Security Extension for Bresson-Chevassut-Pointcheval's model

Chunjie Cao †, Jianfeng Ma †, and Sangjae Moon ††

† Key Laboratory of Computer Networks and Information Security(Ministry of Education), Xidian University, Xi'an 71007, China

†† Mobile Network Security Technology Research Center, Kyungpook National University, Daeyu 702-701 Korea

Summary
The Bresson-Chevassut-Pointcheval (BCP) model is a formalism for the analysis of authenticated group key exchange protocols. Also there are some desired security goals for a practical group key exchange protocol, which are necessary in achieving resistance to active attacks mounted by an increasingly powerful adversary. However, whether a proved secure protocol in the BCP model can meet these security goals remains unknown. Firstly, the relationship between the BCP model and the desired security goals is analyzed in this paper. And it is shown that a protocol proved authenticated key exchange (AKE) security in the BCP model can surely achieve some security goals such as key independence, resistance to all types of passive attacks, Perfect Forward Secrecy (PFS) and implicit key authentication, but can not provide key integrity and known-key security. It is the lack of group key consistency in the definition of AKE security that causes the security flaws. Then, we present new definition of group key (GK) security, and show that a proved GK secure protocol can guarantee all the desired security goals.

Key words:
Provable security, Distributed computing, Formal model, Multicast security

1. Introduction

Group authenticated key exchange (AKE) protocols are designed to provide a group of player communication over an open network with a shared secret key which may later be used to guarantee the confidentiality and integrity of multicast data. Protocols of group AKE are essential for application such as secure video- or tele-conferencing, and also for collaborative and distributed computing applications. Over the years, a number of works [1-7] have examined the problem of extending the two-party Diffie-Hellman protocol [8] to the multiparty setting. A class of generic n-party Diffie-Hellman protocols is defined in [6] and extended to provide implicit key authentication in [1], one practical protocol of which is A-GDH.2. A tree-based Diffie-Hellman group key agreement protocol has been proposed by Kim et al. in [5] which is shown to be secure against passive adversaries.

Provable security is one of the most frequently used methodologies in designing and analyzing group AKE protocols. A protocol is proved secure when it is broken as hard as a hard cryptographic problem solved, such as factoring, the discrete logarithm, the Diffie-Hellman problem, etc. For practical protocols it seems to be the best achievable level of security. A challenge in this approach is to establish all possible attacks which have been taken into account, and this can in fact be equated to solve the identified reference problems. Only recently have Bresson, Chevassut and Pointcheval (BCP) [9, 10] given the first provably secure protocol and model for the dynamic group AKE setting. Previously, the protocol and model for static group AKE setting is presented in [11] by Bresson, et al. The BCP model follows the approach of Bellare and Rogaway [12-14] and has been widely used to analyze group key exchange protocols. It is an important step and is very helpful in analyzing and designing group key exchange protocols.

The main motivation for our work is to show that whether a group AKE protocol proved secure under the BCP model can meet the desirable security goals (such as key independence, implicit key authentication, resistance to all types of passive attacks, perfect forward secrecy and so on) or not. The conclusions indicate that a proved secure protocol under BCP model can provide implicit key authentication, perfect forward secrecy (PFS), key independence and resistance to all types of passive attacks, but it can not provide key integrity and known-key security. The reason is the inappropriate definitions of AKE security. Finally, we give the new definition of security which is referred as group key (GK) security and prove that a GK secure protocol can guarantee all the goals except key confirmation which is not absolutely necessary.

The rest of our paper is organized as follows. First we review the BCP model in Section 2 and the desirable security goals for group authenticated key exchange protocols in Section 3. Next, in Section 4, we give the security analysis of the security model and redefine freshness and AKE security. Finally, we conclude the paper in Section 5.

2. The BCP Model

The BCP model is the first formal model of provable security for authenticated group key exchange and has

Manuscript revised August 22, 2005.

been widely used to analyze group key exchange protocols. The model described in this section is the standard one of [9].

