Deniable Internet Key Exchange*

Andrew C. Yao[a] and Yunlei Zhao[b]**

[a]ITCS, Tsinghua University, Beijing, China
[b]Software School, Fudan University, Shanghai, China

Abstract. In this work, we develop a family of *non-malleable* and *deniable* Diffie-Hellman key-exchange (DHKE) protocols, named deniable Internet key-exchange (DIKE). The newly developed DIKE protocols are of conceptual clarity, provide much remarkable privacy protection to protocol participants, and are of highly practical (online) efficiency.

For the security of the DIKE protocols, we formulate the notion of *tag-based robust non-malleability* (TBRNM) for DHKE protocols, which ensures robust non-malleability for DHKE protocols against concurrent man-in-the-middle (CMIM) adversaries and particularly implies concurrent forward deniability for both protocol participants. We show that the TBRNM security and the session-key security (SK-security) in accordance with the Canetti-Krawczyk framework are mutually complementary, thus much desirable to have DHKE protocols that enjoy both of them simultaneously. We prove our DIKE protocol indeed satisfies both (privacy preserving) TBRNM security and SK-security (with post-specified peers). The TBRNM analysis is based on a variant of the knowledge-of-exponent assumption (KEA), called concurrent KEA assumption introduced and clarified in this work, which might be of independent interest.

1 Introduction

The Internet Key-Exchange (IKE) protocols [21, 22] are the *core* cryptographic protocols to ensure Internet security, which specifies key exchange mechanisms used to establish shared keys for use in the Internet Protocol Security (IPsec) standards [23]. The IPsec and IKE are intended to protect messages communicated in the IP layer, i.e., "layer 3" of ISO-OSI, which process the transmission of messages using the network addresses *possibly without knowing end-user peers' identities*. The IKE and IPsec can in turn be used to offer confidentiality, authentication and privacy for communication protocols in the higher layers of ISO-OSI.

The standard of IKE key-exchange has gone through two generations. The first generation IKEv1 [21] uses public-key encryption as the authentication mechanism, and the IKEv2 [22] uses signatures as the authentication mechanism with the SIGMA protocol [24] serving as the basis.

The IKEv2 protocol is based on DHKE [13], and works in the "post-specified peer" setting [22], where the information of who the other party is does not necessarily exist at

* This work was supported in part by the National Basic Research Program of China Grant Nos.2007CB807900, 2007CB807901, the National Natural Science Foundation of China Grant Nos.60553001, 60703091, and the QiMingXing Program of Shanghai. Preliminary version of this work appeared in Cryptology ePrint Archive 2007/191.

** Contact author. ylzhao@fudan.edu.cn

the session initiation stage and is learnt by the party only after the protocol run evolves. Actually, this is quite a common case for KE protocols in practice, particularly for the purpose of preserving players' privacy. For example, the key-exchange session may take place with any one of a set of servers sitting behind a (url/ip) address specified in the session activation; Or, a party may respond to a request (for a KE session) coming from a peer that is not willing to reveal its identity over the network and, sometimes, even not to the responder before the latter has authenticated itself (e.g., a roaming mobile user connecting from a temporary address, or a smart-card that authenticates the legitimacy of the card-reader before disclosing its own identity) [7].

For key-exchange protocols, both security and privacy are desired. Among privacy concerns, deniability is an essential privacy property, and has always been a central concern in personal and business communications, with off-the-record communication serving as an essential social and political tool [16, 12, 14]. Given that many of these interactions now happen over digital media (email, instant messaging, web transactions, virtual private networks), it is of critical importance to provide these communications with "off-the-record" or deniability capability to protocol participants.

A protocol is called *forward* deniable, if it ensures deniability for both the sender and the receiver simultaneously. Forward deniability essentially implies that the protocol is *statistical* zero-knowledge (ZK) [19] for both the sender and the receiver, in the sense that both the view of the sender and that of the receiver can be statistically simulated by an efficient algorithm alone without any interactions.

Whenever deniability of messages is desired, in general, we can just run a forward deniable authentication protocol [16] for each message to be sent. However, the beauty of using forward deniable key-exchange is that if the key-exchange protocol is deniable, then all the transactions (of *public* messages) using the session-key produced by the key-exchange protocol can be deniable (i.e., simulatable) for both the protocol participants. Moreover, for the IKE protocol that is the core cryptographic protocol to ensure Internet security, offering deniability by IKE running at the IP layer within the IPsec standard [23] is much more desirable, because it enables various privacy services to be offered at the higher layers with uncompromised quality. Note that a privacy problem at the IP layer can cause irreparable privacy damage at the application layer. For example, an identity connected to an IP address, if not deniable, certainly nullifies an anonymous property offered by a fancy cryptographic protocol running at the application level. (If deniability is not desired, for some cases, then a non-repudiable proof, e.g., a signature, can always be issued at the application level.)

1.1 Our contributions

In this work, we develop a family of *non-malleable* [15] and *deniable* DHKE protocols, named deniable Internet key-exchange (DIKE), which adds novelty and new value to the IKE key-exchange standard [21, 22] and the SIGMA protocol [24]. The newly developed DIKE protocols are of conceptual clarity, provide much remarkable privacy protection to protocol participants, are of highly practical (online) efficiency, and of well compatibility with the IKEv2 and SIGMA protocols.

For the security of the DIKE protocols, we formulate the notion of *tag-based robust non-malleability* (TBRNM) for Diffie-Hellman key-exchange protocols, which ensures robust non-malleability for DHKE protocols against concurrent man-in-the-

middle (CMIM) adversaries. Roughly speaking, TBRNM says that a CMIM adversary can successfully finish a session of a DHKE protocol only if it does know both the secret-key and the DH-exponent corresponding to the public-key and the DH-component alleged and sent by the CMIM adversary for that session. The TBRNM formulation takes security and privacy in an integrity, which particularly implies concurrent forward deniability (actually, concurrent non-malleable statistical zero-knowledge CNMSZK) for both the protocol initiator and the protocol responder. We show that the TBRNM security and the session-key security (SK-security) formulated in the Canetti-Krawczyk framework (CK-framework) [6] are mutually complementary, thus much desirable to have DHKE protocols that enjoy both of them simultaneously.

We prove our DIKE protocol is indeed both TBRNM secure and SK-secure (with post-specified peers). The TBRNM analysis is conducted in the *restricted* random oracle (RO) model introduced by Yung, et al [41], in order to bypass the subtleties of deniability loss for simulation with unrestricted ROs [33, 35, 41], and is based on a variant of the knowledge-of-exponent assumption (KEA) [9]. In particular, we revisit the KEA assumption, demonstrate and clarify the subtleties and insufficiency of employing the KEA assumption to argue the security of DH-based *interactive* cryptographic protocols when they are run *concurrently in the public-key model* (as is the focus of this work). This motivates us to introduce a new extended KEA assumption, called concurrent KEA (CKEA) assumption. Interestingly, the CKEA assumption can be viewed as the non-black-box counterpart of the gap Diffie-Hellman (GDH) assumption [34], while the original KEA assumption is that of the computational Diffie-Hellman (CDH) assumption. As we shall show, the CKEA-based approach for achieving concurrent non-malleability and deniability can be a useful paradigm in DH-based cryptographic practice, with a reasonable trade-off between practical efficiency and formal provable security. The SK-security (with post-specified peers) of our DIKE protocol is proved under the GDH assumption in the RO model.

2 Preliminaries

If A is a probabilistic algorithm, then $A(x_1, x_2, \cdots ; r)$ is the result of running A on inputs $x_1, x_2, \cdots$ and coins r. We let $y \leftarrow A(x_1, x_2, \cdots)$ denote the experiment of picking r at random and letting y be $A(x_1, x_2, \cdots ; r)$. If S is a finite set then $x \leftarrow S$ is the operation of picking an element uniformly from S. If α is neither an algorithm nor a set then $x \leftarrow \alpha$ is a simple assignment statement.

On a system parameter n (also written as 1^n), a function $\mu(\cdot)$ is negligible if for every polynomial $p(\cdot)$, there exists a value N such that for all $n > N$ it holds that $\mu(n) < 1/p(n)$. Let $X = \{X(n,z)\}_{n\in N, z\in\{0,1\}^*}$ and $Y = \{Y(n,z)\}_{n\in N, z\in\{0,1\}^*}$ be distribution ensembles. Then we say that X and Y are computationally (resp., statistically) indistinguishable, if for every probabilistic polynomial-time (resp., any power-unbounded) algorithm D, for all sufficiently large n's, and every $z \in \{0,1\}^*$, $|\Pr[D(n,z,X(n,z)) = 1] - \Pr[D(n,z,Y(n,z)) = 1]|$ is negligible in n.

