Mutual Restricted Identification[*]

Lucjan Hanzlik, Kamil Kluczniak, Mirosław Kutyłowski, and Łukasz Krzywiecki

Faculty of Fundamental Problems of Technology,
Wrocław University of Technology
firstname.secondname@pwr.wroc.pl

Abstract. We extend the idea of Restricted Identification deployed in the personal identity documents in Germany. Our protocol, Mutual Restricted Authentication (MRI for short), is designed for direct anonymous authentication between users who belong to the same domain (called also a sector). MRI requires *only one* private key per user. Still there are no limitations to which domain a user may belong and the domains are not fixed in advance. This enables an implementation of MRI when a strictly limited secure memory is available (like for smart cards). MRI guarantees that a user has exactly one identity within a domain, while the identities from different domains of the same user are not linkable. The main difference between RI and MRI is that for MRI the privacy of both participants are protected, while in case of RI the terminal is fully exposed. The protocol is efficient, extremely simple (in particular, it outperforms RI) and well suited for an implementation on resource limited devices such as smart cards.

Keywords: personal ID document, Restricted Identification, privacy, simultability, authentication, AKE.

1 Introduction

In pervasive systems one of the key issues is identifying and authenticating digital artefacts. This concerns all electronic identity documents but also other devices like smartphones, tablets, and identification tokens. So we have to talk about an *electronic-ID* (e-ID for short). In some cases the same e-ID has to play different roles in different subsystems – called from now on *domains* – and use a different identity in each domain. Unless necessary, an *e-ID* device should not use linkable identities in different domains. E.g., professional and private roles should be strictly separated.

For technical and usability reasons wireless communication will play a dominant role for communication with e-ID devices. So protecting information exchange against eavesdroppers becomes a key issue. Information on membership of, and identity in, a domain should also be protected. Moreover, tough rules on personal data protection and social sensitivity in countries like UK and Germany make it necessary to *guarantee* effective protection.

Today, most e-ID systems do not hide the identity of at least one party. This is the case for *machine readable travel documents* (that is, electronic passports and personal

[*] This work is based on business and security analysis done in project O ROB 0027 01/ID 27/2 financed by NCBiR.

S. Katsikas and I. Agudo (Eds.): EuroPKI 2013, LNCS 8341, pp. 119–133, 2014.
© Springer-Verlag Berlin Heidelberg 2014

identity documents) and so the terminals *must* be trusted. However, if e-ID devices wish to interact directly, privacy of both sides should be protected.

Domains and Restricted Identification. The idea of separating activity areas was implemented first in the Austrian *Bürgerkarte* - this system is based on the passwords computed with a symmetric algorithm from the citizen's personal number.

The next step was development of the *nPA*, the new German personal identity document. Restricted Identification (RI for short) protocol [1], allows an nPA to use a single private key to authenticate against any terminal. Each nPA uses its private key to compute its domain specific identifiers. The key feature of RI is unlinkability: two terminals from two different domains cannot determine if they are interacting with the same nPA or with two different nPA's. However, within a single domain all actions of an nPA *must* be attributed to the same anonymous identity.

The protocol from nPA requires a prior execution of the Terminal Authentication (TA) protocol, during which the terminal signs with its private key a nonce provided by the nPA. Thereby the transcript of communication can be used as an *undeniable proof* of interaction with the terminal. Therefore, this protocol is not suitable for the case of peer-to-peer communication between e-IDs.

Design Goals. In this paper we develop Mutual Restricted Identification protocol (MRI) that expands RI [1].

MRI is fully *simultable*, i.e. each side of the protocol can compute a transcript of communication that is indistinguishable from the transcripts obtained from real communications. This resolves the problem stated above, no participant can use a communication transcript as a proof against a third party. MRI provides *unlinkability* for activities in different domains, just as in case of RI. MRI is *symmetric* regarding operations performed by both sides of the protocol. This feature has a positive impact on implementation costs and flexibility. MRI is is *resilient to leakages* – in many scenarios revealing ephemeral keys does not disclose the session keys (which is not true for nPA). This also concerns forward security: revealing long-term secrets does not reveal the session keys. MRI is slightly more *efficient* than RI. Therefore, it is well suited feasible for smart cards implementation (which has been also confirmed by an implementation on Java Cards).

Previous Work. There are many papers on authenticated key exchange (AKE). The AKE protocols secure in the Canetti-Krawczyk (CK) model [2], guarantee that the adversary cannot distinguish established keys from random values, as long as some session secrets (ephemeral keys) are not leaked. In [3] Krawczyk proposed a variant of CK and proved that the HMQV protocol (a hashed version of MQV from [4]) achieves so called weak perfect forward security (wPFS), resilience to key compromise impersonation (KCI) attacks and revealing the ephemeral keys of a single party. The extended Canetti-Krawczyk model (eCK) was proposed in [5] to capture combinations of static and ephemeral keys corruptions (apart from the obvious ones that break security by definition), including revealing both ephemeral keys or both static keys. NAXOS [5], NAXOS+ [6], and CMQV [7] were shown to be secure in eCK model.

The KEA+ protocol [8] was shown to be secure in a model weaker that eCK that allows revealing the long-term key of at most one of the parties.

In the above mentioned protocols each party has prior knowledge on the ID of the other party, or the identifiers are sent during a protocol execution. The later case may lead to privacy violations, thus identity hiding was concerned in the papers [9,10,11,12]. Deniability, as an additional feature was achieved in the PACE|AA protocol [13]. In this protocol each party can create transcripts of protocol runs with the same probability distributions as for the transcripts coming from the real protocol executions. Deniability of SKEME and partial deniability of SIGMA were discussed in [14].

