Volume: 2, Issue: 8, 720-723
Aug 2015
www.allsubjectjournal.com
e-ISSN: 2349-4182
p-ISSN: 2349-5979
Impact Factor: 3.762

P Prabhakaran
Assitant Professor, Department of Computer Science, PSG College of arts and science, Coimbatore, India

S Nandhini
Research Scholar, Department of Computer Science, PSG College of arts and science, Coimbatore, India

Correspondence
P Prabhakaran
Assitant Professor, Department of Computer Science, PSG College of arts and science, Coimbatore, India

Secured Multicasting Technique Using Enhanced Mabs

P Prabhakaran, S Nandhini

Abstract
Multicast is an efficient method to deliver data from a sender to a group of receivers. Authentication is an important issue in multicast communication. 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. The approach of signing and verifying each packet independently raises a serious challenge to resource-constrained devices. In mobile environments, the situation is even worse. The instability of wireless channel can cause packet loss very frequently. The smaller data rate of wireless channel increases the congestion possibility. The congestion will lead to packet loss.MABS-B (Multicast Authentication based on Batch Signature) can be used to improve the performance of multicast authentication which supports the authentication of any number of packets simultaneously with one signature verification, to address the efficiency and packet loss problems. The basic scheme 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. The enhanced scheme combines MABS-B with packet filtering to alleviate the DoS impact in hostile environments. The existing technique MABS provides data integrity, origin authentication, and non-repudiation as previous asymmetric key based protocols. In addition, our proposed technique MABS-B can achieve perfect resilience to packet loss in lossy channels in the sense that no matter how many packets are lost the already-received packets can still be authenticated by receivers.

Keywords: MABS, MABA-B, Multicast Authentication, Batch Digital Signature, SHA-1

1. Introduction
In computer networking, multicast is the delivery of a message or information to a group of destination computers simultaneously in a single transmission from the source creating copies automatically in other network elements, such as routers, only when the topology of the network requires it. Multicast is most commonly implemented in IP multicast, which is often employed in Internet Protocol applications of streaming media and Internet television. In IP multicast the implementation of the multicast concept occurs at the IP routing level, where routers create optimal distribution paths for data grams sent to a multicast destination address. At the Data Link Layer, multicast describes one-to-many distribution such as Ethernet multicast addressing, Asynchronous Transfer Mode point-to-multipoint virtual circuits. It 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.

Overview of MABS
Designing a multicast authentication protocol is not an easy task. Generally, there are some issues in real world challenging the design. First, efficiency need should 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. In the Internet, congestion at routers is a major reason causing packet loss. An overloaded router drops 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

The 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 as possible.

Digital Signature Algorithm: This Standard specifies a Digital Signature Algorithm appropriate for applications requiring a digital rather than written signature. The DSA digital signature is a pair of large numbers represented in a computer as strings of binary digits. The digital signature is computed using a set of rules and a set of parameters such that the identity of the signatory and integrity of the data can be verified. The DSA provides the capability to generate and verify signatures. Signature generation makes use of a private key to generate a digital signature. Signature verification makes use of a public key which corresponds to, but is not the same as, the private key.

DSA Key Generation: Key generation has two phases. The first phase is a choice of algorithm parameters which may be shared between different users of the system: Choose an approved cryptographic hash function H. In the original DSS, H was always SHA-1, but the stronger SHA-2 hash functions are approved for use in the current DSS. The hash output may be truncated to the size of a key pair. Decide on a key length L and N. This is the primary measure of the cryptographic strength of the key. The original DSS constrained L to be a multiple of 64 between 512 and 1024 (inclusive). Recommends lengths of 2048 (or 3072) for keys with security lifetimes extending beyond 2010 (or 2030), using correspondingly longer N specifies L and N length pairs of (1024, 160), (2048, 224), (2048, 256), and (3072, 256).

