International Journal of Modern Agriculture, Volume 9, No. 4, 2020
ISSN: 2305-7246

TO AVOID FALSE DATA INJECTION USING BLOOM FILTER

C. Nalini[1], G L Vara Prasad[2]

[1]Professor, Department of Computer Science Engineering, Bharath Institute of Higher Education and Research University Chennai, India

[2]Ph.D. Scholar, BIST,BIHER,Bharath University, Chennai

Abstract

In today's developing world a large amount of sensor networks are used in various real-time applications. Large amount of Data are streamed from multiple sources through intermediate processing nodes that share information throughout the network. Some of the domains where these Sensor networks are used include cyber physical infrastructure systems, environmental monitoring, power grids, etc. Data goes through various nodes to complete a path from producer system to consumer system.

Malicious attacks are possible in between. This project hereby introduces an in-packet bloom filter, which will keep the track of provenance details of each packet separately. This proposal is about a novel lightweight scheme to securely transmit provenance for sensor data. It introduces efficient mechanisms for provenance verification and reconstruction at the base station through decoding of the provenance details. This extends the secure provenance scheme with functionality to detect packet drop attacks. The goal is to design a provenance encoding and decoding mechanism that satisfies such security and performance needs.

Key words: provenance encoding; primary key, inbloom filters

Introduction

Sensor networks are used in numerous application domains, such as cyber physical infrastructure systems, environmental monitoring, power grids, etc. Data are produced at a large number of sensor node sources and processed in-network at intermediate hops on their way to a base station (BS) that performs decision-making. 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. It is necessary to address security requirements such as confidentiality, integrity and freshness of provenance. Our goal is to design a provenance encoding and decoding mechanism that satisfies such security and performance needs.

Problem Identification

The provenance modeling, collection, and querying has been studied extensively for workflows and curate databases; provenance in sensor networks has not been properly addressed. It does not consider denial of service attacks such as the complete removal of provenance, since a data packet with no provenance records will make the data highly suspicious and hence generate an alarm at the Base S. Any confidentiality cannot gain any knowledge about data provenance by analyzing the contents of a packet. Integrity, acting alone or colluding with others, cannot add or remove non-colluding nodes from the provenance of benign data (i.e., data generated by benign nodes) without being detected. Freshness, cannot replay captured data and provenance without being detected by the Base Station.

It is also important to provide Data-Provenance Binding, i.e., a coupling between data and provenance so that an attacker cannot successfully drop or alter the legitimate data while retaining the provenance, or swap the provenance of two packets. In a distributed aggregate computation to verify that the final result has not been perturbed by more than a small error bound with high probability. It do not address the issue of recovery once a malicious node is detected. If there is an intermediate packet drop, some nodes on the path do not receive the packet. When one-way hash functions are used to insert elements in the BF, the identities of the inserted elements cannot be reconstructed from the BF representation.

Related Works

[1]. *H. Lim, Y. Moon, and E. Bertino, **"Provenance-Based Trustworthiness Assessment in Sensor Networks,"*** As sensor networks are being increasingly deployed in decision-making infrastructures such as battle field monitoring systems and SCADA (Supervisory Control and Data Acquisition) systems, making decision makers aware of the trustworthiness of the collected data is a crucial.

This approach uses the data provenance as well as their values in computing trust scores, that is, quantitative measures of trust worthiness. The provenance similarity is based on the principle that "the more different data provenances with similar values, the higher the trust scores".

[2] *Foster, J. Vockler, M. Wilde, and Y. Zhao, **"Chimera: A Virtual Data System for Representing, Querying, and Automating Data Derivation,"*** A lot of scientific data is not obtained from measurements but rather derived from other data by the application of computational procedures. We hypothesize that explicit representation of these procedures can enable documentation of data provenance, discovery of available methods, and on-demand data generation (so-called "virtual data").

[3] A Chimera virtual data system, which combines a virtual data catalog for representing data derivation procedures and derived data, with a virtual data language interpreter that translates user requests into data definition and query operations on the database.

