Hindawi
Security and Communication Networks
Volume 2017, Article ID 7810352, 13 pages
https://doi.org/10.1155/2017/7810352

WILEY | Hindawi

Research Article

Deniable Key Establishment Resistance against eKCI Attacks

Łukasz Krzywiecki and Tomasz Wlisłocki

Wrocław University of Science and Technology, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland

Correspondence should be addressed to Łukasz Krzywiecki; lukasz.krzywiecki@pwr.edu.pl

Received 29 April 2017; Revised 17 July 2017; Accepted 3 August 2017; Published 24 September 2017

Academic Editor: Vincenzo Conti

In *extended Key Compromise Impersonation* (eKCI) *attack* against *authenticated key establishment* (AKE) protocols the adversary impersonates one party, having the long term key and the ephemeral key of the other peer party. Such an attack can be mounted against variety of AKE protocols, including 3-pass HMQV. An intuitive countermeasure, based on BLS (Boneh–Lynn–Shacham) signatures, for strengthening HMQV was proposed in literature. The original HMQV protocol fulfills the deniability property: a party can deny its participation in the protocol execution, as the peer party can create a fake protocol transcript indistinguishable from the real one. Unfortunately, the modified BLS based version of HMQV is not deniable. In this paper we propose a method for converting HMQV (and similar AKE protocols) into a protocol resistant to eKCI attacks but without losing the original deniability property. For that purpose, instead of the undeniable BLS, we use a modification of Schnorr authentication protocol, which is deniable and immune to ephemeral key leakages.

1. Introduction

An authenticated key establishment (AKE) protocol enables two parties: the initiator (starting the protocol, usually called Alice) and the responder (usually called Bob) to mutually identify themselves and establish a secret shared session key, subsequently used to protect communication channel. The *deniability* property for AKE protocols [1, 2] guarantees that parties still can mutually verify their identities, but the transcript of the protocol cannot be regarded as a proof that the parties have executed the protocol together. We distinguish the *initiator deniability* and the *responder deniability* as the deniability feature can be achieved for each party independently. Deniability may be desirable in various privacy protecting scenarios where the proof of interaction should not be transferable; for example, clients of some Internet services might wish to have the right and real possibility of denying using the service.

Many general AKE schemes have been proposed so far; see, for example, MQV [3], HMQV [4], SIGMA [5], KEA+ [6], NAXOS [7], CMQV [8], SMQV [9], E-NAXOS [10], Huang [11], or Kim et al. [12] with numerous additional modifications. Their security has been analyzed in many models, for example, CK [13], eCK [7], and seCK [9], under various attack scenarios. In Key Compromise Impersonation (KCI) attack scenario [14–16], an adversary, which obtained long term secrets of one party, say Alice, can execute AKE protocol with her, and impersonate her another party, say Bob, without using the long term secret of Bob. This attack is especially devastating when the correct identification is of the paramount importance. Imagine the attacker learning the long term key of a bank. Now, the attacker not only can play a role of the bank to any identity (which is obvious), but also can be authenticated as any identity in front of that bank, for example, can be authenticated to the account of some very rich person and subsequently order the money transfer from that logged-in account to his own.

In [17] Tang and Chen proposed a new impersonation attack type on AKE protocols called extended KCI (eKCI). In this attack, the adversary has access not only to Alice's long term secret, but also to her ephemeral secret, for example, the ephemeral Diffie-Hellman key. With the knowledge of both these keys it can impersonate any party to Alice. This new kind of attack can be mounted against protocols already proven to be secure for regular KCI attacks, for example, NAXOS secure in the extended Canetti-Krawczyk model. In [17] authors exemplified the eKCI attack against HMQV protocol [4]. Subsequently they proposed an intuitive and

elegant countermeasure based on BLS signatures [18]: in the rest of the paper we refer to this solution by BLS-HMQV.

From the design point of view BLS-HMQV is a composition of original HMQV with another layer of authentication, done by the BLS signature scheme. In BLS-HMQV, a party running the protocol sends to its peer an additional signature over some challenge depending on previous messages. The signature forms a proof of identity, since it can be produced only with the secret key corresponding to the certified public key of the signer. Unfortunately, there is one aspect of this solution which in some scenarios can be regarded as a serious drawback: signed messages in the protocol transcript may be used as undeniable proof for a third party where the communication with the signers took place. In this context the modification to HMQV proposed in [17] makes it resistant to eKCI, but at the same time the protocol loses its *deniability* property.

Therefore to achieve the deniability property altogether with the eKCI resistance, we follow two-layer architecture of BLS-HMQV. However in our modified protocol (called mHMQV) we exchange the undeniable layer of BLS with the deniable layer of Schnorr-like protocol from [19]. Therefore our proposition mHMQV is deniable like original HMQV and is eKCI resistant like BLS-HMQV, with its two-layer composition.

As a final remark we recall that secrecy and fairness of values generated by both parties rely on the internal implementation of pseudorandom number generator algorithm, which itself may utilize hardware based randomness or external environmental sources. One of the most comprehensive recommendations for such algorithms can be found in [20]. However note that even algorithms approved for scientific simulations [21], with super long periods, like [22], must be specially tuned for cryptographic purposes [23]. A practical construction for using external source of randomness in AKE protocol, resembling the common reference string model, is given in [24]. Secrecy of values gained in this way can be compromised if an adversary captures the measurements of the external source as well. An example countermeasure for that problem, which uses distributed leader election for selecting a random source of data, was proposed in [25]. As for the internal hardware sources of randomness, the promising approach of using physically unclonable functions is also considered, for example, [26, 27]. Such hardware functions rely on micro differences of the used material and characteristics of processes in production phase, which—as unpredictable and unrepeatable even for the device manufacturer—guarantee the uniqueness of the final results.

1.1. Contribution and Organization of the Paper. The contributions of the paper are the following.

(i) Undeniability of BLS-HMQV. We show that BLS-HMQV protocol from [17], which is BLS based modification of HMQV, although resistant to eKCI is no longer deniable.

(ii) Proposition of mHMQV-eKCI as Resistant and Deniable. This is the main contribution. We propose an extension to HMQV (applicable to similar 2-party protocols) which protects against the eKCI attack and which does not destroy the protocol deniability property: for the initiator and subsequently for the responder. We use for that purpose the modified Schnorr identification scheme [19], which is secure even if the ephemeral secrets of parties are compromised. To the best of our knowledge it is the first proposition of this kind for AKE protocols so far.

(iii) Prototype Implementation. To compare the complexity overhead for deniability and eKCI resistance, we implemented prototypes of HMQV, the BLS based scheme (BLS-HMQV), and our deniable proposition mHMQV.

1.2. Previous Work. In Table 1 we give the comparison showing the eKCI resistance and deniability feature of the majority of AKE protocols, alongside level of complexity (based on required computational effort for used operations) and number of rounds. Note that eKCI resistance in [11, 17, 31] is provided by undeniable signature scheme that is used to identify the parties to each other. In the case of [11, 17] BLS signatures are used: we call these protocols "without NAXOS" and "BLS-HMQV," respectively. We observe and stress here that the scheme of [11] does not withstand repetition attack in the setup of eKCI. Namely, after the protocol execution between parties A and B (with the knowledge of the transcript), an adversary can later impersonate A in front of B if long term and ephemeral secrets of B are leaked during the new sessions (and *vice versa*). Therefore we put "!" instead of "✓" in the table. Finally we denote the protocols we proposed in this paper by "mHMQV."

Beside the typical protocols, securing the session key against the combinations of secrets leakages, comparable in terms of the exponentiation operations for Diffie-Hellman based key exchange and listed in Figure 1, there are schemes that address additional requirements and adversarial assumptions. The AKE schemes in identity-based setup using elliptic curves were analyzed, for example, in [32, 33]. Those 2-round schemes are still vulnerable to eKCI attacks (actually the first one does not withstand the regular KCI as well). Authors of [34] proposed a ring signature based scheme, useful for vehicles key exchange and authentication. Note that they use idea close to one already presented in [2]. However, as it was signaled in [2], the ring signature based authentication makes the schemes vulnerable to KCI and eKCI-adversary knowing the peer long term key can impersonate other parties to that peer. In [35] the lattice based HMQV version for postquantum era was proposed. The proposition exchanges the cryptographic building blocks, preserving the construction design, but as the original version, it is still eKCI vulnerable. It is an interesting open question how this particular postquantum HMQV construction can be improved, as the modification based on [19] proposed in our paper is also vulnerable to quantum attacks. There are also approaches for a partial leakage of cryptographic material and bad randomness. The security model assuming partial leakage of bits of secret keys was analyzed in [36]; however the proposed solution is based on the signatures and as so is undeniable. The next solution from [37], addressing similar problem, results in 2-round protocol which still is not eKCI

Table 1: Protocol comparison.

