregation between several users have been new challenges related with trustworthy agent and privacy protection among providers and users.

2. Privacy Preserving Metering

Fig 2 shows an interaction block between the smart meter stakeholders. The meter readings are certified by smart metering, encrypted using a local symmetric key and uploaded to the servers using a wide-area network. An energy consumer uses typically a web-browser to communicate with the electricity producer, where the meter readings are downloaded and decrypted through the key. Privacy protections like billing, fraud detection and profiling are done in the browser. Then, the results are sent back to the provider for verification and additional tasks. The consumer application can make further use of the meter readings to generate efficiency reports and a rich user interface based on the actual energy consumption of the user. The providers never know the detailed readings, but they can give a rich user experience. They compute highly reliable results on readings to support billing and other processes. Alternative user agents for computations could include smart phones, standalone software clients, as well as third party service providers trusted by the users. In all those cases, as above, certified bills or other computations can be proved correct, ensuring high integrity.

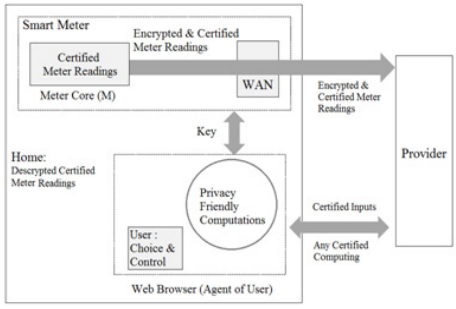

그림 2. 에이전트간의 상호 작용
Fig. 2. Interactions between agents

Currently, the smart meters send all detail energy data to the utilities or the centralized data storage with exposed meter readings. For instance, load monitoring gives the identification of specific electrical devices. It would capture the energy usage style of users such as

meal times, work times, home occupancy times, etc. To encapsulate these activities, privacy impact assessments have been introduced in standard metering process. Smart meters and related billings with privacy protection and high integrity were discussed by Rial and Danezis[6]. They introduced some protocols among the related stakeholders and a special smart meter. The smart meter displays certified reading data to the user, either directly or through a secure network. A final bill is produced through those readings according to a certified tariff scenario by the user. The final bill ensures the correct result to the grid provider exposes no additional information. More complex tariff scenarios are applied over each meter readings and freely time periods. It can be safely done to support the other tasks such as forecasting, profiling, settlement and fraud detection. In particular, the users can delegate the major calculations including their bill to the system without throwing doubt on system integrity. The purpose of the system is to support privacy protection on internet. The scenarios for protecting user privacy in smart metering should satisfy the security basics such as integrity and privacy. The utility provider must be assured that the user reports the correct results of calculations. Also, the provider does not know any information except the result of computations. For the case of billing, the provider is ensured the correct cost is calculated based on the actual readings, without knowing any detailed readings. The scenarios must be relied on cryptographic methods outside the tamperproof part of the smart meter for the integrity or specific calculations like billing. A tamperproof meter M computes the energy consumption data and the related information. A service provider makes a pricing decision, and requests the user to pay the price for total energy consumption at each billing period. The users receive the consumption readings from smart meter and pay the cost to the provider. The pricing decision uses a public function that takes in energy usage data corresponding to the time of consumption and calculates a price. The cost is computed by adding the prices corresponding to the total consumption in a billing period. This pricing decision can be used as aggregates of meter readings, which represent a tariff as a result of consumption per day or week. The provider sends the user a pricing table. The smart meter evaluates the consumption readings by using the related information during a billing period. This result is taken by the user who wants the total cost and sends it again to the provider. The user also gives notice that the cost has been correctly calculated according to the pricing table. The aggregates of smart meter reading are especially watchful in privacy preserving. Three common points are required for privacy preserving aggregation. For the communication network, while a wired communication link to the utility provider is required, smart meters are also assumed to be able to communicate with each other, which can be possible using wireless technologies. Next, a valid credential per smart meter is assumed. Credentials are the required aspect needed by the smart meter to validate the identity of the user. That is, a certification path is required. Finally, the capability of doing cryptographic operations is needed in the smart meter architecture with limited computational power and memory[7]. Though such operation types are somewhat different, but hash functions, pseudo-random number generators, symmetric and asymmetric encryptions are generally used. Zheng suggests a new cryptography technique that combines the digital signature with encryption method for authentication and confidentiality, which is based on discrete logarithm problem[10]. Zheng proposes another encryption method based on elliptic curve, which saves about half computational cost and communication cost than signature encryption method. This scheme satisfies the security basics such as confidentiality, integrity and repudiation[2].

