Open access • Book Chapter • DOI:10.1007/11779360_21

RSA-Based secret handshakes — Source link

Damien Vergnaud

Institutions: University of Caen Lower Normandy

Published on: 14 Mar 2005

Topics: RSA problem, Key (cryptography), Random oracle and Handshake

RSA-Based Secret Handshakes

Damien Vergnaud

RSA-Based Secret Handshakes

Damien Vergnaud

Laboratoire de Mathématiques Nicolas Oresme
Université de Caen, Campus II, B.P. 5186,
14032 Caen Cedex, France
Damien.Vergnaud@math.unicaen.fr

Abstract. A secret handshake mechanism allows two entities, members of a same group, to authenticate each other secretly. This primitive was introduced recently by Balfanz, Durfee, Shankar, Smetters, Staddon and Wong and, so far, all the schemes proposed are based on discrete log systems. This paper proposes three new secret handshake protocols secure against active impersonator and detector adversaries. Inspired by two RSA-based key agreement protocols introduced by Okamoto and Tanaka in 1989 and Girault in 1991, our schemes are, in the random oracle model, provably secure against active adversaries under the assumption that the RSA problem is intractable.

1 Introduction

The concept of *secret handshakes* was introduced in 2003 [1] by Balfanz, Durfee, Shankar, Smetters, Staddon and Wong. The present paper focuses on the proposal of three new practical constructions of secret handshake protocol and the security treatment of them: the schemes are secure against active adversaries assuming the hardness of the RSA problem (and one of its weaker variants for one scheme). These are the first constructs which can be instantiated with the RSA primitive (and thus give a first step towards a problem raised in [5] by Castelluccia, Jarecki and Tsudik)

Background. A secret handshake scheme is a cryptographic primitive recently introduced by Balfanz *et al.* [1] which allows two members of the same group to identify each other secretly, in the sense that each party reveals his/her affiliation to the latter only if the other party is a member of the same group. These protocols model *in silico* the secret handshake in the folklore of the *in vivo* exclusive societies, or guilds.

The protocol proposed in [1] is a simple adaptation of the non-interactive key-agreement scheme of Sakai, Ohgishi and Kasahara [19]. It uses one-time credentials to insure that instances of the handshake protocol performed by the same party cannot be linked. In [22], Xu and Yung proposed secret handshake schemes that achieve *unlinkability* with reusable credentials. However, their schemes offer weaker anonymity to the users who furthermore must be aware of the information of other groups. Recently, Castelluccia *et al.* [5] showed how to build secret

handshake protocols using a novel tool they called *CA-oblivious public-key encryption* which is an encryption scheme such that neither the public key, nor the ciphertext reveal any information about the Certification Authority. Their schemes are secure under a standard cryptographic assumption: the hardness of the classical computational Diffie-Hellman problem. However, they also rely on one-time credentials to reach the unlinkability property. In the first announcement of their results, Castelluccia *et al.* stated that they know how to get CA oblivious encryption scheme based on RSA, but the claim was incorrect and, so far, there do not exist protocols based on RSA which offer full anonymity. In [5], the authors explicitly stated as an open problem the construction of a secret handshake protocol based on RSA. The investigation of this issue is the main purpose of the present paper.

Underlying Technique. As mentioned above, in Balfanz *et al.*'s scheme, completing the secret handshake is essentially equivalent to computing a key in Sakai *et al.*'s non-interactive key agreement protocol, that is particular to the two interacting group members. The basic framework of our constructions builds on very similar ideas from *identity-based* and *self-certified* key agreements protocols. In 1984, Shamir [20] introduced the concept of identity-based cryptosystems in which the public keys can be arbitrary bit strings and in particular can be defined as a part of the publicly known identification information. In 1991, Girault [9] refined the concept and introduced self-certified public keys which are computed by both the authority and the user, so that the certificate is "embedded" in the public key itself, and does not take the form of a separate value.

Our first scheme, that we call OT-SH, is based on an identity-based key agreement protocol proposed in 1989 by Okamoto and Tanaka [16] and whose security relies on the RSA problem [11]. The second and the third construction, called Gi+SH and Gi×SH, relies on the non-interactive self-certified key agreement scheme proposed by Girault in his seminal paper [9] whose security is also based on the difficulty of solving RSA [15]. Adding suitable nonces and hash functions in these schemes (along the lines drawn in [8]), we obtained secret handshake protocols whose security against active impersonator and detector adversaries relies, in the ROM, on the difficulty of solving the RSA problem (with the notable exception of the impersonator resistance of Gi+SH which relies on the hardness of the so-called Difference RSA problem [13]). The scheme OT-SH takes four rounds (as the one in [5] based on CA-oblivious encryption), whereas the schemes Gi-SH and Gi×SH take only three rounds (as the original proposal from [1], and the Diffie-Hellman-based scheme from [5]). In fact, these protocols are very similar to Balfanz *et al.*'s original proposal.

Finally, a protocol providing authentication without key exchange is susceptible to an enemy who waits until the authentication is complete and then takes over one end of the communications line. Therefore, we extend the new schemes such that at the end of the handshake, the parties can set up a temporary session key for securing further communication between them.

2 Preliminaries

2.1 Notations

The set of n-bit strings is denoted by $\{0,1\}^n$ and the set of all finite binary strings is denoted by $\{0,1\}^*$. Concatenation of two strings x_1 and x_2 is denoted by $x_1 \| x_2$ and the empty string is denoted ε.

φ denotes the Euler totient function and λ denotes the Carmichael reduced totient function. For any positive integer k, $\mathsf{Prime}(k)$ denotes the set of prime numbers in $[\![2^k, 2^{k+1}]\!]$ and $2\mathsf{Factors}(k)$ the set of integers n such that $n = pq$ with $p < q < 2p$ and $p, q \in \mathsf{Prime}(k)$. An element of $2\mathsf{Factors}(k)$ for any k is called an *RSA-modulus*.

Let $\mathcal{A}$ be a probabilistic Turing machine running in expected polynomial time (a PPTM, for short), and let x be an input for $\mathcal{A}$. The probability space that assigns to a string σ the probability that $\mathcal{A}$, on input x, outputs σ is denoted by $A(x)$. Given a probability space S, a PPTM that samples a random element according to S is denoted by $x \xleftarrow{R} S$. For a finite set X, $x \xleftarrow{R} X$ denotes a PPTM that samples an element uniformly at random from X.

A *two-party protocol* is a pair of interactive probabilistic Turing machines (A, B). An execution of the protocol (A, B) on input x for A and y for B is an alternating sequence of A-rounds and B-rounds, each producing a message m to be delivered to the other party. We denote by "$\rightsquigarrow m$" the transmission of m from one party to the other. The sequence of such message exchanges is called a *transcript* of this execution of the protocol. If, for all x and y, the length of the transcript, as well as the expected running time of A and B on inputs x and y respectively, are polynomial in the length of x and y, then (A, B) is a *polynomial time two-party protocol*.

2.2 Underlying problems

The security of asymmetric cryptographic tools relies on assumptions about the hardness of certain algorithmic problems. The best known public-key primitive is the RSA function. In order to highlight the fact that our schemes apply to any RSA key generator, we do not pin down any particular generator, but instead parameterize definitions and security results by a choice of generator.

Definition 1. *An* RSA-group generator *is a PPTM that takes a security parameter k as input and outputs a 6-tuple (n, p, q, e, d, g) where $n = pq \in 2\mathsf{Factors}(k)$, $e \in [\![3, 2^k]\!]$, $ed \equiv 1 \mod \varphi(n)$ and $g \in (\mathbb{Z}/n\mathbb{Z})^*$ of order $\lambda(n)$.*

The *RSA assumption* says, roughly speaking, that given a large RSA *modulus* n, an exponent e and α in $(\mathbb{Z}/n\mathbb{Z})^*$, it is hard to find x in $(\mathbb{Z}/n\mathbb{Z})^*$ such that $x^e = \alpha \bmod n$. The *difference RSA assumption* was introduced by Naor in [13]. This non-standard hypothesis deals with the hardness of finding two RSA preimages such that the difference of their images is a given quantity α. The adversary $\mathcal{A}$ has access to a sequence of $m-1$ pairs $(x_i, y_i) \in [(\mathbb{Z}/n\mathbb{Z})^*]^2$ such that

$x_i^e - y_i^e = \alpha \bmod n$ where $\mathcal{A}$ chooses y_i. It should find a new $(x_m, y_m) \in [(\mathbb{Z}/n\mathbb{Z})^*]^2$ such that $x_m^e - y_m^e = \alpha \bmod n$. We denote by $(\alpha + (\cdot)^e)^d$ the oracle that takes input $y \in (\mathbb{Z}/n\mathbb{Z})^*$ and returns $(\alpha + y^e)^d \bmod n$. An adversary solving the difference RSA problem is given oracle access to $(\alpha + (\cdot)^e)^d$.

Definition 2. *Let* Gen *be an RSA-group generator and let* $\mathcal{A}$ *be a PPTM. We consider the following random experiments, where* $k \in \mathbb{N}$ *is a security parameter:*

Experiment $\mathbf{Exp}^{\mathsf{rsa}}_{\mathsf{Gen},\mathcal{A}}(k)$	*Experiment* $\mathbf{Exp}^{\mathsf{diff-rsa}}_{\mathsf{Gen},\mathcal{A}}(k)$
$(n, p, q, e, d, g) \xleftarrow{R} \mathsf{Gen}(k)$	$(n, p, q, e, d, g) \xleftarrow{R} \mathsf{Gen}(k)$
$\alpha \xleftarrow{R} (\mathbb{Z}/n\mathbb{Z})^*$	$\alpha \xleftarrow{R} (\mathbb{Z}/n\mathbb{Z})^*$
$x \xleftarrow{R} \mathcal{A}(n, e, \alpha)$	$(x, y) \xleftarrow{R} \mathcal{A}^{(\alpha+(\cdot)^e)^d}(n, e, y)$
Return 1 *if* $x^e = \alpha \bmod n$, 0 *otherwise*	*Return* 1 *if the following hold and* 0 *otherwise* (1) $x^e - y^e = \alpha \bmod n$ (2) y *was not queried to* $(\alpha + (\cdot)^e)^d$

We define the success *of* $\mathcal{A}$ via $\mathbf{Succ}^{\mathsf{rsa}}_{\mathsf{Gen},\mathcal{A}}(k) = \Pr[\mathbf{Exp}^{\mathsf{rsa}}_{\mathsf{Gen},\mathcal{A}}(k) = 1]$. *(*resp. $\mathbf{Succ}^{\mathsf{diff-rsa}}_{\mathsf{Gen},\mathcal{A}}(k) = \Pr[\mathbf{Exp}^{\mathsf{diff-rsa}}_{\mathsf{Gen},\mathcal{A}}(k) = 1]$*).*

Let $q, \tau \in \mathbb{N}^{\mathbb{N}}$. $\mathcal{A}$ *is a a* τ-RSA-adversary *if for all positive integer* k*, the experiment* $\mathbf{Exp}^{\mathsf{diff-rsa}}_{\mathsf{Gen},\mathcal{A}}(k)$ *ends in expected time less than* $\tau(k)$. $\mathcal{A}$ *is a* (q, τ)-DIFF-RSA-adversary *if for all positive integer* k*, the experiment* $\mathbf{Exp}^{\mathsf{rsa}}_{\mathsf{Gen},\mathcal{A}}(k)$ *ends in expected time less than* $\tau(k)$ *and in this experiment* $\mathcal{A}$ *makes at most* $q(k)$ *queries to the oracle* $(\alpha + (\cdot)^e)^d$.

