OPEN ACCESS
ISSN: 2249-6645

International
Journal
Of Modern Engineering Research (IJMER)

Secure Network Coding In Wireless Sensor Networks

V.Narsing Rao[1], K.Vijaya Babu[2],Challa Mahesh Kumar[3]
[1]*Associate Professor, Department of CSE, CMR Engineering College, Hyderabad*
[2]*Asst. Professor, Department of CSE, CMR Engineering College, Hyderabad.*
[3]*Asst. Professor, Department of CSE, CMR Engineering College, Hyderabad.*

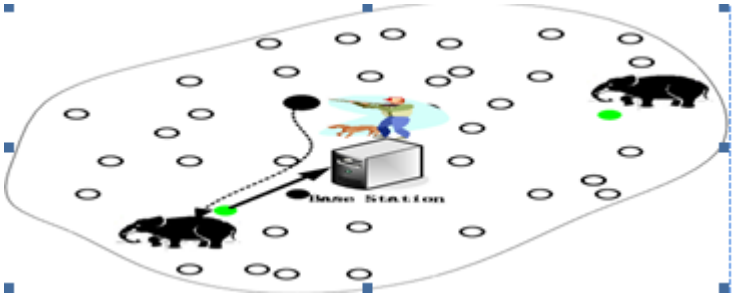
Abstract:*Wireless Sensor Network (WSN) consists of mostly tiny, resource-constraint, simple sensor nodes, which communicate wirelessly and form ad hoc networks in order to perform some specific operation. Due to distributed nature of these networks and their deployment in remote areas, these networks are vulnerable to numerous security threats that can adversely affect their proper functioning. However, their characteristics such as the broadcast nature of the medium, spatial diversity, and significant data redundancy, provide opportunities for new design principles to address these problems. There has been recent interest in employing network coding in wireless networks. This paper explores the case for network coding that offers packet flow un traceability and message content confidentiality against traffic analysis using WSN which gives better performance in WSN life span and nodes energy consumption. Theoretical analysis and simulative evaluation demonstrate the validity and efficiency of the proposed scheme.*

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

I. INTRODUCTION

Wireless sensor networks (WSNs) serve as a significant role to bridge the gap between the physical and logical worlds . Nodes in WSNs are tiny embedded devices which only own limited computing ability, data storage space, constrained battery energy and narrow wireless network band width. Among the most critical issues of WSNs is nodes' energy consumption in general. So, usually, in a WSNs application, to save the battery energy in each node, node can be in power saving model, or in sleeping mode which may lead intermitted network connection and long time delay of transmission even data transmission failure.. 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. 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.

WSNs can be regarded as a kind of MWNs; however, WSNs are also characterized by their unique features such as lack of Privacy threats in MWNs



end-to-end connection, fragmentation, and bundle accumulation. These unique features pose new challenges to the information security, especially privacy preservation, of WSNs. Privacy preservation is a new research topic in WSNs and has received little attention. The existing privacy-preserving technologies such as mix-net and

onion routing are not suitable for WSNs. Considering the unique characteristics of WSNs, we will carefully examine the existing privacy-preserving schemes and design new privacy-preserving schemes for WSNs.

As a kind of MWNs, WSNs characterize themselves with a series of unique features such as lack of end-to-end connection, fragmentation, and bundle accumulation. These three unique features pose great challenges to the information security of WSNs. For example, the lack of end-to-end connection and fragmentation features will severely degrade the data availability in WSNs, which is one of our future research topics. A good bundle accumulation mechanism may greatly enhance the data availability in WSNs. However, a single bundle accumulation mechanism may not able to achieve the desired data availability level. In this sense, a whole-set design of WSNs may be preferred for the enhancement of data availability. On the other hand, network coding has many desired features such as block scheduling easiness and multiple data delivery, which can be utilized for enhancing data availability in WSNs. In this paper, we focus on the privacy issue, i.e., how to prevent traffic analysis/flow tracing and achieve source anonymity in WSNs.

Among all privacy properties, source anonymity is of special interest in WSNs. 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 WSNs, 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, 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, as shown in Fig. 1(B).

It is very challenging to efficiently thwart traffic analysis/flow tracing attacks and provide privacy protection in WSNs. Existing privacy-preserving solutions, such as Onion-based schemes, 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. 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, data transmission in sensor networks, tactical communications in military networks, etc.

Compared with conventional packet forwarding technologies, network coding offers, by allowing and encouraging coding/mixing operations at intermediate forwarders, several significant advantages such as potential throughput improvement, transmission energy minimization, and delay reduction. In addition, network coding can work as erasure codes to enhance the dependability of a distributed data storage system.

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 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;

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.

II. PRELIMINARIES

A. Delayt Tolerant Networks

WSNs deal with communication in extreme and performance-challenged environments, where continuous end-to-end connectivity cannot be assumed. In a WSN, nodes use opportunistic connectivity over intermittent links for communication. Such opportunistic links are generally provided by mobile routers. They offer connectivity by acting as "data mules" to carry data to and from servers with continuous network connectivity (i.e., Internet access). There are many applications for WSNs. In developing regions, especially rural areas, they can be used to provide network access for education, health care or government services. They can also augment low bandwidth Internet connections to transfer large files at low cost, while using the Internet connection for control messages. WSNs are also applicable in vehicular ad-hocnetworks (VANETs) and undersea communication.

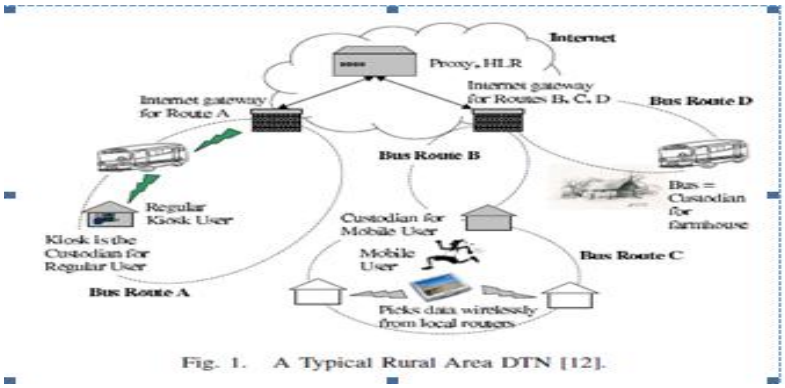


Fig. 1. A Typical Rural Area DTN [12].

Though WSNs arise in many situations and may take many forms, our terminology in this paper is slanted towards the particular example of rural area WSNs. The use of this concrete example aids exposition and provides motivation, but does not reduce the applicability of our work to other types of WSNs. Seth et al. [12] provide a detailed discussion of rural area WSNs. Figure 1 illustrates a typical rural area WSN. We now give a brief overview.

The approach is applicable to villages and rural areas with no Internet connectivity due to geographic or economic constraints.

There is an Internet connection available in a nearby town and a transport medium from the rural area to the town in the form of a vehicle, such as a bus or a car.

The terminal with Internet connectivity is called the gateway. A transport medium that carries data from the end users in a village to a gateway is called a mobile router.

There is also a special static router called a kiosk, which serves as a computing facility for WSN users. The kiosk also provides a persistent data transfer facility, so users do not have to wait for a mobile router to show up.

There are two types of end users, mobile users, who use their own personal devices to connect directly to routers (typically a kiosk), and kiosk users, who use a shared terminal at a kiosk. Our secure and anonymous communication architecture targets mainly mobile users. However, if a kiosk is trusted, our architecture provides equivalent security and anonymity to kiosk users. Achieving security and privacy in such disconnected network is a demanding task, but it is necessary in hostile environments with malicious attackers or even just passive listeners.

In rural area WSNs, security and privacy are necessary to effectively implement concepts like e-governance, citizen journalism, distance education (e.g., aAqua) and telemedicine. In a hostile environment, secure and anonymous WSN communication can provide an efficient way to let informers transfer information while hiding their identity from an enemy. Therefore, the utility of a WSN is greatly expanded when the WSN provides end-to-end security and privacy. The limitations of WSNs require the design of new security and privacy protocols for WSNs, which forms the basis for this work.

B. 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 *k*th 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 $t \in V$).

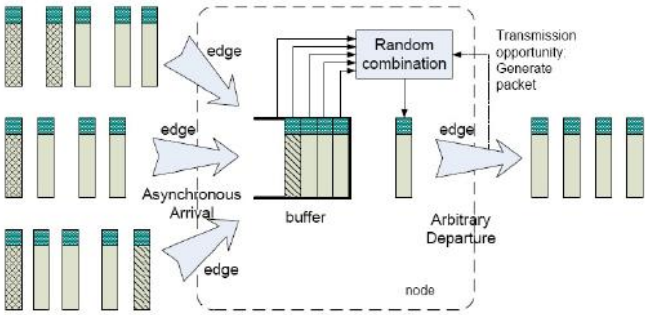


Fig. 2. Random coding (mixing) at intermediate nodes.

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 cipher text.

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 as the HEF to apply encryption to GEVs, since protecting GEVs is generally sufficient to ensure confidentiality network coded message content. 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 $E(m) = gm \cdot rn (mod\ n2)$, where *r* is a random factor.

$E(m)$ satisfies the homomorphic property:

$E(m1) \cdot E(m2) = gm1+m2 \cdot (r1r2)n (mod\ n2) = E(m1 + m2)$.

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.

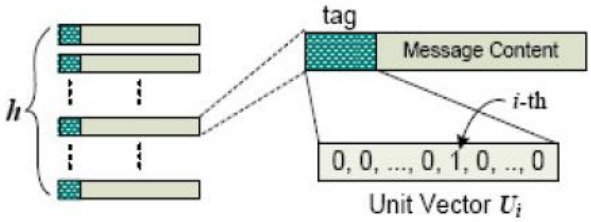


Fig. 5. Packet tagging before source encoding.

Source Encoding: Consider that a source has h messages, say $x1, \cdots, xh$, 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:

$$ci(e) = Eek(gi(e)), (1 \leqslant i \leqslant h)$$
$$c(e) = [c1(e), c2(e), \cdots, ch(e)]$$

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, $g(e) = \Sigma h\, i{=}1 \beta i(e) g(e'i)$. (5)

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

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

(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.

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 $O(h3+h \cdot l)$ 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 $O(Ch\, w(h3 + h \cdot l))$, 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

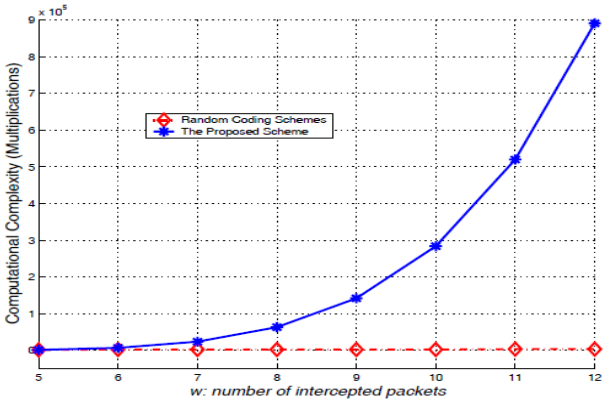


Fig. 6. Privacy enhancement in terms of the order of computational complexity (h=5, l=200).

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 cipher text 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).

IV. 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 hold.

Corollary 2: The invertibility factor sq 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 $\mathcal{O}(q^{-5})$. This corollary can be easily proven by expanding the multiplication of the polynomials. This corollary gives two

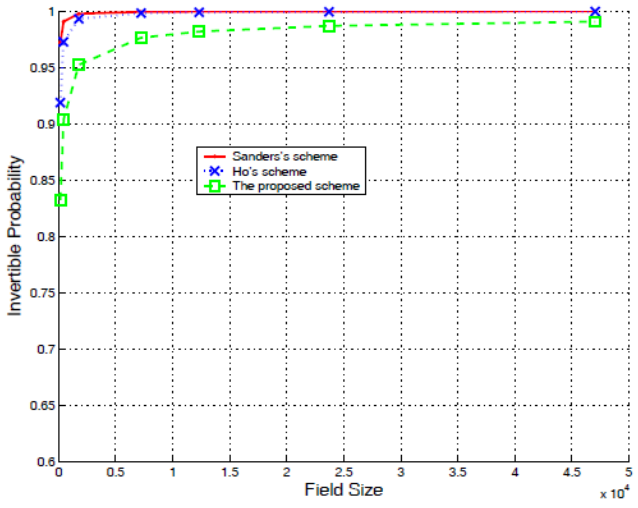


Fig. 7. Invertible probability vs. field size (theoretical analysis).

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 $O(q^{-5})$ is very small and can be omitted.

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

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 □(h2) 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 □(h2 · □□□ □) 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 □(h · □□□ □) in terms of multiplications together. Thus, the computational complexity is □(h2 · □□□ □) for a GEV and □(h3 · □□□ □) 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 □(h· □□□ □) in terms of multiplication operations. Thus, for a whole GEM with h GEVs, the computational overhead is □(h2 · □□□ □) in terms of multiplication.

C. Communication Overhead

Let h messages be generated, and each message is of length □ bits. For source encoding, each message is prefixed with h code words from a ring of size □. Considering the cipher text expansion of the Paillier cryptosystem, we can calculate the communication overhead as 2h · □□□ □/□.

V. 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/intraflow 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.

VI. CONCLUSIONS

In this paper, we have proposed an efficient network coding based privacy-preserving scheme against traffic analysis and flow tracing in delay tolerant networks. We explained Network Coding Based Privacy-Preserving Scheme For WSNs, the lightweight homomorphic encryption on Global Encoding Vectors (GEVs), and the threat models. 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 demonstrate the effectiveness and efficiency of the proposed scheme.

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.

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

[4]. 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.

[5]. 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.

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.

[6]. 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.

[7]. 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.

[8]. 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.

[9]. 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.

[10]. 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.

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

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.

Mr.V.Narsing Rao working as Assoc.Professor at CMR Engineering College. He is having 10 years of experience in teaching . he published 5 papers in various International & National Journals and his areas of interest are Data mining, Network Security and Image Processing.

Mr. K. Vijaya Babu Working as Assistant Professor at CMR Engineering College, Hyderabad, India and done M. Tech. (IT) from JNTUK Vizianagaram Campus. He is having 4 years of teaching experience and published one paper in International Journal. His areas of interest are Computer Networks, Information Security.

Mr.MAHESH KUMAR,Asst.Professor in Department Of Computer Science & Engineering at CMR Engineering College, Hyderabad,Telangana State, India. He is having 7 years of experience in teaching and his areas of interest are Network Security, Image Processing and cloud computing .