Journal of Universal Computer Science, vol. 14, no. 3 (2008), 441-462
submitted: 2/6/07, accepted: 1/11/07, appeared: 1/2/08 © J.UCS

Formal Security Definition and Efficient Construction for Roaming with a Privacy-Preserving Extension

Guomin Yang, Duncan S. Wong, Xiaotie Deng
(Computer Science Department, City University of Hong Kong, Hong Kong
{csyanggm,duncan,deng}@cs.cityu.edu.hk)

Abstract: In a secure roaming scenario, a user U travels to a foreign network and communicates with a foreign server V securely so that no one other than U and V can obtain the messages exchanged between them. U may also want to travel anonymously so that no one including V can find out its identity or trace its whereabouts except its home server H. There have been many key establishment protocols proposed for secure roaming. A typical application of these protocols is the mobile roaming service which may be deployed to interconnected WLAN and 3G networks. Despite the importance of these protocols, most of the protocols are analyzed heuristically. They are lack of formal security treatment.

In this paper, we propose a formal key exchange definition and formalize secure roaming under the Canetti-Krawczyk (CK) model. We also propose a formal model for capturing the notions of user anonymity and untraceability. By using the modular approach supported by the CK-model, we construct an efficient key exchange protocol for roaming and then extend it to support user anonymity and untraceability. The protocols are efficient and each of them requires only four message flows among the three parties U, H and V. For building our protocols, we construct a one-pass counter based MT-authenticator and show its security under the assumption of a conventional MAC secure against chosen message attack.

Key Words: Authenticated Key Exchange, Anonymous Roaming
Category: C.2.2, H.4.3

1 Introduction

Secure key exchange protocols provide the basis of building secure communications using symmetric key cryptography. In 1993, Bellare and Rogaway [Bellare and Rogaway 1994] proposed the first formalized security model for provably secure key exchange protocols under the symmetric key setting. Their work was later extended to a three party case (in which the third party is a key distributor) and the asymmetric setting in [Bellare and Rogaway 1995, Blake-Wilson and Menezes 1997, Blake-Wilson et al. 1997]. In 1998, Bellare et al. proposed a different model which treats authentication and key exchange separately [Bellare et al. 1998]. One of the major advantages of this model is that it supports a modular approach to the construction of secure protocols. During the construction, some proven secure building blocks are used. These building blocks can also be reused for building other protocols. In 2001, Canetti and Krawczyk [Canetti and Krawczyk 2001] followed the work and refined it further. They also changed the definition of secure key exchange from the original simulation-based approach to an indistinguishability-based approach. In this paper, we call their model as the CK-model.

1.1 Secure Roaming

A traditional key exchange protocol has two parties [Bellare and Rogaway 1994] or three parties in a trusted third party setting [Bellare and Rogaway 1995]. With the rapid development of mobile technology, wireless networks have become widely available. People can travel around with their mobile devices without being limited by the geographical coverage of their home networks. This capability is called *roaming*. A typical roaming scenario involves three parties: *a roaming user U*, *a home server H*, and *a foreign server V*. During roaming, U, which is a legitimate subscriber of H, is now in a foreign network and wants to receive services provided by V. In a typical mobile roaming scenario, a mobile user U travels to a foreign network and gets access to V after being authenticated by V that H is indeed the home server of U. The authentication is carried out with the involvement of H. In addition to authentication, U and V also establish a secure channel, by carrying out a key exchange protocol, so that no one except U and V can obtain the messages exchanged between them. U may also want to make sure that the party it is communicating with is indeed V. Hence foreign server authentication may also need to be carried out between U and V. With all these security requirements satisfied, we call this type of roaming activities as *secure roaming*.

1.2 User Anonymity and Untraceability

Another important issue regarding the roaming scenario is user privacy. It concerns about hiding the roaming user's identity and movements from eavesdroppers and even the foreign servers. Informally, user anonymity means that besides U and H, no one including V can find out the identity of U, and user untraceability means that besides U and H, no one including V is able to identify any previous protocol runs which have U involved.

In cellular networks, GSM and 3GPP roaming protocols provide a certain degree of anonymity by using some temporary identity called TMSI (Temporary Mobile Subscriber Identity) rather than the real identity IMSI (International Mobile Subscriber Identity) for each roaming user. However, different sessions of the same user within one foreign domain can be easily linked by the foreign server. Therefore, user untraceability requirement is not satisfied.

There are many other roaming applications that require user privacy, for example, the inter-bank ATM networks and the credit card payment systems [Ateniese et al. 1994]. Ideally, a user should not reveal anything to the serving network (i.e. the foreign server) other than the confirmation of the user's good standing with respect to the user's ATM card or credit card issued by the user's home server. However, current systems are having users given out their personal information inevitably. Some other scenarios which require anonymous roaming include hopping across meshed WLANs (Wireless Local Area Networks) administered by different individuals, joining and leaving various wireless ad hoc networks operated by different foreign operators, etc. With the integration of WLANs and 3G cellular networks, anonymous and secure roaming will become more important in the near future.

1.3 Our Results

We consider a key exchange protocol for secure and anonymous roaming to be a protocol involving three parties: a user and two servers, namely a home server and a foreign server. In each successful protocol execution (i.e. all the parties accept the protocol execution), the following five properties will be attained.

1. **(Foreign Server Authentication)** The user is sure about the identity of the foreign server.

2. **(Subscription Validation)** The foreign server is sure about the identity of the home server of the user.

3. **(Key Establishment)** The user and the foreign server establish a random session key which is known only to them. In particular, the home server should not obtain the key.

4. **(User Anonymity)** Besides the user himself and his home server, no one including the foreign server can tell the identity of the user.

5. **(User Untraceability)** Besides the user himself and his home server, no one including the foreign server is able to identify any previous protocol runs which have the same user involved.

In subsequent sections, we first propose a key exchange definition and secure roaming formalization under the CK-model [Canetti and Krawczyk 2001]. Our definitions formalize the first three properties above. We then capture the last two properties (user anonymity and untraceability) by proposing a general security framework and introducing a security definition for them. We call a scheme satisfying all these properties as an **Anonymous and Authenticated Key Exchange for Roaming (AAKE-R)**.

We focus ourselves on constructing an AAKE-R scheme for a typical roaming scenario such that the user and the foreign server has a direct communication link and the foreign server and the home server has another direct link while the user and the home server do not have a direct link. Also the scheme involves all the three parties. This setting is one of the most common roaming scenarios in practice [Mouly and Pautet 1992, Mu and Varadharajan 1996, Ateniese et al. 1994, Samfat et al. 1995, Buttyan et al. 2000, Go and Kim 2001, Hwang and Chang 2003, 3GPP].

We adopt the approach of the CK-model [Canetti and Krawczyk 2001] and construct a key exchange protocol for secure roaming that satisfies the first three properties above. We then extend it to construct an AAKE-R protocol. The protocols not only are provably secure but also efficient. In each of the protocols, there are only four message flows and only standard cryptographic primitives are used. Hence efficient implementation is possible by choosing appropriate instantiations of the primitives to use. For building our protocols, we construct a one-pass counter based MT-authenticator and show its security under the assumption of having a MAC (Message Authentication Code) algorithm secure against chosen message attack.

Finally, we also present an anonymous authenticated key exchange (AAKE) protocol which is used by the mobile user to communicate with the home server in the home domain. Our AAKE protocol is an abbreviation of the AAKE-R protocol and it requires only two message flows.

