Network Coding Based Privacy Preservation against Traffic Analysis in Multi-Hop Wireless Networks

Yanfei Fan, Yixin Jiang, Haojin Zhu, *Member, IEEE*, Jiming Chen, *Member, IEEE*,
and Xuemin (Sherman) Shen, *Fellow, IEEE*

***Abstract*—Privacy threat is one of the critical issues in multi-hop wireless networks, where attacks such as traffic analysis and flow tracing can be easily launched by a malicious adversary due to the open wireless medium. Network coding has the potential to thwart these attacks since the coding/mixing operation is encouraged at intermediate nodes. However, the simple deployment of network coding cannot achieve the goal once enough packets are collected by the adversaries. On the other hand, the coding/mixing nature precludes the feasibility of employing the existing privacy-preserving techniques, such as Onion Routing. In this paper, we propose a novel network coding based privacy-preserving scheme against traffic analysis in multi-hop wireless networks. With homomorphic encryption on Global Encoding Vectors (GEVs), the proposed scheme offers two significant privacy-preserving features, packet flow untraceability and message content confidentiality, for efficiently thwarting the traffic analysis attacks. Moreover, the proposed scheme keeps the random coding feature, and each sink can recover the source packets by inverting the GEVs with a very high probability. Theoretical analysis and simulative evaluation demonstrate the validity and efficiency of the proposed scheme.**

***Index Terms*—Network coding, homomorphic encryption, privacy preservation, traffic analysis.**

I. Introduction

Wireless access networks, such as Wi-Fi, have been widely deployed due to their convenience, portability, and low cost. However, they still suffer inherent shortcomings such as limited radio coverage, poor system reliability, and lack of security and privacy. Multi-hop Wireless Networks (MWNs) are regarded as a highly promising solution for extending the radio coverage range of the existing wireless networks, and they can also be used to improve the system reliability through multi-path packet forwarding.

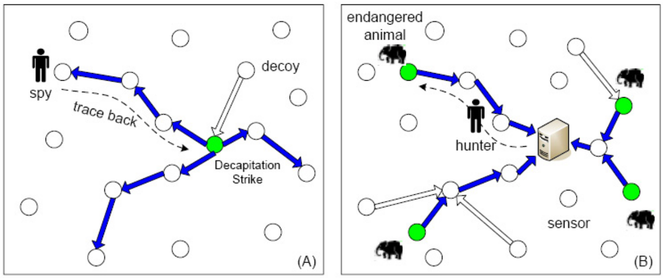


Fig. 1. Privacy threats in MWNs.

However, due to the open wireless medium, MWNs are susceptible to various attacks, such as eavesdropping, data modification/injection, and node compromising. These attacks may breach the security of MWNs, including confidentiality, integrity, and authenticity. In addition, some advanced attacks, such as traffic analysis and flow tracing, can also be launched by a malicious adversary to compromise users' privacy, including source anonymity and traffic secrecy. In this paper, we focus on the privacy issue, i.e., how to prevent traffic analysis/flow tracing and achieve source anonymity in MWNs.

Among all privacy properties, source anonymity is of special interest in MWNs. Source anonymity refers to communicating through a network without revealing the identity or location of source nodes. Preventing traffic analysis/flow tracing and provisioning source anonymity are critical for privacy-aware MWNs, such as wireless sensor or tactical networks. Consider a simple example of multicast communication in military ad hoc networks, where nodes can communicate with each other through multi-hop packet forwarding. If an attacker can intercept packets and trace back to the source through traffic analysis, it may disclose some sensitive information such as the location of critical nodes (e.g., the commanders) and then further it may impair the location privacy. Subsequently, the attacker can take a series of actions to launch the so-called Decapitation Strike to destroy these critical nodes [1], as shown in Fig. 1(A). Another example is the event reporting in wireless sensor networks, where flow tracing can help attackers to identify the location of concerned events, e.g., the appearance of an endangered animal in a monitored area, and then take subsequent actions to capture or kill the animals [2], as shown in Fig. 1(B).

It is very challenging to efficiently thwart traffic analysis/flow tracing attacks and provide privacy protection in MWNs. Existing privacy-preserving solutions, such as proxy-based schemes [3], [4], Chaum's mix-based schemes [5], [6], and onion-based schemes [7], [8], may either require a series of trusted forwarding proxies or result in severe performance degradation in practice. Different from previous schemes, we investigate the privacy issue from a brand new perspective: using network coding to achieve privacy preservation.

Manuscript received January 20, 2010; revised June 9, 2010; accepted August 25, 2010. The associate editor coordinating the review of this paper and approving it for publication was C.-F. Chiasserini.

Y. Fan and X. Shen are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, N2L 3G1, Canada (e-mail: {yfan, xshen}@bbcr.uwaterloo.ca).

Y. Jiang is with the Department of Computer Science and Technology, Tsinghua University, Beijing, 100084, China (e-mail: yixin.tsinghua@gmail.com).

H. Zhu is with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, 200030, China (e-mail: zhu-hj@cs.sjtu.edu.cn).

J. Chen is with the State Key Lab of Industrial Control Technology, Department of Control, Zhejiang University, Hangzhou, 310027, China (e-mail: jmchen@iipc.zju.edu.cn).

Part of this paper was presented at IEEE INFOCOM'09, Rio de Janeiro, Brazil, April 2009.

This work was supported by the Natural Sciences and Engineering Research Council (NSERC) of Canada.

Digital Object Identifier 10.1109/TWC.2011.122010.100087

Network coding was first introduced by Ahlswede et al [9]. Subsequently, two key techniques, random coding [10] and linear coding [11], [12] ([12] gives the first distributed implementation), further promoted the development of network coding. The random coding makes network coding more practical, while the linear coding is proven to be sufficient and computationally efficient for network coding. Currently, network coding has been widely recognized as a promising information dissemination approach to improve network performance. Primary applications of network coding include file distribution and multimedia streaming on P2P overlay networks [13], data transmission in sensor networks [14], tactical communications in military networks [15], etc. Compared with conventional packet forwarding technologies, network coding offers, by allowing and encouraging coding/mixing operations at intermediate forwarders [9], several significant advantages such as potential throughput improvement [16], transmission energy minimization [17], and delay reduction [18]. In addition, network coding can work as erasure codes to enhance the dependability of a distributed data storage system [19]-[22].

The deployment of network coding in MWNs can not only bring the above performance benefits, but also provide a feasible way to efficiently thwart the traffic analysis/flow tracing attacks since the coding/mixing operation is encouraged at intermediate nodes. Similar to Chaum's mix-based schemes [5], [6], network coding provides an intrinsic message mixing mechanism, which implies that privacy preservation may be efficiently achieved in a distributed manner [23]. Moreover, the unlinkability between incoming packets and outgoing packets, which is an important privacy property for preventing traffic analysis/flow tracing, can be achieved by mixing the incoming packets at intermediate nodes. However, the privacy offered by such a mixing feature is still vulnerable, since the linear dependence between outgoing and incoming packets can be easily analyzed. A simple deployment of network coding cannot prevent traffic analysis/flow tracing since the explicit Global Encoding Vectors (GEVs, also known as tags) prefixed to the encoded messages provide a back door for adversaries to compromise the privacy of users. Once enough coded packets are collected, adversaries can easily recover the original packets and then conduct the attacks based on these packets. A naive solution to address this vulnerability is to employ link-to-link encryption. This solution can prevent traffic analysis to a certain degree, but it introduces heavy computational overhead and thus results in significant performance degradation of the whole network system. Additionally, it cannot protect the privacy of users once some intermediate nodes are compromised by adversaries. Such deficiencies motivate us to explore an efficient privacy-preserving scheme for MWNs.