Ⅲ. Noisy weighted Aggregation

The important measurements are total consumption

$C(t)$ and billing $B(t)$ for a certain time instant t, both needed by the electricity producer. For the measurement m_{it} of i-th smart meter index, a general summation $S(t)$ of the meter readings m_{it} can be represented as the following:

$$S(t) = \sum_{M} f(m_{it}) \quad (1)$$

Here $f(\times)$ is an identity function in the case of total consumption, or a given cost function in the case of billing (typically, a linear or piecewise linear function), and M represents the set of involved measurements, either through time slots t, through the meter index i, or through both variables. The sensitivity of these measurements needs some privacy preserving schemes in order to protect them from the grid operator, the electricity producer or the aggregator. This scheme should not hinder the aggregator from calculating $S(t)$ and also avoid some fraud possibilities. $S(t)$ can provide many functions of interest with either the grid operator or the electricity producer. Our goal is to enable the utility company to compute the total consumption without revealing the individual measurements. The measurements are mostly kept secret through some encryptions. A plain text m is encrypted with the following function:

$$H_{pk}(m) = g^m \cdot r^n \mod n^2,\ g \in Z^*_{n^2} \quad (2)$$

Here n is a product of two large primes, p and q, g is a random number with an order n, meaning that $g^n \mod n^2=1$.Thetuple(g, n) is the public key. The random number r is chosen such that $gcd(r, n) = 1$. Using a different random value for every encryption ensures that the cipher text for the same plain text will be different in each case. Note that the recipient of the cipher text does not need to know r to decrypt the message since $r^{ns} \mod n^2=1$,where$s$ is the secret key and it is $lcm(p-1, q-1)$. Therefore, any r that is prime number to n can be removed easily if it is raised first to the power of n and then s. This encryption scheme is additively homomorphism, meaning that multiplication of cipher texts of two messages results in an encryption of the sum of these two messages:

$$H_{pk}(m_1) \cdot H_{pk}(m_2) = H_{pk}(m_1 + m_2) \quad (3)$$

The additive homomorphism encryption is suitable to the aggregation of the encrypted measurements. However, the same key must be used for the aggregation by this encryption. In particular, the measurement aggregation from different smart meters does not provide privacy protection due to using the same key for encryption, and thus another technique must be considered. A privacy-preserving scheme based on secret sharing can be used[3]. The main concept of secret sharing is to divide a secret s into m pieces called shares. Each of these shares is then sent to a user in a secure scheme. A union of some users can reconstruct the secret later. A threshold secret sharing scheme which any combination of k shares out of m can be used to reconstruct the secret. This scheme is based on generating random points on a polynomial of degree k whose constant term is the secret. Clearly, when any k points are joined, the polynomial can be reconstructed, and the secret can be revealed. The secret sharing achieves the privacy goal as the utility company cannot access the private individual measurements. However, since this encryption scenario relies on secret sharing, it increases the amount of data. The complexity analysis with respect to this scheme is not reasonable since the total number of encryptions and modular operations is in either case $O(N^2)$,where$N$ is the number of smart meters. As each cipher text is in the order of thousands of bits, the communication cost is also expensive. Our scenario begins with each user splitting their measurements into random shares, one share for each user:

$$m_{it} = \sum_{j=1}^{n} m_{it}^{j} \mod B \quad (4)$$

Here B is a big integer, and m_{it}^{i} the measurement for

i-th user. The i-th user sends m_{it}^{i} to the utility company after encrypting them with public keys of the other users. The other user s also repeat the same steps with their shares. When the utility company receives encrypted shares from the users, the shares are added, which are encrypted by the same key , using the homomorphism property of the encryption scheme:

$$H_{ik}(m'_{it}) = \prod_{j \neq i} H_{ik}(m_{jt}^{i}) = H_{ik}\left(\sum_{j \neq i} m_{jt}^{i}\right) \quad (5)$$

Here ik denotes the public key of i-th user. The utility company sends H_{ik} to i-th user, who decrypt it by using own public key. To obtain the above equation in crisp text, i-th user adds own share m_{it} to m_{it}^{i} . The other user also repeats the same works, and sends it to the utility company. The utility company adds all plain texts and obtains the entire energy consumption. This simple scheme achieves perfectly the privacy basics, because the utility company cannot access the private individual data. However, the cryptographic scenario relies on secret sharing, which increases the amount of data.

Assume that $f(\times)$ is a non-linear function computed as the hash of a unique identifier such as a serial number or time and date of the measurement. Let c_i, $i = 1, \cdots, n$, be the noisy values weighted in such a way that their product is a constant and used for hindering the measurements. Under these assumptions, all users calculate $c_i f(m_{it})$, respectively. Then, the utility company can easily aggregate the measurements from all users.

$$S(t) = \sum_{i=1}^{n} c_i f(m_{it}) \mod B \quad (6)$$

However, the utility company cannot easily compute the actual sum in case that f requires solving a discrete-log problem. Assume that Diffie-Hellman key exchange is used to obtain the noisy values, and that each user has unique identification index i and a secret key K_i. To obtain the c_i values, a generator of a Diffie-Hellman group h_t is calculated by using a hash function, with t being the time period for computing the total consumption. Then, each smart meter computes the public key $exp(K_i * log(h_t))$ and distributes it to others with valid certificates. After verification of the public keys, all users compute the following expression:

$$c_i = \exp\left(\sum_{j \neq i} -I_{j<i} K_i K_j \cdot \log(h_t)\right) \quad (7)$$

Here $I_{j<i}$ is 1 if the index of meter j is smaller than the index of meter i, and zero otherwise. Clearly the product of all c_i becomes h_t.

$$\prod_{i=1}^{n} c_i = \prod_{i=1}^{n} \exp\left(\sum_{j \neq i} -I_{j<i} K_i K_j \cdot \log(h_t)\right) = h_t \quad (8)$$

Once these noise values are generated, all users can repeat with the computation of the total consumption as explained above. The time complexity of the suggested scenario is the following: The number of messages to be exchanged is $O(n^2)$,as each smart meter has to access a new Diffie-Hellman key for the aggregation in the more secure form of the scenario. The number of modular operations is $O(n)$ and the number of exponentiations is $O(n)$. Notice that the computations are on a Diffie-Hellman group, for which the key length is suggested to be 256 bits. Compared to the Paillier cryptosystem which relies on very large key sizes, the small size of the key shows significant performances.

Ⅳ. Java Implementation

In case of applying Diffie and Hellman key exchange, c_i can be reduced as $\exp(r_i \times \log(h_t))$, where the number r_i is represented in such a way that the sum of r_i is zero. Once these numbers are selected, all users can continue with the computation of the total consumption as explained before. So, we only focus on Diffie and Hellman key-exchange implementation for evaluating these numbers. The key agreement for Diffie and Hellman key-exchange can be done as the following steps. First, an agreement between user i and

user j is made on two large prime numbers, p and g. Next, the user i selects a number I, and also the user j selects a number J, where each must be less than p - 2. The user i then computes I_o and sends the result to the user j.

