Indian Journal of Science and Technology, Vol 9(48), DOI: 10.17485/ijst/2016/v9i48/107972, December 2016
ISSN (Print) : 0974-6846
ISSN (Online) : 0974-5645

Updating Distributed Cache Mechanism using Bloom Filter for Asymmetric Cryptography in Large Wireless Networks

S. Kavitha, R.Thanuja* and A. Umamakeswari

[1]School of Computing, SASTRA University, Thanjavur - 613401, Tamil Nadu, India;
kavithasuthalson@gmail.com, thanuja.r@cse.sastra.edu, aum@cse.sastra.edu

Abstract

Objectives: Public key cryptography helps in achieving information security in wireless networks. Public key will be issued by CA (Certificate Authority). But having CA in wireless network makes it vulnerable to various attacks. Hence distributed cache was proposed where each node in network caches public key of some nodes existing in network in its cache and request to obtain keys not cached will be made via trusted nodes. Due to limited memory only few keys can be cached. There exists an optimal threshold value for caching keys so that equal number of both local nodes and remote nodes public keys can be cached. **Methods:** Random replacement policy was used for updating cache contents, where if a node's local/ remote key cached ratio exceeds optimal ratio then a key will be randomly chosen from the cache and will be removed. Here we propose implementation of bloom filter in cache to cache public key of nodes existing in network. **Findings:** By using bloom filter of large size, more number of public keys can be cached thereby providing space and time advantage and reduces query cost with increased usage of memory. MAC (Media Access Control) id and PSN (Processor Serial Number) are used for uniquely identifying each node in network and are stored in encrypted format. When mismatch is found in encrypted MAC id and PSN, then that node is determined to be malicious node and its id will be notified to all nodes in network thus excluding the node from making any communication to other nodes in the network. **Applications:** Used for checking an element's presence in the set thus reducing I/O lookup when large data set is used. Used as an alternative to cache as it uses limited memory more efficiently than cache does.

Keywords: Bloom Filter, Certificate Authority, Distributed Cache, PSN (Processor Serial Number), Public Key Cryptography

1. Introduction

Use of wireless network has become popular in today's world as communication between devices is based on radio signals and does not involve any cables to be installed. Using wireless network, people can access the network from any location. In MANET (Mobile Ad hoc Network), network topology changes over time and it is the responsibility of the nodes to discover topology and deliver message among themselves. In Wireless Sensor Network (WSN), multiple sensor nodes monitor environmental conditions and transmit these data to main location. Research works have been carried on such large wireless networks which focuses on using optimum ratio for caching keys[1] network performance based on parameters like throughput, energy efficiency and lifetime of network[2–5] and security[5,6]. These large wireless networks can be prone to passive attack (where data being passed through network can only be read by third party but cannot modify the message being passed) and active attacks (where data being passed through network can be read and modified by third party). These attacks can be made by compromised nodes which are present either outside or inside the network. Hence proper security mechanisms have to be employed in order to prevent critical information being leaked to attackers. Public key cryp-

*Author for correspondence

tography can be used for secure transmission of message. Here each node present in network has two keys namely public key (all nodes existing in network knows this) and private key (node keeps it as a secret). By public key of receiver, the message is encrypted and with private key of receiver, the message is decrypted. Usually CA issues certificates (containing public key and node id) to all nodes in network. But having a CA in large wireless network is not suitable as it may be prone to attacks like wormhole, black hole and jamming attacks. Moreover wireless network have dynamic topology where nodes may be joining and leaving the network at any point of time thus making CA unsuitable. Each node acts as CA to other nodes present in the network by storing public keys of nodes in its cache. Due to limited memory, not many keys can be stored. Hence in addition to regular nodes, anchor nodes were deployed which dedicates its complete memory only for public key storage. In this work, caching public keys using bloom filter is proposed where bloom filter is implemented in cache and by this more number of keys can be stored on each node with less memory. Query cost is reduced and success of finding queried keys is increased by using bloom filter. Bloom filter caches public keys of both local nodes (nodes within transmission range) and remote nodes (nodes outside transmission range). Based on bloom filter size and multiple hash functions, more number of public keys can be cached.