2.1 Adversarial Model

Let $U = \{U_1, U_2, \ldots U_n\}$ be a set of n users. Each user can execute the protocol multiple times with different partners: this is modeled by allowing each user an unlimited number of *instances* with which to execute the protocol. We denote instance t of U_i, called an oracle, as $\prod_i^t$ for an integer $t \in Z$.

Queries. Normally, the security of a protocol is related to the adversary's ability, which is formally modeled by queries issued by the adversary. We assume that a probabilistic polynomial time (PPT) adversary A can completely control the communications and make queries to any instance. We now explain the capability that each kind of query captures.

***Setup* (*U*), *Remove* (Ψ, *U*), *Join* (Ψ, *U*)**: These queries model the abilities of adversary A to initiate a new group, remove some members from U and enable some members to join in U respectively. Note that these queries can be simulated by *Send* queries.

***Execute* (Ψ)**: This query models passive attacks in which the adversary eavesdrops an execution of the protocol. A gets back the complete transcripts of an honest execution among the users in group Ψ. The number of group members is chosen by the adversary. (Although the *Execute* query can be simulated via repeated *Send* queries, the adversary is allowed to make this query for tighter distinguishing passive attacks from active attacks.)

***Send* ($\prod_i^t$, *M*)**: This query allows the adversary to make the user U_i run the protocol normally and send message M to instance $\prod_i^t$ which will return a reply.

***Reveal* ($\prod_i^t$)**: This query models the adversary's ability to find session group keys. If an oracle has accepted, holding a session key K, then K is returned to the adversary. Note that we say that an oracle accepts when it has enough information to compute a session key. At any time an oracle can accept and it accepts at most once in executing an operation. As soon as an oracle accepts in executing an operation, the session key is defined.

***Corrupt* (U_i)**: This query models the attacks revealing the long-term private key S_i. This does not output any internal data of U_i.

***Test* ($\prod_i^t$)**: This query models the semantic security of a session key. This query is allowed only once by the adversary A. A random bit b is chosen; if $b = 1$ then the session key is returned, otherwise a random value is returned.

In this model we consider two types of adversaries according to their attack types. The attack types are simulated by the queries issued by the adversaries. A passive adversary is allowed to issue *Execute*, *Reveal*, *Corrupt*, and *Test* queries, while an active adversary is additionally allowed to issue *Send* queries.

2.2 Security Notions

Definition 1: Partner IDS. Partner identities for instance $\prod_i^t$ which consists of the users (including U_i himself) with whom $\prod_i^t$ intends to establish a session key. The Partner IDS of instance $\prod_i^t$ is denoted by PID ($\prod_i^t$)**.**

Definition 2: Session IDS. The Session IDS is a protocol specified function of all communication sent and received by $\prod_i^t$, which is denoted by SID ($\prod_i^t$).

Definition 3: Freshness. An oracle is called fresh (or holds a fresh key) if the following two conditions are satisfied. First, nobody in U has ever been asked for a *Corrupt* query from the beginning of the game. Second, in the current operation execution, $\prod_i^t$ has accepted and neither U_i nor his partners have been asked for a *Reveal* query.

Definition 4: Authenticated Key Exchange (AKE) security. In an execution of protocol P, We say that the event *Succ* occurs if the adversary asks a single *Test* query to a fresh oracle and correctly guesses the bit b. The advantage of an adversary A in attacking protocol P is defined as

$$Adv_A^P(k) = | 2 \cdot Pr[Succ] - 1 |.$$

A protocol P is a secure AKE protocol if for any PPT adversary A, $Adv_A^P(k)$ is negligible. Note that A function $\varepsilon(k)$ is negligible if for every $c>0$ there exists a $k_c>0$ such that for all $k>k_c$, $\varepsilon(k)<k^{-}$.

Definition 5: Mutual Authenticated (MA) security. In an execution of P, we say adversary A violates mutual authentication if there exists an operation execution wherein a player U_i terminates holding $SID(\prod_i^t)$, $PID(\prod_i^t)$ and $|PID(\prod_i^t)| \neq |\Psi| - 1$**,** where Ψ is the multicast group. We denote the MA *success* as $Succ_P^{MA}(A)$ and say protocol P is MA security if $Succ_P^{MA}(A)$ is negligible.