Related Work

Network Model

In this module the network is constructed with one server, router and four clients. Client-server computing or networking is a distributed application architecture that partitions tasks or workloads between service provider's servers and service requesters, called clients. Often clients and servers operate over a computer network on separate hardware. A server machine is a high-performance host that is running one or more server programs which share its resources with clients. In this module the network is constructed by connecting the server with the four clients through a router using their IP address or using default local host.

Key Generation

In this module keys are generated. Key generation has two phases. The first phase is a choice of algorithm parameters which may be shared between different users of the system. Choose an approved cryptographic hash function H. In the original DSS, H was always SHA-1, hash functions are approved for use in the current DSS [7]. The hash output may be truncated to the size of a key pair. Decide on a key length L and N. This is the primary measure of the cryptographic strength of the key. The original DSS constrained L to be a multiple of 64 between 512 and 1024. In this module the module the input file is divided into number of blocks. Then using hash algorithm the message digests and keys are generated [6, 7].

Algorithm

The following steps are used to generate keys:

1. p, a prime longer than 512 bits.
2. q, a 160-bit prime divisor of p - 1.
3. g, a generator of Z, p with order q, i.e., g^p = 1 mod p.
4. x, the private key of the signer, 0 < x < q.
5. y, the public key of the signer, y = g^x mod p.
6. h(), a hash function generating an output in Z.

Digital Signature

Digital signatures employ a type of asymmetric cryptography. For messages sent through an insecure channel, a properly implemented digital signature gives the receiver reason to believe the message was sent by the claimed sender (Figure 1). In this system we are using DSA algorithm to generate digital signatures [11]. Digital signatures are equivalent to traditional handwritten signatures in many respects; properly implemented digital signatures are more difficult to forge than the handwritten type. Digital signature schemes [4] in the sense used here are cryptographically based, and must be implemented properly to be effective.

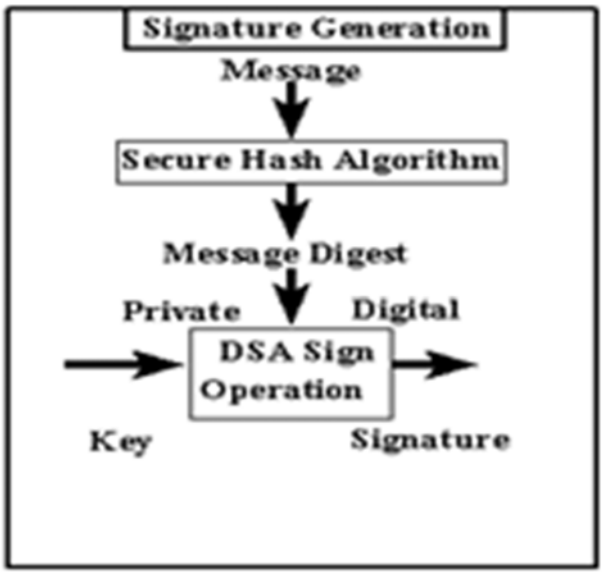

Fig 1: Signature Generation

Digital signatures can also provide non-repudiation, meaning that the signer cannot successfully claim they did not sign a message, while also claiming their private key remains secret [6]. Batch DSA can efficiently handle larger batches, but it has a tradeoff between batch size and key size. Each key variant requires specifying a full-sized y value, while with Batch RSA the variants just required listing a small e value. This will limit Batch DSA in most circumstances to similar batch sizes of on the order of tens of messages; otherwise the keys become unreasonably large.

Batch DSA for Signature Generation

Given a message m, the signer generates a signature by:

1. x randomly selecting an integer k with 0 < k < q,
2. Computing h = h(m)
3. Computing r = (g^k mod p) mod q, and
4. Computing s = rk-hx mod q. The signature form is (r, s).

Signature Verification

Signature verification may be performed by any party using the signatory's public key (Figure 2). This signature verification is done at the client side [9]. It is done by using DSA signature verification. A signatory may wish to verify that the computed signature is correct, perhaps before sending the signed message to the intended recipient. The intended recipient verifies the signature to determine its authenticity. Prior to verifying the signature of a signed message, the domain parameters, and the claimed signatory's public key and identity shall be made available to the verifier in an authenticated manner [2, 3].

