A Web Metering Scheme for Fair Advertisement Transactions

Ren-Chiun Wang[1], Wen-Shenq Juang[2] and Chin-Laung Lei[1,3]

Department of Electrical Engineering[1]
National Taiwan University
No. 1, Sec. 4, Roosevelt Rd., Taipei, Taiwan 106, R.O.C.
rcwang@fractal.ee.ntu.edu.tw, lei@cc.ee.ntu.edu.tw

Department of Information Management[2]
National Kaohsiung First University of Science and Technology
No. 2, Jhuoyue Rd., Nanzih District, Kaohsiung 811, Taiwan, R.O.C.
wsjuang@ccms.nkfust.edu.tw

Corresponding author[3]

Abstract

Since the rapid development of the Internet, many advertisers would want to introduce their goods on web sites. For achieving fair network advertisement payment, one of payment systems may need to evaluate the number of visited clients for particular web pages. However, advertisers fear that web servers inflate the number of metering. Also, web servers fear to receive a forged witness from a malicious client. If one of the above situations happened, the payment of network advertisement is unfair. In this paper, we propose a user-efficiency and fair web metering scheme for ubiquitous environments, where clients can use various intelligent devices to obtain their desired services at any time and any place.

1. Introduction

Since the rapid development of the Internet, more and more advertisers want to introduce their goods on popular web sites. A popular method to measure the popularity of a web site is to evaluate the number of visited clients during a certain time frame (a day or a month). For achieving a fair payment, several security threats must be prevented between advertisers and web servers. Advertisers want to prevent web servers from inflating the number of a metering. Web servers also fear to receive a forged witness from a malicious client.

In 1998, Naor and Pinkas [9] suggested metering schemes based on Shamir's secret sharing [12]. In their schemes, a web server can create a proof by collecting service requests of clients. Then the server can employ the proof to charge the advertisement fees from a trusted audit agency. However, Naor and Pinkas's schemes were insecure [10]. A web server can generate an incorrect proof from the cooperation of two malicious clients. Then the web server cannot make a claim to the advertisement fee. Later, several schemes were proposed based on the computational Diffie-Hellman assumption of the discrete logarithm and the bilinear pairings for enhancing the efficiency and being suitable to multi-server environments [7, 10]. In other words, in those schemes, the communication and computation cost of a client is heavy. Today, a client may use various intelligent devices to obtain his desired services at any time and any place. For convenience, most of these devices are small and of limited power and computation capacity. Therefore, an admired scheme should take these into consideration.

Also, several efficient web metering schemes were proposed based on a one-way hash function and simple bit exclusive OR operation [2, 4, 6]. In those schemes, a web server and a trusted audit agency have to pre-share an extra secret seed each other, the real identity of a client is sent over insecure networks and the server has to send a visited proof to an audit agency for each client. Finally, those schemes rely on a single server. It is not suitable to the real network environments, in which a client may visit many web sites for obtaining the desired services.

As the above description, a fair and efficient web metering scheme should provide the following requirements: (1) A web server cannot create an inflated visiting proof; (2) No one can send a forged witness to cheat a web server for obtaining the desired service; (3) Any valid and malicious client cannot make a web server from creating an incorrect visiting proof even if two or more clients collude; (4) The communication and computation cost of a client is low.

In this paper, we propose a user-efficiency web metering scheme. Our scheme not only provides the above requirements, but also does not require a security-sensitive verifier in the web server side. Hence, our scheme provides a fair network advertisement transaction.

The rest of this paper is organized as follows. In the next section, we propose a user-efficiency web metering scheme. In Section 3, we analyze the security of our scheme. In Section 4, we evaluate the performance of our scheme. Finally, we conclude this paper in last Section.

2. An Efficient Web Metering Scheme

There are two major techniques used in our scheme. One is the concept of hashing chain [3]; the other is the concept of Shamir's polynomial secret sharing scheme [12]. In our scheme, there are a trusted audit agency A, z clients (client i), and s web servers S_j, where $1 \le i \le z$ and $1 \le j \le s$. The scheme is used to meter m time frames and consists of four phases. Those are the initialization phase, the beginning of time frame phase, the interaction phase, and the end of time frame phase. We introduce them as follows.

The initialization phase

1. First, A selects a prime number q and two integer elements a and b, where $q > 2^{160}$ and $4a^3 + 27b^2 \bmod q \neq 0$. Then A also selects an elliptic curve equation over finite field q: $y^2 = x^3 + ax + b \bmod q$. Let G be a base point of the elliptic curve with a prime order n and O be a point of the elliptic curve at infinite, where n multiplies G is equal to O, and $n > 2^{160}$.
2. A issues a pseudo identity TID_i and $h(TID_i, \alpha_i)$ for Client i, where h() is a one-way hash function and α_i is a random number and is also kept secretly by A. Then A sends TID_i and $h(TID_i, \alpha_i)$ back to Client i through a secure channel.
3. A also calculates $h^m(h(TID_i, \alpha_i), S_j)$ and sends it with TID_i to the server S_j.

