Survey on Real Time Broadcast Authentication Schemes for Command and Control Messages

Sonia Thomas, Ms Elisabeth Thomas

Department of computer science and engineering
AJCE, Kanjirapally, Kerala, India

***Abstract* — a broadcast authentication protocol enables the receivers to verify that a received packet was really sent by the claimed sender. Data confidentiality, authenticity, integrity, and non repudiation are basic concerns of securing data delivery over an insecure network. Large and distributed systems with time critical applications require immediate and secure authentication of command and control messages to work efficiently. This paper compare various message authentication schemes like symmetric, asymmetric cryptographic methods, delayed key disclosure methods, signature amortization techniques, one time signatures, online-offline signatures and rapid authentication scheme. On analyzing we found that rapid authentication is more efficient for broadcast authentication of command and control messages with real time applications.**

Keywords—Broadcast authentication, digital signature, verification.

A.

I. Introduction

Large and distributed Real time systems broadcast control and command messages to direct its peripherals for performing its functionality. This broadcasting must be done in an efficient, secure and scalable manner inorder to prevent adversaries from forcing catastrophic decisions that may cause disastrous consequences. Authentication of this command and control messages ensures integrity of messages.ie it assures that messages are received as sent, with no duplication, insertion, modification, recording or replays. Example of such system include cyber physical infra structures like power grid, smart grid, disaster response systems like earth quake warning systems, fire sensors etc. Real time systems require immediate verification and also resource constrained receivers does not support expensive operations.

Simply deploying the standard point-to-point authentication mechanism (i.e., appending a message authentication code (MAC) to each packet, computed using a shared secret key) does not provide secure broadcast authentication. The problem is that any receiver with the secret key can forge data and impersonates the sender. Consequently, it is natural to look for solutions based on asymmetric cryptography to prevent this attack; a digital signature scheme is an example of an asymmetric cryptographic protocol. Indeed, signing each data packet provides secure broadcast authentication; however, it has high overhead, both in terms of the time required to sign and verify, and in terms of the bandwidth. Several schemes were proposed that mitigate this overhead by amortizing a single signature over several packets. However, none of these schemes is fully satisfactory in terms of bandwidth overhead, processing time, scalability, robustness to denial-of-service attacks, and robustness to packet loss. Even though some schemes amortize a digital signature over multiple data packets, a serious denial-of-service attack is usually possible where an attacker floods the receiver with bogus packets supposedly containing a signature. Since signature verification is often computationally expensive, the receiver is overwhelmed verifying bogus signatures.

II. Literature Survey

In this section we analyse various message authentication schemes. We also discuss pros and cons of each technique.

A. symmetric cryptographic methods

In symmetric cryptographic methods a message M transmitted from source A to destination B is encrypted using a secret key K shared by A and B. If no other party knows the key, then confidentiality is provided. ie no other party can recover the plain text of the message. It relays on message authentication code (MAC) to achieve computational efficiency. To be considered secure, a MAC function must resist existential forgery under chosen-plaintext attacks. This means that even if an attacker has access to an oracle which possesses the secret key and generates MACs for messages of the attacker's choosing, the attacker cannot guess the MAC for other messages without performing infeasible amounts of computation. The originator of a message x computes a MAC $h_k(x)$ over the message using a secret MAC key k shared with the intended recipient, and sends both (effectively $x||h_k(x)$). The recipient determines by some means the claimed source identity, separates the received MAC from the received data, independently computes a MAC over this data using the shared MAC key, and compares the computed MAC to the received MAC. The recipient interprets the agreement of these values to mean the data is authentic and has integrity – that is, it originated from the other party which knows the shared key, and has not been altered in transit. It is very simple to implement since we only need to share a key between senders and receivers. But this pair wise key distribution between signer and verifier is not practicable in large systems. And also due to symmetric nature it is not publically verifiable and hence it cannot achieve non repudiation property.