From the above mentioned protocols based on DH key exchange (without pairings) none fully satisfied the required goals:

- the following protocols are not deniable: NAXOS, NAXOS+, JFKi, JFKr, SIGMA,
- the following protocols are not identity hiding: MQV,HMQV,CMQV,
- the following protocols use prior knowledge of the partner's ID: KEA+, NAXOS, NAXOS+, SKEME.

On the other hand the protocols [10] and [11] are based on pairings, and it is not clear how could they be adjusted for restricted identification.

The Restricted Identification protocol has its variant called ChARI, which redefines initial steps and eliminates so called *group keys* shared by many e-IDs. The price paid is a slight loss of efficiency and the use of separate certificates, whitelists or blacklists for domains. Below we present an efficiency comparison for RI, ChARI and MRI:

Table 1. Efficiency comparison for RI protocols

protocol	exponentiations on a smart card	exponentiations on terminal	communication rounds	number of private keys on a smart card
RI [1]	2 + 2	2 + 1	3	2
ChARI [15]	2 + 2	3 + 1	3	1
MRI (this paper)	3	3	2	1

2 Mutual Restricted Identification

Below we use a cyclic group $\mathcal{G}$ of a prime order q where the Discrete Logarithm Problem is hard.

Domains. Two users can authenticate themselves if they belong to the same *domain*. On the other hand, a user may belong to any number of domains. For a domain S there is a uniquely defined generator $g_S \in \mathcal{G}$ used by all users. g_S must be derived in a way that the discrete logarithm of g_{S_1} with respect to g_{S_2} is unknown for any domains S_1, S_2, $S_1 \neq S_2$. For instance, we can use a hash function mapping the legal names to $\mathcal{G}$, that is, $g_S = H(S)$.

Table 2. Mutual Restricted Identification protocol

Alice		Bob
x_A - private key $y_A = g^{x_A}$ - public key $cert_A$ - certificate for y_A		x_B - private key $y_B = g^{x_B}$ - public key $cert_B$ - certificate for y_B
	OPTIONAL SETUP	
recompute g $y_A := g^{x_A}$ - set public key fetch $cert_A$ and check y_A		recompute g $y_B := g^{x_B}$ - set public key fetch $cert_B$ and check y_B
	MAIN PROCEDURE	
choose a at random		choose b at random
$h_A := H(a\mid 0)$		$h_B := H(b\mid 0)$
$c_A := y_A^{h_A}$	$\xrightarrow{c_A}$	$c_B := y_B^{h_B}$
	$\xleftarrow{c_B}$	
$K := {c_B}^{x_A h_A}$		$K := {c_A}^{x_B h_B}$
$K_A := H(K\mid 1)$, $K_B := H(K\mid 2)$	$\xrightarrow{Enc_{K_A}(a, cert_A)}$	$K_A := H(K\mid 1)$, $K_B := H(K\mid 2)$
		reject if $c_A \neq y_A^{H(a\mid 0)}$ or $cert_A$ invalid
reject if $c_B \neq y_B^{H(b\mid 0)}$ or $cert_B$ invalid	$\xleftarrow{Enc_{K_B}(b, cert_B)}$	
$K_s := H(K\mid 3)$		$K_s := H(K\mid 3)$

Initialization. The protocol is described on Fig. 2. Note that some initial steps are omitted: such as negotiating the encoding format, the communication parameters, the algorithms and group used, etc. This stage must be based on a temporal ad hoc identity and there must be a very limited number of behavior profiles during this phase in order to eliminate identification.

In the following description we assume that the communication is within domain S with the generator $g_S = g$. The certificates for the public keys of, respectively, Alice and Bob in the domain S will be denoted by $cert_A$ and $cert_B$.

Protocol Idea. The first part of the protocol is deriving the master session key K by the Diffie-Hellman protocol based on the values c_A and c_B. At this stage the identities of the participants are not revealed. At the first look it may appear that derivations of K depend on the participants' identities. However, c_A and c_B are in fact equal to $g^{x_A h_A}$ and $g^{x_B h_B}$, and as h_A and h_b are in some sense "random", so are $x_A h_A$ and $x_B h_B$ modulo q.

Note that the key K depends on the domain parameter $g = g_S$. Indeed, if A uses g and B uses a different key g', then A derives $(c_B)^{x_A h_A} = (g')^{x_B h_B x_A h_A}$, while B derives $(c_A)^{x_B h_B} = g^{x_A h_A x_B h_B}$. So the results are different.

The master key K is used to get a number of keys by applying a hash function with different parameters. We follow a frequent practice to yield "independent" keys by hashing a shared secret expanded with different parameters.

The second stage of the protocol is communicating the values a and b. The purpose is the following: knowing the session key by A is an evidence of knowledge of the discrete logarithm of c_A with respect to g. So, as A knows the discrete logarithm of c_A with respect to y_A, we conclude that A may easily derive the discrete logarithm of y_A with respect to g. Thereby after terminating the protocol execution in an accepting state we may conclude that A knows the secret key x_A.

Note that while c_A is "random", finding a such that $c_A = y_A^{h_A}$ is infeasible, if c_A has not been computed in this way. Indeed, possibility of deriving h_A would break the Discrete Logarithm Problem. Namely, we would challenge the adversary with $c_A = g^r$ for r chosen at random, get back h such that $y_A^h = c_A$, and then derive $y_A = g^{r/h}$.

Encrypting a and b has two goals. First, it protects identity information from an eavsdropper. Second, verification is possible only if the recipient knows the decryption key, and therefore has been participating in the whole interaction.

