社団法人 情報処理学会 研究報告
IPSJ SIG Technical Reports

2008－DD－67（6）
2008/7/25

過去の暗号通信を考慮した動的鍵共有方式

辻　貴介†　清水　明宏††

† 株式会社トリニティーセキュリティーシステムズ
〒101-0031 東京都千代田区東神田 1-7-8 アルテビル東神田 8F
†† 高知工科大学
〒782-8502 高知県香美郡土佐山田町宮ノ口 185
E-mail: †tsuji@trinity-ss.com, ††shimizu.akihiro@kochi-tech.ac.jp

あらまし 近年，実生活により作成された様々な情報のディジタル化が進んでおり，個人情報などの秘密情報をインターネットを介し，生成，加工，保存，閲覧するサービスが増加している．そのようなサービスにおいて，秘密情報の安全性を確保するため，送受する際に暗号方式が必要である．しかしながら，暗号解読技術の発達により，過去の通信情報の安全性が損なわれる危険性がある．本稿では，過去の通信情報の安全性をより向上させるための鍵共有方式を提案する．本方式では，攻撃者が通信情報を取得し通信者の鍵を盗んだとしても，過去の暗号通信が解読され難い．
キーワード 秘密情報，インターネット，安全性，暗号，鍵共有

An Authenticated Key Sharing Scheme to Protect Past Secrets

Takasuke TSUJI† and Akihiro SHIMIZU††

† Trinity Security Systems, Inc.,
Alte Building Higashi Kanda 8F, 1-7-8 Higashikanda Chiyoda-ku, Tokyo, 101-0031, Japan
†† Kochi University of Technology,
185 Miyanokuchi, Tosayamada-cho, Kami-shi, Kochi, 782-8502, Japan
E-mail: †tsuji@trinity-ss.com, ††shimizu.akihiro@kochi-tech.ac.jp

Abstract Various information are used in real life, and those information are changed to digital data for using many computer services. Secret data such as users' life logs are used in many computer systems, in which the user can use services conveniently. In those computer systems, secret data are created, changed, stored, and read through the Internet. Cryptosystems are necessary for safety internet communications. However, it has risks that past communication data are decipherable if the same encryption key is used all the time. In this manuscript, we propose an authenticated key sharing scheme to protect past communication data. In this scheme, the past encryption data can be difficultly guessed if the attacker intercepts communication data and steals keys of users.
Key words secret data, internet, security, cryptosystem, key sharing scheme

1. Introduction

Computer systems have been developed and people can use many services using computer systems. For example, people can make and see their reports in office information management systems. In those systems, secret data such as users' life logs are included in the reports. Internet communications are increasing, and those secret data are created, changed, stored, and read through the Internet.

Data through the Internet can be easily wiretapped and can be seen secret data, which are included in sending data. Cryptosystems such as the AES(Advanced Encryption Standard)[1] are necessary to protect the communication data through the Internet. Encryption key has to be securely shared among users before encryption communications using cryptosystems.

The Diffie-Hellman key agreement method[2] is the most famous key agreement scheme. However, this scheme has a security problem, the Man-In-the-Middle attack, in which the attacker gets key negotiations of users, and she/he can easily create encryption keys to communicate with each user. The AKE(Authenticated Key Exchange) solves this problem.

In the AKE, the user authenticates the other party and exchanges an encryption key. W. Diffie *et al.* have solved the Man-In-the-Middle attack by using digital signatures [3]. In this scheme, each public key initially have to get, and the modulus operator must be used.

The EKE(Encrypted-Key-Exchange) [4] is a key exchange scheme, in which a session key for encryption communications is shared by using a password. The EKE has variation; public key cryptosystems the RSA [5] or the ElGamal [6] is applied to the EKE. In each variation, key negotiation data are encrypted by using a password and are sent. Then the users calculate the session key from key negotiation data. In the EKE, key negotiation data are sent by using a password. Thus, those data can be seen if the password is leak out. If session key isn't protect, the attacker can see the session key from key negotiation data. If a public key cryptosystem is applied to the EKE, the attacker can see the session key by using secret key.

Existing key exchange scheme the EKE is applied symmetric key cryptosystems. Moreover, the EKE adds public key cryptosystems to symmetric key cryptosystems for the against security problems. One-time password authentication protocols [7], [8] are simple and secure. Those protocols against weakness of authentication protocols, which are based symmetric key cryptosystems. In this manuscript, an authenticated key sharing scheme is proposed. This scheme isn't based on symmetric key cryptosystems, is applied on one-time password authentication mechanism.