$$I_o = g^I \bmod p \tag{9}$$

Then, the user j computes J_o and sends the result to the user I.

$$J_o = g^J \bmod p \tag{10}$$

The user i computes the key.

$$K = \exp(I \cdot \log(I_o)) \bmod p \tag{11}$$

The user j computes the key.

$$K = \exp(J \cdot \log(J_o)) \bmod p \tag{12}$$

The only calculations that are actually seen across the network are I_o and J_o. From I_o and J_o alone, the key cannot be obtained. The user needs the p and g to do the calculations. The p and g are usually agreed upon through a secure means, but without the user ever seeing I or J, the key cannot be derived. The security providers that come with Sun's implementation of Java provide two types of asymmetric keys: DSA and RSA. JCE provides an additional type of asymmetric key: Diffie and Hellman. Each of these has their own interface that allows you to determine fundamental information about the key. For most programmers, however, keys are opaque objects, and specific features of keys are not needed. Fig 3 demonstrates a key agreement code and the result for Diffie and Hellman key-exchange. Here, we used a simple example of p with 71 and g with 41. Also, we used the random numbers with sum zero. The result shows that user i and j derived the same secret key with only opening to each other a piece of their information. Note that the simple pass of p with 71 and g with 41 would not have worked. The p and g that is passed in the algorithm must be 512 bits. Java class Big Integer supports this problem. The Diffie and Hellman algorithm is part of an engine class and has an associated service provider. Fig 4 denotes a method of letting the algorithm select the lager values of p and g. Fig 5 denotes a Diffie and Hellman key generation with larger primes, in which p and g are chosen and passed into the algorithm. We observed the keys that the Diffie and Hellman generate normally since the output in Fig 5 has the same number for the Fig 4. A key can be generated with 512 bits to 2048 bits as long as the intervals in between are multiples of 64 bits. The engine class will call a service provider that will actually implement the algorithm. In the Fig 5, the following code will pass the p and g variables into the algorithm: DHParameterSpec param = new DHParameterSpec(p, g); The p and g must be 512 bits because the algorithm is specified to use 512 bits in the following code: kpg.initialize(512).

For our noisy adding energy aggregation scheme, this Diffie and Hellman key-exchange can be used for obtaining keys without passing the key itself, but there are some flaws such as man-in-the-middle attack. The man-in-the-middle attack is possible because there is no authentication that user i or user j is actually user