3 Security Assumptions

Definition 1 (DDH Assumption). *Let $\mathcal{G}$ be a cyclic group of a prime order q'. The* Decisional Diffie-Hellman Problem *(DDH Problem) is hard for $\mathcal{G}$ if there is no probabilistic polynomial-time algorithm $\mathcal{A}_{\mathrm{DDH}}$ that with a non-negligible probability distinguishes between the distributions $D_0 = (\tilde{g}, \tilde{g}^{\alpha}, \tilde{g}^{\beta}, \tilde{g}^{\gamma})$ and $D_1 = (\tilde{g}, \tilde{g}^{\alpha}, \tilde{g}^{\beta}, \tilde{g}^{\alpha\beta})$, where α, β, γ are chosen at random from $\{1, \dots, q' - 1\}$. That is, for any probabilistic polynomial-time algorithm $\mathcal{A}_{\mathrm{DDH}}$ the adversary's advantage*

$$\mathbf{Adv}(\mathcal{A}_{\mathrm{DDH}}) = |\Pr[\mathcal{A}_{\mathrm{DDH}}(D_1) = 1] - \Pr[\mathcal{A}_{\mathrm{DDH}}(D_0) = 1]|$$

is at most ϵ_{DDH} for a negligibly small ϵ_{DDH}.
The Computational Diffie-Hellman Problem *(CDH Problem) is to derive $g^{\alpha\beta}$ given g^{α} and g^{β}. If the DDH Problem is hard, then there is no efficient algorithm solving the CDH Problem.*

In order to model requirements for a hash function we use the notion of correlated-input secure hash functions [16].

Definition 2. *A hash function H is* correlated-input secure *if for a random r and any Boolean circuits $C_1, \dots, C_n$ there is no adversary such that given $H(C_1(r)), \dots, H(C_{n-1}(r))$, it distinguishes between $H(C_n(r))$ and a random R of the same length with a non-negligible probability within realistic time.*

For the encryption function we use the Ideal Cipher Model, closely related to Random Oracle Model.

Definition 3. *In the* Ideal Cipher Model *encryption is modelled by an oracle $\mathcal{O}$ that holds a table T storing triples (m, k, c), where m stands for a plaintext, k stands for an encryption key, and c stands for a ciphertext. Initially, T is empty.*

Given a query Encrypt(m,k), the oracle $\mathcal{O}$ checks if there is an entry of the form (m, k, c) in T. If yes, then $\mathcal{O}$ responds with c. Otherwise, $\mathcal{O}$ chooses c' at random, but different from all z such that there is an entry (h, k, z) in T. Then $\mathcal{O}$ responds with c' and inserts (m, k, c') in T.

Given a query Decrypt(c,k), the oracle $\mathcal{O}$ checks if there is an entry of the form (m, k, c) in T. If yes, then $\mathcal{O}$ responds with m. Otherwise $\mathcal{O}$ chooses m' at random, but different from all z such that there is an entry (h, k, z) in T. Then $\mathcal{O}$ responds with m' and inserts (m', k, c) in T.

4 Privacy Issues

Proofs of Interaction. One of the key privacy problems is that a transcript of a protocol can be used by a communicating party or by an eavesdropper to prove that an interaction with a certain party has occurred. This provides motivation to solutions based on the Zero-Knowledge Proof principle, where an interaction can be perfectly simulated and therefore is useless for proving anything. The paper [17] states this property more explicitly as *simultability* of protocol executions.

Proposition 1. *B (respectively, A) can generate a proof consisting of all data transferred during an alleged execution of the MRI protocol together with all internal values used by B (A) without any interaction with A but with exactly the same probability distribution as for the real interactions.*

Proof. B creates a fake transcript by performing all steps on behalf of A and B. The only difference is that B does not attempt to derive K as it is done by A. However, this is unnecessary, since B can compute K using its own procedure.

Creating a fake transcript by A is similar. □

Another possibility is that an eavesdropper holding neither x_A nor x_B presents an interaction transcript. Potentially, it can contain some data that cannot be created without involvement of A or B – in this case we have a proof that either this is a real transcript or a simulated one created by either A or B. However, if we may assume that A and B are honest, then we get a proof of interaction between A and B. Below we show that there is no such a danger for the MRI protocol.

Proposition 2. *In the Ideal Cipher Model under the DDH Assumption, given a transcript of an interaction consisting of c_A, c_B, Enc$_1$ and Enc$_2$, it is infeasible to identify the protocol participants. More precisely, the advantage of the adversary to win the following game is negligible: for arbitrary participants A_0, B_0 and A_1, B_1:*

- *the challenger chooses a bit u at random,*
- *the challenger presents a record T consisting of the messages exchanged between A_u, B_u during a real execution of the MRI protocol,*
- *the adversary responds with u'. He wins if $u' = u$.*

Proof. The original game can be formalized as follows:

Game 0.

```
choose $u$, $a$, $b$ at random
$h_A := H(a|0)$, $h_B := H(b|0)$, $c_A := y_{A_u}^{h_A}$, $c_B := y_{B_u}^{h_B}$
$K := g^{x_{A_u} x_{B_u} h_B h_A}$, $K_A := H(K|1)$, $K_B := H(K|2)$
$E_1 := \mathit{Enc}_{K_{A_u}}(a, \mathit{cert}_{A_u})$, $E_2 := \mathit{Enc}_{K_{B_u}}(b, \mathit{cert}_{B_u})$
$u' := \mathcal{A}(c_A, c_B, E_1, E_2)$
```

The encryption operations above are understood as calls to the encryption oracle $\mathcal{O}$. Now we replace the encryption results with random variables:

Game 1.

```
    choose $u$, $a$, $b$ at random
    $h_A := H(a|0), h_B := H(b|0), c_A := y_{A_u}^{h_A}, c_B := y_{B_u}^{h_B}$
    $K := g^{x_{A_u} x_{B_u} h_B h_A}, K_A := H(K|1), K_B := H(K|2)$
    choose $E_1$ and $E_2$ at random and inform oracle $\mathcal{O}$ about them
    $u' := \mathcal{A}(c_A, c_B, E_1, E_2)$
```

In Game 1 we simply reverse the order of operations concerning encryption oracle. Instead of asking $\mathcal{O}$ during an encryption we create the values and demand to include these values in the table kept by $\mathcal{O}$. This may lead to conflicts with already existing values and thereby to a fault event. However, this is very unlikely.

