IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 2, Issue 2, Apr-May, 2014
ISSN: 2320 – 8791 (Impact Factor: 1.479)
www.ijreat.org

Secure Web-Based Service in Content Distribution Network

Rajeswari.K[1], Kavitha.M[2]

[1, 2]**Department of Computer Science and Engineering, Sriram Engineering College, Perumalpattu–602 024, Tamil Nadu, India**

Abstract

Many web-based services involve distribution of content like digital audio, video, software, games, stock quotes, streaming presentations, and live news feeds through distributed networking technologies, like content distribution networks (CDN's), multicast networks, and peer-to-peer networks. Unfortunately, these modern distributed systems, designed to distribute content to a large group of users, also provide a platform for adversarial users to launch a myriad attacks with widespread consequences. Adversaries can masquerade as legitimate content providers to distribute malicious content possibly infected with worms, viruses, etc. Authenticating the content plays a crucial role in preventing these attacks. The task of protecting content is the responsibility of content providers. Today majority of mass-viewed content is dynamically generated and rich in multimedia that include combination of text, audio, still images, animation, video, and interactive content forms that are gathered from a myriad of sources, assembled and presented to the user. In such scenarios, malicious modification of content by malicious sources becomes a legitimate threat.

Keywords: *Stream authentication, cryptography, content distribution, digital signatures, signature amortization, trapdoor hash functions.*

1. INTRODUCTION

Adversaries can masquerade as legitimate content providers to distribute malicious content possibly infected with worms, viruses, etc. Adversaries can also place themselves in the content distribution path, for example, by compromising in the content distribution path, for example, by compromising web caches [1], and modify the content in ways that can potentially harm client devices. Authenticating the content plays a crucial role in preventing these attacks. Although CDNs like Akamai employ mechanisms for providing physical security, host system security, access control, software reliability and integrity, and monitoring and role in preventing these attacks. Although CDNs like Akamai employ mechanisms for providing physical security, host system security, access control, software reliability and integrity, and monitoring and response, these mechanisms are primarily designed for providing security of the CDN's service network infrastructure and ensuring proper functions. Today majority of mass-viewed content is dynamically software reliability and integrity, and monitoring and response; these mechanisms are primarily designed for providing security of the CDN's service network infrastructure and ensuring proper functions.

Today majority of mass-viewed content is dynamically generated and rich in multimedia that include combination of text, audio, still images, animation, video, and interactive content forms that are gathered from a myriad of sources, assembled and presented to the user. In such scenarios, malicious modification of content by malicious sources becomes a legitimate threat.

It highlight this point, recently, Google and MSN (Microsoft(R)) were observed to be distributing malware after attackers were able to trick the networks by masquerading as a legitimate advertising provider and inserting malicious advertisements (by exploiting two Internet Explorer, one Java, and four Adobe Reader flaws) that installed the HDD Plus malware [3].

YouTube was also a victim of an attack where malicious code was inserted into pages (by exploiting a cross-site scripting vulnerability) displaying the targeted videos that would launch When users opened the video clip redirecting users to pornographic sites and display falsified news alerts. In conventional techniques for message authentication require the sender and the receiver to have the ability to store the entire message before processing the message. However, in most instances of distributing content like digitized multimedia, the

content provider transmits the content in the form of digital streams that receivers consume at more or less the stream arrival rate without excessive delay.

These trends strongly motivate the need to design a stream authentication scheme that incurs low overheads for signing (to reduce server load), communication (to reduce bandwidth consumption), and verification (to ensure mobile devices can cope with the additional processing).

Authentication of delay-sensitive streams requires high verification rates which translate to requiring minimal computational overhead to verify individual blocks and avoiding excessive accumulation of data in buffers before verification can proceed. For instance, to maintain jitter-free playback of on demand media distributed through CDNs, per-block verification rates at client devices must equal or exceed the rate at which blocks arrive at the device. at which blocks are generated.

Stream transmission is typically done using unreliable transport protocols like UDP to provide a high throughput, which can cause loss of diagrams during transmission. Thus, stream authentication mechanisms must be designed to tolerate arbitrary loss of data blocks without affecting the ability of a receiver to verify remaining blocks.

2. SECURING CONTENT DISTRIBUTION: A MOTIVATING APPLICATION

It present an application of using signature amortization for stream authentication in securing on demand and real-time content typically distributed via CDN's. We highlight specific challenges associated to efficiently authenticating content and later, present a viable approach towards solving existing problems. This application is not meant to be the sole motivation for the signature amortization mechanism that we are introducing, but only an illustration of some of its potential uses and benefits. The proposed signature amortization technique can also be applied to provide authentication in multicast and peer-to-peer networks.

