Math-Net.Ru

Общероссийский математический портал

T. Q. Phong, N. Q. Toan, K. X. Thanh, Доказательство защищенности протокола обмена ключами, *Матем. вопр. криптогр.*, 2016, том 7, выпуск 4, 117–132

DOI: <https://doi.org/10.4213/mvk207>

Использование Общероссийского математического портала Math-Net.Ru подразумевает, что вы прочитали и согласны с пользовательским соглашением
<http://www.mathnet.ru/rus/agreement>

Параметры загрузки:

IP: 3.95.254.165

26 апреля 2023 г., 10:41:25

A security proof for key exchange protocols

T. Q. Phong, N. Q. Toan, K. X. Thanh

Institute of Cryptography Science and Technology, Government Information Security Committee,
Viet Nam

Получено 10.VI.2016

Abstract. We present a technique of modular security proofs for key exchange protocols in the modified Bellare–Rogaway (mBR) model introduced by C. Kudla and K. G. Paterson. We have applied this technique to modify Protocol 1 in their paper to Protocol 2 and analyze its security in this model. These two protocols differ only in the combination of the ephemeral and long-term private keys in the computation of the session key. Our method for proving strong partnering of Protocol may be applied to prove a same result (stated without proof) by C. Kudla and K. G. Paterson.

Keywords: modular method, security proof, strong partnering, mBR model, Bellare–Rogaway model, Canetti–Krawczyk model

Доказательство защищенности протокола обмена ключами

Т. К. Фонг, Т. К. Тоан, К. К. Танх

*Институт криптографических наук и технологий, Государственный комитет
защиты информации, Вьетнам*

Аннотация. Предлагается модулярный метод доказательства защищенности протоколов обмена ключами в модифицированной модели Беллара–Рогавея, введенной К. Кудла и К. Г. Патерсоном. Этот метод использован для преобразования Протокола 1 из их статьи в Протокол 2 и анализа его защищенности в этой модели. Протоколы 1 и 2 различаются только комбинацией разового и постоянного секретных ключей при вычислении сеансового ключа. Наш метод доказательства того, что Протокол обеспечивает сильное партнерство, можно применить при доказательстве аналогичного утверждения К. Кудла и К. Г. Патерсона.

Ключевые слова: модулярный метод, доказательство защищенности, сильное партнерство, модель Беллара–Рогавея, модель Канетти–Кравчука

Citation: *Mathematical Aspects of Cryptography*, 2016, v. 7, № 4, pp. 117–132 (Russian)

© Академия криптографии Российской Федерации, 2016 г.

1. Introduction

The security proof for key exchange protocols has been known as hard problem. The first formal concepts of security for key exchange protocols were introduced by Bellare and Rogaway (the BR93 model). A “modular” method for constructing the key exchange protocols was developed by Bellare, Canetti and Krawczyk. The modified security models and new ones introduced later include the BR95 [5], BPR2000 [2] and Canetti-Krawczyk (CK) [6] models.

The modular method requires constructing a secure protocol in the “authenticated links” model, and then using “authenticators” to all the protocol flows to obtain a secure protocol in the standard “unauthenticated links” model.

Related works

In [1], C. Kudla and K. G. Paterson considered protocols which are not designed in a modular way but which we nevertheless wish to prove secure, because of their efficiency. These two authors introduced a technique using the hardness of a Gap problem by which the proof process of a large class of key exchange protocols can be simplified and then they presented a security proof using their technique to a modified version of Protocol 4 in [4].

In [7], C. Boyd and coworker also used the modular method to prove the security of key exchange protocol in CK model.

Moreover in [8], P. Morrissey, N. P. Smart and B. Warinschi provided a same approach as that of C. Kudla and K. G. Paterson to prove the security of SSL/TLS protocol.

Organization

This paper includes five Sections: Section 1 introduces the background and the related works on the security proof for key exchange protocols. Section 2 provides the necessary preliminaries and the techniques of C. Kudla and K. G. Paterson for proving the security of key exchange protocols. In Section 3, we will present results on Protocol 1 in [1]. Section 4 is our main contribution: we present a slightly modified version of Protocol 1 and apply the above technique to analyze the security of the modified protocol. Section 5 is on our future research. Finally, the conclusion is presented in Section 6.