In Game 1 the adversary gets 4 values that are uniformly distributed. However, these values are entangled by entries that exist in the table of the encryption oracle $\mathcal{O}$. Disclosing these relations is possible only after asking the oracle $\mathcal{O}$ with the key K_A or K_B. Assume that this is possible with a non negligible probability. We show that then we would be able to solve the DDH Problem. Indeed, for a given instance (g, C, D, Z) we play Game 1 with $y_{A_0} = C$, $y_{B_0} = D$, and observe the queries to $\mathcal{O}$. If any key equals $H(Z^{h_A h_B}|1)$ or $H(Z^{h_A h_B}|2)$, then we have an indication that (C, D, Z) is a Diffie-Hellman triple. □

Passive Adversary and Linking Attempts

Definition 4 (passive adversary privacy model). *We assume that $A_1, \dots, A_k$ can communicate within domains $S_1, \dots, S_u$. During the time period considered, the adversary observes t interactions, say $T_1, \dots T_t$, and participates itself in some number of interactions T (at arbitrary time moments). The adversary knows the participants $A_1, \dots, A_k$ and their public keys for each domain. An elementary event in the probability space Ω is a mapping that indicates for each transaction the communicating parties and the domain used:*

$$R : \{T_1, \dots, T_k\} \longrightarrow \{A_1, \dots, A_k\}^2 \times \{S_1, \dots, S_u\}$$

A priori knowledge of the adversary is a probability distribution π on Ω.

The probability distribution π on Ω models the knowledge resulting from the real conditions. E.g. if the transmission times of T_i and T_{i+1} overlap, then usually we may conclude that the participants of T_i and T_{i+1} are different.

Definition 5 (attack model for the passive adversary). *Let D be the list of all messages exchanged during some protocol executions observed by the adversary. We consider the distributions π and $\pi|D$ (the probability distribution π conditioned by the data D observed). We say that the protocol is* secure against linking, *if the distributions π and $\pi|D$ do not differ in a non-negligible way. That is, given a sample drawn from distribution π or $\pi|D$, the adversary has no non-negligible advantage to guess whether the sample has been drawn from π or $\pi|D$.*

Definition 5 says that the data sent by the protocol does not add substantial *new* knowledge for determining *who is talking with whom*. Note also that π might be arbitrary, as real conditions and users' behavior is hard to predict. In particular, we are not making the artificial assumption that π is the uniform distribution.

Unlinkability - Sketch of the Proof. The security of the MRI protocol against linking follows from similar considerations as in the proof of Proposition 2. However, now within the game we take into account all interactions, each game concerns choice of participants as well as domain used, and the adversary is given all transcripts.

Before we proceed let us introduce the following concept. For a pair of participants A and B holding the public keys $y_A = g_S^{x_A}$, $y_B = g_S^{x_B}$ for a domain S we define their *hidden public key* as $g_S^{x_A x_B}$.

Proposition 3. *Given the hidden public key $g_S^{x_A x_B}$ for A and B and domain S, one can generate transcripts of an interaction between A and B within S with exactly the same probability distribution as for the real interactions.*

Proof. The fake transcripts are created by following exactly the operations of A and B from the description of the protocol. The only exception is computing the key K (as neither x_A nor x_B is available). However, one can compute K using the equality $K = (g_S^{x_A x_B})^{h_A h_B}$. □

Obviously, ability of the adversary to distinguish between distributions π and $\pi|D$ from Definition 5 can only increase, if for each pair of participants A and B and each domain the adversary learns the hidden public key $g_S^{x_A x_B}$. From now on we assume that the adversary knows the hidden public key for each pair of participants and domain.

Assume that the adversary applies algorithm $\mathcal{A}$ to break privacy. The overall strategy to show that advantage of $\mathcal{A}$ is negligible is as follows:

We consider behavior $\mathcal{A}$ separately for different *cases*. A *case* is determined by fixing the value of R. (Note that according to our assumptions, the probabilities of cases may differ.) However, if we succeed to show that in each case we can replace the transcripts by random transcripts with a negligible change of behavior of $\mathcal{A}$, then $\mathcal{A}$ may skip the input regardless of the case.

Now consider a case C, and assume that the last interaction T_k is between the participants A and B. Then we consider two kinds of inputs to $\mathcal{A}$: the original transcripts and the transcripts with the last interaction T_k replaced by four random messages. As in the proof of Proposition 2 we show that the behavior of $\mathcal{A}$ cannot differ non-negligibly for these two kinds of inputs. Assume conversely that $\mathcal{A}$ behaves in a different way. Then we use it to build a distinguisher between the random transcripts and the transcripts between participants A and B. Indeed, given a transaction T which is either random or between A and B, we build a case for $\mathcal{A}$, by adding transcripts $T_1, \ldots, T_{k-1}$ where the participants of the interactions are indicated by C. Creating the transcripts is possible due to Proposition 3.

We proceed in the same way, in each phase we replace the next T_i by random transcripts and we argue that the behavior of $\mathcal{A}$ cannot change in a non-negligible way. Finally we are left with random transcripts, but $\mathcal{A}$ behaves almost in the same way as for the original inputs for the case C.