1.4 Related Work

There have been many key establishment protocols proposed for secure roaming [Mouly and Pautet 1992, Ateniese et al. 1994, Mu and Varadharajan 1996, Samfat et al. 1995, Buttyan et al. 2000, Hwang and Chang 2003, 3GPP]. This type of protocols may potentially be adopted widely in emerging applications such as roaming over interconnected WLAN and 3G networks. Despite the importance of these protocols, most of the protocols are analyzed heuristically and lack of a formal security treatment. Also, there have been a number of work on secure roaming with user anonymity and untraceability [Ateniese et al. 1994, Samfat et al. 1995, Go and Kim 2001]. In [Samfat et al. 1995], there is a session key established in each protocol execution between a user and a foreign server. However, the key is also known to the user's home server. This is undesirable because when a roaming user is visiting a foreign server, services are actually provided by the foreign server to the user but not the home server. The home server is called in only as a guarantor for giving a promise that the user is indeed a legitimate subscriber of the home server. For example, in the WLAN Roaming, when a user accesses the Internet through a foreign server, the user may not want his home server to know which network sites he is visiting. Beside this undesirable feature, in all the protocols of [Samfat et al. 1995], the key value remains unchanged for all the sessions between the user and a particular foreign server. This allows the foreign server to trace the user easily. In [Go and Kim 2001], a protocol was designed for protecting a roaming user's identity from all entities other than his home server and the serving foreign server. However, according to results from [Wong 2005], a malicious server which is not communicating with the roaming user can launch an active attack to reveal the user's identity. In addition, it is recently found that both [Samfat et al. 1995] and [Go and Kim 2001] cannot provide *Subscription Validation* as both of them are vulnerable to the Deposit-Case Attack [Yang et al. 2005].

In the construction of our key exchange protocol for secure roaming under the CK-model, we propose and use a counter based MT-authenticator. Our counter based MT-authenticator is an improvement of the one proposed in [Bellare et al. 1998]. The counter based MT-authenticator in [Bellare et al. 1998] cannot be used in the bidirectional setting where both sides use the same shared key but each side has two different counters (one for sending and the other for receiving messages), because it is vulnerable to the reflection attack. Our improved counter based MT-authenticator removes this limitation.

Organization. In Sec. 2, we review the CK-model and introduce a new MT-authenticator called the one-pass counter based MT-authenticator. In Sec. 3, we propose a formal key exchange definition for secure roaming under the CK-model and construct a key exchange protocol that satisfies the first three properties of above. In Sec. 4, we propose a formal definition for user anonymity and untraceability. This is followed by the construction of an anonymous version of our protocol. We also present an anonymous authenticated key exchange (AAKE) protocol which is used by the mobile user to communicate with the home server in the home domain in Sec. 5. Finally, we conclude the paper in Sec. 6.

2 The Modular Approach and A One-pass Counter Based MT-authenticator

2.1 The CK-Model

In the Canetti-Krawczyk (CK) model [Canetti and Krawczyk 2001], there is a system of n parties denoted by P_1,...,P_n, which may carry out multiple concurrent executions of a message-driven protocol in an adversary controlled network. In each protocol execution, the adversary activates the protocol in some of the parties either by action requests which model invocations coming from other programs run by the parties or incoming messages which model messages coming from the network.

A key exchange (KE) protocol π is a message-driven protocol. Each execution of the KE protocol is modelled as a series of activations within two parties, P_i and P_j. For example, when activating the protocol within P_i, the program running in P_i will receive the identity of P_j (with whom the key is to be established), a session ID s and a *role* which can be either initiator or responder. The program of

P_i will then fork a sub-process and pass in $(P_i, P_j, s, role)$ to the sub-process. The sub-process will be responsible for handling that particular protocol execution. The forked sub-process is called a *KE-session* with input $(P_i, P_j, s, role)$. A KE-session is *completed* when the corresponding sub-process returns with output (P_i, P_j, s, κ) where κ is non-null. In this model, each program can fork multiple sub-processes to handle multiple KE-sessions. If party P_i has a KE-session with input $(P_i, P_j, s, role)$ and party P_j has a KE-session with input $(P_j, P_i, s', role')$, and $s = s'$, then we say that the two KE-sessions are *matching*.

There are two adversarial models.

1. **Unauthenticated-links adversarial model (UM).** A UM adversary $\mathcal{U}$ is a PPT Turing machine which controls all the communication links of the system and schedules all protocol events including the initiation of protocol executions and message delivery. The UM adversary can perform any actions such as injecting or modifying messages. In addition, $\mathcal{U}$ knows all the local output of a party except the secret output, i.e. the session key of a key exchange session. The adversary $\mathcal{U}$ is also given the following additional capabilities by making some appropriate queries to the game simulator. We focus on reviewing the model for KE protocols.

   (a) corrupt party P_i: Upon corruption, $\mathcal{U}$ learns all the current internal information of P_i and takes over P_i. P_i can no longer be activated and does not generate further output.

   (b) session-state reveal of a specific KE-session within party P_i: $\mathcal{U}$ learns all the current state of the KE-session, but not the long-term key or the state information of other KE-sessions of party P_i.

   (c) session-key reveal of a completed KE-session within party P_i: $\mathcal{U}$ learns the secret output (i.e. the session key) of the KE-session.

   (d) session expiration of a completed KE-session within party P_i: The session key of the KE-session will be erased. This query is for capturing the property of perfect forward secrecy.

   A KE-session with input $(P_i, P_j, s, role)$ is called *exposed* if (1) $\mathcal{U}$ corrupts P_i or P_j before s expires or (2) $\mathcal{U}$ performs session-state reveal or session-key reveal on session s of P_i or P_j.

2. **Authenticated-links adversarial model (AM).** An AM adversary has all the capabilities as a UM adversary but the AM adversary is not allowed to inject or modify messages (except that the sender is corrupted or if the message belongs to an exposed session). The adversary is restricted to deliver messages faithfully, and the message is only to be delivered once (i.e. all the messages the adversary received are in a set M of undelivered

messages with format (m,P_i,P_j) where P_i and P_j are the sender and the receiver, m is some message, and once m is delivered, it is removed from M). However, the adversary can choose not to deliver a message.

For the rest of the paper, we denote an AM adversary by $\mathcal{A}$, and a UM adversary by $\mathcal{U}$. Let $AUTH_{\pi,\mathcal{A}}$ be the global output (i.e. the output of the adversary and all the parties in the system) of running π in AM and $UNAUTH_{\pi,\mathcal{U}}$ be that in UM.

Definition 1. Let π and π' be n-party message-driven protocols in AM and UM, respectively. We say that π' **emulates** π in UM if for any UM-adversary $\mathcal{U}$ there exists an AM-adversary $\mathcal{A}$ such that $AUTH_{\pi,\mathcal{A}}$ and $UNAUTH_{\pi',\mathcal{U}}$ are computationally indistinguishable.

Since the authentication in AM is explicitly ensured, if π' emulates π in UM, the authentication in UM is also ensured.

Definition 2 (Authenticator). An authenticator $\mathcal{C}$ is an algorithm such that for any protocol π in AM, the protocol $\mathcal{C}(\pi)$ emulates π in UM.

