Stream Authentication Using Trapdoor Hash Function

Dodla Padmaja[1], P. Srilatha[2]

[1]M.Tech student, Dept of Computer Science and Engineering, Anurag Group of Institution, Hyderabad, India

[2]Assistant Professor, Dept of Computer Science and Engineering, Anurag Group of Institution, Hyderabad, India

Abstract: ***Web based services engage distribution of contents like digital audio, video, games, software, stock quotes, Streaming presentations and live news feeds throughout distributed networking technologies, like Content Distribution Networks (CDN's), multicast networks, and peer-to-peer networks. We guard delay sensitive streams against malicious attacks, auditing mechanisms and security mechanisms need to be designed to efficiently process long sequence of bits. We have proposed a novel signature amortization technique based on trapdoor hash functions for authenticating each and every individual data blocks in the stream. Our technique affords for each and every intermediate blocks in the stream we want to stay away from the transmission loss and we will afford constant memory requirements for sender as well as receiver and we want to authenticate and verify the stream to keep away from unauthenticated user and to avoid malicious content.***

Keywords: Stream authentication, cryptography, content distribution network, trap door functions**.**

1. Introduction

In this paper, we have focused on the problem of efficient stream authentication and stream verification and auditing using digital signatures. The goal is to afford [1] origin authentication, integrity, non repudiation and auditing each and every individual data blocks that comprise a digital stream.

2. Problem Definition

The Problem which has faced by efficient authentication of stream poses several challenges:

1. The first problem which has faced by authentication of delay sensitive streams requires more verification rates for verify each and every individual data block in the stream[2].
2) The second problem which has faced by stream authentication for signature and other hash value mechanisms requires high excessive bandwidth utilization and requires high and more size for transmitted signed streams.
3) The third Problem for stream authentication for transmission of stream using untrustworthy transmission protocols like User Datagram protocol (UDP) leads to loss of datagram's during transmission.

In Existing approach the problems faced by stream authentication should be solved for one sender and receiver. Each sender and receiver must be of the same opinion on a secret code with message authenticating code (MAC) to ensure authenticating each and every packet. In case of the several receivers it is harder to solve the symmetric approach either sender or receiver desires to any one holding a key. In order to keep away from this proposed system use digital signature for sender to sign each and every packet with its private key.

In stream authentication security[3] problems are still harder with many receivers this directs to loss in the data streams. The data loss will be based on the bandwidth of the receivers. With high packet loss directs to low bandwidth. We want to ensure the authenticity of the data in the loss of high packets. We have to design the stream authentication without failure of data blocks for the receiver.

In Existing paper for stream authentication they proposed different schemes

1. The first scheme is TESLA (Timed Efficient Stream Loss-tolerant Authentication)[4] and it suggests authentication for the sender, provide high scalability and nominal overhead. It is using symmetric cryptographic primitives for a Message authentication codes (MACs) and Pseudorandom Functions(PRFs) and it based on release of keys for the
2. Sender by time to time. The second scheme is EMSS (Efficient Multi-chained Stream Signature) [8]and it suggests origin of non repudiation and it affords high loss resistance and it provides the cost of slightly delayed verification. It is utilized to signing a
3. Limited number of special packets in a data stream. Each and every packet supposed to be linked as a signed packet via multiple hash chains. This can be achieved by affixing the hash of each packet and also include the appending hashes of earlier packets to the number of subsequent packets.

3. Proposed System

Digital streaming Internet applications such as online gaming, presentations, Multimedia playback, news feeds, and stock quotes involve end-users with very little tolerance for low data rates, high latency and playback interruption. To defend such delay sensitive streams against malicious attacks, security mechanisms require to be designed to efficiently process long sequence of bits. We study the problem of well-organized authentication for delay-sensitive and real-time streams commonly seen in multicast, content distribution and peer-to-peer networks. We proposed a novel signature amortization technique based on trapdoor hash functions for authenticating individual data blocks in a stream.

The advantages of the proposed technique:

- To tolerate the unusable arrival rates and the transmission losses must be resilient in intermediate blocks Without affecting the remaining blocks in the stream.
- The transmitting stream must be minimizes for the block verification process and the block signing process.
- The communication overhead must be limited while sending the authenticated message and for each block in the stream.
- The bandwidth must be limited with multiple blocks by sending authentication information.

4. System Architecture Overview

Fig 1 shows the system architecture of content distribution network. The elements include the core data center, web cache and it serving the various clients and the back end of the content distribution network is WAN or internet and the data centers. The caches must be distributed widely and serving the requested clients. Both the web caches and core data centers contains media server and the content distribution manager.

The media content must be stored in the media servers and it should serve the content in both the real time as well as on demand users. Clients can include [7] tablets, laptops and mobile phones.

