Journal of The Korea Society of Computer and Information
Vol. 17 No. 4 , April 2012

2012-17-4-3-3

Yi et al.'s Group Key Exchange Protocol : A Security Vulnerability and its Remediation

Youngsook Lee*, Jeeyeon Kim**, Dongho Won**

Yi등이 제안한 그룹 키 교환 프로토콜의 보안 취약성 및 개선 방법

이 영 숙*, 김 지 연** , 원 동 호**

Abstract

A group key exchange (GKE) protocol is designed to allow a group of parties communicating over a public network to establish a common secret key. As group-oriented applications gain popularity over the Internet, a number of GKE protocols have been suggested to provide those applications with a secure multicast channel. Among the many protocols is Yi et al.'s password-based GKE protocol in which each participant is assumed to hold their individual password registered with a trusted server. A fundamental requirement for password-based key exchange is security against off-line dictionary attacks. However, Yi et al.'s protocol fails to meet the requirement. In this paper, we report this security problem with Yi et al.'s protocol and show how to solve it.

▸ Keyword : group key exchange, password, dictionary attack, identity-based cryptography

요 약

그룹 키 교환 프로토콜은 일련의 그룹을 형성하는 다수의 통신 참여자들이 공개된 통신망을 통해 그룹의 공통 비밀키를 설정하기 위한 목적으로 설계된다. 그룹 지향적인 응용분야들이 인터넷상에서 인기를 더해감에 따라서 이들 응용분야에 안전한 멀티캐스트 채널을 제공하기 위해 많은 그룹 키 교환 프로토콜이 제안되었다. 그동안 제안된

● 제 1저자 : 이영숙 ● 교신저자 : 원동호
• 투고일 : 2011. 12. 23, 심사일 : 2012. 01. 17, 게재확정일 : 2012. 02. 01.
* 호원대학교 사이버수사 경찰학부(Department of Cyber Investigation Police, Howon University)
** 성균관대학교 정보통신공학부(School of Information and Communication Engineering, Sungkyunkwan University)
※ 이 논문은 2012년 호원대학교 연구비 지원을 받은 것임

그룹 키 교환 프로토콜 중에 최근 Yi 등이 발표한 패스워드 기반 프로토콜이 있다. 이 프로토콜에서는 각 프로토콜 참가자가 자신의 패스워드를 가지고 있으며 이 패스워드는 신뢰할 수 있는 서버에 등록되어 있다고 가정된다. 패스워드 기반 키 교환에서 가장 근본적인 보안 요구사항은 오프라인 사전 공격에 대한 안전성이라 할 수 있다. 그러나 Yi 등이 제안한 프로토콜은 패스워드에 기반한 프로토콜임에도 불구하고 이 요구사항을 만족하지 못하는 것으로 드러났다. 본 논문에서는 Yi 등의 프로토콜에서 발견되는 이러한 보안 문제점을 지적하고 그 해결책을 제시한다.

▸ Keyword : 그룹키 교환, 패스워드, 사전공격, 신원기반 암호

I. Introduction

The increasing ubiquity of computer networks is accelerating the development of group-oriented applications in which a group of parties communicate collaboratively to achieve their common interest or objective. Typical group-oriented applications include video/audio teleconferencing, distributed multiplayer games, grid computing, collaborative workspaces, and social networking services. In particular, social networking services such as Twitter[1] and Facebook[2] have recently gained tremendous popularity and are redefining our sense of community. The proliferation of group-oriented applications has led to a growing concern in security of group communications. The current Internet, by design, is an open network which might be controlled by an adversary. Today's adversaries are equipped with more powerful computing resources and attacking tools than ever before. The situation gets even worse when we consider malicious insiders. In general, we cannot expect complete trust among all group members just because they collaborate to achieve a specific purpose; collaboration does not imply full trust. Perhaps malicious insiders pose the most serious security threat to many organizations and enterprises.

One valuable tool for protecting group communications is protocols for group key exchange (GKE). A group of parties communicating over a public network can generate a common secret key (called a session key) by running a GKE protocol. Once a session key has been established, the parties can use this key to encrypt and/or authenticate their subsequent multicast messages. This represents a typical way of communicating confidentially and with integrity over a public channel. The session key, of course, must be known only to the intended parties at the end of the protocol run, because otherwise the whole system becomes vulnerable to all manner of attacks. Roughly stated, a key exchange protocol satisfying this requirement is said to be authenticated. Any protocol for authenticated key exchange inherently requires that the protocol participants establish their long-term authentication secrets (either low-entropy passwords or high-entropy cryptographic keys) before they ever run the protocol.

