Computers and Mathematics with Applications 60 (2010) 272–275

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

journal homepage: www.elsevier.com/locate/camwa

On the security of metering scheme

Huang Lin *, Zhenfu Cao

Department of Computer Science and Engineering, Shanghai Jiao Tong University, 800 Dongchuan Road, Shanghai 200240, PR China

ARTICLE INFO

Keywords:
Metering scheme
Turnover
Hash function
Analysis
Weakness

ABSTRACT

In 2001, Harn and Lin [4] proposed a non-repudiation metering scheme. In this paper, we reveal two security weaknesses in their scheme, which could make the scheme either too inefficient or incapable of presenting the exact visiting number of a server. An improved scheme will be presented in this paper to avoid these weaknesses in the metering scheme.

1. Introduction

As web advertisement plays a more and more important role in the global net economy, attention has been drawn to find out a suitable way to measure the popularity of a server. Metering scheme is a simple way aiming to measure the visiting number of a certain server. A naive example of the metering scheme is to give each client a certified key and require it to present a certification to the server while visiting the server, but there are two main disadvantages of this naive scheme. First, it is far too inefficient, the size of a server's proof and the work of a third party to verify the claimed number of a server (could be an audit agency) should be the same order as the total visiting number of all the clients, plus it also implies that each client needs to perform a public key signature during every visit. Secondly, it could not prevent the invasion of privacy of the clients since it could be easy to trace the clients by verifying the certification of a certain client. In 1998, Naor and Pinkas [1] proposed a metering scheme based on secret sharing scheme for the first time. In 2000, Ogata and Kurosawa [2] proposed a modified version of Naor and Pinkas' metering scheme. In 2001, Masucci and Stinson [3] presented a flexible metering scheme with pricing. All of the schemes mentioned above were constructed based upon secret sharing scheme.

In 2001, Harn and Lin [4] first proposed a non-repudiation metering scheme based on Lamport's password-chaining technique [5]. Compared with the previous metering scheme, their scheme could work without the participation of the audit agency. The security of their scheme is based on the security of one-way hash chaining and the digital signature scheme.

However, we will state that with the help of a corrupt client a server could easily inflate its visiting number. If the third party wants to get the true visiting number of the server, it needs to verify all the signatures of the proof of the server which means there will not be any significant difference between Harn and Lin [4]'s scheme and the naive version of metering scheme.

2. Review of Harn and Lin's scheme

There are mainly two characters in their scheme: client and server. Let $f(x)$ be a hash function and $f^m(x) = f(f(\cdots(f(x))\cdots))$ be the composition of m f's, and $f^0(b) = b$. The whole scheme could be divided into the following steps:

– *Step* 1
During the initialization, each client selects an integer b and registers $f^m(b)$ to the server.

* Corresponding author.
E-mail addresses: Faustlin@sjtu.edu.cn (H. Lin), zfcao@cs.sjtu.edu.cn (Z. Cao).

0898-1221/$ – see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.camwa.2010.01.008

– *Step* 2
In the first visit, the client submits $f^{m-1}(b)$ to prove itself to the server. Then, the server will replace $f^m(b)$ in its database by $f^{m-1}(b)$.
– *Step* 3
In the next jth visit where $j = 2, 3, \ldots, m$, the client will present $f^{m-j}(b)$ to the server and the server will also update its database with $f^{m-j}(b)$. In order to provide non-repudiation proof to the server, each client also needs to sign on the hash value of $f^m(b)$ and present the signature to the server, and here M represents certain information about, such as the expired date and the identity of the associated server and so on. The server could use the combination of $f^{m-j}(b)$ and the signature of $(f^m(b), M)$ as evidence of jth visit made by the client.