[4] *K. Muniswamy-Reddy, D. Holland, U. Braun, and M. Seltzer, **"Provenance-Aware Storage systems,"*** A Provenance-Aware Storage System (PASS) is a storage system that automatically collects and maintains provenance or lineage, the complete history or ancestry of an item. We discuss the advantages of treating provenance as meta-data collected and maintained by the storage system, rather than as manual annotations stored in a separately administered database. We describe a PASS implementation, discussing the challenges it presents, performance cost it incurs, and the new functionality it enables. We show that with reasonable overhead, we can provide useful functionality not available in today's file systems or provenance management systems.

[5] *Y. Simmhan, B. Plale, and D. Gannon, **"A Survey of Data Provenance in E-Science,"*** Data management is growing in complexity as large scale applications take advantage of the loosely coupled resources brought together by grid middleware and by abundant storage capacity.

Metadata describing the data products used in and generated by these applications is essential to disambiguate the data and enable reuse. Data provenance, one kind of metadata, pertains to the derivation history of a data product starting from its original sources.

Existing System

There are two main parts of the existing system.

- Pedigree captures provenance for network packets in the form of per packet tags that store a history of all nodes and processes that manipulated the packet.

- ExSPAN describes the history and derivations of network state that result from the execution of a distributed protocol.

Few Points To Overcome

- This system does not address security concerns and is specific to some network use cases. This becomes a disadvantage.

- This system traces the source of a stream long after the process has completed. It reflects the importance of issues we addressed, it is not intended as a security mechanism, hence, does not deal with malicious attacks.

Proposed System

1. The proposed technique relies on in-packet Bloom filters to encode provenance.

2. Here gets introduced an efficient mechanisms for provenance verification and reconstruction at the base station. In addition, we extend the secure provenance scheme with functionality to detect packet drop attacks staged by malicious data forwarding nodes.

3. It propose an in-packet Bloom filter (iBF) provenance-encoding scheme. The design efficient techniques for provenance decoding and verification at the base station. The detailed security analysis and performance evaluation of the proposed provenance encoding scheme and packet loss detection mechanism.

4. A multihop wireless sensor network, co/nsisting of a number of sensor nodes and a base station that collects data from the network

5. Each data packet contains 1) a unique packet sequence number, 2) a data value, and 3) provenance.

The sequence number is attached to the packet by the data source, and all nodes use the same sequence number for a given round . The sequence number integrity is ensured through MACs.

OVERALL ARCHITECTURE

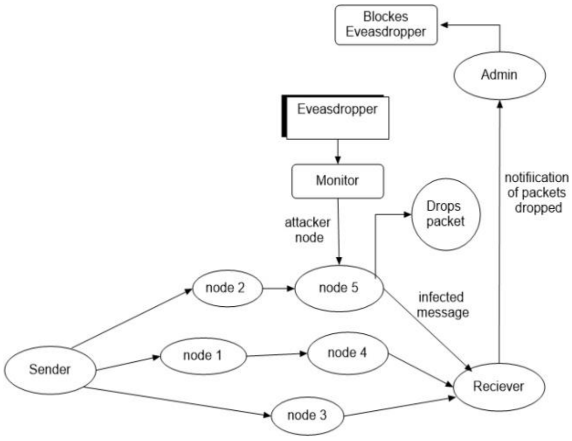


Figure1. Wireless sensor network showing the malicious attacker in between the regular nodes.

MODULES

Provenance Verification

In verification of the modules it processes the Key generation, decryption, and also the key exchanging, sent to receiver module. The RSA used here involves a public key and a private Key**.** The public key can be known to everyone and is used for encrypting the messages. Messages encrypted with the public key can only be decrypted using the private key which will act as the enhancing point of this paper. The keys for the RSA algorithm are generated. In Provenance Collection, receiver module receives a packet data suspicious means place in suspicious box suppose, the data means placed in province box. The Base Station conducts the verification process not just only to verify its knowledge of provenance but also to check the integrity of the transmitted provenance data of all the data packets involved.

1. Data Provenancce

Data provenance represents a key factor in evaluating the trustworthiness of sensor data.

Setup: the data producer sets up its signing key k and data consumer sets up its verification key k0 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, k0): the data consumer uses key k0 to verify the signature sig of received data D to ensure its origin, and rejects the data if the verification fails.

2. Provnenance Encoding and Decoding

In provenance encoding strategy whereby each node on the path of a data packet actually securely embeds the provenance information within a Bloom filter (BF) which will then be transmitted along with the data. As soon as we receive the packet, the Base Station extracts and verifies the provenance information provided. This paper also devises an extension of the provenance encoding scheme that allows the Base Station to detect if a packet drop attack was done by aay malicious node through the transmission path. For a data packet, provenance encoding refers to generating the vertices in the provenance graph and inserting them into the iBF.