Protocols for password-authenticated key exchange are designed to work even when the authentication secrets are human-memorable passwords chosen from a small known set of values. These password-based protocols, despite their practical significance in today's computing environments, are notoriously hard to design right. The major hurdle to password-authenticated key exchange is (off-line) dictionary attacks in which an adversary exhaustively enumerates all possible passwords in an off-line manner to find out the correct one. Indeed, many protocols, even some with a claimed proof of security, have been found to be vulnerable to a dictionary attack years after they were published. In this letter, we present another instance of the vulnerability that can be identified in the password-authenticated GKE protocol proposed recently by Yi et al. [3]. Like the previous protocols of [4][5][6], Yi et al.'s protocol assumes a

Kerberos-like authentication model in which each client, who is a potential participant of the protocol, shares a password with a trusted server but not with any other clients. This model enjoys the obvious practical advantage that no matter how many different session keys for different groups a client wants to generate, he/she does not need to hold multiple passwords but only needs to remember a single password shared with the server. Yi et al.'s protocol differs from previous designs [4][5][6] in two aspects: (1) it can be constructed generically from any GKE protocol secure against passive adversaries and (2) it employs identity-based cryptography where an arbitrary identity like an email address can serve as a public key. Despite its practicality and uniqueness, Yi et al.'s protocol should not be adopted in its present form. Due to a fatal flaw in its design, Yi et al.'s protocol fails to protect the passwords of its participants against a dictionary attack. We here report this critical problem with Yi et al.'s protocol and present how to solve it.

II. Preliminaries

As already mentioned, Yi et al.'s protocol [3] is based on identity-based cryptography where an arbitrary identity serves as a public key. In this section, we revisit the relevant terminology and definitions from [3]. No originality is claimed for this section.

1. Identity-Based Encryption

An identity-based encryption (IBE) scheme is specified by four randomized algorithms: Setup, Extract, Encrypt, Decrypt as follows.

- Setup: On input a security parameter k, it returns *params* (public system parameters) and *master-key* (known only to the "Private Key Generator").
- Extract: On inputs params, master-key and a public identity ID $\in \{0, 1\}*$, it returns a private key dID.
- Encrypt: On inputs params, ID, and a message $M \in \mathbf{M}$ (the plaintext space), it returns a ciphertext $C \in \mathbf{C}$ (the ciphertext space).
- Decryption: On inputs params, $C \in \mathbf{C}$, and a private key dID, it returns $M \in \mathbf{M}$.

Chosen ciphertext security is the standard acceptable notion of security for a public key encryption scheme. An IBE scheme is semantically secure against the adaptive chosen ciphertext attack if no polynomial bounded adversary A has a non-negligible advantage against the challenger in the following game:

- *Initialize*: The challenger runs the Setup algorithm, gives *params* to the adversary, but keeps the *master-key* to itself.
- *Phase* 1: The adversary adaptively asks a number of different queries q_1, q_2, ..., q_m, where q_i is either Extract(ID_i) or Decrypt(ID_i, C_i).
- *Challenge*: Once the adversary decides that Phase 1 is over, it outputs a pair of equal length plaintexts (M_0, M_1) and an identity ID on which it wishes to be challenged, where ID must not appear in Phase 1. The challenger picks a random bit $b \in \{0,1\}$ and sends C = Encrypt(ID, M_b) as the challenge to the adversary.
- *Phase* 2: The adversary issues more queries q_{m+1}, q_{m+2}, ..., q_n adaptively as in Phase 1, except that the adversary may not request a private key for ID or the decryption of (ID, C).
- *Guess*: Finally, the adversary outputs a guess $b' \in \{0, 1\}$ and wins the game if $b' = b$.

