L-032

Integrity for the In-flight Web Page Using Fragile Watermarking

Peng Gao* Takashi Nshide† Yoshiaki Hori† Kouichi Sakurai†
*Graduate School of Information Science and Electrical Engineering,
Kyushu University, Japan, Fukuoka
Email: gao@itslab.csce.kyushu-u.ac.jp
† Graduate School of Information Science and Electrical Engineering,
Kyushu University, Japan, Fukuoka
Email: {nishide,hori, sakurai}@inf.kyushu-u.ac.jp

Abstract

In recent years, it has been found that middle modifications and attacks widely exist when web pages are transferred from a web server to a user based on HTTP. And the reason is that HTTP does not guarantee network traffic's integrity. This paper presents an idea and proposes an approach which provides integrity of the in-flight web page and can have better performances than HTTPS by supporting web cache technology. By employing fragile watermarking scheme, it can furthermore save transmission bandwidth and storage space of the web server than existed solutions.

Keyword-Webpage, Integrity Protection ,Fragile Watermarking

I. INTRODUCTION

Nowadays, with online shopping gets more popular, the integrity of web pages becomes seriously important for both users and content providers. Meanwhile, the result of experiment results shows that the "in-flight changes" of web page widely exist in HTTP [1]. This means some of the web pages are modified by a middle access point when they fly from a web server to client's browser. For example, the advertisements which were injected by Internet Service Providers (ISPs) to increase revenues; the pop-up were blocked by the firewall which would decrease potential revenue of the website; even more, the malware injection can be added to the original page.

Usually, to prevent such changes are using the HTTPS [2]. However, it also has disadvantages. First of all, the TLS will increase web servers and users' performance overheads. Secondly, for lots of portal websites, and web applications like YouTube, HTTPS seems it's too restrict since confidentiality is not so critical important. And the most important thing is that HTTPS provides the end-to-end security by symmetrically encrypting each document the user requested. So it does not support web document caching technology, which brings about that web contents only can be served by web servers but not any local proxy server. This breaks current web distributed architecture and increase bandwidth cost in the Internet.

Web Cache. There are two types of Web caches: a browser cache and a proxy cache. Here, we mean the proxy cache, which is near users and stores copies of web contents of host server passing through it, such as HTML pages, images, even web applications. When subsequent requests for these contents arrive, the cache will deliver the locally stored copy of the content to avoid repeating the download from the host server. With this technology, it cannot only efficiently reduce host server bandwidth and workload but also let users get the response quickly. With service oriented web application gets more popular, Web caching plays a more important role in improving service quality for a large range of Internet users.

Based on these reasons we believe there is a need to design a new protocol which focuses on providing the integrity of Web page and also support the web cache technology to improve the performances in comparison to HTTPS solution. This paper is organized as follows. Section 2 describes related works in this domain and Section 3 presents a new protocol. In section 4 we conduct a comparison to show the effectiveness of the proposed protocol and Section 5 concludes the paper.

II. RELATED WORKS

Web Tripwires. In [1], authors design a toolkit named "Web Tripwires" to provide the web page integrity of a web page using JavaScript code. The web server sends three parts to users: a web page, a JavaScript (the web tripwire) and a well known representation (e.g. Hash value of a webpage). This toolkit automatically computes a well known representation of web page and then compares it with the original received one to see whether there are any changes. The advantage of this solution is that it is flexible and less expensive than switching to HTTPS and do not require changes to current browsers. And it also allows the user to detect precisely which modifications have been done by the attacker. However, this approach has one major disadvantage. It's not security enough. If there are attackers have been noticed this technique, The JavaScript code can be also easily removed or modified with the web page together.

SINE protocol. In [3], authors develop a new protocol called “SINE” to protect the integrity of transferred web contents and also support Web cache. The main idea is based on a Hash-chain verification scheme for sinning digital streams [4].The Web server, firstly, divides the web contents into k equally-sized blocks. Secondly, starts to generate the Hash-chain from the bottom block to top block. Each block is attached the Hash of the next one which means that if assume that block, i is correct, then can verify the block i+1 through computing the Hash value of block i+1 and compare it with the one attached in block i. Finally, the Hash value of top block is signed by a digital signature to provide the integrity of the entire chain. When user gets this document, it can be verified from top to button one by one. However, even this method can grantee the integrity of the document and also support the web cache technology, it has a main disadvantage while in the critical network bandwidth environment, it will be costing to store the Hash value in the server and its transmission.

PCA-based web page watermarking. Principal Component Analysis (PCA) [5] is a multivariate technique, which can transform a number of related variables to other uncorrelated variables, which can represent the original characters before the transformation took place. PCA has two main properties: (1) the principal vectors (PVs) are projection axes of the original data and (2) the projected vectors can express the most features of the original data.

In [6], authors give a fragile watermarking scheme for web page’s tamper-Proof based on PCA algorithm. The main advantage of this scheme is that the watermarked web page can detect the web page’s change and do not increase web page’s size. However, this scheme does not concern the in-flight change but only pay attention to the Host-targeted malicious modifications. It cannot protect the integrity for in-flight web page. The most important drawback is that when the web page’s size becomes larger, the computing time is exponential growth.