[Paper] protocol	Complexity	Rounds	eKCI resistance	Deniability
[6] KEA+	3	2	—	√
[6] KEA+C	3	3	—	√
[7] NAXOS	4	2	—	√
[28] NAXOS+	5	2	—	√
[10] E-NAXOS	5	2	—	√
[8] CMQV	3	2	—	√
[9] SMQV	3	2	—	√
[12] Prot.1	3	2	—	√
[12] Prot.2	5	2	—	√
[29] AMA	4	4	—	√
[30] MRI	3	4	—	√
[4] HMQV	3	3	—	√
[2] Mod. Σ_0	$2 + RS + RV$	3	—	Initiator
[2] Mod. Σ_1	$2 + RS + RV$	4		Responder
[31] Σ_0	$2 + S + V$	3	√	—
[31] Σ_1	$2 + S + V$	4	√	—
[11] without NAXOS	3	2	!	—
[17] BLS-HMQV	4	3	√	—
mHMQV-1	5	3	√	Initiator
mHMQV-2	6	4	√	√

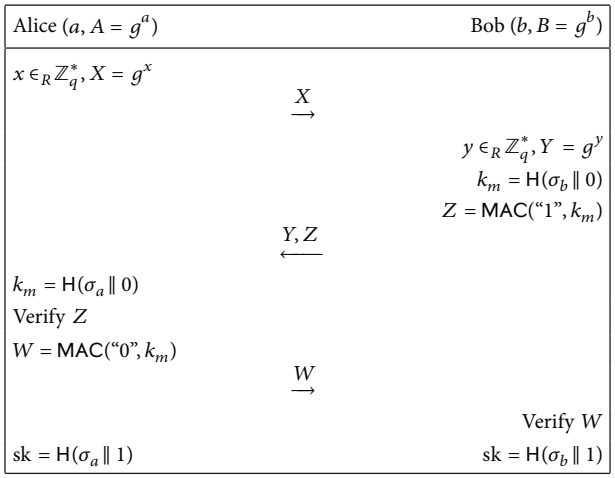

Figure 1: 3-pass HMQV.

resistant. Another 2-round protocol from [38], addressing the "bad randomness" problem for pseudorandom number generators in user devices, is also not eKCI resistant. Another AKE construction, secure without ROM under the hardness of integer factorization problem, code-based problems, or learning with errors problems, was proposed in [39]. Note that this proposition also is not secure to aKCI attacks. In [40] the authors analyzed security model with the adversary registering arbitrary bit strings as keys. They showed generic results for protocols that achieve security even if some keys have been produced maliciously in this way. However this also does not solve the eKCI resistance for typical protocols; for example, the strengthened version of CMQV presented there is still eKCI vulnerable.

To the best of our knowledge the problem of construction of an AKE protocol, both deniable and withstanding eKCI, as stated in Section 1.1, is still open in literature, since the original eKCI introduction in [17]. Please note, additionally, that in the context of immunizing AKE protocols to eKCI attacks, the construction [41], which follows up the paper [19] and is a modification of Okamoto identification scheme, can be also taken into consideration as the authentication layer: as it is deniable and resistant to ephemeral values leakage and setup.

Organization of the Paper. The paper is organized in the following way. In Section 2.2 we recall the HMQV protocol and discuss its deniability property. In Section 3 we recall the eKCI attack on HMQV and the defense method proposed in [17]. We discuss how that approach breaks the deniability of the original HMQV. In Section 4 we propose a solution to the eKCI attack on HMQV based on the modified Schnorr authentication protocol from [19]. We recall the original Schnorr authentication protocol, discuss its deniability property, and show that it is inadequate in setups where the ephemeral keys can be leaked. Then we propose using its modified version to get *initiator deniability*. Subsequently we show how the protocol can be modified further to achieve the *responder deniability*. We prove the security of our claims. In Section 6 we discuss the *proof-of-concept* implementation of our protocols.

2. Preliminaries

2.1. Notation. Presented AKE protocols are based on Diffie-Hellman (DH) key exchange, so we assume that corresponding computations are done within a group $G = \langle g \rangle$ of prime order q, where computational Diffie-Hellman assumption (CDH) holds.

Let I (denotes initiator called Alice) and R (denotes responder called Bob) be two peer parties of the key exchange protocol π. Alice as initiator is the party which starts (sends the first message) the protocol π. Bob is the other party. Let (a, A) and (b, B) denote pairs of long term secret/public keys of Alice and Bob, respectively, randomly chosen according to the key generating algorithm. Usually, apart from the long term keys, each party in protocol π coins additional random secret key, called ephemeral key, used in computation during protocol execution. Let x, y denote ephemeral keys of Alice and Bob, respectively. Thus $\pi(I(a, B, x), R(b, A, y))$ denotes the protocol run between the initiator (Alice) having the secret key a, the ephemeral key x, and the public key B of Bob and the responder (Bob) having the secret key b, the ephemeral key y, and the public key A of Alice.

Typical requirements after the authenticated key establishment protocol

$\pi(I(a, B, x), R(b, A, y))$ is completed (i.e., after both parties finished their computations successfully) are the following:

(i) Both parties mutually identified themselves. We denote that

$\pi(I(a, B, x), R(b, A, y)) \rightarrow (I \,\mathsf{accepts}\, R)$ initiator speaks with responder of identity *Bob*.

$\pi(I(a, B, x), R(b, A, y)) \rightarrow (R \,\mathsf{accepts}\, I)$ Bob knows he speaks with Alice.

(ii) Both parties have computed the same session key.

(iii) The session key is secret; that is, it is known only to the parties of the protocol.

The eKCI attack proposed in [17] affects the first requirement. Intuitively we demand that each party should use its secret key to perform the protocol and be accepted by its peer party. In eKCI attack the adversary can use the peer party secret to impersonate another party.

Definition 1. One says that AKE protocol π is eKCI vulnerable if there exists an efficient adversary algorithm $\mathscr{A}$ such that at least one of the probabilities

$$\begin{aligned} &\Pr\left[\pi\left(I\left(a, B, x\right), \mathscr{A}\left(a, A, B, x, y\right)\right)\right. \\ &\quad \left. \longrightarrow (I \,\mathsf{accepts}\, \mathscr{A} \text{ as Bob})\right] \\ &\Pr\left[\pi\left(\mathscr{A}\left(b, B, A, x, y\right), R\left(b, A, y\right)\right)\right. \\ &\quad \left. \longrightarrow (R \,\mathsf{accepts}\, \mathscr{A} \text{ as Alice})\right] \end{aligned} \tag{1}$$

is nonnegligible.

Remark 2. In the first event $\mathscr{A}(a, B, A, x, y)$ denotes the adversary which possesses Alice's secrets but does not have Bob's long term secret key b. It is identified falsely by Alice as Bob. Similarly in the second event $\mathscr{A}(b, B, A, x, y)$ denotes the adversary which possesses Bob's secrets but does not have Alice's long term secret key a. It is identified falsely by Bob as Alice. Note that this reflects the scenario in which a hacker, knowing secrets of the bank, can impersonate any user in front of that bank, subsequently ordering malicious money transfers on behalf of this user.

Deniability Model. In this point we recall the deniability model from [1], which is applicable to authenticated key establishment protocols.