We define the adversary A's advantage in attacking the IBE scheme as a function of the security parameter k, $\mathrm{Adv}_A(k) = |\Pr_A[b'=b] - 1/2|$, where the probability is over the random bits used by the challenger and the adversary. The most efficient identity-based encryption schemes are currently based on bilinear pairings on elliptic curves, such as the Weil or Tate pairings. Boneh and Franklin [7][8] were the first to give an IBE

scheme from Weil pairing and prove it to be adaptive chosen-ciphertext security in the random oracle model. More recently, several new IBE schemes from pairing (e.g., [9][10]) were proposed and proven to be adaptive chosen-ciphertext security in the standard model. A common feature of the latest IBE schemes is that the plaintext space is a cyclic group of prime order.

2. Identity-Based Signature

An identity-based signature (IBS) scheme can be described by four algorithms Setup, Extract, Sign, Verify as follows.

- Setup: On input a security parameter k, it returns params (public system parameters) and master-key (known only to the "Private Key Generator").
- Extract: Given params, master-key and a public identity ID $\in \{0, 1\}*$, it returns a private key d_{ID}.
- Sign: Given a message M, params, ID and a private key d_{ID}, it generates a signature σ of the user (with identity ID) on M.
- Verify: Given a signature σ, a message M, and *params*, ID, it outputs accept if σ is a valid signature of the user (with identity ID) on M, and outputs reject otherwise.

An IBS scheme is existential unforgeability under the chosen message attack [11] if no polynomial bounded adversary A has a non-negligible advantage against the challenger in the following game:

- *Initialize*: The challenger runs the Setup algorithm, gives params to the adversary, but keeps the *master-key* to itself.
- *Queries*: The adversary adaptively asks a number of different queries $q_1, q_2, ..., q_m$, where qi is either Extract(ID$_i$) or Sign(ID$_i$, M).
- *Forgery*: Once the adversary decides that queries are over, it outputs a message M', an identity ID′ and a string σ′. The adversary succeeds (denoted as Succ) if Verify(ID′, M', σ′) = 1, where ID′ cannot appear in Extract queries and (ID′, M') cannot appear in Sign queries.

We define the adversary A's advantage in attacking the IBS scheme as a function of the security parameter k, $Adv_A(k) = Pr_A[Succ]$, where the probability is over the random bits used by the challenger and the adversary.

A generic approach to construct IBS schemes is to use an ordinary (i.e., nonidentity-based) signature scheme and simply attach a certificate containing the public key of the signer to the signature [12]. An IBS scheme with provable security in the standard model was given by Paterson and Schuldt in [13].

3. Squaring Decisional Diffie-Hellman Problem

The squaring computational Diffie-Hellman (SCDH) problem in a cyclic group G with a prime order q and a generator g is: Given g, g^a where a is randomly chosen from Z_q, determine g^{a^2}. The problem is as hard as Diffie-Hellman problem [14][15][16].

The squaring decisional Diffie-Hellman (SDDH) problem in a cyclic group G with a prime order q and a generator g is to distinguish between two distributions (g, g^a, g^{a^2}) and (g, g^a, z), where a is randomly chosen from Z_q and z is randomly chosen from G. This problem is not harder than the decisional DH problem, but it is believed that this problem can still be hard, that is, we can assume that the advantage of any PPT algorithm A that outputs $b \in \{0,1\}$ in solving the SDDH problem is negligible, namely,

$$|Pr[A(g, g^a, g^{a^2}) = 0] - Pr[A(g, g^a, z) = 0]|$$

is negligible, where the probability is over the random choice of a in Z_q and z in G, and the random bits consumed by A.

III. Yi et al.'s Group Key Exchange

This section reviews Yi et al.'s password-authenticated GKE protocol PGKE [3]. There are three kinds of entities involved in PGKE: (1) a set of n clients C1,...,Cn who wish to establish a common session key; (2) a server S who provides the clients with a centralized

authentication service; (3) a private key generator PKG who generates global system parameters as well as S's long-term private keys. Both S and PKG are trusted to behave in an "honest but curious" manner; that is, S and PKG may attempt to learn the session key only by passive eavesdropping.