2. Preliminaries

2.1. Mathematical background

We informally recall a family of Gap problems presented in [1].

Let $f: X \times Y \rightarrow \{0, 1\}^k$ be any relation on sets X and Y . The computational problem of f is, given $x \in X$, to compute any $y \in Y$ such that $f(x, y) = 1$ if such y exists, or to return **Fail** otherwise.

The decisional problem of f is, given $(x, y) \in X \times Y$, to decide whether $f(x, y) = 1$ or not.

Definition 1 (Definition 1, [1]). The Gap problem of f is to solve the computational problem of f using an oracle which solves the decisional problem of f .

In addition, the computational Diffie-Hellman (CDH), decisional Diffie-Hellman (DDH) and gap Diffie-Hellman (GDH) problems are also defined in [1] via a relation.

2.2. The modified Bellare-Rogaway model

We now start by describing a modified Bellare-Rogaway (mBR) model for authenticated key exchange protocols defined in [1], this model is closely related to the BPR200 model in [2]. The mBR model contains a set $\{U\}$ of distinct participant IDs and uses parameters listed below,

P_U	The long-term public key of participant U .
S_U	The long-term private key of participant U .
Π_U^i	The i -th session at participant U and considered as an oracle.
$role$	The <i>initiator</i> or <i>responder</i> role of a session.
$pid_{\Pi_U^i}$	The ID of the oracle which Π_U^i is communicating.
$sid_{\Pi_U^i}$	The session ID of Π_U^i .
$sk_{\Pi_U^i}$	The session key of oracle Π_U^i .

Oracles follow the rules of the protocol, responding to input messages (from the adversary). Each oracle Π_U^i maintains a public transcript $T_{\Pi_U^i}$ which records all messages they have sent or received as a result of queries they have answered. Besides, Π_U^i may accept at any time, and once accepted it should hold a tuple $(role, pid_{\Pi_U^i}, sid_{\Pi_U^i}, sk_{\Pi_U^i})$ as above.

2.2.1. The mBR game

The security of a key exchange protocol is modelled via the game between a challenger $\mathcal{C}$ and an adversary $\mathcal{E}$ described in [1] as follows.

The challenger $\mathcal{C}$ runs a Setup algorithm on a security parameter k to create: the public parameters, a set of participants $\{U\}$ and oracles Π_U^i to model instances of each participant U , and to distribute long-term keys to each participant. Also $\mathcal{C}$ randomly selects a bit b .

The adversary $\mathcal{E}$ is given all the participants' public keys and has access to all the participants' oracles as well as any random oracles in the game. $\mathcal{E}$ can make the following queries:

- **Send** (U, i, M) : E can send a message M to the oracle Π_U^i . If Π_U^i has $pid = U'$, then Π_U^i assumes that M has come from U' and respond according to the protocol. E may also make a special Send query λ to an oracle Π_U^i which instructs U to initiate a protocol run with its partner U' . An oracle Π_U^i sets $role_U = initiator$ and is called an *initiator oracle* if the first message it has received is λ . If Π_U^i did not receive a message λ as its first message, then it sets $role_U = responder$ and is called a *responder oracle*.
- **Reveal** (U, i) : this allows E to ask the oracle Π_U^i to reveal the session key (if any) it recently holds to E .
- **Corrupt** (U) : this allows E to ask participant U to reveal its long-term private key.

An oracle exists in one of the following possible states:

- **Accepted**: an oracle has accepted if it decides to accept, holding a session key, after receipt of properly formulated messages.
- **Rejected**: an oracle has rejected if it decides not to establish a session key and to abort the protocol.
- **State***: an oracle is in state* if it has not made any decision to accept or reject.
- **Revealed**: an oracle is revealed if it has answered a **Reveal query**.
- **Corrupted**: an oracle is corrupted if it has answered a **Corrupt query**.