3. Security Goals for Group Key Exchange Protocols

The primary motivation for obtaining a group key is the ability to communicate securely and efficiently once a key is established. If all group members share a key, they can communicate using symmetric encryption. This is more efficient than schemes not requiring key establishment. To satisfy the requirements of security group communication, we need a key exchange protocol providing[1, 5, 15]:

Key Independence. An adversary who knows a set of group keys cannot discover any other group keys. Informally, this means that old keys cannot be known to new members and new keys cannot be known to former members.

Resistance to all types of passive attacks. A passive

attack involves an adversary who attempts to defeat a cryptographic technique *by simply recording data and thereafter analyzing it*.

(Implicit) Key Authentication. If each member in U is assured that no other party aside from U may gain access to a particular secret key, implicit key authentication is independent of the actual possession of such key by all parties, and in fact, it needs not involve any action whatsoever by other parties. For this reason, it is usually referred to more precisely as implicit key authentication.

Perfect Forward Secrecy (PFS). Loss of a long-term key does not compromise the semantic security of previously distributed session keys.

Resistance to known-key attacks. A protocol is said to be vulnerable to a known-key attack if compromise of past session keys allows either a passive adversary to compromise future session keys, or impersonation by an active adversary in the future.

Key Integrity. A party is assured that its particular secret key is a function of only the individual contributions of all protocol parties. In particular, extraneous contributions to the group key cannot be tolerated even if it does not afford the attackers with any additional knowledge. Key integrity is orthogonal to both key authentication and key confirmation. A key exchange protocol may offer one or both of the latter while not guarantee key integrity at the same time.

Key Confirmation. A party is assured that its peer (or a group of peers) actually has possession of a particular secret key.

Explicit key authentication. We say a group key exchange protocol satisfies explicit key authentication if implicit key authentication and key confirmation are both provided. Note that key confirmation and explicit key authentication are not absolutely necessary.

All of these are necessary to achieve resistance to active attacks mounted by an increasingly powerful adversary. Any group key exchange protocol should meet the security goals for group key exchange no matter what designing and analyzing methodologies are used.

4. Security Analysis of the BCP Model

4.1 Description of Security Goals in the BCP Model

Before the security analysis of the BCP model, we define a basic game and then specialize in it for the purpose of modeling each single security goal, which a group key exchange protocol should satisfy. In the basic game, a simulator S that on parameters n (number of users) and security parameter l is described. Each game is an adversarial setting that determines the capabilities and possible actions of the attacker. We follow the general formalism of [9, 11]. Without any loss of generality, we assume that there are many concurrent executions of protocol P.

Game 0: Basic Game

– S chooses the initialization information and sets the long-term keys for each of the n users according to the security parameter l.

– On any activation by A the simulator S performs the operations of P on behalf of users. The answers to the oracle queries made by A as explained below.

Send **($\prod_i^s$, *m*)**: S answers the request on behalf of U_i. If needed in P, S signs the response flow with the long term key assigned to U_i. (S realizes *Setup, Join* and *Leave* queries by *Send* queries in this game.)

Execute **(Ψ)**: S returns all the messages in the honest execution of group Ψ, where $\Psi \subseteq U$.

Reveal$(\prod_i^s)$: S answers the request with the session key of *SID* ($\prod_i^s$).

Corrupt **(U_i)**: S returns the long-term key of U_i.

Test **($\prod_i^s$)**: S chooses a random bit b; if $b = 1$ then the session key is returned, otherwise a random value is returned.

– At the end of the game, adversary A outputs a bit b'.

Based on Game 0, the security goals are defined by the following games between the adversary A and an infinite set of oracles $\prod_i^s$ for $U_i \in U$ and $s \in N$.

Game 1: Key Independence. This game simulates a real setting in which the adversary may issue any attacks that can obtain part of the group session keys. Then with this knowledge the adversary attempts to successfully compute the group key.