Key management can be done either by using CA or in a distributed manner. By using CA methodology, it is the role of the central authority to generate and revoke keys in a periodic fashion. But CA is prone to several attacks. Hence distributed methods can be used in[7]. Distributed method does not have CA. Using trust metric of nodes[8], public keys of other nodes can be obtained using node id and its trust level. In TOMS[9] (Trust Computation and Management System), uses trust value among distributed nodes for establishing trust. When node transmits message successfully its trust value increases else decreases. How long a node stays in the network determines trust of that node. Certificates[10] will be assigned to other nodes based on secret key which is initialized by public key of the node. Trust value[11] is present between nodes and based on this trusted path is obtained. MAC is used for transferring public key between nodes to achieve integrity and authenticity. ECC[12] (Elliptical Curve Cryptography) is used for key management. Digital signatures are verified by sensor using public key of gateway. Certificate chain[13] is established for public key authentication between source and destination node by collecting all the certificates present in certificate chain. Certificates[14] are signed based on node's private key share and obtained from system dealer. Using ant colony[15] public keys are managed and trust established based on demand where nodes issues certificates to neighbor nodes signed with its private key. Keys are managed using ID where network's master key is distributed among shareholders using threshold cryptography[16]. Using cooperative caching[17] data item that benefits network is cached. But in selfish caching, node caches data item that benefits itself. Network devices[18] caches chunks in a distributed cache over cache paths using probabilistic algorithm by determining probability of in-network caching using cache weight. Protocols[19] perform task of interchanging keys and the confidential keys are swapped between communicating peers for mobile nodes. In[20] Table is kept by server involving aspects of clients and when server is revived, the data are updated on those clients for preserving cache stability.

2. System Model

The Figure 1 represents node's having Bloom Filter for caching public keys of other nodes. By passing node id to multiple hash functions, the public keys are stored onto bloom filter. The Figure 2 represents KREQ (Key Request) message format. The Figure 3 represents KREP (Key Reply) message format.

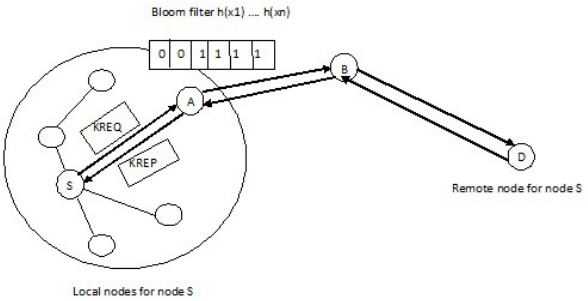

Figure 1. Illustration of caching public key using bloom filter.

3. Caching Public Key

Notations Used:

N = Estimated quantity of elements in bloom filter.
m = Bloom filter's capacity.
h = Count hash functions used.

Y = Count of bits that are placed to 1.
n = Count of elements inserted in bloom filter.
p = Likelihood of faulty positive.

Source (S)	Destination (D)	TTL	List of routers (R)

Figure 2. Key Request (KREQ) message format.

Source (S)	Destination (D)	Public key of destination	List of routers (R)

Figure 3. Key Reply (KREP) message format.

3.1 Establishing Wireless Network

Area is found out for all nodes present in the network. This area is used to find the transmission range of each and every node by which local nodes or remote nodes are determined. Local nodes are nodes inside communication radius and remote nodes are nodes outside communication radius. Nodes generate its own set of public key and private key using Diffie-Hellman algorithm.

3.2 Processing KREQ Message

Assume all nodes contain public key of few other nodes in its cache. Say node S requires sending data to node D. For that node S must possess public key of node D so that it can encrypt the data. Assume node S does not have node D's public key in its cache. Node S sends KREQ message via trusted nodes to obtain node D's public key. Nodes receiving KREQ message checks whether it has public key of sender node in its cache. If yes, it processes the message; else ignores. Thus link of trusted nodes is established.

Algorithm for processing KREQ:

- i^{th} node encounters KREQ message.
- j -> last node in router list.
- if public key of j not cached or mismatch in MAC then.
- ignores the message and exits.
- end if.
- if destination node public key cached then.
- prepares KREP and sends to .j
- if condition ends.
- Reduce TTL by 1.
- If TTL is greater than 0 then.
- Construct KREQ and broadcast.
- End if.

3.3 Processing KREP Message

Node caching D's public key sends KREP message to sender node by encrypting the message using immediate receiver node's public key. If receiving node does not cache D's public key then it attaches its ID to router list; generates new MAC with its private key and broadcasts it to its one hop neighbor. KREP message will be sent to sender node and intermediate nodes in chain of trust will decrypt message with its own private key and stores and forwards node D's public key in its cache.

Algorithm for processing KREP message:

- Node k receives KREP message.
- Decrypts the message using its private key.
- If k = source then.
- Obtain public key and exit.
- Else.
- Prepare KREP by encrypting the message with public key of immediate receiver node.
- Send KREP message.
- End if.

3.4 Updating using Bloom Filter

Bloom filter is probabilistic in nature and is used to examine if an element exists in set or not.

