Enhancing CK-Model for Key Compromise Impersonation Resilience and Identity-based Key Exchange

Robert W. Zhu, Xiaojian Tian, and Duncan S. Wong

Department of Computer Science
City University of Hong Kong
Hong Kong, China
{zhuwei,xjtian,duncan}@cs.cityu.edu.hk

Abstract. In 2001, Canetti and Krawczyk proposed a security model (CK-model) for authentication protocols. They also gave an indistinguishability-based definition for key exchange protocols. Since then the model has almost exclusively been used for analyzing key exchange protocols, although it can be applied to authentication protocols in general. The model not only captures a large class of attacks but also provides a modular approach to the design of authentication protocols. However, the model does not capture the property of Key Compromise Impersonation (KCI) Resilience. Until now, analysis concerning this property has mostly been done heuristically and restricted to key exchange protocols only. Previous attempts on formalizing KCI have mostly been done in some ad hoc manner and additional proofs have to be given specifically for the security of KCI resilience. In this paper, we propose an extension to the CK-model, which allows, for the first time, the KCI attacks to be considered in authentication protocols in general, rather than restricted to key exchange protocols, and no more additional proofs are required specifically for KCI security. With the revival of interest in identity-based (ID-based) cryptography, there have been many new ID-based key exchange protocols proposed. Despite the fact that some of them have been proven in some restricted versions of a model proposed by Bellare and Rogaway in 1993 and some others have been proven in the original CK-model, there is no rigorous model specifically for ID-based key exchange security. In particular, forward secrecy against compromised Key Generation Server (KGS-FS) has never been captured even though this notion is more important and stronger than the perfect forward secrecy in ID-based key exchange. For this, we further extend our model to ID-based setting and capture the property of KGS-FS for ID-based key exchange security.

1 Introduction

In 1993, the first security model using a computational approach to proofs of key exchange (KE) protocols was proposed by Bellare and Rogaway [4]. The model captures a hostile environment with an active adversary which is able to launch various interleaving attacks [5,16] and has the capabilities of corrupting communicating parties and revealing session keys. In 1998, Bellare, Canetti and Krawczyk [1] proposed a different model which was later extended by Canetti and Krawczyk [11] in 2001. One of the most significant components of these two models is a *modular approach* which is used for both designing and analyzing protocols. By using this modular approach, a protocol is first shown to satisfy a certain set of security requirements in an ideally authenticated network, then the protocol is transformed using some reusable building blocks called authenticators in such a way that the transformed protocol will then be satisfying the same set of security requirements but in a more realistic unauthenticated network. One major advantage of this approach is that authentication protocols can be analyzed in general rather than restricted to KE protocols. The functionalities provided by the protocols depend on the set of security requirements specified. The authenticators are responsible for making sure that the final transformed protocol supports mutual authentication in the realistic unauthenticated network. In this paper, we refer to the model proposed by Canetti and Krawczyk [11] as the CK-model.

The CK-model captures a large class of practical attacks and desirable security properties. The model is also flexible enough so that it can make some security properties optional. In this paper, we propose a new approach to extend the CK-model so that it will support a security property called **Key Compromise Impersonation (KCI) Resilience** [9,6,18] and show that our approach is more general and natural on capturing KCI Resilience than the comparable approaches. We also show how to extend the model further so that it can capture the security properties of protocols in the identity-based settings.

KCI Resilience is a security property that compromising the long-term secret of one party should not allow an adversary to masquerade *to* the party as a different party. Let us consider a two-party protocol: suppose an adversary has compromised the long-term secret information of a party A, but not another party B. Obviously, this allows the adversary to impersonate A to B. If, in addition, the adversary is able to impersonate *to* A as B, then such a protocol is said to be vulnerable to KCI attack. We consider KCI attacks against public-key cryptographic protocols only, rather than symmetric key cryptographic protocols. This is because in the symmetric key setting, once the adversary has compromised the long-term secret shared between A and B, the adversary can obviously impersonate A to B as well as B to A [9].

Although the CK-model has captured a large class of attacks and desirable security properties, it does not capture KCI resilience. Until now, analysis concerning this property has mostly been done heuristically and restricted to KE protocols only. Also, other attempts on formalizing KCI have mostly been done in some ad hoc manner and additional proofs have to be given specifically for the security of KCI resilience (please refer to Sec. 2 for details). In addition, it is not commonly aware that KCI attacks actually apply to authentication protocols in general. They belong to a type of attacks that compromises the authentication of a protocol rather than session key security. In this paper, we propose an extension to the CK-model, which allows, for the first time, the KCI attacks to be considered in authentication protocols in general, rather than restricted to KE protocols. In addition, the extension also retains the support of all other properties currently supported by the original model.

In recent years, with the revival of interest in identity-based (ID-based) cryptography [25], there have been many new ID-based KE protocols proposed (to name a few, they include [27,13,8,15,22,29,14,30]). Despite the fact that some of them have been proven in some restricted versions of a model proposed by Bellare and Rogaway in 1993 and some others have been proven in the original CK-model, there is no rigorous model specifically for ID-based key exchange security. In particular, Forward Secrecy against compromised Key Generation Server (KGS-FS) [17,13,22] has never been captured. The property KGS-FS requires forward secrecy to be maintained even after the *master secret* of the KGS is compromised. KGS-FS is a stronger notion than that of PFS as compromising the master secret of the KGS implies compromising the secret keys of all parties in the context of ID-based cryptography. In this paper, we further extend our enhanced CK-model to ID-based setting and in particular capture the property of KGS-FS for ID-based key exchange security.

Contributions. We propose a new approach to extend the CK-model so that it captures, for the first time, the property of KCI Resilience for authentication protocols in general, rather than restricted to KE protocols. In addition, the extension also retains the support of all other properties currently supported by the original model. Our approach allows KCI related attacks to be captured naturally and no additional proof is required specifically for KCI security. It is more general and natural than the comparable approaches proposed previously (reviewed in Sec. 2).

We further propose an ID-based extension on our KCI Resilience extended CK-model. The new extension captures the general setting of ID-based authentication protocols (hence including ID-based KE protocols). The model, when used for analyzing ID-based KE protocols, also captures the property of KGS-FS, which has never been formalized before. To exemplify the use of this new extension, we provide a proof of security for a recently proposed non-pairing based ID-based KE protocol [30] and show that it supports both KCI Resilience and KGS-FS.

Paper Organization. In Sec. 2, we give a brief historical retrospect on the related work. In Sec. 3, we describe the enhancement of the CK-model for supporting KCI Resilience. In Sec. 4, we further extend the model to support ID-based settings and KGS-FS for ID-based KE protocols. In Sec. 5, we review an ID-based KE protocol and show its security under our model. To be self-contained, we also provide a brief review for the CK-model in Appendix A.

2 Related Work

Although KCI has been studied elaborately with respect to key exchange protocols, analysis concerning this property has mostly been performed heuristically [9]. In the following, we focus on reviewing those few results on formal security analysis related to KCI.

