Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 109C (2017) 553–559

Procedia Computer Science

www.elsevier.com/locate/procedia

The 8th International Conference on Ambient Systems, Networks and Technologies (ANT 2017)

An Efficient Broadcast Authentication Scheme in Wireless Sensor Networks

Bacem Mbarek[a], *Aref Meddeb*[b], *Wafa Ben Jaballah*[c] *and Mohamed Mosbah*[d]

a National Engineering School of Tunis, University of Tunis El Manar, Tunis, Tunisia, Email:bacem.mbarek1@gmail.com
b National Engineering School of Sousse, NOCCS Laboratory, University of Sousse, Tunis, Tunisia
c Orange, France
d LaBRI, Bordeaux INP, University of Bordeaux, CNRS, France

Abstract

Broadcast source authentication is a challenging topic in wireless sensor networks. This security service allows senders to broadcast messages to multiple receivers in a secure way. Although several authentication mechanisms have been proposed to address the need for security in WSNs, most of them are resource consuming and are inadequate for constrained environments. In this paper, we shed the light to the security vulnerabilities of symmetric key based authentication mechanisms, and their inability to tackle memory DoS attacks. Moreover, we provide a new efficient broadcast authentication scheme based on a Bloom filter data structure in order to reduce the communication overhead. Finally, we run a thorough set of simulations to assess the efficiency of our approach compared to some state of the art solutions in terms of energy consumption, communication and computation overhead. Our results provide insight into the suitability of our approach for use in WSNs.

Peer-review under responsibility of the Conference Program Chairs.

Keywords:
Source Authentication, Broadcast authentication, Wireless Sensor Networks, Bloom filter

1. Introduction

Wireless sensor networks (WSNs) are frequently used for data gathering applications, such as military sensing and tracking, environment monitoring, patient monitoring, etc. WSN is in general, more vulnerable to attacks and unauthorized access than traditional (wired) networks. The sensor nodes are often deployed in hostile environments where they can be easily captured, compromised, or manipulated by an adversary. Therefore, the security becomes extremely important that provide confidentiality and authentication are critical for the operation of many sensor ap-

* B.Mbarek, National Engineering School of Tunis, University of Tunis El Manar, Tunis, Tunisia, Email: bacem.mbarek1@gmail.com
E-mail address: author@institute.xxx

Peer-review under responsibility of the Conference Program Chairs.
10.1016/j.procs.2017.05.340

plications. For this reason, the implementation of secure techniques in WSN are an important research topic[1,2]. Networks should collect data from the sensors for long periods of time without requiring human intervention. In addition, it could be impossible or inconvenient to recharge the battery, because nodes may be deployed in a hostile or unpractical environment. The sensors must be low in cost, thus will have constrained battery power, limited storage and low computational capacity[3]. Due to these constraints it is difficult to directly employ the existing security approaches to the area of WSNs. Therefore, security protocols for WSNs are focused on conquest of these constraints. In this work, we focus on broadcast authentication as it is a fundamental security service that enables a sender to broadcast critical data to receiver nodes in an authenticated way such that an attacker is unable to forge broadcasted messages.

Due to their inherent limitations, WSNs are especially sensitive to severe Denial of Service (DoS) attacks[3,4,5]. Compared to traditional networks, a WSN is more resource constrained, subject to open wireless communication, and prone to the physical risks of in-situ deployment. These factors increase the susceptibility of WSNs to DoS attacks. An adversary could either execute signal jamming attack; or overwhelm nodes to quickly exhaust their energy, communication bandwidth, memory and CPU of sensor nodes. In this paper, we address the vulnerability of WSNs to DoS attacks when providing authentication mechanisms. We propose a novel bloom filter scheme, for broadcast authentication protocols in wireless sensor networks. The novel bloom sheme reduce the hash function colLision by using a collision resolution scheme for the values stored in the filte, and have a very hight resistant against collision attaks that the attacker could muster. These protocol implementations are evaluated and validated in terms of authentication delay, authentication probability, resilience against DoS attacks, memory and energy consumption overhead.

The remainder of this paper is organized as follows. Section 2 describes work related to broadcast authentication and presents various authentication protocols used for WSN. In Section 3, we present our solution, and Section 4 provides a thorough performance evaluation. Finally, Section 5 concludes the paper.

2. RELATED WORK

Many secure broadcast authentication based schemes have been proposed for resource constrained networks[6,3,7,4,8,9]. Broadcast authentication schemes could be classified into three groups based on the main cryptographic primitive employed: (1) Message Authentication Code (MAC), (2) signature amortization and (3) one-time signature.

