Secure and Robust Metering in the Web Advertising

Soon-Seok Kim, *Member, KIMICS*

Abstract—In this paper, we present robust and secure metering scheme to measure the number of interactions between clients and servers in the web, especially the web advertising. In most cases the web advertising is consists of advertisers, clients, servers, and an audit agency. The metering scheme should always be secure against fraud attempts by servers which maliciously try to inate the number of their visits and against clients that attempt to disrupt the metering process. We propose robust and secure metering scheme based on cryptographic techniques. By analyzing the proposed scheme we show that our scheme is more robust and secure than the previous schemes [1,2,4,5].

Index Terms—Security, Web Advertising, Metering, Cryptographic one-way hash function, WWW

I. INTRODUCTION

On the WWW, advertisers need to know a way to measure the exposure of their advertisements in order to perceive the effectiveness of their advertisements. This measurement has a large affect on the cost needed for advertising. This affect can also be observed in traditional advertising such as newspapers and televisions. In such traditional forms of advertising, measurement is performed via statistical sampling of clients or certified audits of the popularity. Statistical sampling of clients is really applicable when there is a relatively limited selection that the user can make. However on the Internet this is not the case, where the user has millions of choices to select from. Therefore, statistical samplings of clients only have meaning when applied to the popular websites and for the rest it is meaningless. These popular and larger websites are usually not suspected of tampering with metered data.

The smaller sites, which have a considerable amount of advertisements, could do with a bit more. One of the main reasons that advertisers deter from smaller sites could be the lack of a secure and efficient metering scheme. A metering as such could increase and elaborate the overall advertising on the Web by providing advertisers with essential feedback. Being able to reach 100,000 people with a single site of interest is the advantage of the Web and could be enough to attract advertisers, provided there are reliable statistics. The schemes we present in this paper allow such metering.

Metering schemes for Web advertising measure the interaction between clients and Web servers (or web site publishers). Deciding the amount to be paid to Web servers hosting advertisements from advertisers (or their audit agency) can be useful. The fee paid to Web servers hosting advertisements should depend on the number of clients visiting the server. Therefore Web advertising requires measurements of the popularity of Web services and the measurement methods should be efficient so that clients, servers and audit agencies would be willing to apply it. Metering is also used in commercial advertising. For example, the interaction between a server and predefined clients (e.g. between a site with art information and artists) could be measured to decide royalty fees for copyright related material usage. A billing mechanism for usage based accounting between data networks can be an example application.

Automatic software modules, which are installed at Web servers to collect access information, are used to perform metering on the Web. However, as the server has control of information collection and data storage serious security flaws can occur. By manipulating the visit rates to higher values, the owner of the server could charge higher rates for advertisements, thereby earning more profits. This can be easily done by changing the unsecure metered data collected and stored on the server. Also, by using automated programs, individuals could generate automatic visits to particular sites to increase the visit rate.

Naor and Pinkas [1] introduced secure and efficient schemes in 1998. They considered metering in which there is an audit agency that measures the number of client visits received by web servers. In these schemes a server can produce a proof for the number of visits it has received only if it has received a predetermined number of visits by different clients. Later, Masucci and Stinson [2] introduced more efficient metering schemes. In these schemes a server was able to construct a proof for any number of clients visiting it. Also they provided lower bounds on the size of the information distributed to clients and servers, and on the number of random bits needed by the audit agency to set up a metering scheme. The two schemes above are based on a modified version of the *polynomial secret sharing* scheme of Shamir [3] in order to accurately measure the number of clients.

Manuscript received February 1, 2007.

Soon-Seok Kim is with the Department of Computer Engineering, Halla University, Wonju, Kang-won, 220-712, Korea (Tel: +82-33-760-1289, Fax: +82-2-33-760-1314, Email: sskim@halla.ac.kr)

In [4], we introduced efficient metering scheme that was cryptographically secure and prevent servers from inflating the count of their. Contrary to the previous schemes, our scheme was based on secure cryptographic *one-way hash function* and *bit-wise XOR* operations for metering. Moreover a server was able to construct a proof for *any* number of clients visiting it. So our scheme was more efficient than the previous ones in computational complexity.