In [21], Kudla and Paterson extended the original Bellare-Rogaway model [4] and defined a modified version called the mBR model for KE protocols. In their model, the party corruption query can be used to reveal the long-term secret information of the concerned party which can still be chosen as an attacking target in the test oracle query afterwards. Although the model captures KCI, further extension of this model is required in order to capture the PFS (perfect forward secrecy) as well. In addition, the model only restricts to the security analysis of KE protocols. In this paper, we will see that we have a more general approach in formalizing KCI security and this allows us to not only capture the KCI security of authentication protocols in general, but also have one single model to support both KCI resilience and the property of PFS for key exchange security.

In [20], Krawczyk proposed a variant of MQV protocol and showed that it is secure under the CK-model. In [20, Sec. 6.1], the property of KCI Resilience was also considered, but again, the methodology used to show KCI Resilience can only apply to KE protocols, where a test-session should be (and must be) chosen in order to let the underlying adversary launch KCI attacks. In addition, an additional security proof has to be given specifically for KCI. In other words, KCI attacks are not 'integrated' in the model and this type of attacks has to be evaluated separately every time when a new protocol is to be analyzed. Similar technique has also been used in a restricted version of Bellare-Rogaway model [13, Theorem 2] for showing the property of KCI Resilience of an identity-based KE protocol. In Sec. 4, we will discuss in detail that KCI is a type of attacks that not just applies to KE protocols, but it generally applies to authentication protocols. We extend the CK-model which explicitly separates the notions of authentication and key exchange, and this approach allows us to evaluate KCI attacks with other interleaving attacks against the authentication protocols in one single security proof. We no longer need to give any additional proof specifically for evaluating the KCI Resilience.

In [10], Canetti proposed a generic framework for analyzing and constructing cryptographic protocols such that the security of protocols is maintained under a general operation called "universal composition"(UC). This framework facilitates the definition of KCI Resilience, but a formal definition of KCI Resilience is not yet to be given. Furthermore, the current UC framework does not support ID-based setting. We believe that our results presented in this paper can also help formalize KCI attacks and ID-based setting in the UC framework, and we will leave this as a further research problem. In particular, we consider the property of KCI resilience in authentication protocols in general, rather than restricted to KE protocols. This understanding helps us formalize KCI attacks in some proper place of UC framework.

Since the first set of ID-based KE protocols were proposed in late '80s and early '90s, there has been a revival of interest in this area. However, due to the lack of a rigorous model of ID-based key exchange security, many of these protocols do not have any formal security analysis for supporting their security claims, and some others (such as [26,15]) have been shown to be insecure with respect to their original security claims. In [7], a summary of many recently proposed ID-based KE protocols on whether they have attempted to give security proofs or not has been given. As noted, many of these protocols that have security proofs actually have proofs only in a restricted version of the original model (exclusively using the Bellare-Rogaway model [4] mentioned above). In the restricted version, the adversary is prevented from asking any Reveal query. For some other protocols [8,14,29,30], although security proofs have been given in the original Bellare-Rogaway model or CK-model, properties that are only specific to ID-based KE protocols are not captured in these models and therefore, have not been proven. In particular, KGS forward secrecy (KGS-FS) has never been captured even though this notion is more important and stronger than perfect forward secrecy in ID-based key exchange security. In this paper, we also extend our KCI Resilience enhanced CK-model to ID-based setting and capture the property of KGS-FS for ID-based key exchange security.

3 The KCIR-enhanced CK-model

In Appendix A, we provide a self-contained review for the CK-model. For readers who are familiar with the CK-model, the appendix can safely be skipped as we use the same set of notations as in [11].

For a KE protocol, the property of Key Compromise Impersonation (KCI) Resilience [9,6,18] requires that compromising the long-term secret of a party A should not allow an adversary to masquerade *to* A as another party B, under the assumption that the adversary does not have the long-term secret of B. KCI Resilience is an important property for KE protocols. In the CK-model, KCI Resilience is not captured. As an example of a proven SK-secure KE protocol in the CK-model but being vulnerable to KCI attacks, we refer to [8, Protocol 1]. The protocol does not support KCI Resilience as knowing the long-term secret of either one of the two communicating parties will allow the adversary to impersonate any of these two parties.

To see why a proven SK-secure KE protocol in the CK-model does not necessarily imply resistance to KCI attacks, we notice that compromising the long-term secret of a party in the CK-model corresponds to corrupting the party and this party can no longer participate in any further protocol interactions. As a result, the CK-model cannot capture KCI attacks in which the adversary is trying to impersonate *to* a party whose long-term secret is compromised.

A New Query. In order to incorporate the notion of KCI Resilience into the CK-model, we introduce a new query called key compromise query into the model. When an adversary issues a key compromise query for a specified party, say P_i, the adversary will only learn the long-term secret of P_i but not any other internal information of P_i. A special note will also be written in P_i's local output which indicates that P_i is *compromised*. Sometimes, we may say that a session is compromised (or uncompromised). In that case, it is referring to the corresponding party of the session of being compromised (or not being compromised yet).

The query is different from the existing party corruption query. First, the key compromise query reveals only the long-term secret of the party rather than all the state information of the party. Moreover, a compromised party can still be activated and queried. This property differentiates itself from the weaker type of party corruption query described in [3,19] under a model called "weak-corruption" model. In particular, in our model, we allow a party to be uncorrupted while compromised.

For defining a secure authentication protocol, the original definition of protocol emulation (reviewed in Appendix A as Def. 4) can be applied directly to the KCIR-enhanced CK-model above for supporting KCI Resilience. The following definition is the replacement of Def. 4.

Definition 1. *In the enhanced CK-model described above, let π and π' be two n-party message driven protocols. We say that π' emulates π in UM if given any adversary $\mathcal{U}$ in UM against protocol π', there exists an adversary $\mathcal{A}$ in AM against π such that $AUTH_{\pi,\mathcal{A}}$ and $UNAUTH_{\pi',\mathcal{U}}$ are computationally indistinguishable.*

There is no additional definition required specifically for capturing KCI Resilience. And also, this is the first time that KCI Resilience can be formally analyzed without being tied up with a key exchange protocol.

With the key compromise query added, this enhanced model 'integrates' KCI attacks into the CK-model. We explain this statement in three steps by considering two parties P_i and P_j. Firstly, in order to model the KCI attack, the adversary is given the capability of obtaining the long-term secret of P_i and is still allowed to interact with P_i (by issuing the key compromise query to P_i). Secondly, in AM, the KCI attack can never happen: by knowing the compromised party P_i's long-term secret, the AM adversary cannot masquerade to P_i as P_j since the adversary cannot inject any message as the sender (P_j here) unless the sender is also corrupted or compromised, or the message belongs to an exposed session. Thirdly, by the definition of protocol emulation (Def. 1 above), we refer to a protocol π' in UM as "authenticated" if there exists protocol π in AM such that π' emulates π in UM. Therefore, if a protocol π' is authenticated in the enhanced CK-model, then the protocol π' supports KCI Resilience. However, if a protocol π' is only authenticated in the original CK-model, but not in the enhanced CK-model, then π' would not support KCI Resilience.