Let G' be a finite Abelian group of order N, and $G = \langle g \rangle$ be a unique subgroup of G', generated by the generator g, of prime order q which is ensured by requiring $gcd(t, q) = 1$ for $t = N/q$. Denote $Z_q = \{0, 1, \cdots, q-1\}$ and $Z_q^* = \{1, 2, \cdots, q-1\}$, denote by 1_G the identity element of G' and by $G/1_G$ the set of elements of G except

1_G. In the specification of this paper, w.l.o.g., we assume G' is the multiplicative group Z_p^* of order $N = p - 1$ for a large prime p, and G is the unique subgroup of order q for some prime divisor q of $N = p - 1$. Typically, the length of p (i.e., the length of group element for a DL-based cryptographic system), denoted $|p| = n$, is treated as the *system* parameter, and the length of q, denoted $|q| = k$, is treated as the *security* parameter. The value $t = (p - 1)/q$ is called the *cofactor*. The specification can be trivially applicable to the groups based on elliptic curves. In elliptic curve systems, G' is the group of points $E(L)$ on an elliptic curve E defined over a finite field L, and G is a subgroup of $E(L)$ of prime order q. For elliptic curve based groups, the cofactor t is typically very small.

Let $H, H_K : \{0,1\}^* \rightarrow \{0,1\}^{|q|}$ be hash functions, which are modeled as random oracles in security analysis. Here, for presentation simplicity, we have assumed H, H_K are of the same output length. In practice, they may be of different output lengths.

Definition 1 (Computational Diffie-Hellman (CDH) assumption). *Let G be a cyclic group of prime order q generated by an element g, for two elements $X = g^x$, $Y = g^y$ in G, where $x, y \in Z_q$, we denote by $CDH(X,Y) = g^{xy \mod q} \mod p$ (the mod operation is usually omitted for presentation simplicity). An algorithm is called a* CDH solver for *G if it takes as input pairs of elements (X,Y) (and also a generator g of G) and outputs the value of $CDH(X,Y)$. We say the CDH assumption holds in G if for any probabilistic polynomial-time (PPT) CDH solver, the probability that on a pair (X,Y), for $X, Y \leftarrow G$ (i.e., each of x and y is taken uniformly at random from Z_q), the solver computes the correct value $CDG(X,Y)$ is negligible. The probability is taken over the random coins of the solver, and the choice of X, Y uniformaly at random in G.*

The gap DH assumption (GDH) [34] essentially says that in the group G, computing $CDH(X,Y)$, for $X, Y \leftarrow G$, is strictly harder than deciding if $Z = CDH(U,V)$ for an arbitrary triple $(U,V,Z) \in G^3$.

Definition 2 (Gap Diffie-Hellman (GDH) assumption [34]). *Let G be a cyclic group generated by an element g, and a decision predicate algorithm $\mathcal{O}$ be a (*full*) Decisional Diffie-Hellman (DDH) Oracle for the group G and generator g such that on input (U,V,Z), for* arbitrary *$(U,V) \in G^2$, oracle $\mathcal{O}$ outputs 1 if and only if $Z = CDH(U,V)$. We say the GDH assumption holds in G if for any PPT CDH solver for G, the probability that on a pair of random elements $(X,Y) \leftarrow G$ the solver computes the correct value $CDG(X,Y)$ is negligible, even when the algorithm is provided with the (full) DDH-oracle $\mathcal{O}$ for G. The probability is taken over the random coins of the solver, and the choice of X, Y (each one of them is taken uniformly at random in G).*

Definition 3 (Knowledge-of-Exponent Assumption (KEA) [9, 25]). *Let G be a cyclic group of prime order q generated by an element g, and consider algorithms that on input a triple $(g, C = g^c, z)$ output a pair $(Y,Z) \in G^2$, where c is taken uniformly at random from Z_q^* and $z \in \{0,1\}^*$* is an arbitrary string that is generated independently of C. *Such an algorithm $\mathcal{A}$ is said to be a KEA algorithm if with non-negligible probability (over the choice of g, c and $\mathcal{A}$'s random coins) $\mathcal{A}(g, g^c, z)$ outputs $(Y,Z) \in G^2$ such that $Z = Y^c$. Here, $C = g^c$ is the random challenge to the KEA algorithm $\mathcal{A}$, and z captures the auxiliary input of $\mathcal{A}$ that is independent of the challenge C.*

We say that the KEA assumption holds over G, if for every efficient (probabilistic polynomial-time) KEA algorithm $\mathcal{A}$ for G there exists another efficient algorithm $\mathcal{K}$,

referred to as the KEA-extractor, for which the following property holds except for a negligible probability: let (g, g^c, z) *be an input to* $\mathcal{A}$ *and* ρ *a vector of random coins for* $\mathcal{A}$ *on which* $\mathcal{A}$ *outputs* $(Y, Z = Y^c)$*, then on the same inputs and random coins* $\mathcal{K}(g, C, z, \rho)$ *outputs the triple* $(Y, Z = Y^c, y)$ *where* $Y = g^y$*.*

The KEA assumption is derived from the CDH assumption, and is a *non-black-box* assumption by nature [1]. Since its introduction in [9], the KEA assumption has been used in a large body of works, particularly in the literature of deniable authentication and key-exchange (e.g., [20, 2, 1, 11, 25, 10, 12, 38, 39], etc).

3 DIKE Implementation and Advantageous Features

Let $(A = g^a, a)$ (resp., $(X = g^x, x)$) be the public-key and secret-key (resp., the DH-component and DH-exponent) of the initiator $\hat{A}$, and $(B = g^b, b)$ (resp., $(Y = g^y, y)$) be the public-key and secret-key (resp., the DH-component and DH-exponent) of the responder player $\hat{B}$, where a, x, b, y are taken randomly and independently from Z_q^*.

The deniable Internet key-exchange protocol, for the main model of [21–23], is depicted in Figure 1 (page 6), where $CERT_{\hat{A}}$ (resp., $CERT_{\hat{B}}$) is the public-key certificate of $\hat{A}$ (resp., $\hat{B}$) issued by some trusted Certificate Authority (CA) within the underlying public-key infrastructure (PKI), and sid is the session-identifier that is assumed to be set by some "higher layer" protocol that "calls" the KE protocol and ensures no two sessions run at a party are of identical session-identifier [7]. Throughout this work, we assume no proof-of-knowledge/possession (POK/POP) of secret-key is mandated during public- key registration, but the CA will check the non-identity sub-group (i.e., $G/1_G$) membership of registered public-keys. Also, each party checks the $G/1_G$ membership of the DH-component from its peer.

3.1 Some advantageous features of DIKE

Our DIKE enjoys remarkable privacy protection for both protocol participants. Note that all authentic messages, $NMZK(\hat{B}, y)$ and $NMZK(b, y)$ (resp., $NMZK(a, x)$), from $\hat{B}$ (resp., $\hat{A}$) can be computed *merely* from its peer's DH-exponent x (resp., y) and one's own public messages; Furthermore, one party sends the authentic messages involving its secret-key only after being convinced that its peer does "know" the corresponding DH-exponent. This ensures forward deniability for both the protocol participants. IKEv2 and SIGMA do not enjoy these privacy properties, due to the underlying signatures used. Note also that the DIKE protocol works in the post-specified-peer setting, and the messages from one party do not bear the information of its peers's ID and public-key.

Besides some hashing operations and the validation of peer's public-key certificate, the player $\hat{A}$ computes (Y^q, Y^a, Y^x) and (X, B^x), the player $\hat{B}$ computes (X^q, X^b, X^y) and (Y, A^y). Note that the computation of (Y^q, Y^a, Y^x) (resp., (X^q, X^b, X^y)) *in parallel* actually amounts to about 1.5 modular exponentiations. The DH-component X (resp., Y) can always be off-line pre-computed by $\hat{A}$ (resp., $\hat{B}$). Moreover, if the peer's identity is pre-specified, $\hat{A}$ (resp., $\hat{B}$) can further off-line pre-compute the value B^x (resp., A^y). That is, the total computational complexity at each player side is about 3.5 exponentiations, and the on-line computational complexity at each player side can remarkably be only 1.5 exponentiations. We note that if the underlying signatures used