The game is similar to Game 0, but the queries, which the adversary can make, are *Execute*, *Reveal* and *Test* queries. If the advantage $Adv_A^P(k)$ involved in the Test query is negligible, we say an AKE secure protocol can provide key independence.

Game 2: Resistance to all types of passive attacks. This game defines an adversarial setting in which all possible passive attacks (eavesdrop) can be issued to the group key exchange protocol.

The game is similar to Game 0, but the queries, which the adversary can make, are *Execute* and *Test* queries. If the advantage $Adv_A^P(k)$ involved in the *Test* query is negligible, we say an AKE secure protocol can be resistant to all types of passive attacks.

Game 3: Perfect Forward Secrecy (PFS). This game allows the adversary to obtain all long-term keys of group members, and then she tries to compromise the group (session) key with the help of the knowledge of these long-term keys.

The game is similar to Game 0, but the queries, which the adversary can make, are *Execute*, *Corrupt* and *Test* queries. If the advantage $Adv_A^P(k)$ involved in the *Test*

query is negligible, we say an AKE secure protocol can provide PFS.

Game 4: Resistance to known-key attacks. This game simulates the adversarial setting in which any type attacks to obtain past group keys can be issued, and then the adversary uses these group keys to compromise the current group session key or impersonate one group member.

The game is similar to Game 0, but the queries, which the adversary can make, are *Send*, *Execute*, *Reveal* and *Test* queries. We say an AKE secure protocol can resistant to known-key attacks if both of the following are true:

- The advantage $Adv_A^P(k)$ involved in the *Test* query is negligible.
- The adversary fails to impersonate a group member with only past session keys, so all the legitimate members can accept a same group key.

Game 5: Key Integrity. In this game, the adversary tries to forge a legitimate contribution of the group key to compromise the integrity of the group key. So she must have the ability to break the integrity-protection mechanism.

The game is similar to Game 0, but the queries, which the adversary can make, are *Execute*, *Send* and *Test* queries. We say an AKE secure protocol can provide key integrity if both of the following are true:

- The advantage $Adv_A^P(k)$ from the forgery of a contribution is negligible.
- All the legitimate members can accept a same group key.

Game 6: Key Confirmation. This game shows that impersonating a member in MA rounds implies for the adversary to fake an authentication value.

The game is similar to Game 0, but the queries, which the adversary can make, are *Execute* and *Send* queries. We say an AKE secure protocol can provide key confirmation if all participants accept a same group session key at the end of the protocol.

Game 7: Implicit Key Authentication. The goal of the adversary in this game is to obtain the secret through modifying, delaying or injecting messages. Moreover, we assume that any value computed as group session key, by each party, is kept secret.

The game is similar to Game 0, but the queries, which the adversary can make, are *Execute*, *Send* and *Test* queries. If the advantage $Adv_A^P(k)$ involved in the *Test* query is negligible, we say an AKE secure protocol can provide implicit key authentication.

Game 8: Explicit Key Authentication. This property is strong and can be obtained when both implicit key authentication and key confirmation hold. The adversary in this game has to impersonate a group member to compromise the consistency and secrecy of the group key.

The game is similar to Game 0, but the queries, which the adversary can make, are *Execute*, *Send* and *Test* queries. We say an AKE secure protocol can resistant to known-key attacks if both of the following are true:

- All the legitimate members accept a same group key.
- The advantage $Adv_A^P(k)$ involved in the *Test* query is negligible.

4.2 Security Analysis of the BCP Model

Theorem 1. A group key exchange protocol which is proved AKE secure in the BCP model can provide key independence, implicit key authentication, PFS and resistance to all types of passive attacks.

Proof:

Game 1: Because the key independence is that the adversary who knows a set of group keys cannot discover any other group keys, she is allowed to issue *Reveal* query but not *Corrupt* and *Send* queries. We assume that there are many concurrent executions of the group key exchange protocol in network setting, so there also exist many completed but not invalidated sessions at some time. So the adversary A can make *Reveal* query many times to obtain part of the session keys and then she can test any fresh oracle with the help of the knowledge of session keys obtained previously. At the end of the game the adversary outputs a guess. By the definition of AKE security, we know that for every PPT adversary A, the advantage $Adv_A^P(k)$ is negligible. So the protocol proved secure in the security model can guarantee that an adversary who knows a proper subset of group keys cannot discover any other group keys.