A content distribution manager has the following functionalities:

- Auditing or Tracking or monitoring the content usage by the clients and accounting the usage of their Respective clients.
- The contents requested by the clients must be fetched from the media server and partition the file into multiple blocks and transmitting the same requesting blocks into appropriate clients.
- If the client has requested the digital media content the request must be sent to closest web cache of the client. If the web cache includes the request which will be ask by the client it should be fetched and transmits to the client.
- If the request is not present in the web cache means the request must be forwarded to the core data center and obtains the data and it must transmit to the appropriate client.

4.1 Content Uploading

Server should upload the multimedia content was given by the content provider and store in a media Server. The client can be allowed to upload the multimedia content after the registration process is done.

4.2 Stream Authentication

Stream authentication can help avoid some type of attacks by providing the capability to sign and confirm each block in the stream. All content initiates at the core data center and the stream signing mechanism is implemented at the core CDM as part of its content processing service. We assume the subsistence of a Public Key Infrastructure (PKI) responsible for generating certificates for the core CDM, and distributing the public key and the certificate of core CDM to all confirming entities. When a request appears at the core CDM, the content processing service recovers the content from the media server. The core CDM then divides the content it into a stream of blocks, signs each block (using a suitable signature amortization technique), situates the authentication information within the block, and it transmits the signed stream of blocks to the requesting entity.

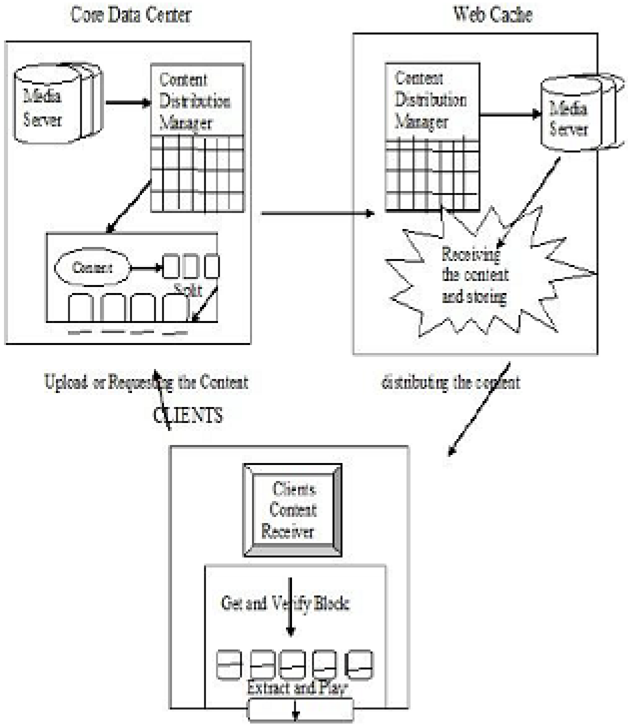

Figure 1: System Architecture Overview

If the content is not produced in real time, the content processing service stores up the signed stream at the media[8]server to avoid redundant signing operations when subsequent requests arrive for the same content.

4.3 Stream Verfication

If we interchange the concatenation [9] order of *r* and *m* in Equation (1) (the Definition of hash operation), the new.

Definition of hash operation is as follows:
$THHK(m; r) = grjjm \bmod n:$
Then the new trapdoor operation would be
$THTK(m1; r1;m2)$
$= r2$
$= 2¡l((m1 ¡ m2) + 2lr1)$
$= 2¡l(m1 ¡ m2) + r1 = x(m1 ¡ m2) + r1;$
where $x = 2¡l \bmod ¸(n):$

The setting of parameters are similarly to the setting of the scheme *TH* reviewed in Section 2, i.e. *n* = *PQ; P* = 2*p* + 1*;Q* = 2*q* + 1*;* and both *p* and *q* are large primes with *jpj* = *jqj*. An element *g* 2 *Z*¤ *n* with order ¸(*n*) = 2*pq* is selected. Then publish the hash key *HK* = (*g; n*). On the other hand, the secret trapdoor key *TK* is computed as *TK* = *x* = 2¡*l* mod ¸(*n*), where *l* is the security parameter, e.g. *l* = 160.

5. Features

We present a security and performance advantages of the authentication of online digitized signature for the authentication of the stream.

- The [10] packet loss must be robustness and the verification of each and every blocks must be depends on the authentication of the stream.
- The arbitrary defeat must be tolerated and the blocks containing signature ought to be reliably delivered to the receiver.
- Computation cost be supposed to be constant for the sender as well as receiver.
- The communication overhead must be constant and there are no multiple copies of the authenticating materials in the multiple blocks.
- The modification of the stream should be thwarted and the forgery signature should be avoided.