In this paper, we propose more secure and *robust* metering scheme that improved the previous scheme [4]. In [4], if very few corrupt or erroneous clients 2 send incorrect information, server cannot verify it and can also generate a fault proof. In other words, if a client send server a fault value V' in [Step 2], server cannot compute correct P by computing $P' \leftarrow P \oplus V'$. In this case, metering system must stop next process and remove information from the client. So the server must verify whether the value V which client sends is correct (in general this feature is called *robustness*). For robustness, we propose a new advanced scheme which modified our previous scheme [4].

The rest of this paper is organized as follows. In Section 2 we review the previous schemes [1,4], and show our advanced scheme for robustness and analyze it in Section 3. Finally, in Section 4 we describe concluding remarks.

II. PREVIOUS WORKS

A. Naor and Pinkas [1]

Naor and Pinkas [1] proposed schemes for performing secure web metering using secret sharing schemes. Their approach has some advantages that provide computationally secure metering and preserve the existing communication pattern. Their scheme can be used to prove visits by k clients in any time frame, where α is a pre-determined system parameter, but cannot be easily extended to support finer grain visit counting or deal with multiple visits by a single client within a single time frame. Compared with their approach, our schemes are that server is able to construct a proof that depends on *any* number of clients visiting it.

In [1], metering measures the number of visits that a server receives (this is equally applied in our scheme). A visit can be defined according to the information that is of interest, e.g. it might be a page hit or any other relevant definition. It is beyond the scope of this paper to define what should be measured.

Here, we simply introduce their basic metering scheme, which is composed of following steps: initialization, regular operation and proof generation. Their scheme has also three parameters α, β and δ which is determined by the number of visits measured in a time-frame α and the security β and δ.

Initialization: The audit agency A generates a random bivariate polynomial $P(x,y)$ over a finite field Z_p, which is of degree α-1 in x and degree β-1 in y. It sends to each client C the univariate polynomial $W_C(y) = P(C,y)$, which is producted from P by substituting the value C for the variable x. That is, W_C is a restriction of $P(x,y)$ to the line $x = C$, and is of degree β-1. Here, the scheme will be used to measure α visits, and the parameter β defines the number of time frames in which the scheme can be securely used.

Regular operation: When client C visits a server S in time frame t, it sends to S the value $W_C(g)$. The input value g is a concatenation of S and t, and for simplicity they assumed that it is in Z_p and that no two pairs $\{S, t\}, \{S', t'\}$ are mapped to the same element.

Proof generation: After α clients have visited the server in time frame t it has α values, $\{P(C_i, g) \mid 1 \le i \le \alpha\}$, and can perform a *Lagrange interpolation* and calculate $P(0, g)$. This value is the proof that the server sends to the audit agency. The audit agency can verify the sent value by evaluating the polynomial P at the point $(0, g)$.

Compared with this scheme, our schemes are based on secure *cryptographic one-way hash function* and *bit-wise XOR operations* for metering. So our schemes are more efficient than the previous ones, requiring the higher computational overheads. In the next subsection, we will describe the previous basic scheme [4].

B. Our Previous Basic Scheme [4]

This scheme is composed of clients (denoted by C), servers (denoted by S), and audit agency (denoted by A). C and S do not necessarily trust each other, but they do trust A for the purpose of metering. The basic idea of our scheme is as follows. A generates and sends a secret information and a temporary user identity to every C. When C approaches S to see an advertisement, C sends the secret information and the temporary user identity to S. S collects them from C and produces a *certificate* and a *roll book* as a proof. Then A, upon receiving the proof, verifies the *certificate* using the *roll book*. The notations used in this, and our previous basic scheme [4], which is depicted in Fig. 1, are shown below.

