

A Simple One-time Limited Authorization Mechanism based on ECDSA

Li Hui-na, Ping Yuan

Department of Computer Science and Technology, Xuchang University, Xuchang, China

leehuina@126.com, pyuan.lhn@gmail.com

Abstract— An one-time limited authorization mechanism based on ECDSA is proposed, which allows password-owner to grant his right to a temporary user without releasing any information about the original password and restrict the time slice or frequency easily. Based on the modification of ECDSA, the mechanism is feasible for being put into practice, can generate limited authorization password safely without being forged or re-used effectively.

Keywords— ECDSA; limited authorization password; access control; nonforgeability

I. INTRODUCTION

Administrator's password is set up in the computer systems in order to protect private information from being fetched by anonymous, which response for the relationship between the behavior of access and resource objects directly. Sometimes we need to transfer our passwords to another one for special purposes, such as we are requiring materials with no ability to login into the remote systems. After an analysis of the risks in two scenarios that the password should be transferred to others, a limited authorization password mechanism [1] was proposed to designate authorized users to visit the system resources. Depending on a limited authorization password, the mechanism should allow a temporary authorized users to visit system in time slice t using password AP_t effectively. However, the restriction is that the temporary authorized users' account, the frequency of visits and time slice requirement should be prearranged.

In this article, based on the Elliptic Curve Digital Signature Algorithm theory[2,3], an improved and controllable mechanism is being proposed.

First, a flexible variable t was introduced as a docket of time slice in which the authorized user would remain in the target system. Then the digital signature of ECDSA was modified from two tuple to three tuple. Obviously, the signature is transformed from the password-owner's password in a special way.

Second, the remote system will use the authentication algorithm to verify the data and grant the proper right to the visitor without releasing any information about the original password.

Two practical characteristics and advantages of the one-time limited authorization password mechanism are listed as

follows:

(1) The time slice and frequency are limited by the password-owner. After login into the system, the temporary authorized user must complete his visit in time; otherwise he must apply to the password-owner for another authorization.

(2) Under the mechanism, the limited authorization password generated by the original password will not disclose the original system password. Also, anyone who want to forge or reuse that authorization will be fund immediately.

II. REVIEW OF LIMITED AUTHORIZATION PASSWORD MECHANISM

The limited authorization password mechanism proposed by Zhan Tao[1] can be described as follows:

If the system password-owner named Alice need to grant right to a temporary user Bob, he should generate a limited authorization password TP and transfer it to Bob. They can achieve that purpose by following the process as follows:

(1) Alice defines the attribute set A to reflect the temporary authorized user's ability in system S . A is defined as $\{T, C, P_1\} (P_1 \subseteq P)$. T is the effective time of TP , C is the upper limit of access to the system S , P_1 is the subset of access permission P owned by Alice.

(2) Alice uses the following formula to calculate the limited authorization password TP .

$$TP = f(A) + Str(h(Str(A)||P)) \quad (1)$$

In the formula, A is mapped to a specific length of identifier ID by function $f(x)$ for storage. For $Str(x)$ is a single-shot function, it has a reverse function described as $Str^{-1}(x)$ to make the establishment of $Str^{-1}(Str(x)) = x$, and h is a hash function.

(3) Alice sends the TP to Bob. After Bob submitting TP , the system S gets A in a special way using $f(A)$ in TP . Then, the system S will calculate TP' as a copy of TP and allows Bob to access the limited resources only if the $TP' = TP$.

III. THE RISKS OF LIMITED AUTHORIZATION PASSWORD MECHANISM

Although easy to implement, the scheme require the limited authorized password TP and attribute set A be arranged and stored in system S before authorizing Bob. According to the definition of A , the mechanisms there are at least two types of

risks as follows:

(1) If the effective time T is a range, then Bob's visit to S will be strictly limited not at other times to visit; Therefore there is no value to add a variable C .