In this paper, based on network coding and Homomorphic Encryption Functions (HEFs) [24], [25], we propose an efficient privacy-preserving scheme for MWNs. Our objective is to achieve source anonymity by preventing traffic analysis and flow tracing. To the best of our knowledge, this is the first research effort in utilizing network coding to thwart traffic analysis/flow tracing and realize privacy preservation. The proposed scheme offers the following attractive features: 1) **Enhanced Privacy against traffic analysis and flow tracing**. With the employment of HEFs, the confidentiality of GEVs is effectively guaranteed, making it difficult for attackers to recover the plaintext of GEVs. Even if some intermediate nodes are compromised, the adversaries still cannot decrypt the GEVs, since only the sinks know the decryption key. Further, the confidentiality of GEVs brings an implicative benefit, i.e., the confidentiality of message content [26], because message decoding only relies on GEVs. On the other hand, with random recoding on encrypted GEVs, the coding/mixing feature of network coding can be exploited in a natural manner to satisfy the mixing requirements of privacy preservation against traffic analysis; 2) **Efficiency**. Due to the Homomorphism of HEFs, message recoding at intermediate nodes can be directly performed on encrypted GEVs and encoded messages, without knowing the decryption keys or performing expensive decryption operations on each incoming packet. The performance evaluation on computational complexity demonstrates the efficiency of the proposed scheme; and 3) **High Invertible Probability**. Random network coding is feasible only if the prefixed GEVs are invertible with a high probability. Theoretical analysis demonstrates that the influence of HEFs on the invertible probability of GEVs is negligible. Thus, the random coding feature can be kept in our network coding based privacy-preserving scheme.

The remainder of the paper is organized as follows. In Section II, preliminaries related to the proposed scheme are given, including network coding, homomorphic encryption functions, and threat models. In Section III, the proposed privacy-preserving scheme is presented in detail. In Sections IV and V, security analysis and performance evaluation/optimization are conducted, respectively. In Section VI, related work is surveyed, followed by the conclusions in Section VII.

II. Preliminaries

A. Network Coding

Unlike other packet-forwarding systems, network coding allows intermediate nodes to perform computation on incoming messages, making outgoing messages be the mixture of incoming ones. This elegant principle implies a plethora of surprising opportunities, such as random coding [10]. As shown in Fig. 2, whenever there is a transmission opportunity for an outgoing link, an outgoing packet is formed by taking a random combination of packets in the current buffer. An overview of network coding and possible applications has been given in [18]. In practical network coding, source information should be divided into blocks with h packets in each block. All coded packets related to the kth block belong to generation k and random coding is performed only among the packets in the same generation. Packets within a generation need to be synchronized by buffering for the purpose of network coding at intermediate nodes.

Consider an acyclic network (V, E, c) with unit capacity, i.e., $c(e) = 1$ for all $e \in E$, meaning that each edge can carry one symbol per unit time, where V is the node set and E is the edge set. Assume that each symbol is an element of a finite field $\mathbb{F}_q$. Consider a network scenario with multicast sessions, where a session is comprised of one source $s \in V$ and a set of sinks $T \subseteq V$ (or one single sink $t \in V$). Let

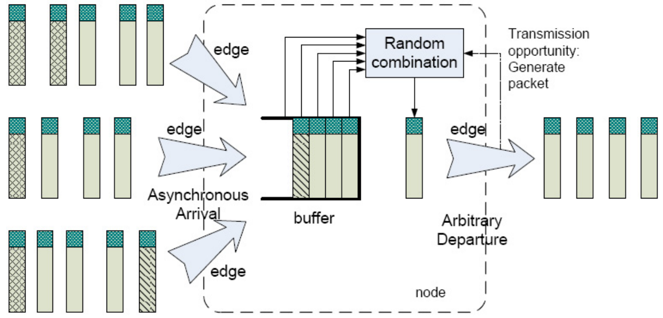


Fig. 2. Random coding (mixing) at intermediate nodes.

$h = MinCut(s, T)$ be the multicast capacity, and $x_1, \cdots, x_h$ be the h symbols to be delivered from s to T.

For each outgoing edge e of a node v, let $y(e) \in \mathbb{F}_q$ denote the symbol carried on e, which can be computed as a linear combination of the symbols $y(e')$ on the incoming edges e' of node v, i.e., $y(e) = \sum_{e'} \beta_{e'}(e) y(e')$. The coefficient vector $\boldsymbol{\beta}(e) = [\beta_{e'}(e)]$ is called *Local Encoding Vector* (LEV). By induction, the symbol $y(e)$ on any edge $e \in E$ can be computed as a linear combination of the source symbols $x_1, \cdots, x_h$, i.e., $y(e) = \sum_{i=1}^{h} g_i(e) x_i$. The coefficients form a *Global Encoding Vector* (GEV) $\mathbf{g}(e) = [g_1(e), \cdots, g_h(e)]$, which can be computed recursively as $\mathbf{g}(e) = \sum_{e'} \beta_{e'}(e) \mathbf{g}(e')$, using the LEVs $\boldsymbol{\beta}(e)$. Suppose that a sink $t \in T$ receives symbols $y(e_1), \cdots, y(e_h)$, which can be expressed in terms of the source symbols as

$$\begin{bmatrix} y(e_1) \\ \vdots \\ y(e_h) \end{bmatrix} = \begin{bmatrix} g_1(e_1) & \cdots & g_h(e_1) \\ \vdots & \ddots & \vdots \\ g_1(e_h) & \cdots & g_h(e_h) \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_h \end{bmatrix} = G_t \begin{bmatrix} x_1 \\ \vdots \\ x_h \end{bmatrix}, \tag{1}$$

where G_t is called *Global Encoding Matrix* (GEM) and the ith row of G_t is the GEV associated with $y(e_i)$. Sink t can recover the h source symbols by inverting G_t and then applying the inverse to $y(e_1), \cdots, y(e_h)$.

In general, each packet can be considered as a vector of symbols $\mathbf{y}(e) = [y_1(e), \cdots, y_N(e)]$. By likewise grouping the source symbols into packets $\mathbf{x}_i = [x_{i,1}, \cdots, x_{i,N}]$, the above algebraic relationships carry over to packets. To facilitate the decoding at the sinks, each message should be tagged with its GEV $\mathbf{g}(e)$, which can be easily achieved by prefixing the ith source packet $\mathbf{x}_i$ with the ith unit vector $\mathbf{u}_i$. Then, each packet is automatically tagged with the corresponding GEV, since

$$\begin{aligned} [\boldsymbol{g}(e), \boldsymbol{y}(e)] &= \sum_{e'} \beta_{e'}(e) [\boldsymbol{g}(e'), \boldsymbol{y}(e')] \\ &= \sum_{i=1}^{h} g_i(e) [\boldsymbol{u}_i, \boldsymbol{x}_i] \end{aligned} . \tag{2}$$

The benefit of tags is that the GEVs can be found within the packets themselves, so that the sinks can compute G_t without knowing the network topology or packet-forwarding paths. Nor is a side channel required for the communication of G_t. 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.

B. Homomorphic Encryption Functions

Homomorphic Encryption Functions (HEFs) have the property of homomorphism, which means operations on plaintext can be performed by operating on corresponding ciphertext. If $E(\cdot)$ is a HEF, $E(x + y)$ can be computed from $E(x)$ and $E(y)$ without knowing the corresponding plaintext x and y. To be applicable in the proposed scheme, a HEF $E(\cdot)$ needs to satisfy the following properties:

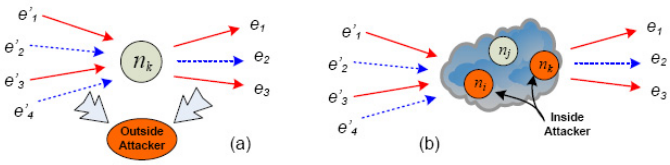


Fig. 3. Attack model: (a) outside attacker; (b) inside attacker.

1) **Additivity**: Given the ciphertext $E(x)$ and $E(y)$, there exists a computationally efficient algorithm $Add(\cdot, \cdot)$ such that $E(x + y) = Add(E(x), E(y))$.

2) **Scalar Multiplicativity**: Given $E(x)$ and a scalar t, there exists a computationally efficient algorithm $Mul(\cdot, \cdot)$ such that $E(x \cdot t) = Mul(E(x), t)$.