The beginning of time frame phase

Before the time frame t beginning, A selects a line $L_j(X) = cX + d \bmod n$ and publishes it publicly, where c and d are constants.

The interaction phase

At the time frame *t*, when Client *i* wants to get a desired service from a web server S_j, the following steps are performed.

1. Client *i* calculates $X_i = h^{m-t}(h(TID_i, \alpha_i), S_j)$ using the secret token $h(TID_i, \alpha_i)$. Then the client sends her/his service request $R = \{TID_i, X_i\}$ to the server S_j.
2. The server S_j first retrieves the verifier $h^m(h(TID_i, \alpha_i), S_j)$. If $h(X_i)$ is the same as the verifier, the identity of the client is authenticated and the server provides the service to the client *i*. Finally, the server updates a new verifier by using $h^{m-t}(h(TID_i, \alpha_i), S_j)$.

The end of time frame phase

During the time frame *t*, when a web server S_j has been visited by *k* or more clients, the server can calculate a visiting proof by using the Lagrange interpolation, where *k* is a threshold value which is pre-determined by *A* and the servers.

1. First, the server collects the service requests X_i of all clients.
2. Then the server calculates $L_j(X_i)$ using X_i.
3. Finally, the server can recover the secret polynomial $F_j(X)$ by calculating $\sum_{h=1}^{k} Y_{i_h} \prod_{l=1, l\neq h}^{k} \frac{x - TID_{i_l}}{TID_{i_h} - TID_{i_l}} \bmod n$, where $Y_i = L_j(X_i)$. The server sends $F_j(0)$ to the audit agency for charging the advertisement fee.
4. *A* also constructs a secret polynomial $F_j(X)$ with degree $(k - 1)$ for the server S_j, where $F_j(X) = \sum_{h=1}^{z} L_j(X_{i_h}) \prod_{l=1, l\neq h}^{z} \frac{x - TID_{i_l}}{TID_{i_h} - TID_{i_l}} \bmod n$, $X_i = h^{m-t}(h(TID_i, \alpha_i), S_j) \bmod n$. If the validation of the proof is true, *A* decides the amount of advertisement fee to the server S_j.

3. Security Considerations

Definition 1: A one-way hash function h() can produce a fixed-length digest value by taking an arbitrary size message. Also, it must satisfy the following properties:

1. It is easily to calculate a message's digest value $h(X)$, where *X* is an input message.
2. Given a digest value $h(X)$, it is computationally infeasible to find the message *X*.
3. Given a message *X*, it is computationally infeasible to find out another message *X'* to satisfy $h(X') = h(X)$.
4. It is computationally infeasible to find out any two different messages $X' \neq X$ to satisfy $h(X') = h(X)$.

Theorem 1: The scheme is secure against the server from inflating the number of the visited clients.

Proof. According to the properties 1, 2 and 3 of the one-way hash function, the following situations happened. First, the client easily calculates the service request $h^{m-t}(h(TID_i, \alpha_i), S_j)$

using the secret token $h(TID_i, \alpha_i)$ at the time frame t. Second, at the server side, it is computationally infeasible to derive $h^{m-t}(h(TID_i, \alpha_i), S_j)$ from the verifier $h^m(h(TID_i, \alpha_i), S_j)$. Third, at the server side, it is computationally infeasible to find out another digest value of a message X' from the received service request, where $h(X') = h^{m-t}(h(TID_i, \alpha_i), S_j)$. Therefore, the server cannot inflate the number of the metering in our scheme. □

Theorem 2: The scheme is secure against any clients from sending a confused service request for obtaining the desired service.

Proof. According to the properties 2 and 3 of the one-way hash function, the following situations happened. First, at a malicious and invalid client side, it is computationally infeasible to derive $h^{m-t}(h(TID_i, \alpha_i), S_j)$ without the secret token $h(TID_i, \alpha_i)$ in the time frame t. Also, it is computationally infeasible to derive $h^{m-t}(h(TID_i, \alpha_i), S_j)$ even if the adversary gets the verifier $h^m(h(TID_i, \alpha_i), S_j)$ from the server side. Second, at a valid malicious client side, it is computationally infeasible to find out another digest value of a message to be equal to $h^{m-t}(h(TID_i, \alpha_i), S_j)$ in the time frame t due to the properties of the one-way hash function. Therefore, our scheme is secure against the forged service request. □

Theorem 3: The scheme is secure against the generation of an incorrect proof from the server even if k or more clients co-operate.