We emphasize that KCI Resilience is a property related to "authentication". It applies to authentication protocols in general, rather than restricted to KE protocols. When compared with all the comparable approaches [21,20,13], which can only capture the property of KCI Resilience for KE protocols and additional proofs must be given particularly for KCI security, our modeling approach is more versatile, systematic, and easier to apply. We will see some examples of applying our KCIR-enhanced CK-model in Sec. 5.

Layered Authenticators. The definition of an authenticator and the construction method of layered authenticators are the same as before except that they are now under the context of the KCIR-enhanced CK-model. Please refer to Sec. A.3 for a brief review.

Theorem 1. *If λ is an MT-authenticator in the KCIR-enhanced CK-model then $\mathcal{C}_\lambda$ constructed based on λ using the layered approach as described in [1] and reviewed in Sec. A.3 is an authenticator.*

The proof is similar to that of [1, Theorem 3]. For examples of secure MT-authenticators in the KCIR-enhanced CK-model, we can consider the signature-based MT-authenticator described in [1, Sec. 3.1]. This authenticator can be shown to be secure in the KCIR-enhanced CK-model. Since the proof can be obtained directly from that of an ID-based variant described in Sec. 4.3, we therefore defer detailed discussions of it in this section.

In the following, we give more details on defining a secure KE protocol which is considered as a type of authentication protocols in the CK-model.

SK Security. With the additional key compromise query, the adversary now has one more method (besides party corruption) to obtain the long-term secret of a party. Therefore, we should adjust the definition of SK security for KE protocols for taking the effect of key compromise into consideration.

First, we change the conditions of making a test-session query by an adversary such that the adversary can only make a test-session query on a KE-session that is *completed*, *unexpired*, *unexposed* and *uncompromised*. After completing the query, the adversary, just like in the original CK-model for SK security, can continually carry out regular actions other than exposing the test-session. The difference between the original model and the KCIR-enhanced model is that the adversary can now compromise the party. Since compromising a party only lets the adversary get the long-term secret of the party, hence once the test-session is completed, the adversary is allowed to issue key compromise to the party, that will have the same effect as corrupting the party after the test-session expires. The definition of an SK-secure KE protocol is then modified as follows.

Definition 2. *A KE protocol π is called SK-secure in the KCIR-enhanced CK-model if the following properties hold.*

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

By following the proof of [11, Theorem 6], it is straightforward to show that if there exists an SK-secure KE protocol in AM, by using an MT-authenticator for the KCIR-enhanced CK-model (such as the signature-based MT-authenticator mentioned above), an SK-secure KE protocol in UM can be derived and the protocol will then support KCI Resilience. For example, Protocol SIG-DH [11] is SK-secure and also resistant to KCI attacks. This is because Protocol 2DH [11] can be shown to be SK-secure in AM under the KCIR-enhanced CK-model. By applying the signature-based MT-authenticator mentioned above to Protocol 2DH, we obtain Protocol SIG-DH.

In all the comparable approaches of formalizing KCI security [21,20,13], KCI security is modeled in some restricted way on key exchange protocols only, where the underlying adversary must first choose some "KCI-attacked" party for the test oracle query, then if the adversary cannot distinguish the session key from a random value, KCI resilience is said to be achieved. However, our method is quite different. We treat the property of KCI resilience to be related to authentication, and separate it from SK-security. We do not need to choose a test session and can apply KCI security to authentication protocols in general. As explained before, the KCI resilience of a protocol π is automatically ensured in AM of our model. If a protocol π' emulates π in UM, the KCI security of π' will also be ensured. Therefore, we believe that our method is more natural and general.

Making PFS Optional. For the scenario where perfect forward secrecy (PFS) is not needed, we need to prevent the adversary from getting the long-term secrets of involving parties of the test-session. Besides requiring that the test-session never expire (for preventing the adversary from issuing party corruption query onto the parties of the test-session or its matching session), we also need to prohibit the adversary issue any key compromise query on these parties.

Making KCI Resilience Optional. There are also some scenarios where KCI Resilience is not necessary or may not be possible. For example, in some applications an adversary would not gain any advantage when masquerading to the compromised party as a different party, or a conventional symmetric-key based KE protocol simply does not support KCI Resilience. In these cases, we should remove key compromise from the adversary's list of allowable queries and the model will fall back to the original CK-model.

In the next section, we further extend the KCIR-enhanced CK-model for supporting the general ID-based setting and KGS-FS for ID-based KE protocols.

4 The ID-based Enhanced CK-model

In an ID-based setting, there is a key generation server (KGS) which has a *master key*. The KGS uses the master key to generate the long-term secret key for each of the parties in the system so that the secret key of any party can be derived directly from the master key or the KGS and the unique identity of the party. In the context of ID-based KE protocols (or authentication protocols in general), additional security concerns are raised by considering this extra party, KGS, in the system and the property above. In particular, for ID-based KE protocols, the notion of KGS forward secrecy (KGS-FS) is to require a session key should remain secure even after the master key of the KGS is compromised. This is at least as strong as the perfect forward secrecy (PFS) of a non-ID-based KE protocol as knowing the KGS' master key implies knowing the secret keys of all parties in the system.

In the following, we describe the changes that need to be made on the KCIR-enhanced CK-model for extending it to an ID-based enhanced model.

4.1 ID-based Message Driven and KE Protocols

An ID-based message driven protocol $\tilde{\pi}$ is a collection of programs, each program is to be run by a different party which is *created* by the KGS. We emphasize that it is the KGS who creates parties during the protocol execution. Each party can perform the same set of actions as described in Sec. A.1. An ID-based KE protocol is an ID-based message driven protocol with similar specification to that of a KE protocol described in Sec. A.1.

4.2 The Adversarial Models

Similar to the CK-model (original or KCIR-enhanced), there are two models, UM and AM, with adversary $\tilde{\mathcal{U}}$ and $\tilde{\mathcal{A}}$, respectively. The sets of actions that can be carried out by these two adversaries are similar to their counterparts in the KCIR-enhanced CK-model. In addition to these, we introduce two additional adversarial activities to our models for capturing the existence of the KGS. They are create party query and corrupt KGS query.

The UM. Let k be a security parameter. The system has the UM-adversary $\tilde{\mathcal{U}}$, a PPT machine called KGS and a number of parties denoted by $P_1, \cdots, P_N$ (also modeled as PPT machines) for the ID-based message driven protocol $\tilde{\pi}$. The number of parties in the system is N where N is a polynomial in k. We assume that $\tilde{\mathcal{U}}$ does not *create* (which will be defined later) more than N parties. The initialization function I is changed to give out a pair of outputs only: $I(r,k) = (I(r,k)_0,\ I(r,k)_{KGS})$, where $I(r,k)_0$ is the public information which becomes known to all parties, the KGS and the adversary while $I(r,k)_{KGS}$ becomes known only to the KGS and it is called the *master key* of the KGS. In the system, we also assume that there is a secure channel between the KGS and each of the N parties. Hence the KGS can make use of this secure channel to send information of a particular party so that the information will only be known to the party and the KGS.