Building Blocks. The cryptographic building blocks of PGKE include:

- *a group key exchange protocol* GKE which is secure against a passive adversary. Every message of GKE is assumed to be sent – via point-to-point links – to all protocol participants. This assumption implies that in GKE, the set of all messages sent and received by each participant is expected to be the same.
- *an identity-based encryption scheme* IBE which is secure against an adaptive chosen ciphertext attack. We let Ecrt and Dcrt be the encryption and decryption algorithms of IBE. The plaintext space of IBE is $\mathbf{M} = \{0,1\}^{\tau}$ for some τ.
- *an identity-based signature scheme* IBS which is existentially unforgeable under an adaptive chosen message attack. We let Sign and Vrfy be the signing and verification algorithms of IBS.

Initialization. Before the protocol PGKE is ever executed, the following initialization is performed to generate public parameters and long-term secrets.

- Public parameters. PKG chooses: (1) a large cyclic group G of prime order q and a generator g of G and (2) two collision-resistant hash functions H1 : $\{0,1\}^* \rightarrow M$ and H2 : $\{0,1\}^* \rightarrow \{0,1\}^{\lambda}$. (Here, λ is the security parameter that determines the length of session identifiers constructed during protocol runs.) This is in addition to generating any public parameters needed for GKE, IBE and IBS.
- Long-term secrets. The server S obtains from PKG its private decryption/signing keys (DK_S, SK_S) corresponding to its public key ID_S. (Here, the public key ID_S is an arbitrary identity of S, and is used both for encryption and verification purposes.) Each client C_i chooses a password pw_i and stores it on the server S.

Protocol Execution. If the protocol GKE takes r rounds of communications, then the protocol PGKE runs in r+2 rounds.

[Round 1 ~ r]: The clients $C_1,...,C_n$ execute the protocol GKE. Let k_i be the key computed by C_i as a result of the execution of GKE. Let sidi be the (ordered) concatenation of all messages sent and received by C_i during the course of the execution.

[Round r+1]: Each client Ci computes $SIDi = H_2(g^{k_i}|sid_i)$ and sets $PID_i = (C_1,...,C_n,S)$. Then C_i computes

$$Auth_i = \mathrm{Ecrt}_{IDS}(H1(SID_i|PID_i|pw_i))$$

and sends the message $M_i = C_i|SID_i|Auth_i$ to the server S. Upon receiving all of $M_1,...,M_n$, the server S sets $SID_S = SID_1$ and $PID_S = (C_1,...,C_n,S)$ and checks that the following equation holds for all $i = 1,...,n$:

$$\mathrm{Dcrt}_{DKS}(Auth_i) = H_1(SID_S|PID_S|pw_i).$$

If any of the checks fails, S terminates the protocol execution.

[Round r+2]: S generates a signature

$$Auth_S = \mathrm{Sign}_{SKS}(PID_S|SID_S)$$

and broadcasts the message $M_S = S|Auth_S$. After receiving M_S, each client C_i checks that

$$\mathrm{Vrfy}_{IDS}(PID_i|SID_i, Auth_S) = 1.$$

If the verification fails, C_i aborts the protocol. Otherwise, C_i computes the session key $K_i = g^{k_i^2}$.

IV. Security Analysis

Resistance against dictionary attacks is the fundamental security requirement that should be satisfied by any password-based protocols for authenticated key exchange. However, the PGKE protocol described above fails to meet the

requirement. In this section, we reveal this security problem with PGKE and then suggest a countermeasure to the attack.

1. Dictionary Attack

Consider an adversary A whose goal is to find out the password of client Ci. Then, the following describes a dictionary attack mounted by A to achieve its goal.

1. As the $(r+1)^{th}$ round of PGKE proceeds, A eavesdrops on the message $M_i = C_i|SID_i|Auth_i$ sent from C_i to S.
2. A next makes a guess pw'_i for the password pw_i and computes $Auth'i = \mathrm{Ecrt}_{IDS}(H_1(SID_i|PID_i|pw'_i))$.
3. A then verifies the correctness of pw'_i by checking that $Auth'_i$ is equal to $Auth_i$. If pw'_i and pw_i are equal, then the equality $Auth'_i = Auth_i$ ought to be satisfied.
4. A repeats steps 2 and 3 until a correct password is found.

This dictionary attack may lead to devastating losses of passwords because: (1) it can be mounted against any of the clients and (2) the steps for verifying password guesses can be performed in an off-line manner by an automated program.

