Journal of Applied Sciences

ISSN 1812-5654

science alert

ANSInet
an open access publisher
http://ansinet.com

Journal of Applied Sciences 8 (10): 1967-1971, 2008
ISSN 1812-5654

Scalable Storage Scheme from Forward Key Rotation

Chunbo Ma and Jun Ao
Information and Communication College, Guilin University of Electronic Technology,
Guilin, Guangxi, 541004, People's Republic of China

Abstract: The encryption scheme based on Forward Key Rotation is such a scheme that only the authorized person is allowed access to the designated files and the previous versions. In this study, we present a Forward Key Rotation storage scheme based on discrete logarithm and prove its security under random oracle model. Moreover, we propose another improved Forward Key storage scheme from pairing on elliptic curves. Compared to the scheme presented by Kallahalla *et al.* our scheme uses relatively short keys to provide equivalent security. In addition, the re-generated keys can be verified to ensure that the keys are valid in the improved scheme.

Key words: Secure storage, public key, scalability

INTRODUCTION

As more information becomes available in digital format, enormous quantities of data are created. With the increasing requirement to both temporarily and permanently retain information, the secure storage technique is attracting much attention. In a data storage system, the storage medium for preserving the information is an important target to a malicious attacker. Once the attacker breaks the system, he can gain unauthorized information, disclose some valuable secrets or prevent the access of legitimate users. In order to avoid these risks, the researchers have proposed many potential systems for securing stored information, such as NASD (Gobioff *et al.*, 1999, 1997), PASIS (Ganger *et al.*, 2001; Wylie *et al.*, 2000), CFS (Blaze, 1993), SNAD (Freeman and Miller, 2000; Miller *et al.*, 2001) and PLUTUS (Kallahalla *et al.*, 2003) and so on. Moreover, all these researches come into being a field of computer research. Currently, the major target of a secure storage system is ensuring information confidentiality, integrity and availability without substantially degrading performance.

Kallahalla *et al.* (2003) present a secure storage scheme, PLUTUS, in 2003. The primary goal of the scheme is to provide file owners with direct control over authorizing access to their files as well as scalable key management. Key revocation is a major problem in secure storage system. Due to the proliferation of keys and the use of file groups, the key revocation is very complicated. Since several files in the file system are encrypted with the same key, key revocation will result in mass re-encryption. However, Kallahalla *et al.* (2003) designed a key rotation scheme to alleviate the negative effects. The key rotation scheme can make the updated secret key relate to the previous versions. For the user of PLUTUS, when he is allowed access to a certain re-encrypted files using given file-group key, then he can generate previous versions from the given key. In other words, any valid user can re-generate the matching key for a given file if he has the latest file-group key. Therefore, we call the key rotate scheme designed by Kallahalla Forward Key Rotate (FKR) scheme.

The security of the FKR presented by Kallahalla *et al.* (2003) is based on the RSA. However, limited by the prime generation technique, producing a pair of suitable keys used in RSA scheme is not an easy thing. Moreover, until now there is no way to show the strength of the RSA is equivalent to that of decomposing a large number. Currently, RSA based schemes use relatively long keys compared to the security they provide. In this aspect, a scheme based on elliptic curves provides much shorter keys. As the requirement of practice, it is necessary to design other KFR storage scheme related to different intractable problems.

In this study, we first present a FKR storage scheme based on discrete logarithm and discuss its security. Subsequently, we present an improved FKR storage scheme from pairings on elliptic curves. Compared to the mechanism presented by Kallahalla *et al.* (2003) the improved scheme provides relatively short keys to perform the same function. Moreover, the user can verify the validity of the re-generated keys via bilinear pairings.

Corresponding Author: Chunbo Ma, Information and Communication College, Guilin University of Electronic Technology, Guilin, Guangxi, 541004, People's Republic of China

KALLAHALLA *et al.*'s FORWARD KEY ROTATION SCHEME

Kallahalla *et al.* (2003) proposed a Forward Key Rotation scheme used in PLUTUS system in 2003. In this section, we will briefly review their FKR scheme. Suppose that there exists a secure RSA encrypt scheme as defined in Rivest *et al.* (1978). Let U_0 be a user who will establish a FKR scheme, e_0 be the public key of U_0 and d_0 be the matching private key.

- User U_0 chooses $k_0 \epsilon Z^*_q$ uniformly at random and computes

$$k_1 = k_0^{d_0}, k_2 = k_1^{d_0}, \cdots, k_l = k_{l-1}^{d_0}$$

- When user $U_{i \neq 0}$ wants to access the FKR scheme, the user U_0 gives $U_{i \neq 0}$ the latest key $k_l = k_{l-1}^{d_0}$. Hence, $U_{i \neq 0}$ has ability to compute $k_{l-1} = k_l^{e_0} = k_{l-1}^{d_0 e_0}$ and get any $k_{z \leq l}$.