When $\tilde{\mathcal{U}}$ makes a create party query with a specified party, say P_i with identity ID_i (chosen by $\tilde{\mathcal{U}}$), the KGS is activated to assign the identity ID_i to P_i, if P_i is not yet created and ID_i has not been assigned to any party yet. Then, KGS computes a long-term secret sk_i from ID_i and the master key of the KGS using an algorithm called user key generation algorithm which should be defined explicitly in the protocol specification. sk_i is then sent to P_i through the secure channel. A special note is also appended to the KGS' local output which specifies that P_i has been created with identity ID_i. Party P_i is said to be *created*. The create party query can only be made once for each of the parties, and the number of created parties is not fixed, instead it grows with the number of create party queries made by the adversary.

Only after a party is created, $\tilde{\mathcal{U}}$ can start activating the party either by incoming message queries or action request queries, and perform all the adversarial actions described in Sec. A.2 and key compromise queries described in Sec. 3 on the party. Note that all queries are only allowed to be made on created parties.

There is an additional adversarial action for capturing the property of KGS-FS. By issuing a corrupt KGS query, $\tilde{\mathcal{U}}$ can learn the master key of the KGS. This event is recorded through a special note in the KGS' local output. From this point on, the KGS cannot be activated anymore and there is no further local output generated. The KGS is said to be *corrupted*.

The global output $UNAUTH_{\tilde{\pi},\tilde{\mathcal{U}}}$ now consists of the cumulative outputs of all *created* parties as well as the outputs of $\tilde{\mathcal{U}}$ and KGS.

The AM. An AM-adversary $\tilde{\mathcal{A}}$ has all the capabilities of $\mathcal{A}$ described in Sec. A.2 and Sec. 3 with the additional adversarial actions, create party and corrupt KGS. The global output $AUTH_{\tilde{\pi},\tilde{\mathcal{A}}}$ is analogous to $UNAUTH_{\tilde{\pi},\tilde{\mathcal{U}}}$, where the computation is carried out in the AM.

Remark: Unlike other ID-based adversarial models such as [2], our model does not need extract query, which allows the adversary to compromise users' secret information. This is because the party corruption query has already included the capability of extract query. Therefore, in our model, additional extract query is not necessary.

4.3 Authenticators

The definition of an authenticator remains unchanged except that it should now be defined under the context of the ID-based enhanced CK-model. To construct an authenticator, we can also use the layered approach to construct a **layered authenticator** from an MT-authenticator as described in Sec. A.3.

Theorem 2. *If λ is an MT-authenticator in the ID-based enhanced CK-model, then $\mathcal{C}_\lambda$ constructed based on λ using the layered approach as described in [1] and reviewed in Sec. A.3 is an authenticator.*

The proof is similar to that of [1, Theorem 3].

As an example of an MT-authenticator in the ID-based enhanced CK-model, we can see that the signature-based MT-authenticator described in [1, Sec. 3.1] can be shown to be an authenticator of the MT protocol in the ID-based enhanced CK-model after changing the signature scheme of the initiator to an ID-based signature scheme [25,23,12,2]. The security requirement of the ID-based signature scheme is the ID-based extension of the conventional existential unforgeability against adaptive chosen message (euf-cma). It concerns about existential unforgeability against adaptive chosen message attack as well as chosen identity attack [12,2] (euf-cma-ida). In the following, we describe the ID-based signature-based MT-authenticator in detail and provide a proof under our model.

Let λ_{SIG}^{ID} be the ID-based signature-based MT-authenticator. When the initialization function I is first invoked (by the game simulator in UM), it runs the master-key generation algorithm of the underlying euf-cma-ida secure ID-based digital signature scheme and outputs $I(r,k) = (I(r,k)_0, I(r,k)_{KGS})$. The public information $I(r,k)_0$ is passed to all parties $P_1, \cdots, P_N$, the KGS and the adversary, while the master key of the KGS, $I(r,k)_{KGS}$, is only passed to the KGS. When the ID-based enhanced UM-adversary $\tilde{\mathcal{U}}$ makes a create party query with some specified party P_i with identity ID_i, for some $i \in \{1, \cdots, N\}$, the KGS is activated to assign ID_i to P_i (which is not yet created and ID_i has not been assigned to any party yet). Then, the KGS computes a long-term secret sk_i from ID_i and $I(r,k)_{KGS}$ using the user-key generation algorithm of the underlying ID-based digital signature scheme. Next, the KGS passes sk_i to P_i using the secure channel defined in UM.

Fig. 1 illustrates the ID-based signature-based MT-authenticator λ_{SIG}^{ID} carried out between two created parties A and B.
$IDSign_{sk_A}$ denotes the signing algorithm of the underlying ID-based digital signature scheme under party A's long-term secret sk_A. When A is activated by an action request for sending message m to B in session s, A first sends "message: s, m" to B and outputs "A sent m to B in session s". B then chooses a random challenge $N_B \in_R \{0,1\}^k$ and sends "challenge: s, m, N_B" to A. Upon receipt of "challenge: s, m, N_B" from B,

$$\begin{aligned}&A \rightarrow B : s,\, m\\ &A \leftarrow B : s,\, m,\ N_B\\ &A \rightarrow B : s,\, m,\ IDSign_{sk_A}(s, m, N_B, B)\end{aligned}$$

Fig. 1. ID-based Signature-based MT-authenticator

A sends "signature: $s, m, IDSign_{sk_A}(s, m, N_B, B)$" to B. Upon receipt of "signature: $s, m, IDSign_{sk_A}(s, m, N_B, B)$" from A, B checks whether the signature is valid. If yes, B accepts the signature and outputs "B received m from A in session s"; otherwise, B rejects the message and terminates session s within B.

Theorem 3. *Suppose that the ID-based digital signature scheme in use is euf-cma-ida secure. Then protocol λ_{SIG}^{ID} emulates MT protocol in UM in the ID-based enhanced CK-model.*

Please refer to Appendix B for the proof.

4.4 SK Security for ID-based KE Protocols

The definition of a secure ID-based KE protocol can be formalized by following Def. 2 and including the impacts brought in by the two additional adversarial actions, i.e. create party and corrupt KGS. First, we require that the adversary can only make a test-session query on a KE-session that is completed, unexpired, unexposed, uncompromised and also with the KGS being *uncorrupted*. After completing the query, the adversary can continually carry out regular actions but is not allowed to expose the test-session. However, the adversary can now compromise the party or even issue a corrupt KGS query. The effect would be similar to corrupting the party corresponding to the test-session after it expires.

The definition of SK-secure should also be modified.

Definition 3. *An ID-based KE protocol $\tilde{\pi}$ is called SK-secure in ID-based enhanced CK-model if the following properties hold.*

1. *With KGS uncorrupted, if two uncorrupted and uncompromised but* ***created*** *parties complete matching sessions, then both output the same session 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.*

By following the proof of [11, Theorem 6], we can also show that if there exists an SK-secure KE protocol in AM of the ID-based enhanced CK-model, by using an MT-authenticator for the ID-based enhanced CK-model, an SK-secure KE protocol in UM can be derived.