2. Proposal Scheme

In many encryption communication systems, the encryption key is shared during users at first, and the communication data are encrypted by using the shared encryption key. In the EKE key exchange scheme, the encryption key is shared by using password, which has been shared at first or has been changed by using first-shared key. In those schemes, the attacker can guess the password from many data such as the encrypted data.

An encryption key, a password, or other secret keys is stored in the computers of users for the next encryption communication and isn't usually changed. Encryption communication data can be easily decrypted if the secret key is stolen. Proposal scheme solves this problem by changing the secret key. This proposal scheme has two core protocols; first one authenticates by using the present secret and second one authenticates by using the next secret. Two core protocols are illustrated below.

2.1 Core Protocol 1

Core protocol 1 has secure algorithms; a seed is securely shared, and the past secret keys aren't calculated from the present secret key.

Before the encryption communication, the registration phase is executed. In this phase, communicators A(Alice) and B(Bob) are share a secret key V_1. This V is calculated by A for the first or the next key sharing. For example, V is calculated as $V_1 = h(C_1)$, where h is a one-way function such as a secure hash function, and C is challenge data for authentication. This challenge data is created by using A's secret such as a random number or password-hidden data. The calculated V_1 is sent from A to B. Then, A stores V_1, C_1, and/or S's seed data, and B stores V_1. Usually, A stores C_1 or a seed data to calculate C_1.

For sharing encryption key, the ith key sharing phase is executed. Before the ith key sharing phase, A stores C_i or a seed of C_i, and B stores V_i. If A stores a seed of C_i, C_i is calculated from a seed of C_i and another data such as A's password.

When A and B share a seed of an encryption key, A sends α and β to B, where

$$\alpha = m_{V_i}(C_i),$$
$$\beta = e_{C_i}(V_{i+1}).$$

Here, m is a masking function, e.g., $m_y(x)$ is the masked data of x by using y. Similarly, e is an encryption function, e.g., $e_y(x)$ is the encrypted data of x by using y. Figure 2.1 shows the ith key sharing phase of core protocol 1.

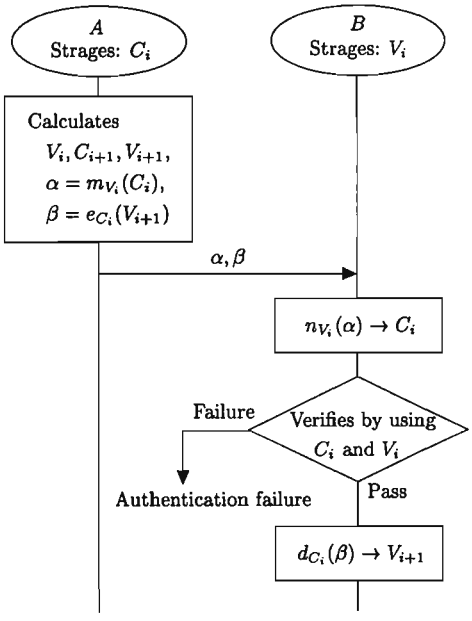

Fig. 2.1 The ith key sharing phase of core protocol 1.

The ith key sharing phase of core protocol 1 is as follows. First, A calculates V_i, C_{i+1}, V_{i+1}, α, and β, and she/he sends

α and β to B. When B receives them, C_i is retrieved from $n_{V_i}(\alpha)$. Here, n is a retrieval function, e.g., $n_y(x)$ is the retrieved data of x by using y. If x equals $m_y(z)$, $n_y(x)$ equals $n_y(m_y(z)) = z$. Next, B authenticates A's challenge data. Here, one of authentication mechanisms is explained. B compares the hashed data of the retrieved C_i and the stores V_i. If they are the same, A is authenticated, and the next procedure is executed. $V_{i+1} = d_{C_i}(\beta)$ is decrypted from β. Here, d is a decryption function, e.g., $d_y(x)$ is the decrypted data of x by using y. If x equals $e_y(z)$, $d_y(x)$ equals $d_y(e_y(z)) = z$. Then, the decrypted V_{i+1} is stored for the next key sharing.

After A and B had shared C_i, they calculate an encryption key K and securely communicate by using K, where K is calculated from C_i by using a one-way function and is calculated by another way of V.

2.2 Core Protocol 2

Core protocol 2 has secure algorithms; a seed is securely shared, and the past secret keys aren't calculated from the present secret key. In this protocol, A calculates a challenge data C, which hasn't be calculated by using only the next secret data V.

