OPEN ACCESS Freely available online

Journal of
Information Technology & Software Engineering

Research Article

Provide a New Algorithm for Encrypting Packets in Wireless Sensor Networks and Authenticating Network Members Using the Bloom Filter

Amir Khaki*, Hamidreza Kermajani*

Department of computer Engineering, Tuyserkan Branch, Islamic Azad University, Tuyserkan, Iran

ABSTRACT

"Network security" and "information security" are mostly used interchangeably. In fact, as the users and staffs do not know much about the security, intruders can easily enter a computer network reaching secret information. Due to the increasing requirement of users to computer networks, the present study aims to present the algorithms and methods to enhance the security of the networks. As a matter of fact, it is tried to increase the security of sending packages according to cryptography algorithms and decrease the package size by compression method. Therefore, traffic load in connection lines will decrease. Besides, authentication is done by bloom filter in this study. It is expected to guarantee the security of sending packages banning intruders to enter the system.

Keywords: Network security; Information security; Cryptography; Intruders

INTRODUCTION

Wireless sensor networks are made of several small sensors spread randomly in the environments which are not easy to reach [1]. They can sense the nearby events. Having a special range, each sensor has primary energy to not only collect data but also send to the destination. These sensors often have little energy and are able to calculate light computations and common processing. Therefore, when they are not working, they can be in on/off mood to save energy. However, it is different for wireless sensor networks. In fact, in addition to energy consumption, the security of the data and authentication of sender and receiver is of considerable importance.

As these networks use wireless connection, it is possible for a person or machine to pretend as a valid member in the network being able to steal or change the data. Therefore, the security of wireless networks is the issue which has recently attracted the attention of researchers. The present study aims to find a way to keep these networks safe. In fact, the techniques of data cryptography and authentication of sender and receiver are applied to protect the networks against threats. As a result, the combination of ASE cryptography algorithm and authentication by bloom filter is used.

Part 2 reviews previous studies about wireless sensor networks. Part 3 is dedicated to the algorithm of cryptography and bloom filter. Part 4 defines algorithm. The findings from algorithm simulation are presented in part 5, and finally conclusion is reported in part 6.

RELATED WORK

The security of sensor networks is an important issue which has attracted the attention of many researchers. Majority of the available techniques depend on the what the organizations require. Basically, these techniques are mostly defined by the available security schemes which are presented here:

Secure sensor node authentication in wireless sensor networks

Using a safe medium by the authority of network, the identity is downloaded by the sensor node. Therefore, the node registration is completed by base station. Then, BS or other nodes identify the node. In fact, authentication is needed for further communication. As a result, authentication can be received leading to the development of the session key of the parties. In this protocol, the authentication request which has been sent by A is accepted by B or BS [2,3].

Secure sensor node authentication in wireless sensor networks

The aforesaid mechanism is used to verify the sensor nodes ensuring the correctness of all the nodes in the network. Monitoring the network, this verification just allows the authorized nodes access the network. Therefore, the intruder is banned to get the secret messages. This studies include the new mechanism of node authentication.

Most attacks, like cloning attack, replay attack, man-in-the-middle attack and node capture attack, often target the nodes of WSN. In

***Correspondence to:** Amir Khaki and Hamidreza Kermajani, Department of computer Engineering, Tuyserkan Branch, Islamic Azad University, Tuyserkan, Iran, E-mail: Amirkhaki64@gmail.com, hr.kermajani@gmail.com

Received: September 10, 2019; **Accepted:** September 23, 2019; **Published:** October 01, 2019

Citation: Khaki A, Kermajani H (2019) Provide a New Algorithm for Encrypting Packets in Wireless Sensor Networks and Authenticating Network Members Using the Bloom Filter. J Inform Tech Softw Eng 9:258. doi: 10.35248/2165-7866.19.9.258

fact, an adversary can easily trap the node and reach the secret data. This program is supposed to certify the proving nodes by several challenges which are provided by the verifier. Therefore, just true nodes are allowed to solve the challenges and reach the network. This scheme is secured.

