Middle-East Journal of Scientific Research 20 (4): 532-536, 2014
ISSN 1990-9233
© IDOSI Publications, 2014
DOI: 10.5829/idosi.mejsr.2014.20.04.12041

Authenticated Multicasting in Mobile Devices Using Batch Signature

Anjali Jose and S. Vinoth Lakshmi

Department of Information and Technology, Tagore Engineering College,
Bharath University, Anna University, Bharath University, India

Abstract: Conventional block-based multicast authentication schemes overlook the heterogeneity of receivers by letting the sender choose the block size, divide a multicast stream into blocks, associate each block with a signature and spread the effect of the signature across all the packets in the block through hash graphs or coding algorithms. The correlation among packets makes them vulnerable to packet loss, which is inherent in the Internet and wireless networks. Moreover, the lack of Denial of Service (*DoS*) resilience renders most of them vulnerable to packet injection in hostile environments. In this paper, a novel multicast authentication protocol, namely MABS is proposed. This eliminates the correlation among packets and thus provides the perfect resilience to packet loss and it is also efficient in terms of latency, computation and communication overhead due to an efficient cryptographic primitive called batch signature, which supports the authentication of any number of packets simultaneously.

Key words: Multicast · Autentication · Signature · MABS

INTRODUCTION

Multicast is an efficient method to deliver multimedia content from a sender to a group of receivers and is gaining popular applications such as real time stock quotes, interactive games, video conference, live video broadcast, or video on demand. Authentication is one of the critical topics in securing multicast in an environment attractive to malicious attacks. Basically, multicast authentication may provide the following security services; data integrity, data origin authentication, non-repudiation [1].

All the three services can be supported by an asymmetric key technique called signature. In an ideal case, the sender generates a signature for each packet with its private key, which is called signing and each receiver checks the validity of the signature with the sender's public key, which is called verifying. If the verification succeeds the receiver knows the packet is authentic [2].

Our target is to authenticate multicast streams from multiple senders to multiple receivers. Generally, the senders are the powerful multicast servers managed by a central authority and can be trustful. The sender signs each packet with a signature and transmits it to multiple receivers through a multicast routing protocol. Each receiver is a less powerful device with resource constraints and may be managed by a no trustworthy person. Each receiver needs to assure that the received packets are really from the sender (*authenticity*) and the sender cannot deny the signing operation (*non- repudiation*) by verifying the corresponding signatures [3].

Designing a multicast authentication protocol is not an easy task. Generally, there are following issues in real world challenging the design. First, efficiency needs to be considered, especially for receivers. Compared with the multicast sender, which could be a powerful server, receivers can have different capabilities and resources. The receiver heterogeneity requires that the multicast authentication protocol be able to execute on not only powerful desktop computers but also resource-constrained mobile handsets. In particular, latency, computation and communication overhead are major issues to be considered. Second, packet loss is inevitable [4].

In the Internet, congestion at routers is a major reason causing packet loss. An overloaded router drops

Corresponding Author: Anjali Jose, Department of Information and Technology, Tagore Engineering College,
Bharath University, Anna University, Bharath University, India.

buffered packets according to its preset control policy. Though TCP provides a certain retransmission capability, multicast content is mainly transmitted over UDP, which does not provide any loss recovery support. In mobile environments, the situation is even worse. The instability of wireless channel can cause packet loss very frequently. Moreover, the smaller data rate of wireless channel increases the congestion possibility. This is not desirable for applications like real-time online streaming or stock quotes delivering. End users of online streaming will start to complain if they experience constant service interruptions due to packet loss and missing critical stock quotes can cause severe capital loss of service subscribers. Therefore, for applications where the quality of service is critical to end users, a multicast authentication protocol should provide a certain level of resilience to packet loss. Specifically, the impact of packet loss on the authenticity of the already-received packets should be as small [5].

In view of the problems regarding the sender-favored block-based approach, we conceive a receiver-oriented approach by taking into account the heterogeneity of the receivers. As receiving devices have different computation and communication capabilities, some could be powerful desktop computers, while the others could be cheap handsets with limited buffers and low-end CPUs. Mixed with various channel loss rates, this heterogeneity poses a demand on the capability of adjusting the buffer size and authenticating buffered packets any time when the high layer application requires at each receiver [6].

