ISSN: 2321-7782 (Online)
Impact Factor: 6.047

Volume 4, Issue 7, July 2016

International Journal of Advance Research in Computer Science and Management Studies

Research Article / Survey Paper / Case Study

Available online at: www.ijarcsms.com

A secure model for detecting origin forgery and packet drop attacks in WSN

T. Poovayee[1]
Research scholar
H.H The Rajah's College (autonomous),
Pudukkottai, India

K. Karpagam[2]
Research guide,Assistant Professor of Computer Sciences
H.H.The Rajah's College (autonomous),
Pudukkottai, India

Abstract: The basic operation of wireless sensor network is the efficient gathering and transmission of sensed data to a base station for advance processing. The life of such a sensor system is the time during which it can gather information from all the sensors to the base station. A fundamental challenge in data gathering is to maximize the system lifetime, given the energy constraints. As sensor networks are being all the time more deployed in decision-making the process that on in packet Bloom filters to encode provenance of the information. This proposed work introduces efficient tools for provenance verification method and reconstruction method at the base station with the functionality to detect packet drop attacks or by malicious data forwarding nodes. To propose a novel lightweight scheme to securely transmit provenance for sensor data. In addition the secure provenance scheme with the functionality to detection packet drop attacks by malicious data from the source to destination node. This work proposes a new lightweight scheme in order to securely transmit provenance with sensor data. This mechanism initially performs provenance at the base station then perform reconstruction of the data at the base station. In addition to this the provenance scheme functionality used to detect packet drop attacks organized by malicious data forwarding nodes. This work describes the effectiveness and efficiency of the Light weight secure provenance scheme in detecting packet forgery and packet loss attacks.

Keywords: Average, Bloom Filter, Count, Multi-Hop WSN, Median.

I. INTRODUCTION

In a multi-hop sensor network, data provenance allows the BS to trace the source and forwarding path of an individual data packet. Provenance must be recorded for each packet, but important challenges arise due to the tight storage, energy and bandwidth constraints of sensor nodes. Therefore, it is necessary to devise a light-weight provenance solution with low overhead. Furthermore, sensors often operate in an untrusted environment, where they may be subject to attacks. Hence, it is necessary to address security requirements such as confidentiality, integrity and freshness of provenance.

The aim of this work goal is to design a provenance encoding and decoding mechanism that satisfy such security and performance needs. This paper a provenance encoding strategy whereby each node on the path of a data packet securely embeds provenance information within a Bloom filter (BF) that is transmitted along with the data. Upon receiving the packet, the BS extracts and verifies the provenance information. This paper also devise an extension of the provenance encoding scheme that allows the BS to detect if a packet drop attack was staged by a malicious node. Sensor networks are used in various areas like such as cyber physical infrastructure system, environmental Weather monitoring, power grids, etc. Data are originated from a huge number of sensor node sources and they are processed at intermediate hops at in networks.

II. LITERATURE REVIEW

Salmin Sultana et al [1] proposes a novel lightweight scheme to securely transmit provenance for sensor data. The proposed technique relies on in-packet Bloom filters to encode provenance. He introduces efficient mechanisms for provenance verification and reconstruction at the base station. In addition, he extend the secure provenance scheme with functionality to detect packet drop attacks staged by malicious data forwarding nodes.

Koustuv Dasgupta et al **[2]**they considers a network of energy-constrained sensors that are deployed over a region. Each sensor periodically produces information as it monitors its vicinity. The basic operation in such a network is the systematic gathering and transmission of sensed data to a base station for further processing.

III. METHODOLOGY

A. Cryptography

Cryptography is the practice and study of techniques for secure communication in the presence of third parties. More generally, it is about constructing and analyzing protocols that overcome the influence of adversaries and which are related to various aspects in information security such as data confidentiality, data integrity, authentication, and non-repudiation. Modern cryptography intersects the disciplines of mathematics, computer science, and electrical engineering. Applications of cryptography include ATM cards, computer passwords, and electronic commerce.

1. Key generation

RSA involves a public key and a private key. The public key can be known to everyone and is used for encrypting messages. Messages encrypted with the public key can only be decrypted using the private key.