Actually, the scalar multiplicativity can be deduced from the additivity, since $E(x \cdot t) = E(\sum_{i=1}^{t} x)$. Benaloh [24] and Paillier [25] cryptosystems are of such an additive HEF, where the addition on plaintext can be achieved by performing a multiplicative operation on the corresponding ciphertext, i.e., $E(x_1 + x_2) = E(x_1) \cdot E(x_2)$. Further, the following two equations can be easily derived:

$$\begin{aligned} E(x \cdot t) &= E^t(x) \\ E(\sum_i x_i \cdot t_i) &= \prod_i E^{t_i}(x_i) \end{aligned} . \tag{3}$$

C. Threat Models

We consider the following two attack models.

Outside Attacker: An outside attacker can be considered as a global passive eavesdropper who has the ability to observe all network links, as shown in Fig. 3 (a). An outside attacker can examine the tags and message content, and thus link outgoing packets with incoming packets. Further, even if end-to-end encryption is applied to messages at a higher layer, it is still possible for a global outside attacker to trace packets by analyzing and comparing the message ciphertext.

Inside attacker: An inside attacker may compromise several intermediate nodes, as shown in Fig. 3 (b). Link-to-link encryption is vulnerable to inside attackers since they may already have obtained the decryption keys and thus the message plaintext can be easily recovered.

Both inside and outside attackers may perform more advanced traffic analysis/flow tracing techniques, including size correlation, time correlation, and message content correlation [27]. Adversaries can further explore these techniques to deduce the forwarding paths [2] and thus to compromise user privacy.

Without loss of generality, we assume that an anonymous secure routing protocol [1] is deployed to assist network nodes to determine forwarding paths. The generation number of a packet can be hidden in the secure routing scheme through link-to-link encryption. In this way, attackers cannot find the generation number of a packet for their further analysis. Notice that secure routing paths are only required to be established at the beginning of each session; during the packet transmission, secure routing paths are not required to change or re-established for each new generation.

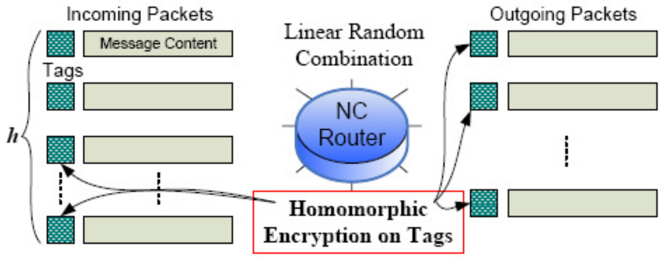


Fig. 4. Homomorphic encryption on packet tags.

III. Network Coding Based Privacy-Preserving Scheme for MWNs

In this section, we propose a novel network coding based privacy-preserving scheme for MWNs, followed by theoretical analysis on the invertibility of GEMs.

A. The Proposed Privacy-Preserving Scheme

Though providing an intrinsic mixing mechanism, the original network coding cannot provide privacy guarantee due to explicit GEVs, since an adversary can recover the original messages as long as enough packets are collected. Link-to-link encryption is vulnerable to inside attackers since they may already have compromised several intermediate nodes and obtained the secret keys. An intuitive way to resolve this issue is to keep GEVs confidential to intermediate nodes by encrypting the GEVs in an end-to-end manner, which can prevent compromised intermediate nodes from analyzing GEVs or recovering the original messages. Such an intuitive approach, however, cannot prevent the adversaries from tracking the message ciphertext since the "mixing" feature of network coding may be disabled by the end-to-end encryption.

To address this issue, we employ the Paillier cryptosystem [25] as the HEF to apply encryption to GEVs, since protecting GEVs is generally sufficient to ensure confidentiality network coded message content [26]. HEF can not only keep the confidentiality of GEVs, but also enable intermediate nodes to efficiently mix the coded messages. In the Paillier cryptosystem, given a message m and the public key $(n,\ g)$, the encryption function can be described as $E(m) = g^m \cdot r^n (mod\ n^2)$, where r is a random factor. $E(m)$ satisfies the homomorphic property: $E(m_1) \cdot E(m_2) = g^{m_1+m_2} \cdot (r_1 r_2)^n (mod\ n^2) = E(m_1 + m_2)$.

With HEFs, intermediate nodes are allowed to directly perform linear coding/mixing operations on the coded messages and encrypted tags, as shown in Fig. 4. In other words, due to the homomorphism of the HEF, one can achieve linear network coding by operating on encoded messages and encrypted GEVs, without knowing the decryption keys or performing the decryption operations.

The proposed scheme consists of three phases: source encoding, intermediate recoding, and sink decoding. Without loss of generality, we assume that each sink acquires two keys, the encryption key *ek* and the decryption key *dk*, from an offline Trust Authority (TA). For supporting multicast, a group of sinks are required to obtain from the TA or negotiate the key pair in advance [28]. Then, the encryption key is published and the decryption key is kept secret.

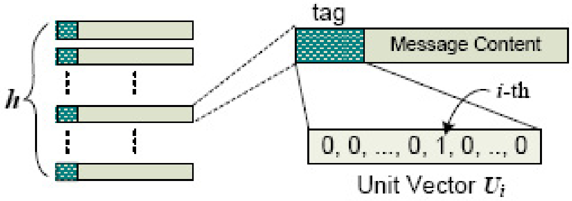


Fig. 5. Packet tagging before source encoding.

Source Encoding: Consider that a source has h messages, say $x_1, \cdots, x_h$, to be sent out. The source first prefixes h unit vectors to the h messages, respectively, as illustrated in Fig. 5. After tagging, the source can choose a random LEV and perform linear encoding on these messages. Then, a LEV can produce an encoded message with the GEV (which is equal to the LEV temporarily) tagged.

To offer confidentiality for the tags, homomorphic encryption operations are applied as follows:

$$\begin{array}{rcl} c_i(e) & = & E_{ek}(g_i(e)), \quad (1 \leqslant i \leqslant h) \\ \boldsymbol{c}(e) & = & [c_1(e), c_2(e), \cdots, c_h(e)] \end{array} . \tag{4}$$

where the notation *ek* denotes the encryption key. Notice that we adopt the strategy of applying HEF to GEVs after (instead of before) linear encoding, which will be discussed in Section IV from the perspective of both security and performance.

Intermediate Recoding: After receiving a number of packets of the same generation, an intermediate node can perform random linear coding on these packets. To generate an outgoing packet, firstly, a random LEV $[\beta_1, \cdots, \beta_h]$ is chosen independently; then, a linear combination of message content of the incoming packets is computed as the message content of the outgoing packet, as shown in Fig. 2.

Since the tags of the h incoming packets are in ciphertext format, and an intermediate node has no knowledge of the corresponding decryption keys, it is difficult for the intermediate node to perform functions such as earliest decoding to get the original message content. However, due to the homomorphism of the encryption function, a linear transformation can be directly performed on the encrypted tags of the incoming packets to generate a new tag for the outgoing packet, namely,

$$g(e) = \sum_{i=1}^{h} \beta_i(e) g(e'_i). \tag{5}$$

The GEV of a new outgoing packet can be calculated according to Eq. (5). By utilizing the homomorphic characteristic of the encryption on GEVs, the ciphertext of the new GEVs for outgoing packets can be calculated as follows:

$$\begin{array}{rcl} E_{ek}(g(e)) & = & E_{ek}(\sum_{i=1}^{h} \beta_i(e) g(e'_i)) \\ & = & \prod_{i=1}^{h} E_{ek}(\beta_i(e) g(e'_i)) \\ & = & \prod_{i=1}^{h} E_{ek}^{\beta_i(e)}(g(e'_i)) \end{array} \tag{6}$$

The ciphertext of new GEVs can be computed from the ciphertext of GEVs of incoming packets without the knowledge of the decryption key. Finally, the ciphertext of a new GEV is prefixed to the corresponding message content to form a new outgoing packet, which is sent out to downstream nodes.