2.2 The Modular Approach

The way to construct an authenticator is given in [Bellare et al. 1998], where a layered approach is used. An authenticator $\mathcal{C}_\lambda$ can be constructed from an **MT-authenticator** λ which emulates the basic message transmission protocol. The basic idea is that whenever a party P_i wants to send or receive a message, we emulate it using λ.

Theorem 3 ([Bellare et al. 1998]). *Let λ be an MT-authenticator (i.e. λ emulates message transmission in unauthenticated networks), and let $\mathcal{C}_\lambda$ be a compiler constructed based on λ as described above. Then $\mathcal{C}_\lambda$ is an authenticator.*

With this powerful compiler on hand, we can first design a secure key exchange protocol in the authenticated link model where we do not need to consider authentication. Then we can use the compiler to derive a new protocol which supports authentication in the unauthenticated link model. In the next, we define session key security by following the definition in [Canetti and Krawczyk 2001].

To define the session key security of a KE protocol, the capability of $\mathcal{A}$ is extended by allowing it to perform a test-session query. At any time during the game, $\mathcal{A}$ can issue a test-session query on a KE-session that is completed, unexpired and unexposed. Let κ be the corresponding session key. A coin $b \overset{R}{\leftarrow} \{0,1\}$ is tossed by the game simulator after receiving a test-session query from $\mathcal{A}$. If $b=0$, κ is returned to $\mathcal{A}$; otherwise, a value chosen according to the distribution of session keys is returned to $\mathcal{A}$. $\mathcal{A}$ can still carry out regular activities on this *test-session* after issuing the query but is not allowed to expose the test-session. However, the attacker *is allowed* to corrupt a partner to the test-session as soon as the test-session (or its matching session) expires at that party. This captures the perfect forward secrecy property of a KE protocol. At the end of its run, $\mathcal{A}$ outputs a bit b' (as its guess for b).

All the session specific internal state information is securely erased once a KE-session is completed. It only leaves the value of the secret local output (the session key) after completing the session.

Definition 4. A KE protocol π is **SK-secure** if the properties below hold.

1. If two uncorrupted parties complete matching sessions, then they both output the same key;
2. The probability that the adversary guesses correctly the bit b' (i.e., $b' = b$) is no more than 1/2 plus a negligible fraction in the security parameter.

The following theorem is very important for the modular approach.

Theorem 5 ([Canetti and Krawczyk 2001]). *Let π be a SK-secure key exchange protocol in AM and let λ be an MT-authenticator. Then $\pi' = \mathcal{C}_\lambda(\pi)$ is a SK-secure key exchange protocol in UM.*

In other words, from the compilation, protocol π' not only achieves authenticity, but also 'inherits' the session key security from π.

Remark. From the proof of Theorem 5, it is not difficult to see that different MT-authenticators can be used for different message flows in π, and the resulting compiler, which is a combination of different MT-authenticators, is still an authenticator. This important feature makes the modular approach much more flexible and powerful.

MT-authenticators can be built from various cryptographic primitives. In case public key cryptosystems are used, it is assumed that each party has its private key and also knows the authentic public keys of other parties. Below is a signature based MT-authenticator [Bellare et al. 1998] which allows party P_i to send a message m to party P_j in an authenticated way.

$$\begin{aligned} &P_i \leftarrow P_j : N_j \\ &P_i \rightarrow P_j : m,\, SIG_{P_i}(m, N_j, P_j) \end{aligned}$$

$N_j \in_R \{0,1\}^k$ is a random challenge (or nonce) and SIG_{P_i} is the signature generation function of P_i, where k is a security parameter. The signature scheme is assumed to be secure against chosen message attack [Goldwasser et al. 1988].

In the subsection below, we propose a new MT-authenticator.

2.3 A One-Pass Counter Based MT-authenticator

We propose a new MT-authenticator. This MT-authenticator will be used in constructing the key exchange protocols for secure roaming in the subsequent sections of this paper. It helps simplify the authentication procedures and improve the protocol efficiency.

Suppose a party P_i shares a random key κ with another party P_j. Each of P_i and P_j initiates a counter starting with 0. Our one-pass counter based MT-authenticator λ_{COUNT} proceeds as follows.

- Whenever P_i wants to send a message m to P_j, P_i increases its local counter $COUNT_{P_i}$ by one, sends $m, COUNT_{P_i}, MAC_\kappa(m, COUNT_{P_i}, P_j)$ to P_j, and adds a message "P_i sent m to P_j" to P_i's local output.
- Upon receiving $m, COUNT_{P_i}, MAC_\kappa(m, COUNT_{P_i}, P_j)$, P_j verifies that the MAC is correct and $COUNT_{P_i} > COUNT_{P_j}$ where $COUNT_{P_j}$ is the local counter of P_j. If all of the verifications succeed, P_j outputs "P_j received m from P_i" and sets $COUNT_{P_j} = COUNT_{P_i}$.

Theorem 6. *If MAC is secure against chosen message attack, λ_{COUNT} is an MT-authenticator.*

Proof. Here we assume that for the two communicating parties, one of them always has the role of initiator and the other always has the role of responder. But the proof also applies to the case when the communication is in bidirectional and both directions are using the same shared key but with two different counters (one for send and one for receive) at each side.

Let $\mathcal{U}$ be a UM adversary interacts with λ_{COUNT}. We define an AM adversary $\mathcal{A}$ which simulates $\mathcal{U}$ as follows.

$\mathcal{A}$ runs $\mathcal{U}$ on a simulated interaction with a set of parties running λ_{COUNT}. First, $\mathcal{A}$ chooses and distributes the shared secret keys for the imitated parties. Then $\mathcal{A}$ proceeds its simulation as follows.

1. When $\mathcal{U}$ activates an imitated party $\tilde{P}_i$ for sending a message m to imitated party $\tilde{P}_j$, $\mathcal{A}$ activates P_i in AM for sending m to P_j.
2. When some imitated party $\tilde{P}_j$ outputs "$\tilde{P}_j$ received m from $\tilde{P}_i$", $\mathcal{A}$ activates party P_j in AM with incoming message m from P_i.
3. When $\mathcal{U}$ corrupts a party, $\mathcal{A}$ corrupts the same party in AM and hands the corresponding information (from the simulated run) to $\mathcal{U}$.
4. $\mathcal{A}$ outputs whatever $\mathcal{U}$ outputs.

Let **E** denote the event that imitated party $\tilde{P}_j$ outputs "$\tilde{P}_j$ received m from $\tilde{P}_i$" where $\tilde{P}_i$ is uncorrupted and the message (m, P_i, P_j) is not currently in the set M of undelivered messages. In other words, either P_i is not activated for sending m to P_j, or P_j has already had the same output before. Since neither $\tilde{P}_i$ nor $\tilde{P}_j$ is corrupted, **E** is also the event that either $\tilde{P}_i$ is not activated for sending m to $\tilde{P}_j$, or $\tilde{P}_j$ has already had the same output before.

If **E** never occurs, then $AUTH_{MT,\mathcal{A}}$ and $UNAUTH_{\lambda,\mathcal{U}}$ are equally distributed. It remains to show that **E** occurs only with negligible probability.

We prove it by contradiction. Assume **E** occurs with non-negligible probability, then we construct a forger $\mathcal{F}$ that breaks the MAC with non-negligible probability.

The forger $\mathcal{F}$ has a MAC oracle O_{MAC} that uses an unknown random key, $\mathcal{F}$ can request O_{MAC} on any message or any verification pair (m, σ). The task of $\mathcal{F}$ is to produce a valid (m, σ) pair but m has not been queried to the oracle before.

