International Journal of Contemporary Research in Computer Science and Technology (IJCRCST) e-ISSN: 2395-5325
Volume 2, Issue 3 (March '2016)

ENHANCING SECURITY BY IDENTIFICATION OF PROVENANCE FORGERY AND PACKET DROP ATTACKS IN WIRELESS SENSOR NETWORKS

M.Linda Alice,
Assistant Professor,
Information Technology,
Velammal Institute of Technology ,Panchetti

S.Karpaga Jothi,
UG Scholars,
Information Technology,
Velammal Institute of Technology ,Panchetti

A.Sathya,
UG Scholars,
Information Technology,
Velammal Institute of Technology ,Panchetti

Abstract: Sensor networks are deployed in numerous application domains, and the data which are collected used in decision making for critical infrastructures. Intermediate nodes are used in data streaming. Data provenance allows the Base Station (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. Hence, it is necessary to address security requirements such as confidentiality, integrity and freshness of provenance. Our goal is to identifying the provenance forgery and packet drop attacks in wireless sensor networks by designing a provenance encoding and decoding mechanism that satisfies such security and performance needs and to rectify them in by sending an email to trusted admin so that administrative user takes the control.

Keywords: provenance, security, sensor networks.

I.INTRODUCTION

Sensor networks , such as cyber physical infrastructure systems, environmental monitoring, power grids, etc.The diversity of data sources creates the need to assure the trustworthiness of data, such that only trustworthy information is considered in the decision process. A multihop wireless sensor network, consisting of a number of sensor nodes and a base station that collects data from the network. Data provenance is an effective method to assess data trustworthiness, since it summarizes the history of ownership and the actions performed on the data. Our goal is to design a provenance encoding and decoding mechanism that satisfies such security and performance needs. This system also does not address security concerns and is specific to some network use cases. SNP extends network provenance to adversarial environments. Since all of these systems are general purpose network provenance systems, they are not optimized for the resource constrained sensor networks. It proposes a chain model of provenance and ensures integrity and confidentiality through encryption, checksum and incremental chained signature mechanism. It extends this method by applying digital signatures to a DAG model of provenance. 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. Provenance in sensor networks has not been properly addressed. To Overcome these drawbacks we enhanced a security based technique in wireless sensor networks. The technique is to propose 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. . We introduce 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. It proposes an in packet Bloom filter (iBF) provenance encoding scheme. From the security point of view, the in-packet Bloom filters act simultaneously as forwarding identifiers and forwarding capabilities. ; introducing a DoS resistant forwarding service. Capabilities enable secure statements attached to packets, allowing forwarding nodes to easily check if a packet has been approved by the receiver The design efficient techniques for provenance decoding and verification at the base station.

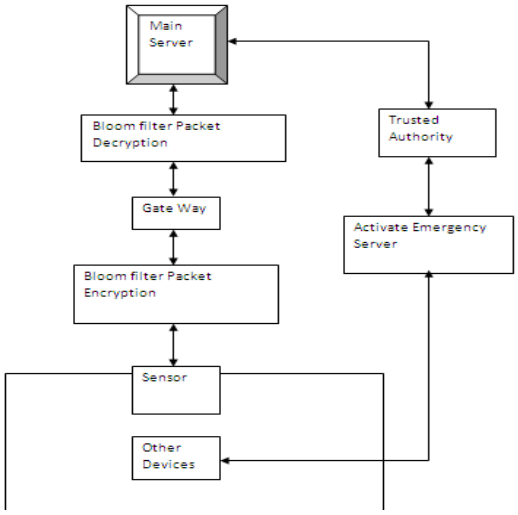
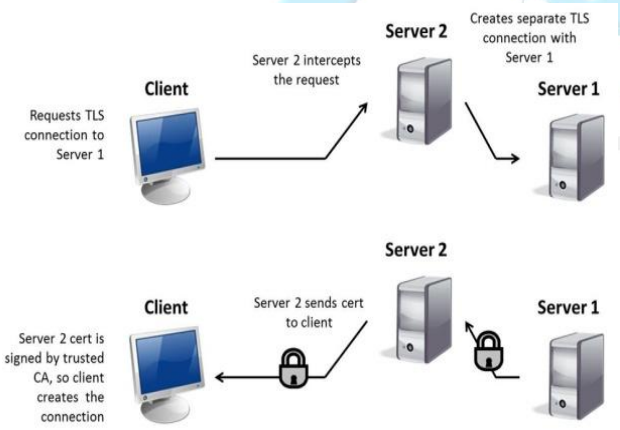