Sink Decoding: After receiving a packet, the sink first decrypts the packet tag using the corresponding decryption key *dk*.

$$\begin{array}{rcl} g_i(e) & = & D_{dk}(c_i(e)) \quad (1 \leqslant i \leqslant h) \\ \boldsymbol{g}(e) & = & [g_1(e), g_2(e), \cdots, g_h(e)] \end{array} \tag{7}$$

Once enough packets are received, a sink can decode the packets to get the original messages. Then, the sink derives the decoding vector, which is the inverse of the GEM, as shown in the following equations.

$$\begin{array}{c} \boldsymbol{G}^{-1} \cdot \boldsymbol{G} = \boldsymbol{U} \\ \boldsymbol{G} = [\boldsymbol{g}(e_1), \boldsymbol{g}(e_2), \cdots, \boldsymbol{g}(e_h)]^T \end{array} \tag{8}$$

Finally, the sink can use the inverse to recover the original messages, shown as follows.

$$\begin{bmatrix} \boldsymbol{x}_1 \\ \vdots \\ \boldsymbol{x}_h \end{bmatrix} = \boldsymbol{G}^{-1} \begin{bmatrix} \boldsymbol{y}(e_1) \\ \vdots \\ \boldsymbol{y}(e_h) \end{bmatrix} \tag{9}$$

For random network coding, a key issue is the invertibility of a GEM. We discuss in detail the invertibility of a GEM as follows.

B. Invertibility of a GEM

Let GEM A be comprised of *h* GEVs with *h* elements in each GEV. $|A|$ and A^* are the determinant and the adjoint of the matrix A, respectively. According to the theory of linear algebra, finding the inverse of a square matrix A is equivalent to solving the corresponding system of linear equation with A being the coefficient matrix. Gaussian elimination can be applied to solve a system of linear congruence equations. Due to the homomorphism of the modulo congruence in terms of the addition, subtraction, and multiplication operations, a system of linear congruence equations can be separated into several single equations with one unknown in each equation as follows:

$$|A|x_i = \sum_{j=1}^{h} (-1)^{i+j} y_j M_{ij} \ (mod\ n) \tag{10}$$

A system of linear congruence equations is solvable if and only if every independent equation is solvable. The difference between solving a system of linear equations and solving a system of linear congruence equations lies in finding the inverse of $|A|$ modulo *n*. In order to further discuss the solutions, we formulate the linear congruence equations as follows:

$$|A|x_i = |\tilde{A}_i| (mod\ n) \quad (i = 1, \cdots, h), \tag{11}$$

where $|\tilde{A}_i| = \sum_{j=1}^{h} (-1)^{i+j} y_j M_{ij} \ (mod\ n)$.

Theorem 1: A system of linear congruence equations has a unique solution only if $|A| \neq 0$.

Proof: See Appendix A. ■

However, this condition is not sufficient for a system of linear congruence equations to have a unique solution, because a solution for $|A|x_i = |\tilde{A}_i|$ does not imply a corresponding solution for $|A|x_i = |\tilde{A}_i| \ (mod\ n)$.

Theorem 2: A system of linear congruence equations has d^h solutions if:

$$\left\{ \begin{array}{c} |A| \neq 0 \\ |\tilde{A}_i| \equiv 0 \ (mod\ d) \quad (i = 1, 2, \cdots, h) \end{array} \right. \tag{12}$$

where $d = gcd(|A|, n)$.

Proof: See Appendix B. ■

Corollary 1: A system of linear congruence equations has a unique solution if and only if:

$$\left\{ \begin{array}{c} |A| \neq 0 \\ gcd(|A|, n) = 1 \end{array} \right. \tag{13}$$

and $x_i = |A|^{-1} |\tilde{A}_i| \ (mod\ n) \quad (i = 1, 2, \cdots, h)$.

Proof: See Appendix C. ■

Theorem 2 and **Corollary 1** indicate that a solvable system of linear congruence equations does not imply the invertibility of the corresponding coefficient matrix. A stronger condition, i.e., $gcd(|A|, n) = 1$, is required for the invertibility of a coefficient matrix A modulo *n*. The above theorems and corollary generally hold whether *n* takes the value of a prime number *q* or the product of two prime numbers *p* and *q*. In section V, we will further give a quantitative analysis on the invertible probability of a coefficient matrix.

IV. Security Analysis

The proposed scheme can provide privacy preservation by means of resisting traffic analysis/flow tracing attacks such as size correlation, time correlation, and message content correlation. Size correlation can be naturally prevented since each message is trimmed to be of the same length in network coding based schemes. Time correlation can be effectively resisted by the inherent buffering technique [18] of network coding. Let the time length of buffering periods be T_b and the average arrival rate of coded packets be λ. The time correlation attack can succeed only when exactly one packet arrives in the buffering period T_b, since zero packets make the attack meaningless and more than one packet can induce the “mixing” operation, making time correlation useless. If coded packets arrive following the Poisson distribution, the probability of a successful time correlation attack can be given as follows:

$$Pr(1, \lambda \cdot T_b) = \lambda \cdot T_b \cdot e^{-\lambda \cdot T_b}. \tag{14}$$

From Eq. (14), it can be seen that the probability decreases exponentially with the time period T_b. On the other hand, the transmission delay increases linearly with the time period T_b. In practice, we can adaptively adjust parameter T_b according to the security and delay requirements.

Message content correlation can be resisted by the “mixing” feature of network coding. With the assistance of HEF, GEVs are kept confidential to eavesdroppers, making it difficult for adversaries to perform linear analysis on GEVs. In addition, HEF keeps the random coding feature, making the linear analysis on message content 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 $\boldsymbol{O}(h^3 + h \cdot l)$ in terms of multiplication, where l is the length of message content in terms of symbols. Thus, the computational complexity to analyze the intercepted w packets is $\boldsymbol{O}(C_w^h(h^3 + h \cdot l))$, which increases exponentially with w, as shown in Fig. 6. It can be seen that, compared with the previous network coding schemes, the proposed scheme significantly enhances

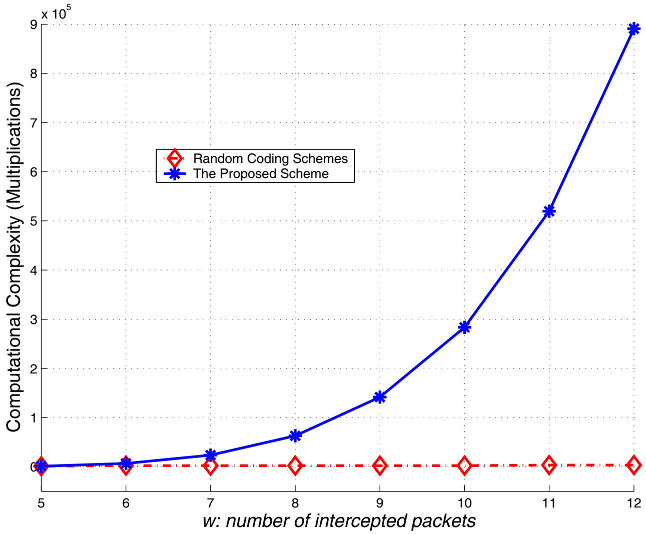


Fig. 6. Privacy enhancement in terms of the order of computational complexity (h=5, l=200).

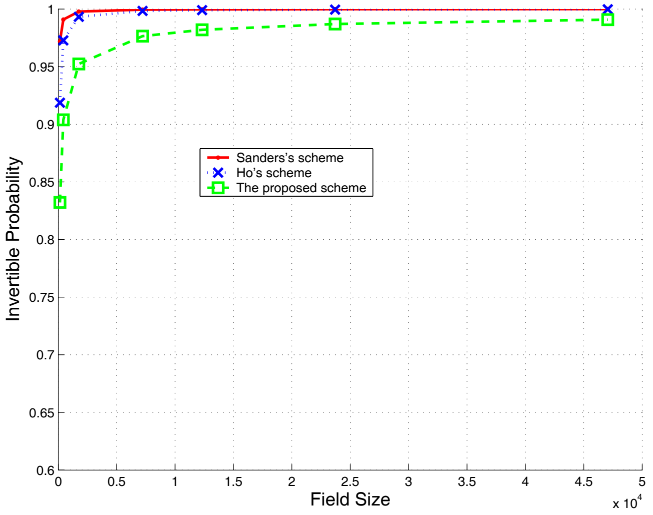


Fig. 7. Invertible probability vs. field size (theoretical analysis).

privacy preservation in terms of computational complexity, which makes the traffic analysis attacks almost impossible.