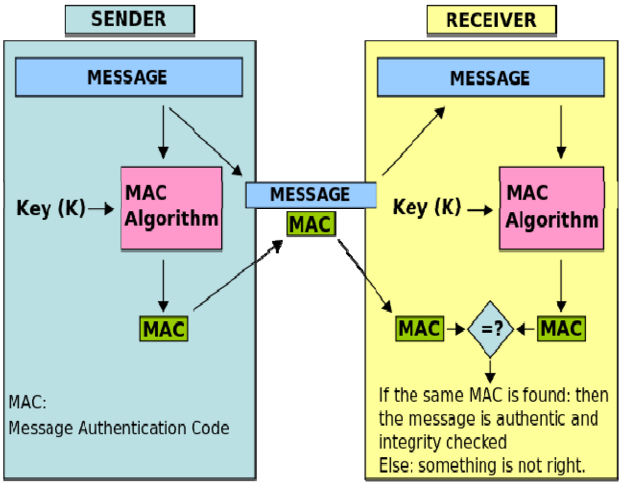

SENDER
MESSAGE
Key (K)
MAC Algorithm
MAC
MAC:
Message Authentication Code
MESSAGE
MAC
RECEIVER
MESSAGE
Key (K)
MAC Algorithm
MAC
=?
MAC
If the same MAC is found: then the message is authentic and integrity checked
Else: something is not right.

Figure 1

Researchers showed that MACs can be efficiently implemented on resource-constrained sensor network nodes, and find that computing a MAC function requires on the order of 1ms on the computation-constrained Berkeley mote platform. Authentication of broadcast messages in sensor networks is much difficult than point-to-point authentication. The symmetric approach used in point-to-point authentication is not secure in broadcast settings, where receivers are mutually untrusted. If all nodes share one secret key, any compromised receiver can forge messages from the sender.

B. Asymmetric cryptographic methods

Asymmetric cryptosystem use different keys at the sender and receiver sides, where one is called as private key and the other is the public key. Anyone can encrypt messages using the public key, but only the holder of the paired private key can decrypt. Security depends on the secrecy of the private key. Even though keys seem different, they are mathematically related. Private Key is used to create signature or to decrypt message and public key is used to verify signature or to encrypt message. Public key algorithms rely on mathematical problems like integer factorization, discrete logarithm and elliptic curve relations. The strength of the system is based on the fact that it is computationally infeasible to obtain the secret key from a properly generated public key. Message authentication involves processing a message with a private key to produce a digital signature. Thereafter anyone can verify this signature by processing the signature value with the signer's corresponding public key and comparing that result with the message. Success confirms the message is unmodified since it was signed, and – presuming the signer's private key has remained secret to the signer – that the signer, and no one else, intentionally performed the signature operation. Since it relay on public infrastructures it is publically verifiable, key size is resilient and is scalable for large systems. Examples include RSA, ECDSA [3]etc.

RSA is a combination of three algorithms key generation, signature generation and verification. The security of RSA is mainly based on integer factorization. Elliptic curve cryptography makes use of elliptic curves in which variables and coefficients are restricted to elements of a finite field. A considerably smaller key size can be used in ECDSA compared to RSA. Their computational effort is also comparable. But these systems require expensive operations like modular exponentiation which makes them impractical for broadcast system with real time applications and resource constrained systems.

C. Delayed key disclosure methods

Most popular delayed key disclosure methods are TESLA, µ-TESLA etc. The idea of delayed key disclosure method is that a MAC code is appended to each and every packet. Key of this MAC is disclosed only in some subsequent packet.

The main idea of TESLA is that the sender attaches to each packet a MAC computed with a key *k* known only to itself. The receiver buffers the received packet without being able to authenticate it. A short while later, the sender discloses *k* and the receiver is able to authenticate the packet. Consequently, a single MAC per packet suffices to provide broadcast authentication, provided that the receiver has synchronized its clock with the sender ahead of time. Each receiver that receives the packet performs the following operation. It knows the schedule for disclosing keys and, since the clocks are loosely synchronized, can check that the key used to compute the MAC is still secret by determining that the sender could not have yet reached the time interval for disclosing it. If the MAC key is still secret, then the receiver buffers the packet. Each receiver also checks that the disclosed key is correct (using self-authentication and previously released keys) and then checks the correctness of the MAC of buffered packets that were sent in the time interval of the disclosed key. If the MAC is correct, the receiver accepts the packet.

Figure 2 depicts the one-way key chain derivation, the MAC key derivation, the time intervals, and some sample packets that the sender broadcasts.