Let $\varepsilon \in [0, 1]^{\mathbb{N}}$. Gen *is said to be* (τ, ε)-RSA-secure *if for any* τ*-RSA-adversary* $\mathcal{A}$ *and any positive integer* k, $\mathbf{Succ}^{\mathsf{rsa}}_{\mathsf{Gen},\mathcal{A}}(k)$ *is smaller than* $\varepsilon(k)$. Gen *is said to be* (q, τ, ε)-DIFF-RSA-secure *if for any* (q, τ)*-DIFF-RSA-adversary* $\mathcal{A}$ *and any positive integer* k, $\mathbf{Succ}^{\mathsf{diff-rsa}}_{\mathsf{Gen},\mathcal{A}}(k)$ *is smaller than* $\varepsilon(k)$.

Finally, the weaker *composite Diffie-Hellman assumption*, related with those described above, is defined as follows:

Definition 3. *Let* Gen *be an RSA-group generator and let* $\mathcal{A}$ *be a PPTM. We consider the following random experiments, where* $k \in \mathbb{N}$ *is a security parameter:*

Experiment $\mathbf{Exp}^{\mathsf{comp-cdh}}_{\mathsf{Gen},\mathcal{A}}(k)$

$(n, p, q, e, d, g) \xleftarrow{R} \mathsf{Gen}(k)$

$(x, y) \xleftarrow{R} (\llbracket 1, 2^{2k+1} \rrbracket)^2$; $\alpha \leftarrow g^x$; $\beta \leftarrow g^y$

$\gamma \xleftarrow{R} \mathcal{A}(n, g, \alpha, \beta)$

Return 1 *if* $\gamma = g^{xy}$, 0 *otherwise*

Let $\tau \in \mathbb{N}^{\mathbb{N}}$. $\mathcal{A}$ *is a* τ-COMP-CDH-adversary *if for all positive integer* k*, the experiment* $\mathbf{Exp}^{\mathsf{comp-cdh}}_{\mathsf{Gen},\mathcal{A}}(k)$ *ends in expected time less than* $\tau(k)$*. We define the* success *of* $\mathcal{A}$ via $\mathbf{Succ}^{\mathsf{comp-cdh}}_{\mathsf{Gen},\mathcal{A}}(k) = \Pr[\mathbf{Exp}^{\mathsf{comp-cdh}}_{\mathsf{Gen},\mathcal{A}}(k) = 1]$. *Let* $\tau \in \mathbb{N}^{\mathbb{N}}$ *and* $\varepsilon \in [0, 1]^{\mathbb{N}}$. Gen *is said to be* (τ, ε)-COMP-CDH-secure *if for any* τ*-COMP-CDH-adversary* $\mathcal{A}$ *and any positive integer* k, $\mathbf{Succ}^{\mathsf{comp-cdh}}_{\mathsf{Gen},\mathcal{A}}(k)$ *is smaller than* $\varepsilon(k)$.

2.3 Ensuring that Handshakes do not reveal the group

A simple observation that seems to be folklore is that standard RSA does not provide anonymity, even if all *moduli* in the system have the same length. One approach to anonymizing RSA, suggested by Desmedt [7], is to add random multiples of the *modulus* n to the ciphertext. This padding removes any information about the size of n and does not interfere with the reduction of the value modulo n. In the following, we assume that such a technique is adopted and that the adversary gains no information on the RSA *modulus* involved in some protocol (in a statistical sense) from the encoding used in the transcript.

2.4 Proofs of knowledge of a discrete logarithm

In the design of his key agreement protocol, Girault [9] needs a zero-knowledge proof of knowledge of a discrete logarithm in a group of unknown order.

To achieve our security reductions, the executions of the protocol have to be simulated in the ROM. Therefore, in the design of our scheme, we rely on a procedure allowing to prove in a non-transferable way the knowledge of a discrete logarithm without revealing information on their value. We make use of non-interactive designated verifier zero-knowledge proof of knowledge of the discrete logarithm of y in base g (of unknown order). The notation is $\mathrm{DVPK}(a : y = g^a)$. We refer the reader to [9, 10] for further details.

3 Definition of Secret Handshakes

3.1 Syntactic definition

Roughly speaking, in a secret handshake scheme, there are group authorities having a public key and a matching secret key (CreateGroup). They can provide any user with a pair of keys based on its identity (AddMember) and they manage a certificate revocation list for the group (RemoveMember). The users can then identify themselves in a protocol (Handshake) in which the parties involved begin by knowing only the claimed identities and their own secret key provided by their authority (not necessarily the same). The formal definition of secret handshake schemes proposed by Castelluccia *et al.* and Xu and Yung in [5, 22] is the following:

Definition 4 ([5, 22]). *A* secret handshake protocol *SH is a 5-tuple* (*Setup*, *CreateGroup*, *AddMember*, *RemoveMember*, *Handshake*) *such that*

- *SH.Setup is a PPTM which takes an integer k as input. The outputs are* the public parameters. *k is called the security parameter.*
- *SH.CreateGroup is a PPTM which takes the public parameters as input and outputs a pair of group keys* $(\mathbf{pk_G}, \mathbf{sk_G})$. *It may also output a data structure CRL called a* certificate revocation list *which is originally empty.*
- *SH.AddMember is a polynomial time two-party protocol* (*Member*, *Group*) *where*

1. *SH.AddMember.Member takes the public parameters, a bit string ID and a group public key* $\mathbf{pk_G}$ *as inputs;*
2. *SH.AddMember.Group takes the public parameters, ID and the matching group secret key* $\mathbf{sk_G}$ *as inputs.*

SH.AddMember.Member outputs a pair of member keys $(\mathbf{pk}_{ID}, \mathbf{sk}_{ID})$ *associated with* $[\mathbf{pk_G}, ID]$.

– *SH.RemoveMember is a PPTM which takes the public parameters, a bit string ID, a group pair of keys* $(\mathbf{pk_G}, \mathbf{sk_G})$ *and the corresponding current CRL as inputs. It outputs an updated CRL which includes the newly revoked certificate ID.*

– *SH.Handshake is a polynomial time two-party protocol* $(Init, Match)$ *where*

1. *SH.Handshake.Init takes the public parameters, a pair of* (ID_I, ID_M)*, a group public key* $\mathbf{pk}_{G_I}$*,* $(\mathbf{pk}_{ID_I}, \mathbf{sk}_{ID_I})$ *a pair of member keys associated with* $[\mathbf{pk}_{G_I}, ID_I]$ *and a member public key* $\mathbf{pk}_{ID_M}$ *as inputs;*
2. *SH.Handshake.Match takes the public parameters, the pair* (ID_I, ID_M)*, a group public key* $\mathbf{pk}_{G_M}$ *and* $(\mathbf{pk}_{ID_M}, \mathbf{sk}_{ID_M})$ *a pair of member keys associated with* $[\mathbf{pk}_{G_M}, ID_M]$ *and* $\mathbf{pk}_{ID_I}$ *as inputs.*

The algorithms jointly output Accept if $\mathbf{pk}_{G_I} = \mathbf{pk}_{G_M} \wedge \{ID_I, ID_M\} \cap \mathrm{CRL} = \emptyset$ *and Reject otherwise.*

3.2 The random oracle debate

The proof of security for our schemes takes place in the random oracle model, introduced in [2]. In this model, cryptographic protocols are designed and proved secure under the additional assumption that publicly available functions, that are chosen truly at random, exist. These random oracles can only be accessed in a black-box way, by providing an input and obtaining the corresponding output. It has been pointed out that a proof of security in the ROM does not guarantee the existence of an instantiation of the random oracle under which the scheme is secure [4]. However, security proofs in the ROM remain "strong indicators" of the security of an analyzed protocol. Note that, to prove the security of the new schemes, there is no need to assume that all the involved hash functions act as random oracles. Indeed, it is possible to model one of them as a *non-programmable random oracle*. The NPROM is known to be strictly weaker than the ROM [14]. In view of the previous remark, it is a legitimate desire to construct protocols relying as little as possible on random oracles. Therefore, even if our security analysis is carried out in the ROM, we will nevertheless modelize some hash functions as (weaker) non-programmable random oracle.

3.3 Security requirements

This subsection recalls the security model formally defined in [5, 22]. Roughly speaking, a secret handshake protocol must satisfy the following properties:

1. completeness: if two members engage in the protocol SH.Handshake with valid pair of keys associated with the same group public key, then both parties output `Accept` at the end of the protocol;

2. impersonator resistance: given a group public key, it is computationally infeasible without the knowledge of some secret key associated with it to successfully execute the protocol SH.Handshake with a member of this group;
3. detector resistance: given a group public key $\mathbf{pk_G}$ and a member public key $\mathbf{pk_{ID}}$, it is computationally infeasible to determine whether $\mathbf{pk_{ID}}$ is associated with $\mathbf{pk_G}$ without the knowledge of a secret key associated with $\mathbf{pk_G}$;
4. indistinguishability to eavesdropper: given two member public keys associated with the same group public key, it is computationally infeasible to distinguish a successful handshake between these members from an unsuccessful one.

In this paper, we consider *active* adversaries against the impersonator resistance (IMP-ACT), the detector resistance (DET-ACT) and the indistinguishability (IND-ACT) of our schemes. The attacker is allowed to run the protocols several times: she can see all exchanged messages, can delete, alter, inject and redirect messages, can initiate communications with another party and can reuse messages from past communications. The adversary is able to trigger several executions of the protocol Handshake. She is also able to interleave these instances, asynchronously and concurrently, with executions with rightful members of the group. The attacker is also allowed to corrupt some users and to ask for additional member keys at any time, and she can do so adaptively.