In the source encoding phase, we apply HEFs to GEVs after (instead of before) linear encoding. From security perspective, this choice is more secure since independent random factors can be chosen for each encryption operation, and these random factors can bring more randomness to the ciphertext of GEVs and make content correlation more difficult. From performance perspective, it is argued that source encoding may be more lightweight if HEFs are applied before linear coding and independent random factors are only chosen for different GEV elements. This argument is not proper since, for each new GEV element, linear coding after encryption requires averagely about h exponentiations and $h-1$ multiplications, which are computationally much more expensive than those of linear coding before encryption (which requires 2 exponentiations and 1 multiplication).

V. Performance Evaluation and Optimization

In this section, we evaluate the performance of the proposed scheme in terms of invertible probability and computational overhead. A performance optimization framework is also developed to minimize the statistical computational overhead.

A. Invertible Probability

Let each element of a LEV be randomly chosen from a field $\mathbb{F}_q$. The following two theorems hold.

Theorem 3: For a Local Encoding Matrix (LEM), which is comprised of h LEVs with h elements in each LEV and each element is from the finite field $\mathbb{F}_q$, the invertible probability of a GEM (also with h vectors) is degraded by $s_q = \prod_{i=1}^{h}(1 - q^{-i})$.

Proof: See Appendix D. ■

Corollary 2: The invertibility factor s_q of an $h \times h$ LEM can be approximated to $1 - q^{-1} - q^{-2}$ when $h \geqslant 4$, and the error of this approximation is within the magnitude of $\boldsymbol{O}(q^{-5})$.

This corollary can be easily proven by expanding the multiplication of the polynomials. This corollary gives two important implications. Firstly, in practical network coding, the min-cut capacity h is much larger than the condition in *corollary 2* and, thus, this corollary can be safely used. Secondly, the field size q is relatively a large number. Therefore, an amount in the magnitude $\boldsymbol{O}(q^{-5})$ is very small and can be omitted. Based on *Theorem 3* and *Corollary 2*, the invertible probability of a GEM can be easily calculated. For a network coding system with a min-cut capacity $h(h \geqslant 4)$, the invertible probability can be approximated as $(1-q^{-1}-q^{-2})^t$, where q is the field size and t is the total coding time from the source to sinks. In practical network coding, since q is a relatively large prime number, the above invertible probability can be further approximated to $1 - tq^{-1}$.

Theorem 3 does not apply to the Paillier cryptosystem, since elements in the cryptosystem are chosen from a ring (another algebraic structure), where and p, q are two prime numbers.

Theorem 3 does not apply to the Paillier cryptosystem, since elements in the cryptosystem are chosen from a ring $\mathbb{R}_n$(another algebraic structure), where $n = pq$ and p, q are two prime numbers.

Theorem 4: For a LEM (comprised of h LEVs), where the elements are randomly chosen from a ring $\mathbb{R}_n(n = pq)$, the invertible probability of a GEM (also with h vectors) is degraded by $s_p + s_q - s_n$.

Proof: See Appendix E. ■

If $|p| \approx |q|$, the integral invertibility factor can be approximately reduced to $1 - p^{-1} - q^{-1}$, with the error confined in $\boldsymbol{O}(p^{-2} + q^{-2})$. If a session performs totally t times of random coding, the invertible probability of GEVs at sinks can be approximately reduced to $1 - t(p^{-1} + q^{-1})$. It can be seen that the invertible probability is dependent on the random coding times, instead of the number of sinks.

We compare the analytical results of the invertible probability from the proposed scheme with those from the random coding schemes in [10] and [12], as shown in Fig. 7. It can be seen that the proposed scheme can maintain a very high invertible probability, which is similar to those of the random coding schemes; in addition, the proposed scheme can offer further privacy enhancement, which is very critical in practical

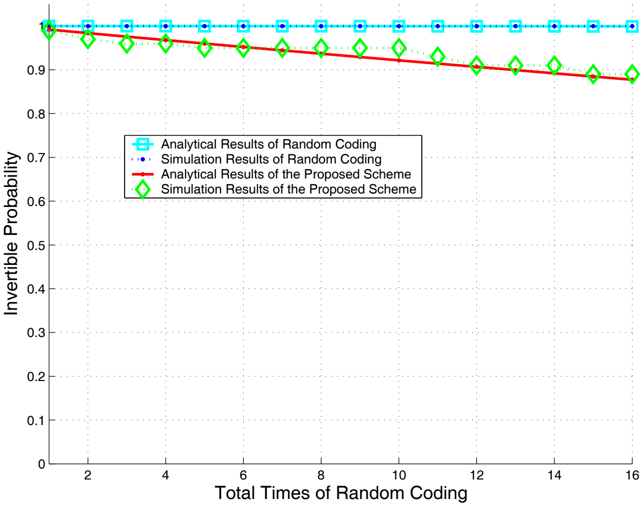


Fig. 8. Invertible probability vs. total times of random coding (p=241, q=251).

applications. Fig. 8 shows the analytical and simulation results of invertible probability versus random coding times. It can be seen that the invertible probability decreases with the increase of the random coding times. Furthermore, both the simulation and the analytical results match very well, which indicates the validity of the performance analysis.

B. Computational Overhead

The computational overhead of the proposed scheme can be investigated respectively from three aspects: source encoding, intermediate recoding, and sink decoding. Since the computational overhead of the proposed scheme is closely related to the specific homomorphic encryption algorithm, in the following analysis, we will take the Paillier cryptosystem as the encryption method when necessary. Note that the computational overhead is counted independent of the underlying network coding framework.

Source Encoding Overhead: Consider h GEVs with h elements in each GEV, which form an $h \times h$ GEM. After source encoding, every element in the GEM is encrypted one by one. Thus, the computational overhead is $\boldsymbol{O}(h^2)$ in terms of encryption operations. Every encryption operation requires 2 exponentiations, 1 multiplication, and 1 modulus operation in the Paillier cryptosystem. Therefore, the computational complexity is $\boldsymbol{O}(h^2 \cdot log\ n)$ in terms of multiplication operations.

Intermediate Recoding Overhead: In intermediate nodes, linear transformation on the elements of GEVs can be performed only by manipulating the ciphertext of these elements because intermediate nodes have no knowledge of 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 $\boldsymbol{O}(h \cdot log\ n)$ in terms of multiplications together. Thus, the computational complexity is $\boldsymbol{O}(h^2 \cdot log\ n)$ for a GEV and $\boldsymbol{O}(h^3 \cdot log\ n)$ for a GEM with h GEVs in terms of multiplication.

Sink Decoding Overhead: After receiving an encoded message, a sink can decrypt the elements in the GEV. According to the Paillier cryptosystem, decrypting an element requires 1 exponentiation, 1 multiplication, and 1 division operation. Therefore, the computational complexity of decrypting a GEV is $\boldsymbol{O}(h \cdot log\ n)$ in terms of multiplication operations. Thus, for a whole GEM with h GEVs, the computational overhead is $\boldsymbol{O}(h^2 \cdot log\ n)$ in terms of multiplication.

C. Communication Overhead

Let h messages be generated, and each message is of length l bits. For source encoding, each message is prefixed with h codewords from a ring of size n. Considering the ciphertext expansion of the Paillier cryptosystem, we can calculate the communication overhead as $2h \cdot log\ n/l$.