III. PROPOSED PROTOCOL DESCRIPTION

The architecture of this new protocol is based on a Watermarking-chain where we employ PCA-based fragile web page watermarking scheme. By the way, we also make some improvements for the PCA scheme itself. We briefly explain this watermarking generation and embedded processes of this scheme as follows.

A. Watermarking generation and embed process

(1). Extract an integer matrix $H^{R\times C}$ from source codes of the original web page by mapping each English character from their corresponding ASCII code.

(2). Amplify the changes by computing $D=HH^t$
(t represents the transpose operation.)

(3). Use a random sequence as a key *K*, here assume it is an N×N matrix. Then use Shannon Diffusion to convolute the D matrix, $I=D \otimes K$

(4) Apply PCA algorithm to the matrix *I*, and choose the first R principal components to generate watermarks.

(a) Compute the covariance matrix *V* according to the equation:

$$V = \sum_{i=1}^{N} (I_i - \bar{I}_R)^t (I_i - \bar{I}_R)$$

I_i is the i th row vector in I, and $\bar{I}_R \in F^{1\times N}$ is the average vector of the row vectors in I, it is computed by

$$\bar{I}_R = \frac{1}{N}\sum_{i=1}^{N} I_i$$

(b) The Eigen decomposition (ED) was applied to *V*: $V=ULU^{-1}$ (U^{-1} is the inverse matrix of U. L is a diagonal matrix with Eigen value of V.)

(c) The subset of u_1, u_2... u_N as basis vectors of a feature space *S*: S=span (u_1, u_2… u_m), m≤N.

(d) Computed some feature vectors by projecting them into *S* using the following equation:

$$Z_i = (I_i - \bar{I}_R)[u_1 u_2 \dots u_m], i = 1,2,\dots,N$$

Where $Z_i \in F^{1\times m}$ is the coordinates of the original data in the feature space S. Z_i is can be called “the Principal Components.”

(e) Choose the first R principal components as Z_1, Z_2... $Z_R \in F^{1\times R}$ and converted them into binary form, for example: a sequence of ‘0’ and ‘1’. Let Z_{ij} denote the *j*th element of Z_i, i=1,2,…, R, j=1,2,…, R. Thus, each z_{ij} is converted to binary form β_{ij}, and all of them are connected to a binary sequence $Wi = \beta_{i1}\beta_{i2}\dots\beta_{iR}$, which was taken as the watermarking for the *i* th text line. So, the watermarking, W, of the whole web page is composed of all *Wi*’s, i=1, 2, R.

(5).Embedded watermarks to through modifying the case of letters in HTML tags (called Upper-Lower Coding, ULC) [6]. For example, in Fig1, W_i= (1001) and T_i=”<title>Web Page</title>”, after being watermarked, T_i is transformed into T_i^w=”<TitLE>Web Page </tiTLe>.”

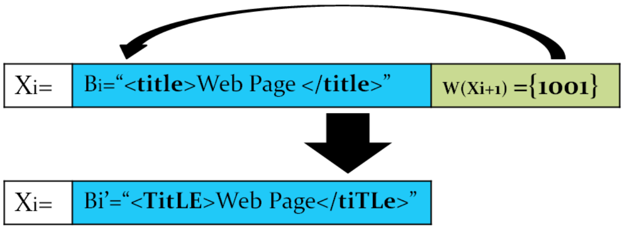

Fig.1 Watermarking embedded by Upper-Lower Coding

B. System Model

Firstly, we present the Client-Cache-Server model considered in our work. The system contains three parts: (a) The Web Server (b) The Proxy Cache Server (c) User. And the traffic relationship is shown in Figure 1.

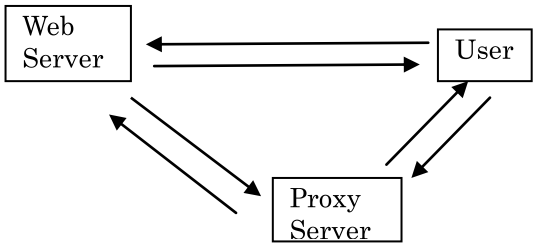

Fig.2 Client- Cache-Server system model

When a user sends a request to the web server base on HTTP, the proxy server will check whether the requested page is available within the cache itself. If the cache has it, the proxy server will response it directly, otherwise the proxy server forwards the request to the web server. Then the Web server gives a response back, the proxy will stores the web page within the cache for next time's user request. When there is a necessary, for example, security concern, the user also can establish a direct connection to the web server base on SSL.

C. Design Requirement

The protocol should satisfy 4 requirements as follows.

First, it should detect any unauthorized modifications to a web page and also should be able to locate the modification as precisely as possible.

Second, it should support web caching technique. The protocol should not change the web proxy architecture.

Third, it should not increase the size of web page for extra communication overhead.

Fourth, it should be security no matter there is intended attack or not. The attacker cannot bypass nor remove the mechanism.