As usual an adversary is formalized by a PPTM $\mathcal{A}$ that is allowed access to different oracles modeling the capacities mentioned above:

- $\mathcal{O}_{\mathrm{CG}}$: it activates a new group authority *via* algorithm SH.CreateGroup. We assume that a group authority is not under $\mathcal{A}$'s control before the new group is established.
- $\mathcal{O}_{\mathrm{AM}}$: given as input the identity of a group authority, it executes the protocol SH.AddMember. The algorithm admits an honest user and assigns it with a unique pseudonym ID.
- $\mathcal{O}_{\mathrm{HS}}$: given as inputs two pseudonyms ID_I and ID_M, it activates the protocol SH.Handshake between the corresponding members, where none, one or both of them may have been corrupt. A corrupt user will execute what the adversary is pleased of.
- $\mathcal{O}_{\mathrm{RM}}$: given as input the identity of a group authority and a bit string ID, it executes SH.RemoveMember to insert ID in the corresponding CRL.
- $\mathcal{O}_{\mathrm{Co}}$: given as input the identity of a group authority G and possibly a pseudonym ID associated with G, the oracle returns, to $\mathcal{A}$, the current internal state information (including all secrets) of G or ID's associated member. Once a group or a member is corrupt it will execute what $\mathcal{A}$ is pleased of, until such a corruption is detected. If the corruption of a member is detected, it is excluded from its group via the algorithm SH.RemoveMember. If the corruption of a group is detected, it is excluded from the system.

Resistance to impersonation attacks. This property captures the requirement that an adversary who does not belong to or does not corrupt a member of a group G managed by an uncorrupt group authority, has only a negligible probability of convincing an honest user of G that she is also a member of

G. We consider an adversary $\mathcal{A}$ in the ROM. She takes the public parameters as input, and outputs a triple $(G^\star, \mathsf{ID}^\star, d^\star)$. $\mathcal{A}$ has access to the random oracle(s) $\mathcal{O}_\mathcal{H}$ and to the oracles $\mathcal{O}_{\mathrm{CG}}$, $\mathcal{O}_{\mathrm{AM}}$, $\mathcal{O}_{\mathrm{HS}}$, $\mathcal{O}_{\mathrm{RM}}$, and $\mathcal{O}_{\mathrm{Co}}$. She succeeds if $\mathsf{ID}^\star$ belongs to $G^\star$; $G^\star$ remains uncorrupt during $\mathcal{A}$'s execution; all corrupt users from $G^\star$ are excluded from $G^\star$ (in particular $\mathsf{ID}^\star$ is uncorrupt) and if the protocol $\mathsf{SH.Handshake}(\mathcal{A}, \mathsf{ID}^\star)$ returns `Accept` when $d^\star = 0$ or if the protocol $\mathsf{SH.Handshake}(\mathsf{ID}^\star, \mathcal{A})$ returns `Accept` when $d^\star = 1$ (*i.e.* $\mathcal{A}$ initiates the final handshake if and only if $d^\star = 0$).

Definition 5. *Let SH be a secret handshake scheme and let $\mathcal{A}$ be a PPTM. We consider the following random experiment, where $k \in \mathbb{N}$ is a security parameter:*

Experiment $\mathbf{Exp}^{\mathsf{imp\text{-}act}}_{\mathsf{SH},\mathcal{A}}(k)$

$\mathcal{P} \xleftarrow{R} \mathsf{SH.Setup}(k)$

$(G^\star, \mathsf{ID}^\star, d^\star) \leftarrow \mathcal{A}^{\mathcal{O}_\mathcal{H}, \mathcal{O}_{\mathrm{CG}}, \mathcal{O}_{\mathrm{AM}}, \mathcal{O}_{\mathrm{HS}}, \mathcal{O}_{\mathrm{RM}}, \mathcal{O}_{\mathrm{Co}}}(\mathcal{P})$

Return 1 if the following hold and 0 otherwise

(1) $G^\star$ was obtained by a $\mathcal{O}_{\mathrm{CG}}$ query

(2) $\mathsf{ID}^\star$ was obtained by a $\mathcal{O}_{\mathrm{AM}}(G^\star)$ query

(3) neither $\mathcal{O}_{\mathrm{Co}}(G^\star)$ nor $\mathcal{O}_{\mathrm{Co}}(G^\star, \mathsf{ID}^\star)$ was queried

(4) if $\mathcal{O}_{\mathrm{Co}}(G^\star, \mathsf{ID})$ for some ID is queried, then $\mathcal{O}_{\mathrm{RM}}(G^\star, \mathsf{ID})$ is queried

(5) if $d^\star = 0$, then $\mathsf{SH.Handshake}(\mathcal{A}, \mathsf{ID}^\star)$ returns `Accept`
if $d^\star = 1$, then $\mathsf{SH.Handshake}(\mathsf{ID}^\star, \mathcal{A})$ returns `Accept`

*Let $q_\mathcal{H}, q_{\mathrm{CG}}, q_{\mathrm{AM}}, q_{\mathrm{HS}}, q_{\mathrm{RM}}, q_{\mathrm{Co}}, \tau \in \mathbb{N}^\mathbb{N}$. $\mathcal{A}$ is a $(q_\mathcal{H}, q_{\mathrm{CG}}, q_{\mathrm{AM}}, q_{\mathrm{HS}}, q_{\mathrm{RM}}, q_{\mathrm{Co}}, \tau)$-*IMP-ACT-adversary *if for all positive integer k, the experiment $\mathbf{Exp}^{\mathsf{imp\text{-}act}}_{\mathsf{SH},\mathcal{A}}(k)$ ends in expected time less than $\tau(k)$ and in this experiment $\mathcal{A}$ makes at most $q_x(k)$ queries to the oracle $\mathcal{O}_x$ for x in $\{\mathcal{H}, \mathrm{CG}, \mathrm{AM}, \mathrm{HS}, \mathrm{RM}, \mathrm{Co}\}$. We define the* success *of the adversary $\mathcal{A}$,* via $\mathbf{Succ}^{\mathsf{imp-act}}_{\mathsf{SH},\mathcal{A}}(k) = Pr\left[\mathbf{Exp}^{\mathsf{imp-act}}_{\mathsf{SH},\mathcal{A}}(k) = 1\right]$.

Let $q_\mathcal{H}, q_{\mathrm{CG}}, q_{\mathrm{AM}}, q_{\mathrm{HS}}, q_{\mathrm{RM}}, q_{\mathrm{Co}}, \tau \in \mathbb{N}^\mathbb{N}$ and let $\varepsilon \in [0,1]^\mathbb{N}$. The scheme SH is said to be $(q_\mathcal{H}, q_{\mathrm{CG}}, q_{\mathrm{AM}}, q_{\mathrm{HS}}, q_{\mathrm{RM}}, q_{\mathrm{Co}}, \tau, \varepsilon)$-resistant to impersonation against active adversary*, if for any $(q_\mathcal{H}, q_{\mathrm{CG}}, q_{\mathrm{AM}}, q_{\mathrm{HS}}, q_{\mathrm{RM}}, q_{\mathrm{Co}}, \tau)$-IMP-ACT-adversary $\mathcal{A}$ and any positive integer k, the function $\mathbf{Succ}^{\mathsf{imp-act}}_{\mathsf{SH},\mathcal{A}}(k)$ is smaller than $\varepsilon(k)$.*

Resistance to detection attacks. This property captures the requirement that an adversary who does not belong to or does not corrupt a member of a group G managed by an uncorrupt group authority, has only a negligible advantage in distinguishing an interplay with an honest member of G from one with a *simulator*. We consider an adversary $\mathcal{A}$ in the random oracle model, which runs in two stages. In the `find` stage, she takes the public parameters $\mathcal{P}$ as input, and outputs a triple $(G^\star, \mathsf{ID}^\star, d^\star)$, with some state information $\mathcal{I}^\star$. In the `guess` stage, she gets the information $\mathcal{I}^\star$, executes $\mathsf{SH.Handshake}$ with the challenger and outputs a bit $b^\star$.

The adversary $\mathcal{A}$ has access to the random oracle(s) $\mathcal{O}_\mathcal{H}$ and to the oracles $\mathcal{O}_{\mathrm{CG}}$, $\mathcal{O}_{\mathrm{AM}}$, $\mathcal{O}_{\mathrm{HS}}$, $\mathcal{O}_{\mathrm{RM}}$, and $\mathcal{O}_{\mathrm{Co}}$. In the `guess` stage, the challenger picks a bit b at random, and the protocol $\mathsf{SH.Handshake}$ is executed with $\mathsf{ID}^\star$ if $b = 0$, or

executed with a simulator SIM delivering values sampled uniformly at random in the suitable encoding space, if $b = 1$. $\mathcal{A}$ succeeds if $\mathsf{ID}^\star$ belongs to $G^\star$; $G^\star$ remains uncorrupt during $\mathcal{A}$'s execution; all corrupt users from $G^\star$ are excluded from $G^\star$ (in particular $\mathsf{ID}^\star$ is uncorrupt) and if $d^\star = b$.

Definition 6. *Let SH be a secret handshake scheme and let $\mathcal{A}$ be a PPTM. We consider the following random experiment, where $k \in \mathbb{N}$ is a security parameter:*

Experiment $\mathbf{Exp}^{\mathsf{det\text{-}act\text{-}b}}_{\mathsf{SH},\mathcal{A}}(k)$

$\mathcal{P} \xleftarrow{R} \mathsf{SH}.\mathtt{Setup}(k)$
$(G^\star, \mathsf{ID}^\star, d^\star, \mathcal{I}^\star) \leftarrow \mathcal{A}^{\mathcal{O}_\mathcal{H},\mathcal{O}_{\mathrm{CG}},\mathcal{O}_{\mathrm{AM}},\mathcal{O}_{\mathrm{HS}},\mathcal{O}_{\mathrm{RM}},\mathcal{O}_{\mathrm{Co}}}(\mathtt{find},\mathcal{P})$
$\mathcal{U}_{d^\star} \leftarrow \mathcal{A}^{\mathcal{O}_\mathcal{H},\mathcal{O}_{\mathrm{CG}},\mathcal{O}_{\mathrm{AM}},\mathcal{O}_{\mathrm{HS}},\mathcal{O}_{\mathrm{RM}},\mathcal{O}_{\mathrm{Co}}}(\mathtt{find},\mathcal{I}^\star)$
if $b = 0$ *then* $\mathcal{U}_{1-d^\star} \leftarrow \mathsf{ID}^\star$, *else* $\mathcal{U}_{1-d^\star} \leftarrow \mathrm{SIM}$
Execute $\mathsf{SH}.\mathsf{Handshake}(\mathcal{U}_0, \mathcal{U}_1)$
$b^\star \leftarrow \mathcal{A}^{\mathcal{O}_\mathcal{H},\mathcal{O}_{\mathrm{CG}},\mathcal{O}_{\mathrm{AM}},\mathcal{O}_{\mathrm{HS}},\mathcal{O}_{\mathrm{RM}},\mathcal{O}_{\mathrm{Co}}}(\mathtt{guess},\mathcal{I}^\star)$
Return 1 if the following hold and 0 otherwise
- *(1)* $G^\star$ *was obtained by a* $\mathcal{O}_{\mathrm{CG}}$ *query*
- *(2)* $\mathsf{ID}^\star$ *was obtained by a* $\mathcal{O}_{\mathrm{AM}}(G^\star)$ *query*
- *(3) neither* $\mathcal{O}_{\mathrm{Co}}(G^\star)$ *nor* $\mathcal{O}_{\mathrm{Co}}(G^\star, \mathsf{ID}^\star)$ *was queried*
- *(4) if* $\mathcal{O}_{\mathrm{Co}}(G^\star, \mathsf{ID})$ *for some* ID *is queried, then* $\mathcal{O}_{\mathrm{RM}}(G^\star, \mathsf{ID})$ *is queried*
- *(5)* $b^\star = b$