2. Encryption

Alice transmits her public key $(n,e)$to Bob and keeps the private key secret. Bob then wishes to send message M to Alice.

Alice first turns M into an integer m, such that $0 \leq m < n$by using an agreed-upon reversible protocol known as a padding scheme. He then computes the ciphertext $c$corresponding to $c = m^e \pmod{n}$.

This can be done quickly using the method of exponentiation by squaring. Bob then transmits c to Alice.

Note that at least nine values of m could yield a ciphertext c equal to m, but this is very unlikely to occur in practice.

3. Decryption

Alice can recover m from c by using her private key exponent d via computing $m = c^d \pmod{n}$. Given m, she can recover the original message M by reversing the padding scheme.

Sign Module

In sign module the following processes are performed. 1. Key generation, 2.encryption, 3.key exchanging 4.signature 5.send to verify module

Provenance Verification

In provenance verify module the following processes are performed. 1. Key generation, 2.decryption, 3.key exchanging 4.send to receiver module

Provenance Collection

If receiver modules receives a packet data suspicious, it is placed in suspicious box, suppose if data is correct it is placed in province box.

Data-provenance

Setup: the data producer sets up its signing key k and data consumer sets up its verification key k_0 in a secure fashion that prevents malware from accessing the secret keys.

Sign(D, k): the data producer signs its data D with a secret key k, and outputs D along with its proof sig.

Verify (sig, D, k_0): the data consumer uses key k_0 to verify the signature sig of received data D to ensure its origin, and rejects the data if the verification fails.

B. Secure Provenance Encoding

This method propose a distributed mechanism to encode provenance at the nodes and centralized algorithm to decode it at the BS. The technical core of our proposal is the notion of in packet Bloom filter. Each packet consists of a unique sequence number, data value, and an iBF which holds the provenance. This paper emphasize that our focus is on securely transmitting provenance to the BS. In an aggregation infrastructure, securing the data values is also an important aspect, but that has been already addressed in previous work. Our secure provenance technique can be used in conjunction with such work to obtain a complete solution that provides security for data, provenance and data-provenance binding.

C. Detecting Packet Drop Attacks

The secure provenance encoding scheme to detect packet drop attacks and to identify malicious node(s). To assume the links on the path exhibit natural packet loss and several adversarial nodes may exist on the path. For simplicity, this method consider only linear data flow paths. Also, it do not address the issue of recovery, once a malicious node is detected. Existing techniques that are orthogonal to our detection scheme can be used, which may initiate multipath routing or build a dissemination tree around the compromised nodes. The augment provenance encoding to use a packet acknowledgement that requires the sensors to transmit more meta-data.

For a data packet, the provenance record generated by a node will now consist of the node ID and an acknowledgement in the form of a sequence number of the lastly seen (processed/forwarded) packet belonging to that data flow. If there is an intermediate packet drop, some nodes on the path do not receive the packet. Hence, during the next round of packet transmission, there will be a mismatch between the acknowledgements generated from different nodes on the path. This fact is utilized to detect the packet drop attack and to localize the malicious node.

IV. ALGORITHM USED

A. Secure Provenance Encoding

The secure provenance technique can be used in conjunction with such work to obtain a complete solution that provides security for data provenance and data-provenance binding. A distributed mechanism is proposed to encode provenance at the nodes and centralized algorithm to decode it at the BS. The technical core of our proposal is the notion of in-packet Bloom filter (iBF).

Each packet consists of a unique sequence number, data value, and an iBF which holds the provenance. This paper emphasize that our focus is on securely transmitting provenance to the Base station. The secure provenance technique can be used in conjunction with such work to obtain a complete solution that provides security for data provenance and data provenance binding.