$\mathcal{F}$ starts by running $\mathcal{U}$ on a set of parties running λ_{COUNT}. $\mathcal{F}$ chooses and distribute shared keys between parties with one exception, $\mathcal{F}$ randomly chooses one pair of users $\tilde{P}_i$ and $\tilde{P}_j$, and whenever one party is required to produce or verify an MAC for some value, $\mathcal{F}$ queries the oracle and hands the result to that party. If $\mathcal{U}$ chooses to corrupt either $\tilde{P}_i$ or $\tilde{P}_j$, $\mathcal{F}$ fails and aborts.

Note that running λ_{COUNT} in this case is equivalent to a regular run. Assume **E** occurs with probability $v(k)$, the probability it occurs between $\tilde{P}_i$ and $\tilde{P}_j$ is then $\frac{2v(k)}{n(n-1)}$ since $\tilde{P}_i$ and $\tilde{P}_j$ are randomly chosen. Also note that when **E** occurs between $\tilde{P}_i$ and $\tilde{P}_j$, neither $\tilde{P}_i$ nor $\tilde{P}_j$ is corrupted.

In the case $\tilde{P}_i$ is not activated for sending m to $\tilde{P}_j$, $\mathcal{F}$ outputs the MAC value $\mathcal{U}$ delivered to $\tilde{P}_j$ in the last message as its forgery. On the other hand, if $\tilde{P}_j$ has already had the same output before, then $\tilde{P}_j$ has received m from $\tilde{P}_i$ with a counter, say $COUNT^{old}_{\tilde{P}_i}$ before. Now when $\tilde{P}_j$ accepts m the second time, then $\tilde{P}_j$ must have accepted another incoming message from $\tilde{P}_i$ with the same m but with a more updated counter value, say $COUNT^{new}_{\tilde{P}_i}$ such that $COUNT^{new}_{\tilde{P}_i} > COUNT^{old}_{\tilde{P}_i}$. However, $\tilde{P}_i$ ($\mathcal{F}$) has never queried with $(m, COUNT^{new}_{\tilde{P}_i}, \tilde{P}_j)$ because each message from $\tilde{P}_i$ to $\tilde{P}_j$ is assumed to be different. Hence $\mathcal{F}$ outputs the MAC value that $\mathcal{U}$ has delivered to $\tilde{P}_j$ in the last message as its forgery as well. Thus $\mathcal{F}$ successfully produces a forgery with probability $\frac{2v(k)}{n(n-1)}$. □

Remark 1. Since we are considering the simple message transfer protocol here, we do not need to consider session-state reveal, session-key reveal or session expiration queries. Also the adversary controls the activations of the parties, so the counter values are also known to the adversary.

Remark 2. A delayed previous message will be rejected by the receiver after a later sent message is accepted in UM, while the delayed message will still be accepted in AM, but it will not affect $\mathcal{A}$ on emulating $\mathcal{U}$.

Remark 3. We do not consider the counter resynchronization problem that may occur if the database of either party collapses. Depending on the nature of different applications, different resynchronization mechanisms may be used, for example, if this MT-authenticator is used in telecommunication protocols, we may build an additional resynchronization protocol to solve the problem.

2.4 SK-secure Key Exchange Protocols in *AM*

Now we consider another building block of the modular approach: SK-secure key exchange protocols in AM. According to [Canetti and Krawczyk 2001], the classical Diffie-Hellman key-exchange protocol is SK-secure under the Decisional Diffie-Hellman (DDH) assumption [Boneh 1998]. Denote G a subgroup of prime order q of a multiplicative group $\mathbb{Z}_p^*$ and g a generator of G. Below is the review of the protocol. P_i and P_j are two parties and s is the session ID.

Diffie-Hellman (DH) Key Exchange in AM:

1. On input (P_i, P_j, s), P_i chooses $x \in_R \mathbb{Z}_q$ and sends $(P_i, s, \alpha = g^x)$ to P_j.
2. Upon receipt of $(P_i,\ s,\ \alpha)$, P_j chooses $y \in_R \mathbb{Z}_q$ and sends $(P_j, s, \beta = g^y)$ to P_i, then computes $\kappa = \alpha^y$, erases y, and outputs the session key κ under session ID s.
3. Upon receipt of (P_j, s, β), P_i computes $\kappa' = \beta^x$, erases x, and outputs the session key κ' under session ID s.

Theorem 7 ([Canetti and Krawczyk 2001]). *Under the Decisional Diffie-Hellman (DDH) assumption, Diffie-Hellman (DH) Key Exchange above is SK-secure in AM.*

3 Authenticated Key Exchange for Roaming (AKE-R)

We now propose a formal key exchange definition for secure roaming under the CK-model. The definition will capture the first three properties for secure roaming listed in Sec. 1.3: Foreign Server Authentication, Subscription Validation, and Key Establishment. We then design a key exchange protocol for satisfying the definition.

Let k be a system-wide security parameter. Let $\mathsf{C}(k) = \{C_1, \cdots, C_{Q_1(k)}\}$ be the set of roaming users (clients) in the system and $\mathsf{S}(k) = \{S_1, \cdots, S_{Q_2(k)}\}$ be the set of servers in the system, where Q_1 and Q_2 are some polynomials and C_i, S_j are the corresponding identities of the parties, for $1 \leq i \leq Q_1(k)$ and $1 \leq j \leq Q_2(k)$.

Subscription. The term 'subscribe' is commonly used to describe some special relationship between a user and a server without clear definition. Based on the widely-used concept of subscription in mobile communications, we give the following definition for subscription.

Definition 8 (Subscribe). Given a security parameter k, 'subscribe' is a computable function $\mathit{Subscribe}$ from $\mathsf{C}(k)$ into $\mathsf{S}(k)$. We say that C_A is 'subscribed' to S_H if $\mathit{Subscribe}(C_A) = S_H$ where $C_A \in \mathsf{C}(k)$ and $S_H \in \mathsf{S}(k)$.

We assume that each user has subscribed to one and only one server, and the subscription is persistent. Hence scenarios related to changing subscriptions of users are excluded.

Based on the terminologies of mobile communications, S_H is said to be the home server of C_A and S_V is said to be a foreign server of C_A if $S_V \neq S_H$. We also assume that the inverse $\mathit{Subscribe}^{-1}$ is computable. Hence for any $S_H \in \mathsf{S}(k)$, $\mathit{Subscribe}^{-1}(S_H)$ is the set of all $C_A \in \mathsf{C}(k)$ such that $\mathit{Subscribe}(C_A) = S_H$.

3.1 The Security Definition of AKE-R

An AKE-R (Authenticated Key Exchange for Roaming) protocol is a message-driven protocol. In the CK-model, each session is modelled by running a subprocess within a party with input $(P_i, P_j, P_\ell, s, \mathit{role})$. We extend the CK-model so that the parties are categorized as roaming users and servers. For a user C_A, the input of his session will be in the form $(C_A, S_V, S_H, s, \mathsf{initiator})$ where the *role* must be $\mathsf{initiator}$. For a server, the *role* can either be $\mathsf{responder}$ or $\mathsf{voucher}$. We say that three sessions of a user and two servers, C_A, S_V and S_H, respectively, are *3-party matching*, if in an execution of the AKE-R protocol, user C_A has a session with input $(C_A, S_V, S_H, s, \mathsf{initiator})$, server S_V has a session with input $(S_V, C_A, S_H, s', \mathsf{responder})$, server S_H has a session with input $(S_H, C_A, S_V, s'', \mathsf{voucher})$, where $S_H \neq S_V$, $s = s' = s''$ and $\mathit{Subscribe}(C_A) = S_H$.