Definition 2 (Partner, [1]) . When running the protocol, if oracles Π_U^i holding $(role_U, sk, sid, pid)$ and $\Pi_{U'}^j$ holding $(role_{U'}, sk', sid', pid')$ have both accepted and the following conditions hold:

1. $sid = sid', sk = sk', pid = U'$ and $pid' = U$.
2. $role_U = initiator$ and $role_{U'} = responder$ or vice versa.
3. No oracle in E 's game besides Π_U^i and $\Pi_{U'}^j$ accepts with session ID equal to sid .

Then Π_U^i and $\Pi_{U'}^j$ are said to be partners.

- **Freshness.** An oracle is called unfresh if it is “revealed”, or it has a “revealed” partner, or if its partner was “corrupted”. If an oracle is not unfresh, then it is fresh.
- **Test Query.** E may make a polynomial number of queries in k . Then at any point E makes a special Test query to an oracle Π_U^i . This oracle must be “accepted” and “fresh”, and it answers as follows. If $b = 0$ then Π_U^i randomly chooses a session key sk and outputs it, otherwise if $b = 1$ then this oracle output the its real session key sk_U^i .

After this point E can continue querying the oracles, except that E can not reveal or corrupt the test oracle or its partner (if it exists). Finally E outputs a guess b' for b .

E 's advantage in the mBR game, denoted $Advantage_{mBR}^E(k)$, is computed as follows:

$$Advantage_{mBR}^E(k) = |Pr[b' = b] - 1/2|.$$

2.2.2. The cNR-mBR game

We now consider a reduced mBR game defined in [1]. This game is identical to the mBR game defined in Section 2.2.1 except that the adversary E is not allowed to make any Reveal and Test queries. Instead, to win the game, the adversary must select an accepted and fresh Test oracle at the end of its computation and output the session key for this oracle. Since the adversary in this game is not allowed to make any Reveal queries and it must actually compute the session key of an oracle, so this game is called a *computational No-Reveals mBR (cNR-mBR)* game. Besides, E 's advantage in this game is also defined, it is probability that E outputs a session key sk such that $sk = sk_{\Pi_U^i}$ where Π_U^i is the Test oracle selected by the adversary. This advantage is denoted by $Advantage_{cNR-mBR}^E(k)$.

2.2.3. The security definitions

The definition of (cNR-)mBR security includes a benign adversary defined in [3]. Informally, a benign adversary is one who simply relays messages between parties without modification. Then, authenticated key exchange (AKE) protocol is defined **(cNR-)mBR-secure** as follows.

Definition 3 (Definitions 2 and 4, [1]). A protocol is an (cNR-)mBR-secure AKE protocol if:

1. In the presence of the benign adversary, two oracles running the protocol both accept holding the same session key and session ID, and the session key is distributed uniformly at random on $\{0, 1\}^k$.
2. For any adversary E , $Advantage_{(cNR-)mBR}^E(k)$ is negligible.

2.3. Modular construction of security proofs

In this section we present the modular construction of security proofs described in [1]. In particular, we will work with key exchange protocols that produce hashed session key on completion of the protocol. In this way, the session key in the key exchange protocol Π will be computed as the hash H of some string which is called the session string ss_{Π} . Each session string for an oracle Π_U^i is denoted by $ss_{\Pi_U^i}$. Here, H is modelled as an random oracle.

2.3.1. Protocol Partnering

Definition 4 (Definition 3, [1]). Suppose Π is a key exchange protocol. If there exists an adversary E , which when attacking Π in an mBR game defined in Session 2.2.1 and with non-negligible probability in the security parameter k can make any two Π_U^i and $\Pi_{U'}^j$, accept holding the same session key when they are not partners, then we say that Π has **weak partnering**. If Π does not have weak partnering, then we say that Π has **strong partnering**.

In [1], in order to guarantee that Π have strong partnering, C. Kudla and K. G. Paterson have included $role_U^i, sid_U^i$ and pid_U^i in the session string $ss_{\Pi_U^i}$.

2.3.2. Handling reveal queries using Gap assumptions

As part of their proof process for a given protocol Π which produces hashed session keys on completion of the protocol, we will consider a related protocol π . Protocol π is identical to Π except that $sk_{\pi_U^i} = ss_{\Pi_U^i}$.