(2) If the effective time T is set to limit the length of access any time, then another problem will exist. Perhaps there is not sufficient number of definitions about A and Bob need more time than T to complete the visit, Alice must set up a new visit attribute set in system S without login into S . Obviously, Alice can do nothing to help him achieving the purpose.

To avoid this problem, Alice must set up a sufficient number of visit attribute set like A in system S and carry the corresponding unique identifier (ID) for the calculation of the new TP later.

Through the above analysis we can see that the password mechanisms require the password holders know the access time and the frequency to the future needs correctly. Clearly, such a requirement in practical applications is very difficult to achieve.

IV. ONE-TIME LIMITED AUTHORIZATION PASSWORD MECHANISM

A. Tuple of Authorization Password Generation

We will introduce the one-time limited authorization password mechanism directly. Initially, the curve parameters (q, FR, a, b, P, n, h) with private key PR_s and public key $PU_s = PR_s \cdot P$ should be generated as in ref [3]. In that parameters group, n is a big primes, and h is a hash function such as $SHA-1$.

If the password-owner Alice need to generate a limited authorization password AP_t for a temporary authorized user Bob. In our scheme, the AP_t is a tuple with three items. A permission mask M used by Bob for mapping a subset of system resources. In order to ensure that Bob has only one chance to login into the system S in the period of t to complete his visit with the AP_t , Alice should follow the steps to complete the initialization and distribute AP_t to Bob:

1) Alice select two random or pseudorandom relative primes $k, t \in [1, n-1]$ respectively and each one would be used only once. For our purpose, we use t to denote a special time slice initialized for Bob to access the target system S . Let the two integers satisfy $t \neq k^{-1} \pmod{n}$.

2) Compute $(tk \pmod{n})P = (x_1, y_1)$ and convert x_1 to an integer $\bar{x}_1$.

3) Compute $r = \bar{x}_1 \pmod{n}$. If $r = 0$ then go to step (1). Compute $k^{-1} \pmod{n}$ and $t^{-1} \pmod{n}$. Then, Alice use the hash function h to generate $h(M)$.

4) Compute $s = t^{-1} \cdot k^{-1}(e + PR_s \cdot r) \pmod{n}$. If $s = 0$ then go to step (1).

5) After that, Alice complete the generation of authorization password $AP_t = (t, r, s)$. Then the tuple of AP_t should be transferred to Bob in a secure way that would not be considered here.

B. Verification and Authorization Phase

We suppose that Bob have receive an authorization password $AP_t' = (t', r', s')$. After having got the input from Bob, the target system S startup the verification and authorization process immediately. This phase must agree upon the follows:

1) System S verify that r and s are integers in the interval $[1, n-1]$.

2) Fetch the permission mask M and compute $e = H(M)$.

3) Then, S compute u_1, u_2 and X as the following formulas:

$$w = (s')^{-1} \pmod{n} \quad (2)$$

$$u_1 = ew \pmod{n} \quad (3)$$

$$u_2 = r'w \pmod{n} \quad (4)$$

$$X = (t')^{-1}[u_1P + u_2 \cdot PU_s] \quad (5)$$

4) If $X = O$, then system S reject AP_t' . Otherwise, convert the x -coordinate x_1 of X to an integer $\bar{x}_1$, and compute $v = \bar{x}_1 \pmod{n}$.

5) System S will accept the requirement from Bob only if $v = r'$ and ensures that Bob would complete his visit in time t and log out on schedule. If necessary, Bob should be re-authorized with a new authorization password from Alice.

Based on the above process, Alice can grant his right to a temporary user without releasing the original password and restrict the time slice and frequency intensely. This one-time limited authorization mechanism can avoid the problems of the limited authorization mechanism effectively.