Definition 9 (SK-Secure AKE-R Protocol). An AKE-R protocol π runs among C_A, S_V, and S_H is called **SK-secure** if the following properties hold.

1. If uncorrupted C_A, S_V and S_H complete 3-party matching sessions, then upon the completion of the protocol, C_A and S_V output the same key.
2. The probability that anyone except C_A and S_V guesses correctly the bit b' (i.e., $b' = b$) in a test-session query (see Def. 4) is no more than 1/2 plus a negligible fraction in the security parameter.

Having an AKE-R protocol SK-secure is not enough in practice. In particular, the definition does not capture Subscription Validation. For example, suppose we extend the two-party Diffie-Hellman key exchange protocol which is proven SK-secure in AM (reviewed in Sec. 2.4) to a three-party version in such a way that C_A and S_V conduct the key exchange. After completed, S_V sends the session ID to S_H. Then S_H accepts and the protocol is completed. This three-party version can be shown to be SK-secure with respect to Def. 9 but obviously not satisfying the requirement of Subscription Validation.

For Subscription Validation, S_V has to make sure that a statement issued by a server has been received claiming that C_A is subscribed to the server and is connecting to S_V. In addition, C_A has to make sure that S_V has received a statement issued by S_H.

Definition 10 (Secure AKE-R Protocol). An AKE-R protocol π run among C_A, S_V, and S_H is secure if the following properties hold.

1. π is a SK-secure AKE-R protocol.
2. Upon the completion of the 3-party matching sessions,
   (a) the matching session of S_H has sent the message below to S_V;
   C_A `is subscribed to` S_H `and is talking to` S_V
   (b) the matching session of S_V has received the message below from S_H;
   C_A `is subscribed to` S_H `and is talking to` S_V
   (c) the matching session of S_V has sent the message below to C_A;
   S_H `claimed that` C_A `is its subscriber and is talking to` S_V
   (d) the matching session of C_A has received the message below from S_V.
   S_H `claimed that` C_A `is its subscriber and is talking to` S_V

We can see that the first three security goals in Sec. 1.3 are captured by Def. 10. *Foreign Server Authentication* is captured by items (c) and (d). *Subscription Validation* is captured by items (a), (b), (c), and (d), we should note that if a server S_H outputs the message no matter the actual home server of C_A is S_H or not, then it relies on C_A to check (item (d) above) if S_H is cheating, this helps detect the Deposit-Case Attack [Yang et al. 2005]. *Key Establishment* is captured by the SK-security.

In the following, we state an important theorem which allows us to reuse all the proven MT-authenticators given in Sec. 2 for constructing secure AKE-R protocols.

Theorem 11. *Let $\mathcal{C}_\lambda$ be an authenticator (Def. 2) constructed from an MT-authenticator λ exemplified in Sec. 2. If π is a secure AKE-R protocol in AM, then $\pi' = \mathcal{C}_\lambda(\pi)$ is a secure AKE-R protocol in UM.*

Proof. We prove it by showing that the following requirements are satisfied.

1. If π is SK-Secure, then π' is also SK-Secure:
   The proof follows that of Theorem 6 in [Canetti and Krawczyk 2001] directly. Since that proof only requires π' emulates π, which is done due to the definition of $\mathcal{C}_\lambda$, and also it does not depend on the parties involved in the protocol. Therefore, if π is SK-Secure, $\pi' = \mathcal{C}_\lambda(\pi)$ "inherits" this property from π.

2. If π satisfies requirement 2 of Def. 10 in AM, π' satisfies the requirement in UM as well:
   Suppose π' does not satisfy this requirement. Then there exists an adversary $\mathcal{U}$ in UM such that the global output of running π' with $\mathcal{U}$ does not follow the requirement, but there is no adversary in AM can do this since π satisfies the requirement. Thus the global outputs are distinguishable, in contradiction to $\mathcal{C}_\lambda$ being an authenticator. □

We now start describing our AKE-R protocol. Our protocol consists of two steps. In the first step, a user carries out a *Pre-authentication* protocol with his home server when he is in the network operated by his home server. In this step, a 'long-term' *user authentication key* will be established. This key will be used in the second step for authenticating the user.

In the second step, the user is roaming and communicating with a foreign server. The roaming key exchange protocol will be carried out by the user, the foreign server and the home server. The purpose of the protocol is to let the roaming user and the foreign server establish a fresh session key so that a secure channel can be built.

3.2 Pre-Authentication

The purpose of pre-authentication is to have C_A and S_H establish a *user authentication key*, $authK_A$. One way to achieve the task is to run a SK-secure key exchange protocol. Alternatively, like in the cellular networks, the key has already been embedded in the SIM card. No matter in which of these two cases, we assume that $authK_A$ is randomly chosen and only shared by C_A and S_H.

After running the pre-authentication protocol, C_A stores $authK_A$ and a *counter* initialized to 0 in some secure and non-volatile memory location. S_H creates an entry for C_A in its own database. In the entry for C_A, attributes such as the identity of C_A, $authK_A$ and a counter value are included. The counter is also initialized to 0, and will be increased in each run of the AKE-R Main protocol below.

3.3 The AKE-R Main Protocol

We first describe our AKE-R Main Protocol in AM. Then we *compile* it to a secure AKE-R protocol in UM using those authenticators described in Sec. 2.3.

3.3.1 A Secure AKE-R Protocol in *AM*

We extend the two-party DH key exchange protocol which is SK-secure in *AM* (Sec. 2.4) to a three-party variant. Since the network is controlled by the adversary, here we use a virtual link between A and H although they are not physically linked in our target applications.

Extended DH Protocol in *AM*: (Fig. 1)

1. A roaming user C_A initiates the protocol execution by choosing $x \in_R \mathbb{Z}_q$ and sends $(C_A, S_H, S_V, s, \alpha = g^x)$ to S_H.

2. Upon receipt of $(C_A, S_H, S_V, s, \alpha)$, S_H checks if C_A is its subscriber, if not, it rejects and halts. Otherwise, S_H sends $(C_A, S_H, S_V, s, \alpha)$ to S_V.

3. Upon receipt of $(C_A, S_H, S_V, s, \alpha)$, S_V checks if S_H is a legitimate server in its server list, if not, it rejects and halts. Otherwise, S_V chooses $y \in_R \mathbb{Z}_q$, sends $(C_A, S_V, S_H, s, \beta = g^y)$ to C_A, then computes $\kappa = \alpha^y$, erases y, and outputs the session key κ under session ID s.