B. Provenance Encoding

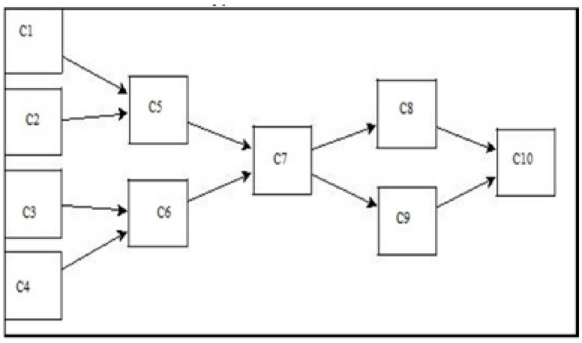


Fig 1 Provenance encoding contributor

The Figure shows that to produce the final result, the contributor C5 uses the outputs of contributors C1 and C2 while contributor of C6 uses the output of contributors C3 and C4. Contributor C7 uses the output of C5 and C6 which later used by C8 and C9. C10 is the final process is executed by that processes the outputs of C8 and C9. After each process is executed and the provenance of the process the had created/generated, the provenance is stored in the provenance database.

C. Provenance Decoding

When the base station receives the data packet station .base know what data packets should be checked. Then, when a packet is received, it is sufficient for the BS to validate the knowledge source coding them in the package.

Algorithm-1 Provenance Verification

```
Input: Received packet with sequence seq and iBF ibf.
Set of hash functions H, Data path P = < n l 1... n 1 , ..., n p >
BF c ← 0 // Initialize Bloom Filter
 for each n i ∈ P do
vid i = generateVID (n i , seq)
insert vid i into BF c using hash functions in H
 endfor
if (BF c = ibf ) then
return true // Provenance is verified
endif
 return false
```

Algorithm-2 Provenance Collection:

```
Input: Received packet with sequence seq and iBF ibf. N Set of nodes (N ) in the network, Set of hash functions H
1. Initialize Set of Possible Nodes S ← ∅ Bloom Filter BF c ← 0 // To represent S
2. Determine possible nodes in the path and build the representative BF
for each node n i ∈ N do
vid i = generateVID (n i , seq)
if (vid i is in ibf ) then
```

```
$S \leftarrow S \cup n i$
insert vid i into BF c using hash functions in H
endif
endfor
3. Verify BF c with the received iBF
if (BF c = ibf ) then
return S // Provenance has been determined correctly
else
return NULL // Indicates an in-transit attack
endif
```

V. Experimental Result

In a multi-hop wireless network, nodes cooperate in relaying/ routing traffic. An adversary can exploit this cooperative nature to launch attacks. For example, the adversary may first pretend to be a cooperative node in the route discovery process. Once being included in a route, the adversary starts dropping packets.

Detecting selective packet-dropping attacks is extremely challenging in a highly dynamic wireless environment. The difficulty comes from the requirement that we need to not only detect the place (or hop) where the packet is dropped, but also identify whether the drop is intentional or unintentional. Specifically, due to the open nature of wireless medium, a packet drop in the network could be caused by harsh channel conditions e.g., fading, noise, and interference, link errors, or by the insider attacker.

This method is implemented and tested by proposed techniques using the .net framework with wsn simulator and to measure the energy consumption. To consider a network of 100 nodes and vary the network diameter from 2 to 14. All results are averaged over 100 runs. First, we look at how effective the secure provenance encoding scheme is in detecting provenance forgery and path changes. Next, we investigate the accuracy of the proposed method for detecting packet loss. Finally, we measure the energy consumption overhead of securing provenance.

A. Bloom Filter

In-packet Bloom-filter is one of the technique used for provenance encoding. Provenance encodes history of data at each node, therefore provenance size increases with the increase in the number of nodes in network. This is inefficient as performance decreases due to high bandwidth consumption. Thus the main focus is to make provenance size light weight, secure transmission with forgery detection and finding packet drop attack using provenance data. This results in decreased bandwidth and energy which is the key factor in WSN. The second goal is to design a provenance encoding and decoding mechanism that satisfies security and performance needs by assuring confidentiality, integrity and originality of provenance.

