PPNC: Privacy Preserving Scheme for Random Linear Network Coding in Smart Grid

Shiming He[1,2], Weini Zeng[3], Kun Xie[4], Hongming Yang[1], Mingyong Lai[1] and Xin Su[2]
[1] School of Computer and Communication Engineering, Hunan Provincial Key Laboratory of Intelligent Processing of Big Data on Transportation, Hunan Provincial Engineering Research Center of Electric Transportation and Smart Distribution Network, Changsha University of Science and Technology, Changsha, 410114 - China
[e-mail: heshiming_hsm@163.com, yhm5218@hotmail.com, laimingyong@126.com]
[2] Hunan Provincial Key Laboratory of Network Investigational Technology, Hunan Police Academy, Changsha, 410138 – China
[e-mail: suxin@hnu.edu.cn]
[3] The 716th Research Institute, China Shipbuilding Industry Corporation, Lianyungang, 222061 – China
[e-mail: zengweini@jari.cn]
[4] Department of Electrical and Computer Engineering, State University of New York at Stony Brook, New York, 30301 – USA
[e-mail:cskxie@hnu.edu.cn]
*Corresponding author: Shiming He

Received September 19, 2016; revised December 2, 2016; revised January 9, 2017; accepted January 29, 2017; published March 31, 2017

Abstract

In smart grid, privacy implications to individuals and their families are an important issue because of the fine-grained usage data collection. Wireless communications are utilized by many utility companies to obtain information. Network coding is exploited in smart grids, to enhance network performance in terms of throughput, delay, robustness, and energy consumption. However, random linear network coding introduces a new challenge for privacy preserving due to the encoding of data and updating of coefficients in forwarder nodes. We propose a distributed privacy preserving scheme for random linear network coding in smart grid that considers the converged flows character of the smart grid and exploits a homomorphic encryption function to decrease the complexities in the forwarder node. It offers a data confidentiality privacy preserving feature, which can efficiently thwart traffic analysis. The data of the packet is encrypted and the tag of the packet is encrypted by a homomorphic

A preliminary version of this paper appeared in IEEE ICA3PP 2015, Nov. 18-20, Zhangjiajie, China. This version includes analysis and results on privacy preserving for random linear network coding. This research was supported by the National Natural Science Foundation of China (Nos. 71331001, 61303045, 71420107027, 91547113 and 61572184), the Science and Technology Projects of Hunan Province and Changsha City (Nos. 2016JC2075, 2016WK2015 and kh1601186), the Research Foundation of Education Bureau of Hunan Province, China (No. 16C0047), and the Open Research Fund of the Hunan provincial Engineering Research Center of Electric Transportation and Smart Distribution Network and the Hunan Provincial Key Laboratory of Network Investigational Technology (No. 2016WLZC016). We express our thanks to Dr. Hong Qiao who checked our manuscript.

https://doi.org/10.3837/tiis.2017.03.015 **ISSN : 1976-7277**

encryption function. The forwarder node random linearly codes the encrypted data and directly processes the cryptotext tags based on the homomorphism feature. Extensive security analysis and performance evaluations demonstrate the validity and efficiency of the proposed scheme.

Keywords: Privacy preserving, network coding, smart grid, homomorphic encryption function, random linear network coding

1. Introduction

With the introduction of information and communication technologies, the smart grid (SG) [1] allows for a two-way flow of information, automation as well as distributed intelligence over the grid. Several SG communication technologies have been proposed [2]. They can be broadly classified in three categories: power line communication (PLC), cable communication (copper or optical fiber), and wireless communication (ad hoc, mesh, and cellular architectures). Given that cable communication involves the development of a dedicated infrastructure with high capital costs, PLC and wireless communications are considered by many utility companies as the most promising alternatives [3]. Nevertheless, practical issues pertaining to these technologies are currently delaying the large-scale deployment of smart meters in distribution systems. In particular, PLC techniques may fail to connect all households (or substations) of the grid because of the strong attenuation of the communication signal [4]. Furthermore, interference is a salient issue for PLC in the distribution grids, as the spectrum is unregulated [5], [6]. With regard to wireless communication, the main challenges are related to medium characteristics of the transmission, including signal fading, noise, and path loss [7]. In fact, studies on advanced metering infrastructures have highlighted that both technologies suffer a lack of reliability because the information loss often exceeds 1% [7], [8], even after employing reliable communication methods.

Researchers have mainly focused on network coding (NC) and random linear network coding to address this bottleneck and enhance network performance in terms of throughput, delay, robustness, and energy consumption in SG [9]–[14]. The wireless communication strategy relies on accessible transmission mediums and is thus subject to security issues, including potential malicious attacks [15] and the provision of privacy guarantees [16]. In SG, privacy implications to individuals and their families are important because of the fine-grained usage data collection. For example, smart meters send their own data to a base station (BS), also called the sink, to gather data. These smart metering data may reveal highly accurate and real-time home appliance energy loads, which may be used to deduce the specific human activities occurring inside the houses. Public outcry about privacy has led to the banning of smart meters in North American cities [17]. Similarly, a planned mandatory deployment of smart meters in the Netherlands was recently derailed. Where smart meters are still deployed, users must now consent to opt in voluntarily.

Users will certainly not opt in if the privacy implications of doing so remain unclear. Several technologies have been proposed for privacy preserving: data encryption [18], [19], data distorting [19], [20] and battery-based load hiding (BLH) [20], [21].

However, random linear network coding introduces a new challenge for privacy preserving due to the encoding of data and updating of coefficients in forwarder nodes. In random linear

network coding, each coded packet contains the coefficients of the original data (named tag) and coded data (named message content). If we only apply end-to-end encryption on data and perform network coding on the encrypted data, then the coefficients will be public, and anyone will be able to linearly analyze the data and perhaps obtain the original encrypted data. If we apply end-to-end encryption on both the data and coefficients and perform network coding on the encrypted data, then the forwarder node will be unable to update the encrypted coefficients. If we apply end-to-end encryption on data and link-to-link encryption on coefficients and perform network coding on the encrypted data, then the coefficients encryption loses efficiency, because the forwarder node needs to decrypt the coefficients, update the coefficients and encrypt the coefficients again before forwarding it.

Several researches [9], [10], [22], [23] have attempted to improve privacy for NC. However, [9] and [10] propose a central scheme for single flow from one source to one destination that is unsuitable for large-scale distributed SG. [22] and [23] are designed for multicast flow, do not consider the converged flows character of SG, and are unsuitable for SG with small packets and numerous nodes. Therefore, the privacy preserving for random linear network coding in SG is still an open problem.

Therefore, to solve the problem, we propose a distributed privacy preserving scheme for random linear network coding in SG, which considers the converged flows character of the SG and exploits a homomorphic encryption function to decrease the complexity in the forwarder node. The data and coefficients of the packet are end-to-end encrypted, and the coefficients of the packet are especially encrypted by the homomorphic encryption function. The forwarder node random linearly codes the encrypted data and directly processes the cryptotext coefficients based on the homomorphism feature. The proposed scheme offers data confidentiality privacy preserving feature that can efficiently thwart traffic analysis. Extensive security analysis and performance evaluations demonstrate the validity and efficiency of the proposed scheme. We have made following contributions in the proposed scheme:

- With the employment of the homomorphic encryption function, the confidentiality of smart meter readings transmitted by the network coding is effectively guaranteed, making it difficult for attackers to recover the plain text of smart meter readings. Given that only the sink knows the decryption key, the attackers still cannot decrypt the smart meter readings even if several forwarder nodes are compromised. Moreover, the coding/mixing feature of network coding can also be naturally exploited to satisfy the requirements of privacy preservation against traffic analysis.
- Given the homomorphism of the homomorphic encryption function, the forwarder node random linearly codes the encrypted data and directly processes the cryptotext coefficients, without knowing the decryption keys or performing expensive decryption operations.
- We have conducted extensive security analysis and performance evaluations. The security analysis demonstrates that the proposed scheme can resist attacks from both inside and outside the network. The performance evaluations on computational complexity demonstrate the efficiency of the proposed scheme.
- We have compared our scheme with the other three schemes in terms of flow model, topology, attack, privacy features and computational overhead.

The rest of this paper is organized as follows. Section 2 briefly reviews the related works. Section 3 states preliminaries about network coding and Boneh–Goh–Nissim cryptosystem. The system model is introduced in Section 4. Section 5 proposes the scheme. Section 6 discusses security analysis and performance evaluations. Conclusions are drawn in Section 7.

2. Related Work

Data privacy has attracted the interest of researchers from various fields, such as in cloud computing [24]-[28] and SG [29], [30]. In SG, several technologies have been proposed for privacy preserving, such as data encryption, data distorting and battery-based load hiding. Data encryption exploits keys between the user and the sink to encrypt packet for end-to-end privacy preserving. Differential privacy is originally proposed by Dwork et al. [31]–[33] as a privacy measure for database queries. The most common way to achieve differential privacy is through data distorting to add noise to the real query result. In battery-based load hiding, a battery is installed for each household and is smartly controlled to store and supply power to the appliances for hiding actual appliance loads from the outsiders.

The existing data encryption obtains low efficiency with network coding because the forwarder nodes need to encode the data and update the coefficients before forwarding it. The privacy preserving scheme [34], [35] that we propose previously for data aggregation cannot work with network coding. Data distorting and battery-based load hiding focus on the collection process of the smart meter. We consider privacy technology in the transmission process after the collection that can work with any technology in the collection process of the smart meter.