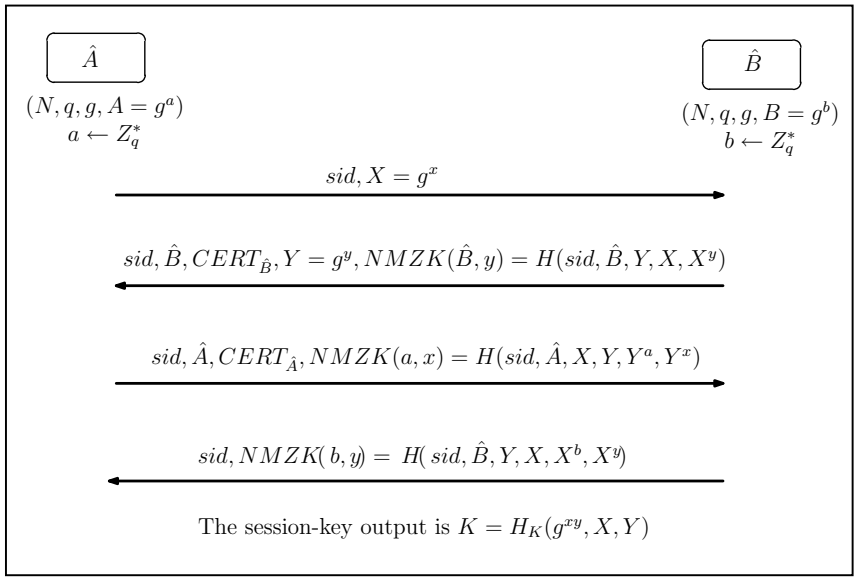

Fig. 1. Deniable Internet Key-Exchange (the main model)

in SIGMA are implemented with the Digital Signature Standard (DSS) [17], the computational complexity of SIGMA is about 4.5 exponentiations at each player side in total, and the online complexity is about 2.5 exponentiations (with offline partial signature generation). For communication complexity, by waiving the use and exchanges of signatures, our deniable IKE is of improved communication complexity, in comparison with that of SIGMA.

Our DIKE protocol is of well compatibility with IKEv2/SIGMA and the (H)MQV protocols [26, 25, 31]. By compatibility with SIGMA/IKEv2, we mean that in case some players are not of discrete logarithm (DL) public-keys, they still can use the Sign-then-MAC mechanism of SIMGA/IKEv2 to authenticate messages from them. In more details, in this case, any one of the last two messages in our deniable IKE can be replaced by the corresponding message flow in SIGMA/IKEv2. By compatibility with (H)MQV, we mean that both (H)MQV and DIKE work for players of DL public-keys, and can be of the same system parameters.

4 Security Formulation and Analysis

In this section, we formulate tag-based robust non-malleability for DHKE protocols based on the CNMZK argument-of-knowledge (CNMZKAOK) formulation [19, 36, 40], investigate the subtleties of the KEA assumption for arguing the security of DH-based interactive protocols running concurrently in the public-key model and introduces the CKEA assumption, and finally show both the TBRNM security and the SK-security of the DIKE protocol.

4.1 Formulating (privacy-preserving) TBRNM for DHKE protocols

We consider an adversarial setting, where polynomially many instances (i.e., sessions) of a DHKE protocol $\langle \hat{A}, \hat{B} \rangle$ are run concurrently over an asynchronous network like the Internet. To distinguish concurrent sessions, each session run at the side of an uncorrupted player is labeled by a tag, which is the concatenation, in the order of session initiator and then session responder, of players' identities, public-keys, and DH-components available from the session transcript; A session-tag is complete if it consists of a complete set of all these components, e.g., $(\hat{A}, A, X, \hat{B}, B, Y)$. Informally speaking, two sessions are matching if they are of the same session-tag.

We assume all communication channels, among all the concurrent sessions of $\langle \hat{A}, \hat{B} \rangle$, are unauthenticated and controlled by a PPT (CMIM) adversary $\mathscr{A}$. This means that the honest player instances cannot directly communicate with each other, since all communication messages are done through the adversary. All honest player instances are working independently with independent random tapes in different sessions (but with the same public-key), and answer messages sent by $\mathscr{A}$ promptly. Once a session is finished, the honest players always erase the ephemeral (private) state information generated during the session, and only keep in privacy the session key output. Sessions can also be expired, and for expired sessions the session keys are also erased.

The CMIM adversary $\mathscr{A}$ (controlling all communication channels) can do whatever it wishes. In particular, $\mathscr{A}$ can interact with polynomial number of instances of $\hat{A}$ in the name of any player playing the role of the responder; such sessions are called the left-sessions. At the same time, $\mathscr{A}$ can interact with polynomial number of instances of $\hat{B}$ in the name of any player playing the role of the initiator; such sessions are called the right-sessions. For presentation simplicity, we assume the number of left-sessions is equal to that of right-sessions, which is $s(n)$ for some positive polynomial $s(\cdot)$. The adversary $\mathscr{A}$ can decide to simply relay the messages of honest player instances. But, it can also decide to block, delay, divert, or modify messages arbitrarily at its wish.

The CMIM adversary $\mathscr{A}$ also takes some arbitrary auxiliary input $z \in \{0,1\}^*$, which captures arbitrary information collected/eavesdropped by $\mathscr{A}$ over the network from the executions of arbitrary (*possibly different*) protocols prior to its actual session interactions with the instances of $\hat{A}$ or $\hat{B}$. For example, z may consist of a CDH triple (X, B, g^{xb}) that is collected over the Internet where $B = g^b$ is the public-key of the player $\hat{B}$, or just the secret-key b in case the CMIM attacker ever broke in the machine of $\hat{B}$. But, the auxiliary input z, collected prior to the actual session interactions of $\langle \hat{A}, \hat{B} \rangle$, is assumed to be independent of the ephemeral DH-components to be generated and exchanged by the instances of $\hat{A}$ and $\hat{B}$ (specifically, we can consider an experiment where the ephemeral DH-components to be exchanged by the instances of uncorrupted players are generated only after the auxiliary string z is fixed.)

We denote by $view_{\mathscr{A}}(1^n, \hat{A}, A, \hat{B}, B, z)$ the random variable describing the view of $\mathscr{A}$ in its concurrent interactions with the instances of $\hat{A}$ and $\hat{B}$, which includes the input $(1^n, \hat{A}, A, \hat{B}, B, z)$, $\mathscr{A}$'s random tape, and all messages received in the $s(n)$ left sessions and the $s(n)$ right sessions (for protocols in the RO model, $\mathscr{A}$'s view also includes the RO, see [3] for more details). Here, for presentation simplicity, we have assumed that $\mathscr{A}$ concurrently interacts with any polynomial number of instances of two players: one is the initiator player $\hat{A}$ and one is the responder player $\hat{B}$. In a way,

the two players $\hat{A}$ and $\hat{B}$ (which can be identical) stand for all uncorrupted players in the system. In general, $\mathscr{A}$ can concurrently interact with any polynomial number of instances of any polynomial number of players. Our definitional framework, as well as the security analysis, can be extended to this general setting, by noting that honest players of different public-keys work independently.

Definition 4 (Tag-based robust non-malleability (TBRNM) for DHKE). *A DHKE protocol, $\langle \hat{A}, \hat{B} \rangle$, is called tag-based robust non-malleable, if for any PPT CMIM adversary $\mathscr{A}$ there exists a PPT simulator/extractor S such that for any sufficiently large n, any pair of uncorrupted players $\hat{A}$ and $\hat{B}$ (of public-key A and B respectively), and any auxiliary string $z \in \{0,1\}^*$, the output of $S(1^n, \hat{A}, A, \hat{B}, B, z)$ consists of two parts (str, sta) such that the following hold, where z captures the arbitrary (possibly public-key dependent) information collected by $\mathscr{A}$ prior to its actual session interactions of $\langle \hat{A}, \hat{B} \rangle$ but is independent of the ephemeral messages (particularly, the DH-components) to be generated and exchanged by the instances of $\hat{A}$ and $\hat{B}$:*