KGS Forward Secrecy (KGS-FS). The original CK-model captures the notion of perfect forward secrecy (PFS), that is, a previously established session key should remain secure even after the long-term secrets of both involving parties are compromised. In the context of ID-based KE protocols, KGS forward secrecy (KGS-FS) [17,13,22] provides an even stronger protection to session keys than that of PFS. By KGS-FS, previously established session keys will still remain secure even after the master key of the KGS is compromised. The ID-based enhanced CK-model above captures the notion of KGS-FS by allowing the adversary to issue corrupt KGS once after the test-session query returns.

For scenarios where KGS-FS is not needed and only PFS is required, we need to remove corrupt KGS from the adversary's list of allowable queries. Note that although the adversary is not allowed to corrupt KGS, he can still issue a party corruption query on a party after the test-session is expired. Hence PFS is captured.

5 SK-Secure ID-Based KE Protocols

In [27], Smart proposed one of the first pairing-based ID-based KE protocols. As noted by Smart in [27], the protocol does not support KGS-FS nor PFS. It only supports *partial* forward secrecy, namely the session key will remain secure if only one of the two involving parties is compromised. In [13], Chen and Kudla proposed several improvements of Smart's protocol in terms of efficiency and the support of KGS-FS. Security proofs of their protocols were also given in a *restricted* Bellare-Rogaway model. The restricted Bellare-Rogaway model is strictly weaker than our ID-based enhanced CK-model. First, the restricted model does not capture KGS-FS. The property of KGS-FS for one of the protocols in [13], which is believed to support KGS-FS, could therefore be only argued heuristically in [13]. Second, the restricted model does not allow the adversary to make any Reveal query. The Reveal query in Bellare-Rogaway model provides the adversary similar capability to the session-output reveal query in the CK-model. If Reveal query is allowed, their protocols cannot be secure (we refer readers to [13] for details), and therefore cannot be secure in our ID-based enhanced CK-model either. Third, the additional capability provided to the adversary by session-state reveal queries in our ID-based enhanced CK-model (that is carried over directly from the original CK-model) further makes protocols in [13] and related protocols insecure due to the session corruption attack [28]. Therefore, the restricted Bellare-Rogaway model is strictly weaker than our ID-based enhanced CK-model.

On the property of KCI resilience, as we have already explained in detail before, the restricted Bellare-Rogaway model does not have KCI attacks 'integrated' in the model and this type of attacks has to be evaluated separately every time when a new protocol is to be analyzed. As we can see in [13, Theorem 2], an addition security proof has to be given for evaluating the KCI resilience of a protocol. In [20], it has the similar issue.

An ID-based KE Protocol supporting KCI Resilience and KGS-FS

In [30], Zhu et al. proposed an ID-based KE protocol which does not use bilinear pairings. They also showed the security of their protocol in the original CK-model. This implies that it supports PFS. However, it does not show that the protocol satisfies KGS-FS. In the following, we use the ID-based enhanced CK-model to show that the protocol supports KGS-FS and also has the property of KCI Resilience.

Let $k \in \mathbb{N}$ be a security parameter. The initial information $I(r,k)_0$ consists of a finite field $\mathbf{F}$, an elliptic curve $\mathscr{C}$ defined over $\mathbf{F}$, an element P of large prime order q in $\mathscr{C}$, and also the public key of the KGS denoted by mpk. This public key is generated as $mpk = mskP$ where $msk \in_R \mathbb{Z}_q$. The secret information $I(r,k)_{KGS}$ for the KGS is therefore the value of msk which is the master key of KGS. There is a non-pairing-based ID-based signature scheme associated with the protocol. For our discussion in this paper, we do not go into the details of the signature scheme but denote a signature generated by the scheme as $IDSign_A(m)$ where m is the message and the subscript

A indicates that the signature is generated by party A using its long-term secret. The signature can be verified using A's "public key", that is, A's identity denoted by ID_A. The signature scheme in [30] has been shown to be euf-cma-ida.

Let A and B be the initiator and responder identified by $ID_A, ID_B \in \{0,1\}^*$. Let sk_A and sk_B be the long-term secrets of A and B, respectively. They are generated by the KGS when A and B are created. A generation is done by using the Schnorr signature scheme [24] on the party's identity as the message under the master key of the KGS. Suppose A and B already have a unique session-id s shared. The protocol proceeds as follows.

1. The initiator, A, on input (A, B, s), chooses $a \in_R \mathbb{Z}_q$ and sends $(A, s, T_A = aP)$ to B.
2. Upon receipt of (A, s, T_A), the responder B chooses $b \in_R \mathbb{Z}_q$ and sends $(B, s, T_B = bP)$ together with its signature $IDSign_B(s, T_B, T_A, A)$; it also computes the session key $K = bT_A$ and erases b.
3. Upon receipt of (B, s, T_B) and B's signature, party A checks the correctness of each component in the incoming message and checks whether the signature is valid with respect to the "public key" ID_B. If the verification succeeds, A sends $(A, s, IDSign_A(s, T_A, T_B, B))$ to B. A then computes $K' = aT_B$, erases a, and outputs the session key K' under session-id s.
4. Upon receipt of (A, s, sig), B checks the correctness of each component in the incoming message and determines if the signature sig is valid with respect to the "public key" ID_A. If yes, B outputs the session key K under session-id s.

Security Analysis. The protocol can be shown to be SK-secure (with KCI Resilience and KGS-FS) in the ID-based enhanced CK-model in two steps. First, an ID-based KE protocol is proposed and proven SK-secure in AM. Second, some appropriate MT-authenticators are applied to the SK-secure KE protocol in AM using the layered approach for obtaining an SK-secure ID-based KE protocol in UM.

For the first step, we begin with Protocol 2DH [11, Sec. 5.1] and describe it in the context of elliptic curve cryptography. It is the classical two-move Diffie-Hellman key exchange protocol. When formalized in AM, we describe it as having the initiator A send $(A, s, T_A = aP)$ to B and then having the responder B send $(B, s, T_B = bP)$ back to A. The session key is abP. According to [11, Theorem 8], Protocol 2DH is SK-Secure in AM under the Decisional Diffie-Hellman assumption. In our ID-based enhanced CK-model, we can also construct a Distinguisher $\tilde{\mathcal{D}}$ which proceeds in the same way as the Distinguisher $\mathcal{D}$ in the proof of [11, Theorem 8] except with the following additional actions: whenever the adversary $\tilde{\mathcal{A}}$ creates a new party, $\tilde{\mathcal{D}}$ creates a party accordingly; whenever a *created* party is compromised or the KGS is corrupted, $\tilde{\mathcal{D}}$ hands the related information of that party or KGS to $\tilde{\mathcal{A}}$, respectively. The proof will then follow and we will get the contradiction we want for showing the SK security of Protocol 2DH in AM of the ID-based enhanced CK-model.

For the second step, in Sec. 4.3, we show that the signature-based MT-authenticator described in [1, Sec. 3.1] can be converted to an MT-authenticator for the ID-based enhanced CK-model after changing the signature scheme to an euf-cma-ida secure ID-based signature scheme. Therefore, we can transform Protocol 2DH in AM to a protocol in UM by applying the ID-based signature-based MT-authenticator of Sec. 4.3 to each of the two messages of Protocol 2DH using the layered approach. By using the optimization technique of [1,11] and eliminating some redundant components, we can obtain the resultant protocol which is identical to Zhu et al.'s protocol reviewed above.

