ISSN(Online):2320-9801
ISSN(Print): 2320- 9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Special Issue 3, July 2014

Achieving Maximum Multicast Throughput in Secured Multi-Hop Wireless Networks

P.Gnana prakash[1], Dr. P.Rajkumar[2]

[1] Dept of Computer Science and Engineering, Info Institute of Engineering, Coimbatore. India.

[2]AP, Dept of Computer Science and Engineering, Info Institute of Engineering, Coimbatore. India.

ABSTRACT: The multi-hop wireless networks always having security problems, the network traffic causes the attacks in inside attackers and outside attackers, to avoid those attacks by using navel network coding. Coding and mixing operation was encouraged in intermediate nodes. Data splitting and transmitting is done in network coding algorithm. The proposed scheme provides the packet flow intractability and message content confidentiality is ensured by threshold secret sharing algorithm.

KEYWORDS: Network coding, Traffic analysis, Threshold secret sharing, TTL (time to live).

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.

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.

In this paper, we seek to bring new insights and efficient solutions to the problem of maximizing information flow rates (or *throughput*) in undirected data networks. We first illustrate the power of *network coding* with respect to achieving maximum throughput. Although previous directions of computing the maximum multicast rates involve solving NP-complete problems, the maximum multicast rates and the corresponding optimal multicast strategy can indeed be computed efficiently in polynomial time, with the unique incurable property ofinformation flows considered. We provide a natural linear programming formulation of the maximum throughput problem, with a polynomial number of variables and constraints. By applying relaxation on the primal linear program (LP), we derive a necessary and sufficient condition for multicast rate feasibility in undirected networks, from a distance labelling perspective.

II. PRELIMINARIES

A. Navel network coding

The 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 takinga random combination of packets in the current buffer.

ISSN(Online):2320-9801
ISSN(Print): 2320- 9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Special Issue 3, July 2014

An overview of network coding and possible applications has been given. 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 carryone 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 $T\subseteq$ (or one single sink $t\in$). Let 836 IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, VOL. 10, NO. 3, MARCH 2011 Fig. 2. Random coding (mixing) at intermediate nodes. $h= Mi(s, T)$ be the multicast capacity, and $x1$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 (□) = [□$\mathit{5}'$ (□)] is called *Local Encoding Vector* (LEV). By induction, the symbol (□) on any edge □$\in$ □can be computed as a linear combination of the source symbols □1, i.e., (□) = Σ □=1 (□)□$\mathit{5}$.

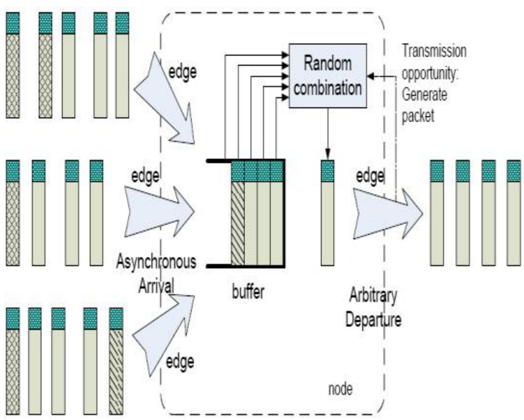

Fig.1. Random coding (mixing) at intermediate nodes.

The coefficients form a *Global Encoding Vector* (GEV) **g**(□) = [□1(□), □□(□)], which can be computed recursively as **g**(□) = Σ □′ □$\mathit{5}'$ (□)**g**(□′), using the LEVs □(□). Suppose that a sink □$\in$ □receives symbols □(□1),□(□), 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 $Gt$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),(e).

