(рабочих, дополнительных и для шифрования хэш-значения) на каждом из 3-х этапов формирования ЭЦП.

В программной реализации для шифрования хэш-значения применен описанный выше нетрадиционный метод [4]. Тогда для этой модели формирования ЭЦП необходим ее полный ключ, состоящий из следующих ключевых гамм каждого этапа:

- системы рабочих оснований $p_1(x), p_2(x),..., p_S(x)$ степени не выше N и порядка их расположения;
- системы избыточных оснований $p_{S+1}(x), p_{S+2}(x),..., p_{S+U}(x)$ степени не выше N_1 и порядка их расположения;
- ключевой последовательности псевдослучайных чисел $G_1(x) = \eta_1(x), \eta_2(x),..., \eta_W(x)$ и обратной к ней гаммы $G_1^{-1}(x) = \eta_1^{-1}(x), \eta_2^{-1}(x),..., \eta_W^{-1}(x)$, системы оснований $r_1(x), r_2(x),..., r_W(x)$ для шифрования хэш-значения с учетом порядка их следования.

При подписывании сообщения ключи каждого этапа алгоритма формирования ЭЦП выбираются из созданной базы полных ключей для шифрования сообщения. Номер записи ключа в базе данных выбирается так же, как при шифровании.

Компьютерная программа генерации и хранения ключей в базе данных является основой разработки комплекса программ по криптографической защите информации при ее хранении и передаче.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Акушский И.Я., Юдицкий Д.И.* Машинная арифметика в остаточных классах. – М.: Советское радио, 1968. – 439 с.

2. *Амербаев В. М., Бияшев Р. Г., Нысанбаева С. Е.* Применение непозиционных систем счисления при криптографической защите информации // Известия Национальной академии наук Республики Казахстан. Сер. физ.-мат. наук. – 2005.. – № 3. – С. 84–89.

3. *Бияшев Р.Г., Нысанбаева С.Е.* Влияние состава полиномиальных оснований непозиционной системы счисления на надежность шифрования // Материалы VIII Международной научно-практической конференции «Информационная безопасность», – Таганрог, Изд-во ТРТУ, 3-7 июля 2006, – С. 66–69.

4. *Нысанбаев Р.К.* Криптографический метод на основе полиномиальных оснований // Вестник Министерства науки и высшего образования и Национальной академии наук Республики Казахстан – Алматы: Гылым, 1999. - №5. – С. 63–65

5. *Бияшев Р.Г.* Разработка и исследование методов сквозного повышения достоверности в системах обмена данными распределенных АСУ: Дис... на соискание уч. степ. докт. тех. наук. – М., 1985. – 328 с.

6. *Моисил Гр. К.* Алгебраическая теория дискретных автоматических устройств. – М.: Издательство иностранной литературы, 1963. – 680 с.

A. G. Chefranov

Russia , Taganrog, Taganrog Institute of Technology, Southern Federal University, and North Cyprus, Gazimagusa, Eastern Mediterranean University

ONE-TIME PASSWORD SCHEME WITH INFINITE AUTHENTICATIONS NUMBER

Authentication of clients to servers is an important problem that has been solved in a number of ways (see, for example, [1, 2]). One time password (OTP) schemes such as [3, 4, 5] address the problem in assumption of not secure channels of communication between clients and servers, and possible compromise of passwords on the server side.