3. Security flaws in Harn and Lin's scheme

For a dishonest server who wants to inflate its visiting number, it could accomplish the inflation by cooperating with a corrupt client. The dishonest server could execute the scheme as follow:

– *Step* 1
The client could first randomly pick n numbers $b_1, b_2, \ldots, b_n$ and register $f^m(b_1), f^m(b_2), \ldots, f^m(b_n)$ to the server. Besides, the client will present the corresponding signature on $(f^m(b_1), M)$, $(f^m(b_2), M), \ldots, (f^m(b_n), M)$ using n forged private keys corresponding to n different fake identities. The expired date in M could be as late as possible.
– *Step* 2
In the jth visit, the client submits $f^{m-j}(b_1), f^{m-j}(b_2), \ldots, f^{m-j}(b_n)$ to the server, where $j = 1, 2, \ldots, m$.
– *Step* 3
Let $Sig_k[c]$ represent the signature on c by identity k. By presenting the combination of $f^{m-j}(b_k)$ and the corresponding signature $Sig_k[(f^m(b_k), M)]$ (where $j = 1, 2, \ldots, m$, $k = 1, 2, \ldots, n$), the server could provide the proof of n visiting clients while only one client visits the server actually.

The server could easily inflate its visiting number by following these above steps. For a web advertiser who wants to know the exact visiting number, he will not be able to find out these n-pair $f^{m-j}(b_k)$ and $Sig_k[(f^m(b_k), M)]$ are actually from the same client unless he verifies the corresponding signature. However, it will be too inefficient if the web advertiser needs to verify the clients' signature one by one.

This scheme also cannot prevent another kind of inflation by the server. This kind of inflation is first mentioned in [1]. Corrupt servers or "entrepreneurs" could possibly organize a large group of clients and sell their services as "visitors-per-pay". Thus, a server could purchase these services and claim it has a large number of visitors while these visitors are from a single group of clients actually.

4. An improved metering scheme

There are two security flaws about Harn and Lin's scheme. The first one could be prevented by adding an audit agency to the scheme. The metering scheme presented in [6] could actually prevent the first inflation. However, the scheme [6] could not avoid the second kind of inflation either. The second kind of inflation could be prevented by the turnover property of a metering scheme [1]. Turnover means the ratio between the number of old and new visiting clients of a server. Since the claimed number of the server in the second kind of inflation is based on a single group of clients, it would not be easy for the dishonest server to provide a good turnover. An improved metering scheme will be presented in this paper, and this scheme is actually a modified version of the metering scheme in [6]. This new metering scheme will add the turnover property in order to prevent the second kind of inflation.

The whole metering scheme could be divided into two states: the normal state and the turnover state. In the normal state, the audit agency A could neglect the turnover of the server. A needs to consider the turnover of the server in the turnover state. The turnover state could last for several time frames randomly chosen by the audit agency A in order to prevent the server from nullifying the inspection of A.

At the beginning of the whole metering scheme, the audit agency A will assign a random key k_i and a user identity Uid_i for the corresponding client C_i, $i = 1, 2, \ldots, n$ and then computes $h(k_i, Uid_i)$ for C_i and transmits it to the corresponding client later. The audit agency is also responsible to send the n pairs of $\langle Uid_i, h^m(k_i, Uid_i)\rangle, i = 1, 2, \ldots, n$ to the server, where m represents that this metering scheme could last for m time frame. The normal state of the scheme works as follow:

– *Step* 1
When C_i visits the server S during time frame t, C_i needs to compute $h^{m-t}(k_i, Uid_i)$ and $V_i = h(S, t, h^{m-t}(k_i, Uid_i))$ and send them to the server. The server will verify the correctness of $h^{m-t}(k_i, Uid_i)$ by executing t times hash functions to check whether $h^t[h^{m-t}(k_i, Uid_i)]$ is equal to $h^m(k_i, Uid_i)$ stored in its database. If the answer is positive, then it will check whether $V_i = h(S, t, h^{m-t}(k_i, Uid_i))$ holds. If this equation also holds, then it will provide the required service for the client. If any of the answer to these previous two check is negative, then it will refuse to provide service for the client.

– *Step* 2
At the end of this time frame, it will present a proof to the audit agency. This proof includes the visiting number l and the visiting clients' identities Uid_i, $i = 1, 2, \ldots, l$. Meanwhile, the server needs to compute the corresponding $P = V_1 \bigoplus V_2 \bigoplus \cdots \bigoplus V_l$ and sends P to A.
– *Step* 3
When A starts to verify the claimed number of the server, it first needs to check the identities Uid_i, $i = 1, 2, \ldots, l$ and computes all the corresponding $V_i' = h(S, t, h^{m-t}(k_i, Uid_i))$, $i = 1, 2, \ldots, l$, and finally it will compute $P' = V_1' \bigoplus V_2' \bigoplus \cdots \bigoplus V_l'$. If the equation $P' = P$ holds, then the audit agency will decide the amount of money it needs to pay for the server.