Algorithm Used

1. Routing Algorithm*:*

Routing is the process of selecting best paths in a network. In the past, the term routing also meant forwarding network traffic among networks. However, that latter function is better described as forwarding. Routing is performed for many kinds of networks, including the telephone network (circuit switching), electronic data networks (such as the Internet), and transportation networks. This article is concerned primarily with routing in electronic data networks using packet switching technology.

This algorithm helps to find the connectivity between the Source and Destination. It checks for all nodes which are connected in network and retrieves to user. So, the user can know about the destination connectivity. In case there is no connectivity between the source and destination, it pops-up to the User. If connection is available, then the source can send the packets to destination through the corresponding path.

Some Common Thoughts

- The people in the digital world think that the data insecurity occurs only through hacking them.
- This is a great misconception.
- In fact, the biggest threat is that even when there is a tight security in the routing process by encoding and decoding, the data gets corrupted.
- This is because the data packets may go through various nodes when the attacker actually corrupts the original data by inserting malicious packets.

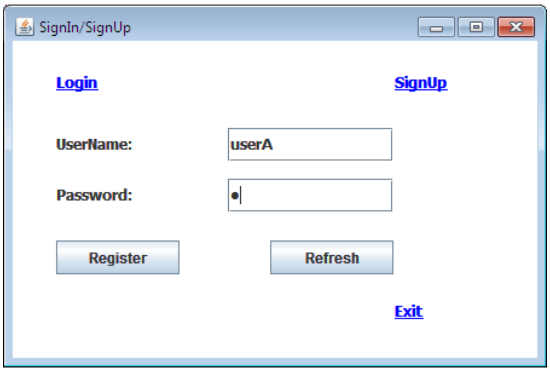
Key Generation algorithm:

Key generation algorithm is one of the algorithm used to generate an unique key to encrypt and decrypt the data. The same key has to be used both the sides to establish a relevant connection.

Modern cryptographic systems include symmetric-key algorithms (such as DES and AES) and public-key algorithms (such as RSA). Symmetric-key algorithms usually use a single shared key; which actually keeps the data secret requires keeping this key secret. Public-key algorithms use a public key and a private key. The public key is made available to anyone (often by means of a digital certificate). A sender encrypts data with the public key; only the holder of the private key can decrypt this data.

Since public-key algorithms tend to be much slower than symmetric-key algorithms, modern systems such as TLS and SSH use a combination of the two: one party receives the other's public key, and encrypts a small piece of data (either a symmetric key or some data used to generate it). The remainder of the conversation uses a (typically faster) symmetric-key algorithm for encryption.

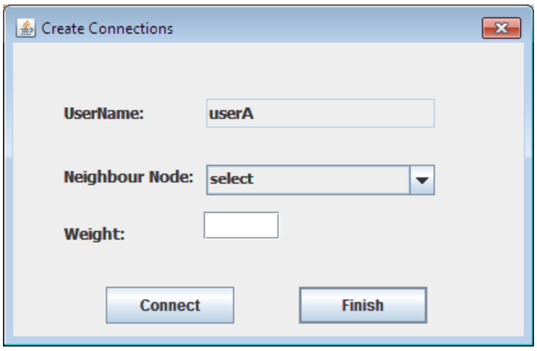
Experimental Setup and Result



UserName: userA

Password: •

Fig2: Register and Login form.

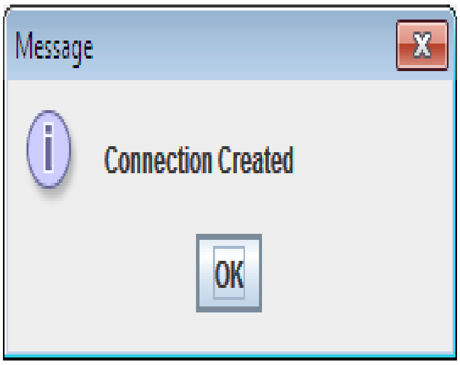


UserName: userA

Neighbour Node: select

Weight:

Fig3: Form to connect the nodes to create various paths.