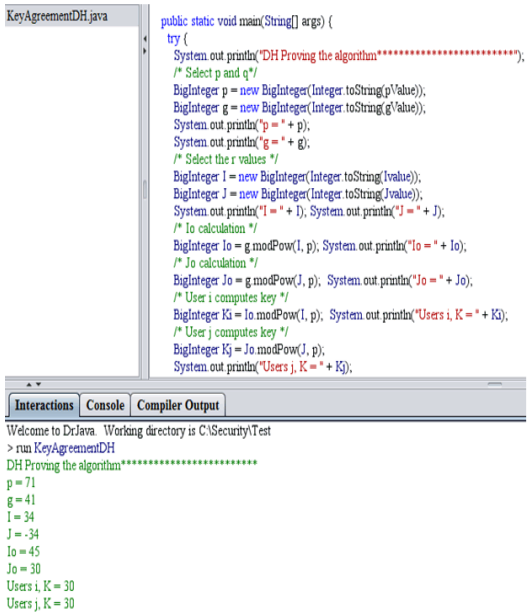

그림 3. 주요 동의 코드와 결과
Fig. 3. Key agreement code and result

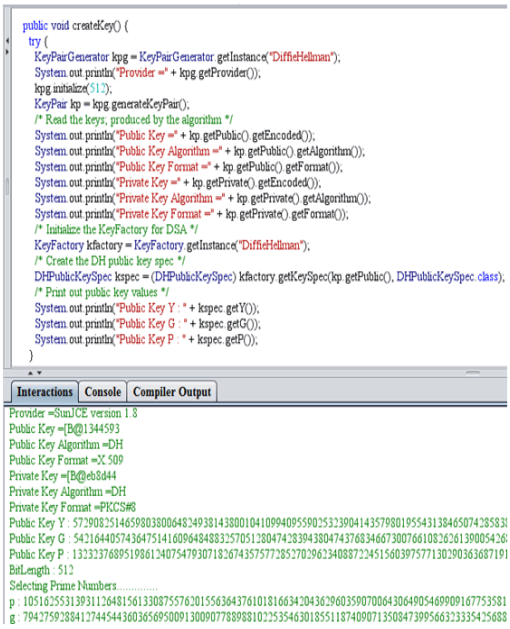

그림 4. 알고리즘으로 더 큰 값 선택
Fig. 4. Choosing the larger values by algorithm

i and user *j*. User *i* can send the shared values to the middle user. The middle user can impersonate user *j* and return the correct values, while building the shared

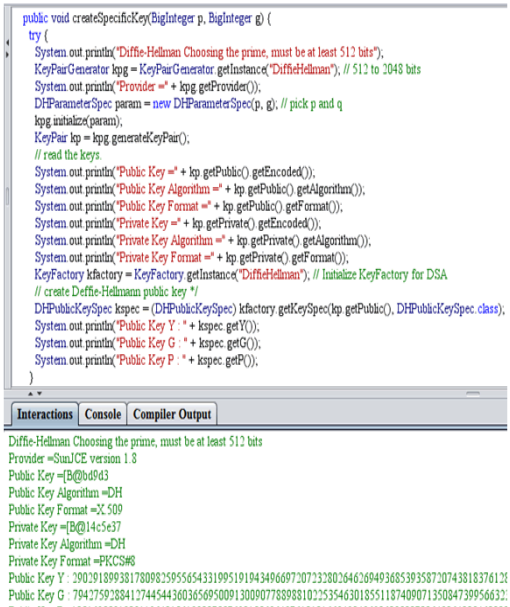

그림 5. 더 큰 값을 전달
Fig. 5. Passing larger values into algorithm

key for their use. The middle user can also be doing the same with user *j*, while impersonating user *i*. After the middle user has the correct keys from real users, the middle user can watch the messages from each in the network and continue to impersonate the other user. It might be some time and many messages later that real users discover that they have not actually communicated directly.

V. Conclusion

Now, some solutions have been demonstrated for privacy protection billing and measurement aggregation protocols based on secure signal processing which protects the smart meter readings and enables the utility company to do data analysis for the smart grid management. Its major purpose is to prevent the unbelievable stakeholders from accessing the private data, and to provide some tools to process the smart meter readings for billing and data analysis. In particular, instead of obtaining the measurements in plaint text, the utility company receives encrypted data from the smart meters. This scheme guarantees the privacy of the users. In order to do the usual task such as billing, the utility company must interact with the smart meters according to a previously described scenario. In this paper, we focused particularly on the smart meter reading aggregation, which is the major research field in the smart meter privacy-preserving. Three points are usually considered for privacy-preserving aggregation: network availability, valid certificate per smart meter and cryptographic system. The encryption types are usually hash functions, pseudo-random number generators, symmetric and asymmetric encryption. Under these situations, we suggested a noisy weighted aggregation scheme to guarantee differential privacy. The noisy weighted values are generated in such a way that their product is one and are used for making the veiled measurements. A Diffie-Hellman generator can be used

to obtain the noisy weighted values. In this case, the noisy values are transformed in such a way that their sum is zero. In particular, the computations on Diffie-Hellman group are usually 512 bits. Compared to the Paillier cryptosystem series which relies on very large key sizes, a significant performance can be obtained. However, the Diffie and Hellman key-exchange has some flaws such as man-in-the-middle attack. To prevent the man-in-the-middle attack, each smart meter must distribute the public key to others with valid certificates. Another problem is an accuracy loss for the information that providers might take as the results of these schemes.