Game 2: The adversary who makes passive attacks simply records data and thereafter analyzes it, so she is only allowed to issue *Execute* and *Test* queries. To achieve her attack, the adversary issues an *Execute* query which helps her get the full session transcript. Then she analyzes the transcript to gain the knowledge of current session key. At some stage the adversary issues a *Test* query to a fresh oracle and eventually outputs the guess. If the adversary wins the game, then the advantage $Adv_A^P(k)$ of A successfully guess the bit b is not negligible. From the definition of AKE security, we know if a group key exchange protocol is proved security in the BCP model, then the advantage $Adv_A^P(k)$ is negligible. The contradistinction shows that a protocol, which is proved to be AKE secure in BCP model, can resist all types of passive attacks.

Game 3: The adversary makes any attacks which can obtain one or more (even all) long-term secret keys of group members, and then tries to compromise the current

group (session) key with the help of the knowledge of these long-term keys. So the adversary is only allowed to issue *Execute*, *Corrupt* and *Test* queries. To obtain these long-term secret keys, she makes *Corrupt* queries. At some stage the adversary issues a *Test* query to a fresh oracle. She may continue to make other queries, eventually outputs the guess *b'* for the bit *b* involved in the *Test* query and terminates. The adversary wins the game if the advantage $Adv_A^P(k)$ is not negligible.

The above reduction procedure of BCP model seems perfect, but it is flawed. Let us recall the definition of *freshness*, which is that no body in *U* has ever been asked for a *Corrupt* query from the beginning of the game and no *Reveal* query has been issued in the current execution of protocol *P*. From the definition, we can learn that there is no chance for the adversary to make *Corrupt* queries *before* she ask a *Test* query to a fresh oracle. To compromise the forward secrecy of the group key, the adversary must corrupt some group members to get long-term secret keys which can help her in guessing *b*. So, the adversary has to corrupt some users *after* the *Test* query has been issued. We can see that the ability of adversary is limited here and has not considered enough. Nevertheless, a protocol proved AKE secure in the BCP model can provide PFS.

Game 7: Implicit key authentication is independent of the actual possession of such key by all parties. In fact, it needs not involve any action whatsoever by other parties. The adversary wants to obtain the secret key by modifying, delaying or injecting messages while not knowing long-term keys or past session keys. So the queries the adversary can make are *Execute*, *Send* and *Test* queries. At some stage she issues a *Test* query to a fresh oracle and eventually outputs the guess. We know, to an AKE secure protocol, the advantage $Adv_A^P(k)$ is negligible. Therefore the implicit key authentication is guaranteed. This completes the proof of Theorem 1. #

Theorem 2. The group key exchange protocol proved AKE secure in the BCP model *can not* provide known-key security and key integrity.

Proof:

Game 4: To achieve this type of attacks, the adversary should have known some past group keys. So she issues *Reveal* queries to past sessions and then tries to compromise the current group key or impersonates one group member. We know that the adversary who impersonates one group member should be allowed the ability to make *Send* query. Therefore an adversary who attacks the known-key security is allowed to ask *Execute, Reveal*, *Send* and *Test* queries.

For the first condition of the adversary winning Game 4, the adversary issues a *Test* query to a fresh oracle and eventually outputs her guess. By the definition of AKE security, we know the advantage $Adv_A^P(k)$ is negligible. So this attack can be prevented. For the second condition, the adversary issues *Send* query to impersonate one group member only by past session keys. As a result, all the legitimate members cannot accept a same group key because impersonation occurs. For example, although the adversary cannot get non-negligible advantage in guessing the bit *b*, she can impersonate U_i. Hence all members in *U* do not accept a same session key at the end of the protocol. Unfortunately, the AKE security cannot guarantee this. As shown in [16], a proved secure protocol is vulnerable to known-key attack.