Proof. According to the properties 3 and 4 of the one-way hash function and Theorem 2, the following situations happened. First, it is computationally infeasible to find out another digest value of a message to be equal to $h^{m-t}(h(TID_i, \alpha_i), S_j)$ in the time frame t from a valid client i. It means that the received service request must be sent from a valid client. Second, the server must identify if the digest value of the service request $h^{m-t}(h(TID_i, \alpha_i), S_j)$ is equal to the verifier in the time frame t. It means that no valid client can forge a digest value of a message to be equal to $h^{m-t}(h(TID_i, \alpha_i), S_j)$ even if k or more valid clients can derive the secret polynomial. Therefore, our scheme is secure against the cooperation of k or more valid malicious clients. □

4. Comparisons

For measuring the computation and communication complexities, we learn the equivalent security levels in key sizes among symmetric cryptosystem (SC), elliptic curve cryptosystem (ECC), Diffie-Hellman key agreement (DH), DSA, and RSA algorithms [5]: 80 (key size) in SC ≅ 160 (key size) in ECC ≅ 1024 (key size) in (DH/DSA/RSA); 128 (key size) in SC ≅ 283 (key size) in ECC ≅ 3072 (key size) in (DH/DSA/RSA); and 192 (key size) in SC ≅ 409 (key size) in ECC ≅ 7680 (key size) in (DH/DSA/RSA).

Standing the equivalent security levels in our comparisons, we assume that the key size of an elliptic curve cryptosystem is in a 163-bit finite field, the modular exponential operation is in a 1024-bits finite field and the operation of bilinear pairing is based on the elliptic curve cryptosystem. Then we also assume that the output length of a one-way hash function is 160-bit such as SHA-1 [1] and the length of a pseudo identity is 32-bit. Finally, some parameters are denoted as follows: T_H is the time of one hash function operation; T_{EXP} is the time of one modular exponential operation; T_{MUL} is the time for one modular multiplication; T_{ECM} is the time for the multiplication of a number over an elliptic curve, T_{ECADD} is the time for the addition of two numbers over an elliptic curve; $T_{\oplus}$ is the time of one exclusive OR operation and T_{BP} is the time for performing a bilinear pairing operation.

As introduced in [8, 11], we also learn a relationship as follows: $1T_{EXP} \cong 2T_{BP}$, $1T_{EXP} \cong 240T_{MUL}$, $1T_{EXP} \cong 600T_H$, $1T_{ECM} \cong 29T_{MUL}$, and $1T_{ECADD} \cong 5T_{MUL}$.

We compare the computation and communication cost with previous schemes [7, 10, 4, 6, 2]. We use Table 1 and Table 2 to evaluate the computation cost of a client at the interaction phase and the end phase. We also use Table 3 to show the comparisons of the communication cost at a client side among related schemes.

Table 1. The comparison of the computation cost of the interaction phase at time frame *t*

	The Interaction Phase	
	At client side	At server side
Our Scheme	$(m - t)T_H \cong 0.4(m - t)T_{MUL}$	$1T_H \cong 0.4T_{MUL}$
Lee and Lee's [7]	$2T_{BP} \cong 240T_{MUL}$	$1T_{BP} \cong 120T_{MUL}$
Ogata and Kurosawa's [10]	$2T_{EXP} \cong 480T_{MUL}$	$2T_{EXP} \cong 480T_{MUL}$
Kim *et al*.'s [4]	$1T_H \cong 0.4T_{MUL}$	$1T_{\oplus}$
Lee and Lee's [6]	$(m - t + 1)T_H \cong 0.4(m - t + 1)T_{MUL}$	$2T_H + 1T_{\oplus} \cong 0.8T_{MUL}$
Blundo and Cimato's [2]	$(m - t)T_H \cong 0.4(m - t)T_{MUL}$	$1T_H \cong 0.4T_{MUL}$

From the above, the communication and computation cost of a client in our scheme is very efficient. The scheme is also suitable for low power computing environments.

Table 2. The comparison of the computation cost of the end phase at time frame *t*

	The Interaction Phase
Our Scheme	$k(T_{ECM} + T_{ECADD}) + kT_{ECM} \cong 63kT_{MUL}$
Lee and Lee's [7]	$kT_{BP} \cong 120kT_{MUL}$
Ogata and Kurosawa's [10]	$kT_{EXP} \cong 240kT_{MUL}$
Kim *et al*.'s [4]*	$k(T_H + 1T_{\oplus}) \cong 0.4kT_{MUL}$
Lee and Lee's [6]*	$k((m - t + 1)T_H + 1T_{\oplus}) \cong 0.4k(m - t + 1)T_{MUL}$
Blundo and Cimato's [2]*	$k(m - \textit{visited times})T_H \cong 0.4k(m - \textit{visited times})T_{MUL}$

*: The web server has to send a charging proof for each client.