2.A. System Architecture Overview

It shows a high-level architecture of a CDN. the components include a core data center, multiple edge caches each serving multiple clients, and the CDN backbone (Internet or WAN). Edge caches are strategically distributed worldwide to lower content delivery costs to requesting clients. Both core data centers and edge caches consist of a media server and a content distribution manager (CDM).

A CDM provides several functionalities including. Usage tracking service that enables logging, accounting, and billing of content usage. Caching service that implements content caching techniques. Content processing service that fetches the requested content from the media server (or stores content into the media server), splits the media file into blocks (or packet flows), and transmits the blocks to other edge caches or to clients. Request processing service that provides navigation of the CDN to locate the content.

When a client requests or subscribes to digital media content, the request is propagated to the edge cache closest to the client. If the cache contains the requested content, the edge cache transmits the content to the client.

If not, then the request is forwarded to the core data center, which sends the content to the client. If content is cacheable, the edge cache fetches the content from the core data center (or alternatively, the core data center pushes cacheable content to the edge cache) using an appropriate caching strategy to serve future requests for the content.

It present an application of using signature amortization for stream authentication in securing on demand and real-time content typically distributed via CDN's. We highlight specific challenges associated to efficiently authenticating content and later, present a viable approach towards solving existing problems.

This application is not meant to be the sole motivation for the signature amortization mechanism that we are introducing, but only an illustration of some of its potential uses and benefits. The proposed signature amortization technique can also be applied to provide authentication in multicast and peer-to-peer networks.

Content processing service that fetches the requested content from the media server (or stores content into the media server), splits the media file into blocks (or packet flows), and transmits the blocks to other edge caches or to clients. Request processing service that provides navigation of the CDN to locate the content.

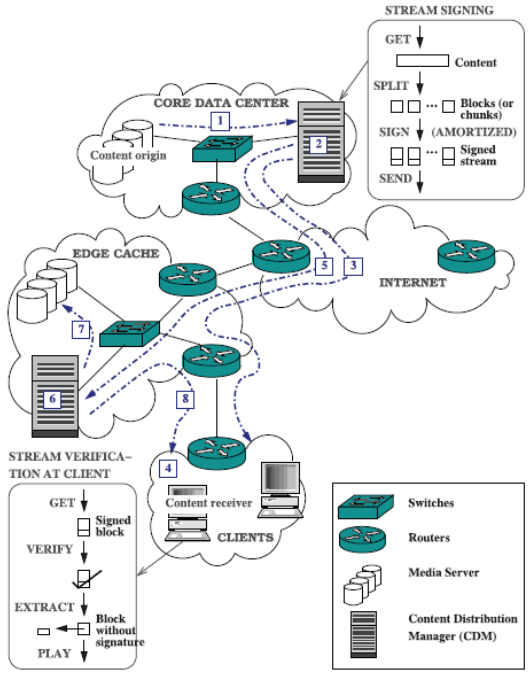

Fig. 1 Architecture of content distribution network

2.B. Stream Authentication in CDN's

Threats involved in distribution of content include: (1) Masquerading attacks, where an adversary poses as an edge cache or a core data center or a third-party provider that serves data for dynamically generated content, to distribute malicious content. Compromising attacks, where an adversary takes control of legitimate content providing hosts (edge cache/ core data center/third-party provider) to inject malicious content.

Man-in-the-middle attacks, where an adversary performs modification of content during transmission from core data center to the edge cache or from the edge cache to the client or from the core data center to the client. Stream authentication can help prevent these attacks by providing the ability to sign and verify each block in the stream. All content originates at the core data center and the stream signing mechanism is implemented at the core CDM as part of its content processing service.

The stream verification mechanism should be implemented by both the CDM at an edge cache and at the client machine. When a signed stream arrives at an edge cache, the content processing service of the CDM verifies individual blocks in the stream stores the signed content into the cache's media server and later, when a request arrives for the content sends the signed stream to the requesting client.

3. PROPOSED SIGNATURE AMORTIZATION TECHNIQUE

In this section, we present a novel trapdoor hash-based signature amortization technique that can be used for authenticating digital streams in CDN's, multicast systems, and peer-to-peer networks. Before going into details of the proposed technique, we present some background information on trapdoor hash functions and its properties.

3.A. Background on Trapdoor Hash Functions

Trapdoor hash functions are collision-resistant hash functions associated with a special trapdoor key that enables the possessor of the key to find collisions between hashes of different messages. A trapdoor hashing scheme consists of four algorithms for parameter generation, key pair generation, hash generation, and collision computation.