Game 5: To compromise the integrity of a group key, the adversary tries to produce a legitimate contribution to the group key. She may get some additional knowledge about the group key or not. So she is allowed the ability to ask *Execute*, *Send* and *Test* query. If a protocol is proved AKE secure in BCP model, then the adversary cannot get any advantage in distinguishing a random value from a real key. But it cannot assure that the forgery events do not occur (as shown in game 4). Furthermore, all legitimate members cannot share a same group key. So AKE security cannot satisfy the second condition in Game 5. This completes the proof of Theorem 2. #

Theorem 3. The group key exchange protocol proved AKE secure in the BCP model *can not* provide key confirmation, and explicit key authentication.

Proof:

By[9, 11], we know that an AKE secure protocol does not contain the MA rounds, so it apparently cannot provide key confirmation and also the explicit key authentication. This completes the proof of Theorem 3. #

A protocol with mutual authentication is transformed from an AKE protocol by adding MA rounds. Then an MA secure protocol is also AKE secure. So we focus on the properties with respect to mutual authentication in the MA rounds. Obviously, the properties involved in mutual authentication are key confirmation and explicit key authentication.

Theorem 4. The group key exchange protocol proved MA secure in the security model can provide key confirmation and explicit key authentication.

Proof:

Game 6: The way of the adversary to attack the property of key confirmation in MA rounds is faking a valid authentication value to impersonate a legitimate group member. She achieves this without the knowledge of past session keys or the long-term secret keys. So she is allowed to ask *Execute* and *Send* query. In the game, the adversary can make any type of attacks to forge a valid authentication transcript, which results in the fact that all the participants cannot share a same group session key at the end of the protocol. By the definition of MA security, the probability $Succ_P^{MA}(A)$ is negligible. This means that

the forge event will not happen in an MA secure protocol. So the MA security can guarantee a protocol provides key confirmation.

Game 8: To break the explicit key authentication of a protocol, the adversary should have the ability to impersonate a legitimate group member without knowing past session keys and long-term keys. If she succeeds in impersonation, the consistency and secrecy of the group key are compromised. Namely, the explicit key authentication is broken. So the adversary is allowed to ask *Execute*, *Send* and *Test* queries in this game.

Then, the adversary issues attacks to the protocol by eavesdropping, modifying, delaying and injecting messages sent among the group members. Firstly, we assume that all group members do not accept a same group key. This implies that there is one or more members which have not correctly executed the protocol. In other words, the adversary successfully impersonates a group member at least. So the probability $Succ_P^{MA}(A)$ is not negligible. This contradicts the definition of MA security. Secondly, we know that the MA security implies the AKE security, then the probability $Adv_A^P(k)$ is also negligible. Therefore, the MA security can guarantee a group key exchange protocol providing explicit key authentication. This completes the proof of Theorem 4. #

4.3 Result Analysis

A protocol proved AKE security in the BCP model can satisfy the security goals involved in secrecy, such as key independence, PFS, implicit key authentication and resistance to all types of passive attacks. By the definitions and games with respect to these goals, we know that the purpose of the adversary is to get non-negligible advantage in correctly guessing the bit b. But the AKE security shows that the advantage $Adv_A^P(k)$ is negligible. So the AKE security can guarantee the secrecy of the group key.

On the other hand, a proved AKE security in the BCP model cannot guarantee the security goals involved in forgery, such as key integrity and known-key security. The adversary attempts to forge a legitimate message, which results in the fact that all the members cannot share a common group key. Although the adversary can not get non-negligible advantage, the protocol fails to achieve its function yet. So, the definition of AKE security only shows the secrecy of group key but not the consistency of group key of each member. In addition, known-key security and key integrity indicate the secrecy of the group key.

From the analysis above, we introduce the following definitions,

Definition 6: Consistency. In the presence of an adversary, all the partner oracles accept the same group key.