The security of the scheme is based on RSA. Kallahalla *et al.* (2003) simplify the key management of PLUTUS using this FKR scheme.

PRELIMINARIES

Complexity assumptions: We assume that a prime p is chosen at random such that p-1 has a large prime factor q. Let g be an element of order q. Define the following problems.

- **Computation diffie-hellman (CDH):** Given g^a, g^b for unknowns a, b ϵZ^*_q, compute $g^{ab} \epsilon Z^*_q$.
- **Decision diffie-hellman (DDH):** Given g^a, g^b, g^c for unknowns a, b, c ϵZ^*_q, decide whether $g^{ab} = g^c$.
- **K-exponent assumption (k-E):** Given $\{g, g^x, g^{x^2}, \cdots, g^{x^k}\}$ for unknown x, compute $g^{x^{k+1}} \in Z^*_p$ (Zhang *et al.*, 2004).

Bilinear maps: Let G_1 be a cyclic multiplicative group generated by g, whose order is a prime q and G_2 be a cyclic multiplicative group of the same order q. Assume that the discrete logarithm in both G_1 and G_2 is intractable. A bilinear pairing is a map: $G_1 \times G_1 \rightarrow G_2$ and satisfies the following properties:

- **Bilinear:** $e(g^a, p^b) = e(g, p)^{ab}$. For all g, p ϵ G_1 and a, b ϵ Z_q, the equation holds.
- **Non-degenerate:** There exists p ϵ G_1, if e(g, p) = 1, then g = 0.
- **Computable:** For g, p ϵ G_1, there is an efficient algorithm to compute e(g, p).

Typically, the map e will be derived either from the modified Weil pairing (Boneh *et al.*, 2001; Boneh and Franklin, 2003) or the tate pairing (Paulo *et al.*, 2002) on an elliptic curve over a finite field. Pairings and other parameters should be selected in proactive for efficiency and security. In group G_1, the CDH and K-E assumptions defined are intractable. However, the DDH assumption is tractable.

General scheme: A FKR storage scheme consists of four algorithms.

- **Initialize:** Given the security parameter l, the algorithm outputs the system parameters.
- **Key generation (i):** Input the number i, output i-th secret key K_i used to encrypt i-th files.
- **Encrypt (K_{i+1}, K_i, $File_i$):** Input two secret keys (K_{i+1}, K_i) and a file $File_i$. The algorithm encrypts $File_i$ using (K_{i+1}, K_i) and outputs the corresponding ciphertext EF_i.
- **Decrypt (K_{i+1}, K_i, K_i):** Input the ciphertext EF_i and the secret keys (K_{i+1}, K_i). The algorithm decrypts the ciphertext using the secret key (K_{i+1}, K_i) and outputs the corresponding plaintext $File_i$.

Security notions: We define adaptively chosen ciphertext security of a FKR storage scheme. Security is defined using the following game between an Attacker and Challenger.

- **Setup:** The Challenger initializes the system. The Challenger gives the Attacker the resulting system parameters.
- **Query phase 1:** The Attacker adaptively issues encrypt queries q_1, q_2, ..., q_m and decryption queries q_1, q_2, ..., q_m, respectively. The Challenger simulates Encrypt and Decrypt, respectively and responds with matching answers.
- **Challenge:** Once the Attacker decides that Query phase 1 is over it outputs two equal length files ($File_0$, $File_1$) to the Challenger. The Challenger picks a random bit $\lambda \epsilon \{0, 1\}$ and encrypts the message $File_\lambda$. It gives ciphertext C to the Attacker.
- **Query phase 2:** The Attacker continues to adaptively issue encryption and decryption queries. The Challenger responds as in the phase 1. These queries may be asked adaptively as in Query phase 1. Note that the decrypt query $q_j = C$ is not permitted, where $0 \leq j \leq n$.

- **Guess:** Finally, the Attacker outputs a guess $\lambda' \in \{0, 1\}$ for λ and wins the game if $\lambda' = \lambda$.

The storage scheme is secure against chosen ciphertext attack, if the Attacker has a negligible advantage $\epsilon = |Pr(\lambda = \lambda')-1/2|$ to win the game.

FORWARD KEY ROTATION STORAGE SCHEME

In this part of study, we will design a FKR storage scheme related to discrete logarithm. We suppose that Server who has a series of files $\{File_1, File_2, ..., File_n\}$ will establish the FKR storage scheme and a user Alice asks to access it. Let p be a large prime and $\langle g \rangle$ be a cyclic multiplicative group generated by g, whose order is a prime q such that q|(p-1). There exists a pair of security algorithms (E, D), where the algorithm E is a secure encryption algorithm based on discrete logarithm and D is the matching decryption algorithm.