V. ANALYSIS

A. Correctness

As applied in the algorithm of ECDSA, the correctness of the verification process can be proved as follows: If the tuple AP_t' was indeed generated by Alice, then $s' = t^{-1}k^{-1}(h(M) + PR_s \cdot r') \pmod{n}$. Then the formula (6) will be true.

$$\begin{aligned} k &\equiv (t')^{-1}(s')^{-1}(h(M) + PR_s \cdot r') \\ &\equiv (t')^{-1}(u_1 + u_2 \cdot PR_s) \pmod{n} \end{aligned} \quad (6)$$

Then X from formula (5) will be $X = (t')^{-1}[u_1P + u_2 \cdot PU_s] = (t')^{-1}(u_1 + u_2PR_s)P = kP$. That means the equation $v = r'$ satisfy the requirement of verification.

B. Security

This paper presents a one-time limited authorization password mechanism based on the ECDSA which is based on the principle of elliptic curve. It has a stronger safety performance than DSA algorithm so that it had been used as the digital signature algorithm for commercial purpose and others[4].

The following analysis set focus on the influences security of the algorithm with a variable t . The common attacks of our proposed are listed as following and we will analyze it briefly.

Theorem 1. As part of public parameters, t, P and r would never disclose any information about the random integers k and private key PR_s more than the original algorithm ECDSA gives.

Proof: In order to get the integer k , one should know tk first or try to solve the problem listed by formula (7). From the

formula $(tk \bmod n)P = (x_1, y_1)$, as the same as the relative of key pairs PR_s and PU_s , the problem of working out tk is based on that finding the discrete logarithm of an elliptic curve element is infeasible[5].

$$\left(kP = (t^{-1} \bmod n) \cdot (x_1, y_1) \right)_P \not\Rightarrow k \quad (7)$$

Theorem 2. It is infeasible for attackers to attack the mechanism by setting $t'' = (ln)t$ where $(l \geq 1)$ to increase the time slice of visit.

Proof: Although attackers can make all formula true in the whole authorization process, the system S would reject any time slice that is greater than a special threshold T or satisfy a simple condition easily.

Theorem 3. It is impossible for attackers attempt to store and replay the tuples of (t, r, s) or use them with the public parameters (q, FR, a, b, P, n, h) and PU_s to construct arbitrary tuples or private key PR_s .

Proof: As being proved in Theorem 1, the intention of working out the private key PR_s is unrealistic. Anyway, the random number k will be changed each time, so that attackers have no the abilities to solve the open problem in the field of ECC for a long time.

VI. CONCLUSIONS

This work mainly investigates a kind of mechanism for one to grant his right to a temporary user with one-time

limited. Under this mechanism, the temporary user will get a authorization password AP_t without any information about the original password to access a target system in time slice t . Based on the modification of ECDSA, the mechanism is feasible for being put into practice, can generate limited authorization password safely without being forged or re-used effectively. In addition to this work, we will set focus on finding a practical scheme to restrict the authority in details to support this mechanism. So that the administrator can not only restrict the time slice but also the details of behavior and the range of objects easily.

ACKNOWLEDGMENT

This work is supported by Foundation of He'nan Educational Committee (0624220058).

REFERENCES

- [1] Zhan Tao, Zhou Xing-she, and Liao Zhi-gang, "Limited Authorization Password Mechanism", Computer Engineering, vol.34, pp.28-30, Feb.2008.
- [2] N. Smart, The discrete logarithm problem on elliptic curves of trace one, Journal of Cryptology, Volume 12, 1999: 193-196.
- [3] Darrel Hankerson, Alfred Menezes, and Scott Vanstone. Guide to Elliptic Curve Cryptography. Springer-Verlag, 2005.
- [4] Douglas R. Stinson, Cryptography Theory and Practice, Third Edition, Chapman & Hall CRC, 2006.
- [5] Ian F. Blake, Gadiel Seroussi, and Nigel P. Smart, editors, Advances in Elliptic Curve Cryptography, London Mathematical Society Lecture Note Series 317, Cambridge University Press, 2005.