D. Overview of New Protocol

On Fig 3, the authenticator for a page is shown. First, the server divides HTML source code of the web page into n blocks which have equally rows. (B_1, B_2...B_n). Secondly, start to generate the Watermarking-chain from down to top with the secret key *K*. We add padding in the bottom block Bn as an END marker to be able to detect the end of the Watermarking-chain in the next verifications process.

Then we add the watermarking of X_i in B_{i+1} and repeat this action until get to the top of the chain. By this way, we create a link between each block of the chain.

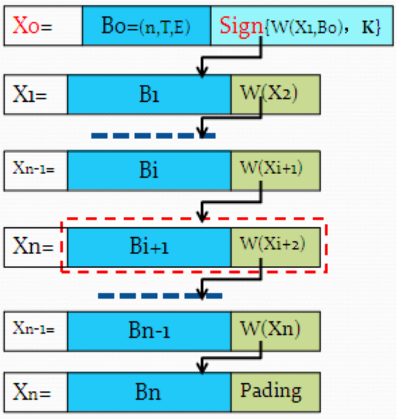

Fig.3 Authenticator

The first part X_0 which is computed as the server's digital signature over X_1, provides the integrity of the entire chain. (n: the number of blocks in file, T: timestamp, E: expiration date, K: secret key of watermarking)

The user can incrementally verify the integrity of each block of the web page by verifying the interdependent one way watermarking values, while the authenticity of the entire document is guaranteed by a digital signature on the top of Watermarking-chain.

E. Process

Here describe the process of the new protocol:

(1). The client sends a GET request to get a web page.

(2). The Web Server creates the Watermarking-chain of the requested web page with a secret key K, and then embeds the watermarking into HTML tag. This operation can be done in advance. After this operation, we get a new page which can detect changes.

(3). The Web Server sent the new page (requested web page and the signed Watermarking-chain) to user through Proxy Server. After the first request for a page, the proxy will store the page and the corresponding Watermarking-chain.

(4).The user gets the first block of the new page. First, extracts the watermarking and then verifies this block by using the public key of Web server.

(5).The authenticator X_i is used to authenticate the subsequent block X_{i+1}. Authentication of blocks Xi for i = 1...n is performed using fast fragile watermarking until the last block of X.

(6).By comparing the number of blocks received with the authenticated n from Y_0. The client verifies whether the entire document has received.

IV. EVALUATION

Since we have not implemented our protocol yet, because it is not feasible to implement the protocol without having an access to the remote web servers and proxy servers, we cannot really compute the detailed performances. Therefore, firstly, we evaluate the performance of the new protocol by comparing to existed approaches.

(1) Because we break the web page into small blocks (typically 1 KB), this will greatly reduce the distance between PCA watermarking with traditional Hash functions (e.g.MD5, SHA...).

(2) Because usually that most important message of the HTML codes is within its content parts and double quotes parts of its tags, we decide to extract only double-quotes parts in HTML tags. This will save a lot time when watermarking generation.

(3) We directly map the English character and numbers into their corresponding ASCII code but not redefine to "0~25" like original PCA watermarking scheme. This can save computing time and also could guarantee the number's integrity in the source code but not only the English characters.

Base on these analyses, we make a comparison about our proposed protocol with the existed approach as following table.

	HTTP	HTTPS	Tripwire	SINE	proposal
Detects changes	×	√	√	√	√
locate changes	×	×	√	√	√
Renders increme-tally	√	√	√	√	√
Web Cache	√	×	√	√	√
Not increase page size	√	√	×	×	√
Security	×	◎	×	○	○
Performa nce		×	◎	○	△

(×:not support √:support ◎: best ○:high level △:middle level)

V. Conclusion

A growing service such as online shopping should consider the integrity of web page to detect and prevent unauthorized in-flight modifications. In this paper, we presented a new security protocol, based on [3] that ensures the integrity of web page and supports web caching technology. And also by replacing the traditional Hash computing to the PCA-based fragile watermarking, the protocol has a main merit that it can save the channel bandwidth of Internet, which makes us believe that it has a significant advantage in the critical network bandwidth situation or environment than existed approaches.

REFERENCES

[1] .C. Reis, S. Gribble, T. Kohno, et al., Nicholas C. Weaver, "Detecting In-Flight Page Changes with Web Tripwires " The 5th USENIX Symposium on Networked Systems Design & Implementation (NDSI2008) Pp. 31–44 .

[2] Wikipedia "HTTP Secure" http://en.wikipedia.org/wiki/HTTP_Secure

[3] . C. Gaspard, E. Bertino, C.Nita-Rotaru, S.Goldberg, W.Itani, "SINE: Cache-Friendly Integrity for the Web." The 5th workshop on Secure Network Protocols (NPSec2009)

[4] R. Gennaro, P. Rohatgi, "How to Sign Digital Streams" Proceedings of the 17th Annual International Cryptology Conference on Advances in Cryptology, Pages: 180 - 197

[5] J.E. Jackson, "A User's Guide to Principal Components" Wiley, New York, 1991.

[6] Q.Zhao, H.Lu, "PCA-based web page watermarking" Pattern Recognition Volume 40, Issue 4 (April 2007) Pp.1334-1341