Definition 3. One says that (KeyGen, I, R) is a concurrently deniable key establishment protocol with respect to the class AUX of auxiliary inputs if, for any adversary $\mathscr{M}$, for any input of public keys $\mathbf{pk} = (\mathsf{pk}_1, \ldots, \mathsf{pk}_\ell)$ and any auxiliary input $\mathrm{aux} \in \mathrm{AUX}$, there exists a simulator $\mathrm{SIM}_{\mathscr{M}}$ that, running on the same inputs as $\mathscr{M}$, produces a simulated view which is indistinguishable from the real view of $\mathscr{M}$. That is, consider the following two probability distributions, where $\mathbf{pk} = (\mathsf{pk}_1, \ldots, \mathsf{pk}_\ell)$ is the set of public keys of the honest parties:

$$\begin{aligned} \mathscr{R}\mathrm{eal}\,(n, \mathrm{aux}) &= \left[\left(\mathsf{sk}_i, \mathsf{pk}_i\right)\right. \\ &\quad \left. \longleftarrow \mathsf{KeyGen}\left(1^n\right); (\mathrm{aux}, \mathbf{pk}, \mathsf{View}_{\mathscr{M}}\left(\mathbf{pk}, \mathrm{aux}\right)\right] \\ \mathscr{S}\mathrm{im}\,(n, \mathrm{aux}) &= \left[\left(\mathsf{sk}_i, \mathsf{pk}_i\right)\right. \\ &\quad \left. \longleftarrow \mathsf{KeyGen}\left(1^n\right); (\mathrm{aux}, \mathbf{pk}, \mathrm{SIM}_{\mathscr{M}}\left(\mathbf{pk}, \mathrm{aux}\right)\right]; \end{aligned} \tag{2}$$

then for all probabilistic polytime machines Dist and all $\mathrm{aux} \in \mathrm{AUX}$

$$\begin{aligned} &\left|\Pr_{x \in \mathscr{R}\mathrm{eal}(n, \mathrm{aux})}\left[\mathsf{Dist}\left(x\right) = 1\right]\right| \\ &\quad - \left|\Pr_{x \in \mathscr{S}\mathrm{im}(n, \mathrm{aux})}\left[\mathsf{Dist}\left(x\right) = 1\right]\right| \leq \mathsf{negl}\left(n\right). \end{aligned} \tag{3}$$

We say that the protocol is *initiator deniable* if there exists the simulator $\mathrm{SIM}_{\mathscr{M}}$, denoted as $\mathrm{SIM}^I_{\mathscr{M}}$, that running on

the same inputs as Bob (and without Alice's secret key) can provide Alice's part of the protocol. That is when Bob can simulate the whole transcript itself. Conversely, we say that the protocol is *responder deniable* if there exists the simulator $\mathrm{SIM}_{\mathscr{M}}$, denoted as $\mathrm{SIM}^R_{\mathscr{M}}$, that running on the same inputs as Alice (and without Bob's secret key) can provide Bob's part of the protocol. That is when Alice can simulate the whole transcript itself.

2.2. Description of the 3-Pass HMQV. Let us recall the 3-pass protocol of the HMQV family from [4], which is proved to be secure against the standard KCI attacks. The two users Alice and Bob agree on a group G of prime order q, a generator g of G, a hash function H, and a message authentication code function MAC. Alice selects her long term private key at random $a \in_R \mathbb{Z}_q^*$ and lets the trusted third party (TTP) certify the public key $A = g^a$. Similarly, Bob selects his long term private key $b \in_R \mathbb{Z}_q^*$ and lets the TTP certify the public key $B = g^b$. The protocol is shown in Figure 1. The values σ_a and σ_b are defined as follows:

$$\begin{aligned} d &= \overline{\mathsf{H}}\left(X \parallel \text{“Bob”}\right), \\ e &= \overline{\mathsf{H}}\left(Y \parallel \text{“Alice”}\right), \\ \sigma_a &= \left(Yg^{be}\right)^{x+da}, \\ \sigma_b &= \left(Xg^{ad}\right)^{y+eb}, \end{aligned} \tag{4}$$

where $\overline{\mathsf{H}}$ outputs the first ℓ bits of the input of the hash function H, and ℓ is a security parameter. Note that $\sigma_a = (Yg^{be})^{x+da} = (g^y g^{be})^{x+da} = g^{(x+da)(y+eb)} = (g^x g^{da})^{y+eb} = (Xg^{da})^{y+eb} = \sigma_b$. Thus the values k_m and the secret session key sk computed independently on both sides are the same.

2.3. Deniability of HMQV

Theorem 4. *HMQV is initiator deniable.*

Proof. We show that the protocol is *initiator deniable* as Bob can produce the transcript of the protocol execution X, Y, Z, W alone, but with the same probability distribution as it would be produced altogether with Alice. Namely, the simulator $\mathrm{SIM}^I_{\mathscr{M}}$ (run with Bob's input) chooses $x \in_R \mathbb{Z}_q^*$ and computes $X = g^x$ and the rest of parameters Y, Z, W, which does not require Alice's secret key a. Observe that for σ_a he does not apply the derivations from the protocol (the private key of Alice would be necessary). Instead, he makes use of the equality $\sigma_a = \sigma_b$. □

It follows that a transcript cannot be regarded as a proof that Alice participates in the protocol execution. Similarly we state the following.

Theorem 5. *HMQV is responder deniable.*

Proof. It is analogical to the proof of Theorem 4. Alice can produce the transcript alone: $\mathrm{SIM}^R_{\mathscr{M}}$ (run with Alice's view and input) chooses $Y \in_R \mathbb{Z}_q^*$ and computes $Y = g^y$ and the rest of parameters Y, Z, W, which does not require Bob's secret key b. □

2.4. eKCI Attack on the 3-Pass HMQV. We recall the original eKCI attack on HMQV from [17]. Suppose that an adversary has access to x and a and mounts an attack against Alice. After obtaining the first message X the adversary computes $\sigma'_b = g^{(x+ad)y} \cdot B^{(x+ad)e} = g^{(x+ad)y} \cdot (g^b)^{(x+ad)e}$. This equals $g^{(x+da)(y+eb)}$. Then it computes the rest of parameters on Bob's side and sends them to Alice, impersonating in this way itself as Bob. Note the fact that the computation of σ_b does not require the knowledge of b. It is straightforward to verify that $\sigma_b = \sigma_a$, and the adversary always succeeds in the attack.

3. Prevention of the Attack: Undeniable Version

Let us recall the method from [17] protecting against the eKCI attack. The idea is that the users (Alice and Bob) should mutually demonstrate the knowledge of their long term private key to each other. The authors propose the use of deterministic BLS signature scheme [18]. We denote the resulting protocol as BLS-HMQV. The construction of that protocol is very intuitive: It can be viewed as two-layer approach:

(i) The first layer is the original HMQV.

(ii) The second layer includes the BLS signatures over the parameters of the HMQV protocol (parties identifiers, messages).

Indeed any adversary algorithm that would break eKCI resistance of BLS-HMQV, that is, would impersonate one party by means of anything but the long term secret key (e.g., the other party parameters) would be immediately used to break unforgeability of BLS signature scheme.

3.1. BLS-HMQV. First let us briefly recall the BLS scheme. Let G, G_T be groups of a prime order q and g be a generator of G. Let $\mathsf{H}_1 : \{0,1\}^* \to G$. We assume that $\hat{e} : G \times G \to G_T$ is a bilinear map, and a signer holds a private/public key pair (α, g^α), where $\alpha \in_R \mathbb{Z}_q^*$. For a message $m \in \{0,1\}^*$, the signature generation and verification procedures are as follows:

(1) The signer computes a signature V, where $V = (\mathsf{H}_1(m))^\alpha \in G$.

(2) The verifier checks whether $\hat{e}(V, g) = \hat{e}(\mathsf{H}_1(m), g^\alpha)$. If so, the signature is accepted.

The BLS-HMQV based solution to the eKCI attack on HMQV is depicted in Figure 2. We follow the notation from [17]. The important part of the protocol extension computed on the responder side is boxed. Similarly respective computations on the initiator side are underlined.

3.2. Loosing Deniability. Although BLS-HMQV is resistant to eKCI attack, we observe that the protocol depicted in Figure 2 is *not initiator deniable*.