In general, each packet can be considered as a vector of symbols **y**(e) = [y1(e), yM(e)]. By likewise grouping the source symbols into packets **x**i= [xi,1,xi,M], the above algebraic relationships carry over to packets. To facilitate the decoding at the sinks, each message should be tagged with its GEV **g**(e), which can be easily achieved by prefixing the *i*th source packet **x**$i$with the *i*th unit vector **u**i. Then, each packetis automatically tagged with the corresponding GEV, since [g(e),y(e)] = Σ e' $\beta e'$ (e)[g(e'),y(e')] = Σxi=1 gi(e)[ui,xi]. The benefit of tags is that the GEVs can be found within the packets themselves, so that the sinks can compute Gtwithout knowing the network topology or packet-forwarding paths. Nor is a side channel required for the communication of Gt. 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

HomomorphicEncryption Functions (HEFs) have the property of homomorphism, which means operations on plaintext Fig. 3. Attack model: (a) outside attacker; (b) inside attacker. can be performed by operating on corresponding

ISSN(Online):2320-9801
ISSN(Print): 2320- 9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Special Issue 3, July 2014

cipher text. If *E* is a HEF, (*x*+ *y*) can be computed from (*x*) and (*y*) without knowing the corresponding plaintext *x* and *y*. To be applicable in the proposed scheme, a HEF *E* needs to satisfy the following properties:

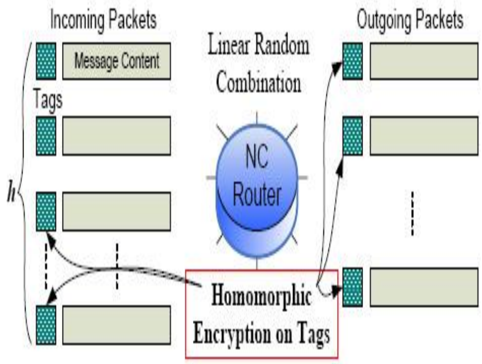

Fig2. Homomorphic encryption on packet tags.

1) **Additively**: Given the cipher text (*x*) and (*y*), there exists a computationally efficient algorithm *A*(E(x),E(y)) such that *E*(*x*+ *y*) = *Add*(*E*(*x*),*E*(*y*)). 2) **Scalar Multiplicatively**: Given (*x*) and a scalar *t*, there exists a computationally efficient algorithm*Mul*(*x*,y) such that *E*(*x*,*t*) = *Mul*(*E*(*x*), *t*). Actually, the scalar multiplicatively can be deduced from the additively, since (*x*,) = (Σ *ti*=1 *x*).

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., (*x*1 + *x*2) = (*x*1),(*x*2). Further, the following two equations can be easily derived: (,) = *E*(*x*) *E*(Σ *ixi*,*ti*) = Π *iEti*(*xi*) . (3)

C. Threshold secret sharing algorithm

Since the previous SSS has been defined on smaller fields, prime numbers or finite fields,*GF* , in our paper, we use the fast algorithm of Discrete Fourier Transform (DFT), which is originally used to transfer from one domain to another. FFT is used heavily in signal and image digital processing, forensic science, interpolation and decimation, linear estimation,pattern recognition, and many other applications [22,23]. In our paper we use it for secretdistribution and sharing.

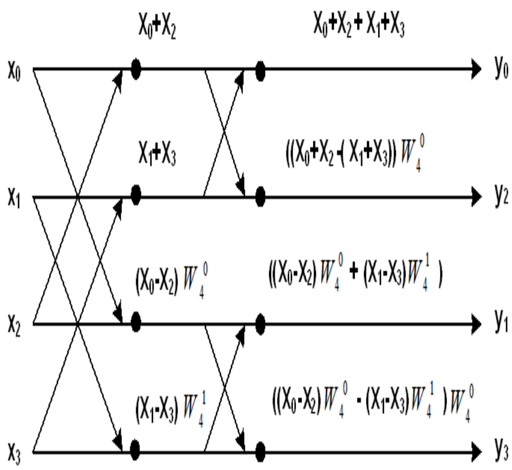

Fig. 3: Generates linear system of equations

ISSN(Online):2320-9801
ISSN(Print): 2320- 9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Special Issue 3, July 2014