Connection Created

Fig4: Dialog box for showing the connection is established.

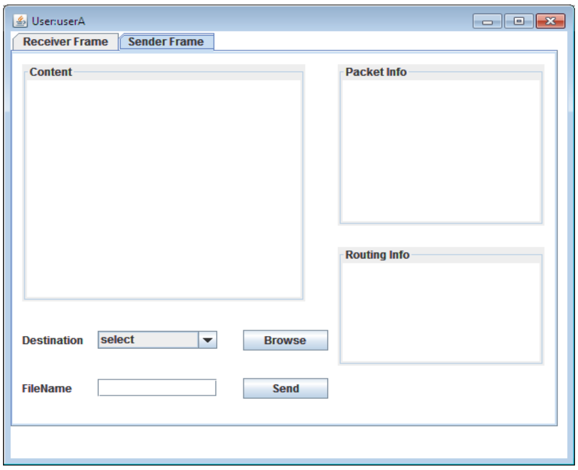
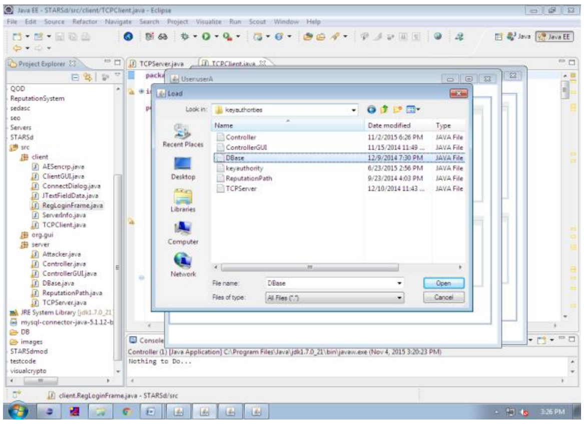


Fig4: User wise Sender and Receiver module to send or receive data.



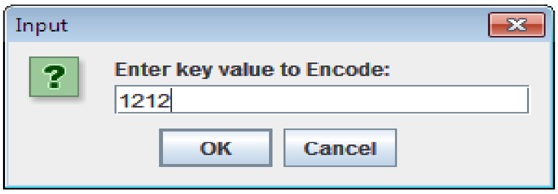
Name	Date modified	Type
Controller	11/2/2015 6:26 PM	JAVA File
ControllerGUI	11/15/2014 11:49 ...	JAVA File
DBase	12/9/2014 7:30 PM	JAVA File
keyauthority	6/23/2015 2:56 PM	JAVA File
ReputationPath	9/23/2014 4:03 PM	JAVA File
TCPServer	12/10/2014 11:43 ...	JAVA File

File name: DBase

Files of type: All Files (*.*)

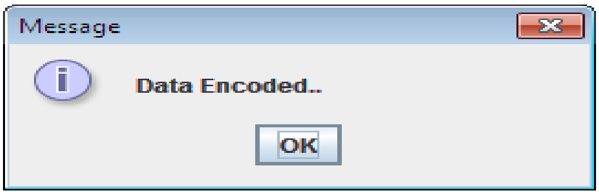
```
Controller (1) [Java Application] C:\Program Files\Java\jdk1.7.0_21\bin\javaw.exe (Nov 4, 2015 3:20:23 PM)
Nothing to Do...
```

Fig4:Select text file from the computer .



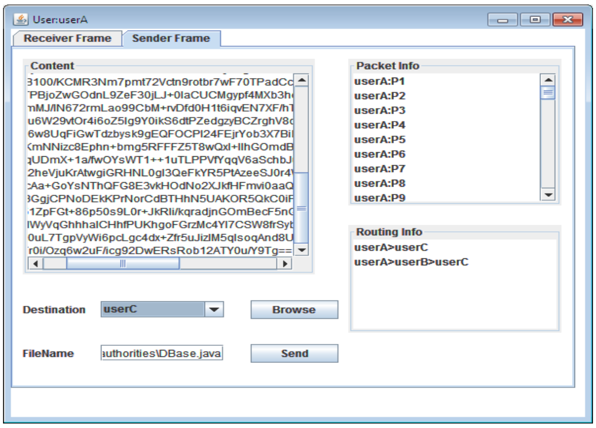
Enter key value to Encode: 1212

Fig7: Dialog box to encoding theory.