For example for packet *Pj*+3, the sender computes a MAC of the data using key

K'i+1

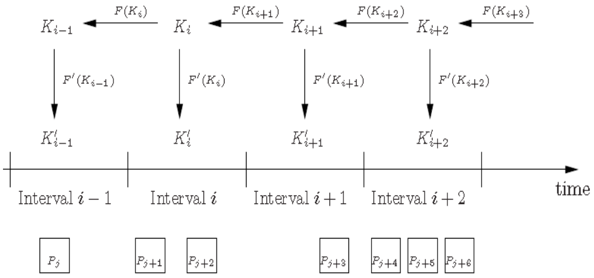


Figure 2

verification since the key is known only after receiving its corresponding packet having the key. And also this method requires time synchronization between sender and all receives. But maintaining continuous time synchronization is a challenging problem in large and distributed system.

D. Signature amortization methods

Amortization schemes for authenticating streamed data have been introduced as a solution to reduce the high overhead that sign-each scheme suffer from. This technique greatly improves signing and verification rates compared to the naïve signature-per packet approach .TESLA is not designed for limited computing environments like sensor networks. μTESLA is designed to solve the following inadequacies of TESLA in sensor networks.

- TESLA authenticates the initial packet with a digital signature, which is too expensive for our sensor nodes. μ_TESLA uses only symmetric mechanisms.
- Disclosing a key in each packet requires too much energy for sending and receiving. μTESLA discloses the key once per epoch.
- It is expensive to store a one-way key chain in a sensor node.μ_TESLA restricts the number of authenticated senders

μ_TESLA requires that the base station and nodes are loosely time synchronized, and each node knows an upper bound on the maximum synchronization error. To send an authenticated packet, the base station simply computes a MAC on the packet with a key that is secret at that point in time. When a node gets a packet, it can verify that the corresponding MAC key was not yet disclosed by the base station (based on its loosely synchronized clock, its maximum synchronization error, and the time schedule at which keys are disclosed). Since a receiving node is assured that the MAC key is known only by the base station, the receiving node is assured that no adversary could have altered the packet in transit. The node stores the packet in a buffer. At the time of key disclosure, the base station broadcasts the verification key to all receivers. When a node receives the disclosed key, it can easily verify the correctness of the key . If the key is correct, the node can now use it to authenticate the packet stored in its buffer.

The aim of this delayed key disclosure is to achieve public verifiability of symmetric primitives, without losing its computational efficiency. But this delayed key disclosure method cannot achieve immediate

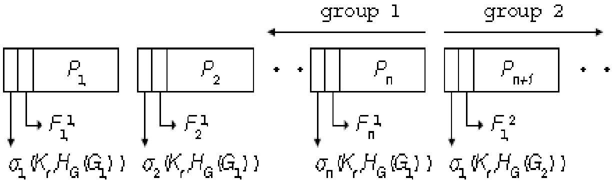


Figure 3

Figure 3 [11] shows authenticated packet streams.

In this a signature is computed for set of packets rather than finding distinct signature for each packet. By doing that cost for signature generation and verification is amortized for a set of packets. If the condition on individual packet verification is relaxed so that the verification of a packet is dependent on other packets within the block, then the communication overhead can be reduced substantially. In this type of an approach, verification of each packet is not guaranteed and instead is assured with a certain probability.

Each packet would include hashes of the previous packets. The signature packet, which contains the hashes of the final few packets along with a signature, is sent at the end of the stream to authenticate all the packets. Tolerance to loss can be increased further by sending multiple copies of a signature packet—copies would be sent with delayed intervals, because packet loss is correlated. To reduce the verification delay at the receiver side, a stream of packets is divided into blocks, and the same process is repeated for every block, i.e., all the data packets within the block are chained with multiple hashes followed by an insertion of one or more signature packets.

Signature amortization method does not allow immediate verification; because the verification can be proceed only after receiving all related messages. Since the messages are related packet loss vulnerability is higher. And also it requires expensive operation for signature generation and verification.