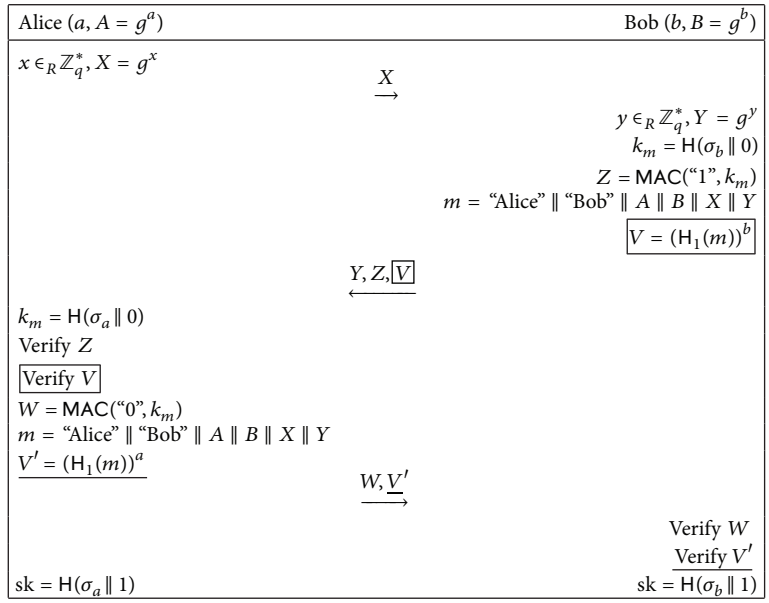

Figure 2: BLS-HMQV: BLS based prevention of the eKCI attack against HMQV.

Theorem 6. *The BLS-HMQV protocol depicted in Figure 2 is not initiator deniable.*

Proof. Indeed, in order to produce a simulated transcript indistinguishable from the original one, a simulator $\mathrm{SIM}_{\mathscr{M}}^{I}$ (run with Bob's input and without the knowledge of Alice's secret key a) would have to create a verifiable signature V'. So it would be used as an efficient forger for the underlying BLS scheme, contradicting BLS security. □

Corollary 7. *The BLS-HMQV protocol in not responder deniable due to the similar reasoning.*

4. Our Proposition: Deniable Prevention to eKCI Attack

In this section we propose the deniable version of the solution to eKCI attack. It is based on exchanging the undeniable BLS layer from BLS-HMQV with the deniable identification (IS) scheme, for example, Schnorr IS. To illustrate the idea of the construction we first show the *initiator deniable* solution based on the Schnorr identification protocol [42]. Next we observe that this particular solution is imperfect in systems where the ephemeral secrets may be leaked: the security of the long term key relies on the security of the ephemeral key; thus once the ephemeral secrets are leaked the long term secrets are also compromised.

4.1. The Basic Schnorr Based Imperfect Solution. Let us recall the Schnorr identification protocol from [42].

Schnorr Identification Protocol. Let G be a group of prime order q and g be a generator of G. Suppose that an authenticator possesses the certified private/public key pair $(a, A = g^a)$, and a verifier already knows the public key $A = g^a$.

(1) The authenticator computes $x \in_R \mathbb{Z}_q^*$, $X = g^x$ and sends X to the verifier.

(2) The verifier choses $c \in_R \mathbb{Z}_q^*$ and sends it to the authenticator.

(3) The authenticator computes $s = x + ac$ and sends s to the verifier.

(4) The verifier accepts the verification iff $g^s = XA^c$.

The *initiator deniable* version of the protocol from Figure 2 augmented with the Schnorr identification protocol is presented in Figure 3. The hash function $\mathsf{H}_2 : \{0,1\}^* \to \mathbb{Z}_q^*$ effectively produces challenge c computed from m_A, which itself contains Y coined at Bob's side.

Deniability of the Basic Schnorr Based Solution. To prove the deniability of the protocol (Figure 3) for Alice it suffices to show the construction of the efficient simulator that produces the protocol transcript without the knowledge of Alice's secret a. Indeed such a simulator exists: Bob simulates the messages of Alice with the distribution indistinguishable from the original one:

(1) Bob chooses randomly $s \in_R \mathbb{Z}_q^*$.

(2) Bob computes $g^x = X \coloneqq (g^s)/A^c$. Thus $s = x + ac$, although Bob does not know the value x.

(3) Having X and s Bob computes the rest of the parameters and protocol messages: Y, Z, V are computed by Bob alone from his secrets; W is computed as $\mathsf{MAC}(\text{“0”}, k_m)$, where values k_m on both sides are equal; hence $\sigma_a = \sigma_b$. Thus he produces the transcript X, Y, Z, V, W, s which has the same distribution as the original transcript that would be produced altogether with Alice.

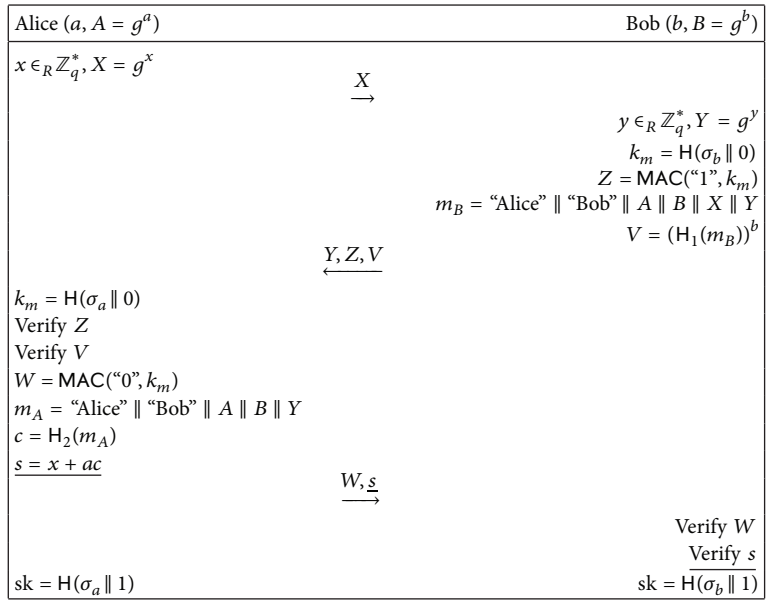

Figure 3: Preventing eKCI-initiator deniable imperfect solution.

Note that message m_A computed on Alice's side does not contain X. Otherwise it would be impossible to compute $X = (g^s)/A^c$ for $c = \mathsf{H}_2(m_A)$. Indeed, this trick was used to provide deniability of PACE|AA protocol from [43].

Imperfection of the Basic Schnorr Based Solution. The solution is imperfect in scenarios where ephemeral keys can be leaked. If the ephemeral secret x is known to the adversary, it can compute Alice's long term secret $a := (s - x)/c$ and impersonate her since then. Therefore in the next section we propose using the secure version from [19].

4.2. Prevention of the Attack: Secure Deniable Solution

Modified Schnorr Identification Protocol from [19]. The idea of that protocol is to perform response computation in the exponent using a new generator $\widehat{g}$. Let s recall the steps:

(1) The authenticator computes $x \in_R \mathbb{Z}_q^*$, $X = g^x$ and sends X to the verifier.

(2) The verifier computes a challenge $c \in_R \mathbb{Z}_q^*$ and sends it to the authenticator.

(3) The authenticator computes $\widehat{g} = \mathsf{H}_1(X \mid c)$, $S = (\widehat{g})^x (\widehat{g})^{ac}$ and sends S to the verifier.

(4) The verifier accepts the verification iff $\widehat{e}(S, g) = \widehat{e}(\mathsf{H}_1(X \mid c), XA^c)$.

Note that we do not require the intermediate computation of s. Such intermediate values can be leaked in some scenarios and together with the leaked ephemerals can be used to compromise the long term keys. The modified HMQV protocol which uses the above technique for initiator is depicted in Figure 4. We denote the protocol as mHMQV-1.

Deniability of the Modified Schnorr Based Solution. The *initiator deniability* property is preserved. We state the following.