Finally notice that after these transformations $\mathcal{A}$ works on the same sets of random inputs with the same probability distribution. Hence $\mathcal{A}$ may skip the actual input and generate random transcripts by itself. It follows directly that $\mathcal{A}$ does not distinguish between π and $\pi|D$. Thereby we get the following result:

Theorem 1. *Assuming the Ideal Cipher Model and hardness of the Decisional Diffie-Hellman Problem, the MRI protocol is secure against linking.*

5 AKE Security of the MRI Protocol

For security of the session key we follow the model originating from [18] and extended by many authors. The model is based on the principle that if one of the legitimate participants (not necessarily both!) enters an *accepting state* with a session key K_s, then it should be infeasible for the adversary to derive K_s. In an accepting state a participant A not only holds the session key but also the identifier of the accepted session and the identity of the other participant B with whom A believes to share K_s.

The adversary $\mathcal{A}$ fully controls the communication channel between any participants A and B. This means that if a message is sent from A to B (or conversely), then $\mathcal{A}$ may prevent the delivery, may modify the message, or deliver a message of its choice. Moreover, $\mathcal{A}$ may deliver a message when no message is sent.

Security Game. We confine ourselves to the case when there are participants A and B holding private keys x_A, x_B. $\mathcal{A}$ controls all other users and holds their private and public keys. $\mathcal{A}$ may obtain the ephemeral keys used by A and B except for the session attacked. The attack consists of the following phases:

Phase 1: a number of times the protocol is executed between A and B as well as between A or B and the participants controlled by $\mathcal{A}$. For each of these interactions $\mathcal{A}$ may demand revealing the ephemeral values.

Phase 2: A and B execute the protocol. $\mathcal{A}$ can manipulate any message transmitted, but cannot ask for ephemeral values.

Phase 3: If neither A nor B enters an accepting state, then $\mathcal{A}$ looses. If A (respectively, B), terminates in an accepting state, it chooses a bit b at random. Then $\mathcal{A}$ obtains either the session key K_s kept by A (if $b = 0$), or a random key R (if $b = 1$).

Phase 4: it is executed exactly as Phase 1.

Finally, $\mathcal{A}$ answers $\overline{b}$ and wins, if $b = \overline{b}$.

Note that inability to distinguish between the session key and a random key witnesses that no substantial property of the session key can be deduced by $\mathcal{A}$.

5.1 Security Proof

We gradually simplify the attack scenario without substantial changes of adversary's advantage. The initial attack game is described in Sect. 5. The core property of authentication is presented by the following lemma:

Lemma 1. *Assume that CDH Problem is hard. Let y be a element such that discrete logarithm of y with respect to g is unknown. Let c be chosen at random. Then it is infeasible to provide an element c' and (K, a) such that K is a solution for CDH Problem for c and c' and simultaneously $c' = y^a$.*

Proof. Assume conversely that it is possible to present such (K, a). Then we show that it would be able to solve CDH Problem. Given an instance (u, v) of CDH choose r at random and set $y := v^r$. Then choose r' at random and set $c := u^{r'}$. In this way we derive a random instance of the problem concerned in Lemma 1. According to the current assumption derive c' and (K, a). So $K = \text{CDH}(u^{r'}, c')$, where $\text{CDH}(\alpha, \beta)$ stands for the solution of CDH Problem for α and β. However, $c' = y^a$ so $K = \text{CDH}(u^{r'}, y^a) = \text{CDH}(u^{r'}, v^{ra}) = \text{CDH}(u, v)^{r'ra}$. Since we know r, r' and a, we can get $\text{CDH}(u, v)$. □

Corollary 1. *Under the same assumptions as in Lemma 1 it is infeasible to create $E_{K_A}(a, cert_A)$ where $K_A{=}H(K|3)$, $K{=}CDH(c, c')$, and $c'{=}y_A^{H(a)}$.*

Proof. According to Ideal Cipher Model, creating the correct ciphertext is possible only if K_A and a are given. According to the correlated-input secure hash, deriving K_A with a non-negligible probability requires using K. So, getting $E_{K_A}(a, cert_A)$ yields (K, a), which is infeasible by Lemma 1. □

Reducing Phases 1 and 4. One can eliminate all correct interactions between either A or B and a participant controlled by $\mathcal{A}$ from Phases 1 and 4. Indeed, according to Proposition 1 $\mathcal{A}$ can generate transcripts of these interactions with exactly the same probability distribution. The next step is to reveal to the adversary $g^{x_A x_B}$ as it can only increase the advantage of $\mathcal{A}$. However, by Proposition 3 this enables to generate transcripts of correct interactions between A and B with exactly the same probability. Thereby, during Phases 1 and 4 only interactions corrupted by the adversary are left.

Now, let us consider an interaction between B (or A) and D (run by $\mathcal{A}$) in Phase 1 or 4, and initiated by D. As D deviates from the protocol, authentication of D fails and B (or A) sends no second message. So the only message sent by the honest party is the random element c_B, and this can be easily simulated.

The case of an interaction initiated by an honest user, say A, with D controlled by $\mathcal{A}$, is more complicated. There are two subcases: the first is that D can solve CDH Problem for c_A and c_D. This case can be perfectly simulated by $\mathcal{A}$: it chooses a, and proceeds as described by the protocol apart from derivation of K which is done according to the subcase assumption. In the other case, the adversary becomes the ciphertext $E_1 = Enc_{K_A}(a, cert_A)$. However, since $\mathcal{A}$ cannot derive K, we can replace K_A by a random key using correlated-input secure hash assumption. Then, according to the Ideal Cipher Model we can replace the ciphertext E_1 with a random string of the same length.

Attacking Interactions between A and B. The only interactions in Phases 1 and 4 that are left are interactions between A and B corrupted by $\mathcal{A}$. As $\mathcal{A}$ controls the communication channel, we may assume that the following messages are exchanged (the elements with an overline come from $\mathcal{A}$):

- between A and $\mathcal{A}$: c_A, $\overline{c_B}$, E_1, $\overline{E_2}$
- between $\mathcal{A}$ and B: $\overline{c_A}$, c_B, $\overline{E_1}$, E_2

We consider a number of cases depending on the behavior of $\mathcal{A}$.