Definition 7: Secrecy. When the adversary asks a single *Test* query to a fresh oracle in an execution of P, the advantage $Adv_A^P(k)$ in correctly guessing the bit b is negligible.

And then the following corollaries:

Corollary 1. The secrecy of the group key implies key independence, PFS, implicit key authentication and resistance to all types of passive attacks.

Corollary 2. The secrecy and consistency of the group key imply key integrity and known-key security.

These two corollaries can be directly deduced from the analysis above.

4.4 Extension for the BCP Model

To enable the BCP model to deal with known-key security, key integrity and PFS better, we redefine the freshness and AKE security which are denoted as freshness and group key security respectively. The new definitions are as follows:

Definition 8: Freshness. An oracle $\prod_i^t$ is called fresh (or holds a fresh key) if the following two conditions are satisfied:

- Nobody in current group Ψ has been asked for a *Corrupt* query before $\prod_i^t$ accepts.
- In the current operation execution, $\prod_U^t$ has accepted and neither $\prod_U^t$ nor his partners have been asked for a *Reveal* query.

As shown in Game 3, the adversary can make *Corrupt* queries at the time after $\prod_U^t$ has accepted and before she issues the *Test* query to $\prod_U^t$. This simulates the nature execution of the protocol more realistically.

Definition 9: Group Key (GK) security. We say that the event *Succ* occurs if the adversary issues *Test* query to a fresh oracle and correctly guesses the bit b. The advantage of an adversary A in attacking protocol P is defined as

$$Adv_A^P(k) = | 2 \cdot Pr[Succ] - 1 |.$$

A protocol P is GK security, if the following two properties are satisfied:

- Consistency: In the presence of an adversary, all the partner oracles accept the same key.
- Secrecy: In the presence of an adversary, $Adv_A^P(k)$ is negligible.

We introduce the consistency of group key into the definition of GK security which makes up the defect of the AKE security. The following theorem shows that the GK security does work.

Theorem 5. The group key exchange protocol proved GK security in the BCP model can provide all the security goals but key confirmation.

Proof: By corollary 1 and corollary 2, the consistency and secrecy of the group key imply key independence, resistance to all types of passive attacks, PFS, known-key security, implicit key authentication and key integrity; so does the GK security.

We know the GK security is independent of MA rounds, so it obviously cannot guarantee that a group key exchange protocol provides key confirmation.

Therefore, a GK secure protocol can provide all the security goals but key confirmation. This completes the proof of Theorem 5. #

5. Conclusion

In this paper, we have presented a security analysis of the relationship between the security model proposed by Bresson, et al and the desired security goals that a group key exchange protocol should satisfy. Firstly, we have shown that a proved authenticated key exchange (AKE) secure protocol in the BCP model can guarantee the implicit key authentication, key independence, PFS, resistance to all types of passive attacks but can not provide key integrity and known-key security. And the reasons have been indicated, which is the lack of consistency of group key to resist forgery in AKE security. So we introduce the new definition of group key (GK) security. Then we show that an authenticated group key exchange proved GK secure in the BCP model can guarantee all the desired security goals but key confirmation which is not absolutely necessary. In addition, we have proven that the MA security implies key confirmation and explicit key authentication.

Acknowledgments

The research was supported by the National Natural Science Foundation of China (Grant No. 90204012, 60503012), the National “863” High-tech Project of China (Grant No. 2002AA143021), the Excellent Young Teachers Program of Chinese Ministry of Education, the Key Project of Chinese Ministry of Education, and the University IT Research Center Project of Korea.

References

[1] G. Ateniese, M. Steiner and G. Tsudik, "New multiparty authentication services and key agreement protocols," IEEE Journal of Selected Areas in Communications, vol.18, pp.1-13, 2000.

[2] M. Burmester and Y. G. Desmedt, "A secure and efficient conference key distribution system," in Proc. of Eurocrypt'94, Springer-Verlag, LNCS vol.950, pp.275-286, 1994.

[3] I. Ingemarsson, D.T. Tang and C.K. Wong, "A conference key distribution system," IEEE Transactions on Information Theory, vol.28, pp.714-720, 1982.

