IOSR Journal of Electronics and Communication Engineering (IOSR-JECE)
e-ISSN: 2278-2834,p- ISSN: 2278-8735.Volume 10, Issue 1, Ver. II (Jan - Feb. 2015), PP 09-13
www.iosrjournals.org

Preventing Malicious Node and Provide Secure Routing In Manet

R. Gayathri[1], J.Maria Sofi Anusuya[2]

P.G. Student, Department of Electronics and communication Engineering, Dhanalakshmi srinivasan Engineering College, perambalur, India[1]

Associate Professor, Department of Electronics and communication Engineering, Dhanalakshmi srinivasan Engineering College, perambalur, India[2]

Abstract: *A mobile ad-hoc network (MANET) is an autonomous system of mobile nodes connected by wireless links. Mobile Ad-hoc Networks (MANET) are characterized by open, distributed and dynamic architectures, built on top of the shared wireless medium; all features contribute to make MANET very vulnerable to attacks at any layer of the Internet model. We consider the following problem: nodes in a MANET must separated data packet using rateless codes but some nodes are assumed to be malicious, i.e., before transmitting a coded packet they may modify its payload. Nodes receiving corrupted coded packets are prevented from correctly decoding the original chunk. We propose SIEVE, a fully distributed technique to identify malicious nodes. SIEVE is based on special messages called checks that nodes periodically transmit. A check contains the list of nodes identifiers that provided coded packets of a chunk as well as a flag to signal if the chunk has been corrupted. SIEVE operates on top of an otherwise reliable architecture and it is based on the construction of a factor graph obtained from the collected checks on which an incremental belief propagation algorithm is run to compute the probability of a node being malicious. We show that SIEVE is very accurate and robustness under several attack scenarios and deceiving actions. We also focused on decentralized networks, called MANETs, where communication between mobile nodes is purely ad-hoc based, exploit- ing rateless coding to minimize data loss due to transmission unreliability, and detecting malicious nodes sending corrupted packets to detect and prevent problem in a strongly distributed environments, using SIEVE.*

I. Introduction

MOBILE Ad-hoc Networks (MANET) are characterized by open, distributed and dynamic architectures, built on top of the shared wireless medium; all features contribute to make MANET very vulnerable to attacks at any layer of the Internet model.

We consider a particular type of active, non cryptography related attack, where insider nodes corrupt data at the application level (this is also known as pollution attack). In this paper we deem as a use case a data dissemination application over a MANET. Nodes generate data chunks to be Disseminated to all participants using rateless codes; some malicious nodes deliberately modify coded packets of a chunk before relaying them to prevent honest nodes from obtaining the original information.

In this paper we propose SIEVE a decen-tralized, accurate and robust technique to identify malicious nodes on top of an Other-wise reliable and attacker free architecture. Each node in SIEVE dynamically creates a bipartite graph whose vertexes are checks and uploading nodes. A check is a report created by a node upon decoding a data chunk; a check contains a variable length list of nodes identifiers that provided parts of the data as well as a flag to signal if the data chunk has been corrupted. Detection of the compromised chunks is achieved exploiting the constraints imposed by linear channel coding. The factor graph is periodically and independently analyzed by each node running an incremental version of the Belief Propagation (BP) algorithm [3]–[6]. The proposed algorithm allows each node to compute the probability of any other node being malicious; these latter probabilities are used to derive a suspect ranking of nodes in the MANET. Each node updates its local factor graph using the checks obtained by its own decoding operations as well as checks that are periodically gossiped by neighbor nodes.

1.1 Our Contributions

The major problem of the paper are the identification malicious nodes in terms of the estimation of the marginal probabilities on a bipartite graph and the proposal of a decentralized and accurate solution based on the BP algorithm. It is worth pointing out that the Selected data dissemination application is just a quite popular use case to a single scenario. In particular, SIEVE can be used in any application that uses multi-party download or collaboration, provided that is possible to detect that a given set of collaborating entities is compromised by at least one malicious node. As opposed to cryptographic/algebraic techniques proposed in the area of network coding based wireless mesh networks, e.g., [9]–[12] SIEVE does not rely on verification tools to check the integrity of every coded block. In SIEVE the BP algorithm is used to infer the identity of the malicious nodes resting upon only on a simple pollution detection mechanism; as an example, in our reference scenario, pollution detection is achieved as a by product of the data dissemination protocol based on rate-less codes. Furthermore, the aforementioned solutions may not be suitable for MANET since mobility is likely to affect key pre distribution, routing mechanisms, attack behavior, etc. SIEVE fits well two key MANETs features that must be accounted for when devising any security solution: it is fully decentralized and does not rely on any infrastructure (as opposed to some solutions in the area of peer-to-peer streaming where special well known nodes are necessary,e.g., [13]). Furthermore, SIEVE requires small computational, storage, and communication costs for implementation.