- ***Statistical simulatability.*** *The following ensembles are statistically indistinguishable: $\{view_{\mathscr{A}}(1^n, \hat{A}, A, \hat{B}, B, z)\}_{n \in N, \hat{A} \in \{0,1\}^*, A \in G/1_G, \hat{B} \in \{0,1\}^*, B \in G/1_G, z \in \{0,1\}^*}$ and $\{S_1(1^n, \hat{A}, A, \hat{B}, B, z)\}_{n \in N, \hat{A} \in \{0,1\}^*, A \in G/1_G, \hat{B} \in \{0,1\}^*, B \in G/1_G, z \in \{0,1\}^*}$, where $S_1(1^n, \hat{A}, A, \hat{B}, B, z)$ denotes (the distribution of) the first output of S, i.e., str.*
- ***Knowledge extraction.*** *sta consists of a set of $2s(n)$ strings, $\{\tilde{w}_1^l, \tilde{w}_2^l, \cdots, \tilde{w}_{s(n)}^l, \tilde{w}_1^r, \tilde{w}_2^r, \cdots, \tilde{w}_{s(n)}^r\}$, satisfying the following:*
  - *For any i, $1 \leq i \leq s(n)$, if the i-th left-session (resp., right-session) in str is aborted or with a tag identical to that of one of the right-sessions (resp., left-sessions), then $\tilde{w}_i^l = \perp$ (resp., $\tilde{w}_i^r = \perp$);*
  - *Otherwise, i.e., the i-th left-session (resp., right-session) in str is successfully completed and is of session-tag different from those of all right-sessions (resp., left-sessions), then $\tilde{w}_i^l = (\tilde{b}_i^l, \tilde{y}_i^l)$ (resp., $\tilde{w}_i^r = (\tilde{a}_i^r, \tilde{x}_i^r)$), where $\tilde{b}_i^l$ (resp., $\tilde{a}_i^r$) is the discrete-logarithm of the public-key $\tilde{B}_i^l$ (resp., $\tilde{A}_i^r$) set and alleged by the CMIM adversary $\mathscr{A}$ for the i-th left-session (resp., right-session) in the name of $\hat{\tilde{B}}_i^l$ (resp., $\hat{\tilde{A}}_i^r$), and $\tilde{y}_i^l$ (resp., $\tilde{x}_i^r$) is the discrete-logarithm of the DH-component $\tilde{Y}_i^l$ (resp., $\tilde{X}_i^r$) set and sent by the CMIM adversary $\mathscr{A}$ in the i-th left-session (resp., right-session).*

Furthermore, we say the DHKE protocol $\langle \hat{A}, \hat{B} \rangle$ is of privacy-preserving TBRNM, if it additionally satisfies: (1) the transcript of each session can be generated merely from the DH-exponents (along with some public system parameters, e.g., players' public-key and identity information, etc); (2) messages from one party do not bear the identity and public-key information of its peer.

TBRNM vs. SK-security. We make some brief comparisons between TBRNM and the SK-security in accordance with the CK-framework.

- At a high level, the SK-security essentially says that a party that completes a session has the following guarantees [6]: (1) if the peer to the session is uncorrupted then the session-key is unknown to anyone except this peer; (2) if the *unexposed* peer completes a matching session then the two parties have the same shared key.
  Roughly speaking, besides others, TBRNM ensures the enhanced guarantee of the above (2): if the *possibly malicious* peer completes a matching session, then the

two parties, not only, have the same shared key, *but also and more importantly, the (possibly malicious) peer does "*know*" both the DH-exponent (and thus the shared session-key) and the secret-key corresponding to the DH-component and public-key sent and alleged by it in the test-session.* We suggest this kind of security guarantee is very essential to DHKE protocols, particularly when they are run concurrently over the Internet.

- The TBRNM formulation follows the simulation approach [19, 36, 40] of adaptive tag-based CNMZKAOK, which can actually be viewed as an extended and much strengthened version of the latter. In particular, TBRNM implies concurrent forward deniability for both the protocol initiator and the responder. The SK-security definition follows the indistinguishability approach, which particularly does not take deniability into account.
- Recall that the TBRNM formulation is w.r.t. any PPT CMIM adversary of *arbitrary* auxiliary input. In particular, the adversary's auxiliary input can be dependent on player's public-key, e.g., consisting of a CDH triple (X, B, g^{xb}) or just the secret-key b. That is, the TBRNM formulation implicitly captures the adversarial leakage of static secret-keys of uncorrupted players. Static secret-key exposure *for uncorrupted players* was not captured by the SK-security in [6] (static secret-key exposure and party corruption were separately treated in [6]). But, the TBRNM formulation does not take into account the following abilities of the CMIM adversary in: exposing ephemeral private state for incomplete sessions, exposing session-keys for completed sessions, and party corruption, which are however captured by the SK-security in the CK-framework.

From the above clarifications, the TBRNM security and the SK-security can be viewed mutually complementary, and thus it is much desirable to have DHKE protocols that enjoy both the SK-security and the TBRNM security simultaneously.

4.2 KEA assumption revisited, and the CKEA assumption

Subtleties of employing the KEA assumption in the public-key model. Note that, for the KEA assumption in Definition 3, the requirement of independence between the challenge C and the auxiliary input z plays a critical role. For example, when using KEA for provable security of cryptographic protocols running concurrently in the public-key model, in some cases the challenge C is actually the player's *public-key*. In this case, a valid answer $(A, B = A^c)$, with respect to the challenge C, could be just got by an adversary $\mathscr{A}$ from its auxiliary input that models arbitrary information collected/eavesdropped by $\mathscr{A}$ over the network from executions of other (possibly *different*) protocols *before the interaction of the protocol at hand takes place*. Note that, in this case, it is impossible to efficiently extract the value a from the internal state and auxiliary input of the adversary $\mathscr{A}$. This shows that for protocols with provable security based on the KEA assumption w.r.t. *public* challenges, the independence requirement between the auxiliary input z and the public challenge C (corresponding to player's public-key) can significantly limit the composability of the protocol in practice. In other words, in practice it is unrealistic to assume adversary's auxiliary input to be independence of player's public-keys, particularly for protocol running concurrently in the public-key model. To bypass this subtlety of KEA with public challenges and

to render robust composability to cryptographic protocols, in this work we insist using *ephemeral* fresh challenges in designing and analyzing protocols in the public-key model with the KEA assumption.

Subtleties of employing the KEA assumption for interactive *protocols in the* concurrent *setting.* The KEA assumption was originally introduced to argue the (non-malleability) security of public-key encryption (that is a non-interactive cryptographic primitive) [9]. But, when arguing the security of *interactive* protocols running concurrently against CMIM adversaries, we note that, in many scenarios (particularly for DH-based authentication and key-exchange as is the focus of this work), the KEA assumption is insufficient. The reason is that, in such settings, the CMIM adversary can potentially get access to a list of (polynomially many) DDH-oracles, with each being w.r.t. an element taken randomly and independently in G by an honest player instance.

For example, consider a two party protocol $\langle \hat{A}, \hat{B} \rangle$, where $\hat{A}$ generates and sends $X = g^x \in G$ and $\hat{B}$ generates and sends $Y = g^y \in G$; After (or during) the exchange of X and Y, each party uses the shared DH-secret g^{xy} to authenticate some values, and aborts in case the authenticated values from its peer are deemed to be invalid. Now, consider a CMIM adversary who, on a system parameter 1^n, simultaneously interacts with $s(n)$ instances of $\hat{A}$ (by playing the role of $\hat{B}$) and $s(n)$ instances of $\hat{B}$ (by playing the role of $\hat{A}$), where $s(\cdot)$ is a positive polynomial. On an arbitrary value $Z \in G$, a random element X_i generated by $\hat{A}$ (or $\hat{B}$), $1 \leq i \leq s(n)$, and another arbitrary element $Y_j \in G$ where Y_j may also be one of the random elements generated by $\hat{A}$ or $\hat{B}$, the CMIM adversary $\mathscr{A}$ can simply use Z (as the supposed DH-secret) to authenticate a value to the party who sends X_i: if the party aborts, $\mathscr{A}$ concludes $Z \neq CDH(X_i, Y_j)$, otherwise it concludes $Z = CDH(X_i, Y_j)$. This simple protocol example demonstrates that in the concurrent settings for (DH-based) interactive protocols, the CMIM adversary can actually get access to polynomially many DDH-oracles.

The concurrent KEA (CKEA) assumption. The above discussion motivates us to introduce the following assumption, named concurrent knowledge-of-exponents assumption (in reminiscence of the motivation for arguing the *concurrent* security of interactive cryptographic schemes against CMIM adversaries).