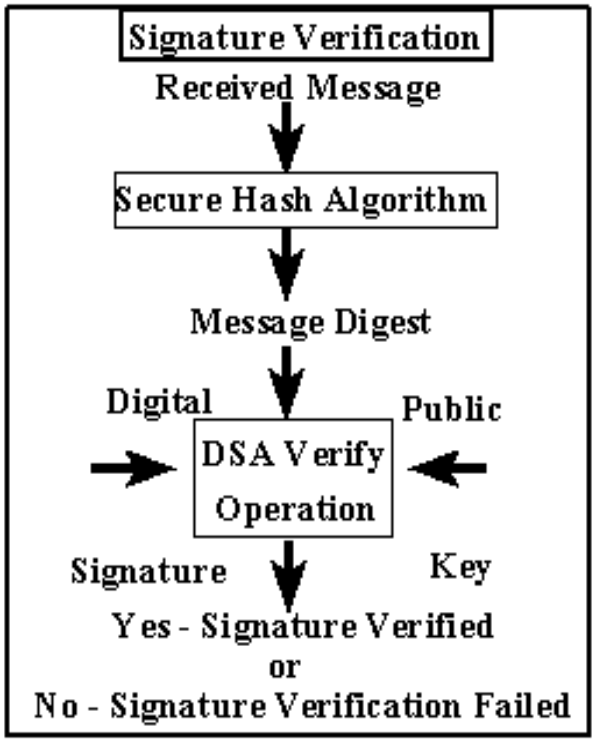

Fig 2: Signature Verification

3.5 Batch DSA for Signature verification

The receiver can verify the signature by first computing [8]
h = h(m) and then checking whether
$((g^{sr-1}y^{hr-1}$) mod p) mod q = r This is because if the packet is authentic, then
$((g^{sr-1}y^{hr-1}$) mod p) mod q
$((g^{(s+hx)r-1}$) mod p) mod q
$(g^{k}$ mod p) mod q = r

Secure Hash Algorithm (SHA1)

The SHA1 encryption algorithm specifies a Secure Hash Algorithm (SHA1) which can be used to generate a condensed representation of a message called a message digest (Figure 3). Both the transmitter and intended receiver of a message in computing and verifying a digital signature uses the SHA1 [5]. When a message of any length $<2^{64}$ bits is input. The SHA1 produces a 160-bit output called a message digest. The message digest can then be input to the Digital Signature Algorithm (DSA), which generates or verifies the signature for the message. Signing the message digest rather than the message often improves the efficiency of the process because the message digest is usually much smaller in size than the message [5].

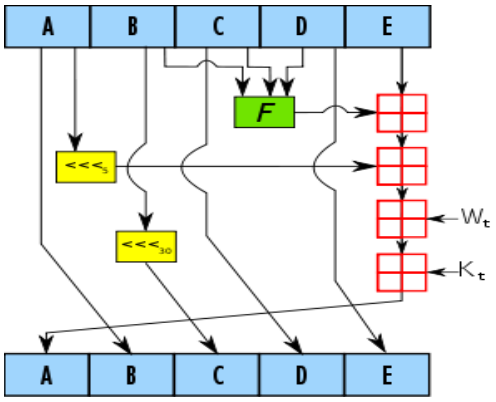

Fig 3: One iteration within the SHA-1 compression function

3.6 Experiments and Results

In this work, the input can be a text file or any other file. By clicking on the Browse button we can choose the file which is to be given as input. The selected file content will be displayed in the Browsed File list box. The selected file path will also be displayed. The network is constructed with one server, one router and four clients. For establishing connection between server and clients we are using socket program. First the server is connected with router using default IP address. The clients are also connected with router using the default IP address. The selected text file is split into number of blocks and the number of blocks split also displayed. After splitting the input file into blocks, the keys are generated.