Case 1: $\mathcal{A}$ cannot derive $\text{CDH}(c_A, \overline{c_B})$.
In this case $\mathcal{A}$ gets a ciphertext E_1 obtained with an unknown key K_A. According to the Ideal Cipher Model assumption, $\mathcal{A}$ cannot get any information about the plaintext or transform it a controlled way. So essentially the adversary may either use $\overline{E_1} = E_1$ or to ignore E_1 when constructing $\overline{E_1}$. In the first case B will accept it provided that $c_A = \overline{c_A}$ and $\text{CDH}(c_A, \overline{c_B}) = \text{CDH}(\overline{c_A}, c_B)$, that is when $c_A = \overline{c_A}$ and $c_B = \overline{c_B}$. In the second case B will not accept $\overline{E_1}$ with a high probability. So we have two cases:

- up to the third step, the execution of the protocol is not disturbed by the adversary,
- there are some modifications by the adversary, but B rejects after getting $\overline{E_1}$ and the ciphertext E_1 can be replaced by a random string.

Consequently, performing the last step (delivery of $\overline{E_2}$) can be done either according to the protocol or simulated by $\mathcal{A}$ (as B is silent).

If the whole protocol is executed without modifications of $\mathcal{A}$, then it can be eliminated from Phases 1 and 4, as already observed. So in all cases we can eliminate such interactions from Phase 1 and 4.

Case 2: $\mathcal{A}$ can derive $\text{CDH}(c_A, \overline{c_B})$.
It means in particular that $c_B \neq \overline{c_B}$. In this case the answer E_1 from A can be simulated by $\mathcal{A}$, as c_A can be generated by an oracle as y_A^a. Consequently, by Lemma 1 the adversary $\mathcal{A}$ cannot create $\overline{E_1}$ that is accepted by B.

Nevertheless, $\mathcal{A}$ can continue interacting with A. However, in this case providing $\overline{E_2}$ and accepting it by A occurs with a negligible probability only. Indeed, the only input from B is a random element c_B which can be simulated.

We conclude that it is possible to simulate the interaction in this case and that neither B nor A enters an accepting state.

Phase 2. We are left with a game consisting of Phases 2 and 3. First we consider the case that B enters an accepting state. This means that E_1 corresponds to c_A received by B and c_B sent by B. According to the Ideal Cipher Model this may occur with a non-negligible probability only if E_1 has been created with the key K_A as computed by B. Indeed, the plaintext contains $cert_A$, which is fixed, so a different key for the same ciphertexts would lead to a plaintext not containing $cert_A$. (Also a can be checked against y_A and c_A.)

The presence of the correct a witnesses that E_1 originates from a party that used the same c_A as received by B. On the other hand, to get K_A it is necessary to use K, apart from a negligible probability. In turn, deriving $K = c_B^{x_A h_A}$ for known (but random) c_B, known h_A, and y_A, but without x_A is equivalent to solving CDH Problem for c_B and y_A. As we assume that the DDH Problem is hard, this is infeasible. So B can assume that E_1 have been created by a party holding the key x_A, that is by A. It means that c_A originates from A.

Now, let us argue why adversary $\mathcal{A}$ cannot distinguish between the right session key and a random key. Note that all messages sent by A and B correspond to a correct protocol execution (maybe the last message from B to A is not delivered correctly). Then we may reveal the values of a, b, K_A, K_B, and refer to correlated-input secure hash function condition, where the random parameters used by the circuits are x_A, x_B.

The same argument can be applied to cover the case that A enters an accepting state.

6 Leaking Ephemeral Keys

Ephemeral values may be implemented in a less secure way than long-time secret keys. Therefore it is necessary to consider consequences of revealing them. In particular, the attack may be performed against A that interacts with B which is controlled by an adversary. We draft here two cases:

Attempt to Learn x_A or x_B. We concern the extreme case that the adversary is getting a, b as well as the private key x_B and attempts to learn x_A. However, in this case the messages exchanged between A and B can be perfectly simulated according to Sect. 4. So any attack executed in this way can be performed off-line with the same effect. In turn, the off-line attack can be used as an attack against the Discrete Logarithm Problem: we choose at random the values a, b, x_B, derive a protocol description and run the off-line adversary on these data.

Attempt to Learn a Session Key. Assume that we are given a transcript of an interaction consisting of c_A, c_B, E_1, E_2 and a and b used for this interaction. Ability to learn anything on the session key is described by the following game:

Game 0.

```
choose $a$, $b$ at random, $h_A := H(a|0)$, $h_B := H(b|0)$, $c_A := y_A^{h_A}$, $c_B := y_B^{h_B}$
$K := g^{x_A x_B h_B h_A}$,   $K_A := H(K|1)$, $K_B := H(K|2)$
$E_1 := Enc_{K_A}(a, cert_A)$, $E_2 := Enc_{K_B}(b, cert_B)$
choose $u$ at random
if $u = 0$, then $R := H(K|3)$, otherwise choose $R$ at random
$u' := \mathcal{A}(a, b, E_1, E_2, R, y_A, y_B)$
```

The adversary wins, if $u' = u$.

Below we consider a modified version of this game, where E_1 and E_2 are generated in a different way.