Definition 5 (Concurrent knowledge-of-exponents assumption (CKEA)). *Suppose G is a cyclic group of prime order q generated by an element g, 1^n is the system parameter, $p(\cdot)$ and $q(\cdot)$ are positive polynomials. Let a decision predicate algorithm $\mathcal{O}_{\mathcal{C}}$ for $\mathcal{C} = \{C_1 = g^{c_1}, \cdots, C_{p(n)} = g^{c_{p(n)}}\}$ (where c_i, $1 \leq i \leq p(n)$, is taken uniformly at random from Z_q^*) be a DDH-Oracle (w.r.t. the random challenge set $\mathcal{C}$) for the group G and generator g, such that on input (X, Y, Z), for arbitrary $(X, Y) \in G^2$, the oracle $\mathcal{O}_{\mathcal{C}}$ outputs 1 if and only if $X \in \mathcal{C}$ and $Z = CDH(X, Y)$. Consider algorithms that on input a triple $(g, \mathcal{C}, z)$, with oracle access to $\mathcal{O}_{\mathcal{C}}$, output a set of triples $\{(X_1, Y_1, Z_1), \cdots, (X_{q(n)}, Y_{q(n)}, Z_{q(n)})\} \subseteq (G^3)^{q(n)}$, where $z \in \{0,1\}^*$ is an arbitrary string that is generated independently of $\mathcal{C}$. (Specifically, we can consider an experiment where the DH-components in the set $\mathcal{C}$ are generated only after the auxiliary string z is fixed.) Such an algorithm $\mathcal{A}^{\mathcal{O}_{\mathcal{C}}}$ is said to be a CKEA algorithm if with non-negligible probability (over the choice of $g, c_1, \cdots, c_{p(n)}$ and $\mathcal{A}$'s random coins) $\mathcal{A}(g, \mathcal{C}, z)$ outputs $\{(X_1, Y_1, Z_1), \cdots, (X_{q(n)}, Y_{q(n)}, Z_{q(n)})\} \subseteq (G^3)^{q(n)}$ such that $X_i \in \mathcal{C}$ and $Z_i = CDH(X_i, Y_i)$ for all i, $1 \leq i \leq q(n)$.*

We say that the CKEA assumption holds over G, if for every PPT CKEA-algorithm $\mathcal{A}^{\mathcal{O}_\mathcal{C}}$ there exists another efficient PPT algorithm $\mathcal{K}$, referred to as the CKEA-extractor, such that for any polynomials $p(\cdot), q(\cdot)$ and sufficiently large n the following property holds except for a negligible probability: let $(g, \mathcal{C}, z)$ be the input to $\mathcal{A}^{\mathcal{O}_\mathcal{C}}$, ρ a vector of random coins for $\mathcal{A}^{\mathcal{O}_\mathcal{C}}$ and ϖ a vector of answers given by $\mathcal{O}_\mathcal{C}$ on queries made by $\mathcal{A}^{\mathcal{O}_\mathcal{C}}$ on which $\mathcal{A}$ outputs $\{(X_1, Y_1, Z_1), \cdots, (X_{q(n)}, Y_{q(n)}, Z_{q(n)})\} \subseteq (G^3)^{q(n)}$ such that $X_i \in \mathcal{C}$ and $Z_i = CDH(X_i, Y_i)$ for all i, $1 \leq i \leq q(n)$, then on the same inputs and random coins and oracle answers $\mathcal{K}(g, \mathcal{C}, z, \rho, \varpi)$ outputs $\{(X_1, Y_1, Z_1, y_1), \cdots, (X_{q(n)}, Y_{q(n)}, Z_{q(n)}, y_{q(n)})\}$ where $Y_i = g^{y_i}$ for all i, $1 \leq i \leq q(n)$.

We note that the CKEA assumption can be viewed as the non-black-box counterpart of the gap Diffie-Hellamn assumption, while the original KEA assumption is that of the CDH assumption. As we shall show in this work, the CKEA assumption is powerful for achieving highly practical cryptographic protocols provably secure against CMIM adversaries in concurrent settings like the Internet. We suggest the CKEA-based approach for achieving concurrent non-malleability can be a useful paradigm in DH-based cryptographic practice, with a reasonable trade-off between practical efficiency and formal provable security.

4.3 Simulation with restricted RO

When employing the simulation paradigm for proving the security of cryptographic protocols in the RO model, the RO is usually programmed by the simulator (i.e., the simulator provides random answers to RO queries, provided that multiple identical RO-queries are answered with the same answer). But a subtlety here is: simulation with (programmable) RO may lose deniability in general [33, 35, 41].

To overcome the deniability loss of simulation with programmable RO, the works of [33, 35] proposed the unprogrammable RO model where all parties have access to an unprogrammable (fixed) RO. A further investigation, made in [41] (particularly for *interactive* protocols), showed that, in most cases (particularly for the subtleties observed in [33, 35, 41]), the problem lies in the ability of the simulator in defining (i.e., programming) the RO on queries (first) made by the simulator itself in order to simulate honest parties of private inputs. Specifically, the simulator runs the underlying adversary as a subroutine and mimics honest parties in its simulation. Typically, honest parties (e.g., honest ZK provers) possess some private inputs and get access to un unprogrammable RO in reality; The simulator (in its simulation) has to take the advantage of its ability in programming the RO (to be more precise, programming the RO on queries first made by the simulated honest parties) in order to successfully simulate messages generated by honest parties. But, the simulated honest-party messages may not necessarily be generated with the unprogrammable RO actually accessed by the honest parties in reality. This is precisely the reason for the problems, particularly the loss of deniability, observed in [33, 35, 41] for simulation with programmable RO.

The work of [41] proposed the *restricted* RO model, where all parties (particularly, all honest parties and the simulator) *except the adversary* get access to an unprogrammable RO but the adversary (who is polynomial-time and possesses no private inputs) is still allowed to access a programmable RO. We can simply view that the restricted RO model is identical to the original RO model, except for that the simulator is confined to programming the RO only on queries first made by the adversary (run

by the simulator as its subroutine). Clearly, the restricted RO model is a hybrid of the original programmable RO model [4] and the unprogrammable RO model [33, 35]. The restricted RO model allows efficient (interactive) protocol implementations, while still reasonably avoiding the loss of some properties (particularly, deniability) caused by simulation with fully programmable RO.

4.4 Security results and overview

For the security of the DIKE protocol (depicted in Figure 1), we prove that it enjoys both the (privacy-preserving) TBRNM security and the SK-security with post-specified peers. Specifically, the DIKE protocol is *privacy-preserving* tag-based robust non-malleable in the restricted RO model under the GDH assumption and the CKEA assumption. In particular, as a warm-up, we show that the DIKE protocol also implies a 3-round adaptive tag-based concurrent non-malleable statistical (straight-line) zero-knowledge argument of knowledge for discrete logarithm (DL), which is presented in Section 5. We then prove that the DIKE protocol, *with exposable DH-exponents and pre-computed DH-components*, is SK-secure in the CK-framework with post-specified peers under the GDH assumption in the random oracle model. We suggest that the (restricted) RO and the CKEA assumption might be unavoidable to achieve *highly practical* DHKE protocols of the TBRNM security (particularly with the SK-security *simultaneously*). But, the proof details of TBRNM and SK-security are somewhat tedious and conceptually less interesting. For space limitation and to avoid potential sidetracking, the reader is referred to the full paper for complete proof details. Below, we mainly provide high-level overviews of the TBRNM analysis (particularly, the tricks of using CKEA assumption and restricted RO in the TBRNM analysis) and the SK-security analysis.

TBRNM analysis overview. For the TBRNM analysis, the polynomial-time simulator S generates DH-components and DH-exponents by itself by emulating honest player instances. But, different from honest player instances, S uses the DH-exponents (generated by S itself) merely for DDH-tests in its simulation. To this end, S maintains a DDH-test list, denoted $\mathcal{L}_{DDH}$, and stores all DDH-test records into $\mathcal{L}_{DDH}$. The key observation is: what can be done by the simulator S can also be done by another efficient oracle machine $S^{\mathcal{O}_\mathcal{C}}$ on the same common input and the random coins of S *except the coins used to generate the DH-components*, where $\mathcal{O}_\mathcal{C}$ is a DDH-oracle and $\mathcal{C} = \{X_1^l, \cdots, X_{s(n)}^l, Y_1^r, \cdots, Y_{s(n)}^r\}$ is the set of all the DH-components generated by S. Specifically, $S^{\mathcal{O}_\mathcal{C}}$ works just as S does, but with the following modifications: (1) $S^{\mathcal{O}_\mathcal{C}}$ just sets the DH-component for the i-th left-session (resp., the j-th right-session) to be the value X_i^l (resp., Y_j^r), $1 \leq i, j \leq s(n)$, given in the set of $\mathcal{C}$, rather than generating them by itself as S does. (2) Whenever $S^{\mathcal{O}_\mathcal{C}}$ needs to perform a DDH-test w.r.t. a DH-component in $\mathcal{C}$, it queries the DDH-test to its oracle $\mathcal{O}_\mathcal{C}$ and stores the record of the DDH-test into $\mathcal{L}_{DDH}$. Whenever $S^{\mathcal{O}_\mathcal{C}}/S$ needs to extract the DH-exponent and/or secret-key corresponding to the DH-component and/or public-key sent and alleged by the CMIM adversary $\mathscr{A}$, $S^{\mathcal{O}_\mathcal{C}}/S$ runs the CKEA-extractor $\mathcal{K}$ on the same common input, the random coins of $S^{\mathcal{O}_\mathcal{C}}$ *that just correspond to the coins of* S *except the coins used to generates the DH-components* X_i^l*'s and* Y_j^r*'s*, and $\mathcal{L}_{DDH}$ that corresponds to the vector of records of DDH-tests performed by $\mathcal{O}_\mathcal{C}$. By the CKEA assumption,

$\mathcal{K}$ will successfully extract the corresponding DH-exponents and/or secret-keys with overwhelming probability.