We now consider a protocol Π which has strong partnering and is such that the related protocol π is cNR-mBR secure. In order to translate these results into a proof of mBR security for Π , we need a challenger $\mathcal{C}$ in an mBR game for Π which is able to answer an adversary E 's Reveal queries. In fact, according to Theorem 2 in [1] we only need the challenger $\mathcal{C}$ that is able to solve the following decisional problem:

“Given the transcript T_U^i of oracle Π_U^i in an mBR game, as well as P_U and $P_{U'}$ (the long-term public keys of U and U' where $pid_U^i = U'$) and a string s , decide whether $s = ss_{\Pi_U^i}$, where $ss_{\Pi_U^i}$ is the session string of oracle Π_U^i ”.

This decisional problem is called the session string decisional problem for protocol Π .

We now recall Theorem 2 in [1], this is an important result for designing and proving a secure protocol in the mBR model.

Theorem 5 (Theorem 2, [1]). *Suppose that key exchange protocol Π produces a hashed session key on completion of the protocol (via hash function H) and that Π has strong partnering. If the cNR-mBR security of the related protocol π is probabilistic polynomial time reducible to the hardness of the computational problem of some relation f , and the session string decisional problem for Π is polynomial time reducible to the decisional problem of f , then the mBR security of Π is probabilistic polynomial time reducible to the hardness of the Gap problem of f , assuming that H is a random oracle.*

3. The Main results for Protocol 1 in [1]

According to Theorem 5, we can see that the technique of C.Kudla and K.G.Paterson can be summarized as follows. In order to prove a protocol Π is secure using the hardness of GDH assumption, we need to prove that this protocol has strong partnering. After that, the cNR-mBR security of the related protocol Π being probabilistic polynomial time reducible to the hardness of the CDH problem also must be specified. Finally, we must prove that the appropriate decisional oracle can be used to solve the session string decisional problem of Π (in general, this is a trivial reduction).

We now present the main results for Protocol 1 in [1].

Protocol 1 in [1]

The Setup algorithm generates primes p and q where $q|(p-1)$. It then chooses G to be a multiplicative subgroup of $\mathbb{Z}_p^*$, where G has order q , and a generator $g \in G$. This algorithm also sets the session ID space $S = G^4$ and select a hash function

$$H: G^2 \times S \rightarrow \{0, 1\}^k.$$

Each participant I selects a long-term private key x_I randomly from $\mathbb{Z}_q$ and sets the long-term public key to be

$$X_I = g^{x_I} \bmod p.$$

Fig. 1. Protocol 1 in [1]

Suppose that A and B are participants with long-term public keys

$$X_A = g^{x_A} \bmod p$$

and

$$X_B = g^{x_B} \bmod p$$

respectively. A and B run the protocol as in Fig. 1.

It is easy to see that if Protocol 1 completes correctly, then

$$sk_A = sk_B = H(g^{ax_B} \bmod p, g^{bx_A} \bmod p, X_A, X_B, T_A, T_B).$$

Let Protocol 1' be the related protocol for Protocol 1, then Protocol 1' has the following property.

Theorem 6 (Theorem 3, [1]). *The cNR-mBR security of Protocol 1' is probabilistic polynomial time reducible to the hardness of the CDH problem in G .*

In [1] C. Kudla and K. G. Paterson had stated without proof that Protocol 1 has strong partnering.

Theorem 7 (Theorem 4, [1]). *Protocol 1 has strong partnering in the random oracle model.*

8Corollary 1, [1] Protocol 1 is secure in the random oracle model assuming the hardness of the Gap Diffie–Hellman problem.

4. A slight modification in Protocol 1

Fig. 2. Protocol 2

It is easy to see that if Protocol 2 completes correctly, then

$$sk_A = sk_B = H(g^{ab} \bmod p, g^{x_A x_B} \bmod p, X_A, X_B, T_A, T_B).$$

By using the similar way to design and analyze the security of Protocol 1, we constructed and analyzed the security of Protocol 2 modified from Protocol 1 in the mBR model. These two protocols are identical except a small modification in the computation of the session key. Protocol 2 is illustrated in Fig. 2.

Let Protocol 2' be the related protocol for Protocol 2. We obtain a same result as that for Protocol 1'.