Our proposed scheme is a special data encryption method. Even though data encryption, data distorting and battery-based load hiding work in different phase of smart meter systems, the main advantages of our proposed scheme are that it can decrease the complexity for the forwarder node in SG with random linear network coding and improve the efficiency of privacy preserving scheme with the same privacy preserving .

Several studies [9], [10], [22], [23] have been conducted privacy preserving in network coding. Hasen et al. [9], [10] develope an enhanced network coding technique for packet routing to hide the source, destination, path, traffic volume, and content information of the packets for the SG system. They introduce the concept of the sub-graph network for this purpose, using a subset of the sub-graphs to transfer the data to improve the energy consumption and system complexity. They eliminate the need to send coefficients of the network coding nodes to the receiver in the decoding process to save bandwidth. Their scheme maintaine multiple favorable privacy preserving metrics, such as anonymity, unlinkability, undetectability and unobservability for communications. Fan et al. [22], [23] propose a novel privacy preserving scheme against traffic analysis in network coding. With homomorphic encryption operation on coefficients, their schemes offer two significant privacy-preserving features, packet flow untraceability and message content confidentiality, to thwart the traffic analysis attacks efficiently. Moreover, the proposed scheme maintaine the random coding feature, and each sink could recover the source packets by inverting the coefficients with a very high probability.

However, the schemes in [9], [10] are central and the topology of SG is pre-known and static which are not suitable for large scale distributed smart grid. The schemes in [22], [23] focus on traffic analysis and are designed for multicast flow which do not consider the converged flows of SG with small packets and numerous nodes in data collection.

Therefore, the privacy preserving for random linear network coding in SGs is still an open problem that we have tried to solve in conference versions [36].

3. Preliminaries

In this section, we briefly recall the ideas of network coding [37] and Boneh–Goh–Nissim cryptosystem [38], which serve as the basis of the proposed scheme.

3.1 Network coding

The idea behind network coding is that forwarder nodes in the network can mix the packets through algebraic operations, breaking the traditional store-and-forward approach. In particular, random linear network coding (RLNC) [39], [40], [41] provides a fully distributed methodology for network coding whereby each node in the network independently and randomly selects a set of coefficients and uses them to form linear combinations of the data that it receives, as shown in **Fig. 1**.

Consider an acyclic network $G(V,E)$, where V is the node set and E is the edge set. We assume that the transmission data can be considered as a vector of the symbols and each symbol is an element of a finite field F_q. Consider a network scenario where a session comprises a set of sources $S \subseteq V$ and a sink $t \in V$. There are h nodes in source set S, where each node sends a symbol to sink t. $x_1,...,x_h$ are the original symbols to be delivered from S to t.

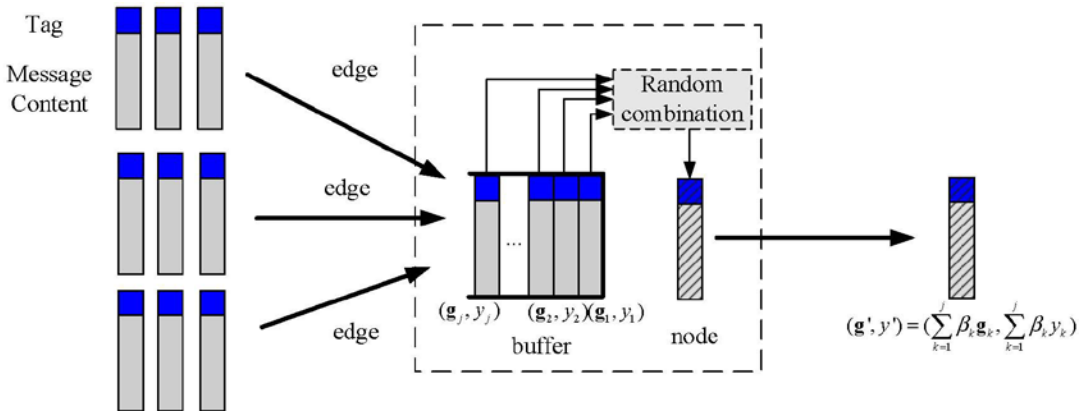

Fig. 1. Random linear network coding

The original symbol can also be considered as a coded symbol. For source s, an original symbol is $y_s = \sum_{i=1}^{h} g_{si} x_i$, where g_{ss} is 1, $g_{si}(i \neq s)$ is set to zero, and x_i s are original symbols' form.

In particular, we assume that the forwarder node v has received coded symbols of the $y_k = \sum_{i=1}^{h} g_{ki} x_i$ form. Let $y' \in F_q$ denote the outgoing symbol, which can be computed as a linear combination of the received coded symbols of node v, i.e., $y' = \sum_{k=1}^{j} \beta_k y_k$, where β_k s are random numbers. The coefficient vector $\boldsymbol{\beta} = [\beta_1,...,\beta_j]$ is called local encoding vector (LEV).

By induction, the outgoing symbol y' can be computed as a linear combination of the sources' original symbols $x_1,...,x_h$, i.e., $y' = \sum_{k=1}^{j} \beta_k y_k = \sum_{k=1}^{j} \beta_k (\sum_{i=1}^{h} g_{ki} x_i) = \sum_{i=1}^{h} (\sum_{k=1}^{j} \beta_k g_{ki}) x_i$. The coefficients form a global encoding vector (GEV) $\mathbf{g}' = [g'_1,...,g'_h]$, which can be computed recursively as

$$\mathbf{g}' = \sum_{k=1}^{j} \beta_k \mathbf{g}_k \ , \tag{1}$$

using the LEV $\boldsymbol{\beta} = [\beta_1,...,\beta_j]$, and $\mathbf{g}_k = [g_{k1},...,g_{kh}]$.

Suppose that the sink t receives symbols $y'_1,...,y'_h$, which can be expressed in terms of the source symbols as

$$\begin{bmatrix} y'_1 \\ \vdots \\ y'_h \end{bmatrix} = \begin{bmatrix} g'_{11} \cdots g'_{1h} \\ \vdots \ \ddots \ \vdots \\ g'_{h1} \cdots g'_{hh} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_h \end{bmatrix} = \begin{bmatrix} \mathbf{g}'_1 \\ \vdots \\ \mathbf{g}'_h \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_h \end{bmatrix} = G_t \begin{bmatrix} x_1 \\ \vdots \\ x_h \end{bmatrix}, \tag{2}$$

where G_t is called global encoding matrix (GEM) and the i^{th} row of G_t is the GEV associated with y'_i. The sink t can recover the h original symbols by inverting the matrix G_t and then applying the inverse to $y'_1,...,y'_h$.

In general, the data of each packet can be considered as a vector of symbols $\mathbf{y}'_t = [y'_{t,1},...,y'_{t,l}]$, where l is the length of data. By likewise grouping the source symbols into packets $x_i = [x_{i,1},...,x_{i,l}]$, the above algebraic relationships are carried over to the packets. To facilitate the decoding at the sinks, each packet should be tagged with its GEV $\mathbf{g}'$, which can be easily achieved by prefixing the i^{th} source packet $\boldsymbol{x}_i$ with the i^{th} unit vector $\boldsymbol{u}_i$. Then, each packet is automatically tagged with the corresponding GEV, as

$$[\mathbf{g}'_t, \mathbf{y}'_t] = \sum_{k=1}^{j} \beta_k [\mathbf{g}_k, \mathbf{y}_k] = \sum_{i=1}^{h} g'_{ti} [\mathbf{u}_i, \mathbf{x}_i] \tag{3}$$

Therefore, each packet contains the GEV (named tag) and coded data (named message content), as shown in Fig. 1. The benefit of tags is that the GEVs can be found within the packets themselves, thus, the sink can compute G_t without knowing the network topology or packet-forwarding paths. Actually, the network can be dynamic, with nodes and edges being added or removed in an ad hoc way. The coding arguments can be time-varying and random. More recently, tunable sparse network coding is introduced in [42], where the coding is done at different levels of sparsity, i.e., more sparse at the beginning of transmission (coding coefficients mostly zero) and denser towards the end, while keeping the transmitted coded packets innovative with high probability. This scheme reduces the delay and decoding complexity. Our work is based on the tunable sparse network.

3.2 Boneh–Goh–Nissim cryptosystem

Several cryptosystems possess the homomorphic feature: Paillier [43] and Boneh-Goh-Nissim cryptosystems [38]. The Boneh–Goh–Nissim cryptosystem is a public key encryption scheme that is proposed by Boneh, Goh and Nissim in 2005 and can provide more homomorphic features than the Paillier cryptosystem. The Boneh–Goh–Nissim system has been widely used in many privacy-preserving applications because it can achieve several nice homomorphic properties.

Given the security parameter $\tau \in Z+$, a bilinear-parameter generation algorithm $F(\tau)$ outputs a tuple (p, q, G, $G1$, e), where p and q are distinct primes with $|p|=|q|=\tau$, G and $G1$ are two cyclic groups of order $n = pq$, and e: $G \times G \rightarrow G1$ is a bilinear map.

The Boneh–Goh–Nissim encryption is comprised of three algorithms: key generation, encryption, and decryption, as follows.