A : the identity of the audit agency.
S : the identity of servers.
C : the real identity of clients.
$\boldsymbol{TID_C}$: a temporary identity of clients, this value is not a real identity of C and it should be different for each of n clients. A must exchange information about TID_C with S in *initialization step*. Supposing A and S exchanged the information, one-way hash function h and seed s, TID_C would be $h(s)$, $h(h(s))$, $h(h(h(s)))$, $h(h...h(s)...)$. Also these values are sorted by S and stored for *MASK* generation.
MASK : an n bit vector (initialization value is 0) which S generates. *MASK* is indexed with the sorted list generated previously. When C visits S it finds *TIDc* using binary search in the sorted list and decides *rank*(*TIDc*), where *rank*(*TIDc*) is the rank of *TIDc* in the sorted list. Hence it sets $MASK[rank(TIDc)] \leftarrow 1$. Consequently, *MASK* is a *roll book*.
$\boldsymbol{\alpha}$: a secret key which A randomly generates.
t : a certain time frame (e.g. a day or a month) which is predetermined by A and S in *initialization step*.

h : a one-way hash function (e.g. SHA-1 or HAVAL)
$\oplus$: a bit-wise exclusive-OR operation.

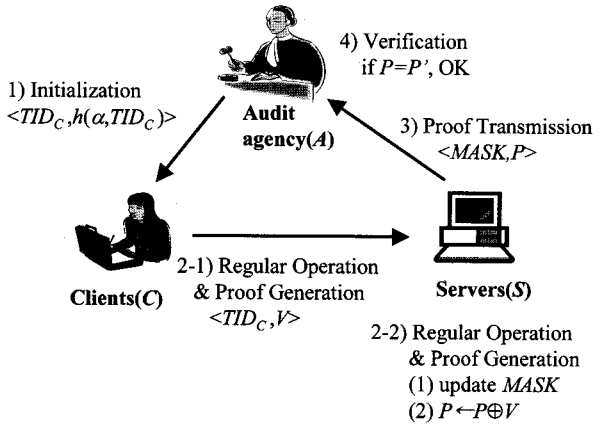

Fig. 1 Our Previous Basic Scheme [4]

[Step 1] Initialization:
The audit agency A chooses a random secret key α and a temporary identity TID_C. It then generates an initialization message $<TID_C, h(\alpha, TID_C)>$ for every client C. This message is sent to C through a secure channel and should be kept secret.

[Step 2] Regular Operation & Proof Generation:
When client C approaches server S in a certain time frame t, C computes V $(=h(S, t, h(\alpha, TID_C))$ and sends a message $<TID_C, V>$ to S. Then S executes the following:

(1) Updates $MASK$ using $TIDc$.
(2) Computes $P \leftarrow P \oplus V$, where P is a *certificate*, whose initialization value is 0.

[Step 3] Proof Transmission:
Server S sends $MASK$ and P at the end of time frame t.

[Step 4] Verification:
The audit agency A generates a certificate P'[1] using the $MASK$ from server S, and verifies whether P equals P'.

III. NEW METERING SCHEME FOR ROBUSTNESS

In the previous scheme [4] if very few corrupt or erroneous clients send incorrect information, server cannot verify it and can also generate a fault proof. In other words, if a client send server a fault value V' in [Step 2], server cannot compute correct P by computing $P' \leftarrow P \oplus V'$. In this case, metering system must stop next process and remove information from the client. So the server must verify whether the value V which client sends is correct. For robustness, we propose a new advanced scheme which slightly modified our basic scheme. The proposed scheme can be summarized as follows. In [Step 1] an audit agency A generates firstly his own secret information β and computes $\omega = h(\beta, V)$. Here V is a value $h(S', t', h(\alpha, TID_C))$ when a client visits a certain server S' in a predetermined time frame t' and it can be precomputed by audit agency. In the basic scheme, V is originally the value when a client visits an arbitrary server S in arbitrary time frame t. However if a client visits a certain server S' in a predetermined time frame t', it is more efficient in implementation of robustness because an audit agency can precompute the V. In this case the proposed scheme can be applied to a special purpose (e.g. in a membership web site such as a metering the number of members which visit the site in a predetermined time frame). However if an audit agency generates each server other value ω and gives each client, the clients may selectively visits an arbitrary servers in a certain time frame. However this scheme is less efficient than the former. Secondly an audit agency A sends β (to servers) and ω (to clients). In [Step 2] clients send servers the V concatenated with ω. In [Step 3] each server verifies whether V is correct by computing ω $(=h(\beta, V))$.