Theorem 9. *The cNR-mBR security of Protocol 2' is probabilistic polynomial time reducible to the hardness of the CDH problem in G .*

Proof. Since Protocol 2 is a slight modification of Protocol 1, so in the proof of this theorem we may use the same idea as in Theorem 3 in [1] with some modifications.

First, assume that for the security parameter k there exists an adversary E for Protocol 2' who can win the cNR-mBR game in time τ which is polynomial in k with advantage η which is non-negligible in k . Suppose that the number of participants is n_P , and n_S is the maximal number of sessions in which each participant may be involved, where n_P and n_S are polynomial functions of k .

We now construct from E an algorithm F which solves the CDH problem in G with non-negligible probability. That is, given $g^x, g^y \in G$ as input elements, we need $\mathcal{F}$ to compute and produce the value $g^{xy} \bmod p$.

Thus, we need to show how an adversary E breaking the mBR-security of Protocol 2' can be used to construct the algorithm $\mathcal{F}$ which can solve the CDH problem in G . Note that the session key of this protocol has form

$$(g^{ab} \bmod p, g^{x_A x_B} \bmod p, X_A, X_B, T_A, T_B)$$

(here a, x_A, T_A, X_A are the ephemeral private, long-term private, ephemeral public, long-term public key of some participant A , respectively; similarly, b, x_B, T_B, X_B are the ephemeral private, long-term private, ephemeral public, long-term public key of some participant B , respectively). We can see that in the session key of this protocol only has the Diffie-Hellman values of the ephemeral keys and the long-term public keys, so if we assign g^y as an ephemeral public key of some participant and g^x as a long-term public key of another (same as in the proof of Theorem 3 in [1]), it seems that $\mathcal{F}$ does not have any advantage to compute $g^{xy} \bmod p$ from the session key which E outputs.

Therefore, in our proof, we need to construct the following $\mathcal{F}$ that differs from the algorithm used to solve the CDH problem in the proof of Theorem 3 in [1].

First, $\mathcal{F}$ simulates a challenger in a cNR-mBR game with E (the game which $\mathcal{F}$ sets up with group G and generator $g \in G$). Next, $\mathcal{F}$ generate a set of participants of size n_P . For each participant I , $\mathcal{F}$ sets I 's long-term private key to be a randomly chosen $x_I \in_R \mathbb{Z}_q$ and sets the long-term public key to be

$$X_I = g^{x_I} \bmod p.$$

After that, $\mathcal{F}$ also picks two random participants $P \neq Q$, and two numbers $i, j \in \{1, \dots, n_S\}$. Finally, $\mathcal{F}$ starts E and answers E 's queries as follows.

- **Send:** E may make a special Send query Π_I^s which sets $pid_I = X_{I'}$ and instructs I to initiate a protocol run with its partner I' . Further, E can also send any oracle Π_I^s a message M , and the oracle responds according to the protocol. However if E initializes or sends a message to oracle Π_P^i , then Π_P^i outputs g^x . And if E initializes or sends a message to oracle Π_Q^j , then Π_Q^j outputs g^y .
- **Corrupt** (U) : $\mathcal{F}$ can answer all of the Corrupt queries, because $\mathcal{F}$ has all of the long-term private keys.

Because P, Q are randomly selected from the set of participants of size n_P and i, j are randomly chosen from the set $\{1, 2, \dots, n_S\}$, so the probability that E queries oracle Π_Q^j for the Test session is equal to

$$\frac{1}{n_P \cdot n_S}$$

and the probability that Π_P^i to be partner of Π_Q^j is equal to

$$\frac{1}{(n_P - 1) \cdot n_S}$$

(eliminate n_S sessions at participant Q). Therefore, the probability that E chooses one of two participants (Π_Q^j and Π_P^i) for the Test session and the other to be the partner of Test session is

$$\frac{2}{(n_P - 1) \cdot n_P \cdot n_S^2}.$$

Finally, E outputs a session key of the form (a, b, c) where $a, b \in G$ and $c \in G^4$. Therefore $\mathcal{F}$ outputs a as its guess for $g^{xy} \bmod p$. It is easy to check that $\mathcal{F}$ solves the CDH problem on input (g^x, g^y) with probability

$$\eta' = \frac{2\eta}{(n_P - 1) \cdot n_P \cdot n_S^2}$$

(non-negligible in k) in polynomial time τ . $\square$

Remark 1. We provide a short comparison between the success probability for reducing the mBR security of Protocol 1' (according to the method of C.Kudla and K.G.Paterson) and Protocol 2' (our way) to the hardness of CDH problem via Table 1.