1) Key generation: Given the security parameter $\tau \in Z+$, run $F(\tau)$ to obtain the tuple (p, q, G, $G1$, e) as described above. Randomly chose two generators $g, x \in G$ and set $k = x^q$. Then k is a random generator of the subgroup of G of order p. The public key is PK = (n, G, $G1$, e, g, k). The private key is $SK = p$.

2) Encryption: Given a message $m \in 0,1,...,T$ where $T << q$ is the bound of the message space, choose a random number $r \in Z_n$. Then the ciphertext can be calculated as $C = g^m \cdot k^r \in G$.

3) Decryption: Given that the private key $SK = p$ and the ciphertext $C \in G$, first compute $C^p = (g^m \cdot k^r)^p = (g^p)^m$. Let $g_p = g^p$, then $C^p = g_p^m$. To recover m, it suffices to compute the discrete logarithm of g_p^m.

Note that when m is a short message, say $m \leqslant T$ for some small bound T, the decryption takes the expected time $O(\sqrt{T})$ using the Pollards lambda method [44]. Note that decryption in this system takes polynomial time in the size of the message space T.

The Boneh–Goh–Nissim cryptosystem has several nice homomorphic properties. It is additively homomorphic. For any ciphertexts $C_1, C_2 \in G$ of messages $m_1, m_2 \in 0,1,...,T$ with random numbers $r_1, r_2 \in Z_n$, it satisfies the following homomorphic property.

$$HE(m_1) \cdot HE(m_2) = (g^{m_1} \cdot k^{r_1}) \cdot (g^{m_2} \cdot k^{r_2}) = g^{(m_1+m_2)} \cdot k^{(r_1+r_2)} = HE(m_1 + m_2) \quad (4)$$

where $HE(.)$ represents the Boneh–Goh–Nissim encryption function. Further, the following two equations can be easily derived.

$$\begin{aligned} HE(t \cdot m) &= HE^t(m) \\ HE(\sum_i t_i \cdot m_i) &= \prod_i HE^{t_i}(m) \end{aligned} \quad (5)$$

4. System Model and Motivation

In this section, we present the system model and the motivation of our work.

4.1 System model

There are different proposed definitions for the privacy. Bob Blakley defines privacy as “The ability to lie about yourself and get away with it” [45], or “The right to be left alone”. The latter definition has been adopted by NIST [46]. Pfitzmann and Hansen provided six features for the privacy [47] as follows: anonymity, unlinkability, undetectability, unobservability, pseudonymity, identity management.

As shown in **Fig. 2**, the network is modeled as a graph represented by a set of source nodes (smart meter, secondary substations and households) and a sink. The sink is considered as the BS, in charge of data gathering, coordination, and control of the source nodes. The BS is aware of the number of online source nodes, and the network topology is multi-hop. The source nodes are responsible for gathering measurements and forwarding packets to subsystems

connected to them. Each source node uploads its data to the sink at constant frequency (e.g., every 15 min), which is a data collection period. We consider a reasonable clock synchronization (a few seconds drift is acceptable) of the BS and the source nodes to support the data collection period. Therefore, the flow model in SG is converged, in which multiple sources and a sink (BS) exist.

We assume that all the nodes have the same performance capabilities in terms of processing and storage. Moreover, the links between the BS and the nodes are noisy and the signal can fade; thus, the data transmitted from the nodes to the BS and vice versa can be lost. Therefore, the BS obtains the data from the source nodes by random linear network coding. In a data collection period, the data packets from the different sources compose a round. On their way to the sink, the packets are coded with the other packets in the same round but from different sources. For example, each source node gathers measurement data and generates an original packet. The arrows represent the routing path. Node 2 sends its original packet to node 3. Nodes 1 and 4 send their original packets to node 5. After receiving the packet from node 2, node 3 generates a coded packet by Eq.(3) in sub-section 3.1 and sends the coded and its original packets to node 10. Similarly, node 5 generates two coded packets and sends the two coded and its original packets to node 10. Node 10 generates five new coded packets and sends them to the BS. The number of coded packets to generate at a forwarder node is equal to the number of received innovative packets. The processes of sub-networks of nodes 6, 7, 8, and 9 are similar.

The concept of downstream node is used, meaning a node that is closer to the sink than the local node. We intend the information to flow as waves towards the sink and compute the downstream nodes by using a secure, anonymous routing protocol [48].

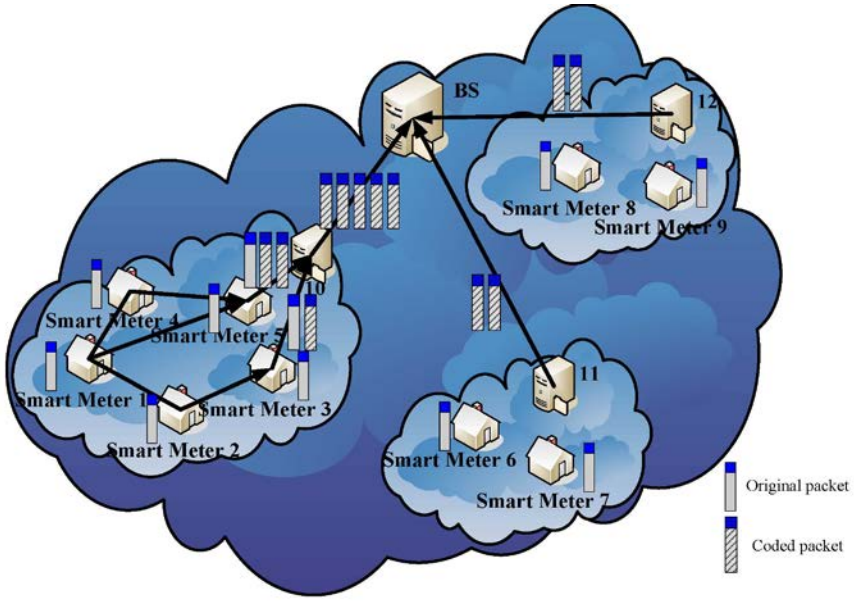

Fig. 2. Smart grid network architecture

4.2 Motivation

Given the benefit of tags, random linear network coding can provide a fully distributed methodology for network coding. However, the tag in random linear network coding

introduces a new challenge for privacy preserving because of updating of tags and encoding of data in the forwarder nodes.

Several direct privacy preserving solutions exists for random linear network coding by end-to-end encryption.

First, we only apply end-to-end encryption on data and perform network coding on the encrypted data. The tag is public. Because of the convergence flow model the tag of packets from the source nodes must be a unit vector. It is easy to obtain the original encrypted data. For the forwarder nodes, according to the tag of each packet, they or the outsider attacker can linearly analyze the data and get the original encrypted data, thereby providing backdoors for traffic analysis.

Second, we apply end-to-end encryption on both the data and the tag and perform network coding on the encrypted data. The forwarder node encodes the encrypted data to generate a new coded packet. The forwarder node needs to calculate the tag of the new packet by the LEV multiplied by the tags (GEVs) of the received packets according to Eq.(1). However, the tags are end-to-end encrypted. The forwarder node has only the encrypted tags and not the plain tags. The LEV and the encrypted tags of the received packets cannot compute the encrypted or plain tag of the new coded packet. This solution is unsuccessful because the forwarder node cannot correctly update the encrypted tag.

Third, we apply end-to-end encryption on the data and link-to-link encryption on the tag and perform network coding on the encrypted data. The encryption obtains low efficiency because the forwarder node needs to decrypt, update, and encrypt the tag again before forwarding it.

Therefore, to reduce the update complex of tag in the forwarder nodes is our main problem. The main challenge is to efficiently update the tag and maintain the secrecy of tag in the forwarder node.

5. Privacy Preserving Scheme for Random Linear Network coding

In this section, we first propose a scheme named **p**rivacy **p**reserving scheme for random linear **n**etwork **c**oding (PPNC) and then take an example to illustrate its process.

5.1 The detail of PPNC

The network coding is used between IP and TCP layer [41]. Our scheme is designed to make sure the privacy for network coding. Therefore, our scheme is also used between IP and TCP layer. PPNC exploits the homomorphism of the homomorphic encryption function to handle the update of the tags in forwarder nodes.