D. Performance Optimization

As described in the previous subsections, the invertible probability and computational overhead of the proposed scheme are $1 - t(p^{-1} + q^{-1})$ and $\boldsymbol{O}(h^3 \cdot log\ n)$, respectively. Thus, the statistical computational overhead for a GEM can be expressed in terms of multiplications as follows:

$$CO = \frac{h^3 \cdot log\ n}{1 - t(p^{-1} + q^{-1})} \tag{15}$$

From Eq. (15), we can see that the computational overhead of the proposed scheme is a monotonically increasing function of h, i.e., the length of a GEV, for any given n and t. As discussed in Section IV, the security of the proposed scheme is also monotonically increasing with the increase of h. Thus, a tradeoff between the security and the computational overhead should be considered in practical deployment. A typical way to deal with this tradeoff is to set the security requirements first and then choose the minimum h to meet the requirements. In this way, the minimum computational overhead can be achieved.

On the other hand, noticing that $n = pq$ and $|p| \approx |q|$, we can approximate Eq. (15) for any given h and formulate it as the following optimization problem:

$$\begin{array}{rl} Minimize & g(n,t) = \frac{log\ n}{1-2t/\sqrt{n}} \\ subject\ to: & n \geqslant 4,\ n\ is\ an\ integer, \\ where & t \geqslant 1,\ t\ is\ an\ integer. \end{array} \tag{16}$$

By solving the ordinary differential equation $\frac{\partial g}{\partial n} = 0$, we have: $n_1 = 4t^2 LambertW^2(-(2e \cdot t)^{-1})$ and $n_2 = 4t^2 LambertW^2((2e \cdot t)^{-1})$. Since the *Lambert-W* function has infinite branches in the complex plane, we only consider the branches which have real-valued solutions with real arguments. In addition, since $t \geqslant 1$(t is the total coding time) and the *Lambert-W* function is single-valued in the real plane for a positive argument, we can determine that $n_2 < 4t^2 \cdot ((2e \cdot t)^{-1})^2 = e^{-2}$, which is in conflict with the condition $n \geqslant 4$. Thus, n_2 can be excluded for further consideration.

n_1 is double-valued in the real plane since the argument of the function $LambertW(x)$, $x = -(2e \cdot t)^{-1}$, is in the region of $(-1/e, 0)$. We denote the double-valued results as

$$n_1(k,t) = 4t^2 LambertW^2(k, \frac{-1}{2e \cdot t}),\ k = -1, 0, \tag{17}$$

where k can be any integer in the complex plane. For a real-valued solution of n_1, k can only be 0 and -1. Similarly, we can determine that $n_1(0,t) < n_1(0,t)|_{t=1} = 0.215$ and this

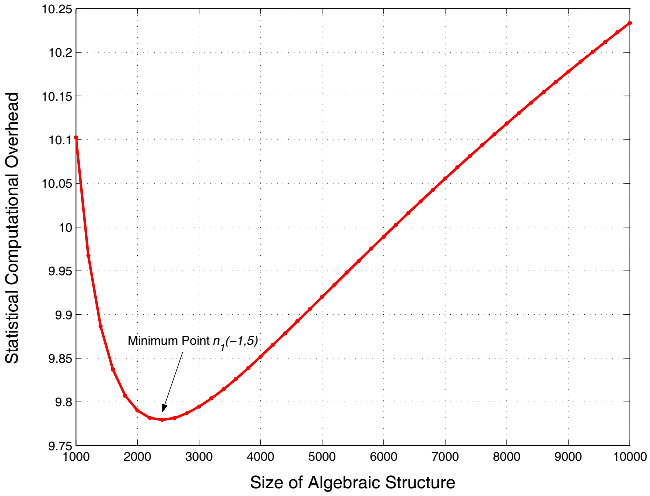


Fig. 9. Computational overhead vs. size of algebraic structure (t=5).

principal value is not in accord with the prescribed condition $n \geqslant 4$. Finally, we can determine that the result $n_1(-1,t)$ is the point where the objective function $g(n,t)$ achieves its minimum for any given parameter t. For example, given $t = 5$, we can get three real-valued results as follows: $n_1(-1,5) = 2390.936$, $n_1(0,5) = 0.146$, and $n_2(0,5) = 0.126$, where only $n_1(-1,5)$ meets the prescribed condition $n \geqslant 4$. Fig. 9 shows the analytical results of the statistical computational overhead versus the size of algebraic structure. After obtaining the minimum point, we can find the closest positive integer n which is the product of two primes p and q, i.e., $n = pq$. The integer n can then be substituted into Eq. (15) to achieve the minimum computational overhead.

VI. Related Work

Several privacy-preserving schemes have been proposed, and they can be classified into three categories: proxy-based, mix-based, and onion-based. Proxy-based schemes include Crowds [3] and Hordes [4]. The common characteristic of these schemes is to employ one or more network nodes to issue service requests on behalf of the originator. In Crowds, for example, servers and even crowd members cannot distinguish the originator of a service request, since it is equally likely originating from any member of the crowd. Chaum's mix based schemes include MorphMix [5] and Mixminion [6]. These schemes commonly apply techniques such as shaping, which divides messages into a number of fixed-sized chunks, and mixing, which caches incoming messages and then forwards them in a randomized order. These two techniques can be used to prevent attacks such as size correlation and time correlation. Onion-based schemes include Onion Routing [7] and Onion Ring [8]. The common feature of these schemes is to chain onion routers together to forward messages hop by hop to the intended recipient. Therefore, every intermediate onion router knows only about the router directly in front of and behind itself, respectively, which can protect user privacy if one or even several intermediate onion routers are compromised.

Network coding has privacy-preserving features, such as shaping, buffering, and mixing. However, network coding suffers from two primary types of attacks, *pollution attacks* [29] and *entropy attacks* [30]. *Pollution attacks* can be launched by untrusted nodes or adversaries through injecting faked messages or modifying authentic messages, which are fatal to the whole network due to the rapid propagation of pollution. In *entropy attacks*, adversaries forge non-innovative packets that are linear combinations of "stale" ones, thus reducing the overall network throughput. The vulnerabilities of inter/intra-flow network coding frameworks are identified, and general guidelines are provided to achieve the security objectives of network coding systems in [31].

To secure network coding, some solutions have been proposed and they can be classified into two categories according to different theoretical bases. Information-theory based schemes [15] can detect or filter out polluted messages at sinks. A new network coding security model and a construction of secure linear network codes are proposed in [32]. Distributed polynomial-time rate-optimal network codes [33] are introduced against Byzantine adversaries with different attacking capabilities. Cryptography-based solutions include homomorphic hashing [30], homomorphic signatures [29], and secure random checksum [30]. These solutions either require an extra secure channel [30], or incur high computation overhead [29]. Another secure network coding scheme based on hash functions are proposed in [34].

In summary, existing studies on secure network coding mainly focus on detecting or filtering out polluted messages [29]. Little attention has been paid to the privacy issues, especially to protect the encoded messages from tracking or traffic analysis.

VII. Conclusions

In this paper, we have proposed an efficient network coding based privacy-preserving scheme against traffic analysis and flow tracing in multi-hop wireless networks. With the lightweight homomorphic encryption on Global Encoding Vectors (GEVs), the proposed scheme offers two significant privacy-preserving features, packet flow untraceability and message content confidentiality, which can efficiently thwart traffic analysis/flow tracing attacks. Moreover, with homomorphic encryption, the proposed scheme keeps the essence of random linear network coding, and each sink can recover the source messages by inverting the GEVs with a very high probability. The quantitative analysis and simulative evaluation on privacy enhancement and computational overhead demonstrate the effectiveness and efficiency of the proposed scheme. In our future work, we will further improve the privacy preservation of the proposed scheme to achieve event source unobservability by employing dummy messages.

Appendix A
Proof of Theorem 1

Proof: From the theory of modulo congruence, for $n > 1$, the mapping $\mathcal{F} : \mathbb{Z} \mapsto \mathbb{Z}_n$ is a homomorphic mapping in terms of the addition, subtraction, and multiplication operations. Therefore, for each solution to $|A|x_i = |\tilde{A}_i|\ (mod\ n)$, there must be one or more solutions to $|A|x_i = |\tilde{A}_i| + k \cdot n\ (k \in \mathbb{Z})$.