II. Sieve

In this paper we consider a MANET created a N wireless nodes moving in a given area. A set of Nsource nodes periodically produces a new data packet to be disseminated to all others once every h seconds. All nodes cooperate to the diffusion of the data chunks by running a distributed dissemination algorithm based on Luby Transform (LT) codes [15]. Data is transmitted by source nodes using LT codes [15]: a packet is divided in K equally sized blocks. The source node then creates and forwards coded packets using LT codes [15], combining random subsets of the K blocks; the size of each coded packet is Scb = S/K.

2.1 LT Encoding And Decoding Process

LT codes have been proposed in [15] this paper is a rateless codes: these are a particular family of erasure codes where the rate is not fixed by design, so that the number of coded packets can be decided and changed on the fly. LT codes are rateless based on the binary Galois field GF(2). In [15] it is shown that selecting the number of blocks to be combined, termed as the packet degree d, according to the Robust Soliton Distribution (RSD), one gets optimal asymptotic decoding performance. By optimal performance we mean that the so called decoding overhead, i.e. the number of coded packets to be received in excess of K, turns to be negligible for asymptotically large K. In particular, the original chunk can be obtained by any node able to collect any set of M = K · (1 + □) coded packets. The decoding algorithm can be viewed as the solution of a system of linear equations with K unknowns (the K original data blocks) and $M \geq K$ equations. In [15] it is shown that a simple method based on the recursive cancellation of equations corresponding to packets with degree 1, i.e. representing one original data blocks, guarantees the desired asymptotic performance.

2.2 LT Dissemination Protocol

The rateless principle and randomness of LT codes is used for spreading data in the MANET by letting source nodes transmit novel coded packets that can be gene-rated randomly and on the fly. A coded packet conveys the XOR payloads of the corresponding original packets as well as a header signaling the indexes of the combined packets. The original chunk can be obtained by any node able to collect any set of K · (1 + □) coded packets without requiring any coordination among the source nodes.

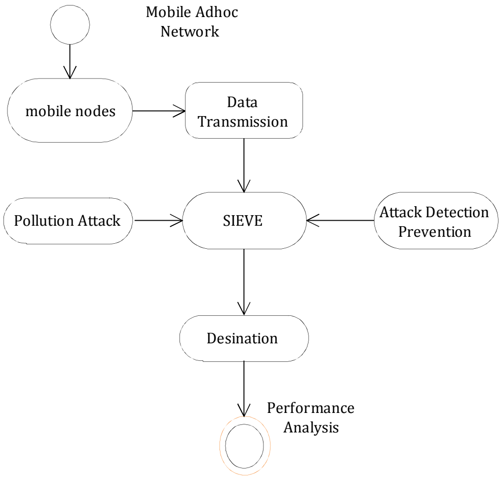

Fig 1.1 SIEVE preventing malicious node.

Fig 1.1 ,if the source node can transmitted the data from source to destination throught.

SIEVE. SIEVE using LT code to prevent data from malicious node.It can mainly identify malicious node produces pollution attack.

2.3 Malicious Nodes

The proposed dissemination protocol is an example of a distributed and collaborative approach that has the potential to simplify and accelerate the spreading of the information in the MANET thanks to node mobility. On the other hand, few malicious nodes may try to break the system by polluting, i.e. modifying some coded packets. In this paper we assume that a subset of $P < N$ nodes is composed of malicious nodes, That deliberately modify the payload of the coded packets to prevent honest nodes from correctly reconstructing the original chunk. In the presence of coding even a single corrupted coded packet can prevent an honest node from decoding the original chunk.

III. POLLUTION IDENTIFICATION PROTOCOL

SIEVE uses LT codes decoding scheme to detect changed packet and exploits the Belief Propagation (BP) algorithm [3] to identify malicious nodes.