The scheme, which is depicted in Fig. 2, is described as follows. Here we assume that a client visits a predetermined server S in a certain time frame t, where a value V is precomputed by audit agency.

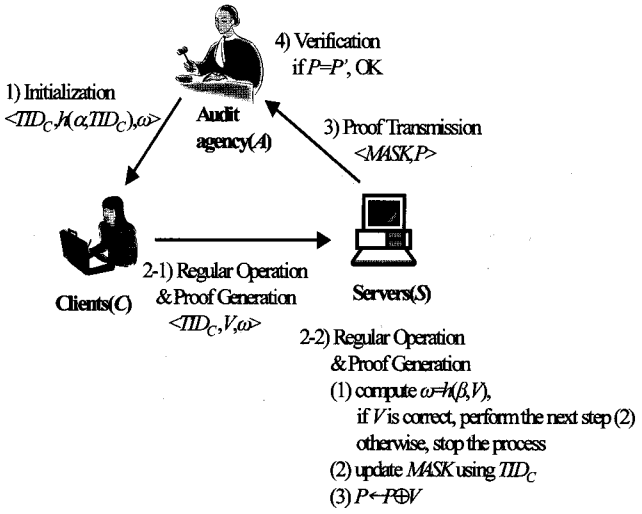

Fig. 2 New Metering Scheme for Robustness

[Step 1] Initialization:
The audit agency A chooses a random secret key α, β and a temporary identity TID_C, and sends β to every server S. A also computes $h(\alpha, TID_C)$, V $(=h(S, t, h(\alpha, TID_C)))$ and ω $(=h(\beta, V))$. It then generates an initialization message $<TID_C, h(\alpha, TID_C), \omega>$ for every client C. This message is sent to C through a secure channel and should be kept secret.

[Step 2] Regular Operation & Proof Generation:
When client C approaches a predetermined server S in a certain time frame t, it computes V $(=h(S, t, h(\alpha, TID_C))$ and sends a message $<TID_C, V, \omega>$ to S. Then S executes the following

(1) Computes ω $(=h(\beta, V))$ and verifies whether V is correct. If V is correct, then performs the next step (2)

[1] This value can be computed using the $MASK$ in the same way server S computed. In initialization step audit agency A can precompute the values V according to a certain time frame t.

otherwise, stops the process.[2]

(2) Updates *MASK* using *TIDc*.

(3) Computes $P \leftarrow P \oplus V$, where P is a *certificate*, whose initialization value is 0.

***[Step 3] Proof Transmission*:**

Server S sends *MASK* and P at the end of time frame t.

***[Step 4] Verification*:**

The audit agency A generates a certificate P' using the *MASK* from server S, and verifies whether P equals P'.

A. Analysis of the Proposed Scheme

***Robustness*:**

The robustness of proposed scheme is based on a secure one-way hash function. In our scheme, if a corrupt client fabricates the V, which is sent to server, he must find the secret key β, though he knows ω and V in $\omega=h(\beta, V)$. However it is computationally impossible for it to find β because of the one-wayness of hash function.