Key generation has two phases. The first phase is a choice of algorithm parameters which may be shared between different users of the system. Second phase is choosing an approved cryptographic hash function. We are using SHA-1 algorithm for this process. The signature is embedded in each and every block. The signature is generated using DSA algorithm. Each block carries a signature. After the key generation and signing on each blocks, the files are sent to the clients through routers. And the signature verification is done at the client side. If the signature generated for each block at the client side using private key matches the signature for each block, the blocks are authenticated. If they don't matches, the blocks are not authentic and they are not sent by intended server. 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 attack. Figure 4 and 5 are the sample results produced using Java Swing with Eclipse Tool.

Fig 4: Key Generation

Fig 5: Verifying Authentication

Conclusion

The proposed, secured multicasting technique MABS-B can be used to improve the performance of multicast authentication which supports the authentication of any number of packets simultaneously with one signature verification, to address the efficiency and packet loss problems. To overcome the existing problems in MABS, we propose a novel authentication scheme MABS-B. This will generate one signature for each blocks and it will overcome the problems like congestion, communication overhead. MABS-B can be perfectly resilient to packet loss due to the elimination of correlation among packets. This project can be enhanced by implementing enhanced scheme called MABS-E, which combines the basic scheme MABS-B and a packet filtering mechanism to tolerate packet injection. In particular, the sender attaches each packet with a mark, which is unique to the packet and cannot be spoofed. The mark design ensures that a packet from the real sender never falls into any set of packets from the attacker, and vice versa.

References

1. A. Perrig, R. Canetti, J.D. Tygar, and D. Song, "Efficient Authentication and Signing of Multicast Streams over Lossy Channels," Proc. IEEE Symp. Security and Privacy (SP '00), pp. 56- 75, May 2000.
2. C. Boyd and C. Pavlovski, "Attacking and Repairing Batch Verification Schemes," Proc. Sixth Int'l Conf. Theory and Application of Cryptology and Information Security Advances in Cryptology (ASIANCRYPT '00), pp. 58-71, Dec. 2000
3. C.H. Lim and P.J. Lee, "Security of Interactive DSA Batch Verification," IEE Electronic Letters, vol. 30, no. 19, pp. 1592-1593,
4. C.K. Wong and S.S. Lam, "Digital Signatures for Flows and Multicasts," Proc. Sixth Int'l Conf. Network Protocols (ICNP '98), pp. 198-209, Oct. 1998.
5. D. Eastlake and P. Jones, "US Secure Hash Algorithm 1 (SHA1)," RFC 3174, Sept. 2001.
6. D.Naccache, D. M.Raihi, S. Vaudenay, and D. Raphaeli, "Can D.S.A. be improved? Complexity Trade Offs with the Digital Signature Standard," Proc. Workshop Theory and Application of Cryptographic Techniques Advances in Cryptology (EUROCRYPT '94)", pp. 77-85, May 1995.
7. FIPS PUB, "Digital Signature Standard (DSS)", May 1994.
8. L. Harn, "Batch Verifying Multiple DSA-Type Digital Signatures," IEE Electronic Letters, vol. 34, no. 9, pp. 870-871, Apr. 1998.
9. L. Harn, "DSA-Type Secure Interactive Batch Verification Protocols," IEE Electronic Letters, vol. 31, no. 4, pp. 257-258, Feb. 1995.
10. P. Judge and M. Ammar, "Security Issues and Solutions in Mulicast Content Distribution: A Survey," IEEE Network Magazine, vol. 17, no. 1, pp. 30-36, Jan./Feb. 2003.
11. R. Gennaro and P. Rohatgi, "How to Sign Digital Streams," Information and Computation, vol. 165, no. 1, pp. 100-116, Feb. 2001.
12. R. Gennaro and P. Rohatgi, "How to Sign Digital Streams," Proc. 17th Ann. Cryptology Conf. Advances in Cryptology (CRYPTO '97), Aug. 1997.
13. Yun Zhou, Xiaoyan Zhu, and Yuguang Fang, Fellow, IEEE. "MABS: Multicast Authentication Based on Batch Signature". geared