As we can see, by applying the ID-based enhanced CK-model, we retain the beauty of the original CK-model and allow a new ID-based KE protocol to be proven effectively using the modular approach. In addition, the properties of KGS-FS and KCI Resilience are added into the CK-model in a modular approach. In particular, the property of KCI resilience can now be evaluated in authentication protocols in general rather than restricted to KE protocols and KGS-FS are included into the CK-model is a natural and straightforward way with the formality of ID-based setting in place as well.

References

1. M. Bellare, R. Canetti, and H. Krawczyk. A modular approach to the design and analysis of authentication and key exchange protocols. In *Proc. 30th ACM Symp. on Theory of Computing*, pages 419–428. ACM, May 1998. Full paper available at the first author's homepage.
2. M. Bellare, C. Namprempre, and G. Neven. Security proofs for identity-based identification and signature schemes. In *Proc. EUROCRYPT 2004*, pages 268–286. Springer-Verlag, 2004. LNCS 3027 (Full paper is available at Bellare's homepage URL: `http://www-cse.ucsd.edu/users/mihir`).
3. M. Bellare, D. Pointcheval, and P. Rogaway. Authenticated key exchange secure against dictionary attacks. In *Proc. EUROCRYPT 2000*, pages 139–155. Springer-Verlag, 2000. LNCS 1807.
4. M. Bellare and P. Rogaway. Entity authentication and key distribution. In *Proc. CRYPTO 93*, pages 232–249. Springer-Verlag, 1994. LNCS 773.
5. R. Bird, I. Gopal, A. Herzberg, P. Janson, S. Kutten, R. Molva, and M. Yung. Systematic design of two-party authentication protocols. In *Proc. CRYPTO 91*, pages 44–61. Springer, 1992. LNCS 576.
6. S. Blake-Wilson, D. Johnson, and A. Menezes. Key agreement protocols and their security analysis. In *Sixth IMA International Conference on Cryptography and Coding*, pages 30–45. Springer-Verlag, 1997. LNCS 1355.
7. C. Boyd and K.-K. R. Choo. Security of two-party identity-based key agreement. In *Mycrypt 2005*, pages 229–243. Springer-Verlag, 2005. LNCS 3715.
8. C. Boyd, W. Mao, and K. G. Paterson. Key agreement using statically keyed authenticators. In *ACNS 2004*, pages 248–262. Springer-Verlag, 2004. LNCS 3089.
9. C. Boyd and A. Mathuria. *Protocols for Authentication and Key Establishment*. Springer-Verlag, 2003.
10. R. Canetti. Universally composable security: a new paradigm for cryptographic protocols. In *Proc. 42th IEEE Symp. on Foundations of Comp. Science*, pages 136–145, 2001. Full version: Cryptology ePrint Archive, Report 2000/067 (Revised Date: 13 Dec 2005).
11. R. Canetti and H. Krawczyk. Analysis of key-exchange protocols and their use for building secure channels. In *Proc. EUROCRYPT 2001*, pages 453–474. Springer-Verlag, 2001. LNCS 2045. (Full version: `http://eprint.iacr.org/2001/040`).
12. J. C. Cha and J. H. Cheon. An identity-based signature from gap Diffie-Hellman groups. In *Public Key Cryptography 2003*, pages 18–30. Springer-Verlag, 2002. LNCS 2567.
13. L. Chen and C. Kudla. Identity based authenticated key agreement protocols from pairings. Cryptology ePrint Archive, Report 2002/184 (Revised Date: 27 May 2004), 2002. `http://eprint.iacr.org/2002/184`.
14. K. Y. Choi, J. Y. Hwang, D. H. Lee, and I. S. Seo. ID-based authenticated key agreement for low-power mobile devices. In *Information Security and Privacy, 10th Australasian Conference (ACISP 2005)*, pages 494–505. Springer, 2005. LNCS 3574.
15. Y. J. Choie, E. Jeong, and E. Lee. Efficient identity-based authenticated key agreement protocol from pairings. *Applied Mathematics and Computation*, 162(1), 2005.
16. W. Diffie, P. C. Van Oorschot, and M. J. Wiener. Authentication and authenticated key exchanges. *Designs, Codes, and Cryptography*, 2(2):107–125, June 1992.

17. C. Günther. An identity-based key exchange protocol. In *Proc. EUROCRYPT 89*, pages 29–37. Springer-Verlag, 2000. LNCS 434.
18. M. Just and S. Vaudenay. Authenticated multi-party key agreement. In *Proc. ASIACRYPT 96*, pages 36–49. Springer-Verlag, 1996. LNCS 1163.
19. J. Katz, R. Ostrovsky, and M. Yung. Forward secrecy in password-only key exchange protocols. In *Proc. of SCN 2002*, pages 29–44. Springer-Verlag, 2002. LNCS 2576.
20. H. Krawczyk. HMQV: a high-performance secure Diffie-Hellman protocol. In *Proc. CRYPTO 2005*, pages 546–566. Springer-Verlag, 2005. LNCS 3621 (Full version: Cryptology ePrint Archive, Report 2005/176 (Revised Date: 5 Jul 2005)).
21. C. Kudla and K. G. Paterson. Modular security proofs for key agreement protocols. In *Proc. ASIACRYPT 2005*, pages 549–565. Springer-Verlag, 2005. LNCS 3788.
22. N. McCullagh and P. S. L. M. Barreto. A new two-party identity-based authenticated key agreement. In *CT-RSA 2005*, pages 262–274. Springer-Verlag, 2005. LNCS 3376.
23. T. Okamoto. Provably secure and practical identification schemes and corresponding signature schemes. In *Proc. CRYPTO 92*, pages 31–53. Springer-Verlag, 1993. LNCS 740.
24. C. P. Schnorr. Efficient identification and signatures for smart cards. In *Proc. CRYPTO 89*, pages 239–252. Springer, 1990. LNCS 435.
25. A. Shamir. Identity-based cryptosystems and signature schemes. In *Proc. CRYPTO 84*, pages 47–53. Springer, 1984. LNCS 196.
26. K. Shim. Efficient id-based authenticated key agreement protocol based on Weil pairing. *IEE Electronics Letters*, 39(8):653–654, April 2003.
27. N. Smart. Identity-based authenticated key agreement protocol based on Weil pairing. *IEE Electronics Letters*, 38(13):630–632, June 2002.
28. X. Tian and D. S. Wong. Session corruption attack and improvements on encryption based MT-authenticators. In *CT-RSA 2006*, pages 34–51. Springer-Verlag, 2006. LNCS 3860.
29. Y. Wang. Efficient identity-based and authenticated key agreement protocol. Cryptology ePrint Archive, Report 2005/108, 2005. `http://eprint.iacr.org/2005/108`.
30. R. W. Zhu, G. Yang, and D. S. Wong. An efficient identity-based key exchange protocol with KGS forward secrecy for low-power devices. In *The 1st Workshop on Internet and Network Economics (WINE 2005)*, pages 500–509. Springer-Verlag, 2005. LNCS 3828.