Related Works:

Mulicast Content Distribution: This paper describes, multicast enables efficient large-scale content distribution by providing an efficient transport mechanism for one-to-many and many-to-many communication. The very properties that make multicast attractive, however, also make it a challenging environment in which to provide content security. It shows how the fundamental properties of the multicast paradigm cause security issues and vulnerabilities. It focuses on four areas of research in security for multicast content distribution: receiver access control, group key management, multicast source authentication and multicast fingerprinting. Also, it briefly highlight other security issues in multicast content distribution including source access control, secure multicast routing and group policy specification.

Multicast Data Origin Authentication: It states, multicasting is an efficient communication mechanism for group-oriented applications such as videoconferencing, broadcasting stock quotes, interactive group games and video on demand. The lack of security obstructs a large deployment of this efficient communication model. This limitation motivated a host of research works that have addressed the many issues relating to securing the multicast, such as confidentiality, authentication, non-repudiation, integrity and access control. Many applications, such as broadcasting stock quotes and video-conferencing, require data origin authentication of the received traffic. Hence, data origin authentication is an important component in the multicast security architecture. Multicast data origin authentication must take into consideration the scalability and the efficiency of the underlying cryptographic schemes and mechanisms, because multicast groups can be very large and the exchanged data is likely to be heavy in volume (*streaming*). Besides, multicast data origin authentication must be robust enough against packet loss because most multicast multimedia applications do not use reliable packet delivery. Therefore, multicast data origin authentication is subject to many concurrent and competitive challenges, when considering these miscellaneous application-level requirements and features. In this article we review and classify recent works dealing with the data origin authentication problem in group communication and we discuss and compare them with respect to some relevant performance criteria [7].

Babra: Batch-Based Broadcast Authentication in Wireless Sensor Networks states that to prevent adversaries from injecting bogus messages, authentication is required for broadcast in wireless sensor networks. *i*TESLA is a light-weight broadcast authentication protocol, which uses a one-way hash chain and the delayed disclosure of keys to provide the authentication service. However, it suffers from several drawbacks in terms of time synchronization, limited broadcast rounds, key chain management at the source node, etc. In this paper, they proposed a novel protocol, called BAtch-based BRoadcast Authentication (BABRA) for wireless sensor networks. BABRA does not require time synchronization, eliminates the requirement of key chain and supports broadcast for infinite rounds. BABRA is an efficient due to the use of symmetric key techniques [8].

Digital Signatures for Flows and Multicasts: This presents chaining techniques for signing/verifying multiple packets using a single signing/verification operation. Flow signing and verification procedures based upon a tree chaining technique are presented. Since a single signing/verification operation is amortized over many packets, these procedures improve signing and verification rates by one to two orders of magnitude compared to the approach of signing/verifying packets individually. These procedures do not depend upon reliable delivery of packets, provide delay-bounded signing and are thus suitable for delay-sensitive flows and multicast applications. To further improve the procedures, several extensions to the Feige-Fiat-Shamir digital signature scheme to speed up both the signing and verification operations are presented, as well as to allow "adjustable and incremental" verification. The extended scheme, called eFFS, is compared to four other digital signature schemes (*RSA, DSA, ElGamal, Rabin*). The comparison of signing and verification times, as well as key and signature sizes is done. It is observed that the signing and verification operations of eFFS are highly efficient compared to the other schemes, eFFS allows a tradeoff between memory and signing/verification time and eFFS allows adjustable and incremental verification by receivers.

Graph-based Authentication of Digital Streams: It describes the authentication of digital streams over a lossy network. The overall approach taken is graph-based, as this yields simple methods for controlling overhead, delay and the ability to authenticate, while serving to unify many previously known hash- and MAC-based techniques. The loss pattern of the network is defined probabilistically allowing both burst and random packet loss to be modeled. The authentication schemes are customizable by the sender of the stream; that is, within reasonable constraints on the input parameters, it provides schemes that achieve the desired authentication probability while meeting the input upper bound on the overhead per packet. In addition, it has been demonstrated that some of the shortcomings of previously known schemes correspond to easily identifiable properties of a graph and hence, may be more easily avoided by taking a graph-based approach to designing authentication schemes [9].