Let $q_\mathcal{H}, q_{\mathrm{CG}}, q_{\mathrm{AM}}, q_{\mathrm{HS}}, q_{\mathrm{RM}}, q_{\mathrm{Co}}, \tau \in \mathbb{N}^\mathbb{N}$. *$\mathcal{A}$ is a* $(q_\mathcal{H}, q_{\mathrm{CG}}, q_{\mathrm{AM}}, q_{\mathrm{HS}}, q_{\mathrm{RM}}, q_{\mathrm{Co}}, \tau)$-$\mathsf{DET\text{-}ACT}$-adversary *if for all positive integer k, the experiment* $\mathbf{Exp}^{\mathsf{det\text{-}act}}_{\mathsf{SH},\mathcal{A}}(k)$ *ends in expected time less than $\tau(k)$ and in this experiment $\mathcal{A}$ makes at most $q_x(k)$ queries to the oracle $\mathcal{O}_x$ for x in $\{\mathcal{H}, \mathrm{CG}, \mathrm{AM}, \mathrm{HS}, \mathrm{RM}, \mathrm{Co}\}$. We define the* advantage *of the adversary $\mathcal{A}$,* via

$$\mathbf{Adv}^{\mathsf{det-act}}_{\mathsf{SH},\mathcal{A}}(k) = \left| Pr\left[\mathbf{Exp}^{\mathsf{det-act}-0}_{\mathsf{SH},\mathcal{A}}(k) = 1\right] - Pr\left[\mathbf{Exp}^{\mathsf{det-act}-1}_{\mathsf{SH},\mathcal{A}}(k) = 1\right]\right|.$$

Let $q_\mathcal{H}, q_{\mathrm{CG}}, q_{\mathrm{AM}}, q_{\mathrm{HS}}, q_{\mathrm{RM}}, q_{\mathrm{Co}}, \tau \in \mathbb{N}^\mathbb{N}$ *and let* $\varepsilon \in [0,1]^\mathbb{N}$. *The scheme SH is said to be* $(q_\mathcal{H}, q_{\mathrm{CG}}, q_{\mathrm{AM}}, q_{\mathrm{HS}}, q_{\mathrm{RM}}, q_{\mathrm{Co}}, \tau, \varepsilon)$-resistant to detection against active adversaries*, if for any* $(q_\mathcal{H}, q_{\mathrm{CG}}, q_{\mathrm{AM}}, q_{\mathrm{HS}}, q_{\mathrm{RM}}, q_{\mathrm{Co}}, \tau)$-$\mathsf{DET\text{-}ACT}$-*adversary $\mathcal{A}$ and any positive integer k, the function* $\mathbf{Adv}^{\mathsf{det-act}}_{\mathsf{SH},\mathcal{A}}(k)$ *is smaller than* $\varepsilon(k)$.

Indistinguishability to eavesdropper. This property captures the requirement that an adversary has only a negligible advantage of distinguishing a successful handshake between uncorrupt members of a group G managed by an uncorrupt group authority from an unsuccessful one. We consider an adversary $\mathcal{A}$ in the ROM, which runs in two stages. In the `find` stage, she takes the public parameters $\mathcal{P}$ as input, and outputs a triple $(G^\star, \mathsf{ID}^\star_I, \mathsf{ID}^\star_M)$ with some state information $\mathcal{I}^\star$. In the `guess` stage, she gets the information $\mathcal{I}^\star$ and $\Upsilon^\star$, generated by the challenger depending on a random bit b. If $b = 0$, then $\Upsilon^\star$ is the transcript of an execution of $\mathsf{SH.Handshake}$ between $(\mathsf{ID}^\star_I, \mathsf{ID}^\star_M)$, otherwise $\Upsilon^\star$ is a bit string sampled uniformly at random in the transcript space. Eventually $\mathcal{A}(\mathtt{guess})$ outputs a bit $d^\star$.

The adversary $\mathcal{A}$ has access to the random oracle(s) $\mathcal{O}_{\mathcal{H}}$ and to the oracles $\mathcal{O}_{\mathrm{CG}}$, $\mathcal{O}_{\mathrm{AM}}$, $\mathcal{O}_{\mathrm{HS}}$, $\mathcal{O}_{\mathrm{RM}}$, and $\mathcal{O}_{\mathrm{Co}}$. It succeeds if $d^\star = b$, if $\mathsf{ID}_I^\star$ and $\mathsf{ID}_M^\star$ belongs to $G^\star$, and if $G^\star$, $\mathsf{ID}_I^\star$ and $\mathsf{ID}_M^\star$ remain uncorrupt during $\mathcal{A}$'s execution.

Definition 7. *Let SH be a secret handshake scheme and let $\mathcal{A}$ be a PPTM. We consider the following random experiment, where $k \in \mathbb{N}$ is a security parameter:*

$\boxed{\textit{Experiment } \mathbf{Exp}^{\mathsf{ind\text{-}act\text{-}b}}_{\mathsf{SH},\mathcal{A}}(k)}$
$\mathcal{P} \xleftarrow{R} \mathsf{SH.Setup}(k)$
$(G^\star, \mathsf{ID}_I^\star, \mathsf{ID}_M^\star, \mathcal{I}^\star) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathcal{H}},\mathcal{O}_{\mathrm{CG}},\mathcal{O}_{\mathrm{AM}},\mathcal{O}_{\mathrm{HS}},\mathcal{O}_{\mathrm{RM}},\mathcal{O}_{\mathrm{Co}}}(\mathcal{P})$
if $b = 0$ *then* $\Upsilon^\star \leftarrow \mathsf{SH.Handshake}(\mathsf{ID}_I^\star, \mathsf{ID}_M^\star)$ *else* $\Upsilon^\star \xleftarrow{R} \mathsf{TranscriptSpace}$
Return 1 if the following hold and 0 otherwise
(1) $G^\star$ *was obtained by a* $\mathcal{O}_{\mathrm{CG}}$ *query*
(2) $\mathsf{ID}_I^\star$ *and* $\mathsf{ID}_M^\star$ *were obtained by a* $\mathcal{O}_{\mathrm{AM}}(G^\star)$ *query*
(3) neither $\mathcal{O}_{\mathrm{Co}}(G^\star)$ *nor* $\mathcal{O}_{\mathrm{Co}}(G^\star, \mathsf{ID}^\star)$ *was queried*
(4) $d^\star = b$

*Let $q_{\mathcal{H}}, q_{\mathrm{CG}}, q_{\mathrm{AM}}, q_{\mathrm{HS}}, q_{\mathrm{RM}}, q_{\mathrm{Co}}, \tau \in \mathbb{N}^{\mathbb{N}}$. $\mathcal{A}$ is a $(q_{\mathcal{H}}, q_{\mathrm{CG}}, q_{\mathrm{AM}}, q_{\mathrm{HS}}, q_{\mathrm{RM}}, q_{\mathrm{Co}}, \tau)$-*IND-ACT-adversary *if for all positive integer k, the experiment $\mathbf{Exp}^{\mathsf{ind\text{-}act\text{-}b}}_{\mathsf{SH},\mathcal{A}}(k)$ ($b \in \{0,1\}$) ends in expected time less than $\tau(k)$ and in this experiment $\mathcal{A}$ makes at most $q_x(k)$ queries to the oracle $\mathcal{O}_x$ for x in $\{\mathcal{H}, \mathrm{CG}, \mathrm{AM}, \mathrm{HS}, \mathrm{RM}, \mathrm{Co}\}$. We define the* advantage *of the adversary $\mathcal{A}$,* via

$$\mathbf{Adv}^{\mathsf{ind-act}}_{\mathsf{SH},\mathcal{A}}(k) = \left| Pr\left[\mathbf{Exp}^{\mathsf{ind-act-real}}_{\mathsf{SH},\mathcal{A}}(k) = 1\right] - Pr\left[\mathbf{Exp}^{\mathsf{ind-act-random}}_{\mathsf{SH},\mathcal{A}}(k) = 1\right]\right|.$$

Let $q_{\mathcal{H}}, q_{\mathrm{CG}}, q_{\mathrm{AM}}, q_{\mathrm{HS}}, q_{\mathrm{RM}}, q_{\mathrm{Co}}, \tau \in \mathbb{N}^{\mathbb{N}}$ and let $\varepsilon \in [0,1]^{\mathbb{N}}$. The scheme SH is said to be $(q_{\mathcal{H}}, q_{\mathrm{CG}}, q_{\mathrm{AM}}, q_{\mathrm{HS}}, q_{\mathrm{RM}}, q_{\mathrm{Co}}, \tau, \varepsilon)$-indistinguishable against active adversaries*, if for any $(q_{\mathcal{H}}, q_{\mathrm{CG}}, q_{\mathrm{AM}}, q_{\mathrm{HS}}, q_{\mathrm{RM}}, q_{\mathrm{Co}}, \tau)$-IND-ACT-adversary $\mathcal{A}$ and any positive integer k, the function $\mathbf{Adv}^{\mathsf{ind-act}}_{\mathsf{SH},\mathcal{A}}(k)$ is smaller than $\varepsilon(k)$.*

4 The new protocols

Let Gen be an RSA-group generator. For any integer $k \in \mathbb{N}$ (*resp.* $r \in \mathbb{N}$), let $[\{0,1\}^* \longrightarrow \{0,1\}^k]$ (*resp.* $[\{0,1\}^* \longrightarrow \{0,1\}^r]$) be a hash function family and $\mathcal{H}_k$ (*resp.* $\mathcal{G}_r$) be a random member of this family. Let $f : \mathbb{N} \longrightarrow \mathbb{N}$, we denote in the following $r = f(k)$ (this map is derived from the security proofs; in practice, for a security requirement of 2^{80} operations, we use the values $k = 1024$ and $r = 160$). For all tuple (n, p, q, e, d, g) output by Gen, and all bit strings ID in $\{0,1\}^*$, with overwhelming probability the integer x, whose binary encoding is $\mathcal{H}_k(\mathsf{ID})$, is invertible modulo n. For the sake of simplification, we suppose that this always happens. Moreover, let us recall that an appropriate mechanism is used in order to ensure that encodings of integers modulo n do not reveal the value of n.