For the use of restricted RO, whenever S needs to send one of the values $NMZK(\hat{B}, y)$, $NMZK(a, x)$ and $NMZK(b, y)$, it first checks whether the value has been defined by checking all RO queries made by $\mathscr{A}$ and performing corresponding DDH-tests. If the value to be sent has already been defined (by $\mathscr{A}$'s RO query), the value is set to be the already defined one, otherwise, S sends a random value. If S sends a random value, from this point on whenever $\mathscr{A}$ makes an RO query, S checks whether the previously sent random value is the answer to the RO query (again by performing DDH-tests). Note that in the later case (i.e., the value to be sent has not been defined), S does not try to use its knowledge of DH-exponents (generated by itself) to honestly generate such values, to ensure that those DH-exponents are used merely for DDH-tests in order to comply with the CKEA assumption. If $\mathscr{A}$ never makes an RO query with the previously sent random value as the RO answer, the RO on this point remains undefined. In particular, S never defines it on its own, which ensures S works in the restricted RO model. By the above tricks, the simulator S works in *strict* polynomial-time and its simulation is *straight-line* (without rewinding $\mathscr{A}$).

SK-security analysis overview. The core of the SK-security analysis is to prove that any PPT CMIM attacker can successfully finish an *unexposed* session in the name of some uncorrupted player only if that uncorrupted player (impersonated by the CMIM attacker) does indeed send the authenticated value, say, $NMZK(a, x)$ or $NMZK(b, y)$, in the corresponding matching session. In more details, we prove that: for the DIKE protocol $\langle \hat{A}, \hat{B} \rangle$ (depicted in Figure 1) *with exposable DH-exponents and pre-computed DH-components*, where the players $\hat{A}$ and $\hat{B}$ may be identical, the probability of the following events is negligible under the GDH assumption in the random oracle model:

Event-1. The CMIM adversary $\mathscr{A}$ successfully finishes the j-th right-session for some j, $1 \leq j \leq s(n)$, where $\mathscr{A}$ sends $\widetilde{NMZK}(a, \tilde{x}_j^r)$ in the third-round in the name of $\hat{A}$ (actually, any uncorrupted player) with respect to the DH-component Y_j^r sent by the uncorrupted player $\hat{B}$ in the second-round, while the uncorrupted player $\hat{A}$ did not send $\widetilde{NMZK}(a, \tilde{x}_j^r)$ in any left-session and $\mathscr{A}$ does not know the discrete-logarithm of Y_j^r (i.e., $\mathscr{A}$ did not make the state-reveal query against the j-th right-session at the uncorrupted player $\hat{B}$ in accordance with the CK-framework).

Event-2. The CMIM adversary $\mathscr{A}$ successfully finishes the i-th left-session for some i, $1 \leq i \leq s(n)$, where $\mathscr{A}$ sends $\widetilde{NMZK}(b, \tilde{y}_i^l)$ in the fourth-round in the name of $\hat{B}$ (actually, any uncorrupted player) with respect to the DH-component X_i^l sent by the uncorrupted player $\hat{A}$ in the first-round, while the uncorrupted player $\hat{B}$ did not send $\widetilde{NMZK}(b, \tilde{y}_i^l)$ in any right-session and $\mathscr{A}$ does not know the discrete-logarithm of X_i^l (i.e., $\mathscr{A}$ did not make the state-reveal query against the i-th left-session at the uncorrupted player $\hat{A}$ in accordance with the CK-framework).

Now, suppose the DIKE protocol is not SK-secure, which roughly means that $\mathscr{A}$ can distinguish the session-key $H_K(X, Y, g^{xy})$ of an *unexposed* test-session, say a left-session $(\hat{A}, sid)$ at the side of the uncorrupted player $\hat{A}$, from a random value. Let $X = g^x$ (resp, $Y = g^y$) be the DH-component sent by $\hat{A}$ (resp., $\hat{B}$), and $NMZK(b, y) = H(sid, \hat{B}, Y, X, X^y, X^b)$ be the authentication value sent by $\hat{B}$ (maybe impersonated

by $\mathscr{A}$) in the fourth-round of this test-session. By the above discussions, we have that with overwhelming probability the uncorrupted player $\hat{B}$ does indeed send $NMZK(b, y)$ in one of right-sessions. This implies that in the RO model with overwhelming probability, the (left) test-session has matching (right) session $(\hat{B}, sid)$ in which $\hat{B}$ sends Y in the second-round (after receiving X in the first-round but not necessarily in the peer name of $\hat{A}$) and $NMZK(b, y)$ in the fourth-round.

As the session-key is computed as $H_K(X, Y, g^{xy})$ and H_K is a random oracle, there are only two strategies for the adversary $\mathscr{A}$ to distinguish $H_K(X, Y, g^{xy})$ from a random value:

- *Key-replication attack:* $\mathscr{A}$ succeeds in forcing the establishment of a session (other than the test-session or its matching session) that has the same session-key output as the test-session. In this case, $\mathscr{A}$ can learn the test-session key by simply querying that session to get the same key (without having to expose the test-session or its matching session).
- *Forging attack:* At some point in its run, $\mathscr{A}$ queries the RO H_K with (X, Y, g^{xy}).

The possibility of the key-replication attack is trivially ruled out in the RO model, by observing that X is only sent by $\hat{A}$ in the test-session and Y is only sent by $\hat{B}$ in the matching session.

The success of the forging attack says $\mathscr{A}$ can successfully output $(X, Y, CDH(X, Y))$. Recall that, with overwhelming probability, X and Y are only sent *by uncorrupted players* in the test-session and its matching session. As both the test-session and its matching session are assumed to be unexposed in accordance with the CK-framework (and thus $\mathscr{A}$ does not know the DH-exponent x or y), then we can exploit the assumed ability of $\mathscr{A}$ in performing the successful forging attack to break the CDH assumption (with the assistance of the DDH-oracle $\mathcal{O}_X$ or $\mathcal{O}_Y$), which in turn violates the GDH assumption.

4.5 Discussions on the resistance against some concrete attacks

The both TBRNM security and SK-security of our DIKE protocol imply the resistance to most concrete yet essential security attacks against DHKE protocols (some of which are beyond the SK-security), particularly, unknown key share (UKS), key compromise impersonation (KCI), cutting-last-message attack, perfect forward security (PFS), reflection attacks, etc. In this section, we make informal discussions on the resistance to some of these concrete attacks, with more details deferred to the full paper.

Resistance against unknown key share attack. Informally speaking, by a successful UKS attack an adversary can successfully make two uncorrupted parties compute the same session-key in two sessions but have different views of who the peer to the exchange was, even if the adversary actually does not know the corresponding session-key.

For a successful UKS attack against our DIKE protocol, between two sessions of different pairs of players, we have the following observations: As the session-key is derived from $H(X, Y, g^{xy})$ and the two sessions are of the same session-key, with overwhelming probability in the RO model these two sessions must be of the same DH-components, say (X, Y), and furthermore, in the same (initiator and responder) order. Note that, with overwhelming probability, there are at most two sessions (involving uncorrupted players) of the same DH-components exchanged in the same order, as uncorrupted players generate DH-components randomly and independently. In other words,

besides the two sessions suffering from the UKS attack, there exist no other sessions of the same (ordered) DH-components (X, Y). This implies that each one of the two sessions (suffering from the UKS attack) is of a *distinct* tag, i.e., different from the tags of all other sessions. By the tag-based robust non-malleability, the adversary must know both of the corresponding DH-exponents x and y (and also the secret-keys corresponding to the public-keys alleged by the adversary in the two sessions). This particularly implies that the adversary does know the session-key $H(X, Y, g^{xy})$, which violates the assumed success of the UKS attack.

Resistance against cutting-last-message attack. Suppose the player $\hat{B}$ sends the last message in the run of a DHKE protocol $\langle \hat{A}, \hat{B} \rangle$, the cutting-last-message attack, suffered by IKEv2, works as follows [29]: A man-in-the-middle $\mathscr{A}$ interacts with the uncorrupted $\hat{B}$ in the name of $\hat{A}$ in a session (referred to as the test-session), while concurrently interacting with the uncorrupted $\hat{A}$ in the name of $\hat{M} \neq \hat{B}$ in another session (referred to as the matching session). $\hat{M}$ just relays messages between $\hat{A}$ and $\hat{B}$ in these two sessions, but aborts the matching session after receiving the last message from $\hat{B}$ in the test-session. Such a simple attack results in authentication failure as follow: $\hat{B}$ is perfectly fooled to believe that it has shared a session key with $\hat{A}$ in the test-session, while $\hat{A}$ thinks it only ever took part in an aborted session with $\hat{M}$ in the matching session. (As suggested in [7], this cutting-last-message attack can be prevented by adding an additional fifth-round of "acknowledgement" from $\hat{A}$ to $\hat{B}$, but increasing the system complexity.)