Also, approach [5] is robust to possible channel failure. These OTP schemes use hash chains $h(h(..h(p)..)) = h^n(p)$ applied to a secret password phrase *p*, where *n* is length of the hash chain. The hash function *h* is assumed to be a one-way function [1] that is easy to calculate when given an argument, and is computationally infeasible to invert, i.e. find its argument when given its value. The hash chains have a property that it is easy to go forward, but it is not possible to go backward not having secret information. Authentication in OTP schemes is based on providing a server with the server-side password, $h^n(p)$, where n, for example, 1000. A client is authenticated to the server by sending decrementing powers of $h(p)$ ($h^{n-1}(p)$ in the first session), which the server uses an authenticator in the current session (by checking that $h(authenticator) = server-side-password$), and, as a server-side password, in the next authentication session. OTP schemes have two major problems: 1) significant computational complexity imposed by the protocols on a client side, and 2) necessity of re-initialization of a password phrase after some certain number of authentications. Some measures reducing client side computational efforts are proposed in [3]. Restricted hash chains (RHC) used in [3], contrary to used in [5] chains, restrict also going in the forward direction to certain number of steps, if secret information is not available. However, mentioned above problems are not resolved radically. An OTP scheme using asymmetric encryption for hash function calculation and providing not limited number of uses before re-initialization is suggested in [6]. It is subject to compromise of the client's public key which is to be kept on the server side; the server is not authenticated to the client. Work [7] uses generated by both the client and the server hash chains as session keys and exchange by encrypted messages to authenticate each other; users' passwords are kept by the server RSA-encrypted. The problem with this scheme is in the knowledge of users' passwords by the server: they can be compromised if the server's database is compromised; long enough hash chains need to be calculated, RSA and DES algorithms are used in its implementation that is time consuming. Work [8] proposes an OTP scheme with mutual authentication of a client and server; it does not use encryption and is resistant to small number attack. Scheme [8] requires re-initialization after limited number of authentications and uses long hash chains. Work [9] shows that the scheme [8] is vulnerable to preplay attack. We introduce RHC allowing stepping forward only by k steps if secret information is not available (RHC-k), that is a generalization of RHC [3], and present a one-time password scheme based on RHC-1. The scheme improves [8] in the following: number of authentications before re-initialization is not limited, it has better performance, and it is not vulnerable to preplay attack. The rest of the paper is organized as follows. We present OTP scheme [8]. Then we introduce and analyze the proposed scheme; RHC-k is introduced, also. Then conclusions are presented.

We follow [9] in description of [8].

1. Registration stage

1.1. Client <- Server: *SEED*

1.2. Client <- Server: *N*, $SEED \oplus D_0$, $h(D_0)$

1.3. Client -> Server: $p_0 \oplus D_0$

where $p_0 = h^N(K \oplus SEED)$, *K* is a secret password phrase, $\oplus$ is a bitwise XOR operation, *N* is the number of login times before re-initialization, $N >> 1$, $SEED$ is a random number generated by Server, D_0 is the first nonce.

2. Login and authentication stage

2.1. Client <- Server: M1=(M_{11}, M_{12}, M_{13}) = ($N-t, SEED \oplus D_t, h(D_t) \oplus p_{t-1}$), $0 < t < N$ - integer.

2.2. Client: if $h(M_{12} \oplus SEED) \oplus p_{N-M_{11}-1} \neq M_{13}$ then Client decides that message M1 is not from Server and stops the procedure.

2.3. Client -> Server: $U_t = p_t \oplus D_t$

where $p_t = h^{N-t}(K \oplus SEED)$ is calculated by Client new one-time password that should replace the previous one, p_{t-1}, that was previously delivered to Server.

2.4. Server: if $h(U_t \oplus D_t) = p_{t-1}$ then Server authenticates Client, where p_{t-1} is one-time password that is already kept by Server. Finally, new password, $p_t = U_t \oplus D_t$, replaces p_{t-1} on the server side, and *t* is incremented.

This scheme is vulnerable to preplay attack [9].

In the protocol below we denote Client as C, and Server as S.