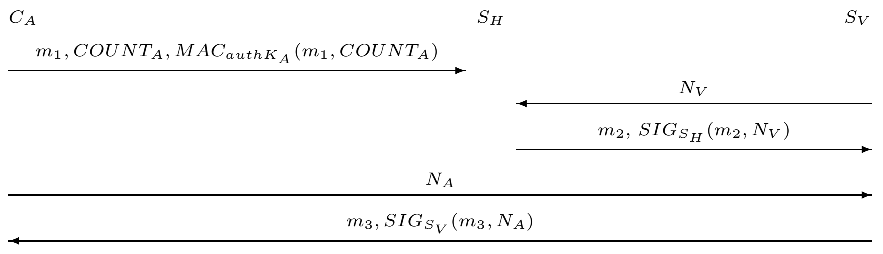
4. Upon receipt of $(C_A, S_H, S_V, s, \beta)$, C_A checks if S_V is the correct server it wants to communicate with and S_H is indeed its home server. If either verification fails, it rejects and halts. Otherwise, it computes $\kappa' = \beta^x$, erases x, and outputs the session key κ' under session ID s.

$$\begin{tikzcd}
C_A \arrow[r, "{m_1 = (C_A, S_H, S_V, s, \alpha = g^x)}"] & S_H \arrow[r, "{m_2 = (C_A, S_H, S_V, s, \alpha)}"] & S_V \arrow[ll, bend left, "{m_3 = (C_A, S_H, S_V, s, \beta = g^y)}"]
\end{tikzcd}$$

Figure 1: Protocol 1 – Extended DH protocol in *AM*

Corollary 12. *Under the DDH assumption, the Extended DH protocol is a secure AKE-R in AM.*

According to Def. 10, the proof should contain two parts. The proof for the first part, i.e., SK-security, is straightforward by following Theorem 7. The second condition of Def. 10 is achieved by checking the identities in each message.


C_A
S_H
S_V
$m_1, COUNT_A, MAC_{authK_A}(m_1, COUNT_A)$
N_V
$m_2,\ SIG_{S_H}(m_2, N_V)$
N_A
$m_3, SIG_{S_V}(m_3, N_A)$

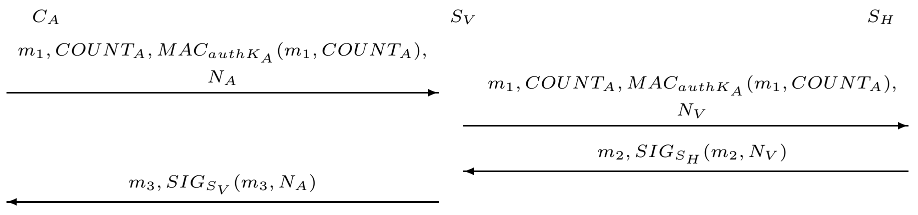
$m_1 = (C_A, S_H, S_V, s, \alpha = g^x)$ $m_2 = (C_A, S_H, S_V, s, \alpha)$ $m_3 = (C_A, S_H, S_V, s, \beta = g^y)$

Figure 2: Protocol 2 – Extended DH protocol in UM

3.3.2 A Secure AKE-R Protocol in UM

An AKE-R protocol in UM can be derived by applying MT-authenticators to the Extended DH protocol in AM. We apply the one-pass counter based MT-authenticator to m_1, the signature based MT-authenticator to m_2 and m_3. The resulting protocol is in Fig. 2.

After deriving the AKE-R protocol in UM, an optimization [Tin et al. 2003] of message flows can be applied. And the final AKE-R protocol in UM is illustrated in Fig. 3.


C_A
S_V
S_H
$m_1, COUNT_A, MAC_{authK_A}(m_1, COUNT_A),$
N_A
$m_1, COUNT_A, MAC_{authK_A}(m_1, COUNT_A),$
N_V
$m_2, SIG_{S_H}(m_2, N_V)$
$m_3, SIG_{S_V}(m_3, N_A)$

$m_1 = (C_A, S_H, S_V, s, \alpha = g^x)$ $m_2 = (C_A, S_H, S_V, s, \alpha)$ $m_3 = (C_A, S_H, S_V, s, \beta = g^y)$

Figure 3: Protocol 3 – Optimized Extended DH protocol in UM

We can see that Protocol 3 preserves the same message flows as Protocol 1, if we treat S_V as a router. And Protocol 3 uses the same MT-authenticator for each authentication step as Protocol 2 does, therefore, Protocol 3 maintains the security of Protocol 2.

4 Anonymous and Authenticated Key Exchange for Roaming (AAKE-R)

We now start specifying the security definition of the *anonymous* version of an AKE-R protocol. This version of AKE-R protocol will satisfy all the five properties listed in Sec. 1.3. In particular, the protocol should provide User Anonymity and User Untraceability. We call such a protocol an Anonymous and Authenticated Key Exchange for Roaming (AAKE-R) protocol.

4.1 The Security Definition of User Anonymity and Untraceability

Game A: "The game is carried out by a simulator $\mathcal{S}$ which runs an adversary $\mathcal{U}$. It is based on the adversarial model UM.

1. $\mathcal{S}$ sets up a system with users in $\mathsf{C}(k)$ and servers in $\mathsf{S}(k)$.
2. $\mathcal{S}$ then runs $\mathcal{U}$ and answers $\mathcal{U}$'s queries.
3. $\mathcal{U}$ can execute the AAKE-R protocol on any parties in the system by activating these parties and making queries.
4. Among all the parties in the system, $\mathcal{U}$ picks two users $C_i, C_j \in \mathsf{C}(k)$ and two servers $S_V, S_H \in \mathsf{S}(k)$ such that $Subscribe(C_i) = Subscribe(C_j) = S_H$.
5. $\mathcal{U}$ sends a test query by providing C_i, C_j, S_V and S_H.
6. The simulator $\mathcal{S}$ simulates one AAKE-R protocol run among C_i, S_V and S_H, and another one among C_j, S_V and S_H. $\mathcal{S}$ also updates the state information of each party due to the simulation. Then $\mathcal{S}$ tosses a coin b, $b \overset{R}{\leftarrow} \{0,1\}$. If $b = 0$, the simulation transcript with C_i is returned to $\mathcal{U}$, otherwise, that with C_j is returned to $\mathcal{U}$. Denote T the transcript $\mathcal{U}$ receives, and sid the session ID in T.
7. After receiving the response of the test query, $\mathcal{U}$ can still launch all the allowable attacks through queries and also activate parties for protocol executions as before.
8. At the end of $\mathcal{U}$'s run, it outputs a bit b' (as its guess for b)."

$\mathcal{U}$ wins the game if (1) S_H, C_i and C_j are uncorrupted, (2) for the session sid in step 6 above, $\mathcal{U}$ can only perform session-state reveal, session-key reveal and session expiration queries to S_V. (3) $\mathcal{U}$ guesses correctly the bit b (i.e. outputs $b' = b$.). Define $\mathbf{Adv}_{\pi,\mathcal{U}}(k) = \Pr[\mathcal{U} \text{ wins the game}] - 1/2$.

Definition 13 (User Anonymity and Untraceability). An AKE-R protocol provides user anonymity and untraceability, if for sufficiently large security parameter k, $\mathbf{Adv}_{\pi,\mathcal{U}}(k)$ is negligible.

The formulation of Def. 13 is very powerful and can be shown to ensure both user anonymity and user untraceability required by a good AAKE-R protocol. It guarantees that as long as the home server is uncorrupted, the adversary can neither tell the identity from the messages of one session nor link that session to another one.

4.2 A Secure AAKE-R Protocol

Based on the secure AKE-R protocol (in UM) proposed in Sec. 3, we now modify it so that it also provides user anonymity and untraceability.