[4] Y. Kim, A. Perrig and G. Tsudik, "Simple and fault-tolerant key agreement for dynamic collaborative groups," in ACM CCS'00, pp.235-244, 2000.

[5] Y. Kim, A. Perrig and G. Tsudik, "Tree-based group key agreement," ACM Transactions on Information and System Security, vol.7, pp.60-96, 2004.

[6] M. Steiner, G. Tsudik and M. Waidner, "Diffie-Hellman key distribution extended to groups," in ACM CCS'96, pp.31-37, 1996.

[7] M. Steiner, G. Tsudik and M. Waidner, "Key agreement in dynamic peer groups," IEEE Transactions on Parallel and Distributed Systems, vol.11, pp.769-780, 2000.

[8] W. Diffie and M. E. Hellman, "New directions in cryptography," IEEE Transactions on Information Theory, vol.22, pp.644-654, 1976.

[9] E. Bresson, O. Chevassut and D. Pointcheval, "Provably authenticated group Diffie-Hellman key exchange -the dynamic case," in Proc. of Asiacrypt'01, Spinger-Verlag LNCS vol.2248, pp.290-309, 2001.

[10] E. Bresson, O. Chevassut and D. Pointcheval, "Dynamic group Diffie-Hellman key exchange under standard assumptions," in Proc. of Eurocrypt'02, Spinger-Verlag, LNCS vol.2332, pp.321-336, 2002.

[11] E. Bresson, O. Chevassut, D. Pointcheval and J. Quisquater, "Provably authenticated group Diffie-Hellman key exchange," in ACM CCS'01, pp.255-264, 2001.

[12] M. Bellare and P. Rogaway, "Random oracles are practical: a paradigm for designing efficient protocols," in ACM CCS'93, pp.62-73, 1993.

[13] M. Bellare and P. Rogaway, "Entity authentication and key distribution," in Proc. of Crypto'93, Spinger-Verlag, LNCS vol. 773, pp.232-249, 1993.

[14] M. Bellare and P. Rogaway, "Provably-secure session key distribution: the three party case," in Proc. of STOC'95, pp.57-66, 1995.

[15] A. Menezes, P. van Oorschot and S. Vanstone,: Handbook of applied cryptography. CRC Press series on discrete mathematics and its applications, 1997.

[16] J. Nam, S. Kim and D. Won, "A weakness in the Bresson-Chevassut-Essiari-Pointcheval's group key agreement scheme for low-power mobile devices," IEEE Communication Letters, vol.9, pp.429-431, 2005.

Chunjie Cao received the B.E and M.E. degrees in computer science and technology from Xidian University in 2001 and 2004 respectively. Currently, he is pursuing his Ph.D. degree in computers with Their Applications at Xidian University. His research interests include Information security and Cryptography.

Jianfeng Ma received the B.E. degree in mathematics from Shaaxi Normal University (Xi'an) in 1985, and obtained his M.E. and Ph.D. degrees in computer software and communications engineering from Xidian University (Xi'an) in 1988 and 1995 respectively. Since 1995 he has been with Xidian University as a lecturer, associate professor and professor. From 1999 to 2001, he was with Nanyang Technological University of Singapore as a research fellow. Currently, Prof. Ma is the director of the Ministry of Education Key Laboratory of Computer Networks and Information Security, and he is the dean of the school of computer of Xidian University. His research interests include information security, coding theory and cryptography.

Sangjae Moon received the B.E. (1972) and M.E. (1974) degrees in electronic engineering from Seoul National University, Korea, and the PhD (1984) degree in communication engineering from the Department of Electrical Engineering of the University of California, Los Angeles. He is currently a professor in the School of Electronic, Electrical and Computer Science, Kyungpook National University, Korea. He was president of the Korea Institute of Information Security and Cryptology from February 2001 to January 2002. Currently, he is the director of the Mobile Network Security Research Center, which is financially supported by the Ministry of Information and Communication, Korea. His research interests currently are in the areas of cryptography, network security, and security applications.