Before the encryption communication, the registration phase is executed. In this phase, communicators A and B are share a secret key V_1. This V is calculated by A for the first or the next key sharing. For example, V is calculated by using A's secret such as a random number or password-hidden data. The calculated V_1 is sent from A to B. Then, A stores V_1, C_1, and/or S's seed data, and B stores V_1. Usually, A stores C_1 or a seed data to calculate C_1.

For sharing encryption key, the ith key sharing phase is executed. Before the ith key sharing phase, A stores C_i or a seed of C_i, and B stores V_i. If A stores a seed of C_i, C_i is calculated from a seed of C_i and another data such as A's password.

When A and B share a seed of an encryption key, A sends α and β to B, where

$$\alpha = m_{V_i}(C_i),$$
$$\beta = e_{C_i}(V_{i+1}).$$

Figure 2.2 shows the ith key sharing phase of core protocol 2.

The ith key sharing phase of core protocol 2 is as follows. First, A calculates V_i, C_{i+1}, V_{i+1}, α, and β, and she/he sends α and β to B. When B receives them, C_i is retrieved from $n_{V_i}(\alpha)$. Next, $V_{i+1} = d_{C_i}(\beta)$ is decrypted from β. Then, B authenticates A's challenge data. Here, one of authentication mechanisms is explained. B compares the hashed data of the retrieved C_i and the retrieved V_{i+1}. If they are the same, A is authenticated, and the next procedure is executed. Then, the decrypted V_{i+1} is stored for the next key sharing.

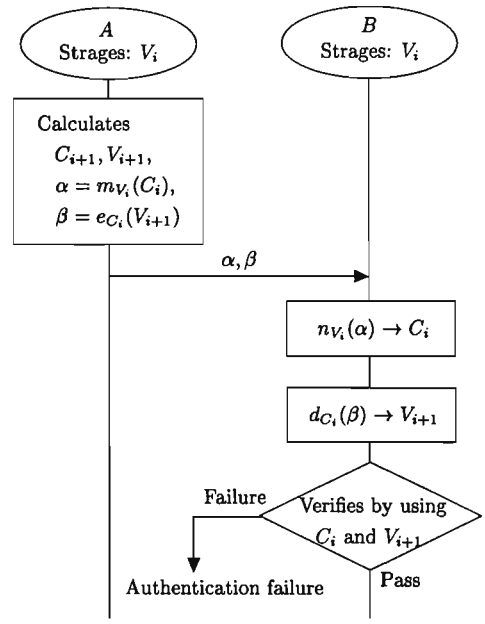

Fig. 2.2 The ith key sharing phase of core protocol 2.

After A and B had shared C_i, they calculate an encryption key K and securely communicate by using K, where K is calculated from C_i by using a one-way function.

3. Security and Performance Considerations

The EKE scheme is general because this scheme can use several authentication schemes. However, the EKE scheme is applied symmetric key cryptosystems, the past communication data can be seen if the shared key is stolen. Proposal scheme is applied one-time password authentication mechanism and another one way mechanism of symmetric key cryptosystems. Therefore, the past encryption key can't be calculated by using the shared key, i.e., the past communication data is safety if the shared key is stolen.

The AKE scheme using the scheme with a digital signature scheme can protect the past communication data. However, the AKE scheme can't use other authentication schemes. Proposal has equivalent security to the AKE scheme and can be applied various authentication schemes such as one-time password authentication scheme, and other scheme based on symmetric key cryptosystems or public key cryptosystems. If proposal scheme doesn't adopt authentication schemes using modulus operators, this scheme doesn't need to calculate by using modulus operators. Moreover, proposal scheme can share the encryption key with only one way challenge against over two way challenges of the AKE. Thus, proposal scheme can protect past secrets and has general and useful characteristics.

4. An Application on an Authentication Scheme

The SAS-2[8] is a simple and secure authentication scheme. In this section, proposal scheme is applied to the SAS-2. Key sharing protocol on the SAS-2 is as follows.

Before the encryption communication, the registration phase is executed. In this phase, communicators A and B are share a secret key V_1. This V_1 is calculated by A as $V_1 = h(P, N_1)$, where P is A's password, and N is a random number. The calculated V_1 is sent from A to B. Then, A stores N_1, and B stores V_1.

For sharing encryption key, the ith key sharing phase on the SAS-2 is executed. Before the ith key sharing phase, A stores N_i, and B stores V_i. Here N_{i+1} is a random number, which is used in the ith key sharing phase.