Game 1.

```
choose $a$, $b$ at random, $h_A := H(a|0)$, $h_B := H(b|0)$, $c_A := y_A^{h_A}$, $c_B := y_B^{h_B}$
$K := g^{x_A x_B h_B h_A}$,   $K_A := H(K|1)$, $K_B := H(K|2)$
choose $E_1$ and $E_2$ at random
choose $u$ at random
if $u = 0$, then $R := H(K|3)$, otherwise choose $R$ at random
$u' := \mathcal{A}(a, b, E_1, E_2, R, y_A, y_B)$
```

A difference between Game 0 and Game 1 may be observed only if $\mathcal{A}$ asks the encryption oracle $\mathcal{O}$ a query containing K_A or K_B. Then decrypting E_1 or E_2 may yield wrong results (in the Game 1, $\mathcal{O}$ does not know a and b, so with a high probability it will choose the plaintext inconsistently). However, if $\mathcal{A}$ may generate K_A or K_B

with a non-negligible probability, then we can construct a distinguisher for the DDH Problem, just as in the proof of Proposition 2.

Now let us clean up by eliminating parameters unused by the adversary or random. Thereby we get the following game:

Game 2.

choose a, b at random, $h_A := H(a|0)$, $h_B := H(b|0)$, $K := g^{x_A x_B h_B h_A}$
choose u at random
if $u = 0$, then $R := H(K|3)$, otherwise choose R at random
$u' := \mathcal{A}(a, b, R, y_A, y_B)$

Now, it is easy to see that Game 2 could be directly used for solving the DDH Problem: given a candidate triple (U, V, Z), we choose a, b, r_1, r_2 at random, put $y_A := U^{r_1}$, $y_B := V^{r_2}$ and $R := H(Z^{r_1 r_2 h_A h_B}|3)$. Then we give a, b, R, y_A, y_B to $\mathcal{A}$. (Note that r_1, r_2 are used to randomize the input.)

Note that if Z is random, then R created as above is not a random value, but a hash value of a random value. However, any difference in behavior of $\mathcal{A}$ in case of random R and $R := H(Z^{r_1 r_2 h_A h_B}|3)$ for a random Z would lead to a procedure that distinguishes the values of the form $H(S|3)$ from the random strings of the same length. For correlated-input secure hash functions this is impossible.

The above argument can be extended to the case when we have a number of interactions between A and B and the corresponding ephemeral keys. In this case we formulate the following game for k interactions:

Game 0'.

choose a_i, b_i at random, $h_{i,A} := H(a_i|0)$, $h_{i,B} := H(b_i|0)$, for $i \leq k$,
$c_{i,A} := y_A^{h_{i,A}}$, $c_{i,B} := y_B^{h_{i,B}}$, for $i \leq k$,
$K_i := g^{x_A x_B h_{i,B} h_{i,A}}$, for $i \leq k$,
$K_{i,A} := H(K_i|1)$, $K_{i,B} := H(K_i|2)$, for $i \leq k$,
$E_{i,1} := \mathit{Enc}_{K_{i,A}}(a_i, \mathit{cert}_A)$, $E_{i,2} := \mathit{Enc}_{K_{i,B}}(b_i, \mathit{cert}_B)$, for $i \leq k$,
choose u at random, choose S at random
if $u = 0$, then $R_i := H(K_i|3)$, otherwise $R_i := H(S^{h_{i,B} h_{i,A}}|3)$, for $i \leq k$
$u' := \mathcal{A}(a_1, \ldots, a_k, b_1, \ldots, b_k, E_{1,1} \ldots, E_{k,1}, E_{1,2} \ldots, E_{k,2}, R_1, \ldots, R_k)$

After making essentially the same transformations we get a proof for the following theorem:

Theorem 2. *Assume that H is a correlated-input secure hash function and that the DDH Problem is hard. Then, in the Ideal Cipher Model it is infeasible to derive any information on the session keys of MRI given the messages exchanged and the ephemeral keys a, b used for these interactions.*

7 Forward Security

Another problem we have to concern is that at some moment the private key x_A is disclosed. This may occur due to physical attack with techniques unknown at the time of the system deployment. In this scenario the adversary has no access to the ephemeral keys – as they should be stored in a volatile memory or erased after usage. So the attack scenario can be described by the following game:

Game 0.

```
choose $a$, $b$ at random, $h_A := H(a|0)$, $h_B := H(b|0)$
$c_A := y_A^{h_A}$, $c_B := y_B^{h_B}$
$K := g^{x_A x_B h_B h_A}$, $K_A := H(K|1)$, $K_B := H(K|2)$
$E_1 := \mathit{Enc}_{K_A}(a, \mathit{cert}_A)$, $E_2 := \mathit{Enc}_{K_B}(b, \mathit{cert}_B)$
choose $u$ at random
if $u = 0$, then $R := H(K|3)$, otherwise choose $R$ at random
$u' := \mathcal{A}(x_A, x_B, E_1, E_2, R)$
```

The adversary wins if $u' = u$. Following almost exactly the same argument as in the proof of Theorem 2 we get the following result (in fact, the results holds also under assumption of semantic security):

Theorem 3. *Assume that H is a correlated-input secure hash function and that Decisional DDH Problem is hard. Then in the Ideal Cipher Model it is infeasible to derive any information on the session key of MRI executed between A and B, given the messages exchanged and the private keys x_A, x_B.*

7.1 Malicious Implementations

If a protocol is implemented in a black box device, then a user is exposed to malicious implementations that behave like the original protocol – no procedure based on the regular output may detect any difference – but a party holding appropriate secret (not stored in the device) gets access to private data of the user (see *kleptographic attacks*, e.g. [19]). The key mechanism of kleptographic attacks is to use a pseudorandom parameter that can be derived by the device (from its internal values) and the attacker (from the previous output of the device and the secret of the attacker). As an authentication protocol cannot be deterministic it seems that there is always room for such an attack.