Step 1: Initialize a bloom filter in the form of bit array; say m bits, with all bits initially assigned to 0.

Step 2: Define a set of hash functions that has to be applied to insert an element or check for an element's existence in bloom filter.

Step 3: Insert elements into bloom filter by passing on those elements to multiple hash functions and set the coinciding bit position in the array from 0 to 1.

Step 4: When a new element is detected and has to be inserted in to the bloom filter, pass it on to multiple hash functions and if any of the hash function returns 0 then it indicates element does not exist in bloom filter; so add the element to the corresponding bit array and set the bit from 0 to 1.

Step 5: If all hash functions returns 1 then that indicates the element is already present in the bloom filter.

Step 6: Elements cannot be deleted from bloom filter.

For approximating quantity of elements present in bloom filter Equation (1) is used:

$$N = \frac{-m \mathrm{h}\left[1-\frac{Y}{m}\right]}{h} \quad (1)$$

To reduce likelihood of faulty positive Equations (2) to (7) are used:

Consider bloom filter size to be m, then probability that particular hash function has not set a particular bit to 1 while inserting an element is:

$$1-\frac{1}{m} \quad (2)$$

Consider count of hash functions used to be h, then probability that none of the hash function places bit to 1:

$$\left(1-\frac{1}{m}\right)^{h} \quad (3)$$

After inserting n elements, likelihood of certain bit placed to 0:

$$\left(1-\frac{1}{m}\right)^{hn} \quad (4)$$

Likelihood of a bit placed to 1 after inserting n elements is:

$$1-\left(1-\frac{1}{m}\right)^{hn} \quad (5)$$

Probability of all the bits being set to 1 is:

$$\left[1-\left(1-\frac{1}{m}\right)^{hn}\right]^{h} \approx \left(1-e^{-hn/m}\right)^{h} \quad (6)$$

Say p as fraction of m where bits are 0 after inserting all the elements in to the bloom filter. i.e. count of bits that are 0 is *pm*.

To verify an element is not present in set by any hash function h, then likelihood of bit being placed to 1 will be 1-*p*.

Probability that bit placed to 1 by all hash functions h,

$$(1-p)^{h} \quad (7)$$

To determine optimal count of hash functions to minimize probability of false positive Equation (8) is used:

$$-\ln 2 \quad (8)$$

To determine bloom filter's capacity for reducing likelihood of faulty positive, Equation (9) is used:

$$\frac{n\ln(p)}{\left(\ln(2)\right)} \quad (9)$$

Algorithm for constructing bloom filter:

- Procedure Bloom_Filter (set X, hash_fn, int m).
- Return filter.
- Filter = set m bits with all bits initialized to 0.
- For every element x_i in X.
- Apply hash function h_i.
- Filter $[h_i(x_i)]$ sets bits from 0 to 1.
- End for.
- End for.
- Return filter.

Algorithm for testing an element is present in set or not:

- Procedure Testing_for_element (element, filter, hash_fn).
- Returns present or not present.
- For each hash function h_i.
- If $[h_i(element) \neq 1]$ then return element not present.
- End for.
- Returns element present in set.

3.5 Protecting against Attacks

PSN and MAC are used for uniquely identifying a node in network. If PSN or MAC is found to be modified during transit, the node is said to be malicious node and presence of malicious node will be notified to all nodes present in the network thereby preventing the message from being passed through malicious node. When mismatch is found between encrypted MAC and message, the message will be ignored.

4. Results and Discussion

The Figure 4 represents decrease in likelihood of fault positive as number of hash functions used increases. The Figure 5 represents number of keys that can be cached using bloom filter. As the size of the bloom filter increases, more number of keys can be cached.

5. Conclusion and Future Enhancement

While using asymmetric cryptography sender has to know the destination's public key in order to encrypt the

message. Usually CA will issue public key of all nodes to all other nodes present in the network. But having CA is not secure in wireless networks as it is prone to multiple types of communication attacks. Hence each node is made to act as CA by caching public keys of few nodes present in network. Bloom filter is proposed where more number of keys can be stored as it stores bits and not the actual data thus reducing query cost. Efficiency of bloom filter depends on number of bits in array and number of hash functions. It provides fast membership checking. Once malicious node is detected, public key of the malicious node has to be revoked and this work is left out for future research.

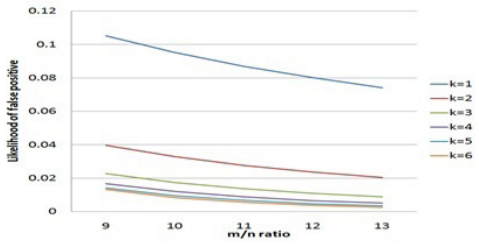

Figure 4. Decrease in likelihood of fault positive.