Three steps exist in PPNC: sending the packet, forwarding the packets, and decoding and decrypting the packets. Our scheme phases are as follows:

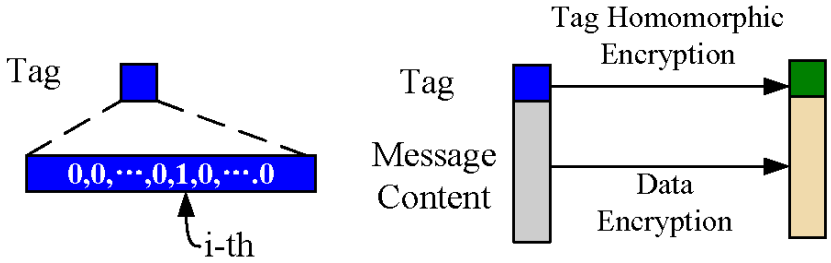

Fig. 3. Setting tag and encryption at source node *i*

Algorithm 1. Sending the packet on node *i*

Input: The public key of node BS, $\mathrm{PubK}_{\mathrm{BS}}$;
The plain packet to be sent, x_i;
The homomorphic encryption function, $HE(\cdot)$;
The encryption function, $E(\cdot)$;
Output: The sent packet with coded encrypted tag and data, $(\overline{g}_i, \overline{x}_i)$;

1. $\overline{x}_i \leftarrow E_{PubK_{BS}}(x_i)$ {Data encryption}
2. $g_i \leftarrow (0,\cdots,1,\cdots 0)$ {The i^{th} value set to one, each value in g_i represents the coefficient of one packet}
3. $\overline{g}_i \leftarrow HE_{PubK_{BS}}(g_i)$ {Tag encryption}
4. $(\overline{g}_i, \overline{x}_i) \rightarrow$ downstream node {Sending encrypted tag and data}

1) Sending the packet: Each source node per data collection period sends only one packet. Both the tag and the data must be protected. As shown in **Fig. 3**, each node encrypts the data by the end-to-end encryption function with the public key of the BS. Then each node sets the coefficients of the packet on tag. Simply, the i^{th} value is set to 1 for the packet of node *i*. In order to protect the coefficients information in the packet, we encrypt the tag by the end-to-end homomorphic encryption function with the public key of the BS. Finally, the encrypted tag and the encrypted data are sent, as shown in Algorithm 1. $HE(\cdot)$ is the homomorphic encryption function, and $E(\cdot)$ is the encryption function.

2) Forwarding the packet: The arrival of a new packet triggers the forwarder node to generate a new coded packet and send it to the next-hop node according to the routing protocol. To do so, the forwarder node creates a random linear combination of the received coded packets and forwards it to the downstream node, as shown in **Fig. 4**. In **Fig. 4**, the forwarder node can generate different coded packets from one set of received packets $(y_1,...,y_{j-1})$ by different LEV($\boldsymbol{\beta} = \left[\beta_1,...,\beta_{j-1}\right]$). Only the arrival of a new packet y_j triggers the forwarder node to generate a new coded packet, which means that the new coded packet includes the new arrived packet information, and its LEV is $\boldsymbol{\beta} = \left[\beta_1,...,\beta_j\right]$.

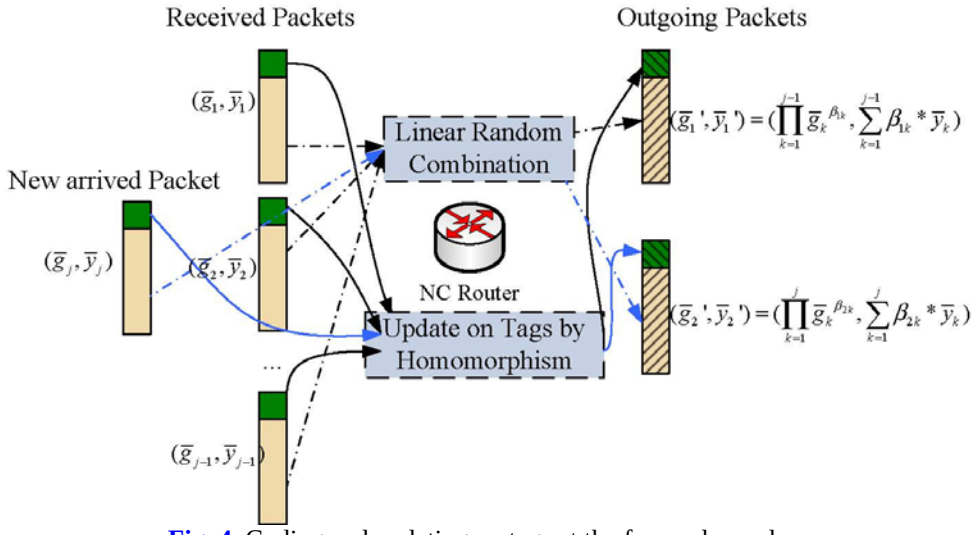

Fig. 4. Coding and updating on tags at the forwarder node

Algorithm 2. Forwarding the packets on node *i*

Input: The received packets, $(\overline{g}_1, \overline{y}_1)...(\overline{g}_j, \overline{y}_j)$;

Output: The sent packet with new coded encrypted tag and data, $(\overline{g}', \overline{y}')$;

1. for $k \to j$
2. $(\overline{g}_k, \overline{y}_k)$ {Received the k^{th} encrypted packet}
3. $\overline{y}' \leftarrow \sum_{k=1}^{j} \beta_k * \overline{y}_k$ {Encoding the encrypted data}
4. $\overline{g}' = \prod_{k=1}^{j} \overline{g}_k^{\beta_k}$ {Updating the encrypted tag}
5. $(\overline{g}', \overline{y}') \to$ downstream node{Sending new coded tag and data}

First, the forwarder node randomly chooses the LEV $\boldsymbol{\beta}$ for the received packets. Second, it linearly encodes the encrypted data of all the received packets according to the LEV to generate the data $\overline{y}'$ of a new packet. Then it generates the encrypted tag $\overline{g}'$ of the new coded packet by the encrypted tags of the received packets according to Eq.(6).

$$\overline{g}' = \prod_{k=1}^{j} \overline{g}_k^{\beta_k} \tag{6}$$

where the received packets are $(\overline{g}_1, \overline{y}_1)...(\overline{g}_j, \overline{y}_j)$, $\overline{g}_k, \overline{y}_k$ is the encrypted tag and data of the received packet *k*, respectively.

Theorem 1. The encrypted tag $\overline{g}'$ of the new coded packet can be calculated by Eq. (6) on the encrypted tags of the received packets.

Proof. According to Eqs.(1) and (5), we can get that

$$\begin{aligned} \overline{g}' &= HE_{PubK_{BS}}(g') = HE_{PubK_{BS}}(\sum_{k=1}^{j} \beta_k \cdot g_k) \\ &= \prod_{k=1}^{j} HE_{PubK_{BS}}^{\beta_k}(g_k) = \prod_{k=1}^{j} \overline{g}_k^{\beta_k} \end{aligned} \tag{7}$$

Finally, the forwarder node sends the new packet with the encoded data $\overline{y}'$ and the encrypted tag $\overline{g}'$ to the downstream node, as shown in Algorithm 2.

3) Decoding and decrypting the packets: As shown in **Fig. 5**, after the BS receives the packets, the BS utilizes its own homomorphic private key to decrypt the header to obtain tag of the packets. The BS discards non-innovative packets because they do not contain new information. After receiving the *h* innovative packets, the BS obtains the reverse value of the transfer matrix *Gt*, decodes the received packets by the transfer matrix, and obtains the encrypted original data. Finally, the BS obtains plain original data by the private key of BS, as shown in Algorithm 3.

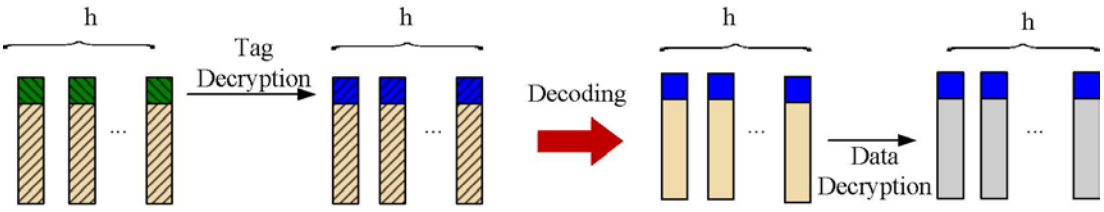

Fig. 5. Decoding and decrypting at sink node

Algorithm 3. Decoding and decrypting the packets

Input: The private key of node BS, $\mathrm{PrvK}_{\mathrm{BS}}$;
The matrix of received innovative packets with size of "$1*h$", $\overline{Y}$;
The homomorphic decryption function, $HD(\cdot)$;
The decryption function, $D(\cdot)$;
Output: The matrix of plain original packets, X;

1. $(\overline{g}_i, \overline{y}_i)$ {Received the i^{th} encrypted packet}
2. for $i \to h$ do
3. $Gt_i \leftarrow HD_{PrvK_{BS}}(g_i)$ {Tag decryption, Gt is the transfer matrix}
4. $\overline{X} \leftarrow Gt^T \times \overline{Y}$ {Decoding}
5. for $i \to h$ do
6. $x_i \leftarrow D_{PrvK_{BS}}(\overline{x}_i)$ {Data decryption}

5.2 An example

We use the left sub-network of **Fig. 2** as an example to illustrate the process of our approach (Algorithms 1 to 3). Five source nodes and one sink node are present in the **Fig. 6**. We assume that the data of the source is same as the *id* of source node, that is, source node *i*'s plain packet is ((0,..,1,..0), *i*). According to Algorithm 1, the sent packet is $(HE_{PubK_{BS}}((0,\cdots,1,\cdots 0)), E_{PubK_{BS}}(i))$. The arrows are the selected routing path. That is, nodes 1 and 4 send packets to node 5, node 2 sends packets to node 3, nodes 3 and 5 send packets to node 10, and node 10 sends packets to BS.

According to Algorithm 2, after node 3 receives the packet from node 2, node 3 stores its packet $(HE_{PubK_{BS}}((0,0,1,0,0)), E_{PubK_{BS}}(3))$ and the received packet from node 2 $(HE_{PubK_{BS}}((0,1,0,0,0)), E_{PubK_{BS}}(2))$. Then node 3 random chooses LEV $\boldsymbol{\beta} = [1,1]$ and generates a new coded packet $(HE_{PubK_{BS}}((0,1,0,0,0))HE_{PubK_{BS}}((0,0,1,0,0)), E_{PubK_{BS}}(2) + E_{PubK_{BS}}(3))$ to forward to node 10. At the same time, node 3 sends its packet to node 10.