	Protocol 1'	Protocol 2' (our modification)	Comment
The probability that $\mathcal{F}$ solves CDH on input (g^x, g^y) (where η is probability that E breaks the mBR-security)	$\frac{\eta}{n_P^2 \cdot n_S}$	$\frac{2\eta}{(n_P - 1)n_P \cdot n_S^2}$	According to these probabilities, the ability for reducing the mBR-security of Protocol 1' to the hardness of CDH is greater than that of Protocol 2'

Table 1. The comparison of reduction probability.

Theorem 10. *Protocol 2 has strong partnering in the random oracle model.*

We will prove this theorem in Appendix A.

It is easy to see that a decisional Diffie–Hellman oracle can be used to solve the session string decisional problem for Protocol 1. Therefore, applying Theorem 7, 9 and 10, we obtain the immediate result as follows.

Corollary 11. *Protocol 2 is secure in the random oracle model assuming the hardness of the Gap Diffie–Hellman problem.*

5. Our future research

In the mBR model, it was considered only the model for secure authenticated key exchange protocols, that is, the exchange of messages between participants in the protocol is performed by some benign adversary (always delivers messages correctly between participants). Hence, the first requirement (produces the same session key of two accepted oracles) in the definition of an mBR secure protocol is always obvious if the protocol is correct. However, in CK model, the definition of security is not bounded by the authenticated world, but also accounts for the unauthenticated world (the delivery of messages between participants is executed by an un-benign adversary who can add or modify messages to send).

Then, the requirement for producing the same session key of a protocol in the definition of security in the unauthenticated world (in CK model) is not trivial in general and should be carefully checked.

By the above reason we consider an extension of mBR model with the presence of the un-benign adversary as in the CK model. Therefore, it is necessary to make some modifications for the protocol to be proved to be secure in the mBR model but without effecting the second requirement in Definition 3.

We will consider a proposal that suggest to include MAC in the message flows as the idea of construction of Protocol 1 in [4]. In particular, the modification will transform Protocol 1 into Protocol 3 (see Fig. 3).

Fig. 3. Protocol 3

The inclusion of MAC in the message flows in general does not effect the structure of Protocol 1. Thus, Protocol 3 is not only to guarantee the second requirement in the definition of mBR security, but also supply the information that supports participants ability to check the validity of the received message flows.

In our knowlegde, the security of Protocol 3 may be proved by combining the technique for proving the security presented in [1] and the technique for proving the security based on the proof of Theorem 8 in [4]. This will be our future study together with considering the combination between the digital signature and the message authentication code (SIGN and MAC) of key exchange protocol SIGMA in the CK model.

In [9] Stanislav V. Smyshlyaev and his coworkers used the approach of Bellare and Rogaway to produce the adversary model for the password-based key exchange protocol SESPake (this protocol is approved in the standardization system of the Russian Federation by TC26). And, another our future work is to study the security proofs for SESPake protocol.

6. Conclusion

In the technique of C. Kudla and K. G. Paterson, strong partnering of a protocol is very important. However, in their security proof for the modified version of Protocol 4 in [4], they skipped proving strong partnering for this protocol. In this paper we proved strong partnering for the protocol mentioned above. In addition, we applied their technique to modify Protocol 3 in [4] to Protocol 2 in our paper and then prove the security for this protocol. Moreover, according to Remark 1, we can see that, in the design of key exchange protocol, a small modification in the computation of session key may imply changes in the security level of the protocol.

Although our work just stops at the technical application of C. Kudla and K. G. Paterson to work with a specific example, it also helps us understand more about this technique. We hope that, in the future, we will be able to learn more about this technique to get more tools for demonstrating the security of key exchange protocols, as well as to obtain new results.