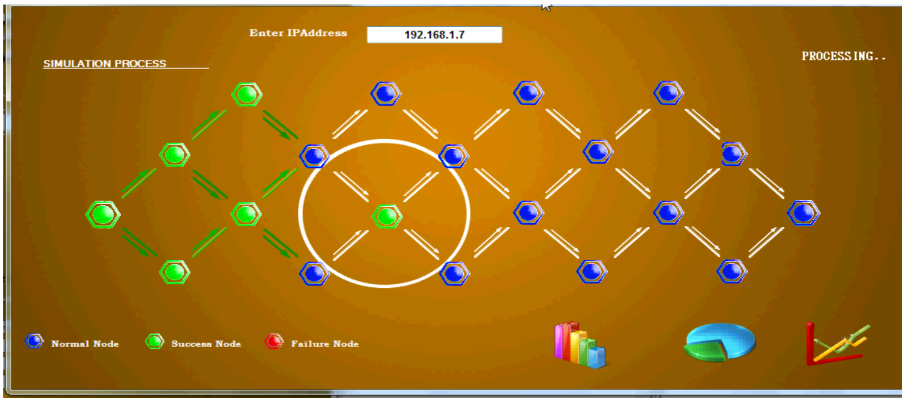
B. Provenance verification

The BS first executes the provenance verification process upon receiving a packet. The BS knows

1) The current data path for the packet (decoded from the provenance of the previous packet in the flow), and

2) The preceding packet sequence number forwarded by each node in the path. In this context,

The BS assumes that each node in the path saw and forwarded the same packet in the last round, and that this packet's sequence number is the same one as recorded at the BS. Thus the verification is bound to fail when pSeq and pSeqb do not match, which also indicates a possible packet loss and suffices to execute provenance collection process directly skipping the verification.

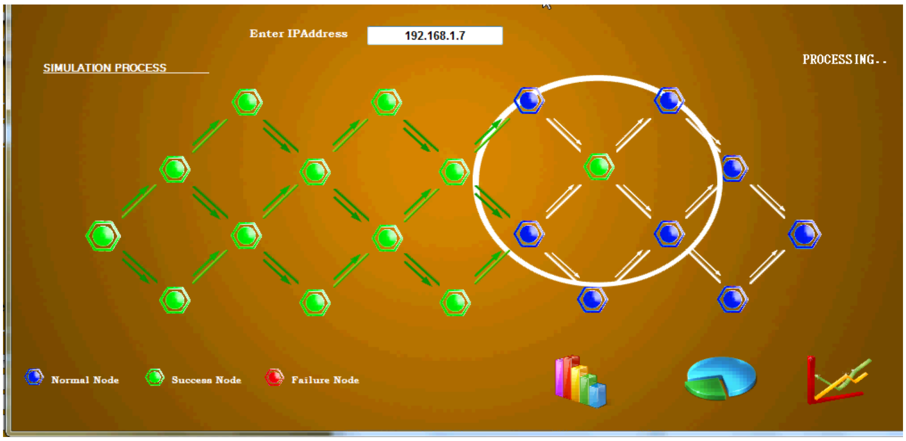


Enter IPAddress 192.168.1.7

Fig 2. provenance verification process

C. *Provenance collection*

Collection attempts to retrieve the nodes from the encoded provenance, confirm a packet loss and identify the malicious node that dropped the packet. It also distinguishes between the packet drop attack and other attacks that might have altered the iBF. Note that, in case of a path change, the new nodes can be easily learnt through an iteration of ibf membership testing over all the nodes.



Enter IPAddress 192.168.1.7

Fig 3 Provrnance collection from the encoded provenance

D. *Space Complexity and Energy Consumption*

The provenance length in SSP and MP increases linearly with the path length. For our scheme, the empirically determine the BF size which ensures no decoding error. Although then they also measure the energy consumption for both the basic provenance scheme and the extended scheme for packet drop detection, while