To provide user anonymity, the identity of the user should not be sent in clear. In addition, it should not be known to the foreign server according to the anonymity definition above. To do so, we first change the identities in m_1, m_2 and m_3 as follows:

$$\begin{aligned} m_1 &= alias, C_A, S_H, S_V, s, g^x \\ m_2 &= alias, S_H, S_V, s, g^x \\ m_3 &= alias, S_H, S_V, s, g^y \end{aligned}$$

Here *alias* acts as a *temporary ID* for the roaming user which is a fixed-length binary string chosen randomly from $\{0,1\}^k$. Also, encrypt the first message of the protocol using S_H's public key.

In addition, for anonymity, all the counters used in the system should have the same length. We define a counter $COUNT \in \{0,1\}^{Q_3(k)}$ for some polynomial Q_3 and assume that the value of $COUNT$ would not reach $2^{Q_3(k)} - 1$ in the lifetime of the system.

The complete AAKE-R main protocol is illustrated in Fig. 4.

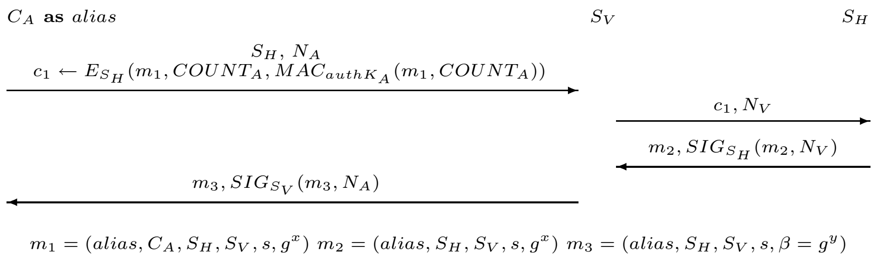

C_A **as** $alias$
S_V
S_H
S_H, N_A
$c_1 \leftarrow E_{S_H}(m_1, COUNT_A, MAC_{authK_A}(m_1, COUNT_A))$
c_1, N_V
$m_2, SIG_{S_H}(m_2, N_V)$
$m_3, SIG_{S_V}(m_3, N_A)$
$m_1 = (alias, C_A, S_H, S_V, s, g^x)$ $m_2 = (alias, S_H, S_V, s, g^x)$ $m_3 = (alias, S_H, S_V, s, \beta = g^y)$

Figure 4: The AAKE-R Main Protocol

Theorem 14. *If E_{S_H} is CCA-secure, $\boldsymbol{Adv}_{\pi,\mathcal{U}}(k)$ is negligible.*

Proof. We prove it by contradiction. Namely, if the protocol is not anonymous, that is, if $\mathcal{U}$ wins the game with non-negligible advantage, $v(k)$, over random guess (which is half chance), we construct a distinguisher $\mathcal{D}$ to break E_{S_H}.

We start by describing a game for the distinguisher $\mathcal{D}$. First, $\mathcal{D}$ adaptively queries a decryption oracle with any ciphertext. Then $\mathcal{D}$ chooses two messages msg_0 and msg_1 and asks the game simulator for a ciphertext. The simulator randomly picks $b \overset{R}{\leftarrow} \{0,1\}$ and gives $\mathcal{D}$ the ciphertext c such that $c = E_{S_H}(msg_b)$. After receiving c, $\mathcal{D}$ adaptively queries the decryption oracle with any ciphertext except c. $\mathcal{D}$ is to output a value $b' \in \{0,1\}$ as its guess for b.

Now we construct $\mathcal{D}$ which simulates $\mathsf{Game\ A}$. First, $\mathcal{D}$ sets up the system appropriately by creating a set $\mathsf{C}(k)$ of users and another set $\mathsf{S}(k)$ of servers. It then initializes all the users in $\mathsf{C}(k)$ with randomly chosen authentication keys from $\{0,1\}^k$ and counters which are with initial value 0, and initializes all the servers in $\mathsf{S}(k)$ with randomly chosen public key pairs for encryption and another set of public key pairs for signature. Afterwards, $\mathcal{D}$ randomly picks a server, S_H, and replace its encryption public key such that it corresponds to E_{S_H}.

$\mathcal{D}$ runs $\mathcal{U}$ and answers all its queries and simulates all the responses of party activation due to protocol execution. If $\mathcal{U}$ picks S_H as the home server, two users C_i, C_j such that $Subscribe(C_i) = Subscribe(C_j) = S_H$, and some server S_V as the foreign server during the test query, $\mathcal{D}$ answers the query by providing the transcript of a protocol run constructed as follows.

First, $\mathcal{D}$ randomly chooses x in $\mathbb{Z}_q$, $alias$ in $\{0,1\}^k$, a session ID s in $\{0,1\}^k$, and constructs two messages msg_0 and msg_1 as follows.

$$msg_0 = alias || \ C_i || \ S_H || \ S_V \ || \ s \ || \ g^x || \ COUNT_i + 1 || \ MAC_i$$
$$msg_1 = alias || \ C_j || \ S_H || \ S_V \ || \ s \ || \ g^x || \ COUNT_j + 1 || \ MAC_j$$

$\mathcal{D}$ queries the CCA-security simulator with msg_0 and msg_1. Suppose the CCA-security simulator returns a ciphertext c. Then, $\mathcal{D}$ constructs

$$message_1 = \langle \ S_H, N_A, c \ \rangle$$
$$message_2 = \langle \ c, N_V \ \rangle$$
$$message_3 = \langle \ alias, S_H, S_V, s, g^x, SIG_{S_H}(alias, S_H, S_V, s, g^x, N_V) \ \rangle$$
$$message_4 = \langle \ alias, S_H, S_V, s, g^y, SIG_{S_V}(alias, S_H, S_V, s, g^y, N_A) \ \rangle.$$

where $N_A, N_V \in_R \{0,1\}^k$. $\mathcal{D}$ also updates the counter values in the internal states of C_i, C_j and S_H.

The transcript returned by $\mathcal{D}$ to $\mathcal{U}$, as the response for $\mathcal{U}$'s test query, is $(message_1, message_2, message_3, message_4)$. $\mathcal{D}$ continues the game by answering all the queries made by $\mathcal{U}$ and simulating all the responses of party activation due to protocol execution. If $\mathcal{U}$ asks a $\mathsf{session\text{-}state\ reveal}$ query to S_V with session ID s, the simulator $\mathcal{S}$ returns the random coins in generating N_V, g^y and

$SIG_{S_V}(alias, S_H, S_V, s, g^y, N_A)$ to $\mathcal{U}$. If $\mathcal{U}$ corrupts S_V, $\mathcal{S}$ returns the long-term keys of S_V, and the internal state of S_V which includes the state information of session s, to $\mathcal{U}$.

When $\mathcal{U}$ outputs a bit value b' as its guess, $\mathcal{D}$ outputs b' and halts.

If $\mathcal{U}$ does not pick S_H as the home server in his test query, $\mathcal{D}$ just randomly picks a value $b' \overset{R}{\leftarrow} \{0, 1\}$, outputs it and halts.