Theorem 8. *The mHMQV-1 protocol depicted in Figure 4 is initiator deniable.*

Proof. We have to show how the simulator $\mathrm{SIM}_{\mathscr{M}}^I$ (with Bob's view) would produce the transcript X, Y, Z, V, W, S which has exactly the same distribution as the transcript produced by two parties Alice and Bob together. The simulator $\mathrm{SIM}_{\mathscr{M}}^I$ computes values X, Y, Z, V, W, S, where $S = (\mathsf{H}_1(X \mid c))^s = (\mathsf{H}_1(X \mid c))^{x+ac}$ in the following way: It computes everything in the generator g first. It takes s, and y randomly computes $Y = g^y$, $m_A = \text{“Alice”} \| \text{“Bob”} \| A \| B \| Y$, and $c = \mathsf{H}_2(m_A)$. Afterwards it is able to compute the commitment of the first message $X = g^x = g^s/A^c$ accordingly (as in the example from Section 4.1). Y, Z, V are computed by Bob alone from his secrets as in HMQV. W is computed as $\mathsf{MAC}(\text{“0”}, k_m)$, because values k_m on both sides are equal as $\sigma_a = \sigma_b$. Then it computes $S = (\mathsf{H}_1(X \mid c))^s$. Note that it does not need to compute $(\mathsf{H}_1(X \mid c))^a$: this value is not a part of the transcript. Therefore the resulting transcript has exactly the same distribution as the transcript computed by Alice and Bob together. □

Proving of Interaction for Initiator. Note that in the *initiator deniable* version of the protocol mHMQV-1 in Figure 4 the transcript could have been produced by Bob alone, or together by Alice and Bob really interacting with each other. Therefore the simple trick can be made by Alice to have a proof of interaction. She simply has to remember $x = \log_g(X)$ as the commitment to the value X she uses in the first message. Usually ephemeral values are deleted once they are not needed anymore. However Alice may record the ephemeral value x and produce it in front of the judge

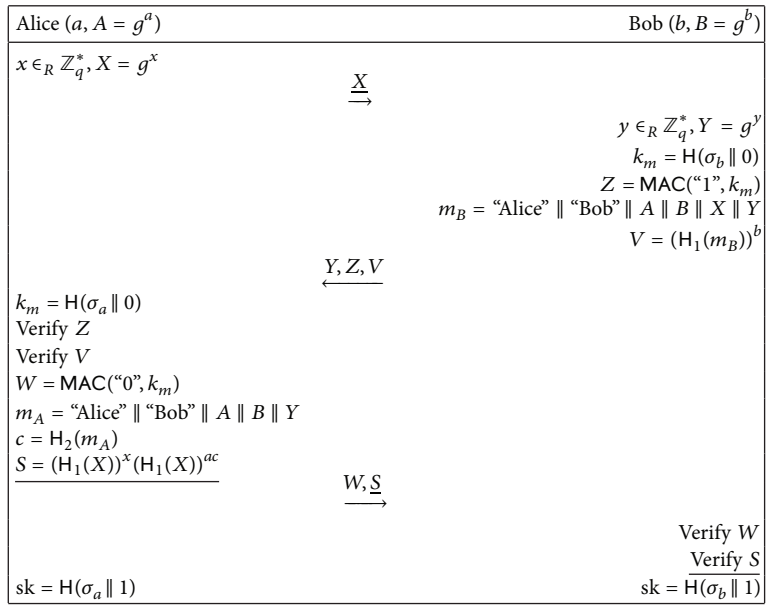

Figure 4: mHMQV-1: preventing eKCI-initiator deniable version.

to prove that the transcript, and particularly X, was not computed by Bob's simulation. Indeed if Bob is to present x he will have to break DLP problem for $X = g^s/A^c$. Still, if Alice does not store x, then no algorithm can tell if the transcript was the result of the protocol interaction or Bob's simulation.

Achieving Responder Deniability. The deniability of the responder also can be achieved; however it requires a slight modification of the protocol. The mechanism is symmetrical. The procedures of Bob mimic/reflect the behavior of Alice: this also requires an additional message from Bob at the end (so 4 messages in total). Note that storing values x and y enables Alice and Bob to prove the interaction according to reasoning from Section 4.2.

We state that the modified protocol depicted in the Figure 5 provides deniability for both Alice and Bob:

(i) The transcript can be simulated by the responder alone (Alice deniability).

(ii) The transcript can be simulated by the initiator alone (Bob deniability).

We call the protocol mHMQV-2. It is deniable for both the initiator and the responder.

Theorem 9. *The mHMQV-2 protocol depicted in Figure 5 is "initiator deniable."*

Proof. Essentially it is as the proof of Theorem 8. The only difference is that the value V is computed by the simulator as $(\mathsf{H}_1(Y \mid d))^y (\mathsf{H}_1(Y \mid d))^{bd}$ and included in the last fourth message (not in the second). □

Theorem 10. *The mHMQV-2 protocol depicted in Figure 5 is "responder deniable."*

Proof. Analogically it is as above. The simulator $\mathrm{SIM}_{\mathscr{M}}^R$ for the responder, with Alice's secrets, produces the transcript X, Y, Z, W, S, V, where $V = (\mathsf{H}_1(Y))^v$ should be equal to $(\mathsf{H}_1(Y \mid d))^{y+bd}$ for $c = \mathsf{H}_2(m_B)$: It starts with v and x uniformly at random, computes $X = g^x$, and sets $m_B = \text{“Alice”} \| \text{“Bob”} \| A \| B \| X$, and $h = \mathsf{H}_2(m_B)$. Then it computes $Y = g^y$ as g^v/B^d. Z is computed as $\mathsf{MAC}(\text{“1”}, k_m)$, because values k_m on both sides are equal as $\sigma_a = \sigma_b$. The parameters W, S can be easily computable with the input of Alice. Subsequently it computes $V = (\mathsf{H}_1(Y \mid h))^v$. □

5. Key Security and eKCI Resistance

In this point we discuss the security aspects of the proposed modification.

(i) Ephemeral Key Leakage Does Not Compromise Long Term Keys. This addresses the problem with the regular Schnorr authentication signalized in Section 4.1.

(ii) eKCI Resistance. The mHMQV protocols, extended with the proposed modification of Schnorr identification scheme, are resistant against eKCI attack, that is, are immune against impersonation attacks of the adversary authenticator which learns both the long term key and the ephemeral key of the verifier.

(iii) Session Key Security. The resulting protocol mHMQV still fulfills the session key security of the original unmodified version. In other words, the proposed modifications do not affect and impair the original AKE security.

The following theorem states that leakage of authenticator's ephemeral secret gives no advantage to the adversary whose goal is to extract the long term key.

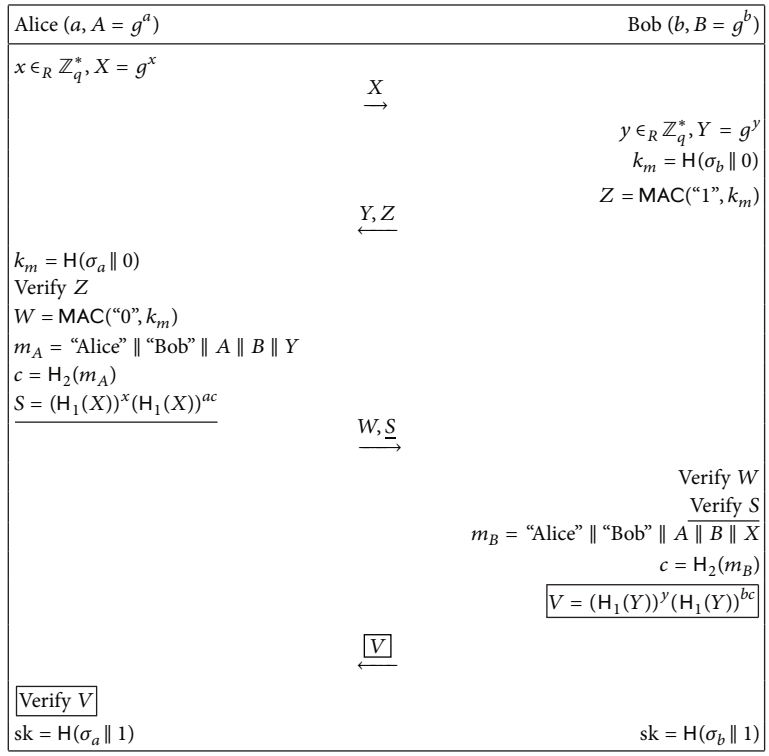

Figure 5: mHMQV-2: preventing eKCI-deniability for both the initiator and the responder.