Step 1: Server chooses a random number $r \in Z_q^*$ and computes $K_i = g^{r^i} \bmod p$ as a secret key.

Step 2: Server generates the encrypted files $EF_i = E_{K_{i+1}}(K_i \| File_i)$, where a || b denotes the concatenate of a and b. Thereafter, server publishes all encrypted files.

When Alice asks to access to the FKR storage scheme at the point j, Server sends the secret key K_{j+1} to Alice. Since Alice can obtain any encrypted files, she can compute

$$K_j \| File_j = D_{K_{j+1}}(EF_j)$$

and get K_j and $File_j$. Similarly, Alice can obtain K_{j-1} and $File_{j-1}$ via K_j and EF_{j-1}.

However, by the K-E assumption defined already, Alice can't produce the secret key K_{j+2} using K_j and K_{j+1}. In other words, Alice doesn't have ability to decrypt EF_{j+1} even though she has secret key $K_{i\neq j+1}$.

SECURITY

The security of the scheme is partly based on the algorithms (E, D). Thereby, we assume that these two algorithms are secure. Given $\{g^r, g^{r^2}, \cdots, g^{r^i}\}$, one can't deduce $g^{r^{i+1}}$ by the K-E assumption mentioned already. It means that if Server gives an access point at j, one can't access to the files $File_k$, where k>j.

Then we will give the following theorem to show the security of the proposed storage scheme.

Theorem: We assume that an attacker Eve who can, with success probability ϵ, break the FKR storage scheme within a time τ by asking Encrypt and Decrypt oracles at most q_E and q_D queries respectively, then there exists a challenger Alice who running in a time τ' can solve the DDH problem with success probability ϵ', where

$$\varepsilon' = \varepsilon,\ \tau' = \tau + (q_E + q_D + 1)t_{ED}$$

where, t_{ED} is the time for performing an encryption or decryption algorithm.

Proof: We assume that Eve is an attacker who has the ability to break the storage scheme. Then there exists a challenger Alice who can solve the DDH problem by running Eve as a subroutine. The system chooses a random bit $b \in \{0, 1\}$ and a random number $r \in Z_q^*$. The challenger Alice is given $\{g^r, g^{r^2}, \cdots g^{r^{i-1}}, T, g^{r^{i+1}}, \cdots, g^{r^k}\}$. If b = 1, the system sets $T = g^{r^i}$, otherwise, chooses a random number $T \in Z_p^*$.

The attacker Eve is allowed to issue Encrypt, Decrypt and Challenge queries. The challenger Alice will simulate the corresponding oracles to output the answers.

Query phase 1: The attacker Eve is allowed to issue following queries.

Encrypt queries: Eve chooses a random number $j \in [l, k]$ and takes the file $File_j$ and then issues query on $(j, File_j)$.

- If $j \neq i$, Alice computes $EF_j = E_{K_{j+1}}(K_j \| File_j)$ and outputs EF_j as the answer, where $K_{j+1} = g^{r^{j+1}}$ and $K_j = g^{r^j}$.
- If j = i, Alice outputs error messages and halts.

Decrypt queries: Eve chooses a random number $j \in [l, k]$ and takes the ciphertext EF_j and then issues query on (j, EF_j).

- If $j \neq i$, Alice computes $K_j \| File_j = D_{K_{j+1}}(EF_j)$ and outputs K_j and $File_j$ as the answer, where $K_{j+1} = g^{r^{j+1}}$ and $K_j = g^{r^j}$.
- If j = i, Alice outputs error message and halts.

Since above simulation is perfect, the attacker Eve can't distinguish the simulated result from the actual results. The above queries can be asked several times. When Eve decides this phase is over, he issues challenge query.

Challenge query: Eve outputs two equal length files $File_0$ and $File_1$ to Alice. Upon receiving the two files, Alice chooses a random bit $\lambda \in \{0, 1\}$ and computes

$$EF_i = E_{K_{i+1}}(T \| File_\lambda)$$

where, $K_{i+1} = g^{r^{i+1}}$. Thereafter, Alice sends EF_j to Eve as the answer.

Note that the Challenge query is allowed only once.

Query phase 2: The attacker Eve continues to adaptively issue encryption and decryption queries. The challenger Alice will respond as in the phase 1. However, decryption query $q_j = EF_i$ is not permitted. We assume that Eve issues at most q_E encrypt queries and q_D decrypt queries.

Guess: After receiving the answer from Alice, Eve outputs his guess λ'. If $\lambda' = \lambda$, Alice decides b = 1, otherwise b = 0.

Since Eve has ability to break the scheme with non-negligible probability ϵ, i.e., outputs $\lambda' = \lambda$ with probability ϵ, then the challenger Alice can solve DDH with the same probability.