Список литературы

- [1] Kudla C., Paterson K. G., “Modular security proofs for key exchange protocols”, *Lect. Notes Comput. Sci.*, **3788**, 2005, 549–565.
- [2] Bellare M., Pointcheval D., Rogaway P., “Authenticated key exchange secure against dictionary attacks”, *Lect. Notes Comput. Sci.*, **1807**, 2000, 139–155.
- [3] Bellare M., Rogaway P., “Entity authentication and key distribution”, *Lect. Notes Comput. Sci.*, **773**, 1994, 232–249.
- [4] Blake-Wilson S., Johnson D., Menezes A., “Key exchange protocols and their security analysis”, *Lect. Notes Comput. Sci.*, **1355**, 1997, 30–45.
- [5] Bellare M., Rogaway P., “Provably secure session key distribution: The three party case”. In: “*Proc. 27th ACM Symp. Theory of Computing — ACM STOC*”, N. Y.: ACM Press, 1995, 57–66.
- [6] Canetti R., Krawczyk H., “Analysis of key-exchange protocols and their use for building secure channels”, *Lect. Notes Comput. Sci.*, **2045**, 2001, 453–474.
- [7] Hitchcock Y., Boyd C., González-Nieto J., “Modular proofs for key exchange: Rigorous optimizations in the Canetti-Krawczyk model”, *Appl. Alg. in Eng., Commun. and Comput.*, **16**:6 (2006), 405–438.
- [8] Morrissey P., Smart N. P., Warinschi B., “The TLS handshake protocol: A modular analysis”, *J. Cryptology*, **23**:2 (2010), 187–223.
- [9] Smyshlyaev S. V., Oshkin I. B., Alekseev E. K., Ahmetzyanova L. R., “On the security of one password authenticated key exchange protocol” (2015), <https://eprint.iacr.org/2015/1237>.

Appendix A

Proof of Theorem 10. For Protocol 2 in the random oracle model, we will show that if two distinct oracles Π_A^i and Π_B^j both accept holding a same session key then they must be partners in the definition of Partner as above except with negligible probability. We will prove this statement via two following cases.

Case 1: $A \neq B$.

If Π_A^i and Π_B^j both accept holding a same session key then Π_A^i and Π_B^j must held (sk_A, sid_A, pid_A) and (sk_B, sid_B, pid_B) , respectively ($sk_A = sk_B$ from the assumption). In Protocol 2,

$$pid_A = X_{B'}, \quad sk_A = H(g^{ab'} \bmod p, g^{x_A x_{B'}} \bmod p, sid_A),$$

where B' is some participant, $x_{B'}$ and b' are the long-term and ephemeral private keys of B' , respectively. Similarly,

$$pid_B = X_{A'}, \quad sk_B = H(g^{a'b} \bmod p, g^{x_{A'} x_B} \bmod p, sid_B),$$

where A' is some participant, $x_{A'}$ and a' are the long-term and ephemeral private keys of A' , respectively. Since $sk_A = sk_B$, we have

$$H(g^{ab'} \bmod p, g^{x_A x_{B'}} \bmod p, sid_A) = H(g^{a'b} \bmod p, g^{x_{A'} x_B} \bmod p, sid_B).$$

On the other hand, H is a hash function which returns the random values, so the probability that H returns a same value for two different queries is negligible. Therefore, we obtain $sid_A = sid_B$ except with negligible probability (otherwise, it is easy to find two different queries that H returns the same value with non-negligible probability).

We next prove that $role_A \neq role_B$ except with negligible probability, i.e. if Π_A^i (or A) is the initiator then Π_B^j (or B) must be the responder or vice versa. Indeed, if we suppose that both A and B are the initiator with non-negligible probability, then according to the description of Protocol 2, $sid_A = X_A, X_{B'}, T_A, T_{B'}$ and $sid_B = X_B, X_{A'}, T_B, T_{A'}$ (here, $X_{A'}, X_{B'}$ are the long-term public key of A' and B' respectively; $T_{A'}$ and $T_{B'}$ are the ephemeral public key of A' and B' respectively); on the other hand, because A and B are two different entities, so $sid_A \neq sid_B$ with non-negligible probability. This contradicts the above result that $sid_A = sid_B$ with a probability being “very close” to 1. Therefore, it is impossible that both A and B are the initiator with non-negligible probability. Similarly, by comparing sid_A and sid_B according to the description of Protocol 2, we also

obtain that A and B are not both two responders. Hence, the second requirement of Definition 2 is satisfied.