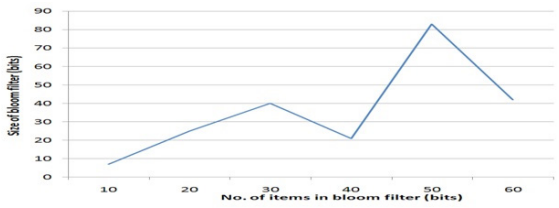

Figure 5. Keys cached increases with respect to size of bloom filter.

6. Acknowledgement

The authors wish to express their sincere thanks to the Department of Science & Technology, New Delhi, India (Project ID: SR/FST/ETI-371/2014). The authors also thank SASTRA University, Thanjavur, India for extending the infrastructural support to carry out this work.

7. References

1. Yao L, Deng J, Wang J, Wu G. A-CACHE: An anchor based public key caching scheme in large wireless networks. Journal of Computer Networks. 2015 Jul; 87:78–88.
2. Weber S, Andrews JG, Jindal N. An overview of the transmission capacity of wireless networks. IEEE Transactions on Communications. 2010 Dec; 58(12):3593–604.
3. Andrews JG, Ganti RK, Haenggi M, Jindal N, Weber S. A primer on spatial modeling and analysis in wireless network. IEEE Communications Magazine. 2010 Nov; 48(11):156–63.
4. Di Francesco M, Das SK, Anastasi G. Data collection in Wireless Sensor Network with mobile elements: A survey. ACM Transactions on Sensor Networks. 2011 Aug; 8(1):1–34.
5. Zhou X, Ganti RK, Andrews JG, Hijorungnes A. On the throughput cost of physical layer security in decentralized wireless network. IEEE Transactions on Wireless Communications. 2011 Aug; 10(8):2764–75.
6. Vasudevan S, Goeckel D, Towsley DF. Security–capacity trade–off in large wireless network using keyless secrecy. Proceedings of the Eleventh ACM International Symposium on Mobile Ad hoc Networking and computing; 2010 Sep. p. 21–30.
7. Koo JH, Kim BH, Lee DH. Authenticated public key distribution scheme without trusted third party. Springer Berlin Heidelberg; 2005 Dec. p. 926–35.
8. Hamouid K, Adi K. Efficient certificate less web of trust model for public key authentication in MANET. Computer Communications. 2015 Jun, 63, pp.24 – 39.
9. Boukerche A, Ren Y. A trust- based security system for ubiquitous and pervasive computing environments. Computer Communications. 2008 Dec; 31(18):4343–51.
10. A trust based threshold cryptography key management for Mobile Ad hoc Networks. 2009. Available from: http://ieeexplore.ieee.org/document/5379089/
11. Maity S, Hansdah RC. Certificate-less on-demand public key management (clpkm) for self-organized manets. Springer Berlin Heidelberg; 2012 Dec. p. 277–93.
12. Cho JH, Chan KS, Chen IR. Composite trust-based public key management in Mobile Ad hoc Networks. Proceedings of the 28th Annual ACM Symposium on Applied Computing; 2013 Mar. p. 1949–56.
13. Memarmoshrefi P, Seibel R, Hogrefe D. Bio inspired self organized public key authentication mechanism for Mobile Ad hoc Networks. Springer Berlin Heidelberg; 2010 Dec. p. 375–86.
14. Steady status study of distributed data caching in ad hoc networks. 2013. Available from: http://ieeexplore.ieee.org/document/6614197/
15. Learning distributed caching strategies in small cell networks. 2014. Available from: http://ieeexplore.ieee.org/document/6933484/
16. Wave: Popularity based and collaborative in network caching for content oriented networks. 2012. Available from: http://ieeexplore.ieee.org/document/6193512/

17. Steady status study of distributed data caching in ad hoc networks. 2013. Available from: http://ieeexplore.ieee.org/document/6614197/
18. Psaras I, Chai WK, Pavlou G. Probabilistic in-network caching for information centric networks. Proceedings of the 2nd ed. of the ICN workshop on Information Centric Networking; 2012. p. 55–60.
19. Vyas P, Trivedi B, Patel A. A survey on recently proposed key exchange protocols for mobile environment. Indian Journal of Science and Technology. 2015 Nov; 8(30):1–5.
20. Kanchana A, Gayathri P. A survey on push based server initiated update mechanism for caching in wireless network. Indian Journal of Science and Technology. 2015 Nov; 8(32):1–5.