The function TH is said to be associated with the (long-term) hash key HK. Our definition extends traditional definitions for trapdoor hashing schemes [8] by allowing constructions of trapdoor hash functions that can generate collisions using ephemeral keys. Security notions associated with trapdoor hashing schemes include collision forgery resistance, ephemeral collision forgery resistance, semantic security, and key exposure resistance that are formally defined in.

A trapdoor hashing scheme is said be key exposure resistant if given the system parameters params and the tuple it is computationally infeasible to compute the trapdoor key TK corresponding to the hash key HK.

3.B. Trapdoor Hash-Based Signature Amortization Technique

The proposed signature amortization technique works by authenticating the initial block of a stream using a signature on trapdoor hash of the block's contents, and authenticating subsequent blocks of the stream by finding trapdoor collisions with the hash of the signed (initial) block. The PKI is responsible for distribution and maintenance of public keys and certificates to all entities in the system. We assume that each block is augmented with information that will help prevent replay attacks. This information can include session identifier and component numbers (like nonce, time stamps, etc.) Within each block along with mechanisms to check the freshness of the information.

The proposed signature amortization technique can be divided into two phases: Stream signing and stream verification. During the stream signing phase, the sender generates the authenticating information for each block in the stream as follows: Initial block generation. The sender generates a signature using SK on the trapdoor of the contents m0 of first block p0 in the stream. Otherwise, the receiver sends an error message to the signer and aborts.

3.C. Features of the Proposed Signature Amortization Technique

It present a brief discussion on security and performance merits of the proposed trapdoor hash-based signature amortization technique for stream authentication. Robustness against packet loss.

In the proposed technique, verification of a block only depends on the authenticating information contained in that block itself and no other block in the stream to compute the trapdoor hash value of the contents mi of a block pi, the verifier needs to know ri which is contained within block pi.

Moreover, verification of each subsequent block pi depends on trapdoor hash value of any block pj received prior to pi.Thus, the proposed technique can tolerate arbitrary loss of blocks in the stream as long as the initial block containing the signature is reliably delivered to the receiver—any subsequent block can be verified by comparing its trapdoor hash value with the trapdoor hash value of the initial block.

In the proposed technique, verification of a block only depends on the authenticating information contained in that block itself and no other block in the stream to compute the trapdoor hash value of the contents mi of a block pi, the verifier needs to know ri which is contained within block pi.

TABLE 1
The DL-MTH Trapdoor Hashing Scheme

Algorithm	Description
ParGen	Entities choose and agreed upon common system public parameters Params = {p, q, a.H} where p, q are 1024- bits, 160-bit primes respectively, ql p-1, a is an element of order q in Z* p and H ;{ 0, 1}*-> Z*q is a cryptographic hash function.
KeyGen	An entity uses system public parameters params, to generate is (private), Trap door key and (public) hash key pair, (TK, HK)
HashGen	An entity generates a trap door hash of a message using a hash key, by choosing an element and computing hash as a key. The function TH **is** associated with hash hey Y.
TrapColGen	Given the trap door key and hash key pair, and an additional message

4. AN EFFICIENT DL-BASED TRAPDOOR HASH FUNCTION

The DL-MTH trapdoor hashing scheme achieves key exposure-freeness by splitting the (trapdoor, hash) key pair into two components, one permanent and the

other ephemeral, and using the ephemeral keys to computer trapdoor hash collisions between digest values of two distinct messages. Unlike prior schemes by Chen et al.and the DL-MTH scheme ensures that it is computationally infeasible for a third party to compute an additional collision with the knowledge of a message pair whose trapdoor hash values are equal. In the schemes by an entity computes a trapdoor hash value using an ephemeral component, called a transaction identifier.

Thus, the schemes by Chen et al. and signer authenticates a new message by finding collisions between the hashes of the original signed-message and the hash of new message that needs to be signed. Unlike the schemes by Chen et al. and, the DL-MTH scheme allows collisions of the form the tuple hm; r; ;m0; r0;HK0i, does not allow a third party to forge additional collisions or reveal the trapdoor key .

5. ANALYSIS OF PROPOSED DISCRETE LOG-BASED SIGNATURE AMORTIZATION SCHEME

We now present a thorough analysis of the proposed signature amortization scheme DL-SA including its correctness, security, and performance evaluation.