Authentication framework for WSN using identity-based signatures

This scheme leads to the following findings:

- The entire nodes inside the system make this mechanism hold active broadcast and multicast authentication. Moreover, it designs a secure network and shoots the troubles like disaster handling, environmental monitoring and traffic control beyond the involvement of the BS.
- An active and safe identity-based authentication proposal is suggested to develop BS involvement [4].

Authentication simple, secure, efficient, lightweight and token based protocol for mutual authentication in wireless sensor networks

The aforesaid protocol incudes two phases:

- Registration phase: Here, BS sends the token to sensor node containing Pseudo random number NumA. Therefore, authorized entity is just able to make the token.
- Authentication phase: authentication request AR is sent by a including not only the token but also hash of NumA which is supposed to be authenticated by B. Both A and B follow the similar method. In other words, A and B are both authenticated mutually and their hashes are compared. In case it matches, the token is accepted. If not, authentication is not successful [4].

Security improvement on biometric based authentication scheme for wireless sensor networks using fuzzy extraction

Choi et al. suggested biometric-based authentication system being susceptible to the attacks of stolen smart card and user impersonation **[5].**

Three-factor user authentication and key agreement using elliptic curve cryptosystem in wireless sensor networks

Park et al. proposed the attacks of user impersonation and some security drawbacks in the revocation/reissue phase Choi et al. scheme [6].

Improving biometric-based authentication schemes with smart card revocation/reissue for wireless sensor networks

Moon et al. determine impersonation attack and showed the weakness of smart card revocation/reissue phase of Park et al.'s scheme [2,3,7-10].

THE ACKNOWLEDGMENT OF BLOOM FILTER AND CRYPTOGRAPHY ALGORITHM

The acknowledgment of bloom filter

An empty bloom filter contains m bit array whose elements are all zero. Besides, a number of functions K are available in which each number peers to a cube of bloom filter array. The following operation can be done in bloom filter:

Element insertion: As the first step to insert an element, the pointed number is tangled to function K. Number of K determines the situation in the array for this number. The array in the specified situations is changed to one. Finally, the intended number is inserted in bloom filter.

In the picture Figure 1, as an example, K equals 3 (K=3). Each one of X, Y, and Z is sent to 3 functions of hash. Therefore, tangled function becomes an index of the array in which number 1 should be inserted.

The element search: Suppose that we are going to test the presence or absence of X in filter bloom. At first, the element is given to the tangled function K which determines the location of this number in bloom array. Then, the values of the array in this K is studied. If there is zero in one or more box, the filter will definitely show that this element does not exist in this array. In other words, if there were any element, the K should be 1. In contrast, if all were 1, it would not be clear enough whether this 1 is due to the existence of this element or the insertions of other elements. Therefore, this element may exist in this filter. It shows that the more the element is inserted in the array, the more probable the answer is wrong.

Element deletion: Since negative error is not allowed in this filter, the element cannot be deleted. It means that as the absence of this element should be clearly reported, deletion is not possible. In fact, when we are intended to delete an element, we should change all paired bits of this element to zero. As a result, the element is removed. However, it is possible that the places of other numbers share some similarities with this element which are changed to zero by mistake. Therefore, if the filter shows the absence of an element, it may not be true. In fact, it can be due to the removal of the elements which have had situational similarities with this member. As a result, to stop negative errors, it is not possible to remove an element in this structure.

Cryptography algorithm

Cryptography is an operation in which the obvious or main text is changed to a new form applying various procedures. They do not have a specific content. The only people who have the code key can reach the main text. Cryptography algorithms are classified as symmetric and asymmetric groups. In the former, cryptograph and de-cryptography have the same but reverse process. In the present study, AES cryptography algorithm, as a symmetric cryptography, is applied.

Figure 1: Insert operation in Bloom Filter.