4.1 Heuristic of the constructions

As described above, the adversaries are afforded enormous power. Therefore, it appears prudent to follow general principles in the design of secret handshake protocols (paraphrased from [8]):

- **asymmetry principle:** *asymmetry in a protocol is desirable.*
  Symmetries in a protocol should be used with caution, due to both the possibility of *reflection attacks*, and attacks in which responses from one party can be re-used within a protocol.
- **chronology principle:** *messages within a particular protocol run should be logically linked or "chained" in some manner.*
  This principle is aimed at precluding *replay attacks* and *interleaving attacks.*
- **"add your own salt" principle**: *a party should be able to incorporate into the data being operated on a reasonable amount of data which he himself randomly selects.*
  In other words, a protocol should not require a party to carry out a cryptographic operation on inputs which may be entirely under the control of an adversary. The objective is to prevent the so called *chosen-ciphertext attacks.*

Moreover, mutual authentication is hardly an end in itself: a party goes through a secret handshake protocol in order to then conduct some transaction that is allowed only to member of his group. In an open setting, the authentication by itself is largely useless because an adversary can "hijack" the ensuing session. Therefore, to have secure transactions, some information from the handshake protocol must be used to authenticate flows in the transaction. We will come back to this issue in section 7, but note *hic et nunc* that this session key exchange should be designed with caution:

- **link principle:** *authentication and key exchange should be linked.*
  If mutual authentication and key exchange are independent, then an attacker could allow two parties to carry out authentication unhindered, and could take over one's party role in key exchange.

4.2 Description of OT-SH

Okamoto and Tanaka's Identity-Based Key Agreement Scheme. The identity-based key exchange protocol proposed by Okamoto and Tanaka in 1989 [16] is described as follows: on input a security parameter k, a trusted authority (the Private Key Generator, PKG) runs the RSA-group generator Gen, publishes (n, e, g), and keeps (p, q, d) secret. Let Alice and Bob be two entities of which identification information is ID_A and ID_B, respectively.

Alice keeps $s_A = \mathcal{H}_k(\mathsf{ID}_A)^{-d} \bmod n$ computed by the PKG as her own secret (and so does Bob with $s_B = \mathcal{H}_k(\mathsf{ID}_B)^{-d} \bmod n$). The scheme is then basically the Diffie-Hellman scheme implemented over $(\mathbb{Z}/n\mathbb{Z})^*$ (see figure 1). The security of this scheme remained open until 1998 when a solution was given by Mambo and Shizuya [11].

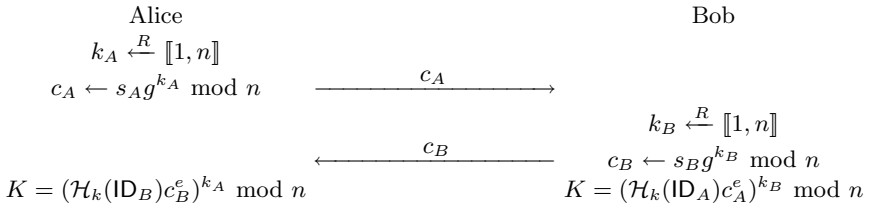

Fig. 1. Description of Okamoto-Tanaka identity-based key exchange protocol

Description of OT-SH. Our first new secret handshake scheme OT-SH is a simple variant of the previous protocol. The Setup algorithm establishes as public parameters the RSA-group generator and the hash functions $\mathcal{H}_k$ and $\mathcal{G}_r$. The algorithm CreateGroup is identical to the PKG key generation and the protocol AddMember to a user registration. The 4-round protocol Handshake is designed thanks to appropriate used of nonces and hash values (following the principles exemplified in the previous section) in the key exchange protocol. The scheme is described with all details in figure 2.

Algorithm Setup
Input: $k \in \mathbb{N}$
Output: $\mathcal{P}$
$\mathcal{P} = (k, \mathsf{Gen}, \mathcal{H}_k)$

Algorithm CreateGroup
Input: $\mathcal{P}$
Output: $(\mathbf{pk_G}, \mathbf{sk_G})$
$(n, p, q, e, d, g) \xleftarrow{R} \mathsf{Gen}(k)$
$\mathbf{pk_G} \leftarrow (n, e, g)$ $\mathbf{sk_G} \leftarrow d$

Protocol AddMember
Common Input: $\mathcal{P}, \mathbf{pk_G} = (n, e, g), \mathsf{ID}$
Member Input: ε
Group Input: $\mathbf{sk_G} = d$
Output: $(\mathbf{pk}_{\mathsf{ID}}, \mathbf{sk}_{\mathsf{ID}})$
$\mathbf{pk}_{\mathsf{ID}} \leftarrow \varepsilon$
$\mathbf{sk}_{\mathsf{ID}} \leftarrow (\mathcal{H}_k(\mathsf{ID}))^{-d} \bmod n$

Protocol Handshake	
Common Input: $\mathcal{P}$, $\mathsf{ID_I}$, $\mathsf{ID_M}$, $\mathbf{pk}_{\mathsf{ID_I}}$, $\mathbf{pk}_{\mathsf{ID_M}}$	
Init Input: $\mathbf{pk_{G_I}} = (n_\mathsf{I}, e_\mathsf{I}, g_\mathsf{I})$, $\mathbf{sk}_{\mathsf{ID_I}}$ Match Input: $\mathbf{pk_{G_M}} = (n_\mathsf{M}, e_\mathsf{M}, g_\mathsf{M})$, $\mathbf{sk}_{\mathsf{ID_M}}$	
Output: $(b_\mathsf{I}, b_\mathsf{M}) \in \{\mathtt{Accept}, \mathtt{Reject}\}^2$	
① Init's round: $k_\mathsf{I} \xleftarrow{R} \{0,1\}^{n_\mathsf{I}}$, $c_\mathsf{I} \leftarrow \mathbf{sk}_{\mathsf{ID_I}} \cdot g_\mathsf{I}^{k_\mathsf{I}} \bmod n_\mathsf{I}$	$\rightsquigarrow c_\mathsf{I}$
❶ Match's round: $k_\mathsf{M} \xleftarrow{R} \{0,1\}^{n_\mathsf{M}}$, $c_\mathsf{M} \leftarrow \mathbf{sk}_{\mathsf{ID_M}} \cdot g_\mathsf{M}^{k_\mathsf{M}} \bmod n_\mathsf{M}$	$\rightsquigarrow c_\mathsf{M}$
② Init's round: $V_\mathsf{I} \leftarrow \mathcal{G}_r\left[(\mathcal{H}_k(\mathsf{ID_M}) \cdot c_\mathsf{M}^{e_\mathsf{I}})^{k_\mathsf{I}} \bmod n_\mathsf{I} \| \mathsf{ID_I} \| \mathsf{ID_M} \| c_\mathsf{I} \| c_\mathsf{M} \| 0\right]$	$\rightsquigarrow V_\mathsf{I}$
❷ Match's round	
If $\mathcal{G}_r\left[(\mathcal{H}_k(\mathsf{ID_I}) \cdot c_\mathsf{I}^{e_\mathsf{M}})^{k_\mathsf{M}} \bmod n_\mathsf{M} \| \mathsf{ID_I} \| \mathsf{ID_M} \| c_\mathsf{I} \| c_\mathsf{M} \| 0\right] = V_\mathsf{I}$	
then $b_\mathsf{M} = \mathtt{Accept}$; $V_\mathsf{M} \leftarrow \mathcal{G}_r((\mathcal{H}_k(\mathsf{ID_I}) \cdot c_\mathsf{I}^{e_\mathsf{M}})^{k_\mathsf{M}} \bmod n_\mathsf{M} \| \mathsf{ID_I} \| \mathsf{ID_M} \| c_\mathsf{I} \| c_\mathsf{M} \| 1)$	
else $b_\mathsf{M} = \mathtt{Reject}$; $V_\mathsf{M} \xleftarrow{R} \{0,1\}^r$	$\rightsquigarrow V_\mathsf{M}$
• Init's execution ending	
If $\mathcal{G}_r\left[(\mathcal{H}_k(\mathsf{ID_M}) \cdot c_\mathsf{M}^{e_\mathsf{I}})^{k_\mathsf{I}} \bmod n_\mathsf{I} \| \mathsf{ID_I} \| \mathsf{ID_M} \| c_\mathsf{I} \| c_\mathsf{M} \| 1\right] = V_\mathsf{M}$	
then $b_\mathsf{I} \leftarrow \mathtt{Accept}$ else $b_\mathsf{I} \leftarrow \mathtt{Reject}$	

Fig. 2. Description of the scheme OT-SH

4.3 Description of Gi+SH and Gi×SH

Girault's Self-Certified Key Agreement Scheme. In 1991, Girault [9] proposed a key agreement which allows to obtain a shared secret key without exchange of messages (but it does require that the public keys of each party are known to the other), based on his new idea of self-certified public key. Rigorous security of this protocol (and variants) has been made clear only recently [15] by Oh, Mambo, Shizuya and Won. With the notations from section 4.2, Alice picks uniformly at random $s_A \in \mathbb{Z}/n\mathbb{Z}$, computes $S = g^{s_A} \bmod n$ and sends S_A together with a designated verifier proof of knowledge of s_A to the authority. If the proof is valid, the authority computes Alice's public key P_A as the RSA signature of the difference (*resp.* quotient) of S_A and $\mathcal{H}_k(\mathsf{ID}_A)$: $P_A = (S_A - \mathcal{H}_k(\mathsf{ID}_A))^d \bmod n$ (*resp.* $P_A = (S_A/\mathcal{H}_k(\mathsf{ID}_A))^d \bmod n$), therefore $g^{s_A} = P_A^e + \mathcal{H}_k(\mathsf{ID}_A) \bmod n$ (*resp.* $g^{s_A} = P_A^e \cdot \mathcal{H}_k(\mathsf{ID}_A) \bmod n$). Similarly, Bob's public key is P_B such that $g^{s_B} = P_B^e + \mathcal{H}_k(\mathsf{ID}_B) \bmod n$ (*resp.* $g^{s_B} = P_B^e \cdot \mathcal{H}_k(\mathsf{ID}_B) \bmod n$) and s_B is known only to Bob. Alice and Bob can thereafter simply exchange an authenticated key by choosing

$$\begin{aligned} K &= (P_A^e + \mathcal{H}_k(\mathsf{ID}_A))^{s_B} = (P_B^e + \mathcal{H}_k(\mathsf{ID}_B))^{s_A} \mod n \\ \big(\textit{resp. } K &= (P_A^e \cdot \mathcal{H}_k(\mathsf{ID}_A))^{s_B} = (P_B^e \cdot \mathcal{H}_k(\mathsf{ID}_B))^{s_A} \mod n\big) \end{aligned}$$

This protocol is clearly related to Diffie-Hellman's one, but contrary to it, makes Alice sure that she shares K with Bob and conversely.

Description of Gi+SH and Gi×SH. The new secret handshake schemes Gi-SH and Gi×SH are completely analogous to the one proposed in [1] by Balfanz *et al.*: the non-interactive key-agreement protocol of Sakai *et al.* being replaced by Girault's additive and multiplicative scheme (respectively). They are described in figure 3.