The Protocol

1. Registration stage

1.1. Server: $n_S = 0$; generate randomly $SEED_S$

1.2. Client <- Server: $SEED_S$

1.3. Client: $n_C = 0$; generate randomly $SEED_C$

1.4. Client -> Server: $SEED_C$

1.5. Client: $p = h(K \oplus SEED_S)$, $p_1 = h(p)$, $hp = h(p_1)$

1.6. Client -> Server: *hp*

2. Login and authentication stage

2.1. Client: generate nonce D_C

2.2. Client -> Server: M1=($M_{11}, M_{12}, M_{13}, M_{14}$) =

($p_1 \oplus h(D_C)$, $h(p \oplus h(p_1)) \oplus h(D_C)$,

$h^2(p \oplus h(p_1)) \oplus D_C$, $SEED_C \oplus D_C$

)

2.3. Server: $D_C = SEED_C \oplus M_{14}$; if $h(M_{11} \oplus h(D_C)) = hp$ and $h(M_{12} \oplus h(D_C)) = M_{13} \oplus D_C$, then Server authenticates Client, else – Server stops the protocol

2.4. Server: $V = D_C + 1$, $hp = M_{12} \oplus h(D_C)$, $n_S = n_S + 1$

2.5. Client <- Server: M2=($V \oplus SEED_S$, $V \oplus SEED_C$) = (M_{21}, M_{22})

2.6. Client: if $M_{21} \oplus SEED_S = M_{22} \oplus SEED_C = D_C + 1$ then Client decides that authentication is completed successfully, otherwise the protocol stops

2.7. Client: $p_1 = p \oplus h(p_1)$, $n_C = n_C + 1$

In the protocol, Client randomly generates nonce D_C, and hash of a new password of the client $h(p \oplus h(p_1))$ = *h(new password)*, and sends to Server message M1.

Server, upon receiving M1, reveals $D_C = SEED_C \oplus M_{14}$, $p_1=M_{11} \oplus h(D_C)$, and *h(new password)=*$M_{12} \oplus h(D_C)$, calculates $h(p_1)$, and compares it against *hp* kept in its database. Authenticity of M1 is checked by comparison of $h(M_{12} \oplus h(D_C))$ against $M_{13} \oplus D_C$. If the conditions hold, Server decides that a client is actually Client. In this case, Server sets $V = D_C + 1$, *hp=h(new password)*, $n_S=n_S+1$, and sends message M2 to Client. Client, on receiving M2, reveals *V* and compares it against D_C *+1*. Authenticity of M2 is checked by: $M_{21} \oplus SEED_S$ = $M_{22} \oplus SEED_C$ = $D_C + 1$. If the conditions hold, Client sets $n_C = n_C + 1$, $p_1 = p \oplus h(p_1)$, and authentication completes successfully.

Considered above scheme is not limited to a finite number of interactions before a password re-initialization, has low computational complexity both on a client and server sides. Performance of the suggested protocol is defined by three hash function calculations on the client side (Step 2.2), and three hash function calculations on the server side (Step 2.3). (Calculation of $h(p_1)$ is not counted in Steps 2.2 and 2.7, because it was calculated in Registration stage, Step 1.5.) Thus, total number of hash function calculations per authentication is six. Scheme [8], for *t*-th iteration, requires *N-t+4* hash function calculations. Similar estimates hold for other OTP schemas such as [5, 7]; better performance shows [3] but still it is worse than that of presented here schema.

The protocol is resistant to compromising of a server side password database. New passwords are generated by a client's side only with the simple procedure (client side password in Step 2.7, and server side password in Step 2.2).

The procedure of password generation may be represented as follows:

1. $p = h(K \oplus SEED_S)$, client side password: $p_1 = h(p)$, server side password: $hp_1 = h(p_1)$

2. client side password: $p_{t+1} = p \oplus h(p_t)$, server side password: $hp_{t+1} = h(p_{t+1})$, $t = 1,2,..$