DES was the victim of US Federal support. Even if it had not had this destructive support, it would not have found an appropriate place for itself in IT in the third millennium. NIST1 organization did not want to repeat the unsuccessful experience of DES. Holding a multi-stage competition, started in January in 1997 and continued to November in 2001, it tried to choose the new standard of cryptography out of real and legal characters. In fact, it was going to remove any interference in the process of designing. As the stages were passed, two Belgian men won the competition. Their algorithm was standardized by NIST called AES.

THE PRESENTED ALGORITHM

In this part, an algorithm for cryptography and confirmation of user identity in a wireless sensor network is introduced. It is hypothesized that there is a fixed wireless sensor network with some sensors. In this network, the sensors have been divided into some clusters and each one has a head cluster. Besides, the information gathered from the environment is exchanged among the sensors wirelessly. Most studies have tried to find a way to decrease the energy use in sensor networks, although the present study aims to introduce a way to crypt the data and confirm the identity of the senders. Algorithm will be defined.

The presented algorithm has three phases including data cryptography, data transformation, and the sender confirmation. Each phase is explained as follow.

a. The phase of data cryptography

In the first stage, the collected data are codified by AES cryptography algorithm which is one the standard cryptography algorithms. The way is presented as ion a study [1]. Improving cryptography operation, some s-boxes are used in this study which is defined as the following.

1. At first, a linear array named S is made. The size is 256 entry including the s-box values by AES method. The length is put in Δ (Δ=256).
2. Chaos linear function of piece-wise is done N0 times. The function is explained as follow.

$$(n+1)=\begin{cases}\dfrac{x(n)}{p} & 0<x(n)£p\\ \dfrac{1-x(n)}{1-p} & p<x(n)<1\end{cases} \quad (1)$$

Here, X (0)=0.3571, p=0.587 and N0=1000 which are used as the primary values; the key of chaos function.

It should be mentioned that the result of this stage is a sequence with the length of 1000.

1. Cnt variable equals 1. It is the counter of ring.
2. The values of chaos function were counted in stage 2. It is repeated once more finding a new variable, x, which will be used in next stage.
3. Random number m which is in [1, k] collection is computed by the following method:

$$m=\left\{\text{floor}\left(x\times 10^{10}\right)\right\}\bmod(k)+1 \quad (2)$$

In which k=Δ -cnt+ 1

4. Two elements of S array which are in m and k situations are moved.
5. Now, if cnt is smaller than 256, 1 unit is added entering stage 4.
6. The stages 3 to 7 on the present S array are repeated for c times in which c equals 3.
7. The moving result of linear S array is moved to a 16*16 table which is the final s-box.

These 9 stages are in a 1000-one ring which is repeated 1000 times to make 1000 different s-boxes from the main s-box. One of these 1000 s-boxes is randomly chosen and used in a stage in which the output of the second phase is replaced by successor tables.

b. The phase of data transformation

In the next stage, ID code of sensor knot is given to the tangled function (hash) to find the abstract. In this way, SHA-a algorithm is used to find the value of hash. Figure 2 shows the code-like of SHA-1 algorithm.

The results of tangled function are put in hash1, hash 2 and hash 3 respectively. The values are the numbers in a range from 1 to MBF which reveal the index of BF array, 1 in the boxes.

The cashed data is sent to destination knot along with bloom filter vector (Figure 3).