A The Canetti-Krawczyk Model (CK-Model)

We review the model proposed by Canetti and Krawczyk [11] to the extent that it is sufficient for understanding the rest of the paper. For full details, we refer readers to the full papers of [1,11].

A.1 Message Driven and Key Exchange (KE) Protocols

The CK-model is defined over message driven protocols. An n-party message driven protocol is a collection of n programs, each of them is run by a different party. A program is first invoked with some initial input which includes a security parameter and some random input. Once invoked, it can be activated by two types of events:

1. The arrival of an incoming message.
2. An action request which models information coming from other programs run by the party. Typical examples of action requests include requesting for sending a message or triggering a key exchange request with some specified party. For each action request, the response of the program should be defined explicitly in the protocol specification.

At the end of each activation, a local output value is generated.

A Key Exchange (KE) protocol is a message driven protocol. Each execution of the KE protocol is modeled as a series of activations on two parties, P_i and P_j. For example, after all the programs of parties have been invoked, the program running in P_i will take an action request of the form establish session$(P_i, P_j, s, role)$ where P_j is the party with whom the key is to be exchanged, s is the session ID, and $role$ is either initiator or responder. The program of P_i will then fork a sub-process and ask it to handle the subsequent execution procedure of the protocol. The sub-process is called a KE-session with input $(P_i, P_j, s, role)$. After the sub-process is forked, if P_i is activated by an action request to send a message m to P_j in session s (i.e. there is a send-message action request received with inputs m, P_j and s), the corresponding KE-session will handle it. The KE-session is *completed* when the corresponding sub-process returns with output (P_i, P_j, s, κ) where κ is a non-null *session key*. The KE-session can label κ as 'secret' which prevents an adversary from getting the value of it. When a session *completes*, the internal state of the corresponding sub-process is assumed to be erased.

In this model, each program running in a party can fork multiple sub-processes to handle multiple KE-sessions. If in one KE protocol execution, 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*.

The Initialization Function I. This function in a message driven protocol models a perfectly secure bootstrapping of cryptographic functions. The function takes a random input r and a security parameter k, and outputs a vector $I(r, k) = I(r, k)_0 \cdots I(r, k)_n$, where $I(r, k)_0$ will become known to all parties and the adversary; $I(r, k)_i$, for $i > 0$, will become known only to party P_i.

A.2 Two Adversarial Models: UM and AM

For capturing various capabilities and activities of adversaries, in the CK-model, there are two adversarial models defined. They are the *unauthenticated-links model (UM)* and the *authenticated-links model (AM)*.

The Unauthenticated-links Adversarial Model (UM). Consider a message driven protocol π with n parties $P_1, \cdots, P_n$. A UM-adversary $\mathcal{U}$ is a PPT (probabilistic polynomial-time) Turing machine which controls all the communication links of the system and schedules almost all the protocol activities, including party activation and message delivery (except the protocol initialization which is controlled by the initialization function I). In particular, $\mathcal{U}$ can activate π within some party P_i by either incoming messages or action requests. $\mathcal{U}$ can also arbitrarily generate, inject, modify, and deliver any messages of his own will. In addition, $\mathcal{U}$ can access all the local output of a party except a secret output, e.g. the session key of a key exchange session. The adversary $\mathcal{U}$ can also do the following activities by making some appropriate queries.

- party corruption on P_i: $\mathcal{U}$ learns all the internal state information of P_i. It includes the long-term secret of P_i and all the state information of its sessions. P_i will no longer be activated and does not generate further local output. P_i is said to be *corrupted*.
- session-state reveal: $\mathcal{U}$ learns all the current state of a specified session.
- session-output reveal: $\mathcal{U}$ learns all outputs that are labeled 'secret', from a specified session.

The global output denoted by $UNAUTH_{\pi,\mathcal{U}}$ of a protocol is the cumulative output of $\mathcal{U}$ and $P_1, \cdots, P_n$. The output of the adversary is specified in the protocol specification. For example, in the definition of session-key security in Sec. A.4, the output of $\mathcal{U}$ is its guess for the coin value tossed by the game simulator.

Remark: For KE protocols, the session-key reveal query is used instead of the session-output reveal query and an additional adversarial activity called session expiration query is added. A session-key reveal query can be scheduled by $\mathcal{U}$ for a *completed* KE-session. In this case, $\mathcal{U}$ learns the session key of the specific KE-session. $\mathcal{U}$ can also issue a session expiration query for any *completed* KE-session. In this case, the session key is erased from the party's memory and $\mathcal{U}$ is no longer allowed to issue a session-key reveal query on an *expired* KE-session.

Session Exposure. A KE-session with input $(P_i, P_j, s, role)$ is called *locally exposed* if $\mathcal{U}$ performs any of the following activities: (1) a session-state reveal (2) a session-key reveal (3) a party corruption on P_i before the session expired. However, a party corruption on P_i *after* session $(P_i, P_j, s, role)$ expired is not taken into account. A KE-session is called *exposed* if either the session or its matching session is *locally exposed*.

The Authenticated-links Adversarial Model (AM). An AM-adversary $\mathcal{A}$ is similar to the UM-adversary $\mathcal{U}$ while $\mathcal{A}$ is not allowed to inject or modify messages unless the message sender is corrupted or the message belongs to an exposed session. $\mathcal{A}$ is restricted to deliver messages faithfully, and each message can only be delivered *at most* once. Also note that the sender of a message can never be changed by $\mathcal{A}$ even if the origin session is exposed (which includes the case that the sender is corrupted). The global output denoted by $AUTH_{\pi,\mathcal{A}}$ is the cumulative output of $\mathcal{A}$ and $P_1, \cdots, P_n$.

Definition 4 ([11]). *Let π and π' be two n-party message driven protocols. We say that π' emulates π in UM if given any adversary $\mathcal{U}$ in UM against protocol π', there exists an adversary $\mathcal{A}$ in AM 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.

A.3 Authenticators

An authenticator $\mathcal{C}$ is an algorithm that for any protocol π in AM, the protocol $\mathcal{C}(\pi)$ emulates π in UM. One way of constructing an authenticator is given in [1], where a layered approach is used. According to [1, Theorem 3], an authenticator $\mathcal{C}_\lambda$, which is called a **layered authenticator** in [1], can be constructed from an MT-authenticator λ which emulates the basic message transmission (MT) protocol. The basic idea is that whenever a party P_i wants to send or receive a message (using an MT protocol) in AM, the MT-authenticator emulates it in UM using λ. Formally, an MT protocol in AM is carried out as follows. Upon activation within a party P_i on an action request of the form send(P_i, P_j, s, m), P_i sends the message (P_i, P_j, s, m) to party P_j, and outputs "P_i sent m to P_j in session s". Upon receipt of a message (P_i, P_j, s, m), P_j outputs "P_j received m from P_i in session s". Suppose we have an MT-authenticator λ which emulates this MT protocol. For a protocol π, a layered authenticator $\pi' = \mathcal{C}_\lambda(\pi)$ can be constructed as follows. For each message that π sends, λ is invoked and activated with an action request for sending that message to the corresponding recipient in UM. When

π' is activated with some incoming message, $\mathcal{C}_\lambda$ also activates λ with the corresponding incoming message. When λ outputs, for example, "P_j received m from P_i in session s", π is activated with incoming message m from P_i.