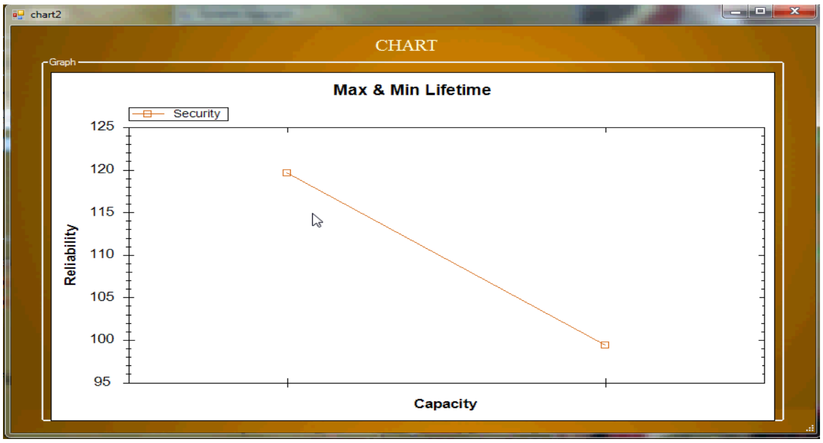


Fig 4 Provenance length in SSP and MP

Varying hop counts for packet drop attack, the set the malicious link loss rate as 0.03. Note that, modern sensors use ZigBee pecification for high level communication protocols which allows up to 104 bytes as data payload. Hence, SSP and MP can be used to embed provenance (in data packet) for maximum 2 and 14 nodes, respectively. The results confirm the energy efficiency of our solutions.

E. *Bloom filter based provenance scheme (BFP)*

This scheme uses a fixed size Bloom filter (BF) to encode the provenance of a packet. It embeds all the nodes on a packet's path in the BF using a set of hash functions.

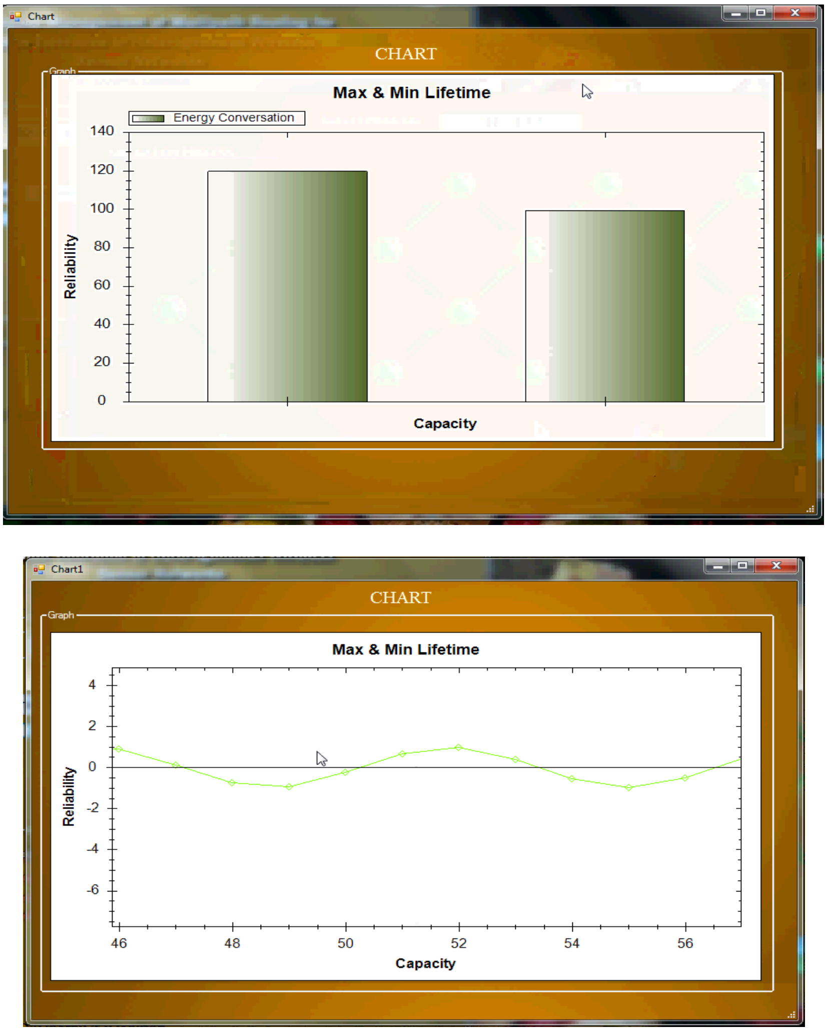


Fig 5 Performance of BFP

In Figure 5 shows the network overhead for separate Bloom filter lengths with increasing number of receivers for the dataset. The x-axis on the bottom of the figure shows the number of receivers whereas the values on the top show the number of links in the Bloom filter, respective to the number of receivers. It shows the comparative performance analysis with existing filtering protocols along with the reliability of the proposed work.