Similarly, after node 5 receives the packet from nodes 1 and 4, node 5 stores its packet $(HE_{PubK_{BS}}((0,0,0,0,1)), E_{PubK_{BS}}(5))$ and the received packets from node 1 and 4 $(HE_{PubK_{BS}}((1,0,0,0,0)), E_{PubK_{BS}}(1))$, $(HE_{PubK_{BS}}((0,0,0,1,0)), E_{PubK_{BS}}(4))$. Then node 5 random chooses two LEVs $\boldsymbol{\beta}_1 = [1,1,1], \boldsymbol{\beta}_2 = [1,2,3]$ and generates two new coded packets $(HE_{PubK_{BS}}((1,0,0,0,0))HE_{PubK_{BS}}((0,0,0,1,0))HE_{PubK_{BS}}((0,0,0,0,1)), E_{PubK_{BS}}(1) + E_{PubK_{BS}}(4) + E_{PubK_{BS}}(5))$, $(HE_{PubK_{BS}}((1,0,0,0,0))HE^2_{PubK_{BS}}((0,0,0,1,0))HE^3_{PubK_{BS}}((0,0,0,0,1)), E_{PubK_{BS}}(1) + 2E_{PubK_{BS}}(4) + 3E_{PubK_{BS}}(5))$ to forward to node 10. At the same time, node 5 sends its packet to node 10.

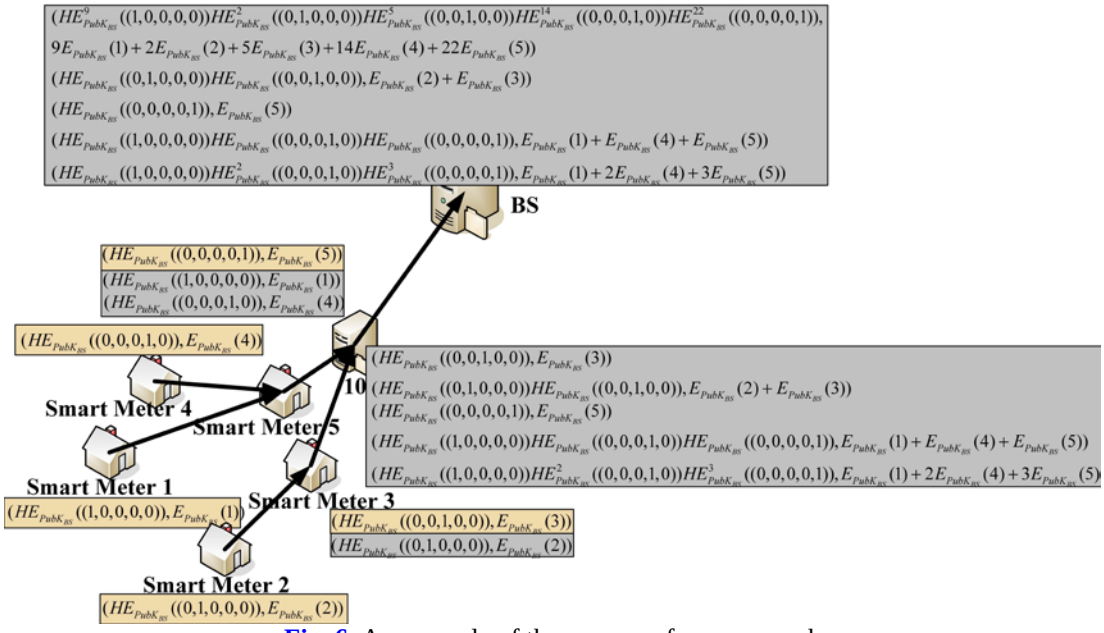

Fig. 6. An example of the process of our approach

Then node 10 receives five coded packets. According to Algorithm 2, node 10 needs to generate five new coded packets to BS. For example, node 10 randomly chooses a LEV $\boldsymbol{\beta}_1 = [1,2,3,4,5]$. The data of the first new coded packet is

$$\begin{aligned}&(E_{PubK_{BS}}(3)) + 2(E_{PubK_{BS}}(2) + E_{PubK_{BS}}(3)) + 3(E_{PubK_{BS}}(5)) + 4(E_{PubK_{BS}}(1) + E_{PubK_{BS}}(4) + E_{PubK_{BS}}(5))\\ &+5(E_{PubK_{BS}}(1) + 2E_{PubK_{BS}}(4) + 3E_{PubK_{BS}}(5)) \\ &= 9E_{PubK_{BS}}(1) + 2E_{PubK_{BS}}(2) + 5E_{PubK_{BS}}(3) + 14E_{PubK_{BS}}(4) + 22E_{PubK_{BS}}(5))\end{aligned},$$

and the tag of the first new coded packet is

$$\begin{aligned}&(HE_{PubK_{BS}}((0,0,1,0,0)))*(HE_{PubK_{BS}}((0,1,0,0,0))HE_{PubK_{BS}}((0,0,1,0,0)))^2*(HE_{PubK_{BS}}((0,0,0,0,1)))^3\\ &*(HE_{PubK_{BS}}((1,0,0,0,0))HE_{PubK_{BS}}((0,0,0,1,0))HE_{PubK_{BS}}((0,0,0,0,1)))^4\\ &*(HE_{PubK_{BS}}((1,0,0,0,0))HE^2_{PubK_{BS}}((0,0,0,1,0))HE^3_{PubK_{BS}}((0,0,0,0,1)))^5\\ &= HE^9_{PubK_{BS}}((1,0,0,0,0))HE^2_{PubK_{BS}}((0,1,0,0,0))HE^5_{PubK_{BS}}((0,0,1,0,0))HE^{14}_{PubK_{BS}}((0,0,0,1,0))HE^{22}_{PubK_{BS}}((0,0,0,0,1))\end{aligned}.$$

The LEVs of the other four coded packets are $\boldsymbol{\beta}_2 = [0,1,0,0,0]$, $\boldsymbol{\beta}_3 = [0,0,1,0,0]$, $\boldsymbol{\beta}_4 = [0,0,0,1,0]$, and $\boldsymbol{\beta}_5 = [0,0,0,0,1]$, respectively.

According to Algorithm 3, after the BS receives five innovative coded packets, the BS decrypts the header to obtain the tag of the packets. The tag of the first received coded packet is (9, 2, 5, 14, 22). The tags of other four coded packets are (0, 1, 1, 0, 0), (0, 0, 0, 0, 1), (1, 0, 0, 1, 1), and (1, 0, 0, 2, 3), respectively. The transfer matrix *Gt* is

$$\begin{bmatrix} 9 & 2 & 5 & 14 & 22 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 2 & 3 \end{bmatrix}.$$

The BS performs Gaussian elimination to recover the encrypted original data $E_{PubK_{BS}}(1)$, $E_{PubK_{BS}}(2)$, $E_{PubK_{BS}}(3)$, $E_{PubK_{BS}}(4)$, and $E_{PubK_{BS}}(5)$. Finally, the BS decrypts the encrypted data to obtain the original data.

6. Security analyses

In this section, in order to demonstrate the properties of the proposed scheme, we present an analysis from the privacy point of view and estimate the computational and communication overhead.

6.1 Privacy performance analysis

The attackers may launch various security attacks against the SG networks. However, these attacks have no one-for-all solution. Therefore, this paper separately investigates such attacks and assumes that the attackers aim to obtain private data. We refer to the Dolev–Yao model [49] to design our two adversary models, which include outside and inside attackers as shown in **Fig. 7**, in the case of the SG system. We also discuss the replay attack and man-in-the-middle attack. For the six feature for privacy defined in section 4.1, in our paper, we focus on the four former features.

1) Outside attacker: An outside attacker is an external party and is not an entity of the system. An outside attacker can be considered as a global passive eavesdropper who has the ability to observe all network links. The attacker receives all of the packets and examines the tags and data entering into a node (smart meter) and departing from the node. Furthermore, even if messages are encrypted in an end-to-end manner, it is still possible for a global outside attacker to trace packets by analyzing and comparing the ciphertext message. For instance, the attacker knows the public keys of the entire parties and has detailed knowledge about network topology. Moreover, the attacker knows the detail design of our proposed privacy mechanism shown by Algorithms 1-3. The goal of the outside attacker is to obtain information about the original packet.

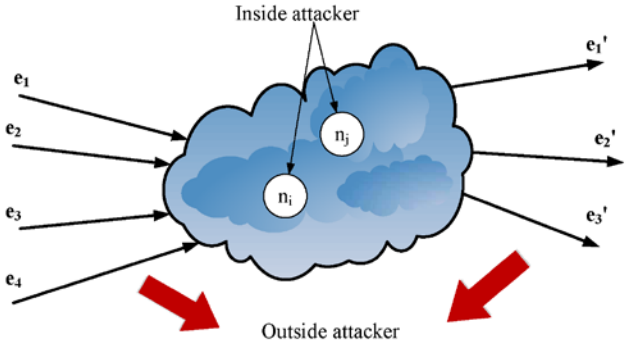

Fig. 7. Attack model: outside and inside attackers.