Remark 1. Note that, in the schemes Gi⊕SH (with $\oplus = +$ or $\oplus = \times$), after the run of the protocol Gi⊕SH.AddMember, the group authority does not learn the secret key of the member and thus in contrast to the previous scheme, cannot impersonate that member. However, in [18], Saeednia described a shortcoming of Girault's self-certified model that may be exploited by the authority to compute users' secret keys. This attack applies as well on Gi⊕SH, but it is easy to make this attack ineffective by taking additional precautions. This might be interesting for high-security needs (*e.g.* in traitor tracing).

5 Security results

The following theorems state that the advantage of any active impersonator, detector or distinguisher adversary against the schemes OT-SH and Gi⊕SH in the ROM can be upper-bounded *via* an explicit function related to the success of solving the RSA problem, the difference RSA problem or the composite Diffie-Hellman problem in Gen. More precisely, the hash function $\mathcal{H}_k$ is modeled by a

Algorithm Setup	Algorithm CreateGroup
Input: $k \in \mathbb{N}$	Input: $\mathcal{P}$
Output: $\mathcal{P}$	Output: $(\mathbf{pk_G}, \mathbf{sk_G})$
$\mathcal{P} \leftarrow (k, \mathsf{Gen}, \mathcal{H}_k)$	$(n, p, q, e, d, g) \xleftarrow{R} \mathsf{Gen}(k)$; $\mathbf{pk_G} \leftarrow (n, e, g)$; $\mathbf{sk_G} \leftarrow d$

Protocol AddMember
Common Input: $\mathcal{P}, \mathbf{pk_G}, \mathsf{ID}$ Group Input: $\mathbf{sk_G}$
Output: $(\mathbf{pk}_{\mathsf{ID}}, \mathbf{sk}_{\mathsf{ID}})$
① Member's round: $\mathbf{sk}_{\mathsf{ID}} \xleftarrow{R} [\![2^k, 2^{k+1}]\!]$, $S \leftarrow g^{\mathbf{sk}_{\mathsf{ID}}}$, $\mathsf{p} \xleftarrow{R} \mathrm{DVPK}(\gamma \mid S = g^\gamma) \rightsquigarrow (S, \mathsf{p})$
❶ Group's round: Verify p ; $\mathbf{pk}_{\mathsf{ID}} \leftarrow (S \ominus \mathcal{H}_k(\mathsf{ID}))^d \bmod n$ $\rightsquigarrow \mathbf{pk}_{\mathsf{ID}}$

Protocol Handshake
Common Input: $\mathcal{P}$, $\mathsf{ID_I}$, $\mathsf{ID_M}$, $\mathbf{pk}_{\mathsf{ID_I}}$, $\mathbf{pk}_{\mathsf{ID_M}}$
Init Input: $\mathbf{pk}_{\mathbf{G_I}} = (n_\mathsf{I}, e_\mathsf{I}, g_\mathsf{I})$, $\mathbf{sk}_{\mathsf{ID_I}}$
Match Input: $\mathbf{pk}_{\mathbf{G_M}} = (n_\mathsf{M}, e_\mathsf{M}, g_\mathsf{M})$, $\mathbf{sk}_{\mathsf{ID_M}}$
Output: $(b_\mathsf{I}, b_\mathsf{M}) \in \{\mathtt{Accept}, \mathtt{Reject}\}^2$
① Init's round: $r_\mathsf{I} \xleftarrow{R} \{0,1\}^r$ $\rightsquigarrow r_\mathsf{I}$
❶ Match's round: $r_\mathsf{M} \xleftarrow{R} \{0,1\}^r$;
$V_\mathsf{M} \leftarrow \mathcal{G}_r\left[(\mathbf{pk}_{\mathsf{ID_I}}{}^{e_\mathsf{M}} \oplus \mathcal{H}_k(\mathsf{ID_I}))^{\mathbf{sk}_{\mathsf{ID_M}}} \bmod n_\mathsf{M} \| \mathsf{ID_I} \| \mathsf{ID_M} \| r_\mathsf{I} \| r_\mathsf{M} \| 0\right]$ $\rightsquigarrow (r_\mathsf{M}, V_\mathsf{M})$
② Init's round: $K_I \leftarrow (\mathbf{pk}_{\mathsf{ID_M}}{}^{e_\mathsf{I}} \oplus \mathcal{H}_k(\mathsf{ID_M}))^{\mathbf{sk}_{\mathsf{ID_I}}} \bmod n_\mathsf{I}$
If $\mathcal{G}_r\,[K_I \| \mathsf{ID_I} \| \mathsf{ID_M} \| r_\mathsf{I} \| r_\mathsf{M} \| 0] = V_\mathsf{M}$
then $b_\mathsf{I} \leftarrow \mathtt{Accept}$; $V_\mathsf{I} \leftarrow \mathcal{G}_r\,[K_I \| \mathsf{ID_I} \| \mathsf{ID_M} \| r_\mathsf{I} \| r_\mathsf{M} \| 1]$
else $b_\mathsf{I} \leftarrow \mathtt{Reject}$; $V_\mathsf{I} \xleftarrow{R} \{0,1\}^r$ $\rightsquigarrow V_\mathsf{I}$
• Match's execution ending
If $\mathcal{G}_r\left[(\mathbf{pk}_{\mathsf{ID_I}}{}^{e_\mathsf{M}} \oplus \mathcal{H}_k(\mathsf{ID_I}))^{\mathbf{sk}_{\mathsf{ID_M}}} \bmod n_\mathsf{M} \| \mathsf{ID_I} \| \mathsf{ID_M} \| r_\mathsf{I} \| r_\mathsf{M} \| 1\right] = V_\mathsf{I}$
then $b_\mathsf{M} \leftarrow \mathtt{Accept}$; else $b_\mathsf{M} \leftarrow \mathtt{Reject}$

Fig. 3. Description of the schemes Gi+SH and Gi×SH ($(\oplus, \ominus) = (+, -)$ and $(\times, /)$)

random oracle, while the hash function $\mathcal{G}_r$ is modeled by a non-programmable random oracle (more precisely, we only suppose that $\mathcal{G}_r$ has the *evaluation point knowledge* (EPK) property: it is not possible to learn the value of what $\mathcal{G}_r(\sigma)$ without knowing the bit string σ).

In all cases but two, the proof is more or less routine: the main difficulty of the analysis comes from the introduction of the RSA challenge into the public base. Indeed in the reductionist security proofs, the element g, supposed to be of maximal order in $(\mathbb{Z}/n\mathbb{Z})^*$, is replaced by a random element of $(\mathbb{Z}/n\mathbb{Z})^*$. Lemma 1 asserts that with probability close to $\exp(-1)$, among $(k+1)^2$ such random elements there is an element with order $\lambda(n)$.

Lemma 1. *Let $k \geq 1$ be an integer, and let $n \in 2\mathsf{Factors}(k)$. If an integer g is drawn uniformly at random from $[\![1, n]\!]$, then* $\Pr[\mathrm{ord}(g) = \lambda(n)] > 1/(110 \ln k)$.

Proof. This lemma is a simple consequence of [17, Theorem 15]. □

We illustrate the application of this lemma in the proof of the impersonation resistance of OT-SH (*cf.* Theorem 1). The impersonation resistance (as the detection resistance) of Gi+SH rely on the difference RSA problem. Since this is a less classical assumption, we carry out the arguments in full detail (*cf.* Theorem 2). The remaining four security results are left to the reader and will follow similar proofs for the two given here. In the theorems, $T_{\mathrm{Exp}}(k)$ denotes the time complexity for an exponentiation modulo a $2k+1$-bit integer.

5.1 Security of the scheme OT-SH

Theorem 1. *Let Gen be an RSA-group generator, let $f : \mathbb{N} \to \mathbb{N}$ and let OT-SH be the secret handshake protocol instantiated with Gen and f.*
Let $q_{\mathcal{H}}, q_{\mathrm{CG}}, q_{\mathrm{AM}}, q_{\mathrm{HS}}, q_{\mathrm{RM}}, q_{\mathrm{Co}}, \tau \in \mathbb{N}^{\mathbb{N}}$.

1. *Let $\mathcal{A}$ be a $(q_{\mathcal{H}}, q_{\mathrm{CG}}, q_{\mathrm{AM}}, q_{\mathrm{HS}}, q_{\mathrm{RM}}, q_{\mathrm{Co}}, \tau)$-IMP-ACT-adversary against OT-SH, in the ROM. There exist a τ'-RSA adversary $\mathcal{B}$ such that*

$$\begin{cases} \mathbf{Succ}^{\mathsf{rsa}}_{\mathsf{Gen},\mathcal{B}}(k) \geq \mathbf{Succ}^{\mathsf{imp-act}}_{\mathsf{OT\text{-}SH},\mathcal{A}}(k)/(110\ln(k)\cdot q_{\mathrm{CG}}(k) q_{\mathcal{H}}(k)) \\ \tau'(k) \leq [\tau(k) + (q_{\mathcal{H}}(k) + q_{\mathrm{CG}}(k) + q_{\mathrm{AM}}(k) + 3q_{\mathrm{HS}}(k) + O(1))T_{\mathrm{Exp}}(k)] \end{cases}$$

   for all positive integers k.
2. *Let $\mathcal{A}$ be a $(q_{\mathcal{H}}, q_{\mathrm{CG}}, q_{\mathrm{AM}}, q_{\mathrm{HS}}, q_{\mathrm{RM}}, q_{\mathrm{Co}}, \tau)$-DET-ACT-adversary against OT-SH, in the ROM. There exist a τ'-RSA adversary $\mathcal{B}$ such that*

$$\begin{cases} \mathbf{Succ}^{\mathsf{rsa}}_{\mathsf{Gen},\mathcal{B}}(k) \geq \mathbf{Adv}^{\mathsf{det-act}}_{\mathsf{OT\text{-}SH},\mathcal{A}}(k)/(220\ln(k)\cdot q_{\mathrm{CG}}(k) q_{\mathcal{H}}(k)) \\ \tau'(k) \leq [\tau(k) + (q_{\mathcal{H}}(k) + q_{\mathrm{CG}}(k) + q_{\mathrm{AM}}(k) + 3q_{\mathrm{HS}}(k) + O(1))T_{\mathrm{Exp}}(k)] \end{cases}$$

   for all positive integers k.
3. *Let $\mathcal{A}$ be a $(q_{\mathcal{H}}, q_{\mathrm{CG}}, q_{\mathrm{AM}}, q_{\mathrm{HS}}, q_{\mathrm{RM}}, q_{\mathrm{Co}}, \tau)$-IND-ACT-adversary against OT-SH, in the ROM, making at most $q_{\mathcal{G}}(k)$ queries to the non-programmable random oracle $\mathcal{O}_{\mathcal{G}}$ in the experiments $\mathbf{Exp}^{\mathsf{ind\text{-}act\text{-}b}}_{\mathsf{OT\text{-}SH},\mathcal{A}}(k)$ (for $b \in \{0,1\}$ and $k \in \mathbb{N}$). There exist a τ'-COMP-CDH adversary $\mathcal{B}$ such that*

$$\begin{cases} \mathbf{Succ}^{\mathsf{comp-cdh}}_{\mathsf{Gen},\mathcal{B}}(k) \geq \mathbf{Adv}^{\mathsf{ind-act}}_{\mathsf{OT\text{-}SH},\mathcal{A}}(k)/[2q_{\mathcal{G}}(k)] \\ \tau'(k) \leq \tau(k) + (q_{\mathcal{H}}(k) + 3q_{\mathrm{HS}}(k) + O(1))T_{\mathrm{Exp}}(k) \end{cases}$$

   for all positive integers k.