3.1 LT codes Verification

The every node keeps collecting from different uploaders sets of coded blocks corresponding to different chunks. LT codes can be exploited to detect if modified blocks have been collected without the need of any supplementary verification mechanism. Indeed, a node is able to detect pollution as soon as an inconsistency is found in the solution of the underlying system of linear equations. This is achieved by observing that a coded packet with a degree 1 equation represents a data packet in the clear. Such data packet can be simplified from all the incoming equations. Since LT codes have a certain overhead some coded packets that are linearly dependent on the ones received previously are always collected before successful decoding; this amount to the reception of some equations whose terms are all already known. As soon as this condition is met the LT decoder can check the consistency of the payload carried by the coded packet; in other words, the same linear combination must be obtained combining a set of already known packets. If this constraint is violated the whole chunk is recognized as corrupted. Please note that the receiver node is not able to identify the corrupted block(s) but only that at least one of them has been maliciously manipulated.

3.2 Check Construction and Reporting

SIEVE is based on the concept of checks that are reports created by nodes upon decoding a chunk. A check contains the list of the identifiers of nodes that provided coded blocks of a chunk and a flag to label such chunk as corrupted or not. A check describing a corrupted chunk is called a positive check while it is termed a negative check otherwise. Each nodes n maintains a list of all checks created that is denoted as Ln. Each node, besides accumulating the checks from its local decoding operations, gossips them in the neighborhood.

3.3 performance analysis

Sieve can be identified three main performance analysis as shown in the graph.
The mainly consider the three performance such as throughput, packet delivery ratio and end to end delay.

3.4 Average End To End Delay

Fig 1.2 shows the accurately identify the performance analysis of average end to end delay

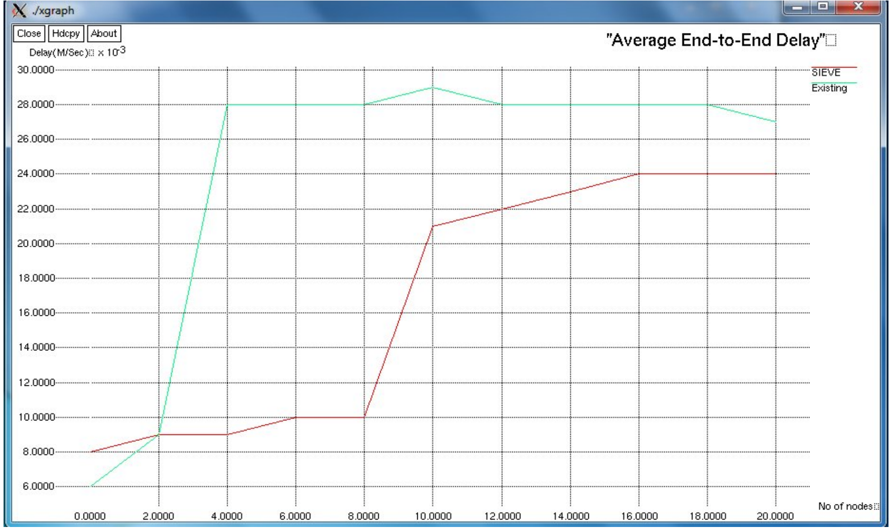

Fig 1.2 Average End To End Delay

3.5 packet delivery ratio

Fig 1.3 shows the accurately identify the performance analysis of packet delivery ratio