Butterfly-Graphy Based Stream Authentication: In butterfly-graph based stream authentication scheme for lossy networks, the streaming packets could be lost in both random and burst ways. Due to the nice properties of butterfly graph, the proposed scheme is quite robust and efficient. Theoretical analysis and simulation results show that the proposed scheme outperforms existing schemes in terms of overhead and authentication probability while maintaining the same levels of sender / receiver delay and robustness.

Signature Amortization: This describes a novel method for authenticating multicast packets that is robust against packet loss. It focuses to minimize the size of the communication overhead required to authenticate the packets. The approach is to encode the hash values and the signatures with Rabin's Information Dispersal Algorithm (*IDA*) to construct an authentication scheme that amortizes a single signature operation over multiple packets. This strategy is especially efficient in terms of space overhead, because just the essential elements needed for authentication (*i.e., one hash per packet and one signature per group of packets*) are used in conjunction with an erasure code that is space optimal. To evaluate the performance of the scheme, this technique is compared with four other previously proposed schemes using analytical and empirical results. Two different bursty loss models are considered in the analyses.

Video Stream Authentication: It is well known that packets may be lost when video stream is transmitted over wireless network. To authenticate real time multicast streams with less overhead but at higher probabilities, most of previous stream authentication schemes insert packet hashes into the packet bodies explicitly. This scheme enables to remove the packet hashes from the packet overhead in lossy networks. It encodes the packet data with single encoding operation and only encapsulates the parity symbols into the packets overhead. Thus, it reduces the communication overhead as well as encoding time.

Proposed System Overview

Multicast Establishment: The receiver requests the sender to join in that group to collect the data. The multicast group is established using MABS protocol. The request is in the form of IP address and port number. The receiver will also send its buffer size to the sender. The request will be send via a datagram socket. Sockets are used for interprocess communication. The IP address for multicast is of class D. Multicast Authentication using Batch Signature (*MABS*) utilizes an efficient asymmetric key primitive called batch signature. Supports authentication to any number of packets simultaneously.

MABS Uses Per-packet Signature Instead of Per-block: signature and thus eliminates the correlation among packets. The packet independency makes MABS perfect resilient to packet loss. The Internet and wireless channels tend to be lossy due to congestion or channel instability, where packets can be lost according to different loss models, such as random loss or burst loss. In MABS, however, no matter how many packets are lost, the already received packets can still be authenticated by each receiver. This is a significant advantage over previous schemes.

Signature Generation: Sender generates a signature using Batch BLS algorithm. Then it sends the public key and generator to the receiver. The sender generates a signature for each packet with its private key, which is called signing. The receiver checks the validity of the signature with the sender's public key, which is called verification. If the verification succeeds, the receiver knows the packet is authentic.

In most RSA implementations, the public key e is usually small while the private key d is large. Therefore, the RSA signature verification is efficient while the signature generation is expensive. This poses a challenge to the computation capability of the sender because the sender needs to sign each packet. Choosing a small private key d can improve the computation efficiency but compromise the security. If the sender does not have enough resource, a pair of {e,d} with comparable sizes can achieve a certain level of trade-off between computation efficiency and security at the sender part. If the sender is a powerful server, then signing each packet can be affordable in this scenario.

The BLS signature scheme uses a cryptographic primitive called pairing, which can be defined as a map over two cyclic groups G1 and G2, e : G1 _ G1 ! G2, satisfying the following properties:

Bilinear: For all u; v $\in$ G1 and a; b $\in$ Z, we have $e(u^a,v^b) = e(u,v)^{ab}$

Non-degenerate: For the generator g1 of G1, i.e., $g_1^p = 1\_G1$, where p is the order of G1, we have e (g1, g1) $\neq 1 \in$ G2

BLS Algorithm

Key Generation: Sender chooses a random integer 'x' and compute

$y = g1^x \in G1$ 'x'→ private
Key, 'y'→ public key

Signing:

- Given a message 'm', the sender first computes h=h(m), where h() is a hash function
- Then it computes $\sigma = h^x$ where σ is the signature of 'm'

Verification:

- Receiver first computes $h = h(m) \in G1$

  Then it check whether $e(h,y) = e(\sigma,g1)$
  i.e., $e(h,y) = e(h,g1^x) = e(h,g1)^x = e(h^x,g1) = e(\sigma,g1)$
  {Bilinear property=>$e(u^a,v^b) = e(u,v)^{ab}$}

Batch BLS Algorithm: Given 'n' packets $\{m_i, \sigma_i\}$, i=1,….,n, the receiver can verify the batch of BLS signatures by first computing $h_i = h(m_i)$, i=1,…….,n and then checking whether $e(\pi^n_{i=1} h_i,y) = e(\pi^n_{i=1} \sigma_{i,}g1)$. This is because if all messages are authentic then,

$$e(\pi^n_{i=1} h_i,y) = e(\pi^n_{i=1} h_i,g_1^x)$$
$$= e(\pi^n_{i=1} h_i^x,g1)$$
$$= e(\pi^n_{i=1} \sigma_{i,}g1)$$

One merit of the BLS signature is that it can generate a very short signature.

Data Multicasting: The receivers receives the data from the sender. After receiving the data, the receiver verify the signature for a batch of n packets. If the signature is verified the data is loaded to client and the receiver receives the authenticated packets. If the packets are not authentic, the receiver drops the packet.

CONCLUSION

To reduce the signature verification overheads in the secure multimedia multicasting, block-based authentication schemes have been proposed. Unfortunately, most previous schemes have many problems such as vulnerability to packet loss and lack of resilience to denial of service (*DoS*) attack. To overcome these problems, we develop a novel authentication scheme MABS. We have demonstrated that MABS is perfectly resilient to packet loss due to the elimination of the correlation among packets and can effectively deal with DoS attack. Moreover, we also show that the use of batch signature can achieve the efficiency less than or comparable with the conventional schemes. Finally, we further develop the batch signature scheme based on Batch BLS, which is more efficient than the batch RSA signature scheme.

REFERENCES

1. Judge, P. and M. Ammar, 2003. Security Issues and Solutions in Mulicast Content Distribution: A Survey, IEEE Network Magazine, 17(1): 30-36.
2. Challal, Y., H. Bettahar and A. Bouabdallah, 2004. A Taxonomy of Multicast Data Origin Authentication: Issues and Solutions, IEEE Comm. Surveys & Tutorials, 6(3): 34-57.
3. Zhou, Y. and Y. Fang, 2006. BABRA: Batch-Based Broadcast Authentication in Wireless Sensor Networks, Proc. IEEE GLOBECOM.
4. Wong, C.K. and S.S. Lam, 1998. Digital Signatures for Flows and Multicasts, Proc. Sixth Int'l Conf. Network Protocols (ICNP '98), pp: 198-209.
5. Miner, S. and J. Staddon, 2001. Graph-Based Authentication of Digital Streams, Proc. IEEE Symp. Security and Privacy (SP '01), pp: 232- 246.
6. Zhang, Z., Q. Sun, W.C. Wong, J. Apostolopoulos and S. Wee, 2006. A Content-Aware Stream Authentication Scheme Optimized for Distortion and Overhead, Proc. IEEE Int'l Conf. Multimedia and Expo (ICME '06), pp: 541-544.
7. Park, J.M., E.K.P. Chong and H.J. Siegel, 2002. Efficient Multicast Packet Authentication Using Signature Amortization, Proc. IEEE Symp. Security and Privacy (SP '02), pp: 227-240.
8. Wu, Y. and T. Li, 2006. Video Stream Authentication in Lossy Networks, Proc. IEEE Wireless Comm. and Networking Conf. (WCNC '06), 4: 2150-2155.
9. Kaliyamurthie, K.P., D. Parameswari and R. Udayakumar, 2013. Reducing Web crawler overhead using mobile crawler, Middle-East Journal of Scientific Research, 15(12): 1830-1833. ISSN, pp: 1990-9233.