Where 2 cos(2) sin(2) *j w e j* □ □ □ □ □ □ □ □ □ □ □ □ □ □ isconstant, *i*and *k*, □ □ .From Fig. 1. Weremark that each output
I y depends on all inputs1 2 *n*(*x* , *x* , ..., *x*) . So, at least half of the pairs*i k*(*x* , *y*) , where *k*, *i* □ {0,1, ...,*n*}, are mandatory to solve the corresponding equationsto find the remaining pairs and *k* is the bitreversal of *i* , for example (1,4)=(001,100),(3,6)=(011, 110) .We don't need to build the linear system ofequations as Blakely's scheme does. Theequations are generated by expanding Eq. 1automatically.There are some mathematical programs thatoffer functions to solve a set of linear equationsdirectly such as Mathematica and MATLAB.
Let *n* is the number of shares. Then *n*complex numbers should be chosen randomly.

D.TTL Field

Time to live (**TTL**) or **hop limit** is a mechanism that limits the lifespan or lifetime of data in a computer or network. TTL may be implemented as a counter or timestamp attached to or embedded in the data. Once the prescribed event count or timespan has elapsed, data is discarded. In computer networking, TTL prevents a data packet from circulating indefinitely. In computingapplications, TTL is used to improve performance of caching or to improve privacy.

Ip packets: Under the Internet Protocol, TTL is an 8-bit field. In the IPv4 header, TTL is the 9th octet of 20. In the IPv6 header, it is the 8th octet of 40. The maximum TTL value is 255, the maximum value of a single octet.

In theory, under IPv4, time to live is measured in seconds, although every host that passes the datagram must reduce the TTL by at least one unit. In practice, the TTL field is reduced by one on every hop. To reflect this practice, the field is renamed *hop limit* in IPv6.

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 [18] 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: $Pr(1, \lambda, Tb) = \lambda Tbe{-}\lambda Tb$. (4) From Eq. (4), 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 Tbaccording 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 (h3+hl) 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(Cw(h3 + hl)), which increases exponentially with w, 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. Fromperformance 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

ISSN(Online):2320-9801
ISSN(Print): 2320- 9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Special Issue 3, July 2014

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. CONCLUSION

In this paper, we have proposed an efficient navel network coding based Achieving maximum multicast throughput and flow tracing in multi-hop wireless networks With the lightweight homomorphic encryption. The proposed scheme offers two significant privacypreserving features, packet flow intractability and message content confidentiality, which can efficiently thwart traffic analysis/flow tracing attacks. The threshold secret sharing Algorithm provides such futures. The quantitative analysis and simulative evaluation on privacy enhancement and computational overhead demonstrate the effectiveness and efficiency of the proposed scheme. In ourfuture work, we will further improve the privacy preservation of the proposed scheme to achieve event source unobservability by employing dummy messages.

REFERENCES

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A Survey on Sensor Networks," IEEE Comm. Magazine, vol. 40, no. 8, pp. 102-114, Aug. 2002.

[2] C.L. Barrett, S.J. Eidenbenz, L. Kroc, M. Marathe, and J.P. Smith, "Parametric Probabilistic Sensor Network Routing," Proc. ACM Int'l Conf. Wireless Sensor Networks and Applications (WSNA), pp. 122-131, 2003.

[3] M. Burmester and T.V. Le, "Secure Multipath Communication in Mobile Ad Hoc Networks," Proc. Int'l Conf. Information Technology: Coding and Computing, pp. 405-409, 2004.

[4] T. Claveirole, M.D. de Amorim, M. Abdalla, and Y. Viniotis, "Securing Wireless Sensor Networks Against Aggregator Compromises," IEEE Comm. Magazine, vol. 46, no. 4, pp. 134-141, Apr. 2008.

[5] D.B. Johnson, D.A. Maltz, and J. Broch, "DSR: The Dynamic Source Routing Protocol for Multi-hop Wireless Ad Hoc Networks," Ad Hoc Networking, C.E. Perkins, ed., pp. 139-172, Addison-Wesley, 2001.

[6] Y. Wu, P. A. Chou, and S.-Y. Kung, "Minimum-energy multicast inmobile 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] 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.

[9] 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.

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

[11] 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.

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

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

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

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

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

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

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