E. One time signature methods(OTS)

In cryptography Lamport one time signature scheme is a method for constructing a digital signature. Lamport signatures can be built from any cryptographically secure one-way function. Usually a cryptographic hash function is used. Lamport signatures with large hash function is more secure than other schemes but Lamport key can be used to sign a single message. Since they rely on one way hash functions without any trapdoors they are publically verifiable and also computationally efficient. But it requires large public and private key sizes. Then more schemes have been proposed. It includes BiBa, HORS TV-HORS etc.

BiBa use one way hash function without trap doors. It features low verification overhead and relatively small signature size. But it requires large public key and signature generation overhead. BiBa stands for Bins and Balls signature. BiBa exploits birthday paradox such that signer has more balls to which results in a high probability to find a signature but adversary has few balls so has low probability to forge a signature. BiBa is useful in systems where signer can send public to verifier efficiently and the verifier is constrained with computational power. But the public key size and signature generation overhead is more in BiBa.

HORS which is hash to obtain random subset is a modification of BiBa. It maintains the verification efficiency of BiBa and also it achieves speeding up of signing times. HORS uses a cryptographically strong hash function H to map each message M to a K element subset of a T element set. The private key is T, the public key is the set created by applying a one-way function to each element of T and the signature is the K element subset that is mapped to M. Since it is a onetime signature scheme distribution of key must be done at each time that cause higher communication overhead. And size of the public remains large which causes certification or chaining overheads.

TV-HORS is an extension of time valid one time signature schemes. The basic idea is applying TV_OTS model to HORS signature scheme to make it to a time valid v-time signature scheme, and then using one-way hash chains to link multiple key pairs together to enable authentication of a large number of streaming packets. TV-HORS provides short end-to-end computational latency, perfect tolerance to packet loss, and strong resistance against malicious attacks. But it requires large key size which makes it impractical for resource constrained system.

F. Online offline signatures

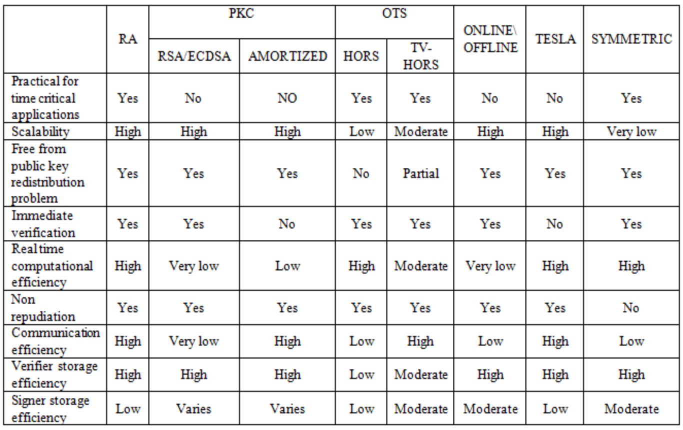
Here signature generation is performed in two phases. First phase is the offline phase which is before actual message to be signed is given. Second is the online phase is performed after obtaining the message to be signed. On-line/offline signature schemes are useful, since in many applications the signer has a very limited response time once the message is presented, but able to carry out costly computations between consecutive signing requests. In hash sign switch scheme hash the given message using a trapdoor hash value and then sign the hashed value using a given signature scheme. Offline phase uses original signature scheme to sign the hash value of random message and random number pair. Online phase uses that precomputed signature to sign the original message by using the trapdoor to find the collision of hash values, ie hash of random number and random number should be same as the hash value of original message and newly selected random number. Online phase is completely independent of original signature scheme and consists only of finding a collision of trapdoor hash function. Expensive computations required to generate the signature are performed at offline phase. And thereby it removes computation overhead at the online phase. Some system use one time signature scheme as the building block. Such system inherits all the drawbacks of that system.

G. *Rapid authentication (RA)*

RA is meant for authenticating command and control messages in a broadcasting environment. It make use of online offline signature method and uses RSA and Condensed RSA for key generation, signature generation and verification. It uses already existing structures in command and control messages for generating signature at the offline phase. During online phase signature generated at the offline phase is combined with the original message using condensed RSA scheme. The idea of RSA is based on the fact that number of possible sub messages in a command and control messages are limited. So it is possible to pre compute RSA signature on those sub messages during the offline phase. Verification is also efficient because it uses RSA verification scheme with few modular multiplication. It also provides a signature masking technique to secure individual message signatures that are combined during the online phase. It achieves several desirable properties including fast signature generation and verification, immediate verification without verification, small key and signature sizes, high scalability, high packet loss tolerance, provable security and being free from synchronization requirement.