The turnover state of the scheme works as follow:

– *Step* 1
The principle of the first three steps could be the same to that of the counterpart in the normal state.
– *Step* 2
In order to estimate the turnover of a certain sever, A needs to store the identities sets of the server in the different time frames of turnover state. It is A's responsibility to assign the clients ID, thus A knows the correspondence between the client and the client's ID. In consequence, A could easily compute the ratio between the number of new clients and that of the old clients, i.e., the turnover of the server. Since the audit agency might supervise many servers during a certain period, it is easy for A to establish a criteria in term of the turnover. Those servers with the turnover below a certain level should be paid a lower unit price per visiting client as a punishment. Those servers with the turnover above the level should be paid a higher unit price per visiting client as a rewards. Plus, the advertisers might consider those servers with low turnover as the candidates when they decide to terminate their pacts with certain unprofitable servers to ensure a high profits from the advertisements.

5. Security and efficiency analysis

The security analysis could be twofold according to the two kind of inflations.

1. It is impossible for a single client to impersonate multiple clients since the issue authority of the clients' identities belongs to A. Without the help of the audit agency, the one-way chaining algorithm can prevent one client to figure out the secret key K_i of the other clients and thus successfully impersonate the other clients since it is computationally infeasible for these clients to compute backward values from the published one-way value.
2. It is hard for the server to benefit from the second kind of inflation due to the rewards and punishment policy based on the turnover properties of the metering scheme. The server might still win certain amounts of profits from the advertisers while only a single group of clients visit the server. However, the audit agency will easily find out the trick of the server by observing the turnover of this server. Eventually the advertisers will cancel the contract with this server.

The efficiency of our construction is comparable to that of Harn and Lin's construction.

In our construction, when a client visits a server, it needs to accomplish two hash function computation while it needs to complete one hash function and generate one public key signature in Harn and Lin's scheme. The generation of a public key signature is rather costly compared with the symmetric hash function computation. However, the client could generate the signature in advance and store it in its own computer if the client visits the same server repeatedly. In order to make sure that the client presents the correctly formed proof, the server has to complete $t + 1$ hash function computations while this is not needed in Harn and Lin's scheme. However, there's no guarantee of robustness in Harn and Lin's construction respectively.

At the end of a time frame, the server in our construction is required to complete l XOR computations while in Harn and Lin's scheme there's no need for the server to do any computation.

When the audit agency A verifies the proof presented by the server, it needs to complete $l \times (m - t)$ hash function computations and l XOR computations while A has to accomplish $l \times t$ hash function computations and l public key signature verifications in Harn and Lin's scheme. It is observable that our construction is more efficient in this step.

6. Conclusion

In this paper, the security flaws of [4] are stated out, and an improved metering scheme is presented to remedy these flaws. The security analysis shows that this improved scheme does have the claimed advantage.

Acknowledgement

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant Nos. 60972034, 60970110 and 60773086.

References

[1] M. Naor, B. Pinkas, Secure and efficient metering, Lecture Notes in Computer Science 1403 (1998) 576–589.
[2] W. Ogata, K. Kurosawa, Provably secure metering system, in: Proc. Advances in Cryptology—ASIACRYPT 2000, pp. 388–398.
[3] B. Masucci, D.R. Stinson, Efficient metering schemes with pricing, IEEE Transactions on Information Theory 47 (7) (2001) 2835–2844.
[4] L. Harn, H.Y. Lin, A non-repudiation metering scheme, IEEE Communications Letters 5 (12) (2001) 486–487.
[5] L. Lamport, Password authentication with insecure communication, Communications of the ACM 24 (11) (1981) 770–772.
[6] N.Y. Lee, M.F. Lee, Secure and efficient Web metering scheme, IEE Proceedings. Communications 152 (3) (2005) 262–264.