The password generation procedure generates each new password using secret value of *p*. Used here approach is based on RHC similar to used in [3]. We consider chains of the form

$$\begin{aligned} &A_0 = h(p), S_0 = h(A_0), \\ &A_{t+1} = p \oplus h(A_t), S_{t+1} = h(A_{t+1}) \end{aligned}, \quad (1)$$

whereas chains [5] can be represented as follows:

$$\begin{aligned} &A_0 = h(p), S_0 = h(A_0), \\ &A_{t+1} = h(A_t), S_{t+1} = h(A_{t+1}) \end{aligned}, \quad (2)$$

where A_t is *t*-th authenticator (client-side password), and S_t is *t*-th server-side password. Distinction between (1) and (2) is in the use in (1) of secret information, *p*, in each step of generating of new authenticator A_{t+1}. With our chains, not having secret information, it is possible to advance the chain only in one step (from an authenticator to a server-side password). We call such hash chains RHC-1. Introduced RHC-1 can be easily generalized to RHC-k, allowing k consecutive advancing steps without possessing secret information:

$$A_0 = h(p), S_0 = h(A_0),$$
$$A_{nk-i} = h(A_{nk-i-1}), S_{nk-i} = h(A_{nk-i}), i = \overline{1, k-1}, n = 1,2,.. \quad (3)$$
$$A_{nk} = p \oplus h(A_{nk-1}), S_{nk} = h(A_{nk}), n = 1,2,..$$

Definition of RHC-k, given by (3), does not need bounding the full hash chain length, contrary to [3]. Definition (3) is a generalization of (1) and (2).

According to (1), a client conveys to server not only its authenticator, but also a server-side password for the next authentication session. The proposed OTP scheme generates RHC-1 in the forward direction contrary to schemes such as [5, 3] that pass hash chain in backward direction (from the last member to the first one).

Replay attack is not possible since each time password changes. Small number attack is not possible because a challenge (an authentication number) is not sent by a server. Preplay attack is not possible since password information goes only in one way, from a client to a server. A server is authenticated by a client with the help of nonce D_C : only the server could understand value of D_C , increase and close it with known to the server $SEED_C$ and $SEED_S$.

The scheme is resistant also to system failures, but restoration procedure using two hash function calculations will be considered elsewhere.

REFERENCES

1. *DIFFIE, W., HELLMAN, M.E.* 1976. New directions in cryptography. IEEE Trans. Inform. Theory v. IT-22, No 6, 644-654.

2. *EVANS, A., KANTROWITZ, W.*, WEISS, E. 1974. A user authentication scheme not requiring secrecy in the computer. Comm. ACM 17, No 8, 437-442.

3. *GOYAL, V., ABRAHAM, A., SANYAL, S., HAN, S.Y.* 2005. The N/R one time password system. In Proceedings of International Conference on Information Technology: Coding and Computing (ITCC'05), 4-6 April, 2005, v. 1, 733-738.

4. *HALLER, N.* 1995. The S/KEY one-time password system. *RFC 1760*, available from http://www.ietf.org.

5. *LAMPORT L.* 1981. Password authentication with insecure communication. Comm. ACM 24, No 11, 770-772, available from http://cmpe.emu.edu.tr/chefranov/cmpe552-06/Lecture%20Notes/Lamport81.pdf

6. *BICAKCI, K., BAYKAL, N.* 2002. Infinite length hash chains and their applications. In Proceedings of 11th IEEE Int. Workshops on Enabling Technologies: Infrastructure for Collaborating Enterprises (WETICE'02), 57-61.

7. *XIAO-RONG, C., QI-YUAN, F., CHAO D., MING-QUAN, Z.* 2005. Research and realization of authentication technique based on OTP and Kerberos. In Proceedings of 8th Int. Conference on High-Performance Computing in Asia-Pacific Region (HPCASIA'05), 5 pp.

8. *YEH, T.C., SHEN, H.Y., HWANG, J.J.* 2002. A secure one-time password authentication scheme using smart cards. IEICE Trans. Commun. v. E85-B, No. 11, 2515-2518.

9. *YUM, D.H., LEE, P.J.* 2005. Cryptanalysis of Yeh-Shen-Hwang's one-time password authentication scheme. IEICE Trans. Commun., v. E88-B, No. 4, 1647-1648, available from http://cmpe.emu.edu.tr/chefranov/cmpe552-06/Lecture%20Notes/Yum05.pdf