Figure 1: Architecture Diagram

II.CLIENT AND SERVER CONFIGURATION

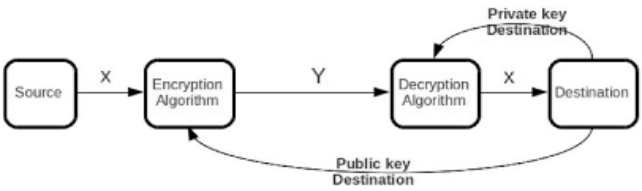
Client server configuration consists of two elements they are client and server. server is a centralized element which provides resources to the client. It manages data, printers ,network traffic. client manages local resources by users such as monitor, keyboard, cpu, pheripherals.we create wireless sensor network (wsn) ,using this network sensors can transmit data to observer via gate way. Gateway is intermediate server in between wireless sensor network and main server where it translates between different protocol and suites. Sensors send packet when detect abnormal condition, server take action based on that value.

To establish a client/server environment between two SAS sessions . the required SAS software products must be installed SAS/CONNECT® on the client SAS/CONNECT® or SAS/SHARE® on the server SAS/SECURE™ if encryption is required SAS/ACCESS® if the data access will be to non-SAS databases. the underlying network must be functional (outside of SAS) the client must be able to identify the server . both client and server must use the same communications protocol . the client must be able to start a SAS session on the server. Common Gateway Interface (CGI) program that resides on the web server.



Packet encryption:

We split large data into small packets, each packet contain sensor province, sensor value, sensor id . we have used bloom filter data structure for effectively data transmission between the source and destination , and also each packet contain unique identification number. That number must be encrypted before start transaction.



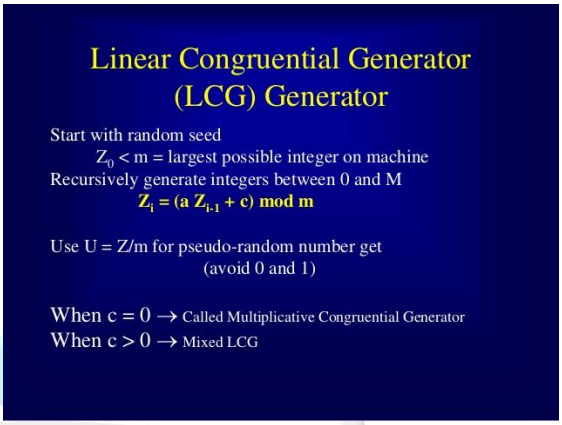
For encryption we use linear congruential generator.

Linear congruential generator:

An LCG is essentially a formula of the following form:

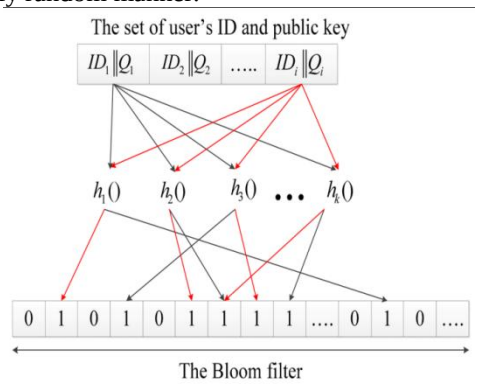
$$\text{Number} = (a * \text{number} + c) \bmod m$$

In other words, we begin with some start or "seed" number which ideally is "genuinely unpredictable", and which in practice is "unpredictable enough". For example, the number of milliseconds— or even nanoseconds— since the computer was switched on is available on most systems. Then, each time we want a random number, we multiply the current seed by some fixed number, *a*, add another fixed number, *c*, then take the result modulo another fixed number,*m*. The number *a* is generally large.



Packet split using bloom filter:

Bloom filters are serializable. Bloom filter is used to reduce the size of MACs and ensure their security.simple, space-efficient probabilistic data structure that succinctly represents a set in order to support membership queries. a Bloom filter is implemented as a bit-array of m bits associated with h different hash functions, each of which maps a set element to one of the m array positions in a uniformly random manner.