Let us discuss shortly susceptibility of the MRI protocol to such attacks. As the long time secrets x_A are used for exponentiations only, x_A can be implemented in ROM with no access to other operations. In particular, for ROM it is impossible to manipulate the code. The code for the remaining parts of MRI may be included e.g. in a smart card applet, where manipulations are much easier. Nevertheless, at worst the applet may serve as an oracle for computing values d^{x_A}, where the numbers d are given. This may slightly ease a cryptanalytic attack against x_A, but not expose x_A directly.

The other target of the adversary is to derive a session key. Note that leaking the ephemeral value a (or h_A) without x_A does not enable to derive a session key: given c_B and h_A we still need x_A to obtain $K = c_B^{h_A x_A}$. So the leakage must be more complicated than just based on malicious way of computing h_A.

Finally, we have to be aware that MRI, like any other protocol with pseudorandom values, enables a limited hidden channel. Simply, in order to leak a short bit string $\kappa = k_0 k_1 \dots k_m$ we leak a few bits in each c_A. Namely, the malicious implementation chooses a until $H(Y^{h_A})$ has $k_0 \dots k_m$ as leading bits. If $Y = g^z$ and z is held by the adversary, then κ can be recomputed from $H(c_A^z)$. On average, 2^m trials are necessary, so m cannot be large, especially for smart cards.

References

1. BSI: Advanced Security Mechanisms for Machine Readable Travel Documents 2.1, parts 1-3. Technische Richtlinie TR-03110-1 (2012)
2. Canetti, R., Krawczyk, H.: Analysis of key-exchange protocols and their use for building secure channels. In: Pfitzmann, B. (ed.) EUROCRYPT 2001. LNCS, vol. 2045, pp. 453–474. Springer, Heidelberg (2001)
3. Krawczyk, H.: HMQV: A high-performance secure Diffie-Hellman protocol. Cryptology ePrint Archive, Report 2005/176 (2005)
4. Law, L., Menezes, A., Qu, M., Solinas, J., Vanstone, S.: An efficient protocol for authenticated key agreement. Designs, Codes and Cryptography 28(2), 119–134 (2003)
5. LaMacchia, B.A., Lauter, K., Mityagin, A.: Stronger security of authenticated key exchange. In: Susilo, W., Liu, J.K., Mu, Y. (eds.) ProvSec 2007. LNCS, vol. 4784, pp. 1–16. Springer, Heidelberg (2007)
6. Lee, J., Park, J.H.: Authenticated key exchange secure under the computational Diffie-Hellman assumption. Cryptology ePrint Archive, Report 2008/344 (2008)
7. Ustaoglu, B.: Obtaining a secure and efficient key agreement protocol from (H)MQV and NAXOS. Cryptology ePrint Archive, Report 2007/123 (2007)
8. Lauter, K., Mityagin, A.: Security analysis of KEA authenticated key exchange protocol. In: Yung, M., Dodis, Y., Kiayias, A., Malkin, T. (eds.) PKC 2006. LNCS, vol. 3958, pp. 378–394. Springer, Heidelberg (2006)
9. Aiello, W., Bellovin, S.M., Blaze, M., Canetti, R., Ioannidis, J., Keromytis, A.D., Reingold, O.: Just fast keying: Key agreement in a hostile internet. ACM Trans. Inf. Syst. Secur. 7(2), 242–273 (2004)
10. Cheng, Z., Chen, L., Comley, R., Tang, Q.: Identity-based key agreement with unilateral identity privacy using pairings. In: Chen, K., Deng, R., Lai, X., Zhou, J. (eds.) ISPEC 2006. LNCS, vol. 3903, pp. 202–213. Springer, Heidelberg (2006)
11. Chien, H.-Y.: ID-based key agreement with anonymity for ad hoc networks. In: Kuo, T.-W., Sha, E., Guo, M., Yang, L.T., Shao, Z. (eds.) EUC 2007. LNCS, vol. 4808, pp. 333–345. Springer, Heidelberg (2007)
12. Krawczyk, H.: SIGMA: The 'SIGn-and-MAc' approach to authenticated Diffie-Hellman and its use in the IKE-protocols. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 400–425. Springer, Heidelberg (2003)
13. Bender, J., Dagdelen, Ö., Fischlin, M., Kügler, D.: The PACE|AA protocol for machine readable travel documents, and its security. In: Keromytis, A.D. (ed.) FC 2012. LNCS, vol. 7397, pp. 344–358. Springer, Heidelberg (2012)
14. Raimondo, M.D., Gennaro, R., Krawczyk, H.: Deniable authentication and key exchange. In: Juels, A., Wright, R.N., di Vimercati, S.D.C. (eds.) ACM Conference on Computer and Communications Security, pp. 400–409. ACM (2006)
15. Hanzlik, L., Kluczniak, K., Kubiak, P., Kutyłowski, M.: Restricted identification without group keys. In: Min, G., Wu, Y., Liu, L.C., Jin, X., Jarvis, S.A., Al-Dubai, A.Y. (eds.) TrustCom, pp. 1194–1199. IEEE Computer Society (2012)
16. Goyal, V., O'Neill, A., Rao, V.: Correlated-input secure hash functions. In: Ishai, Y. (ed.) TCC 2011. LNCS, vol. 6597, pp. 182–200. Springer, Heidelberg (2011)
17. Bender, J., Dagdelen, Ö., Fischlin, M., Kügler, D.: Domain-specific pseudonymous signatures for the German identity card. In: Gollmann, D., Freiling, F.C. (eds.) ISC 2012. LNCS, vol. 7483, pp. 104–119. Springer, Heidelberg (2012)
18. Bellare, M., Rogaway, P.: Entity authentication and key distribution. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 232–249. Springer, Heidelberg (1994)
19. Young, A., Yung, M.: The dark side of "black-box" cryptography, or: Should we trust capstone? In: Koblitz, N. (ed.) CRYPTO 1996. LNCS, vol. 1109, pp. 89–103. Springer, Heidelberg (1996)