Protocols in the first group are symmetric authentication schemes such as TESLA[10], its simplified version for resource limited networks μTESLA[11], and the enhancements of μTESLA such as[7]. These schemes provide broadcast authentication by using MACs and require time synchronization between the nodes and the sink. Moreover, these schemes are vulnerable to DoS attack. Another shortcoming of μTESLA is the difficulty of establishing the initial trust between the nodes and the sink.

Schemes in the second group of broadcast authentication protocols employ signature amortization. One of the first protocols in this group is SAIDA [12]. This protocol is not robust against false packet injection and packet modification attacks. The designers of SAIDA have proposed Reed Solomon codes to handle the packet modification attack. However, this kind of coding is too complex for the low-power processor of the nodes. The one-time signature BiBa[13] and an improvement of BiBa, called HORS[14], are among the schemes in the second group. The major drawback of using one-time signature schemes in wireless networks is that the public key has to be frequently updated to maintain security. This requirement significantly adds to the communication overhead of the protocol. Moreover, broadcast authentication schemes based on one-time signatures are not suitable for designing node-to network multihop broadcast protocols. Broadcast communications of any node has to be handled by the sink as an intermediary.

The third category employs symmetric keys[4,8,15], however it implements time asymmetry to tackle the source authentication problem. An efficient time asymmetry scheme based on key disclosure is μTESLA[11], a simplified version of TESLA[10]. The main idea is to use key disclosure delay to keep the authentication key secret until the expiration of a given time interval, and then it will be disclosed. This approach is referred to as temporal asymmetry authentication TESLA[7]. Such approaches are adequate for non real time applications in which the actual reception of the packet and its verification depend on key disclosure delay. All these schemes use MACs and require time synchronization between the nodes and the sink. These schemes are vulnerable to DoS attack and exhibit some difficulties to establish initial trust between the nodes and the sink.

Because of their adequacy to WSNs in terms of complexity, in this paper, we focus on the third category (temporal asymmetric approaches)[4,8]. Our objective is to reduce the authentication delay of such approaches without adding complexity. We aim to minimize the impact of DoS attacks by reducing in the receiver buffer the delay of forged packets. Essentially, we want to decrease the computational overhead and communication error rate of the authentication protocol by using an efficient XOR-based Bloom filter data structure. Specifically, multiple hash digests are XORed and utilized as the index to search within the Bloom Filter. We show that our proposal outperforms the Staggered-TESLA schemes.

In this Section we present an overview of Bloom Filter and Staggered TESLA on which we based our contribution.

2.1. Bloom filter

A Bloom filter is an array of m bits for representing a set $S = \{x_1, x_2, x_3, ..., x_n\}$ of n elements, initially all set to 0. Each Bloom filter needs k independent hash function h_1,..., h_k [16]. These k hash functions range between 0 and $m-1$ and each element is mapped to $[0, ..., m-1]$. For each element x in S, the $h_i(x)$ bits are set to 1 for $1 \leq i \leq k$.

To verify the presence of y in S, we test whether all bits $h_i(y)$ are set to 1. If yes, y is assumed to be a member of S. If not, y is not a member of S. As a consequence, the Bloom filter gives a false positive suggesting that y may be in S infact it is not. However, the Bloom filter does not suffer from false negatives i.e., a negative presence of an object is always confirmed.

2.2. Staggered TESLA

The Staggered TESLA is an extended version of TESLA proposed in [17]. The Staggered TESLA reduces the delay of forged packets within the buffer of the receiver by splitting the time into equal intervals. An authentication key corresponds each time interval for all the generated packets in that interval. We derive all the keys a public one-way function. As a matter of fact, the Staggered TESLA utilizes various MACs from successive TESLA keys. As a consequence, most malicious nodes can not counterfeit every generated MACs.

The key disclosure delay is presumed to be $d = 3$ time intervals, like in [8]. Adversaries are less likely to forge the MACs constructed utilizing both K_{i-1} and K_{i-2} keys rather than those generated using K_i. The scheme that uses MACs from successive TESLA keys is referred to as Staggered TESLA. The format of the j^{th} packet transmitted to the interval i can be:

$$\left\{ \begin{matrix} MAC(K_i, M), MAC(K_{i-1}, M), \\ ..., MAC(K_{i-d-1}, M) \end{matrix} \right\} \tag{1}$$

We identify two problems with the Staggered TESLA. First, it necessitates additional computation and communication with the objective of verifying and transmitting the additional MACs compared with the traditional TESLA. Even if the MACs are based on symmetric cryptography making them computationally efficient, in some applications where several broadcasts may be needed (such as emergency situations), this computation cost may become prohibitive. Second, the Staggered TESLA is vulnerable to replay attacks. e.g, an adversary is able to put in a j packet from an i interval, and surely speaking conterfeit an $i+1$ interval.