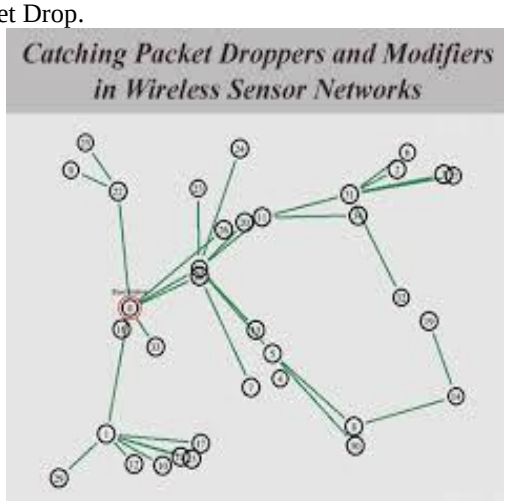


All bits in an initial Bloom filter are set to 0, standing for an empty set. To insert an element u into a set represented by a Bloom filter BF, h array positions are calculated by hash functions on u and the bits at those positions in BF are set to 1. Correspondingly, when it is required to check the

membership of an element v within the Bloom filter BF, supplying v to hash functions outputs h array positions; if any of the bits at the h positions is 0, then element v does not belong to the set; otherwise, theelement is claimed to be a member of the set. we derive the advantage of an adversary against secure Bloom filter with parameters (m, h). Intuitively, for 80-bit security, the selections of (m, h) should satisfy Adv(m, h) ≤ 2 −80. Without loss of generality, we suppose that adversary A is required to output a secure Bloom filter BFA which should be identical to BFu,e for node u claiming event Ne. Let p1(i) be the probability of the number of bit one in BFu,e being i. Obviously p1(i) is none-zero only if 1 ≤ i ≤ h. Based on the pseudo-randomness of cryptographic hash function, we have p1(i) = m i τ (h, i) mh , where τ (h, i) is equal to the output of the following problem: How many distinct words we can get if we use i different letters to construct words of length h while it is required that all i letters are used. If HMAC is secure, then the adversary only can output random BFA with the number of bit one that he selects. When the bit one number of bloom filter u,e is i, the probability that a random BFA with i bits of one equals BFu,e is then p2(i) = 1 m i . Subsequently, the overall advantage that adversary can achieve is Adv(m, h) = max 1≤i≤h p1(i)p2(i) = m−h max 1≤i≤h τ (h, i) . Consider τ (h, i) in the context of the word-counting problem, it is easy to get the recursive formula: τ (h, i) = (i h − Pi−1 j=1 i j τ (h, j) if i > 1 1 if i = 1 . Bloom filters allow false positives but the space savings often outweigh this drawback when the probability of an error is made sufficiently low.

III.PACKET DROP DETECTION

Packet loss can be caused by congestions due to heavy traffic, collisions at link layer, buffer overflows, etc. In WSNs, congestions and buffer overflows seem unlikely to happen because of low traffic rates.The packet drops are identified easily in the Main Server , when the hacker alters the packet data in the Gate Way then the value of the packets will be altered.Once the data is transmitted to the Main server from the Gate Way, then transmitted packets are decrypted. During the decryption of the packets if there are any alteration in the packet in the Gate Way there will be the Packet Drop.



When the packets are determined as hacked packets the information is sent to the trusted authority.The trusted authority in turn activate the emergency server.The emergency server checks the interconnected devices with the sensor in the room and turn it to the actual state.

Detection of a packet-dropping path.

```
Detect_compromised_path(s, d)
Begin
    Get ns,nr;
     While(True)
      if ns - nr > 0 then
Guess that packets are being dropped by malicious nodes on
the source-to-destination path.
return TRUE
else
return FALSE
Wait till next verification cycle.
End.
```

IV.CONCLUSION

We 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 scheme ensures confidentiality, integrity and freshness of provenance. We extended the scheme to incorporate data-provenance binding, and to include packet sequence information that supports detection of packet loss attacks. Experimental and analytical evaluation results show that the proposed scheme is effective, light-weight and scalable. In future work, we plan to implement a real system prototype of our secure provenance scheme, and to improve the accuracy of packet loss detection, especially in the case of multiple consecutive malicious sensor nodes.

V.REFERENCES

[1]. karpaga Jothi S, and Sathya A "Enhancing Security By identification of provenance forgery and packet dropping attacks in wireless sensor networks.

[2]. H. Lim, Y. Moon, and E. Bertino, "Provenance-Based Trustworthi-ness Assessment in Sensor Networks," Proc. Seventh Int'l WorkshopData Management for Sensor Networks, pp. 2-7, 2010.

[3]. S. Sultana, E. Bertino, and M. Shehab, "A Provenance BasedMechanism to Identify Malicious Packet Dropping Adversariesin Sensor Networks," Proc. Int'l Conf. Distributed ComputingSystems (ICDCS) Workshops, pp. 332-338, 2011.

[4]. Sensor Hierarchical In-Network Aggregation in Sensor Networks .H.chan , A.Perrig and D.song in 2006.