Proof. We prove only the first part of the theorem. Our method of proof is inspired by Shoup [21]: we define a sequence of games $\mathsf{Game}_0, \ldots, \mathsf{Game}_5$ starting from the actual adversary $\mathcal{A}$ and modify it step by step until we reach a final game whose success probability has an upper bound related to solving the RSA problem. All the games operate on the same underlying probability space: the setup, the public and private keys of the groups and the members, the coin tosses of $\mathcal{A}$ and the random oracle.

Let k be a security parameter, let (n, p, q, e, d, g) be a 6-tuple generated by $\mathsf{Gen}(k)$ and let $\alpha \in (\mathbb{Z}/n\mathbb{Z})^*$ be a random instance of the RSA problem. We construct a reduction algorithm $\mathcal{B}$ which on input (n, e, α) outputs an element $x \in (\mathbb{Z}/n\mathbb{Z})^*$ aimed to satisfy $x^e = \alpha \bmod n$.

Game_0 We consider a $(q_{\mathcal{H}}, q_{\mathrm{CG}}, q_{\mathrm{AM}}, q_{\mathrm{HS}}, q_{\mathrm{RM}}, q_{\mathrm{Co}}, \tau)$-IMP-ACT-adversary $\mathcal{A}$, against the scheme OT-SH instantiated with the RSA-group generator Gen, in the random oracle model. $\mathcal{A}$ takes the public parameters as input, and outputs a triple $(G^\star, \mathsf{ID}^\star, d^\star)$. $\mathcal{A}$ has access to the random oracle $\mathcal{O}_{\mathcal{H}}$ and to the oracles $\mathcal{O}_{\mathrm{CG}}$, $\mathcal{O}_{\mathrm{AM}}$, $\mathcal{O}_{\mathrm{HS}}$, $\mathcal{O}_{\mathrm{RM}}$, and $\mathcal{O}_{\mathrm{Co}}$. $\mathcal{A}$ succeeds if $\mathsf{ID}^\star$ belongs to $G^\star$, $G^\star$ remains uncorrupt during $\mathcal{A}$'s execution, all corrupt users from $G^\star$ are excluded from $G^\star$ and if in the protocol OT-SH.Handshake$(\mathcal{A}, \mathsf{ID}^\star)$ (*resp.* OT-SH.Handshake$(\mathsf{ID}^\star, \mathcal{A})$), the member $\mathsf{ID}^\star$ returns `Accept` when $d^\star = 0$ (*resp.* when $d^\star = 1$).
In any game Game_i ($i \in [\![1, 5]\!]$), we denote by Imp_i this event. Without loss of generality we can suppose that any time $\mathcal{A}$ makes a query involving a pseudonym ID to one of the oracles $\mathcal{O}_{\mathrm{CG}}$, $\mathcal{O}_{\mathrm{AM}}$, $\mathcal{O}_{\mathrm{HS}}$, $\mathcal{O}_{\mathrm{RM}}$, and $\mathcal{O}_{\mathrm{Co}}$, $\mathcal{A}$ has previously queried ID to the random oracle $\mathcal{O}_{\mathcal{H}}$. In particular, we suppose that $\mathcal{A}$ has queried ID^* and $\mathsf{ID}_{\mathcal{A}}$ (the pseudonym used by $\mathcal{A}$ in the final execution of OT-SH.Handshake) to the random oracle $\mathcal{O}_{\mathcal{H}}$. By definition, we have $\Pr[\mathsf{Imp}_0] = \mathbf{Succ}^{\mathsf{imp-act}}_{\mathsf{OT\text{-}SH},\mathcal{A}}(k)$.

Game_1 The algorithm $\mathcal{B}$ picks uniformly at random an index $\ell_1 \in [\![1, q_{\mathrm{CG}}(k)]\!]$ and aborts if the group $G^\star$ was not obtained at the ℓ_1's query to the oracle $\mathcal{O}_{\mathrm{CG}}$. We obtain $\Pr[\mathsf{Imp}_1] = \Pr[\mathsf{Imp}_0]/q_{\mathrm{CG}}(k)$.

Game_2 $\mathcal{B}$ simulates the $\mathcal{O}_{\mathrm{CG}}$ oracle. For the i-th query ($i \in [\![1, q_{\mathrm{CG}}(k)]\!] \setminus \{\ell_1\}$), $\mathcal{B}$ executes $\mathsf{Gen}(k)$ and gets $(n_i, p_i, q_i, e_i, d_i, g_i)$. It outputs (n_i, e_i, g_i) as the answer to $\mathcal{A}$'s i-th query for the group denoted by G_i and stores d_i.
For the ℓ_1-th query, $\mathcal{B}$ picks uniformly at random $r' \in (\mathbb{Z}/n\mathbb{Z})^*$, computes $r = r'^e \bmod n$ and outputs $(n, e, (r\alpha)^e)$. We denote by Λ the event that $(r\alpha)^e$ is of order $\lambda(n)$ in $(\mathbb{Z}/n\mathbb{Z})^*$. If Λ happens, then the distribution of the group authorities keys is unchanged. Therefore, we have $\Pr[\mathsf{Imp}_2|\Lambda] = \Pr[\mathsf{Imp}_1]$.

Game_3 The algorithm $\mathcal{B}$ picks uniformly at random an index $\ell_2 \in [\![1, q_{\mathcal{H}}(k)]\!]$. For the i-th bit string ID queried to $\mathcal{O}_{\mathcal{H}}$, $\mathcal{B}$ picks at random $s_{\mathsf{ID}} \in (\mathbb{Z}/n\mathbb{Z})^*$ and $\mathcal{B}$ sets $h_{\mathsf{ID}} = s_{\mathsf{ID}}^{e^2}$ if $i \neq \ell_2$, otherwise it sets $h_{\mathsf{ID}} = s_{\mathsf{ID}}^{e} \cdot \alpha$. $\mathcal{B}$ returns h_{ID} as the answer to the oracle call and stores the 4-tuple $(\mathsf{ID}, h_{\mathsf{ID}}, s_{\mathsf{ID}}, i)$ in the H_k-List. $\mathcal{B}$ discards execution such that $\mathsf{ID}_{\mathcal{A}}$ was not the ℓ_2-th query to the oracle $\mathcal{H}$. We obtain $\Pr[\mathsf{Imp}_3] = \Pr[\mathsf{Imp}_2]/q_{\mathcal{H}}(k)$.

Game_4 Now, $\mathcal{B}$ simulates the $\mathcal{O}_{\mathrm{AM}}$ and the $\mathcal{O}_{\mathrm{Co}}$ oracles. The queries $\mathcal{O}_{\mathrm{AM}}(G_i)$ for $i \in [\![1, q_{\mathrm{CG}}(k)]\!]$ are easily simulated since OT-SH.AddMember outputs only a secret key associated with a pseudonym.
For all $i \in [\![1, q_{\mathrm{CG}}(k)]\!] \setminus \{\ell_1\}$, thanks to the knowledge of d_i, $\mathcal{B}$ can perfectly simulate the member key generation protocol for the group G_i, and therefore can reply to $\mathcal{A}$'s queries involving members of G_i. For a member of $G^\star$ with pseudonym ID queried to the $\mathcal{O}_{\mathrm{AM}}$ or the $\mathcal{O}_{\mathrm{Co}}$ oracle, we know that $\mathsf{ID} \neq \mathsf{ID}_{\mathcal{A}}$. Therefore $\mathcal{B}$ can retrieve s_{ID} in the H_k-List and computes the d-th power of $\mathcal{O}_{\mathcal{H}}(\mathsf{ID})^{-1} = h_{\mathsf{ID}}^{-1}$ as s_{ID}^{-e}. Thanks to the knowledge of this value $\mathcal{B}$ can reply to $\mathcal{A}$'s queries involving members of $G^\star$. This perfectly simulates the oracles, therefore we have $\Pr[\mathsf{Imp}_4|\Lambda] = \Pr[\mathsf{Imp}_3|\Lambda]$.

Game_5 Finally, $\mathcal{B}$ simulates the handshake protocols for members with pseudonyms say ID_I and ID_M. The only difficulty happens when $\mathsf{ID}_I = \mathsf{ID}_{\mathcal{A}}$ or $\mathsf{ID}_M = \mathsf{ID}_{\mathcal{A}}$. In this case, $\mathcal{B}$ picks uniformly at random $u \in [\![1, n]\!]$, sets $c_I = (r\alpha)^{eu} \cdot s_{\mathsf{ID}_I}^{-e}$

or $c_M = (r\alpha)^{eu} \cdot s_{\mathsf{ID}_M}^{-d}$ (respectively). The algorithm $\mathcal{B}$ picks uniformly at random $v \in [\![1, n]\!]$, sets $c_{\mathcal{A}} = s_{\mathsf{ID}_{\mathcal{A}}}^{-1} r'(r\alpha)^v$ (so that $[c_{\mathcal{A}}(s_{\mathsf{ID}_{\mathcal{A}}}\alpha^d)] \in \langle g \rangle$) and $K = (c_{\mathcal{A}}^e h_{\mathsf{ID}_{\mathcal{A}}})^u$. Querying these suitable values to the non-programmable random oracle $\mathcal{G}_r$, $\mathcal{B}$ can perfectly simulate the $\mathcal{O}_{\mathrm{HS}}$. Therefore, we have $\Pr[\mathsf{Imp}_5|\Lambda] = \Pr[\mathsf{Imp}_4|\Lambda]$.

To summarize, when the Game_5 terminates, $\mathcal{A}$ outputs the triple $(G^\star, \mathsf{ID}^\star, d^\star)$ such that $G^\star$'s public key is $(n, e, (r\alpha)^e)$ and $\mathcal{O}_{\mathcal{H}}(\mathsf{ID}^\star) = s_{\mathsf{ID}^\star}^{e^2}$. Equiped with the pseudonym $\mathsf{ID}_{\mathcal{A}}$, $\mathcal{A}$ interacts with the reduction algorithm $\mathcal{B}$ emulating the member with pseudonym $\mathsf{ID}^\star$ to execute the protocol $\mathsf{OT\text{-}SH.Handshake}(\mathcal{A}, \mathsf{ID}^\star)$ (or $\mathsf{OT\text{-}SH.Handshake}(\mathcal{A}, \mathsf{ID}^\star)$ depending on $d^\star$).
From the simulation, we know that $\mathcal{O}_{\mathcal{H}}(\mathsf{ID}_{\mathcal{A}}) = s_{\mathsf{ID}_{\mathcal{A}}}^e \alpha$. Whatever the bit $d^\star$ is, $\mathcal{B}$ picks uniformly at random $t \in [\![1, n]\!]$ such that t is coprime to e and sets $c_{\mathsf{ID}^\star} = (r\alpha)^t s_{\mathsf{ID}^\star}^e$ in the execution of the handshake protocol.