6. Future Enhancement of Online STREAM Authentication Technique

To allow users to be timely and exactly informed about their data usage, the distributed logging mechanism is complemented by an innovative auditing mechanism. We have supported the two complementary auditing modes:
1. Push mode,
2. Pull mode.

6.1 Push Mode

In this mode, the logs are periodically pushed to the data owner (or auditor) by the harmonizer. The push action will be generated by either type of the following two events: one is that the time elapses for a certain period according to the temporal timer inserted as part of the JAR file, another is that the JAR file go beyond the size stipulated by the content owner at the time of their creation. After the logs are sent to the data owner,[11] the log files can be dumped, so as to free the space for upcoming access logs. This mode serves two essential functions in the logging architecture:
a. It make sure that the size of the log files does not go off.
b. It allows timely detection and correction of any loss or damage to the log files.

6.2 Pull Mode

This mode can allow the auditors to retrieve the logs anytime, when they desire to check the recent access to the own data. The pull message consists simply of an FTP[12] pull command, which will be issues from the command line. For naïve users, the wizard comprising a batch file can be simply built. The request will be sent to the harmonizer, and the user can be informed of the data's locations and acquire an integrated copy of the authentic and sealed log file.

7. Conclusion

The authentication stream in the content distribution network avoids malicious modification or threats in the middle of the data transmission. The challenging task is the verification and signing for the on demand content and the tolerance next to the transmission loss and the communication overhead must be small per block. We present the authentication of online digitized signature using trap door hash function method that challenges that meet real time streaming in content distribution and afford efficient authentication of delay sensitive streams. Here, the authentication of online digitized signature method by authenticating from primary blocks in the stream using signature on the trap door hash function and by authenticating subsequent blocks in the stream.

References

[1] B.M. Luettmann and A.C. Bender, "Man-in-the-Middle Attacks on Auto-Updating Software," Bell Labs Technical J., vol. 12, no. 3, pp. 131-138, 2007.

[2] Akamai, "Akamai Information Security Management System Overview: Securing the Cloud," White Paper, http://stagwwwweb01.akamai.com/dl/whitepapers/Akamai_ ISMS.pdf?campaign_id=AANA-65TPAC, 2012.

[3] H. Krawczyk, and T. Rabin, \Chameleon signatures," *Symposium on Network and Distributed Systems Security*, pp. 143-154, 2000..

[4] T. Okamoto, M. Tada, and A. Miyaji, \Efficient 'on the °fly' signature schemes based on integer factoring," *Proceedings of the 2nd International Conference on Cryptology in India, Indocrypt*, LNCS 2247,pp. 275-286, 2001 [5] K. Skaugen, "Cloud 2015," Proc. Interop, http://www.interop.com/lasvegas/2011/presentations/free/136-kirk-skaugen.pdf, 2012.

[5] Cisco, "Cisco Visual Networking Index: Global Mobile DataTraffic Forecast Update, 2011-2016," White Paper, http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ ns827/-520862.pdf, 2012.

[6] D. Grabham, "Intel: New Server Needed for Every 120 TabletsSold,"Techradar,http://www.techradar.com/news/computingcomponents/processors/intel-new server-needed-for-every-120-tablets-sold-1069021, 2012.

[7] A. Shamir and Y. Tauman, "Improved Online/Offline Signature Schemes," CRYPTO '01: Proc. 21st Ann. Int'l Cryptology Conf.,pp. 355-367,2001.

[8] G. Brassard, D. Chaum, and C. Cre´peau, "Minimum Disclosure Proofs of Knowledge," J. Computer and System Sciences, vol. 37,no. 2, pp.156-189, 1988.

[9] H. Krawczyk and T. Rabin, "Chameleon Signatures," Proc.Network and Distributed System Security Symp. (NDSS), 2000.

[10] S. Even, O. Goldreich, and S. Micali, "Online/Offline Digital Schemes," CRYPTO: Proc. Ninth Ann. Int'l Cryptology Conf., pp. 263-275, 1989.

[11] R. Rivest, A. Shamir, and L. Adleman, \A method for obtaining digital signature and public key

cryptosystems," *Communications of the ACM*, vol. 21,no. 2, pp. 120-126, 1978.

Author Profile

Dodla Padmaja received the B. Tech degree in computer science and Engineering from Dr.M.G.R University, Chennai, 2008 and pursuing M. Tech degree in Computer science and Engineering from JNTU Hyderabad.

P. Srilatha is a assistant professor in CVSR College of Engineering from Anurag Group of Institutions, Hyderabad, Before this she was a lecturer in Eswari Engineering College, Chennai. She has received the Bachelor of Technology in Computer Science and Engineering from Shadan College, JNTU, Hyderabad and M. Tech in Software Engineering from School of IT, JNTUH, Hyderabad..