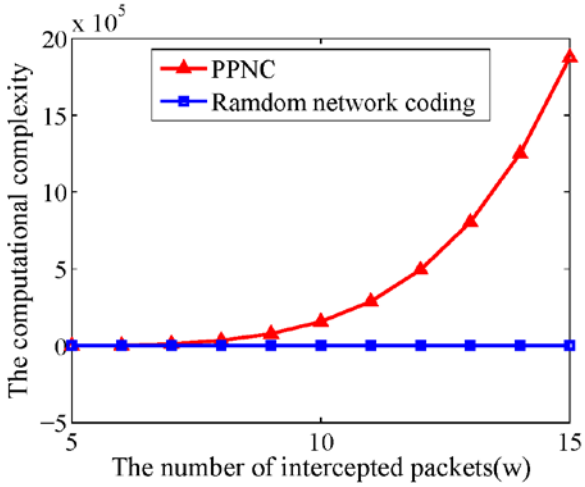

Fig. 8. Privacy enhancement in terms of the computational complexity

Discussion: For a global outside attacker, it is still possible to trace the packets by analyzing and comparing the ciphertext message, even if messages are encrypted in an end-to-end manner and the encrypted message remains the same during its forwarding. The mixing feature of network coding can resist the data correlation used in traffic analysis (undetectability). In our scheme, the encrypted message is changed after getting through every node; thus, it is hard to trace the path or to find the source of a message (unlinkability, anonymity, which yields to unobservability). With the assistance of the homomorphic encryption function, the tags are kept confidential to eavesdroppers, making it difficult for attackers to perform linear analysis on tags. In addition, homomorphic encryption function keeps the random coding feature, making linear analysis on data almost computationally impossible. Let the number of intercepted packets be *w*. The computational complexity for attackers to examine if a packet is a linear combination of *h* messages is $O(h^3 + hl)$ in terms of multiplication, where *l* is the length of the data. Thus, the computational complexity to analyze the intercepted *w* packets is $O(C_w^h(h^3 + hl))$, which increases exponentially with *w*, as shown in **Fig. 8**, where $h = 5$ and $l = 100$.

2) Inside attacker: The inside attacker is an internal party and may compromise several forwarder nodes. The malicious node is already authenticated and receives the system parameters and its own private key; thus, the inside attacker possess these information. The malicious node is under control of the attacker and performs Algorithm 2. Link-to-link encryption is vulnerable to an inside attacker because they may already obtain the decryption keys and reveal plain text message. The goal of the inside attacker is to gain access to the information of the neighbor nodes by receiving their data for relay.

Discussion: Having access to a malicious node only improves the attacker situation on modifying its data. The forwarder nodes only mix the packets and do not perform any encryption and decryption. Consequently, his behave is almost the same as the previous scenario.

3) Replay attack: A valid data transmission is maliciously repeated or delayed. This is carried out either by the sources or by an adversary who intercepts the data and re-transmits it, possibly as part of a masquerade attack by the IP packet substitution.

Discussion: When a packet is repeated, the forwarder node generates more than one new coded packet. The BS may receive more than *h* packets, but includes non-innovative packets.

The BS discards non-innovative packets because they do not contain new information. Therefore, after the BS receives h innovative packets, it can obtain the information by Algorithm 3. When a packet is delayed, the BS needs to wait the delay for receiving h innovative packets.

4) Man-in-the-middle attack (MITM): The attacker secretly relays and possibly alters the communication between two parties who believe they are directly communicating with each other.

Discussion: In our scheme, we use the end-to-end encryption for the tag and the data of packet. All cryptographic systems that are secure against MITM attack require mutual authentication. How to authenticate is beyond the scope of our paper. To fight MITM attack, we can authenticate before the key distribution of the tag homomorphic encryption and data encryption.

6.2 Computational Overhead

The computational overhead of the proposed scheme can be investigated from three aspects, including source node, forwarder node and sink node. Since the computational complexity of the proposed scheme is closely involved with the specific homomorphic encryption algorithm in the following analysis, we will take the Boneh–Goh–Nissim cryptosystem as the encryption method when necessary. Our scheme can work with any encryption algorithms of the data encryption, such as AES [50]. Therefore, in the overhead analysis we ignore the encryption and decryption of the data and focus on tag homomorphic encryption, decryption, and coding computation of the data and the tag.

1) Source node overhead: For transmitting the smart meter reading, the source node i needs one encryption operation. According to the Boneh–Goh–Nissim cryptosystem, every encryption operation requires 2 exponentiations and 1 multiplication operation. Therefore, the computational complexity is $O(\log n)$ in terms of the multiplication operations.

2) Forwarder node overhead: In forwarder nodes, linear transformation on the elements of GEVs can only be performed by manipulating the ciphertext of these elements because forwarder nodes have no knowledge of the decryption keys. According to Eq.(6), the computational complexity of producing one element in new GEVs is h exponentiations and $h - 1$ multiplications on the ciphertext, which is $O(h \log n)$ in terms of multiplications together. Thus, the computational complexity is $O(h^2 \log n)$ for a GEV and $O(h^3 \log n)$ for a GEM with h GEVs in terms of multiplication operations.

3) Sink node overhead: After receiving a packet, the sink can decrypt the elements of the GEV in the tag. According to the Boneh–Goh–Nissim cryptosystem, decrypting an element takes an expected time $O(\sqrt{T})$ using the Pollards lambda method. Therefore, the computational complexity of decrypting a GEV is $O(h\sqrt{T})$ in terms of the multiplication operations.

After receiving h packets, which requires $O(h^2\sqrt{T})$ multiplication operations to decrypt the h GEVs, the sink node can start to check the linear dependence of the GEVs. A method is the Gaussian elimination algorithm, which requires $O(h^3)$ multiplication operations.

If h GEVs are linearly independent or innovative, we can further derive the inverse of the corresponding GEM. Based on Gaussian elimination, the computational complexity to find the inverse of a matrix is $O(h^3)$ in terms of multiplication operations. With the inverse of a matrix, the sink can recover the encrypted original data by decoding the encoded data. The

computational complexity for the recovery is $O(h^2 l)$ in terms of multiplication, where l is the length of the data.

In summary, the computational complexity for the sink to decode h messages is $O(h^2(\sqrt{T}+h+l))$ in terms of multiplication operations, as shown in **Fig. 9** where l = 100 and T = 200.

However, as the number of source nodes increases, the length of the tag increases because of random linear network coding. The tag may thus be longer than the data, which reduces transmission efficiency.

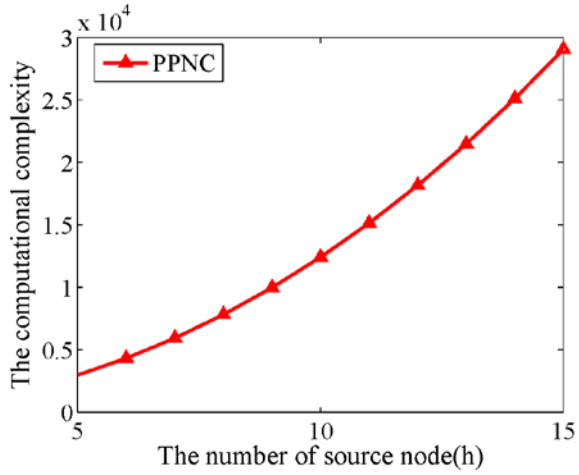

Fig. 9. Computational complexity for the sink to decode messages

6.3 Communication overhead

Let h packets be generated, and the length of data is l bits. For source encoding, each packet is prefixed with h code words from a group of size n. Considering the ciphertext expansion of the Boneh–Goh–Nissim cryptosystem, we can calculate the communication overhead as $2h\log n / l$.

6.4 Comparisons

As a retrospect, in the following we compare our scheme with the other three schemes: basic random linear network coding, the privacy preserving scheme against traffic analysis in network coding proposed by Fan et al. [22], [23], and the privacy-preserving approach for the SG system proposed by Hasen et al. [9], [10].

The result of the comparison is shown in **Table 1.** First, the four schemes are suit for the different flow models and topologies. In single flow model, there is only one source and one destination. In multicast flow model, there are a source and multiple destinations. There are multiple sources and a destination in converged flow model. Network coding can work with three flow model, [22], [23] is designed for multicast flow, [9], [10] is designed for single flow model, and our scheme is for converged flow model. Second, we claim that all the four schemes can prevent inside attack, and that only network coding cannot prevent outside attack. In the Dolev-Yao model [46], the goal of outside and inside attackers is to obtain information about the original packet. The inside attackers only modify its data, and do not pollute other forwarding packets. Therefore, in network coding, although the GEV is public, the inside attackers cannot get the information of the coded packets without h innovative packets. However, for the outsider attackers, it may receive all of the packets and examines the tags and

the data entering to the BS. It can receive h innovative packets and decode the packets to gain the information. Therefore, in the Dolev-Yao model, network coding cannot prevent outside attacker, but can prevent inside attacker. Finally, we compare privacy features. We consider two types of the attackers such as a neighbor and a forwarder node [9], [10]. Several of the schemes may deliver the anonymity in case of forwarder nodes; however, the data is not anonymous for a neighbor. We also use the following symbols to describe each deliverable:
"√": Delivers the feature against all nodes.
"○": Delivers the feature only against forwarder nodes.
"×": Does not deliver the feature.