As above, we obtained the second requirement of Definition 2. However, we don't have the third requirement, and in the first requirement it is remain to show that $pid_A = X_B$ and $pid_B = X_A$ except with negligible probability. Without loss of generality, assume that Π_A^i is the initiator and Π_B^j is the responder. Therefore, $sid_A = X_A, X_{B'}, T_A, T_{B'}$ and $sid_B = X_{A'}, X_B, T_{A'}, T_B$. Because the probability that $sid_A = sid_B$ is "very close" to 1, so $X_A = X_{A'}$ and $X_B = X_{B'}$ except with negligible probability. Note that, because of $pid_A = X_{B'}$ and $pid_B = X_{A'}$, we obtain $pid_A = X_B$ and $pid_B = X_A$ except with negligible probability. Therefore, we completely obtain the first requirement of Definition 2.

The remainder of the proof is to prove the third requirement of Definition 2. That is, we show that no oracle Π_C^t in E 's game besides Π_U^i and $\Pi_{U'}^j$ accepts with session ID equal to X_A, X_B, T_A, T_B with non-negligible probability (without loss of generality, assume that Π_A^i is the initiator and Π_B^j is the responder). Indeed, in otherwise, we can infer that $C \equiv A$ or $C \equiv B$ except with negligible probability. We consider two following cases.

- If $C \equiv A$ and $t \neq i$, then Π_A^t is the initiator of the session and accepts ID session X_A, X_B, T_A, T_B with the responder oracle Π_B^j . Hence, in the step of selecting the random value a^* of A for oracle Π_A^t , we must obtain $g^{a^*} = T_A = g^a \mod p$ (or $a^* = a$) with non-negligible probability. Note that a is also a random value of A but for oracle Π_A^i , so the probability that $a = a^*$ occurs, is negligible. Therefore, we get a contradiction.
- If $C \equiv B$ and $t \neq j$, then Π_B^t is the responder session and accepts ID session X_A, X_B, T_A, T_B with the initiator oracle Π_A^i . Hence, in the step of selecting the random value b^* of B for oracle Π_B^t , we must obtain $g^{b^*} = T_B = g^b \mod p$ (or $b^* = b$) with non-negligible probability. Note that b is also a random value of B but for oracle Π_B^j , so the probability that $b = b^*$ occurs is negligible. Therefore, we get a contradiction.

Hence, we have proved that no-oracle Π_C^t in E 's game besides Π_A^i and Π_B^j accepts with session ID equal to X_A, X_B, T_A, T_B .

Case 2: $A \equiv B$.

In this case, using again the property of the hash function H in random oracle model, we find that two oracles Π_A^i and Π_A^j both accept with same session ID ($i \neq j$). Using the same argument to satisfy the third requirement in Case 1, we obtain that Case 2 occurs with negligible probability.

Therefore, if two distinct oracles Π_A^i and Π_B^j both accept holding a same session key then they must be partners in the definition of Partner, except with negligible probability. $\square$

Remark 2. Strong partnering of Protocol 1 may be completely proved as that of Protocol 2. Indeed, in our proof of strong partnering of Protocol 2, we only use the argument that if two distinct oracles have the same *sid* then they must be partners, because of the structure of *sid* in this protocol. On the other hand, the inclusion *sid* of each oracle in the computation of session key gives us a result, that is, if two oracles Π_A^s and Π_B^t accept holding the same session key then they have the same *sid*. Hence, strong partnering of Protocol 2 is ensured. Although two above protocols have some differences, the structure of *sid* and inclusion *sid* of each oracle in the computation of session key are identical. Thus, strong partnering of Protocol 1 may be specified as that of Protocol 2. Therefore, Theorem 7 may be proved in the same way as Theorem 10.