C) The phase of sender confirmation

In the main cluster, identity confirmation is done first to prevent

```
h0 = 0x67452301
h1 = 0xEFCDAB89
h2 = 0x98BADCFE
h3 = 0x10325476
h4 = 0xC3D2E1F0
ml = message length in bits (always a multiple of the number of bits in a character).
append the bit '1' to the message e.g. by adding 0x80 if message length is a multiple of 8 bits.
append 0 ≤ k < 512 bits '0', such that the resulting message length in bits
   is congruent to −64 ≡ 448 (mod 512)
append ml, the original message length, as a 64-bit big-endian integer.
   Thus, the total length is a multiple of 512 bits.
break message into 512-bit chunks
for each chunk
   break chunk into sixteen 32-bit big-endian words w[i], 0 ≤ i ≤ 15
   for i from 16 to 79
      w[i] = (w[i-3] xor w[i-8] xor w[i-14] xor w[i-16]) leftrotate 1
   a = h0
   b = h1
   c = h2
   d = h3
   e = h4

   for i from 0 to 79
      if 0 ≤ i ≤ 19 then
         f = (b and c) or ((not b) and d)
         k = 0x5A827999
      else if 20 ≤ i ≤ 39
         f = b xor c xor d
         k = 0x6ED9EBA1
      else if 40 ≤ i ≤ 59
         f = (b and c) or (b and d) or (c and d)
         k = 0x8F1BBCDC
      else if 60 ≤ i ≤ 79
         f = b xor c xor d
         k = 0xCA62C1D6

      temp = (a leftrotate 5) + f + e + k + w[i]
      e = d
      d = c
      c = b leftrotate 30
      b = a
      a = temp

   h0 = h0 + a
   h1 = h1 + b
   h2 = h2 + c
   h3 = h3 + d
   h4 = h4 + e

hh = (h0 leftshift 128) or (h1 leftshift 96) or (h2 leftshift 64) or (h3 leftshift 32) or h4
```

Figure 2: Hash function pseudo-code.

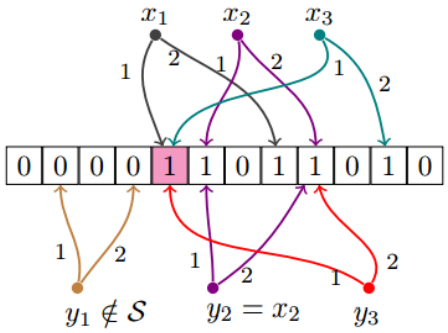

Figure 3: Collision in Bloom Filter.

the knot from decrypting invalid data and to save energy. Therefore, the person's ID is asked first to search the validity of the users. Then, this ID is sent to three tangled functions of the program determining three indices. If there is number 1 in three boxes of this filter, the validity of this ID will be announced. If there is o in all boxes, the ID will be rejected. Besides, if at least one of the boxes has number 1, the ID may be valid. However, the validity is not announced with certainty. As a matter of fact, the box with number 1 may have got that number due to the insertion of other usernames.

Bloom filter uses the inclusive operator XOR which decreases the connection overload securing the network against DOS. Two mechanisms are used to do this approach. First phase is key gathering in which each key is considered for different temporal parts. All keys are gathered as a linear vector. Second phase is collision management in which the collision is managed by inclusive operator XOR and bloom filter on network data. As a matter of fact, the abstracts of tangled functions (hash) become XOR and then applied in bloom filter.

The bloom filter applied in this study uses the system of authentication and public spread in wireless sensor networks. This network is made of a base station with enough power and a memory for secure communications. Each sensor knot not only saves the bloom filter in its memory but also has a tangled function K (hash). Each normal knot can make the authentication messages spread for all knots of the network. In this model, it is hypothesized that all knots have fixed locations and situations. Furthermore, different bloom filter protocols are considered in this algorithm. In the protocol, the sender sensor knot saves a collection of S components in BF bloom filter. The size of this bloom filter equals BF along with tangled function N (hash).

First phase - key gathering: At first, all bits of BF bloom filter are equalized to zero. If tangled function for each exiting element in K, shown by Ki, is hj, hj (k)=I, then the i^{th} bit of BF bloom filter is adjusted with number 1(BF[i] hj=1. Here BF[i]hj is the i^{th} value in BF vector.) The keys are saved in bloom filter due to decreasing the communication extra load among sensor knots. In this phase, the keys and MACs are not sent separately but as a package and all together. A package of P data is the corresponding to M message sent Ti times can be made by:

P=P (MACi)| BFi (3)

BFi represents the mapping of K key K=<k1, k2... kd> on bloom filter vector.

Second phase - collision management: Collision happens when 2 keys are tangled in one situation in BF. The following picture is an example of bloom filter operation in the presence of collision. In this example, a filter bloom is considered in the size of 12 bit. It means that m=1 and two tangled functions are used, k=2.