VI. Conclusion

In this work ensure the problem secure transmission source sensor networks, and propose a Bloom filter based on lightweight source coding and decoding scheme to detect forgery and packet drop attacks in WSN. The program ensures the freshness of the confidentiality, integrity and origin. The have expanded to include links to data sources, and include support for packet attack detection sequence information packet loss programs. Experimental and analytical results of the evaluation show that the program is effective, lightweight and scalable.

References

1. Salmin Sultana, Gabriel Ghinita, Elisa Bertino, Fellow, Mohamed Shehab, “A LightWeight Secure Scheme for Detecting Provenance Forgery and Packet Drop Attacks in Wireless Sensor Networks” , Published by the IEEE Computer Society Issue No.03 - May-June (2015 vol.12)
2. Koustuv Dasgupta, Konstantinos Kalpakis Parag Namjoshi,”An Efficient Clustering based Heuristic for Data Gathering and Aggregation in Sensor Networks”. 0-7803-7700-1/03/$17.00 (C) 2003 IEEE
3. Summary Cache: A Scalable Wide-Area Web Cache Sharing Protocol by Li Fan,Pei Cao, Jussara Almeida, and Andrei Z. Broder, 0-7803-7700-1/03/$17.00 (C) 2003 IEEE
4. In-packet Bloom filters: Design and networking applications by Christian Esteve Rothenberg, Carlos A. B. Macapuna, Mauricio F. Magalhaesa, F´abio L. Verdib, AlexanderWiesmaier, In Proceedings of International Conference on Communications, 2009
5. Secure Hierarchical In-Network Aggregation in Sensor Networks by HaoWen Chan, Adrian Perrig and Dawn Song IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 8, NO. 3, JUNE 2000
6. Yu, S. Kallurkar, “A demspter shafer approach to provenance awareness trust assessment,” in CTS 2008: International Sym-posium on Collaborative Tech. and System, pp. 383–390, May 2008.
7. Carbunar, I. Ioannidis and C. Nita-Rotaru. JANUS: Towards Robust and Malicious Resilient Routing in Hybrid Wireless Networks. WiSe 2004, pp. 11-20.
8. S. Sultana, M. Shehab, E. Bertino. Secure Provenance Transmission for Streaming Data. SUBMITTED in IEEE Transaction on Knowledge and Data Engineering (TKDE), 2011.
9. Groth, P., Jiang, S., Miles, S., Munroe, S., Tsasakou, S., Moreau, L.: An architecture for provenance systems.(Nov. 2006)
10. Buneman, P., Khanna, S., Tan, : Why and where: A characterization of data provenance. In: ICDT. (2001) 316–330
11. Hasan, R., R., Winslett, M.: Preventing history forgery with secure provenance. ACM Transactions on Storage 5(4) (December 2009) 12:1–12:43
12. Hasan, R., Sion, M .: The case of the fake picasso: Prevent against history forgery with secure provenance. In: FAST. (2009) 1–14
13. Salmin Sultana,Gabriel Ghinita, and Mohamed Shehab, *Member*” A LightWeight Secure Scheme for Detecting Provenance Forgery and Packet Drop Attacks in Wireless Sensor Networks” *Member, IEEE,* Elisa Bertino, *Fellow, IEEE, , IEEE* [1]H. Lim, Y. Moon, and E. Bertino, “Provenance-based trustworthiness assessment in sensor networks,” in Proc. of Data Management for Sensor Networks, 2010, pp. 2–7.
14. Foster, J. Vockler, M. Wilde, and Y. Zhao, “Chimera: A virtual data system use for representing, querying” in Proc. of the Conf. on Scientific and Statistical Database Management, 2002, pp. 37–46.
15. K. Muniswamy-Reddy, D. Holland, U. Braun, and M. Seltzer, “Provenance aware storage systems,” 2006, pp. 4–4.