Analysis: Let $\mathbf{E}$ be the event that $\mathcal{U}$ picks S_H as the home server in its test query. Since $\mathcal{D}$ chooses S_H from $\mathsf{S}(k)$ in the game uniformly at random, $\Pr[\mathbf{E}] = \frac{1}{Q_2(k)}$. Hence we have

$$\begin{aligned}\Pr[\mathcal{D} \text{ guesses } b \text{ correctly}] &= (\frac{1}{2} + \upsilon(k))\Pr[\mathbf{E}] + \frac{1}{2}(1 - \Pr[\mathbf{E}]) \\ &= \frac{1}{2} + \frac{\upsilon(k)}{Q_2(k)}\end{aligned}$$

which is non-negligible over random guess. □

5 Communications in the Home Domain

In this section, we present an anonymous authenticated key exchange (AAKE) protocol which is used for the mobile user to communicate with the home server in the home domain. Note that in Sec. 4, we consider anonymity and untraceability against the foreign server, which is an insider of the protocol, but in this section, we only need to consider anonymity and untraceability against outsiders. The protocol is presented in Fig. 5.

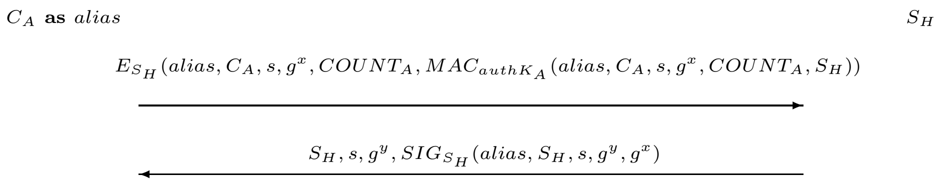

C_A **as** $alias$
S_H
$E_{S_H}(alias, C_A, s, g^x, COUNT_A, MAC_{authK_A}(alias, C_A, s, g^x, COUNT_A, S_H))$
$S_H, s, g^y, SIG_{S_H}(alias, S_H, s, g^y, g^x)$

Figure 5: Anonymous Authenticated Key Exchange in the Home Domain

It is easy to see that the AAKE protocol is an abbreviation of the AAKE-R protocol and g^x in the first message also plays the role of a nonce in the signature based MT-authenticator. The security proof for this protocol follows that of the AAKE-R protocol in a straightforward way and is omitted here.

6 Conclusion

Based on the modular approach of the CK-model, we build an anonymous and authenticated key exchange protocol for roaming (AAKE-R) which is provably secure and efficient. The performance of a roaming protocol is mainly determined by its communication rounds, our AAKE-R protocol, which requires only four rounds among the three communicating parties, has the optimal round efficiency compared with existing roaming protocols in the literature. As a side-product from our modular construction, we propose a one-pass counter based MT-authenticator and show its security under the assumption that there exists an MAC function which is secure against chosen message attack. Like other proven secure MT-authenticators, this new MT-authenticator can also be reused to construct new protocols in the future.

Throughout the construction of our AAKE-R scheme, we introduced a formal definition for secure roaming under the CK-model, and proposed a definition for user anonymity and untraceability. We hope that the framework and definitions can be further studied and adopted for analyzing new protocols of this kind in the future.

Acknowledgements

We would like to thank Dr. Xiaojian Tian and the anonymous reviewers for their careful reading and useful comments. The work was supported by CityU grants (Project Nos. 7001844, 7001959, 7002001).

References

[Ateniese et al. 1994] Ateniese, G., Herzberg, A., Krawczyk, H., Tsudik, G.: "On Traveling Incognito" Proc. IEEE Workshop on Mobile Systems and Applications (Dec 1994).

[Bellare et al. 1998] Bellare, M., Canetti, R., Krawczyk, H.: "A Modular Approach to the Design and Analysis of Authentication and Key Exchange Protocols" Proc. 30^{th} ACM Symp. on Theory of Computing, ACM (May 1998), 419–428.

[Bellare and Rogaway 1994] Bellare, M., Rogaway, P.: "Entity authentication and key distribution" Proc. CRYPTO 1993, Lect. Notes in Comp. Sci. 773, Springer (1994), 232–249.

[Bellare and Rogaway 1995] Bellare, M., Rogaway, P.: "Provably Secure Session Key Distribution – The Three Party Case" Proc. 27^{th} ACM Symp. on Theory of Computing, ACM (1995), 57–66.

[Blake-Wilson et al. 1997] Blake-Wilson, S., Johnson, D., Menezes, A.: "Key Agreement Protocols and Their Security Analysis" Proc. 6^{th} IMA International Conference on Cryptography and Coding, Lect. Notes in Comp. Sci. 1355, Springer (1997), 30–45.

[Blake-Wilson and Menezes 1997] Blake-Wilson, S., Menezes, A.: "Entity Authentication and Authenticated Key Transport Protocols Employing Asymmetric Techniques" Proc. Security Protocols Workshop, Lect. Notes in Comp. Sci. 1361, Springer (1997), 137–158.

[Boneh 1998] Boneh, D.: "The Decision Diffie-Hellman Problem" Proc. 3^{rd} Algorithmic Number Theory Symposium, Lect. Notes in Comp. Sci. 1423, Springer (1998), 48–63.

[Buttyan et al. 2000] Buttyan, L., Gbaguidi, C., Staamann, S., Wilhelm, U.: "Extensions to an Authentication Technique Proposed for the Global Mobility Network" IEEE Trans. on Communications, 48(3), IEEE (2000), 373–376.

[Canetti and Krawczyk 2001] Canetti, R., Krawczyk, H.: "Analysis of Key-Exchange Protocols and Their Use for Building Secure Channels" Proc. EUROCRYPT 2001, Lect. Notes in Comp. Sci. 2045, Springer (2001), 453–474.

[Go and Kim 2001] Go, J., Kim, K.: "Wireless Authentication Protocol Preserving User Anonymity" Proc. 2001 Symposium on Cryptography and Information Security (SCIS 2001), 159–164.

[Goldwasser et al. 1988] Goldwasser, S., Micali, S., Rivest, R.: "A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attack" SIAM J. Computing, 17(2), April 1988, 281–308.

[Hwang and Chang 2003] Hwang, K., Chang, C.: "A Self-encryption Mechanism for Authentication of Roaming and Teleconference Services" IEEE Trans. on Wireless Communications, 2(2), IEEE (2003), 400–407.

[Mouly and Pautet 1992] Mouly, M., Pautet, M.: "The GSM System for Mobile Communications" Published by the authors, 1992.

[Mu and Varadharajan 1996] Mu, Y., Varadharajan, V.: "On the design of security protocols for mobile communications" Proc. 1^{st} Australasian Conference on Information Security and Privacy, Lect. Notes in Comp. Sci. 1172, Springer (1996), 134–145.

[Samfat et al. 1995] Samfat, D., Molva, R., Asokan, N.: "Untraceability in Mobile Networks" Proc. MobiCom 1995, 26–36.

[3GPP] Technical Specification Group (TSG) SA: "3GPP TS 33.102: 3rd Generation Partnership Project (3GPP), 3G Security, Security Architecture" Oct, 2003.

[Tin et al. 2003] Tin, Y., Boyd, C., Gonzalez-Nieto, J.: "Provably Secure Key Exchange: An Engineering Approach" Proc. Australasian Information Security Workshop (AISW2003), 97–104.

[Wong 2005] Wong, D.: "Security Analysis of Two Anonymous Authentication Protocols for Distributed Wireless Networks" Proc. 3^{rd} IEEE Intl. Conf. on Pervasive Computing and Communications Workshops, IEEE (2005), 284–288.

[Yang et al. 2005] Yang, G., Wong, D., Deng, X.: "Deposit-Case Attack Against Secure Roaming" Proc. 10^{th} Australasian Conference on Information Security and Privacy, Lect. Notes in Comp. Sci. 3574, Springer (2005), 417–428.