Theorem 11. *No adversary can extract the long term secret key of the authenticator given public parameters, transcript of the protocol, and the ephemeral secret of the authenticator.*

Proof. The proof is by contradiction. W.l.o.g. let the authenticator be Alice, whose ephemeral key x is leaked. Now suppose that some algorithm $\mathcal{A}(A, B, X, Y, Z, V, W, S, x)$, when given the public parameters A, B, transcript of the protocol X, Y, Z, V, W, S, and the ephemeral secret of Alice x, outputs Alice's long term secret a in nonnegligible probability. Then we can use it as a subprocedure to break the DLP problem for a given value, say $U = g^u$, for unknown u. We have to prepare the input for $\mathcal{A}$, including U as public key of Alice, and S, as it would be computed by corresponding Alice's secret key u. We set up the system in which U is the public key of Alice and a random x is her ephemeral key. We simulate the transcript which would be indistinguishable from the real one. Hence we know x and we can compute X. Values Y, Z, V, W are also easily computable. The only problem here is to produce the suitable S. Indeed in ROM we program $(\mathsf{H}_1(X \mid c))$ as g^r for randomly chosen r. Then we compute $S = (g^r)^x (U)^{rc}$, which equals $(g^r)^x (g^u)^{rc} = (g^r)^x (g^r)^{uc} = (\mathsf{H}_1(X \mid c))^x (\mathsf{H}_1(X \mid c))^{uc}$. Then verification holds: $\widehat{e}(S, g) = \widehat{e}(\mathsf{H}_1(X \mid c), XU^c)$, and we obtain a perfect simulation in ROM. Now we treat the value output from $\mathcal{A}$ as the discrete logarithm of U. □

5.1. eKCI Resistance of mHMQV-1 and mHMQV-2. The eKCI resistance requires that the attacker cannot launch the impersonation attack, even if

(1) the attacker knows the long term key of the verifier,

(2) the ephemeral key of the verifier, after it is coined, is also leaked to the attacker as soon as it is coined.

The attacker is required to possess and use the secret key corresponding to the public key of the authenticator with identity ID, to be positively verified and accepted with this identity ID.

Remark 12. It is of the paramount importance, here, to strictly follow the protocol scheduled steps and implement the protocol in the designed order. Indeed, if the verifier carelessly changes the protocol schedule and prepares the challenge $Y = g^y$ before the very first step of the protocol (before receiving the commitment message X) and if the ephemeral y is leaked to the attacker before the first message, then it possible to impersonate any ID, say with public key U, but without corresponding secret u. In this case the attacker follows the simulator $\mathrm{SIM}_{\mathcal{M}}^{I}$: it starts with random s and the leaked y computes $Y = g^y$, $m_D = \text{“Dorothy”} \| \text{“Bob”} \| U \| B \| Y$, and $c = \mathsf{H}_2(m_U)$. Then it computes $X = g^x$ as g^s/U^c. Subsequently it computes $S = (\mathsf{H}_1(X \mid c))^s$. Then in the first message it sends to Bob precomputed X and later on after receiving Y it sends back precomputed S, impersonating itself in this way to Bob.

Theorem 13. *No adversary can authenticate as Alice in front of responder without the knowledge of the secret key "a" corresponding to public $A = g^a$ in mHMQV-1 and mHMQV-2 protocols.*

Proof.

(1) Reduction to Security of Mod-Schnorr [19]. The proof is an immediate consequence of the security of the mod-Schnorr identification scheme: any attacker that would impersonate Alice without her keys in mHMQV-1 and mHMQV-2 protocols would be used to break the underlying security of the mod-Schnorr identification scheme [19]. Conversely assume that there is an effective adversary $\mathcal{A}$ that impersonates Alice, without her secret key a, in front of Bob in mHMQV-1 protocol with nonnegligible probability. We use that adversary as a subprocedure to break mod-Schnorr in the following way: We play the role of Bob for $\mathcal{A}$. After obtaining X from $\mathcal{A}$ we forward it to our challenger as the first message. Then after obtaining c from our challenger we compute the values on Bob's side and send the second message to $\mathcal{A}$. Now after the adversary $\mathcal{A}$ issues an oracle query $\mathsf{H}_2(m_A)$ we set $\mathsf{H}_2(m_A) \leftarrow c$ in ROM table return value c. After $\mathcal{A}$ outputs S we forward it as the third message to our challenger. Note that if $\mathcal{A}$ is successfully accepted in mHMQV-1 then it is also accepted in mod-Schnorr.

(2) Reduction to CDH. Below we show how that adversary can be used to break the instance (g, g^α, g^β) of the underlying CDH problem, as in original paper [19]. Suppose the adversary $\mathcal{A}$ plays Alice in front of Bob without the knowledge of her secret key and is accepted. We give the adversary the secret key of Bob. Note that Bob's ephemeral key y can only be given (leaked to the adversary only ASAP after it is created on Bob's side). Since then y is another representation of the challenge $Y = g^y$. We set up the system for $\mathcal{A}$ with $A = g^\alpha$ as the public key of Alice. Then we use a *rewinding technique* (as in regular Schnorr identification): we fix the random value x used in $X = g^x$ by the algorithm $\mathcal{A}$ and let $\mathcal{A}$ interact twice with Bob, choosing each time a different random y, say y_1 and y_2. These will result with m_{A1}, c_1, S_1 and m_{A2}, c_2, S_2 accordingly. Note that on $\mathcal{A}$'s query to $\mathsf{H}_1(X \mid c)$ we answer with the value g^β. If Bob accepts both times we have $S_1 = (g^\beta)^x (g^\beta)^{\alpha c_1}$ and $S_2 = (g^\beta)^x (g^\beta)^{\alpha c_2}$. Thus we have $S_1/S_2 = (g^\beta)^{\alpha c_1 - \alpha c_2}$, so we can compute $g^{\alpha\beta} = (S_1/S_2)^{(c_1 - c_2)^{-1}}$. □

Theorem 14. *No adversary can be authenticated as Bob in front of Alice without the knowledge of the secret key b corresponding to public $B = g^b$ in mHMQV-2 protocol.*

Proof. The proof is similar to the proof of Theorem 13. We omit it to save the space. □

As a simple conclusion from Theorems 13 and 14 we state the following.

Corollary 15. *The protocols mHMQV-1 and mHMQV-2 are resistant to eKCI attacks.*

Now we address the security of the session key. This refers to the requirement that the session key established by the parties in the course of the protocol execution is known only to those parties. Usually the security model for the session key defines the so-called *session key security game*, in which the attacker is allowed to issue queries to various oracles, about the long term keys, and ephemeral keys of both parties. Usually the attacker is allowed to issue any combination of such queries, except those which would trivially reveal the session key. Eventually the attacker should not be able to distinguish whether the *test-key,* it was given, is the real established session key or some unrelated random value. However if it does distinguish that, with nonnegligible probability, it wins the security game, and the protocol is considered broken.

5.2. Session Key Security. To show the session key security we follow the same approach as in [17]. It is based on the actual HMQV security proven in [44]. Now observe that extension from [17] that immunes HMQV against eKCI only adds BLS layer for authentication purposes and does not affect the underlying session key security of HMQV. We follow the same approach. We want to show that our modifications do not spoil the session key security of the original HMQV. Our modified version adds some additional computation on each side, providing extra deniable authentication steps, against eKCI attack. This extra computation does not affect the session key security of the original HMQV. We take for granted that HMQV is "session-key-secure"; that is, no adversary $\mathcal{A}^{\mathrm{HMQV}}$ can learn the session key for the completed session between uncorrupted parties (refer [45] for proof of that in Canetti-Krawczyk model). Note that these extra computations can be easily simulated in ROM. Thus the execution of original HMQV can be easily transformed in execution of our mod versions. Now any attacker breaking the session key security of mHMQV could be used to break the session key security of org HMQV. We state the following.

Theorem 16. *If the original AKE protocol is "session-key-secure," then the modified protocol, extended with the authentication method proposed in Section 4.2, is also "session-key-secure" assuming programmable random oracle model.*

Proof. The proof is by contradiction. Assume that there exists an efficient adversary algorithm $\mathcal{A}^{\mathrm{mod}}$ that breaks the security of the modified protocol. We can use it as a subprocedure, to build the adversary algorithm $\mathcal{A}^{\mathrm{org}}$, which breaks the session key security of the original "unmodified" protocol. Observe that each oracle query from $\mathcal{A}^{\mathrm{mod}}$ can be served by $\mathcal{A}^{\mathrm{org}}$ via forwarding question and answers to/from corresponding oracles for org protocol. The only exception is queries concerning values S and V. These however can be easily simulated in ROM: for S we set $\mathsf{H}_1(X \mid c) \leftarrow g^r$ for some random r and compute $S = (\mathsf{H}_1(X \mid c))^x (\mathsf{H}_1(X \mid c))^{ac} = (g^r)^x (g^r)^{ac}$ as $X^r A^{rc}$ (for V we simulate similarly). This way we transform the transcript of the original protocol "org" into the transcript of the modified protocol "mod." Now any answer from $\mathcal{A}^{\mathrm{mod}}$ concerns the session key we output as the answer of $\mathcal{A}^{\mathrm{org}}$. If $\mathcal{A}^{\mathrm{mod}}$ wins the session key security game for mod, then also $\mathcal{A}^{\mathrm{org}}$ wins the security game for org. This would contradict assumption about session key security of org protocol. □

Table 2: Execution times for different protocol versions.