Table 3. The comparison of the communication cost at the client side

	The Communication Cost
Our Scheme	32bits + 2 * 160bits = 352bits
Lee and Lee's [7]	2 * 2 * 160bits = 640bits
Ogata and Kurosawa's [10]	2 * 1024bits = 2048bits
Kim *et al.*'s [4]	32bits + 160bits = 192bits
Lee and Lee's [6]	32bits + 2 * 160bits = 352bits
Blundo and Cimato's [2]	32bits + 160bits = 192bits

5. Conclusions

In this paper, we have proposed a user-efficiency web metering scheme. In our scheme, the web server cannot inflate the number of visited clients and malicious clients cannot forge a login request to cheat the web server even if they co-operate. Therefore, our scheme is fair for advertisement transactions.

6. References

[1] E. Biham, R. Chen, A. Joux, P. Carribault, W. Jalby, and C. Lemuet, "Collisions in SHA-0 and reduced SHA-1", Advance in Cryptology, EUROCRYPT'05, 36-57, 2005.

[2] C. Blundo and S. Cimato, "A software infrastructure for authenticated web metering", IEEE Computer, 37(4):28-33, 2004.

[3] L. Harn and H.-Y. Lin, "A non-repudiation metering scheme", IEEE Communications Letters, 5(12):486-487, 2001.

[4] S. S. Kim, J. Y. Shin and S. K. Kim, "Efficient metering scheme in the WWW", Proceedings 2001 Conf. on Info-tech and Info-net, 5:117-121, Beijing, 2001.

[5] K. Lauter, "The advantages of elliptic curve cryptography for wireless security", IEEE Wireless Communications, 11(1):62-67, Feb. 2004.

[6] N.-Y. Lee and M.-F. Lee, "Secure and efficient web metering scheme", IEE Proceedings Communications, 152(3):262-264, June 2005.

[7] N.-Y. Lee and M.-F. Lee, "Web metering scheme based on the bilinear pairings", IEICE Transactions on Information & Systems, E90-D(3):688-691, March 2007.

[8] Z. Li, J. Higgins, and M. Clement, "Performance of finite field arithmetic in an elliptic curve cryptosystem", 9th International Symposium in Modeling Analysis and Simulation of Comp. and Tel. Sys., 249-256, 2001.

[9] M. Naor and B. Pinkas, "Secure and efficient metering", Eurocrypt'98, 576-590, 1998.

[10] W. Ogata and K. Kurosawa, "Provably secure web metering", Asiacrypt'00, 388-398, 2000.

[11] B. Schneier, "Applied cryptography", 2nd edition, John Wiley & Sons Inc., 1996.

[12] A. Shamir, "How to share a secret", Communications of the ACM, 22:612-613, 1979.

Authors

Ren-Chiun Wang is currently working toward the Ph.D. degree in Department of Electrical Engineering from National Taiwan University, Taipei, Taiwan, R.O.C.. He has been a Lecturer on Department of Information Network Technology, Chihlee Institute of Technology, Taipei, Taiwan, R.O.C., since 2006. He is also a student member of the Institute of Electrical and Electronics Engineers. His current research interests include information security, cryptography, and mobile communications.

Wen-Shenq Juang received his master degree in computer science from National Chiao Tung University in 1993, and his Ph.D. degree in electrical engineering from National Taiwan University in 1998. He is currently an associate professor at the Department of Information Management, National Kaohsiung First University of Science and Technology, Kaohsiung, Taiwan. He is also the deputy secretary-general of Chinese Cryptology and Information Security Association since 2006. Dr. Juang's current research interests include ubiquitous applications, cryptography, information security, and electronic commerce.

Chin-Laung Lei received the B.S. degree in electrical engineering from the National Taiwan University, Taipei, Taiwan, R.O.C, in 1980, and the Ph.D. degree in computer science from the University of Texas, Austin, in 1986.

From 1986 to 1988, he was an Assistant Professor in the Computer and Information Science Department, The Ohio State University, Columbus. In 1988, he joined the faculty of the Department of Electrical Engineering, National Taiwan University, where he is now a Professor. His current research interests include computer and network security, cryptography, parallel and distributed processing, design and analysis of algorithms, and operating system design. Dr. Lei is a member of the Association for Computing Machinery.