Table 1. Comparison among our scheme and the other three schemes

Class	Network coding	[22], [23]	[9], [10]	Our scheme
Flow model	Multicast, Single flow, Converged	Multicast	Single flow	Converged
Topology	Dynamic	Dynamic	Static	Dynamic
Preventing inside attack	√	√	√	√
Preventing outside attack	×	√	√	√
Anonymity	○	√	√	√
Unlinkability	○	○	√	√
Undetectability	○	○	√	√
Unobservability	×	×	√	√

Table 2. Computational overhead of our scheme and the other three schemes

Class	Network coding	[22], [23]	[9], [10]	Our scheme
Setup	/	/	$O(m)$	/
Source	$O(h^2)$	$O(h^2 \log n)$	$O(\log n + m)$	$O(\log n)$
Forwarder	$O(h^2)$	$O(h^2 \log n)$	$O(h^2)$	$O(h^2 \log n)$
Sink	$O(h^3 + h^2 l)$	$O(h^2 \log n + h^3 + h^2 l)$	$O(h^2 \log n + h^3 + h^2 l + m)$	$O(h^2 \sqrt{T} + h^3 + h^2 l)$

We compare their computational overhead in the setup phase, source, forwarder node, and sink as shown in **Table 2**. m is the number of sub-graphing in [9], [10], h is the number of original packets, n is the product of p and q which are distinct primes in cryptosystem, and T is the bound of the coefficient space, l is the length of the data. In [9] and [10], a setup phase is needed to divide sub-graph and assign the transfer matrix of sub-graph. Therefore, there is $O(m)$ overhead in the setup phase. For fairness, only the process on tags is considered, and the encryption and decryption on data in [22], [23] and our scheme are ignored. From last two schemes we can see that, the computational overhead of our scheme is lower than that of the schemes of [9], [10] on setup phase and sink. Then, it is fair to conclude that our scheme outperforms the other three candidates in terms of thwarting outside attacks in SG.

6.5 Simulation

We implement our scheme and the Advanced Encryption Standard (AES) [50] for performance comparisons. AES is a symmetric block cipher that can encrypt and decrypt

information. It is a standardized encryption algorithm and has become the default choice in numerous applications. Different from our scheme, we implement the AES encryption algorithm in the network where each forwarder node needs to make decryption and encryption operation upon a packet's tag. In this comparison, we implement our scheme in the network for data gathering.

We take the end-to-end delay as the performance metric to evaluate performance. The above two algorithms are evaluated through extensive simulations using NS-2. In the simulations, 100 nodes are generated uniformly in a 600 m × 600 m area. The sink node is located in the center of the area. The maximum communication range of each node is set to 70 meters. All the simulation results are obtained by averaging over 20 runs of simulations. To evaluate the performance of the proposed scheme, we set the simulated data size to 128 bits. All the nodes are set as source nodes and the length of tag is set to 400 bits.

Fig. 10 shows the simulation results. The end-to-end delay of our scheme is correlated with the largest hop number from all source nodes to the sink. As expected, the end-to-end delay increases with the increase of the number of hops, as more node participation in data relays would invoke more homomorphic operations. The computational time to solve Eq.(2) with 100 source nodes, i.e. *Gt* with 100 rows and 100 columns is 0.124s with Intel(R) Core(i3) CPU and 4G RAM.

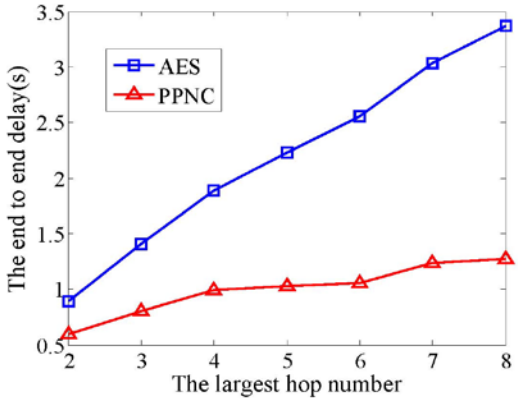

Fig. 10. End to end delay under different hop numbers.

Compared with AES algorithm, our scheme has much lower end-to-end delay. AES algorithm requires every forwarder node to decrypt the received packet's tag before making arithmetical operation on them; and then encrypts the operation results before forwarding. Both encryption and decryption operations introduce extra network latency. In contrast, our scheme frees the forwarder nodes from the complicated encryption and decryption operations. Forwarder nodes carry out arithmetical operation on the ciphertext as if they were plaintext, which saves much time for the whole process.

7. Conclusion

We propose a distributed privacy preserving scheme that considers the converged flows character of smart grid and exploits a homomorphic encryption function to decrease the complexity in forwarder node to solve privacy preserving in SG with random linear network coding. The data of the packet is encrypted and the tag of the packet is encrypted by homomorphic encryption function. Then the forwarder node random linearly codes the encrypted data and directly updates the cryptotext tags based on the homomorphic feature. The

scheme offers data confidentiality privacy preserving feature including anonymity, unlinkability, undetectability and unobservability, which can efficiently thwart traffic analysis, but pseudonymity, identity management. We conduct extensive performance evaluations and security analysis, which demonstrate that our scheme can effectively maintain privacy and have low computation and communication overhead.

References

[1] F. Cohen, "The smarter grid," *IEEE Security and Privacy*, vol. 8, no. 1, pp. 60-63, Feb. 2010. Article (CrossRef Link).

[2] D. Marihart, "Communications technology guidelines for ems/scada systems," *IEEE Transactions on Power Delivery*, vol. 16, no. 2, pp. 181-188, Apr. 2001. Article (CrossRef Link).

[3] G. Deconinck, "An evaluation of two-way communication means for advanced metering in flanders (belgium)," in *Proc. of IEEE IMTC*, pp. 900-905, May 12-15, 2008. Article (CrossRef Link).

[4] Q. Gao, J. Yu, P. Chong, P. So and E. Gunawan, "Solutions for the silent node problem in an automatic meter reading system using powerline communications," *IEEE Transactions on Power Delivery*, vol. 23, no. 1, pp. 150-156, Jan. 2008. Article (CrossRef Link).

[5] B. Sivaneasan, E. Gunawan and P. So, "Modeling and performance analysis of automatic meter-reading systems using plc under impulsive noise interference," *IEEE Transactions on Power Delivery*, vol. 25, no.3, pp. 1465-1475, July 2010. Article (CrossRef Link).

[6] S. Galli, A. Scaglione, and Z.Wang, "For the grid and through the grid: The role of power line communications in the smart grid," in *Proc. of the IEEE*, vol. 99, no. 6, pp. 998-1027, June 2011. Article (CrossRef Link).

[7] V. Gungor, B. Lu and G. Hancke, "Opportunities and challenges of wireless sensor networks in smart grid," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 10, pp. 3557-3564, Oct. 2010. Article (CrossRef Link).

[8] M. Souryal, C. Gentile, D. Griffith, D. Cypher and N. Golmie, "A methodology to evaluate wireless technologies for the smart grid," in *Proc. of IEEE SmartGridComm*, pp. 356-361, Oct. 4-6, 2010. Article (CrossRef Link).

[9] H. Nicanfar, P. TalebiFard, A. Alasaad and V. C. Leung, "Privacy-preserving scheme in smart grid communication using enhanced network coding," in *Proc. of ICC*, pp. 2022–2026, June 9-13, 2013. Article (CrossRef Link).

[10] H. Nicanfar, P. TalebiFard, A. Alasaad and V. C. Leung, "Enhanced network coding to maintain privacy in smart grid communication," *IEEE Transactions on Emerging Topics in Computing*, vol. 1, no. 2, pp. 286-296, Dec. 2013. Article (CrossRef Link).

[11] M. Karthick and K. M. Sivalingam, "Network coding based reliable and efficient data transfer for smart grid monitoring," in *Proc. of IEEE ANTS*, pp. 1-6, Dec. 15-18, 2013. Article (CrossRef Link).

[12] Y. Phulpin, J. Barros and D. Lucani, "Network coding in smart grids," in *Proc. of IEEE SmartGridComm*, pp. 49-54, Oct. 17-20, 2011. Article (CrossRef Link).

[13] G. Rajalingham, Q. D. Ho and T. Le-Ngoc, "Random linear network coding for converge-cast smart grid wireless networks," in *Proc. of QBSC*, pp. 208-212, June 1-4, 2014. Article (CrossRef Link).

[14] R. Prior, D. E. Lucani, Y. Phulpin, M. Nistor, and J. Barros, "Network coding protocols for smart grid communications," *IEEE Transaction on Smart Grid*, vol. 5, no. 3, pp. 1523-1531, May 2014. Article (CrossRef Link).

[15] P. McDaniel and S. McLaughlin, "Security and privacy challenges in the smart grid," *IEEE Security and Privacy*, vol. 7, no. 3, pp. 75-77, May-June 2009. Article (CrossRef Link).

[16] C. Efthymiou and G. Kalogridis, "Smart grid privacy via anonymization of smart metering data," in *Proc. of IEEE SmartGridComm*, pp. 238-243, Oct. 4-6, 2010. Article (CrossRef Link).