***Efficiency*:**

For the verification of robustness, Naor and Pinkas used the equation $V(x,y)=A(x,y)*P(x,y)+B(y)$, where $P(x,y)$ is the polynomial, which is of degree k-1 in x and of degree d-1 in y. $A(x,y)$, of degree c_k in x and c_d in y. And $B(y)$, of degree c_d in y. Here k is the number of clients who visited a server, d is the number of time frames, c_k and c_d is a security parameter in which the scheme can be securely used. The basic idea of their scheme is as follows. Suppose that the audit agency A asks clients C to communicate to S using a value u. A can choose random values a, b to authenticate the value u, compute $v=au+b \bmod p$, and send $<a,b>$ to S and $<u,v>$ to C. Afterwards C sends to S the pair $<u,v>$ and then S can verify that $v=au+b \bmod p$.

In the scheme of Naor and Pinkas, each server needs k computations of polynomial, which is degree of c_k+k-1, for k clients to generate a proof. Moreover each client must additionally perform the computation of polynomial, which is degree of c_d+d-1. However, Our scheme needs only k-times computation of hash function, $h(\beta,V)$. Therefore our scheme is more efficient (see Table 1).

***Security*: *Privacy*:**

The analyses of these features are the same as above the previous basic scheme [4].

IV. CONCLUSIONS

In this paper, we have introduced a new advanced scheme which modified our previous basic scheme [4]. In order to devise robust metering schemes, we applied some cryptographic primitives, such as one-way hash function and bit-wise XOR operations, and analyzed its robustness and efficiency which are essential for implementation of metering scheme.

[2] In this case, even if client C visited server S, we define as a no counting. However, in real application, S can request again correct V to C

Table 1. Comparison of Naor and Pinkas [1] and our proposed scheme (*: this notation means a computation in verification step by the audit agency).

		Naor & Pinkas [1]	Our Advanced Scheme for robustness
Security		Polynomial secret sharing scheme	One-way hash function
Basic idea		Using $V(x,y)=A(x,y)*P(x,y)+B(y)$	Using $\omega=h(\beta, V)$
Efficiency	Audit agency	Evaluate a polynomial of degree d-1 *	k-1 bit-wise XORs *
	Servers	Interpolate a polynomial of degree k-1, k polynomial of degree c_k+k-1	k-1 bit-wise XORs , k binary searches, k hash function
	Clients	Evaluate a polynomial of degree d-1, a polynomial of degree c_d+d-1	Compute one hash function

REFERENCES

[1] M. Naor and B. Pinkas, Secure and efficient metering, *Advances in Cryptology-EuroCrypt '98*, Lecture Notes in Computer Science, vol. 1403, pp. 576-590, May. 1998.

[2] B. Masucci and D. R. Stinson, Efficient metering schemes with pricing, *IEEE Trans. Information Theory*, vol. 47, no. 7, pp. 2835-2844, Nov. 2001.

[3] A. Shamir, How to share a secret, *Comm. ACM*, vol. 22, no. 11, pp. 612-613, Nov. 1979.

[4] Soon Seok Kim, Sung Kwon Kim, and Hong Jin Park, New approach for secure and efficient metering in the web advertising, *Computational Science and Its Applications-ICCSA 2004*, Lecture Notes in Computer Science, vol. 3043, pp. 215-221, May. 2004.

[5] M. K. Franklin and D. Malkhi, Auditable metering with lightweight security, *Financial Cryptography'97*, Lecture Notes in Computer Science, vol. 1318, pp. 151-160, Feb. 1997.

[6] V. Nikov, S. Nikova, B. Preneel and J. Vandewalle, Applying general access structure to metering schemes, *Proceedings of the International Workshop on Coding and Cryptography (WCC 2003)*, 2003.

Soon Seok Kim
Member KIMICS Received B. S. degree in Computer Engineering, Chinju National University, Korea, in 1997. M. S. and Ph. D. Degree in Computer Engineering, Chung-Ang University, Korea, 1999 and 2003. Since 2003, he has been Assistant Professor, Department of Computer Engineering, Halla University, Wonju, Korea. The areas of interest are information security, cryptography application, biometric authentication.