Such cutting-last-message attack is simply ruled out for our DIKE protocol by the tag-based robust non-malleability of DIKE. Specifically, for the above cutting-last-message attack, with overwhelming probability the tag of the completed test-session (i.e., the one in which $\hat{B}$ believes it has shared a session-key with $\hat{A}$) must be *distinct*; In particular, it is different from the tag of the aborted matching session in which $\mathscr{A}$ interacts with $\hat{A}$ in the name of $\hat{M} \neq \hat{B}$. By the tag-based robust non-malleability, it implies that $\mathscr{A}$ has to know $\hat{A}$'s secret-key a and the DH-exponent generated by $\hat{A}$ in the aborted matching session, in order to successfully complete the test-session with $\hat{B}$ in the name of $\hat{A}$. In particular, after receiving the second-round message $(\hat{B}, Y = g^y, NMZ(\hat{B}, y))$ from $\hat{B}$ in the test-session, the CMIM adversary $\mathscr{A}$ cannot compute and send to $\hat{A}$ the message of $(\hat{M}, Y = g^y, NMZK(\hat{M}, y))$ in the name of $\hat{M} \neq \hat{B}$ in the matching session.

Implication of perfect forward secrecy. Informally, a key-exchange protocol is of the PFS property, if the leakage of the static secret-key of an uncorrupted player does not compromise the security of the session-keys established by the player for unexposed yet expired sessions, which have been erased from memory before the leakage occurred [25]. In other words, once an unexposed session is expired and the session-key is erased from its holder's memory, then the session-key cannot be learned by the attacker even if the player is subsequently corrupted. The PFS property of our DIKE protocol is from the observation that: the computation of the session-key $H_K(X, Y, g^{xy})$ does not involve players' secret-keys. Note also that secret-key leakage has already been captured by the TBRNM formulation.

Resistance against reflection attack. In a *reflection* attack, an attacker simply copies the authentic messages from $\hat{A}$ and sends them back to $\hat{A}$ as the messages coming from

the other copy of $\hat{A}$. With respect to the protocol structure of our deniable IKE, to mount a successful reflection attack against the DIKE protocol an adversary has to set $Y = X$ and $\hat{B} = \hat{A}$ so that $(X, Y) = (Y, X)$ and $NMZK(b, y) = NMZK(a, x)$. But, this play is frustrated with our DIKE, by briefly noting that the adversary cannot provide the proof-of-knowledge of $y = x$, i.e., $NMZK(\hat{B}, y) = NMZK(\hat{A}, x)$, in the second-round.

5 Protocol Variants and Implications

Deniable IKE: the aggressive model. In accordance with the aggressive model of IKE [21, 22], we present the 3-round variant of our DIKE protocol in Figure 2.

Most security properties of the deniable IKE of the main model are essentially inherited by this 3-round protocol variant in the aggressive model, except for the full deniability for the responder player $\hat{B}$. Specifically, the player $\hat{B}$ only enjoys *completed-session deniability*, in the sense that if a malicious player $\hat{A}$, denoted as $\hat{A}^*$, completes the session then $\hat{B}$'s deniability will be guaranteed. But if the (possibly malicious) $\hat{A}^*$ just aborts the session after receiving the second-round message, the deniability for $\hat{B}$ is not ensured. Note that the initiator player $\hat{A}$ still has full deniability. We remark that such kind of completed-session deniability for the responder is still very useful and reasonable. For instance, consider the scenario where $\hat{A}$ is a client and $\hat{B}$ is a (bank or shop) server: in such a scenario it is the client $\hat{A}$ who cares more about its privacy and full deniability does guarantee for it, while the server cares less about deniability and the completed-session deniability may still be deemed to be good enough for it.

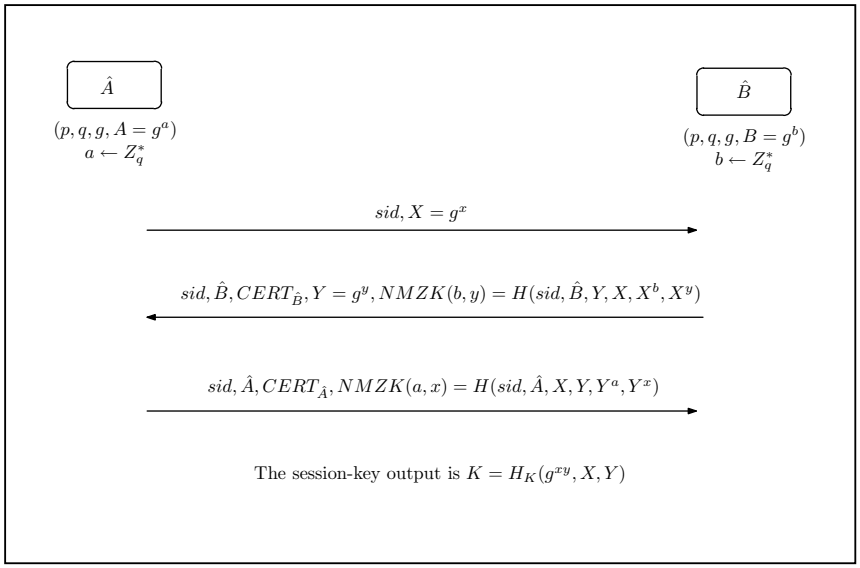

Fig. 2. Deniable Internet Key-Exchange (the aggressive model)

3-round adaptive tag-based concurrent non-malleable (statistical straight-line) zero-knowledge argument of knowledge (CNMZKAOK) for DL. Let $A = g^a \in G$ be the common input (where the group G is specified by the parameters (p, q, g)), $a \in Z_q$ be

the private input of the prover $\hat{A}$, Tag be the session-tag, and H be a hash function that is assumed to be a (restricted) random oracle. The protocol of adaptive tag-based CNMZKAOK for DL is depicted in Figure 3 (page 17), where the DH-component X (resp., Y) is taken randomly and independently from $G/1_G$ by $\hat{A}$ (resp., $\hat{B}$) and each player checks the $G/1_G$ membership of its peer's DH-component.

Note that in the protocol specification, for presentation simplicity, the session-tag Tag is predetermined and known to both the prover $\hat{A}$ and the verifier $\hat{B}$ prior to the protocol run. In an actual adversarial setting, the session-tag may be set by the CMIM adversary adaptively during the protocol run based on its view in all the concurrent (left and right) sessions. The security analysis given in the full paper, which is based upon the CKEA assumption in the restricted RO model, is w.r.t. this general adversarial setting of adaptive tag selection.

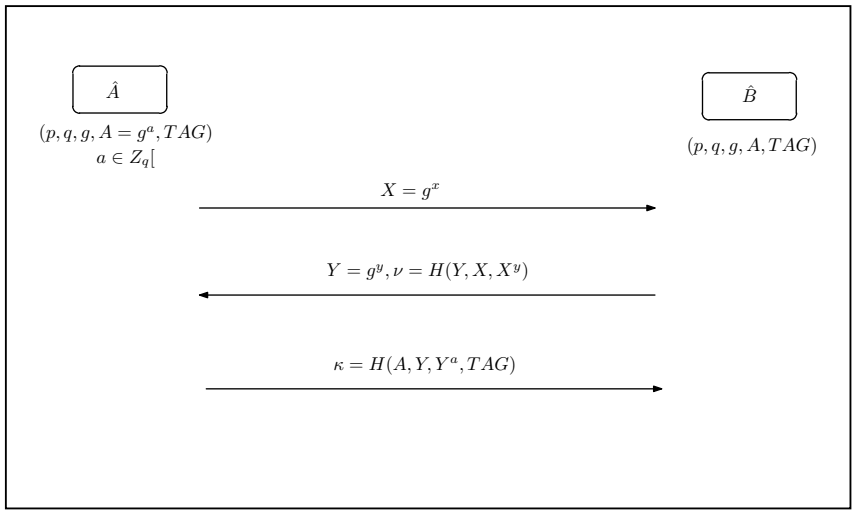

Fig. 3. Adaptive tag-based straight-line CNMZKAOK for DL in the restricted RO model

The 3-round adaptive tag-based CNMZKAOK protocol for DL, depicted in Figure 3, further implies a 3-round concurrent and *forward* deniable authentication protocol [15], based on the CKEA assumption and the DL assumption in the restricted RO model, by viewing messages to be authenticated as the session-tags.