[17] I. Richardson, A. Thomson, D. Infield and C. Clifford, "Domestic electricity use: A high-resolution energy demand model," *Energy and Buildings*, vol. 42, no. 10, pp. 1878-1887, Oct. 2010. Article (CrossRef Link).
[18] L. Chen, R. Lu, Z. Cao, K. AlHarbi and X. Lin., "Muda: Multifunctional data aggregation in privacy-preserving smart grid communications," *Peer-to-Peer Networking and Application*, vol. 8, no. 5, pp. 1-16, June 2014. Article (CrossRef Link).
[19] J. Won, C. Y. T. Ma, D. K. Y. Yau and N. Rao, "Proactive fault-tolerant aggregation protocol for privacy-assured smart metering," in *Proc. of Infocom*, pp. 2804-2812, April 27- May 2, 2014. Article (CrossRef Link).
[20] J. Zhao, T. Jung, Y. Wang and X. Y. Li., "Achieving differential privacy of data disclosure in the smart grid," in *Proc. of Infocom*, pp. 504-512, April 27- May 2, 2014. Article (CrossRef Link).
[21] L. Yang, X. Chen, J. Zhang and H. V. Poor, "Optimal privacy-preserving energy management for smart meters," in *Proc. of Infocom*, pp. 513-521, April 27- May 2, 2014. Article (CrossRef Link).
[22] Y. Fan, Y. Jiang, H. Zhu and X. S. Shen, "An efficient privacy-preserving scheme against traffic analysis attacks in network coding," in *Proc. of Infocom*, pp. 2213-2221, April 19-25, 2009. Article (CrossRef Link).
[23] Y. Fan, Y. Jiang, H. Zhu and X. S. Shen, "Network coding based privacy preservation against traffic analysis in multi-hop wireless networks," *IEEE Transactions on Wireless Communications*, vol. 10, no. 3, pp. 834-843, Dec. 2011. Article (CrossRef Link).
[24] Z. Fu, X. Sun, Q. Liu, L. Zhou and J. Shu, "Achieving efficient cloud search services: Multi-keyword ranked search over encrypted cloud data supporting parallel computing," *IEICE Transactions on Communications*, vol. E98-B, no. 1, pp. 190-200, Jan. 2015. Article (CrossRef Link).
[25] Z. Fu, K. Ren, J. Shu, X. Sun and F. Huang, "Enabling personalized search over encrypted outsourced data with efficiency improvement," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 9, pp.2546-2559, Sept. 2016. Article (CrossRef Link).
[26] Z. Fu, X. Wu, C. Guan, X. Sun and K. Ren, "Towards efficient multi-keyword fuzzy search over encrypted outsourced data with accuracy improvement," *IEEE Transactions on Information Forensics and Security*. vol. 11, no. 12, pp. 2706-2716, Dec. 2016. Article (CrossRef Link).
[27] S. Xie and Y. Wang, "Construction of Tree Network with Limited Delivery Latency in Homogeneous Wireless Sensor Networks," *Wireless Personal Communications*, vol. 78, no. 1, pp. 231-246, Sept. 2014. Article (CrossRef Link).
[28] P. Guo, J. Wang, B. Li and S. Lee, "A Variable Threshold-value Authentication Architecture for Wireless Mesh Networks," *Journal of Internet Technology*, vol. 15, no. 6, pp. 929-936, Nov. 2014. Article (CrossRef Link).
[29] M. Ambrosin, "Verifiable and Privacy-preserving Fine-Grained Data-Collection for Smart Metering," in *Proc. of the 1st Workshop on Security and Privacy in Cybermatics (IEEE CNS 2015 Workshop: SPiCy 2015)*, pp. 655 - 658, Sept. 28-30, 2015. Article (CrossRef Link).
[30] V. Odelu, A. K. Das, M. Wazid and M. Conti, "Provably Secure Authenticated Key Agreement Scheme for Smart Grid", *IEEE Transaction on Smart Grid*, Aug. 2016. Article (CrossRef Link)
[31] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov and M. Naor, "Our data, ourselves: Privacy via distributed noise generation," in *Proc. of EUROCRYPT*, pp.486-503, May 28 - June 1, 2006. Article (CrossRef Link)
[32] C. Dwork, F. McSherry, K. Nissim and A. Smith, "Calibrating noise to sensitivity in private data analysis," in *Proc. of TCC*, pp.265-284, Mar. 4-7, 2006. Article (CrossRef Link)
[33] C. Dwork, "Differential privacy," in *Proc. of ICALP*, pp. 1-12, Jul. 10-14, 2006. Article (CrossRef Link)
[34] W. Zeng, Y. Lin, S. He and J. Yu, "Data aggregation based on the privacy-preserving element in wireless sensor networks," *Journal on Communications*, vol. 33, no.10, pp. 16-25, Oct. 2012. Article (CrossRef Link)
[35] W. Zeng, Y. Lin, L. Wang and S. He, "Privacy-preserving data aggregation scheme based on the p-function set in wireless sensor networks," *Ad Hoc & Sensor Wireless Networks*, vol. 21, no. 1, pp. 21-58, Jan. 2014. Article (CrossRef Link)

[36] S. He, W. Zeng and K. Xie, "Privacy preserving for network coding in smart grid," in *Proc. of The 15th International Conference on Algorithms and Architectures for Parallel Processing (ICA3PP 2015)*, pp. 640-654, Nov. 18-20, 2015. Article (CrossRef Link)
[37] S. Li, R. Yeung and N. Cai, "Linear network coding," *IEEE Transaction Information Theory*, vol. 49, no. 2, pp. 371-381, Feb. 2003. Article (CrossRef Link)
[38] D. Boneh, E. Goh and K. Nissim, "Evaluating 2-dnf formulas on ciphertexts," *Theory of cryptography*, vol. 3378, pp. 325-341, Feb. 2005. Article (CrossRef Link)
[39] T. Ho, M. Médard, R. Koetter, D. R. Karger, M. Effros, J. Shi and B. Leong, "A random linear network coding approach to multicast," *IEEE Transaction Information Theory*, vol. 52, no. 10, pp. 4413-4430, Oct. 2006. Article (CrossRef Link)
[40] S Chachulski, M Jennings, S Katti and D Katabi. "Trading structure for randomness in wireless opportunistic routing," In *Proc. of ACM SIGCOMM 2007*, vol. 37, no. 4, pp.169-180, Aug. 27-31, 2007. Article (CrossRef Link)
[41] J. P. Vilela, L. Lima and J. Barros, "Lightweight security for network coding," in *Proc. of ICC*, pp. 1750-1754, May 19-23, 2008. Article (CrossRef Link)
[42] S. Feizi, D. E. Lucani and M. Mdard, "Tunable sparse network coding," in *Proc. of Int. Zurich Seminar on Communications (IZS)*, pp. 107-110, Feb. 29- Mar. 2, 2012. Article (CrossRef Link)
[43] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proc. of Advances in Cryptology-EUROCRYPT 99*, pp.223–238, May 2-6, 1999. Article (CrossRef Link)
[44] A. Menezes, P. Oorschot, and S. Vanstone, "Handbook of applied cryptography," *CRC press*, Jan. 1996. Article (CrossRef Link)
[45] B. Blakley, "What is Privacy, Really?" *presentation from Digital ID World 2006*. Oct. 23, 2006. Article (CrossRef Link)
[46] National Institute of Standard and Technology. Article (CrossRef Link)
[47] A. Pfitzmann and M. Hansen, "A Terminology for talking about privacy by data minimization: Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management," 2010. Article (CrossRef Link)
[48] J. Shen, H. Tan, J. Wang, J. Wang and S. Lee, "A Novel Routing Protocol Providing Good Transmission Reliability in Underwater Sensor Networks," *Journal of Internet Technology*, vol. 16, no. 1, pp. 171-178, Jan. 2015. Article (CrossRef Link)
[49] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Transaction Information Theory*, vol. 29, no. 2, pp. 198-208, Mar. 1983. Article (CrossRef Link)
[50] N. Fips, "Announcing the advanced encryption standard (aes) ", *National Institute of Standards and Technology (NIST)*, vol. 29, no. 8, pp. 2200-2203, Jan. 2001. Article (CrossRef Link)

Shiming He received the Ph.D. degree in computer application from Hunan University, China, in 2013. She is a lecture in School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha, China. Her current research interests include privacy preserving, wireless network and mobile computing.

Weini Zeng received the Ph.D. degree from Hunan University, China, in 2011. She is a Senior Engineer in the 716th Institute of China Shipbuilding Industry Corporation, China. Her current research interests include sensor networks and information security.

Kun Xie received the Ph.D. degrees in computer application from Hunan University, Changsha, China, in 2007. She is currently an associate professor in Hunan University. Her research interests include wireless network and mobile computing, network management and control, cloud computing and mobile cloud, and big data.

Hongming Yang received the Ph.D. degree in electrical engineering from Huazhong University of Science and Technology in 2003. She is a full Professor in Changsha University of Science and Technology. Her research interests include power system analysis and power markets.

Mingyong Lai received the Ph.D. degree in systems engineering from the National University of Defense Technology in 1997. He is currently a Full Professor with the Changsha University of Science and Technology. His special fields of interest include management systems engineering and industrial economics.

Xin Su received the Ph.D. degree from College of Information Science and Electronic Engineering, Hunan University, Changsha, China, in 2015. He is currently a lecture at the Department of Information Technology, Hunan Academy Police, Changsha, China. His main research interests are in mobile application network traffic analysis and mobile malware detection.