Table 1 give a high level comparison of various authentication schemes.

TABLE 1



	RA	PKC		OTS		ONLINE\ OFFLINE	TESLA	SYMMETRIC
		RSA/ECDSA	AMORTIZED	HORS	TV-HORS			
Practical for time critical applications	Yes	No	NO	Yes	Yes	No	No	Yes
Scalability	High	High	High	Low	Moderate	High	High	Very low
Free from public key redistribution problem	Yes	Yes	Yes	No	Partial	Yes	Yes	Yes
Immediate verification	Yes	Yes	No	Yes	Yes	Yes	No	Yes
Real time computational efficiency	High	Very low	Low	High	Moderate	Very low	High	High
Non repudiation	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
Communication efficiency	High	Very low	High	Low	High	Low	High	Low
Verifier storage efficiency	High	High	High	Low	Moderate	High	High	High
Signer storage efficiency	Low	Varies	Varies	Low	Moderate	Moderate	Low	Moderate

III. CONCLUSION

In this paper we discussed about various cryptographic system for broadcast authentication. Symmetric cryptographic system is not publically verifiable and also impractical for large system. Public cryptographic systems require expensive modular multiplication which makes them impractical for resource constrained systems. Delayed key disclosure methods cannot achieve immediate verification and also require time synchronization between sender and receiver, so it cannot be used in large and real time systems. Signature amortization technique cannot achieve immediate verification. One time requires large key and signature sizes, so it cannot be used in resource constrained systems. RA which is based on RSA and condensed RSA achieves fast signature generation and verification, immediate verification without verification, small key and signature sizes, high scalability etc .But it can be applied to the system in which messages having a predefined structure. It can be used for real time authentication of command and control messages.

REFERENCES

[1] M. LUK, A. PERRIG, AND B. WHILLOCK, "SEVEN CARDINAL PROPERTIES OF SENSOR NETWORK BROADCAST AUTHENTICATION," IN *PROC. 4TH ACM WORKSHOP SECURITY*

[2] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*. Boca Raton, FL, USA: CRC Press, 1996

[3] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, no. 2,pp. 120–126, 1978

*[4] ANSI X9.62-1998: Public Key Cryptography for the Financial Services Industry: The Elliptic Curve Digital Signature Algorithm (ECDSA).*Washington, DC, USA: American Bankers Association, 1999.

[5] D. Liu, P. Ning, S. Zhu, and S. Jajodia, “Practical broadcast authentication in sensor networks,” in *Proc. 2nd Annu. Int. Conf. Mobile Ubiquitous Syst., Netw. Services (MobiQuitous)*, Jul. 2005, pp. 118–129.

[6] A. Perrig, R. Szewczyk, J. D. Tygar, V.Wen, and D. E. Culler, “SPINS: Security protocols for sensor networks,” *Wireless Netw.*, vol. 8, no. 5,pp. 521–534, Sep. 2002.

[7] A. Perrig, “The BiBa one-time signature and broadcast authentication protocol,” in *Proc. 8th ACM Conf. Comput. Commun. Security*,Nov. 2001, pp. 28–37.

[8] L. Reyzin and N. Reyzin, “Better than BiBa: Short one-time signatures with fast signing and verifying,” in *Proc. 7th Austral. Conf. Inf. Security Privacy (ACIPS)*, 2002, pp. 144–153.

[9] A. Shamir and Y. Tauman, “Improved online/offline signature schemes,” in *Proc. 21st Annu. Int. Cryptol. Conf.*, 2001, pp. 355–367.

[10] Attila Altay Yavuz, “*an efficient real time broadcast thentication scheme for command and control messages.”* IEEE transactions on information forensics and security, vol. 9, no. 10, october 2014

[11] J. M. Park, E. K. P. Chong, and H. J. Siegel, “Efficient multicast packet authentication using signature amortization,” in *Proc. IEEE Symp. Security Privacy*, May 2002, pp. 227–240.