According to the theory of linear algebra, the necessary and sufficient conditions for $|A|x_i = |\tilde{A}_i| + k \cdot n\ (k \in \mathbb{Z})$ to have a unique solution is $|A| \neq 0$. Thus, the necessary condition for a system of linear congruence equations to have a unique solution is $|A| \neq 0$.

Appendix B
Proof of Theorem 2

Proof: According to **Theorem 1**, $|A| \neq 0$ is a necessary condition for a system of linear congruence equations to have a unique solution. Therefore, the original system of linear congruence equations can be transformed to $|A|x_i = |\tilde{A}_i| \ (mod\ n)$ $(i = 1, 2, \cdots, h)$ by only applying addition and multiplication operations, which are preserved by the homomorphic mapping $\mathcal{F} : \mathbb{Z} \mapsto \mathbb{Z}_n$.

In equations $|A|x_i = |\tilde{A}_i| \ (mod\ n)$, variables x_i $(i = 1, 2, \cdots, h)$ can be solved independently by employing the theory of linear congruence equations. A linear congruence equation $ax = b \ (mod\ n)$ is solvable if and only if the congruence $b = 0 \ (mod\ d)$ with $d = gcd(a, n)$ is solvable, where $gcd(a, n)$ is the greatest common divisor of a and n. Let one solution of the linear congruence equation be $x_0 < n/d$. Then, the solutions are $x = x_0, \ x_0 + n/d, \ \cdots, \ x_0 + (d-1)n/d$. If $d = 1$, there is only one unique solution which is less than n. According to the theory of linear congruence equations, if $|\tilde{A}_i| \equiv 0 \ (mod\ d)$ $(i = 1, 2, \cdots, h)$, every equation $|A|x_i = |\tilde{A}_i| \ (mod\ n)$ has d solutions. The solutions to a system of linear congruence equations are the combinations of these independent solutions. The combination number is d^h.

Appendix C
Proof of Corollary 1

Proof: From **Theorem 2**, if $d = gcd(|A|, n) = 1$, a system of linear congruence equations has $d^h = 1$ solution, which is the unique solution to the system. In addition, according to the congruence theory, when $gcd(|A|, n) = 1$, the modular inverse of the integer $|A|$, which is denoted as $|A|^{-1}$, can be calculated using the extended Euclidean algorithm. The result satisfies the following equation: $|A|^{-1}|A| = 1 \ (mod\ n)$. The unique solution to a system of linear congruence equations is $x_i = |A|^{-1}|\tilde{A}_i| \ (mod\ n)$ $(i = 1, 2, \cdots, h)$. The solution can also be expressed in a matrix form as $|A|^{-1}|A|X = X = |A|^{-1}A^*Y \ (mod\ n)$.

Appendix D
Proof of Theorem 3

Proof: The invertibility factor of h LEVs depends only on the linear dependence of the h LEVs themselves. Firstly, the elements in the first LEV can be any combinations except for all zeros. Therefore, the invertibility factor of the first LEV is $1 - q^{-h}$. The second LEV should be linearly independent on the first one, where the all-zero vector is also excluded; thus, the invertibility factor of the second LEV is $1 - q/q^h = 1 - q^{1-h}$. The third LEV should be linearly independent on the former two; thus, the invertibility factor is $1 - q^2/q^h = 1 - q^{2-h}$. Similarly, for the remaining LEVs, the invertibility factors are $1 - q^{3-h}, \ 1 - q^{4-h}, \ \cdots, \ 1 - q^{-1}$, respectively. Therefore, the overall invertibility factor of the whole LEM is the product of these individual factors: $s_q = \prod_{i=1}^{h}(1 - q^{-i})$.

Appendix E
Proof of Theorem 4

Proof: The problem can be decomposed into two separate sub-problems in terms of the prime numbers p and q, respectively. As for the sub-problem in terms of p, there is a mapping from the original problem to the problem modulo p. According to **Theorem 3**, the invertibility factor is $s_p = \prod_{i=1}^{h}(1 - p^{-i})$. Similarly, the invertibility factor of the sub-problem in terms of q is $s_q = \prod_{i=1}^{h}(1 - q^{-i})$. The above two sub-problems have overlap, which occurs at these points where the number is congruent to zero modulo n. The invertibility factor related to the overlap area is $s_n = \prod_{i=1}^{h}(1 - n^{-i})$. According to the union principle of the set theory, the overall invertibility factor is: $1 - ((1 - s_p) + (1 - s_q) - (1 - s_n)) = s_p + s_q - s_n$.

References

[1] X. Lin, R. Lu, H. Zhu, P.-H. Ho, X. Shen, and Z. Cao, "ASRPAKE: an anonymous secure routing protocol with authenticated key exchange for wireless ad hoc networks," in *Proc. IEEE ICC'07*, pp. 1247-1253, 2007.

[2] M. Shao, Y. Yang, S. Zhu, and G. Cao, "Towards statistically strong source anonymity for sensor networks," in *Proc. IEEE INFOCOM'08*, pp. 51-55, 2008.

[3] M. K. Reiter and A. D. Rubin, "Crowds: anonymity for web transactions," *ACM Trans. Inf. and System Security*, vol. 1, no. 1, pp. 66-92, Nov. 1998.

[4] C. Shields and B. N. Levine, "A protocol for anonymous communication over the Internet," in *Proc. ACM CCS'00*, pp. 33-42, 2000.

[5] M. Rennhard and B. Plattner, "Introducing MorphMix: peer-to-peer based anonymous Internet usage with collusion detection," in *Proc. ACM Workshop on Privacy in the Electronic Society*, pp. 91-102, 2002.

[6] G. Danezis, R. Dingledine, and N. Mathewson, "Mixminion: design of a type III anonymous remailer protocol," in *Proc. IEEE Symposium on Security and Privacy*, pp. 2-15, May 2003.

[7] D. Goldschlag, M. Reed, and P. Syverson, "Onion routing for anonymous and private Internet connections," *Commun. ACM*, vol. 42, no. 2, pp. 39-41, Feb. 1999.

[8] X. Wu and N. Li, "Achieving privacy in mesh networks," in *Proc. ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN'06)*, pp. 13-22, 2006.

[9] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inf. Theory*, vol. 46, no. 4, pp. 1204-1216, July 2000.

[10] T. Ho, M. Medard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Trans. Inf. Theory*, vol. 52, no. 10, pp. 4413-4430, 2006.

[11] S.-Y. R. Li, R. W. Yeung, and C. Ning, "Linear network coding," *IEEE Trans. Inf. Theory*, vol. 49, no. 2, pp. 371-381, 2003.

[12] P. Sanders, S. Egner, and L. Tolhuizen, "Polynomial time algorithms for network information flow," in *Proc. 15th ACM Symposium on Parallel Algorithms and Architectures (SPAA'03)*, pp. 286-294, 2003.

[13] M. Wang and B. Li, "Network coding in live peer-to-peer streaming," *IEEE Trans. Multimedia*, vol. 9, no. 8, pp. 1554-1567, 2007.

[14] E. Ayday, F. Delgosha, and F. Fekri, "Location-aware security services for wireless sensor networks using network coding," in *Proc. IEEE INFOCOM '07*, pp. 1226-1234, 2007.

[15] K. Han, T. Ho, R. Koetter, M. Medard, and F. Zhao, "On network coding for security," in *Proc. IEEE MILCOM '07*, pp. 1-6, 2007.

[16] Z. Li, B. Li, and L. C. Lau, "On achieving maximum multicast throughput in undirected networks," *IEEE Trans. Inf. Theory*, vol. 52, no. 6, pp. 2467-2485, June 2006.

[17] Y. Wu, P. A. Chou, and S.-Y. Kung, "Minimum-energy multicast in mobile ad hoc networks using network coding," *IEEE Trans. Commun.*, vol. 53, no. 11, pp. 1906-1918, Nov. 2005.

[18] P. A. Chou and Y. Wu, "Network coding for the Internet and wireless networks," *IEEE Signal Process. Mag.*, vol. 24, no. 5, pp. 77-85, Sep. 2007.

[19] Q. Wang, K. Ren, W. Lou, and Y. Zhang, "Dependable and secure sensor data storage with dynamic integrity assurance," in *Proc. IEEE INFOCOM'09*, Rio de Janeiro, Brazil, Apr. 2009.

[20] M. Riemensberger, Y. E. Sagduyu, M. L. Honig, and W. Utschick, "Training overhead for decoding random linear network codes in wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 5, pp. 729-737, 2009.

[21] Z. Zhang, "Linear network error correction codes in packet networks," *IEEE Trans. Inf. Theory*, vol. 54, no. 1, pp. 209-218, 2008.

[22] R. Koetter and F. R. Kschischang, "Coding for errors and erasures in random network coding," *IEEE Trans. Inf. Theory*, vol. 54, no. 8, pp. 3579-3591, 2008.

[23] Y. Fan, Y. Jiang, H. Zhu, and X. Shen, "An efficient privacy-preserving scheme against traffic analysis attacks in network coding," in *Proc. IEEE INFOCOM'09*, Rio de Janeiro, Brazil, Apr. 2009.

[24] J. Benaloh, "Dense probabilistic encryption," in *Proc. Workshop on Selected Areas in Cryptography*, pp. 120-128, 1994.

[25] P. Paillier, "Public-key cryptosystems based on composite degree residuocity classes," in *Proc. EUROCRYPT'99*, vol. 1592, pp. 223-238, 1999.

[26] L. Lima, J. P. Vilela, J. Barros, and M. Medard, "An information-theoretic cryptanalysis of network coding–is protecting the code enough?" in *Proc. ISITA'08*, pp. 1-6, 2008.

[27] P. Venkitasubramaniam and L. Tong, "Anonymous networking with minimum latency in multihop networks," in *Proc. IEEE Symposium on Security and Privacy*, pp. 18-32, 2008.

[28] Y. Challal and H. Seba, "Group key management protocols: a novel taxonomy," *International J. Inf. Technol.*, vol. 2, pp. 105-119, 2005.

[29] Z. Yu, Y. Wei, B. Ramkumar, and Y. Guan, "An efficient signature-based scheme for securing network coding against pollution attacks," in *Proc. IEEE INFOCOM*, 2008.

[30] C. Gkantsidis and P. R. Rodriguez, "Cooperative security for network coding file distribution," in *Proc. IEEE INFOCOM'06*, pp. 1-13, 2006.

[31] J. Dong, R. Curtmola, and C. Nita-Rotaru, "Secure network coding for wireless mesh networks: threats, challenges, and directions," *Computer Commun. (Elsevier)*, vol. 32, no. 17, pp. 1790-1801, Nov. 2009.

[32] N. Cai and R. W. Yeung, "Secure network coding," in *Proc. IEEE ISIT'02*, 2002.

[33] S. Jaggi, M. Langberg, S. Katti, T. Ho, D. Katabi, M. Medard, and M. Effros, "Resilient network coding in the presence of Byzantine adversaries," *IEEE Trans. Inf. Theory*, vol. 54, no. 6, pp. 2596-2603, 2008.

[34] M. Adeli and H. Liu, "Secure network coding with minimum overhead based on hash functions," *IEEE Commun. Lett.*, vol. 13, no. 12, pp. 956-958, 2009.

Yanfei Fan received the B.Sc. degree in computer science from Beijing University of Posts and Telecommunications, Beijing, China, in 2002, the M.Sc. degree in computer science from Tsinghua University, Beijing, China, in 2005, and the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2010. His current research interests include wireless network security and network coding.

Yixin Jiang received the Ph.D. degree from the Department of Computer Science and Technology, Tsinghua University, Beijing, China, in 2006. He was a Post Doctoral Fellow with University of Waterloo, Waterloo, ON, Canada, from 2007 to 2009.

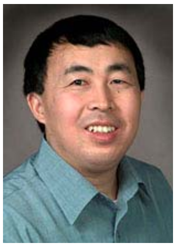
He is an Associate Professor in Tsinghua University. He has served as a Technical Program Committee (TPC) member for main network conferences, such as IEEE/INFOCOM'2010, ICCCN'2009, GLOBECOM, ICC, WCNC, etc. His current research interests include trusted computing, cloud computing, security and privacy in wireless networks. He has received Excellent Backbone Talents Fund Award, Outstanding Doctoral Graduate Award, and Excellent Doctoral Thesis Award of Tsinghua University.

Haojin Zhu (M'09) received the B.Sc. degree in computer science from Wuhan University, Wuhan, China, in 2002, the M.Sc. degree in computer science from Shanghai Jiao Tong University, Shanghai, China, in 2005, and the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2009.

He is currently an Assistant Professor with Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China. His current research interests include wireless network security and distributed system security.

Dr. Zhu was a recipient of the Best Paper Award of IEEE ICC 2007 - Computer and Communications Security Symposium and Chinacom 2008 - Wireless Communication Symposium. He serves as the Technical Program Committee for international conferences such as IEEE/INFOCOM, ICCCN, Globecom, ICC, WCNC and etc.

Jiming Chen (M'08) received the Ph.D. degree in Control Science and Engineering from Zhejiang University in 2005. He was a visiting scholar at INRIA, NUS. He is an associate professor with Institute of Industrial Process Control, and the coordinator of group of Networked Sensing and Control in the State Key Laboratory of Industrial Control Technology at Zhejiang University, China. Currently he is a visiting researcher with the Centre for Wireless Communications, Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. His research interests include estimation and control over sensor network, sensor and actuator network, target tracking in sensor networks, optimization in mobile sensor network. He currently serves as an associate editor for the *International Journal of Communication System* (Wiley), *Ad Hoc & Sensor Wireless Networks* (oldcitypublishing), etc. He also serves as a guest editor for IEEE TRANSACTION ON AUTOMATIC CONTROL, *Wireless Communication and Mobile Computing* (Wiley), etc. He serves as a general symposia Co-Chair of ACM/IWCMC 2009, IWCMC 2010, WiCON 2010, MAC track Co-Chair of Chinacom 2010, Publicity Co-Chair and TPC for IEEE/ICDCS 2010, MASS 2010, INFOCOM 2011, etc.



Xuemin (Sherman) Shen received the B.Sc.(1982) degree from Dalian Maritime University (China) and the M.Sc. (1987) and Ph.D. degrees (1990) from Rutgers University, New Jersey (USA), all in electrical engineering.

He is a Professor and University Research Chair, Department of Electrical and Computer Engineering, University of Waterloo, Canada. Dr. Shen's research focuses on resource management in interconnected wireless/wired networks, UWB wireless communications networks, wireless network security, wireless body area networks and vehicular ad hoc and sensor networks. He is a coauthor of three books, and has published more than 400 papers and book chapters in wireless communications and networks, control and filtering.

Dr. Shen served as the Technical Program Committee Chair for IEEE VTC'10, the Symposia Chair for IEEE ICC'10, the Tutorial Chair for IEEE ICC'08, the Technical Program Committee Chair for IEEE Globecom'07, the General Co-Chair for Chinacom'07 and QShine'06, the Founding Chair for IEEE Communications Society Technical Committee on P2P Communications and Networking. He also served as a Founding Area Editor for IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS; Editor-in-Chief for *Peer-to-Peer Networking and Application*; Associate Editor for IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY; *Computer Networks*; and *ACM/Wireless Networks*, etc., and the Guest Editor for IEEE JSAC, *IEEE Wireless Communications*, *IEEE Communications Magazine*, and *ACM Mobile Networks and Applications*, etc. Dr. Shen received the Excellent Graduate Supervision Award in 2006, and the Outstanding Performance Award in 2004 and 2008 from the University of Waterloo, the Premier's Research Excellence Award (PREA) in 2003 from the Province of Ontario, Canada, and the Distinguished Performance Award in 2002 and 2007 from the Faculty of Engineering, University of Waterloo. Dr. Shen is a registered Professional Engineer of Ontario, Canada, an IEEE Fellow, and a Distinguished Lecturer of IEEE Communications Society.