3. OUR ENHANCED AUTHENTICATION MECHANISM

The Broadcast authentication scheme we propose is based on an *Enhanced Bloom Filter* in order to reduce the authentication delay of data packets, minimize computational and communication costs, and reduce collision probability. The main idea is to reduce the computational overhead and communication error rate of the authentication protocol by using an XOR-Based Bloom filter data structure. Specifically, multiple hash digests are XORed and used as the index to search in a Bloom Filter.

3.1. Overview

In Staggered-TESLA, the sender generates d message authentication codes in time interval T_i, and creates the set S. The format of each send packet is as follows.

$S =< PMAC(K_i, P), PMAC(K_{i-1}, P), ..., PMAC(K_{i-t-1}, P) >$

where $PMAC$ is the packet authentication code (PMAC) of the packet P, and K_i is the commitment key.

The sender utilizes multiple commitment keys in order to be more secure against denial of service attacks. In order to improve the performance of Staggered-Tesla protocol, we utilize a new Bloom filter scheme based in XOR logical function. Our proposed scheme can defeat the drawback of Staggered TESLA protocol in terms of communication overhead, and resilience to DoS attack.

3.2. Description

In order to reduce the communication overhead, we use a Bloom filter vector. In fact, the data packet P corresponding to a message M in time interval $T_{i,j}$ is then constructed as follows: $P = PMAC_{i,j}|BF_{i,j}$, where BF is the commitment key filter, d keys are mapped to a Bloom filter vector $BF_{i,j}$.

We have to map each key $K_{i,j}$ to an m-bit vector with $BF = b_0, b_1, ..., b_m$. Consequently, to reduce the size of Bloom filter, we can have $m < d \times |key|$.

Collision rate. It is possible for a hash function to hash two different items into the same slots (bit positions) and cause a collision, for which it wrongly concludes that an item belongs to a data set, while it actually does not. This occurs because all bits related to the item were previously set to 1 by other items in the data set. Often many correct elements are inserted in the BF but they are considered as false elements because they have a hight collision rate with other elements. Thus, their probability of false positive becomes very high. For this reason we propose a new Bloom Filter vector based in XOR logical function in order to reduce the collision rate.

We use the XOR operation between two encoding values that have been mapped to the same cell in the bloom filter. If one collision occurs in a cell, then the input encoding value is XORed with the previous result of BF. The XOR procedure for a set $S = \{e_1, e_2, ..., e_n\}$ with n elements is as follows.

1. Collision detection: To detect a collision in a cell we use a counter for each cell where 0 means that the cell is empty while 1 means one encoding has been inserted. If the value of 1 occurs more than once in the same cell, it means a collision has been detected in this cell. The counter for each cell is then increased. The first step is to find k hashed cells. The counter of each cell is increased. The second step is to call XOR operation if two encodings are mapped to the same cell.
2. XOR operation: If two MACs hashed in the same cell, a collision will occur. An XOR operation will then be performed between the two encoded values that have been mapped to the same cell in the BF i.e., $b[e_i] \oplus b[e_j]$; where $b[e_i]$ is the encoding value of each inserted element and $B[e_{i-1}]$ is the previous result of the BF.
3. Testing if an element belongs to the BF: The receiver node performs the same operation as the sender with the received elements. Then it will compare between the BFs obtained by the sender and the receiver nodes. If these are equal, then the BF is correct.

4. Performance Evaluation

We used the NS-2.35 simulator [18] to evaluate the performance of Staggered TESLA, Staggered TESLA with Bloom Filter, and XOR Bloom Filter (XBF). We evaluated them in terms of energy consumption, memory overhead, authentication delay, and detection probability of forged filter. We focus on the evaluation of the broadcasted data packets. The simulation was run for $100s$. Since the simulated key disclosure delay was 4 intervals, the maximum number of MACs that could be employed in each packet is 3. To evaluate the four metrics mentioned above, we measure the node energy cost needed to authenticate the source of a transmitted packet, we compute the number of packets in the receivers buffer and calculate the proportion of packets in the buffer that originated from the source, and we compute the average authentication delay. The hash functions used in this work are a member of the class of universal hash functions [19]. Thus, in our implementation, we have selected MD5, SHA1, and CRC32.