IMPROVED SCALABLE FKR STORAGE SCHEME

In practice some devices only have limited capability, so we should design a scheme which can provide shorter keys. Thereby, an improved FKR storage scheme from pairing on elliptic curve is presented part of research. In this scheme the re-generated keys can be verified via bilinear pairings. Thus also provide a measure for a user to verify the validity of the plaintext.

Let G_1 and G_2 be two groups that support a bilinear map. There exists a pair of security algorithms (E, D), where the algorithm E is a secure encryption algorithm based on elliptic curves and D is the matching decryption algorithm.

Step 1: Server chooses a random number $r \epsilon Z^*_q$ and computes $K_i = g^{r^i}$ as a secret key, where $g^{r^i} \in G_1$.

Step 2: Server generates the encrypted files $EF_i = E_{K_{i+1}}(K_i \| File_i)$. Thereafter, server publishes all encrypted files.

When Alice asks to access to the FKR storage scheme at the point j, server sends the secret key K_{j+1} to Alice. Since Alice can obtain any encrypted files, she can compute

$$K_j \| File_j = D_{K_{j+1}}(EF_j)$$

and get K_j and $File_j$. Similarly, Alice can obtain K_{j-1} and $File_{j-1}$ via K_j and EF_{j-1}.

Step 3: After computing K_j and K_{j-1}, Alice performs the following step to verify the validity of the re-generated keys.

$$e(K_{j+1}, K_{j-1}) = e(K_j, K_j)$$

If above equation is true, the re-generated keys are valid. Otherwise, Alice outputs error messages.

CONCLUSIONS

Secure storage is a crucial problem in the Internet. Motivated by Kallahalla's forward key rotation and the requirement of the short key schemes, we present two FKR storage scheme. One is based on discrete logarithm and another is from bilinear pairing on elliptic curve. The latter storage scheme is suitable for implementation in many scenarios, especially those where the storage capability of the users is limited.

REFERENCES

Blaze, M., 1993. A cryptographic file system for unix. First ACM Conference on Communications and Computing Security, pp: 221-230.

Boneh, D., B. Lynn and H. Shacham, 2001. Short signatures from the Weil pairing. Advances in Cryptology-Asiacrypt' 2001, Gold Coast, Australia, Lecture Notes in Computer Science, 2248: 514-532.

Boneh, D. and M.K. Franklin, 2003. Identity-based encryption from the Weil pairing. SIAM. J. Comput., 32 (3): 586-615.

Freeman, W. and E. Miller, 2000. Design for a decentralized security system for network-attached storage. In: Proceedings of the 17th IEEE Symposium on Mass Storage Systems and Technologies, pp: 361-373.

Ganger, G.R., P.K. Khosla, M. Bakkaloglu, M.W. Bigrigg, G.R. Goodson, S. Oguz, V. Pandurangan, A. Craig, N. Soules, J.D. Strunk and J.J. Wylie, 2001. Survivable storage systems. DARPA Information Survivability Conference and Exposition, 2: 184-195.

Gobioff, H., G. Gibson and D. Tygar, 1997. Security for Network Attached Storage Devices. CMU SCS Technical Report CMU-CS-97-185.

Gobioff, H., D. Nagle and G. Gibson, 1999. Embedded Security for Network-Attached Storage. CMU SCS Technical Report CMU-CS-99-154.

Kallahalla, M., E. Riedel, R. Swaminathan, Q. Wang and K. Fu, 2003. PLUTUS: SCALAble secure file sharing on untrusted storage. In: Conference on File and Storage Technology (FAST'03), pp: 29-42.

Miller, E.L., D.D.E. Long, W. Freeman and B. Reed, 2001. Strong security for distributed file systems. In: Proceedings of the 20th IEEE International Performance, Computing and Communications Conference, pp: 34-40.

Paulo, S.L., M. Barreto, H.Y. Kim, B. Lynn and M. Scott, 2002. Efficient algorithms for pairing-based cryptosystems. In Crypto'02, pp: 354-368.

Rivest, R.L., A. Shamir and L. Adleman, 1978. A method for obtaining digital signatures and public-key cryptosystems. CACM., 21 (2): 120-126.

Wylie, J.J., M.W. Bigrigg, J.D. Strunk, G.R. Ganger, H. Kiliccote and P.K. Khosla, 2000. Survivable information Storage systems. IEEE Comput., 33 (8): 61-68.

Zhang, F., R. Safavi-Naini and W. Susilo, 2004. An Efficient Signature Scheme from Bilinear Pairings and Its Applications. Practice and Theory in Public Key Cryptography-PKC 2004, Springer-Verlag, Berlin, pp: 277-290.