> There is a collection of combining three elements S={X1, X2, X3}. The collision happens when X1 and X2 locate their values in the same cell; as it is presented in the figure of a colored cell.

When two functions hashed two different values and are in the same situation leading to collision, two values will be a member of the data collection by mistake. Therefore, most of the correct values have high collision rate, they are recognized as the incorrect values. It is more possible to be at the risk of wrong situation for these values.

Therefore, when collision happens, we use inclusive operator XOR out of two values having the same situation in BF array.

BF[i] h1 XOR BF[i] h2 (4)

BF[i] h2 is the codified output of tangled function h2. BF[i] h1 is also codified output of tangled function h1. For example, if h1 (k1)=3, then BF [3]=BF [3] h1. Then, if h2 (k2)=3, then the collision will occur. The following relationship is used.

BF [3]=BF [3] h1 XOR BF [3] h2=1 XOR 1=0 (5)

XOR operation is applied on a collection like S which has n member. Like the following:

Collision discovery: 0 insertion in bloom filter array means the cell is empty, while 1 insertion means that the output of tangled function is codified. Collision occurred when a cell receives two 1 values simultaneously. In code-like 1, the stages of XOR in bloom filter is presented. A counter is considered for each cell to count the number of inserted 1es inside the cell. In next stage, XOR function is carried out even if the collision occurred.

XOR function: Collision happens when two hashed MAC addresses are located at the same cell in BF. Two values which are mapped to this cell, should become XOR.

BF[i] h1 XOR BF[i] h2 (6)

The receiver sensor knots apply similar rules to produce BF array and then compare the produced rules with received BF arrays from the sender. If both produced arrays are the same, then the received package is valid. The process of testing network members in this filter can be seen in the second code-like.

CONCLUSION

Code-like 1:

For j=0 to j=k do

For i=0 to i=N do

B[ei] ← *hj (ei)*

If Collision.counter > 1 then

(BF[i] h1 XOR BF[i] h2)

End if

I=i+1;

End for

End for

Code-like 2:

For each received key Ki in time interval d:

If the verification of MAC (Ki)=true then

Return (authenticated packet)

Else

Return (authentication failed)

End if

End for each.

When authentication is done, the cashed data are decrypted by a revers way other than codifying which will be saved in the memory of head-cluster for future calculations.

REFERENCES

1. Yasmin R, Ritter E, Wang G. An Authentication Framework for Wireless Sensor Networks using Identity-Based Signatures. 10th Int Conf on Comp and Info Tech (CIT), 2012.
2. Al-Mahmud A, Akhtar R. Secure Sensor Node Authentication in Wireless Sensor Networks. Int Conf Com App, 2012;46(4).
3. Dhawale AD, Chandak MB. Implementation of Rekeying Mechanism for Node Authentication in Wireless Sensor Networks. Intl J Adv Smart Sens Net Syst. 2012;2(4).
4. Rathore R, Hussain M. Simple, Secure, Efficient, Lightweight and Token Based Protocol for Mutual Authentication in Wireless Sensor Networks. Emerge Res Comp Info Commun App.
5. Choi Y, Lee Y, Won D. Security improvement on biometric based authentication scheme for wireless sensor networks using fuzzy extraction. Int J Distrib Sens Net. 2016;116.
6. Park Y, Park Y. Three-factor user authentication and key agreement using elliptic curve cryptosystem in wireless sensor networks. Sensors. 2016;16:2123.
7. Moon J, Lee D, Lee Y, Won D. Improving Biometric-Based Authentication Schemes with Smart Card Revocation/Reissue for Wireless Sensor Networks.Sensors. 2017;17:940.
8. Krishna CM. Fault Detection in Cryptographic Systems in Fault-Tolerant Systems. Cryptographic Algorithm. 2007.
9. Davida GI. Yvo Desmedt Department of Electrical Engineering and Computer Science Advances in Computers. 1990;30;171-222.
10. http://www.umsl.edu/~siegelj/information_theory/projects/HashingFunctionsInCryptography.html