4.1. Energy consumption

We now evaluate the energy consumption of Staggered TESLA with and without Bloom Filter for various network sizes. Figure 1 gives the average energy consumption in Joules needed to authenticate a broadcast message. Clearly,

the Bloom-filter-based scheme consumes a much lower energy. For example, when the network size is 400 nodes, the Bloom filter based scheme always costs $20mJ$. Without Bloom filter, the cost is nearly twice that value i.e., $40mJ$. This may be explained by the fact that in order to authenticate a broadcast message with Staggered TESLA, the sender needs to generate and transmit d MACs for each data packet in time interval T_i , and every sensor node should retransmit the message once. Conversely, the Bloom-filter-based reduces the size of large data sets containing unique identifiers. For each packet, bloom filters generated and concatenated d keys in a single filter. Therefore, we have $m < d \times |key|$ and we can reduce the filter size. These factors can reduce costly radio transmissions. We can also notice that XOR operation incur an additional energy consumption but this is worthy as it is the price to pay in order to significantly increase the detection probability of forged filters, as we will see below.

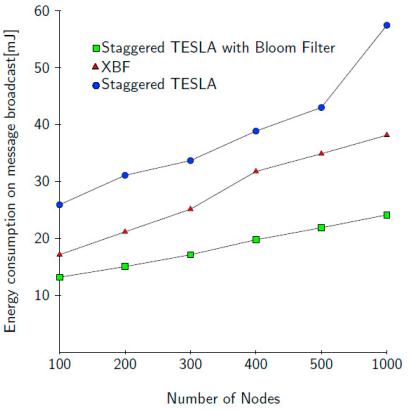

Fig. 1: Energy consumption

4.2. Memory overhead

Figure 2 represents the maximum memory usage for the three approaches in the presence of DoS attacks. We can notice that Bloom Filter exhibits a low memory usage since the verification of the MAC is done frequently i.e., for each packet, we verify the filter size and the probability of false positive, as opposed to Staggered TESLA which waits for all packets of a given message to start authenticating them. Hence, with BF, it is not possible for an adversary to inject malicious packets without being detected. In fact, when an adversary either attempts to inject bogus packets to the network or drop legitimate packets, its worst impact would be increased latency. We can also notice that the XOR operation does not incur a significant cost in terms of memory overhead compared to the simple Bloom Filter approach

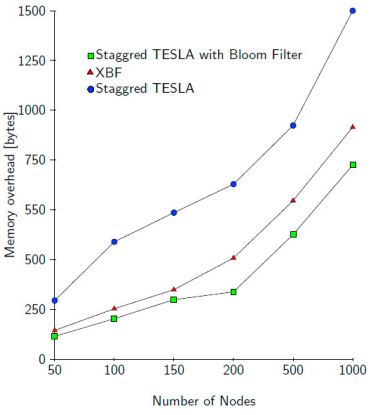

Fig. 2: Memory overhead

4.3. Detection probability of forged filter

Figure 3 depicts the results for the probability of detecting a forged filter with the three broadcast authentication approaches. The comparison between the scenarios with and without Bloom Filter shows that the probability of detecting a forged filter is lower with Bloom filter. This may be explained by the fact that with Bloom filter, a forged packet is quickly detected if the size of the filter is not equal to the defined filter. We can also notice that the XOR operation significantly improves the detection probability of forged filters compared to simple Bloom filter approach, especially when the number of forged filters is high i.e., above 150. For example, for 200 forged filters, we detect 99.7% of them while with simple Bloom filter, only 95.4% of the forged filters are detected. Further, only 80% of the 200 forged filters are detected without bloom filter.

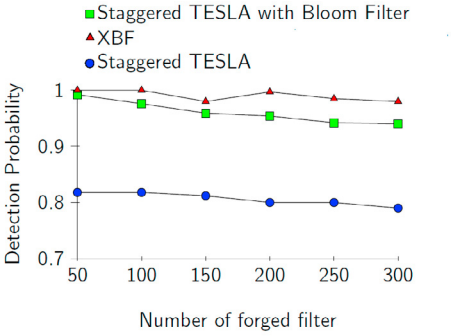

Fig. 3: Detection probability

5. Conclusion

In this paper, we studied the problem of broadcast source authentication in WSN. We proposed a new broadcast authentication mechanism based on XOR operations applied to Bloom filter, referred to as a XOR Bloom Filter Based Authentication (XBF). Our solution reduces the error rate of Bloom filter by using simple XOR operations between hash function values. Our approach also detects DoS attacks by including Staggered Tesla into the Bloom filter. We have shown through analytical and simulation results that XBF provides significant gains in terms of detection probability of forged packets while being comparable to simple Bloom Filter in terms of memory overhead and authentications delay but with slightly additional energy consumption. This additional energy is worthy since it is the price to pay in order to detect forged packets, a key security feature in most mission critical broadcast applications.