Acknowledgment: We are indebted to Frances F. Yao and Bin Zhu for many contributions to the earlier versions of this work, though they finally declined the coauthorship.

References

1. M. Bellare and A. Palacio. The Knowledge-of-Exponent Assumptions and 3-Round Zero-Knowledge Protocols. In *M. Franklin (Ed.): Advances in Cryptology-Proceedings of CRYPTO 2004, LNCS 3152*, pages 273-289, Springer-Verlag, 2004.

2. M. Bellare and A. Palacio. Towards Plaintext-Aware Public-Key Encryption without Random Oracles. In *P. J. Lee (Ed.): Advances in Cryptology-Proceedings of Asiacrypt 2004, LNCS 3329*, pages 48-62, Springer-Verlag, 2004.
3. M. Bellare and P. Rogaway. Entity Authentication and Key Distribution. In *D. Stinson (Ed.): Advances in Cryptology-Proceedings of CRYPTO 1993, LNCS 773*, pages 273-289, Springer-Verlag, 1993.
4. M. Bellare and P. Rogaway. Random Oracles are Practical: A Paradigm for Designing Efficient Protocols. In*ACM Conference on Computer and Communications Security*, pages 62-73, 1993.
5. R. Canetti, U. Feige, O. Goldreich and M. Naor. Adaptively Secure Multi-Party Computation. In *ACM Symposium on Theory of Computing*, pages 639-648, 1996.
6. R. Canetti and H. Krawczyk. Analysis of Key-Exchange Protocols and Their Use for Building Secure Channels. In *Advances in Cryptology-Proceedings of EUROCRYPT 2001, LNCS 2045*, Springer-Verlag, 2001.
7. R. Canetti and H. Krawczyk. Security Analysis of IKE's Signature-Based Key-Exchange Protocol. In *M. Yung (Ed.): Advances in Cryptology-Proceedings of CRYPTO 2002, LNCS 2442*, pages 143-161, Springer-Verlag, 2002.
8. R. Canetti, J. Kilian, E. Petrank and A. Rosen. Black-Box Concurrent Zero-Knowledge Requires $\tilde{\Omega}(log\ n)$ Rounds. In *ACM STOC 2001*, pages 570-579.
9. I. Damgård. Towards Practical Public-Key Systems Secure Against Chosen Ciphertext Attacks. In *J. Feigenbaum (Ed.): Advances in Cryptology-Proceedings of CRYPTO 1991, LNCS 576*, pages 445-456. Springer-Verlag, 1991.
10. A. Dent. Cramer-Shoup Encryption Scheme is Plantext Aware in the Standard Model. In *Advances in Cryptology-Proceedings of EUROCRYPT 2006, LNCS 4004*, pages 289-307. Springer-Verlag, 2006.
11. M. Di Raimondo and R. Gennaro. New Approaches for Deniable Authentication. In proc. of 12nd ACM Conference on Computer and Communications Security (ACM CCS'05), ACM Press, pages 112-121, 2005.
12. M. Di Raimondo, R. Gennaro and H. Krawczyk. Deniable Authentication and Key Exchange. ACM CCS'06, pages 466-475. Full version appears in Cryptology ePrint Archive Report No. 2006/280.
13. W. Diffie and M. Hellman. New Directions in Cryptography. *IEEE Transactions on Information Theory*, 22(6): 644-654, 1976.
14. Y. Dodis, J. Katz, A. Smith and S. Walfish. Composability and On-line Deniability of Authentication. Theory of Cryptography Conference (TCC), pages 146-162, 2009.
15. D. Dolev, C. Dwork and M. Naor. Non-Malleable Cryptography. *SIAM Journal on Computing*, 30(2): 391-437, 2000. Preliminary version in *ACM Symposium on Theory of Computing*, pages 542-552, 1991.
16. C. Dwork, M. Naor and A. Sahai. Concurrent Zero-Knowledge. In *ACM Symposium on Theory of Computing*, pages 409-418, 1998.
17. FIPS Pub 186-2, *Digital Signature Standard (DSS)*, Federal Information Processing Standards Publication 186-2, US Department of Commerce/National Institute of Standard and Technology, Githersburg, Maryland, USA, January 27, 2000.
18. O. Goldreich, S. Micali and A. Wigderson. How to Play Any Mental Game. In *ACM Symposium on Theory of Computing*, pages 218-229, 1987.
19. S. Goldwasser, S. Micali and C. Rackoff. The Knowledge Complexity of Interactive Proof-Systems In *ACM Symposium on Theory of Computing*, pages 291-304, 1985.
20. S. Hada and T. Tanaka. On the Existence of 3-Round Zero-Knowledge Protocols. In *H. Krawczyk (Ed.): Advances in Cryptology-Proceedings of CRYPTO 1998, LNCS 1462* , pages 408-423, Springer-Verlag, 1998.

21. *D. Harkins and D. Carreal (Ed.): The Internet Key-Exchange (IKE)*, RFC 2409, Nov., 1998.
22. C. Kaufman. Internet Key Exchange (IKEv2) Protocol. The Internet Engineering Task Force: INTERNET-DRAFT, October 2002.
23. S. Kent and R. Atkinson. Security Architecture for the Internet Protocol. Request for Comments 2401, 1998.
24. H. Krawczyk. SIGMA: the "SIGn-and-MAc" Approach to Authenticated Diffie-Hellman and Its Use in the IKE-Protocols. Invited Talk at *D. Boneh (Ed.): Advances in Cryptology-Proceedings of CRYPTO 2003, LNCS 2729*, pages 400-425, Springer-Verlag, 2003.
25. H. Krawczyk. HMQV: A High-Performance Secure Diffie-Hellman Protocol. In *V. Shoup (Ed.): Advances in Cryptology-Proceedings of CRYPTO 2005, LNCS 3621*, pages 546-566. Springer-Verlag, 2005.
26. L. Law, A. Menezes, M. Qu, J. Solinas and S. Vanstone. An Efficient Protocol for Authenticated Key Agreement. *Designs, Codes and Cryptography*, 28: 119-134, 2003.
27. Y. Lindell. General Composition and Universal Composability in Secure Multi-Party Computation. In *IEEE Symposium on Foundations of Computer Science*, pages 394-403, 2003.
28. Y. Lindell. Lower Bounds and Impossibility Results for Concurrenet Self Composition. *Journal of Cryptology*, 21(2): 200-249, 2008.
29. W. Mao. Modern Cryptography: Theory and Practice. Prentice Hall PTR, 2004.
30. U. Maurer and S. Wolf. Diffie-Hellman Oracles. In *Advances in Cryptology-Proceedings of CRYPTO 1996, LNCS 1109*, pages 268-282, Springer-Verlag, 1996.
31. A. Menezes. Another Look at HMQV. Cryptology ePrint Archive, Report No. 2005/205.
32. M. Naor and O. Reingold. Number-Theoretic Constructions of Efficient Pseudo-Random Functions. *Journal of the ACM*, 1(2): 231-262 (2004).
33. J. B. Nielsen. Separating Random Oracle Proofs from Complexity Theoretic Proofs: The Non-Committing Encryption Case. In *M. Yung (Ed.): Advances in Cryptology-Proceedings of CRYPTO 2002, LNCS 2442*, pages 111-126, Springer-Verlag, 2002.
34. T. Okamoto and D. Pointcheval. The Gap-Problems: A New Class of Problems for the Security of Cryptographic Schemes. In PKC'01, LNCS 1992, pages 104-118, 2001.
35. R. Pass. On Deniabililty in the Common Reference String and Random Oracle Models. In*D. Boneh (Ed.): Advances in Cryptology-Proceedings of CRYPTO 2003, LNCS 2729*, pages 316-337, Springer-Verlag 2003.
36. R. Pass and A. Rosen. New and Improved Constructions of Non-Malleable Cryptographic Protocols. In *ACM Symposium on Theory of Computing*, pages 533-542, 2005.
37. R. Pass and A. Rosen. Concurrent Non-Malleable Commitments. In *IEEE Symposium on Foundations of Computer Science*, pages 563-572, 2005.
38. D. R. Stinson and J. Wu. An Efficient and Secure Two-Flow Zero-Knowledge Identification Protocol. Cryptology ePring Archive, Report 2006/337.
39. D. R. Stinson and J. Wu. A Zero-Knowledge Identification and Key Agreement Protocol. Cryptology ePring Archive, Report 2007/116.
40. A. C. Yao, M. Yung and Y. Zhao. Adaptive Concurrent Non-Malleability with Bare Public-Keys. Cryptology ePrint Archive, Report 2010/107.
41. M. Yung and Y. Zhao. Interactive Zero-Knowledge with Restricted Random Oracles. Theory of Cryptography Conference (TCC), pages 21-40, 2006.