Protocol	1000 executions	Average time
3-pass HMQV	4,150.30 ms	4.15 ms
BLS-HMQV	16,565.33 ms	16.57 ms
mHMQV-0	18,505.76 ms	18.51 ms
mHMQV-1	27,336.94 ms	27.34 ms
mHMQV-2	33,992.05 ms	33.99 ms

Table 3: Average computation times for basic cryptographic building blocks used in the protocols.

Operation	Average time
Bilinear pairing	4.91 ms
Modular exponentiation	0.66 ms
Hash computing	0.14 ms
Multiplication	<0.01 ms
Addition	<0.01 ms

Table 4: Time complexity assessment for different protocol versions.

Protocol	ModPow	Hashing	Pairing	Other	Total
3-pass HMQV	1.74 ms	0.44 ms	0.00 ms	1.97 ms	4.15 ms
BLS-HMQV	3.29 ms	1.23 ms	9.71 ms	2.34 ms	16.57 ms
mHMQV-0	2.99 ms	1.07 ms	9.43 ms	5.02 ms	18.51 ms
mHMQV-1	5.20 ms	1.78 ms	15.41 ms	4.95 ms	27.34 ms
mHMQV-2	6.89 ms	1.21 ms	21.03 ms	4.86 ms	33.99 ms

6. Performance

Each of the proposed modifications of the scheme strengthens its security but requires performing certain amount of additional computations, which should be expected to affect the overall performance of the protocol. We implemented the basic scheme *(3-pass HMQV)*, the BLS based scheme *(BLS-HMQV)*, the basic Schnorr based imperfect modification *(mHMQV-0)*, the modified Schnorr based initiator deniable version *(mHMQV-1),* and the modified Schnorr based fully deniable version *(mHMQV-2)* in order to measure how much do the proposed improvements extend the execution time of the protocol.

6.1. Implementation. Our implementations have been created using *Python 3* with the *Charm Crypto* library [46], a commonly used open-source cryptographic toolbox providing methods to perform operations on elliptic curves, including bilinear pairings and hashing and the *timeit* for measuring the average execution times. All computations are performed on the same NIST-approved symmetric elliptic curve with a 512-bit base field [47]. In order to measure nothing but the time of computations strictly related to the schemes, each implementation is created as a single program, where the two parties are simulated by interweaving methods.

6.2. Results. Average execution time for each protocol has been measured by running 1000 full rounds of each version on a *Ubuntu 12* virtual machine with *Intel i7* 2.5 GHz and 8 GB RAM. The acquired results are presented in Table 2. As it was expected, each modified version of the protocol is 4 to 8 times slower than the original one. This is intuitively self-explained: each subsequent modification requires additional computing hashes and bilinear pairings.

As a first step in assessing which modifications affect the execution time of the protocols, we had to measure the execution times for every *building block* operation used in the protocol. The results can be seen in Table 3. It emerges that the bilinear pairing operation, crucial for our modifications, requires relatively much computational power. However, to better assess the protocols, we have measured how much time is taken for the most complex *building blocks*, as it has to be noticed that each of them may be used multiple times in a single protocol round.

A detailed assessment of time complexity for every protocol version has been presented in Table 4. It depicts how much time in the protocols is consumed by computing bilinear pairings, hashes, and modular exponents (ModPow operations). One can easily notice that the longer execution time in the modified versions resulted mostly from usage of bilinear pairings. One should take it into consideration while implementing these schemes, as some hardware enhancements could possibly improve the performance of the pairing routines and, as a result, the entire protocol.

Nevertheless, it has to be pointed that the average execution time remains to be just several milliseconds in all the cases, making any of the proposed modification applicable to implementation in real-world usage.

7. Conclusion

In this paper we extended the results from [17]. We observed that the solution from [17], protecting HMQV against the eKCI attack, destroys the deniability property of HMQV. Therefore, following the two-layer construction of [17], we exchange the undeniable BLS signatures layer, with the modified Schnorr identification scheme from [19] resistant to ephemeral key leakages. This way we immune HMQV against eKCI in such a way that the deniability property is preserved. Compared with the undeniable solution from [17], in our *initiator deniable* version of the protocol Alice needs to compute one more exponentiation. The *initiator and responder* deniable version requires two more exponentiations (one more per side) and the additional fourth message. The conducted experiments confirmed that, despite the additional computational effort, the newly proposed protocols remain efficient enough to be implemented in real-world applications.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was partially supported by funding from Polish NCN Contract no. DEC-2013/09/D/ST6/03927.

References

[1] M. Di Raimondo, R. Gennaro, and H. Krawczyk, "Deniable authentication and key exchange," in *Proceedings of the CCS 2006: 13th ACM Conference on Computer and Communications Security*, pp. 400–409, Alexandria, Va, USA, November 2006.

[2] L. Krzywiecki, "Deniable version of SIGMA key exchange protocol resilient to ephemeral key leakage," in *Proceeding of the Provable Security - 8th International Conference, (ProvSec '14)*, S. S. M. Chow, J. K. Liu, L. C. K. Hui, and S. Yiu, Eds., vol. 8782 of *Lecture Notes in Computer Science*, pp. 334–341, Springer, Hong Kong , China, 2014.

[3] L. Law, A. Menezes, M. Qu, J. Solinas, and S. Vanstone, "An efficient protocol for authenticated key agreement," *Designs, Codes and Cryptography*, vol. 28, no. 2, pp. 119–134, 2003.

[4] H. Krawczyk, "Hmqv: A high-performance secure diffie-hellman protocol," in *CRYPTO*, V. Shoup, Ed., vol. 3621 of *Lecture Notes in Computer Science*, pp. 546–566, Springer, Berlin, Germany, 2005.

[5] H. Krawczyk, "SIGMA: The "SIGn-and-MAc" Approach to Authenticated Diffie-Hellman and Its Use in the IKE Protocols," in *Advances in Cryptology - CRYPTO 2003*, vol. 2729 of *Lecture Notes in Computer Science*, pp. 400–425, Springer, Berlin, Germany, 2003.

[6] K. Lauter and A. Mityagin, "Security analysis of kea authenticated key exchange protocol," in *Public key cryptography*, vol. 3958 of *Lecture Notes in Comput*, pp. 378–394, Springer, Berlin, Germany, 2006.

[7] B. A. LaMacchia, K. Lauter, and A. Mityagin, "Stronger security of authenticated key exchange," in *Provable Security*, W. Susilo, J. K. Liu, and Y. Mu, Eds., vol. 4784 of *Lecture Notes in Computer Science*, pp. 1–16, Springer, Berlin, Germany, 2007.

[8] B. Ustaoglu, "Obtaining a secure and efficient key agreement protocol from (H) MQV and NAXOS," *Designs, Codes and Cryptography*, vol. 46, no. 3, pp. 329–342, 2008.

[9] A. P. Sarr, P. Elbaz-Vincent, and J.-C. Bajard, "A new security model for authenticated key agreement," in *Security and Cryptography for Networks*, J. A. Garay and R. D. Prisco, Eds., vol. 6280 of *Lecture Notes in Computer Science*, pp. 219–234, Springer, Berlin, Germany, 2010.

[10] Q. Cheng, C. Ma, and X. Hu, "A new strongly secure authenticated key exchange protocol," in *Advances in Information Security and Assurance*, vol. 5576 of *Lecture Notes in Computer Science*, pp. 135–144, Springer, Berlin, Germany, 2009.

[11] H. Huang, "Strongly secure one round authenticated key exchange protocol with perfect forward security," in *Provable Security*, vol. 6980 of *Lecture Notes in Computer Science*, pp. 389–397, Springer, Berlin, Germany, 2011.

[12] M. Kim, A. Fujioka, and B. Ustaoğlu, "Strongly secure authenticated key exchange without NAXOS' approach," in *Advances in Information and Computer Security*, vol. 5824 of *International Workshop on Security*, pp. 174–191, Springer, Berlin, Germany, 2009.

[13] R. Canetti and H. Krawczyk, "Analysis of key-exchange protocols and their use for building secure channels," in *EUROCRYPT*, B. Pfitzmann, Ed., vol. 2045 of *Lecture Notes in Computer Science*, pp. 453–474, Springer, Berlin, Germany, 2001.