Of course, there is a possibility in the dictionary attack that the adversary A comes up with a password guess pw'_i such that $pw'_i \neq pw_i$ but $H_1(SID_i|PID_i|pw'_i) = H_1(SID_i|PID_i|pw_i)$ and thus $Auth'_i = Auth_i$. However, this possibility should be negligible because otherwise H_1 is not collision-resistant.

2. Countermeasure

The security failure of PGKE is attributed to one obvious flaw in the protocol design: the password pw_i is the only secret included in the computation of $Auth_i = \mathrm{Ecrt}_{IDS}(H_1(SID_i|PID_i|pw_i))$. SID_i can be obtained directly from the message M_i since it is transmitted in the clear. PID_i represents the identities of protocol participants and is generally assumed to be available to the adversary. (However, this assumption is not necessary for our dictionary attack if we think of the adversary A as a malicious client C_j ($\neq C_i$) who also is a protocol participant.) On the basis of this observation, one may suggest that a simple defense against the attack is to transmit SID_i in an encrypted form. This suggestion, of course, is valid if the adversary A does not know the key k_i from which SID_i can be derived. However, notice that A could be any (malicious) client C_j who runs the protocol with client C_i. Hiding SID_i from the public makes no difference to such an inside adversary.

As the discussion above highlights, a proper defense to the dictionary attack must ensure that the password of a client should not be disclosed even to other clients participating in the same protocol run. Keeping this in mind, we recommend to change the $(r+1)^{th}$ round of PGKE as follows:

[Round r+1] (revision): Each client C_i chooses a random $xi \in \{0,1\}\tau$, computes $X_i = \mathrm{Ecrt}_{IDS}(x_i)$ and $SID_i = H2(g^{k_i}|sid_i)$, and sets $PID_i = (C_1,...,C_n,S)$. Then C_i computes

$$Auth_i = \mathrm{Ecrt}_{IDS}(H_1(SID_i|PID_i|pw_i|x_i))$$

and sends the message $M_i = C_i|SID_i|X_i|Auth_i$ to the server S. After receiving the messages $M_1,...,M_n$, the server S sets $SID_S = SID_1$ and $PID_S = (C_1,...,C_n,S)$ and checks that $H_1(SID_S|PID_S|pw_i|\mathrm{Dcrt}_{DKS}(X_i))$ is equal to $\mathrm{Dcrt}_{DKS}(Authi)$ for all $i = 1,...,n$. If any of the checks fails, S terminates the protocol execution.

The other rounds of the protocol remain unchanged. The key change made in our revision is the inclusion of the confounder xi into the computation of $Auth_i$. This change prevents $Auth_i$ from being used as a password verifier. Hence, the dictionary attack is no longer valid against the improved protocol. In Table 1, we compare security properties between the improved Yi et al.'s protocol and the improved EKE-M protocol [17].

V. Conclusion

We have shown that Yi et al.'s password-authenticated group key exchange protocol is vulnerable to an offline dictionary attack

and thus does not guarantee password security. We have also shown that the security vulnerability of Yi et al.'s protocol can be eliminated by slightly modifying the way of generating the messages of clients. Our work highlights again the necessity that active adversaries are to be considered carefully in designing a key exchange protocol, especially when the protocol is password-based authenticated.

Table 1. Protocol Comparison

	Improved Yi et al.'s protocol	Improved EKE-M
Generic construction	yes	no
Off-line dictionary attacks	secure	secure
Undetectable on-line dictionary attacks	secure	vulnerable
Unknown key share attacks	secure	vulnerable
Perfect forward secrecy	provides	provides
Known key security	provides	provides

참고문헌

[1] Twitter, http://twitter.com

[2] Facebook, http://www.facebook.com

[3] X. Yi, R. Tso, E. Okamoto, "ID-Based group password-authenticated key exchange," Advances in Information and Computer Security - 4th International Workshop on Security, LNCS vol. 5824, pp. 192-211, 2009.

[4] J. Byun, D. Lee, "N-party encrypted Diffie-Hellman key exchange using different passwords," in Proceedings of 3rd International Conference on Applied Cryptography and Network Security, LNCS vol. 3531, pp. 75-90, 2005.

[5] J. Byun, S. Lee, D. Lee, D. Hong, "Constant-round password-based group key generation for multi-layer ad-hoc networks," in Proceedings of 3rd International Conference on Security in Pervasive Computing, LNCS vol. 3934, pp. 3-17, 2006.