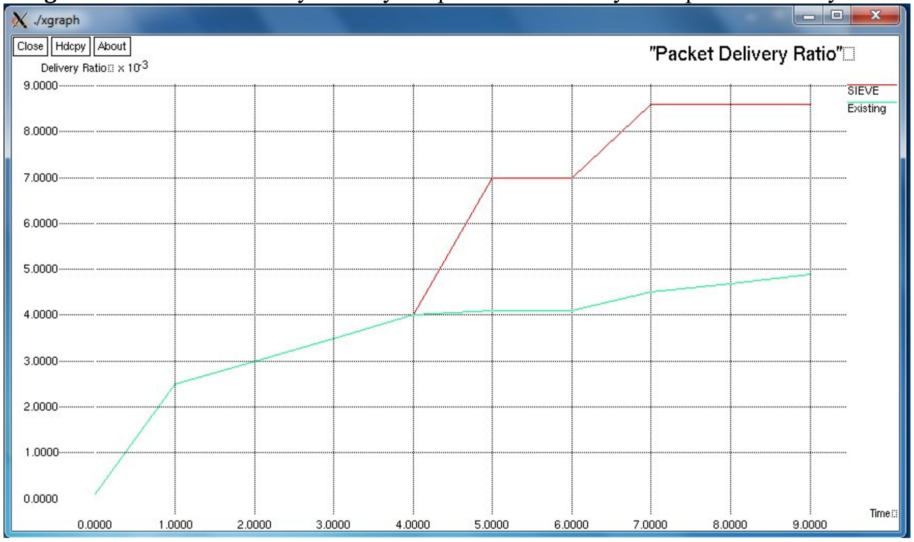

Fig 1.3 Packet Delivery Ratio

3.6 THROUGHPUT RATIO

FIG 1.4 shows the accurately identify the performance analysis of throughput ratio

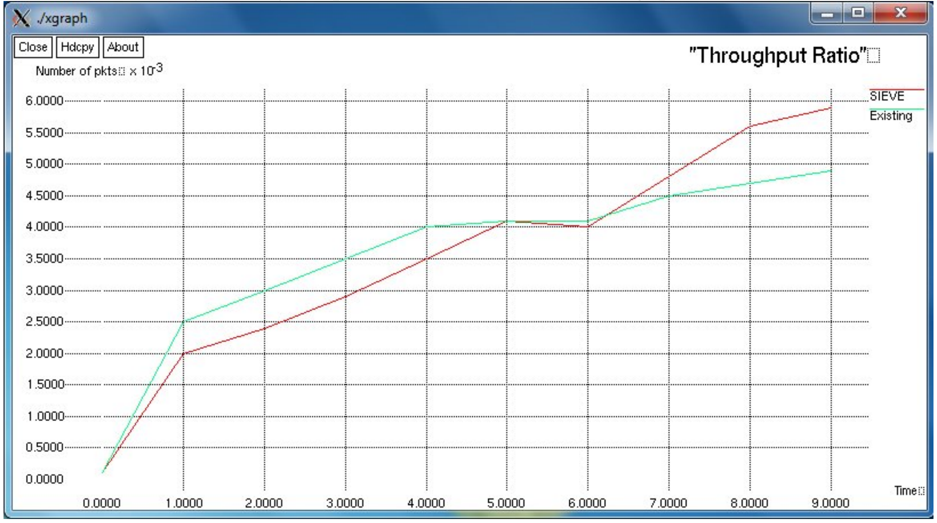

Fig 1.4 throughput ratio

IV. Results

The simulation methodology and the indexes we defined to evaluate the accuracy, reactivity, and robustness of SIEVE. In particular, we show the packet delivery ratio, average end to end delay and throughput ratio of the factor graph and mobility impact the performance of SIEVE as well as how robust it is with respect to several deceiving actions operated by malicious nodes. Finally, we show a trade-off between coding efficiency and SIEVE accuracy.

V. Conclusion

In this we used sieve technique to accurately identified the malicious node. The malicious node produces pollution attack to a original packet. A data can be separated application based on LT code. Fortunately, the LT decoding procedure can be used by each node to detect that a data chuck has been attacked; nonetheless, since parallel downloading form multiple nodes is used, such detection does not allow to identify the malicious nodes. This latter represents the core problem solved by SIEVE. In SIEVE node collaborations are

represented by a bipartite graph linking nodes and detection opportunities, the checks. It is worth pointing out that such representation is quite general and can be used in many other collaborative scenarios
other than the data dissemination use case analyzed in this paper.

Future works will be focused in two main directions. From the one hand we will complete the design and experimentation of a full system to counteract an active attack in MANET. The techniques adopted for the identification and the following removal of malicious nodes clearly require a joint and careful design to optimize the overall performance.

Reference

[1] B. Wu, J. Chen, J. Wu, and M. Cardei, "A survey of attacks and countermeasures in mobile ad hoc networks," in Wireless Network Security, Y. Xiao, X. S. Shen, and D.-Z. Du, Eds. New York, NY, USA: Springer, 2007, pp. 103–135.

[2] H. Yang, H. Luo, F. Ye, S. Lu, and L. Zhang, "Security in mobile ad hoc networks: Challenges and solutions," IEEE Wireless Commun., vol. 11, no. 1, pp. 38–47, Feb. 2004.

[3] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. San Francisco, CA, USA: Morgan Kaufmann Publishers, Inc., 1988.