When A and B share a seed of an encryption key, A sends α and β to B, where

$$\alpha = m_{V_i}(C_i),$$
$$\beta = e_{C_i}(V_{i+1}, R_i),$$

where R is a random number for verification. Figure 4 shows the ith key sharing phase on the SAS-2.

The ith key sharing phase on the SAS-2 is as follows.

First, A inputs A's password, and calculates $V_i = h(P, N_i)$. Then, A calculates N_{i+1}, V_{i+1}, R_i, $C_i = h(V_{i+1}, R_i)$, α, and β, and she/he sends α and β to B. When B receives them, C_i is retrieved from $n_{V_i}(\alpha)$. Next, V_{i+1} and R_i are decrypted from β with the retrieved C_i. To authenticate A, B calculates $h(V_{i+1}, R_i)$ by using the retrieved V_{i+1} and R_i. Then, B compares the calculated data and the retrieved C_i. If they are the same, A is authenticated, and the next procedure is executed.

For a mutual authentication, B calculates $\gamma = h(C_i)$ and sends it to A. When A receives γ, A calculates $h(C_i)$. Then, A compares the calculated data and the received γ. If they are the same, B is authenticated. Then, the decrypted V_{i+1} is stored for the next key sharing.

After A and B had shared C_i, they calculate an encryption key K and securely communicate by using K, where K is calculated from C_i by using a one-way function.

5. Conclusion

An authenticated key sharing scheme using one-time password authentication mechanism is proposed in this manuscript. This scheme can be applied various authentication schemes and can perform by only one way challenge. Thus, this is scheme general and useful. We will apply this scheme to other authentication schemes.

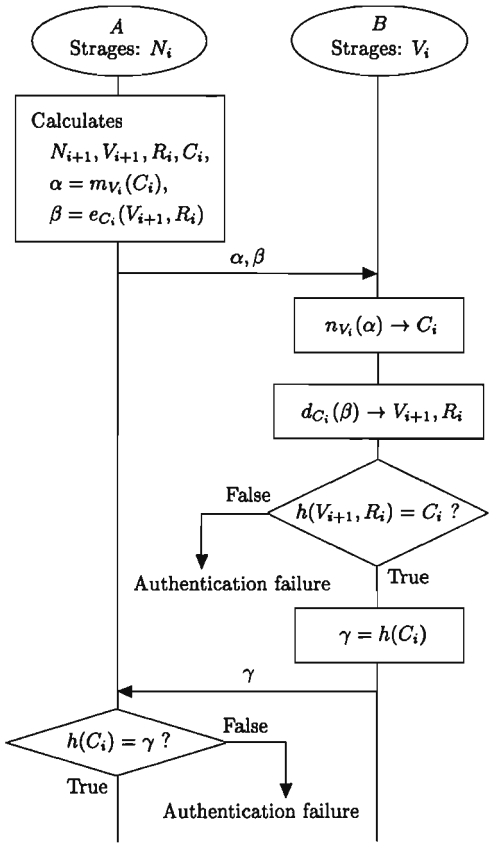

Fig. 4 The ith key sharing phase on the SAS-2.

References

[1] National Institute of Standards and Technology, "Announcing the Advanced Encryption Standard (AES)," FIPS Publication 197, Nov. 2001.

[2] W. Diffie and M. E. Hellman, "New directions in cryptography," IEEE Trans. on Information Theory, vol.IT-22, no.6, pp.644–654, Nov. 1976.

[3] W. Diffie, P.C. van Oorschot, and M.J. Wiener, "Authentication and authenticated key exchanges," Designs, Codes and Cryptography, vol.2, pp.107–125, Mar. 1992.

[4] S.M. Bellovin and M. Merritt, "Encrypted key exchange: password-based protocols secure against dictionary attacks," Proceedings of the 1992 IEEE Computer Society Conference on Research in Security and Privacy, pp.72–84, May 1992.

[5] R.L. Rivest, A. Shamir, and L.M. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," Commun. ACM, vol.21, no.2, pp.120–126, 1978.

[6] T. ElGamal, "A public-key cryptosystem and a signature scheme based on discrete logarithms," IEEE Trans. on Information Theory, vol.IT-31, no.4, pp.469–472, Jul. 1985.

[7] L. Lamport, "Password authentication with insecure communication," Commun. ACM, vol.24, no.11, pp.770–772, 1981.

[8] T. Tsuji and A. Shimizu, "A one-time password authentication protocol for mobile communications and internet protocols," IEICE Trans. Commun., vol.E87-B, no.6, pp.1594–1600, Jun. 2004.