5.A. Correctness

The proposed discrete-log-based signature amortization scheme, DL-SA is correct if the initial block p0 and all subsequent blocks pi in the stream pass the verification procedure at the receiver, provided. All entities honestly choose and agree upon the system public parameters params ¼ hp. The sender honestly executes the key generation algorithm to generate its public and private key pair

5.B. Security

Security of the proposed DL-based signature amortization scheme depends on the following two factors forgery resistance of the signature scheme used for signing the initial block in the stream and collision forgery, ephemeral collision forgery, and key exposure resistance of the trapdoor hashing scheme used for signing each subsequent block in the stream.

The proposed DL-based signature amortization scheme uses the Schnorr signature scheme to sign the initial block of the stream. Security of the Schnorr signature scheme against forgery is based on the following well-known theorem.

Theorem 1. The Schnorr variant of the ElGamal family of signatures is secure against an adaptive chosen message attack in the random oracle model under the discrete logarithm assumption.

5.C. Performance

In this section, we present an experiment-based performance evaluation of the proposed scheme, DL-SA. First, we provide a brief description of the system setup used for our experiments that simulates a wider-scale approach for real world content authentication. Next, we describe our approaches for improving the efficiency of the proposed scheme in our simulations. Finally, we compare the performance of the proposed scheme, DL-SA with the WL scheme and the CHFS
Schemes based on results obtained through our experiments.

We choose the WL and CHFS schemes for comparison as they are the only known signature amortization techniques in the literature that exhibit features that are comparable to the proposed scheme, which include, the ability to independently verify each block in the stream (and thus, are robust against arbitrary loss of intermediate blocks) and the capability to authenticate real-time streams. Shortcomings of other schemes in providing the independent verifiability and real-time authentication support are elaborated in Section 2.

Our performance comparison involves experiments to test the performance of each scheme which include the average per-block signing and verification timings, the average number of additional bits that needs to be appended to a block to provide authentication, and the size of the public key (representing the storage overhead) that is required for verifying the authenticity of each block in the stream.

On a wider-scale test bed, the signing and verification mechanisms will be implemented as shared libraries or objects (possibly remote) at the server-side and client-side as part of the operating system to allow loading the subroutines of a library into multiple applications at runtime. The flexibility provided by dynamic linking of shared libraries will permit multiple flavors of server-side CDM software and client-side stream playback software to utilize the authentication mechanism. Another alternative is to implement the stream authentication procedure as an extension to the transport layer protocols that are used to packet size the stream before transmission.

6. CONCLUSION

Mechanisms of flow authentication in content distribution networks help prevent masquerading attacks and malicious modification of content during transmission. However, efficient authentication of live, on-demand content is a challenging task, and requires fast signing and verification, tolerance against transmission loss and small per-block communication overhead.
They presented a novel trapdoor hash-based signature amortization technique that meets these challenges to provide efficient authentication of delay sensitive and real-time streams in content distribution, multicast, and peer-to-peer networks. Our signature amortization technique works by authenticating the initial block in a stream using a signature on the trapdoor hash of the block contents and authenticating subsequent blocks of the stream by finding trapdoor collisions with the hash of the signed initial block.

It demonstrated the efficacy of the proposed technique to build practical instances by presenting a discrete log-based instantiation of the proposed technique called DL-SA. The DL-SA scheme was designed to allow both individual verification of a single block in the stream and batch verification of multiple blocks to reduce the average verification cost per block.

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://stag-wwwweb01.
akamai.com/dl/whitepapers/Akamai_ISMS.pdf?campaign_id=AANA-65TPAC, 2012.

[3] P. Bright, "Google, Microsoft Distribute Malware After Domain Name Trickery," Ars Technica, http://arstechnica.com/security/ news/2010/12/google-microsoft-distribute-malware-after-domain- name-trickery.ars, 2010.

[4] A. Gonsalves, "YouTube Confirms Justin Bieber Hack Attack," InformationWeek, http://www.informationweek.com/news/
security/attacks/showArticle.jhtml?articleID= 225702490, 2010.

[5] K. Skaugen, "Cloud 2015," Proc. Interop, http://www.interop.com/lasvegas/2011/presentations/free/136-kirk-skaugen.pdf, 2012.

[6] Cisco, "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2011-2016," White Paper, http://www.
cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/
ns705/ns827/white_paper_c11-520862.pdf, 2012.
[7] D. Grabham, "Intel: New Server Needed for Every 120 Tablets Sold," Techradar, http://www.techradar.com/news/computingcomponents/
processors/intel-new-server-needed-for-every-120- tablets-sold-1069021, 2012.

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

[9] 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.

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