It is assumed that each message transmitted in the network contains the identities of the sender and the receiver, as well as the session IDs of the sender's and the receiver's sessions. Throughout this paper, we also assume that the sender and the receiver share the same session ID and each message contains only one copy of it. When the identities of the sender and the receiver are implicitly specified in the context, we omit them in our description. In the original papers of [1,11], the authors use m to denote a message which comprises both the actual message content and a session ID. This causes certain inconvenience in presentation, as an attacker may choose to modify only the session ID or only the actual message content. In this paper, we adopt a more explicit approach. We use m to denote the actual content of a message only. It does not include sender or receiver identity, or the session ID.

A.4 Session-Key (SK) Security

For defining the session key security of a KE protocol, the capability of the adversary ($\mathcal{U}$ in UM or $\mathcal{A}$ in AM) is extended by allowing it to make a test-session query on a KE-session that is *completed*, *unexpired* and *unexposed*. Let κ be the session key (labeled as 'secret' in the session output) of the queried KE-session. A coin $b \in_R \{0,1\}$ is tossed by the game simulator. If $b = 0$, κ is returned to the adversary; otherwise, a random value chosen according to the probability distribution of session keys of the protocol is returned. The adversary can still carry out regular actions but is not allowed to expose the test-session (namely, neither of the test-session and its matching session can be locally exposed). Note that the adversary will be allowed to corrupt a partner to the test-session as soon as the test-session (or its matching session) expires at that party. This helps capture the forward secrecy property of a KE protocol. At the end of its run, the adversary outputs a bit b' (as its guess for b).

Definition 5 ([11]). *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 session key;*
2. *The probability that the adversary guesses correctly the bit b' (i.e., $b' = b$) is no more that $1/2$ plus a negligible fraction in the security parameter.*

According to [11, Theorem 6], it states that if π is a SK-secure KE protocol in AM and λ is an MT-authenticator, $\pi' = \mathcal{C}_\lambda(\pi)$ is a SK-secure KE protocol in UM. Therefore, a modular approach of using SK-secure KE protocol in AM and MT-authenticators to the design of SK-secure KE protocols in UM is obtained. Also, the MT-authenticators can be reused to construct new KE protocols.

B Proof of Theorem 3

Proof. We target to prove that for any ID-based enhanced UM-adversary $\tilde{\mathcal{U}}$ against λ_{SIG}^{ID}, there exists an ID-based enhanced AM-adversary $\tilde{\mathcal{A}}$ against MT protocol such that $AUTH_{MT,\tilde{\mathcal{A}}}$ and $UNAUTH_{\lambda_{SIG}^{ID},\tilde{\mathcal{U}}}$ are computationally indistinguishable.

$\tilde{\mathcal{A}}$ runs $\tilde{\mathcal{U}}$ on a simulated interaction with a set of N parties and the KGS, that are running λ_{SIG}^{ID}. For any party P (or session s, KGS) in AM, we denote the corresponding imitated party (or imitated session, imitated KGS) in UM as P' (or s', KGS'). The initialization function I is first invoked according to the description of λ_{SIG}^{ID} above. $\tilde{\mathcal{A}}$ then proceeds as follows:

1. Whenever $\tilde{\mathcal{U}}$ activates KGS' with a create party query to create some party A' with identity $ID_{A'}$ in UM, $\tilde{\mathcal{A}}$ activates KGS with a create party query to create the corresponding party A with identity ID_A in AM accordingly.
2. Whenever $\tilde{\mathcal{U}}$ activates a party A' with an action request to send message m to party B' in UM, $\tilde{\mathcal{A}}$ activates the corresponding party A with the same action request to send m to B in AM. Recall that this entails the tuple (A, B, s, m) being put into set M of undelivered messages.
3. Whenever B' outputs "B' received m from A' in session s'"in UM, $\tilde{\mathcal{A}}$ activates session s in B with the incoming message m from A in AM, provided that the tuple (A, B, s, m) is in set M. Then, (A, B, s, m) is deleted from set M.
4. Whenever $\tilde{\mathcal{U}}$ issues a party corruption query on a party A' in UM, $\tilde{\mathcal{A}}$ issues a party corruption query on A accordingly in AM. Then, $\tilde{\mathcal{A}}$ hands all the current internal information (i.e. the long-term secret, session states and secret session outputs) of A to $\tilde{\mathcal{U}}$.
5. Whenever $\tilde{\mathcal{U}}$ issues a session-state reveal query on a session s' within a party A' in UM, $\tilde{\mathcal{A}}$ also issues a session-state reveal query on the corresponding session s within A in AM accordingly. $\tilde{\mathcal{A}}$ then hands the current session state of s in A to $\tilde{\mathcal{U}}$. Note that if the corresponding session s has not been generated within A, $\tilde{\mathcal{A}}$ will first generate it and then issue a session-state reveal query on s.
6. Whenever $\tilde{\mathcal{U}}$ issues a session-output reveal query on a session s' within a party A' in UM, $\tilde{\mathcal{A}}$ also issues a session-output reveal query on the corresponding session s within A in AM. $\tilde{\mathcal{A}}$ then hands any output labeled secret of session s in A to $\tilde{\mathcal{U}}$.
7. Whenever $\tilde{\mathcal{U}}$ issues a key compromise query on a party A' in UM, $\tilde{\mathcal{A}}$ issues a key compromise query on A in AM. $\tilde{\mathcal{A}}$ then hands the long-term secret of A to $\tilde{\mathcal{U}}$.
8. Whenever $\tilde{\mathcal{U}}$ issues a corrupt KGS query on KGS' in UM, $\tilde{\mathcal{A}}$ issues a corrupt KGS query on KGS accordingly in AM. Then, $\tilde{\mathcal{A}}$ hands the master key of KGS to $\tilde{\mathcal{U}}$.
9. $\tilde{\mathcal{A}}$ outputs whatever $\tilde{\mathcal{U}}$ outputs.

As in the proof of [1, Proposition 4], we can see that an obstacle on the simulation above occurs only if the following event $\mathcal{B}$ happens.

Let $\mathcal{B}$ be the event that, for party A' (*created, uncorrupted, uncompromised*) and party B' (*created, uncorrupted*), B' outputs "B' received m from A' in session s'" in UM, and either A has not been activated by $\tilde{\mathcal{A}}$ to send m to B, or B has output "B received m from A in session s" before in AM. We need to show that event $\mathcal{B}$ happens only with negligible probability. This can be done by contradiction. Suppose that event $\mathcal{B}$ happens with non-negligible probability, then we may construct a forger $\mathcal{F}$ who can break the euf-cma-ida secure ID-based signature scheme with non-negligible probability. The technical details follow directly from that in [1, Proposition 4].

In event $\mathcal{B}$ above, note that B' can be *compromised* through a key compromise query. This implies that the adversary can obtain the long-term secret of B', but still being unable to masquerade to B' as another party A', provided that A''s long-term secret is not compromised. □