Data Encoded..

Fig8: Dialog box to show that the data has been encoded.



Content

```
3100/KCMR3Nm7pmt72Vctn9rotbr7wF70TPadCc
PBjoZwGOdnL9ZeF30jLJ+0IaCUCMgypf4MXb3h
nMJ/IN672rmLao99CbM+rvDfd0H1t6iqvEN7XF/hT
u6W29vtOr4i6oZ5Ig9Y0ikS6dtPZedgzyBCZrghV8
6w8UqFiGwTdzbysk9gEQFOCPI24FEjrYob3X7Bi
KmNNizc8Ephn+bmg5RFFFZ5T8wQxl+llhGOmdB
qUDmX+1a/fwOYsWT1++1uTLPPVfYqqV6aSchbJ
2heVjuKrAtwgiGRHNL0gI3QeFkYR5PtAzeeSJ0r4
Aa+GoYsNThQFG8E3vkHOdNo2XJkfHFmvi0aaQ
3GgjCPNoDEkKPrNorCdBTHhN5UAKOR5QkC0iF
1ZpFGt+86p50s9L0r+JkRli/kqradjnGOmBecF5n
IWyVqGhhhaICHhfPUKhgoFGrzMc4YI7CSW8frSy
0uL7TgpVyWi6pcLgc4dx+Zfr5uJizlM5qIsoqAnd8U
r0i/Ozq6w2uF/icg92DwERsRob12ATY0u/Y9Tg==
```

Packet Info

userA:P1
userA:P2
userA:P3
userA:P4
userA:P5
userA:P6
userA:P7
userA:P8
userA:P9

Routing Info

userA>userC
userA>userB>userC

Destination userC

FileName authorities\DBase.java

Fig9: Receiver modules receive the data.

Conclusion

This paper addressed the problem of securely transmitting provenance for sensor networks, and proposed a light-weight provenance encoding and decoding scheme based on Bloom filters. The proposed system ensures confidentiality, integrity and freshness of provenance. Furthermore, the paper extended the scheme to incorporate data-provenance binding, and to include packet sequence information that supports detection of packet loss attacks. Till now we have come across the algorithms going to be used in the prevention of packet drops. Hence the data provenance will be used to detect packet drops and overcome it to enhance security.

References

1. A Lightweight Secure Scheme for Detecting Provenance Forgery and Packet Drop Attacks in Wireless Sensor Network Salmin Sultana, Gabriel Ghinita, Member, IEEE , Elisa Bertino, Fellow, IEEE .

2. 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.

3. Mohamed Shehab, Member, IEEE Computer Society

4. Foster, J. Vockler, M. Wilde, and Y. Zhao, "Chimera: A virtual data system for representing, querying, and automating data derivation," in Proc. of the Conf. on Scientific and Statistical Database Management, 2002, pp. 37–46

5. K. Muniswamy-Reddy, D. Holland, U. Braun, and M. Seltzer, "Provenance-aware storage systems," in Proc. of the USENIX Annual Technical Conf., 2006, pp. 4–4.

6. "A Survey of Data Provenance in e-Science Yogesh L. Simmhan" by Beth Plale and Dennis Gannon

7. K. Dasgupta, K. Kalpakis, and P. Namjoshi, “An efficient clustering based heuristic for data gathering and aggregation in sensor networks,” in Proc. of Wireless networks. Communications and Networking Conference, 2003, pp. 1948–1953.

8. R. Hasan, R. Sion, and M. Winslett, “The case of the fake picasso: Preventing history forgery with secure provenance,” in Proc. of FAST, 2009, pp. 1–14.

9. C. Rothenberg, C. Macapuna, M. Magalhaes, F. Verdi, and A. Wiesmaier, “In-packet bloom filters: Design and networking applications,” Computer Networks, vol. 55, no. 6, pp. 1364 – 1378, 2011.

10. Liu and P. Ning, “TinyECC: A configurable library for elliptic curve cryptography in wireless sensor networks,” in Proc. of IPSN, 2008, pp. 245–256.

11. S. Madden, J. Franklin, J. Hellerstein, and W. Hong, “TAG: A Tiny Aggregation Service for Ad-Hoc Sensor Networks,” ACM SIGOPS Operating Systems Rev., vol. 36, no. SI, pp. 131-146, Dec. 2002.