[14] S. Blake-Wilson, D. Johnson, and A. Menezes, "Key agreement protocols and their security analysis," in *Cryptography and coding (Cirencester, 1997)*, M. Darnell, Ed., vol. 1355 of *Lecture Notes in Computer Science*, pp. 30–45, Springer, Berlin, Germany, 1997.

[15] C. Boyd and A. Mathuria, "Authentication and key transport using public key cryptography," in *Protocols for Authentication and Key Establishment*, pp. 107–135, Springer, Berlin, Germany, 2003.

[16] A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, New York, NY, USA, 1997.

[17] Q. Tang and L. Chen, "Extended KCI attack against two-party key establishment protocols," *Information Processing Letters*, vol. 111, no. 15, pp. 744–747, 2011.

[18] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the weil pairing," *Journal of Cryptology*, vol. 17, no. 4, pp. 297–319, 2004.

[19] L. Krzywiecki, "Schnorr-like identification scheme resistant to malicious subliminal setting of ephemeral secret in," in *Innovative Security Solutions for Information Technology and Communications - 9th International Conference, SECITC, 2016*, I. Bica and R. Reyhanitabar, Eds., Lecture Notes in Computer Science, pp. 137–148, Bucharest , Romania, 2016.

[20] E. B. Barker and J. M. Kelsey, "Recommendation for random number generation using deterministic random bit generators," National Institute of Standards and Technology NIST SP 800-90a, Gaithersburg, MD, USA, 2012.

[21] M. Matsumoto, M. Saito, H. Haramoto, and T. Nishimura, "Pseudorandom number generation: impossibility and compromise," *Journal of Universal Computer Science*, vol. 12, no. 6, pp. 672–690, 2006.

[22] M. Matsumoto and T. Nishimura, "Mersenne Twister: a 623-dimensionally equidistributed uniform pseudo-random number generator," *ACM Transactions on Modeling and Computer Simulation*, vol. 8, no. 1, pp. 3–30, 1998.

[23] M. Matsumoto, T. Nishimura, M. Hagita, and M. Saito, "Cryptographic mersenne twister and fubuki stream/block cipher,"

IACR Cryptology ePrint Archive, vol. 165, 2005, http://eprint.iacr.org/2005/165.

[24] C. Ye, S. Mathur, A. Reznik, Y. Shah, W. Trappe, and N. B. Mandayam, "Information-theoretically secret key generation for fading wireless channels," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 2, pp. 240–254, 2010.

[25] G. Lo Re, F. Milazzo, and M. Ortolani, "Secure random number generation in wireless sensor networks," *Concurrency and Computation: Practice and Experience*, vol. 27, no. 15, pp. 3842–3862, 2015.

[26] A. Sadr and M. Zolfaghari-Nejad, "Physical unclonable function (PUF) based random number generator," in *CoRR abs/1204.2516*, http://arxiv.org/abs/1204.2516.

[27] S. S. Zalivako and A. A. Ivaniuk, "The use of physical unclonable functions for true random number sequences generation," *Automatic Control and Computer Sciences*, vol. 47, no. 3, pp. 156–164, 2013.

[28] J. Lee and J. H. Park, *Authenticated key exchange secure under the computational diffie-hellman assumption*, 2008, http://eprint.iacr.org/2008/344.pdf.

[29] L. Hanzlik, K. Kluczniak, L. Krzywiecki, and M. Kutylowski, "Mutual chip authentication," in *Proceedings of the 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, (TrustCom '13)*, pp. 1683–1689, Melbourne, VIC, Australia, July 2013.

[30] L. Hanzlik, K. Kluczniak, M. Kutylowski, and L. Krzywiecki, "Mutual restricted identification," in *Public Key Infrastructures, Services and Applications*, S. K. Katsikas and I. Agudo, Eds., vol. 8341 of *Lecture Notes in Computer Science*, pp. 119–133, Springer, Berlin, Germany, 2014.

[31] R. Canetti and H. Krawczyk, "Security analysis of IKE's signature-based key-exchange protocol," Cryptology ePrint Archive, Springer, Berlin, Germany, 2002, http://eprint.iacr.org/.

[32] S. H. Islam and G. P. Biswas, "Design of two-party authenticated key agreement protocol based on ECC and self-certified public keys," *Wireless Personal Communications*, vol. 82, no. 4, pp. 2727–2750, 2015.

[33] H. Sun, Q. Wen, and W. Li, "A strongly secure pairing-free certificateless authenticated key agreement protocol under the CDH assumption," *Science China Information Sciences*, vol. 59, no. 3, 2016.

[34] C. Büttner and S. A. Huss, "A novel anonymous authenticated key agreement protocol for vehicular ad hoc networks," in *Proceedings of the 1st International Conference on Information Systems Security and Privacy, ESEO (ICISSP '15)*, O. Camp, E. R. Weippl, C. Bidan, and E. Ameur, Eds., pp. 259–269, SciTePress, Angers, Loire Valley, France, 2015, https://doi.org/10.5220/0005238902590269.

[35] J. Zhang, Z. Zhang, J. Ding, M. Snook, and Ö. Dagdelen, "Authenticated key exchange from ideal lattices," in *Advances in cryptology-EUROCRYPT 2015*, vol. 9057, pp. 719–751, Springer, Berlin, Germany, 2015.

[36] R. Chen, Y. Mu, G. Yang, W. Susilo, and F. Guo, "Strong authenticated key exchange with auxiliary inputs," *Designs, Codes and Cryptography*, vol. 85, no. 1, pp. 145–173, 2017.

[37] R. Chen, Y. Mu, G. Yang, W. Susilo, and F. Guo, "Strongly leakage-resilient authenticated key exchange," in *Topics in Cryptology - CT-RSA 2016*, vol. 9610 of *Lecture Notes in Computer Science*, pp. 19–36, Springer International Publishing, Cham, Switzerland, 2016.

[38] M. Feltz and C. Cremers, "Strengthening the security of authenticated key exchange against bad randomness," *Designs, Codes and Cryptography*.

[39] A. Fujioka, K. Suzuki, K. Xagawa, and K. Yoneyama, "Strongly secure authenticated key exchange from factoring, codes, and lattices," *Designs, Codes and Cryptography*, vol. 76, no. 3, pp. 469–504, 2015.

[40] C. Boyd, C. Cremers, M. Feltz, K. G. Paterson, B. Poettering, and D. Stebila, "Asics: authenticated key exchange security incorporating certification systems," *International Journal of Information Security*, vol. 16, pp. 151–171, 2017.

[41] L. Krzywiecki and M. Kutylowski, "Security of okamoto identification scheme: a defense against ephemeral key leakage and setup," in *Proceedings of the Fifth ACM International Workshop on Security in Cloud Computing, (SCC@AsiaCCS '17)*, C. Wang and M. Kantarcioglu, Eds., pp. 43–50, Abu Dhabi, UAE, April 2017.

[42] C. P. Schnorr, "Efficient signature generation by smart cards," *Journal of Cryptology*, vol. 4, no. 3, pp. 161–174, 1991.

[43] J. Bender, Ö. Dagdelen, M. Fischlin, and D. Kügler, "The pace|AA protocol for machine readable travel documents, and ts security," in *Financial Cryptography and Data Security - 16th International Conference, (FC '12)*, A. D. Keromytis, Ed., vol. 7397 of *Lecture Notes in Computer Science*, pp. 344–358, Februray 27-March 2, 2012.

[44] H. Krawczyk, "HMQV: a high-performance secure Diffie-Hellman protocol," in *Advances in Cryptology—CRYPTO 2005*, vol. 3621 of *Lecture Notes in Computer Science*, pp. 546–566, Springer, Berlin, Germany, 2005.

[45] R. Canetti and H. Krawczyk, "Security analysis of IKE's signature-based key-exchange protocol," in *Advances in Cryptology - CRYPTO 2002, 22nd Annual International Cryptology Conference*, M. Yung, Ed., vol. 2442 of *Lecture Notes in Computer Science*, pp. 143–161, Santa Barbara, Calif, USA, 2002.

[46] J. A. Akinyele, C. Garman, I. Miers et al., "Charm: a framework for rapidly prototyping cryptosystems," *Journal of Cryptographic Engineering*, vol. 3, no. 2, pp. 111–128, 2013.

[47] C. NIST, "The digital signature standard," http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf.