Enhanced Network coding based privacy preservation against traffic analysis in multi-hop wireless networks

SHANKAR MADABHAVI
Dept. of ECE
KLS GIT
BELGAUM, INDIA
MADABHAVI08@GMAIL.COM

KRISHNA HUKKERI
Dept. of ECE
KLS GIT
BELGAUM, INDIA
KPHUKKERI@GMAIL.COM

ABSTRACT

Due to the open wireless medium in the multi-hop wireless networks, attacks such as traffic analysis and flow tracing can be easily launched by a malicious adversary. 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 intractability and message content confidentiality, for efficiently preventing 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.. The algorithm will also incorporate the concept of "fake packets" to further confuse the adversary with false information about routing path. This enhancement will be measured for its increased level of security that it now provides.

***Keywords**: 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. 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

Network coding was first introduced by Ahlswede et al [1]. Subsequently, two key techniques, random coding and linear coding 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 [2], data transmission in sensor networks [3], tactical communications in military networks [4], etc. Compared with conventional packet forwarding technologies, network coding offers, by allowing and encouraging coding/mixing operations at intermediate forwarders [1], several significant advantages such as potential throughput improvement [5], transmission energy minimization [6], and delay reduction [7].

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 [8],.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 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) [9], [10], we propose an efficient privacy-preserving scheme for MWNs. Our objective is to achieve source anonymity by preventing traffic analysis. The proposed scheme offers the following attractive features:

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. 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, 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 [7]. 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 $t \in$). Let $h = (s, T)$ be the multicast capacity, and $x1, \cdot \cdot \cdot , xh$ 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., $y(e) = \Sigma e' \beta e' (e) y(e')$. The coefficient vector $(e) = [\beta e' (e)]$ 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 $x1, \cdot \cdot \cdot , xh$, i.e., $y(e) = \Sigma h\ i{=}1\ gi(e)xi$. The coefficients form a *Global Encoding Vector* (GEV) $\mathbf{g}(e) = [g1(e), \cdot \cdot \cdot , gh(e)]$, which can be computed recursively as $\mathbf{g}(e) = \Sigma e' \beta e' (e)\mathbf{g}(e')$, using the LEVs $\beta(e)$. Suppose that a sink $t \in T$ receives symbols $y(e1), \cdot \cdot \cdot , y(eh)$, which can be expressed in terms of the source symbols as where Gt is called *Global Encoding Matrix* (GEM) and the *i*th row of

1) Enhanced Privacy against traffic analysis and flow tracing: With the employment of HEFs, the confidentiality of GEVs is effectively guaranteed, asking 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 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 sink knows the decryption key.

Gt is the GEV associated with $y(ei)$. Sink t can recover the h source symbols by inverting Gt and then applying the inverse to $(e1), \cdot \cdot \cdot , y(eh$

II. The Proposed Privacy-Preserving scheme

Though providing an intrinsic mixing mechanism, the original network coding cannot provide privacy guarantee due to explicit GEVs, since 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 cipher text 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 $(m) = gm \cdot \ (mod\ n2)$, where r is a random factor. (m) satisfies the homomorphic property: $(m1) \cdot (m2) = gm1{+}m2 \cdot (r1r2)n(mod\ n2) = E(m1 + m2)$.. With HEFs, intermediate nodes are allowed to directly perform linear coding/mixing operations on the coded messages and encrypted tags. 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.Then, the encryption key is published and the decryption key is kept secret

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

$(e) = (g_i(e)), (1 \leqslant i \leqslant h)\ c(e)$

$= [c_1(e), c_2(e), \cdots, c_h(e)]$

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.

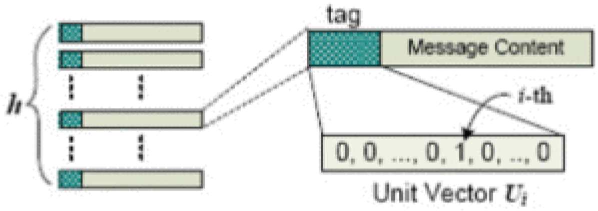

Fig1: *packet tagging.*

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 $[\beta 1, \cdot \cdot \cdot , \beta h]$ 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 cipher text format, and an intermediate node has no 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$(e) = \Sigma h\ i=1\ \beta i(e) g_i(e'i)$. 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 cipher text of the new GEVs for outgoing packets can be calculated as follows:$Eek(g_i(e)) = Eek(\Sigma h\ i=1\ \beta i(e) g_i(e'i)) = \Pi h\ i=1\ Eek(\beta i(e) g_i(e'i)) = \Pi h\ i=1\ E\beta i(e) ek\ (g_i(e'i))$ The cipher text of new GEVs can be computed from the cipher text of GEVs of incoming packets without the knowledge of the decryption key. Finally, the cipher text of a new GEV is prefixed to the corresponding message content to form a new outgoing packet, which is sent out to downstream nodes.

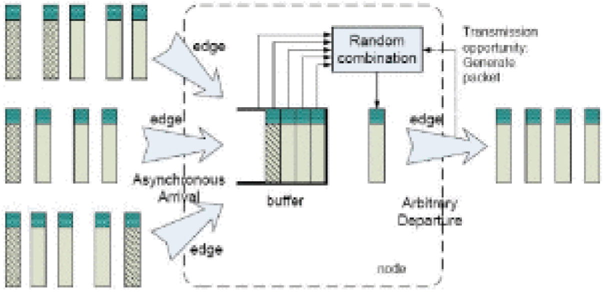

Fig2: *Random coding at intermediate nodes.*

Sink Decoding: After receiving a packet, the sink first decrypts the packet tag using the corresponding decryption key *dk*. $g_i(e) = Ddk(c_i(e))\ (1 \leqslant i \leqslant h)\ g_i(e) = [g1(e), g2(e), \cdot \cdot \cdot , gh(e)]$ 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 $G{-1} \cdot G = U\ G = [(e1), (e2), \cdot \cdot \cdot , g_i(eh)]T$ Finally, the sink can use the inverse to recover the original messages, shown as follows. IV.

III. 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 of network coding. Let the time length of buffering periods be Tb 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 Tb, 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 $Pr(1, \lambda \cdot Tb) = \lambda \cdot Tb \cdot e{-\lambda \cdot Tb}$. From Eq. it can be seen that the probability decreases exponentially with the time period Tb. On the other hand, the transmission delay increases linearly with the time period Tb. In practice, we can adaptively adjust parameter Tb 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 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 $(h3+h \cdot l)$ 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 $(Ch(h3 + h \cdot l))$, which increases exponentially. It can be seen that, compared with the previous network coding schemes, the proposed scheme significantly enhances 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).

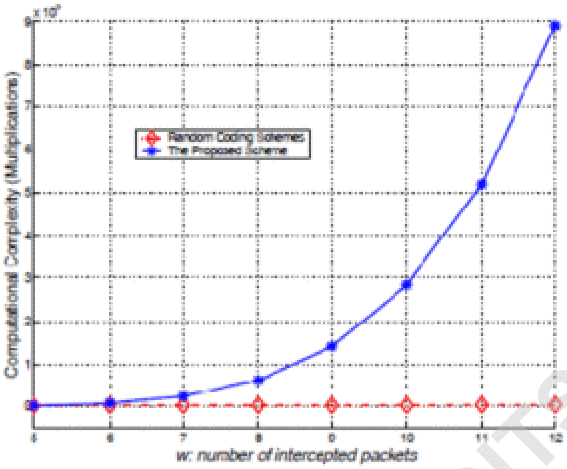

Fig3: *Privacy enhancement in terms of the order of computational complexity*

V. 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 intractability and message content confidentiality, which can efficiently prevent 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.

VI. REFERENCES

[1] 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.

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

[3] 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.

[4] 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.

[5] 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

[6] 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.

[7] 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.

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

[9] 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.

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