References

[1] R. Anderson and S. Fuloria. “On the security economics of electricity metering”. In The 9th Workshop on the Economics of Information Security, 2010.

[2] C. Castelluccia, E. Mykletun, and G. Tsudik. “Efficient aggregation of encrypted data in wireless sensor networks”. *I*n Proceedings of the The Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, Washington, DC, USA, IEEE Computer Society. pp. 109 - 117, 2005.
DOI : doi>10.1109/MOBIQUITOUS.2005.25

[3] F. D. Garcia and B. Jacobs. “Privacy-friendly energy-metering via homomorphic encryption”. In J. C. et al., editor, 6th Workshop on Security and Trust Management (STM 2010), volume 6710 of Lecture Notes in Computer Science, pp. 226 - 238. Springer Verlag, 2010.
DOI : doi> 10.1007/978-3-642-22444-7_15

[4] M. Jawurek, M. Johns, and F. Kerschbaum. “Plug-in privacy for smart metering billing”. In PETS, pp. 192 - 210, 2011.

[5] M. Kohlweiss and G. Danezis. “Differentially private billing with rebates”. In Information Hiding Conference, LNCS, pp. 148-162. Springer, May 18-20 2011.
DOI : doi>I 10.1007/978-3-642-24178-9_11

[6] A. Rial and G. Danezis. “Privacy-preserving smart metering”. In Proceedings of the 10th annual ACM workshop on Privacy in the electronic society, WPES’ 11, New York, NY, USA, ACM. pp. 49 - 60, 2011.
DOI: doi>10.1145/2046556.2046564

[7] A. J. Menezes, S. A. Vanstone, and P. C. V. Oorschot. “Handbook of Applied Cryptography“. CRC Press, Inc., Boca Raton, FL, USA, 1st edition, 1996.

[8] P. Paillier. “Public-Key Cryptosystems Based on Composite Degree Residuosity Classes”. In J. Stern, editor, Advances in Cryptology - EUROCRYPT ’99, vol 1592 of LNCS, pp. 223 - 238. Springer, May 2-6, 1999.
DOI : doi> 10.1007/3-540-48910-x_16

[9] S. Peter, K. Piotrowski, and P. Langendoerfer. “On concealed data aggregation for wireless sensor networks”. In 4th IEEE Consumer Communications and Networking Conferences, 2007.
DOI: doi>10.1.1.61.5158

[10] S. Alishahi, S. M. Seyyedi, M. H. Yaghmaee and M. Alishahi, “Preserving integrity and privacy of data in smart grid communications”, CIRED Workshop - Rome, 11-12, Paper 0225, June 2014.

[11] Y.G. Kim, K.I. Moon, “Adaptive Time Delay Compensation Process in Networked Control System”, The Journal of The Institute of Internet, Broadcasting and Communication(JIIBC), Vol. 5 No. 1, pp.34-46, 2016.
DOI: doi>dx.doi.org/10.7236/IJASC2016.5.1.34

저자 소개

김 용 길(정회원)

- 1990년 호남대학교 전산통계학과 졸업(이학사)
- 1992년 광주대학교 대학원 컴퓨터학과 졸업(공학석사)
- 2014년 ~ 현재 : 조선이공대학교 컴퓨터보안과 조교수

<관심분야> : 네트워크보안, 통신시스템, 정보보호

문 경 일(정회원)

- 1982년 서울대학교 계산통계학과 졸업(이학사)
- 1980년 서울대학교 대학원 계산통계학과 졸업(이학석사)
- 1991년 서울대학교 대학원 계산통계학과 졸업(이학박사)
- 1987년 ~ 현재 : 호남대학교 컴퓨터공학과 교수

<관심분야> : 지능 시스템, 복잡성 과학