[6] J. Kwon, I. Jeong, K. Sakurai, D. Lee, "Password-authenticated multi-party key exchange with different passwords," Cryptology ePrint Archive, Report 2006/476, 2006.

[7] D. Boneh, M. Franklin, "Identity-based encryption from the weil pairing," in Proceedings of Crypto'01, LNCS vol. 2139, pp. 213 - 229, 2001.

[8] D. Boneh, M. Franklin, "Identity based encryption from the Weil pairing," SIAM Journal of Computing, vol. 32, no. 3, pp. 586 - 615, 2003.

[9] B. Waters, "Efficient identity-based encryption without random oracles," in Proceedings of Eurocrypt'05. LNCS vol. 3494, pp. 114 - 127, 2005.

[10] C. Gentry, "Practical identity-based encryption without random oracles," in Proceedings of Eurocrypt'06, LNCS vol. 4004, pp. 445 - 464, 2006.

[11] S. Goldwasser, S. Micali, R. Rivest, "A digital signature scheme secure against adaptive chosen-message attack," SIAM Journal of Computing, vol. 17, no. 2, pp. 281 - 308, 1988.

[12] D. Galindo, J. Herranz, E. Kiltz, "On the generic construction of identity-based signatures with additional properties," in Proceedings of Asiacrypt'06, LNCS vol. 4284, pp. 178 - 193, 2006.

[13] K. Paterson, J. Schuldt, "Efficient identity-based signatures secure in the standard model," in Proceedings of Acisp'06, LNCS vol. 4058, pp. 207 - 222, 2006.

[14] U. Maurer, S. Wolf, "Diffie-Hellman oracles," in Proceedings of Crypto'96, LNCS vol. 1109, pp. 268 - 282, 1996.

[15] M. Burmester, Y. Desmedt, J. Seberry, "Equitable key escrow with limited time span," in Proceedings of Asiacrypt'98, LNCS vol. 1514, pp. 380 - 391, 1998.

[16] F. Bao, R. Deng, H. Zhu, "Variations of Diffie-Hellman problem," in Proceedings of ICICS'03, LNCS vol. 2836,

pp. 301 - 312, 2003.
[17] J. Byun, D. Lee, J. Lim, “Password-based group key exchange secure against insider guessing attacks,” in Proceedings of 2005 International Conference on Computational Intelligence and Security, LNAI vol. 3802, pp. 143-148, 2005.

저 자 소 개

이 영 숙

1987: 성균관대학교 정보공학과 공학사.
2005: 성균관대학교 정보보호학과 공학석사.
2008: 성균관대학교 컴퓨터공학과 공학박사.
2010~2011.6 호원대학교 기획조정처 경영평가 실장
현 재: 호원대학교 사이버수사경찰학부 학부장
관심분야: 암호프로토콜, 네트워크 보안, 스마트폰 보안, 디지털포렌식
Email : ysooklee@howon.ac.kr

김 지 연

1995: 성균관대학교 정보공학과 공학사.
1997: 성균관대학교 정보공학과 공학석사.
2006: 성균관대학교 전기전자및컴퓨터공학과 공학박사.
현 재: KISA, ISMS, PIMS 인증 심사원
관심분야: 암호프로토콜, 암호이론, 정보보호관리체계 인증
Email : jeeyeonkim@paran.com

원 동 호

1976~1988: 성균관대학교 전자공학과 공학사, 석사, 박사.
1978~1980: 한국전자통신연구원 전임연구원.
1985~1986: 일본 동경공업대 객원연구원.
1988~2003: 성균관대학교 교학처장, 전기전자 및 컴퓨터공학부장, 정보통신대학원장, 정보통신기술연구소장, 연구처장.
1996~1998: 국무총리실 정보화추진위원회 자문위원.
2002~2003: 한국정보보호학회장.
2002~2008: 대검찰청 컴퓨터범죄수사 자문위원, 감사원 IT감사자문위원.
현 재: 성균관대학교 정보통신공학부 교수, BK21 사업단장, 한국정보보호학회 명예회장.
관심분야: 암호이론, 정보이론, 정보보호
Email : dhwon@security.re.kr