[4] D. MacKay, Information Theory, Inference and Learning Algorithms. Cambridge, U.K.: Cambridge University Press, 2003.

[5] J. Yedidia, W. Freeman, and Y. Weiss, "Constructing free-energy approximations and generalized belief propagation algorithms," IEEE Trans. Inf. Theory, vol. 51, no. 7, pp. 2282–2312, Jul. 2005.

[6] J. Yedidia, W. Freeman, and Y. Weiss, "Understanding belief propagation and its generalizations," in Exploring Artificial Intelligence in the New Millennium, San Francisco, CA, USA: Elsevier, 2003.

[7] T. Schierl, S. Johansen, A. Perkis, and T. Wiegand, "Rateless scalable video coding for overlay multisource streaming in manets," J. Vis. Commun. Image Represent., vol. 19, no. 8, pp. 500–507, 2008.

[8] V. R. Syrotiuk, C. J. Colbourn, and S. Yellamraju, "Rateless forward error correction for topology-transparent scheduling,"IEEE/ACM Trans. Netw., vol. 16, no. 2, pp. 464–472, Apr. 2008.

[9] S. Agrawal and D. Boneh, "Homomorphic MACs: MAC-based integrity for network coding," in Applied Cryptography and Network Security. Berlin, Germany: Springer, 2009, pp. 292–305.

[10] J. Dong, R. Curtmola, and C. Nita-Rotaru, "Practical defenses against pollution attacks in intra-flow network coding for wireless mesh networks," in Proc. 2nd ACM Conf. WiSec, Zurich, Switzerland, 2009, pp. 111–122.

[11] Y. Li, H. Yao, M. Chen, S. Jaggi, and A. Rosen, "Ripple authentication for network coding," in Proc. IEEE INFOCOM, San Diego, CA, USA, 2010, pp. 1–9.

[12] A. Newell and C. Nita-Rotaru, "Split null keys: A null space based defense for pollution attacks in wireless network coding," in Proc. 9th IEEE SECON, Seoul, Korea, 2012, pp. 479–487.

[13] Q. Wang, L. Vu, K. Nahrstedt, and H. Khurana, "MIS: Malicious nodes identification scheme in network-coding-based peer-topeer streaming," in Proc. IEEE INFOCOM, San Diego, CA, USA, Mar. 2010, pp. 1–5.

[14] R. Gaeta, M. Grangetto, and R. Loti, "SIEVE: A distributed, accurate, and robust technique to identify malicious nodes in data dissemination on manet," in Proc. IEEE ICPADS, Washington, DC, USA, 2012, pp. 331–338.

[15] M. Luby, "LT codes," in Proc. 43rd FOCS, Washington, DC, USA, 2002, pp. 271–280.

[16] R. Gallager, Low-Density Parity-Check Codes. Cambridge, U.K.: MIT Press, 1963.

[17] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael, "Learning low-level vision," Int. J. Comput. Vis., vol. 40, no. 1, pp. 25–47, 2000.

[18] M. Luby, M. Mitzenmacher, M. Shokrollahi, and D. Spielman, "Improved low-density parity-check codes using irregular graphs," IEEE Trans. Inf. Theory, vol. 47, no. 2, pp. 585 –598, Feb. 2001.

[19] G. F. Riley and T. R. Henderson, "The NS-3 network simulator," in Modeling and Tools for Network Simulation. Berlin, Germany:Springer, 2010, pp. 15–34.

[20] M. N. Krohn, M. J. Freedman, and D. Mazieres, "On-the-fly verification of rateless erasure codes for efficient content distribution,"in Proc. IEEE Symp. Security Privacy, 2004.

[21] C. Gkantsidis and P. Rodriguez, "Cooperative security for network coding file distribution," in Proc. IEEE INFOCOM, Barcelona, Spain, 2006.

[22] E. Kehdi and B. Li, "Null keys: Limiting malicious attacks via null space properties of network coding," in Proc. IEEE INFOCOM, Rio de Janeiro, Brazil, 2009.

[23] Z. Yu, Y.Wei, B. Ramkumar, and Y. Guan, "An efficient scheme for securing xor network coding against pollution attacks," in Proc. IEEE INFOCOM, Rio de Janeiro, Brazil, 2009.

[24] T. Ho et al., "Byzantine modification detection in multicast networks with random network coding," IEEE Trans. Inf. Theory, vol. 54, no. 6, pp. 2798–2803, Jun. 2008.