References

1. W. B. Jaballah, M. Mosbah, H. Youssef, and A. Zemmari, "Lightweight secure group communications for resource constrained devices," *International Journal of Space-Based and Situated Computing*, vol. 5, no. 4, pp. 187–200, 2015.
2. W. B. Jaballah, M. Mosbah, and H. Youssef, "Performance evaluation of key disclosure delay-based schemes in wireless sensor networks," *IEEE Pervasive Computing and Communications Workshops (PERCOM Workshops)*, pp. 566–571, 2013.
3. M. Jan, P. Nanda, M. Usman, and X. He, "Pawn: a payload-based mutual authentication scheme for wireless sensor networks," *Concurrency and Computation: Practice and Experience*, 2016.
4. B. Mbarek, A. Meddeb, W. B. Jaballah, and M. Mosbah, "A secure authentication mechanism for resource constrained devices," *IEEE/ACS, 12th International Conference of Computer Systems and Applications (AICCSA)*, pp. 1–7, 2015.
5. W. B. Jaballah, A. Meddeb, and H. Youssef, "An efficient source authentication scheme in wireless sensor networks," *IEEE/ACS International Conference on Computer Systems and Applications (AICCSA)*, pp. 1–7, 2010.
6. N. R. Potlapally, S. Ravi, A. Raghunathan, and N. K. Jha, "Analyzing the energy consumption of security protocols," in *Proceedings of the international symposium on Low power electronics and design*, 2003, pp. 30–35.
7. V. Khanaa, K. Thooyamani, and R. Udayakumar, "A secure and efficient authentication system for distributed wireless sensor network," *World Applied Sciences Journal (Computer Sciences, Engineering and Its Applications)*, pp. 304–308, 2014.
8. W. B. Jaballah, M. Mosbah, H. Youssef, and A. Zemmari, "Lightweight source authentication mechanisms for group communications in wireless sensor networks," *IEEE Advanced Information Networking and Applications (AINA)*, pp. 598–605, 2013.

9. M. Medwed, "Iot security challenges and ways forward," in *Proceedings of the 6th International Workshop on Trustworthy Embedded Devices*, 2016, pp. 55–55.
10. A. Cárdenas, S. Radosavac, J. S. Baras *et al.*, "Performance comparison of detection schemes for mac layer misbehavior," in *IEEE 26th International Conference on Computer Communications (INFOCOM 2007)*, 2007, pp. 1496–1504.
11. A. Perrig, R. Szewczyk, J. Tygar, V. Wen, and D. E. Culler, "Spins: Security protocols for sensor networks," *Wireless networks*, vol. 8, no. 5, pp. 521–534, 2002.
12. J. M. Park, E. K. Chong, and H. J. Siegel, "Efficient multicast stream authentication using erasure codes," *ACM Transactions on Information and System Security (TISSEC)*, vol. 6, no. 2, pp. 258–285, 2003.
13. A. Perrig, "The biba one-time signature and broadcast authentication protocol," in *Proceedings of the 8th ACM conference on Computer and Communications Security*, 2001, pp. 28–37.
14. L. Reyzin and N. Reyzin, "Better than biba: Short one-time signatures with fast signing and verifying." Springer, 2002, pp. 144–153.
15. B. Mbarek, A. Meddeb, W. B. Jabalah, and M. Mosbah, "A broadcast authentication scheme in iot environments," in *IEEE/ACS 13th International Conference on Computer Systems and Applications (AICCSA)*. IEEE, 2016, pp. 1–6.
16. B. H. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Communications of the ACM*, vol. 13, no. 7, pp. 422–426, 1970.
17. Q. Li and W. Trappe, "Staggered tesla: a multicast authentication scheme resistant to dos attacks," in *Global Telecommunications Conference (GLOBECOM'05)*, vol. 3, 2005, pp. 6–pp.
18. I. T. Downard, "Simulating sensor networks in ns-2," DTIC Document, Tech. Rep., 2004.
19. A. Ostlin and R. Pagh, "Uniform hashing in constant time and linear space," in *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, 2003, pp. 622–628.