- If the protocol is successful, thanks to the EPK property of $\mathcal{G}_r$, $\mathcal{B}$ retrieves in its transcript K and $c_{\mathsf{ID}_{\mathcal{A}}}$ such that $K = g^{ek^\star k_{\mathcal{A}}}$, $c_{\mathsf{ID}_{\mathcal{A}}} = g^{k_{\mathcal{A}}} s_{\mathsf{ID}_{\mathcal{A}}}^{-1} \alpha^{-d}$ and $c_{\mathsf{ID}^\star} = g^{k^\star}(s_{\mathsf{ID}^\star}^{-e^2})^d = g^{k^\star} s_{\mathsf{ID}^\star}^{-e} = (r\alpha^e)^{k^\star} s_{\mathsf{ID}^\star}^{-e}$.
  If Λ happens, then $ek^\star \equiv t \bmod \lambda(n)$, $c_{\mathsf{ID}_{\mathcal{A}}}^t = g^{tk_{\mathcal{A}}} s_{\mathsf{ID}_{\mathcal{A}}}^{-t} \alpha^{-dt} = K s_{\mathsf{ID}_{\mathcal{A}}}^{-t} \alpha^{-dt}$ and thus we have $(c_{\mathsf{ID}_{\mathcal{A}}}^t K^{-1} s_{\mathsf{ID}_{\mathcal{A}}}^t)^e = \alpha^{-t}$. Finally, using the extended Euclidean algorithm $\mathcal{B}$ finds integers u, v such that $ev - tu = 1$, and computes $x = (c_{\mathsf{ID}_{\mathcal{A}}}^t K^{-1} s_{\mathsf{ID}_{\mathcal{A}}}^{-t})^u \alpha^v \bmod n$ such that

$$\left[(c_{\mathsf{ID}_{\mathcal{A}}}^t K^{-1} s_{\mathsf{ID}_{\mathcal{A}}}^{-t})^u \alpha^v\right]^e = \alpha^{-tu}\alpha^{ev} = \alpha \bmod n.$$

- Otherwise, $\mathcal{B}$ picks uniformly at random $x \in [\![1, n]\!]$.

The output of $\mathcal{B}$ is x and with probability at least $\Pr[\mathsf{Imp}_5|\Lambda] + 2^{-2k-1}$ it is equal to the e-th root of the RSA challenge α. This probability is greater than $\mathbf{Succ}^{\mathsf{imp-act}}_{\mathsf{OT\text{-}SH},\mathcal{A}}(k)/(q_{\mathrm{CG}}(k)q_{\mathcal{H}}(k))$. By lemma 1, we get the claimed bounds for $\mathbf{Succ}^{\mathsf{rsa}}_{\mathsf{Gen},\mathcal{B}}$ and τ'. □

5.2 Security of the scheme Gi+SH

Theorem 2. *Let Gen be an RSA-group generator, let $f : \mathbb{N} \to \mathbb{N}$ and let Gi+SH be the secret handshake protocol instantiated with Gen and f.*
Let $q_{\mathcal{H}}, q_{\mathcal{G}}, q_{\mathrm{CG}}, q_{\mathrm{AM}}, q_{\mathrm{HS}}, q_{\mathrm{RM}}, q_{\mathrm{Co}}, \tau \in \mathbb{N}^{\mathbb{N}}$.

1. *Let $\mathcal{A}$ be a $(q_{\mathcal{H}}, q_{\mathrm{CG}}, q_{\mathrm{AM}}, q_{\mathrm{HS}}, q_{\mathrm{RM}}, q_{\mathrm{Co}}, \tau)$-IMP-ACT-adversary against Gi+SH, in the ROM. There exist a τ'-DIFF-RSA adversary $\mathcal{B}$ such that*

$$\begin{cases} \mathbf{Succ}^{\mathsf{diff-rsa}}_{\mathsf{Gen},\mathcal{B}}(k) \geq \mathbf{Succ}^{\mathsf{imp-act}}_{\mathsf{Gi+SH},\mathcal{A}}(k)/[\exp(2)q_{\mathrm{CG}}(k)(q_{\mathrm{AM}}(k) + q_{\mathrm{Co}}(k) + 1)] \\ \tau'(k) \leq \tau(k) + (2q_{\mathcal{H}}(k) + 2q_{\mathrm{CG}}(k) + q_{\mathrm{AM}}(k) + q_{\mathrm{HS}}(k) + O(1))T_{\mathrm{Exp}}(k) \end{cases}$$

   for all positive integers k.

2. *Let $\mathcal{A}$ be a $(q_{\mathcal{H}}, q_{\mathrm{CG}}, q_{\mathrm{AM}}, q_{\mathrm{HS}}, q_{\mathrm{RM}}, q_{\mathrm{Co}}, \tau)$-DET-ACT-adversary against Gi+SH, in the ROM. There exist a $(q_{\mathrm{AM}} + q_{\mathrm{Co}}, \tau')$-DIFF-RSA adversary $\mathcal{B}$ such that*

$$\begin{cases} \mathbf{Succ}^{\mathsf{diff-rsa}}_{\mathsf{Gen},\mathcal{B}}(k) \geq \mathbf{Adv}^{\mathsf{det-act}}_{\mathsf{Gi+SH},\mathcal{A}}(k)/[2\exp(2)q_{\mathrm{CG}}(k)(q_{\mathrm{AM}}(k) + q_{\mathrm{Co}}(k) + 1)] \\ \tau'(k) \leq (2q_{\mathcal{H}}(k) + 2q_{\mathrm{CG}}(k) + q_{\mathrm{AM}}(k) + q_{\mathrm{HS}}(k) + O(1))T_{\mathrm{Exp}}(k) \end{cases}$$

for all positive integers k.

3. *Let $\mathcal{A}$ be a $(q_{\mathcal{H}}, q_{\mathrm{CG}}, q_{\mathrm{AM}}, q_{\mathrm{HS}}, q_{\mathrm{RM}}, q_{\mathrm{Co}}, \tau)$-IND-ACT-adversary against Gi+SH, in the ROM, making at most $q_{\mathcal{G}}(k)$ queries to the non-programmable random oracle $\mathcal{O}_{\mathcal{G}}$ in the experiments $\mathbf{Exp}^{\mathsf{ind\text{-}act\text{-}b}}_{\mathsf{Gi+SH},\mathcal{A}}(k)$ (for $b \in \{0,1\}$ and $k \in \mathbb{N}$). There exist a τ'-COMP-CDH adversary $\mathcal{B}$ such that*

$$\begin{cases} \mathbf{Succ}^{\mathsf{comp-cdh}}_{\mathsf{Gen},\mathcal{B}}(k) \geq \mathbf{Adv}^{\mathsf{ind-act}}_{\mathsf{Gi+SH},\mathcal{A}}(k)/[2q_{\mathcal{G}}(k)] \\ \tau'(k) \leq \tau(k) + (q_{\mathcal{H}}(k) + O(1))T_{\mathrm{Exp}}(k) \end{cases}$$

for all positive integers k.

Proof. Again, we prove only the first part of the theorem. Let k be a security parameter, let (n, p, q, e, d, g) be a 6-tuple generated by $\mathsf{Gen}(k)$ and let $\alpha \in (\mathbb{Z}/n\mathbb{Z})^*$ be a random instance of the RSA problem. We construct a reduction algorithm $\mathcal{B}$ which on input (n, e, α) outputs a pair $(x, y) \in [(\mathbb{Z}/n\mathbb{Z})^*]^2$ aimed to satisfy $x^e - y^e = \alpha \bmod n$.

We follow essentially the previous proof with games Game_0 and Game_1 analogous to the corresponding games. With the same notation, we get

$$\Pr[\mathsf{Imp}_1] = \mathbf{Succ}^{\mathsf{imp-act}}_{\mathsf{GI+SH},\mathcal{A}}(k)/q_{\mathrm{CG}}(k).$$

Game_2 $\mathcal{B}$ simulates the $\mathcal{O}_{\mathrm{CG}}$ oracle. For the i-th query ($i \in [\![1, q_{\mathrm{CG}}(k)]\!] \setminus \{\ell\}$), $\mathcal{B}$ executes $\mathsf{Gen}(k)$ and gets $(n_i, p_i, q_i, e_i, d_i, g_i)$. It outputs (n_i, e_i, g_i) as the answer to $\mathcal{A}$'s i-th query for the group denoted by G_i and stores d_i. For the ℓ-th query, $\mathcal{B}$ outputs (n, e, g^{e^2}) as $G^\star$'s public key. The distribution of the group authorities keys is unchanged, and therefore, we have $\Pr[\mathsf{Imp}_2] = \Pr[\mathsf{Imp}_1]$.

Following Coron's technique [6], a random coin decides whether $\mathcal{B}$ introduces the challenge α in the hash answer to the oracle $\mathcal{O}_{\mathcal{H}}$, or an element with a known preimage. Let $\delta \in [0, 1]$.

Game_3 In this game, for each fresh bit string ID queried to $\mathcal{O}_{\mathcal{H}}$, $\mathcal{B}$ picks at random a bit $t_{\mathsf{ID}} \xleftarrow{R} B_\delta$, where B_δ is the probability distribution over $\{0,1\}$ where 0 is drawn with probability δ and 1 with probability $1 - \delta$. $\mathcal{B}$ stores in a list denoted by H_k-List a 4-tuple $(\mathsf{ID}, \mathcal{O}_{\mathcal{H}}(\mathsf{ID}), \perp, t_{\mathsf{ID}})$. $\mathcal{B}$ discards execution which outputs a triple $(G^\star, \mathsf{ID}^\star, d^\star)$ such that $t_{\mathsf{ID}^\star} = 1$ or $t_{\mathsf{ID}_{\mathcal{A}}} = 0$.
Since for each bit string ID queried to $\mathcal{O}_{\mathcal{H}}$, the bit t_{ID} is picked independently of the execution of the game Game_4, we have $\Pr[\mathsf{Imp}_3|\Lambda] = \delta(1-\delta)\Pr[\mathsf{Imp}_2|\Lambda]$.

Game_4 Now, $\mathcal{B}$ immediately aborts if $\mathcal{A}$ makes a query to $\mathcal{O}_{\mathrm{AM}}$ or $\mathcal{O}_{\mathrm{Co}}$ involving a pseudonym ID such that $t_{\mathsf{ID}} > 0$. Thus $\Pr[\mathsf{Imp}_4|\Lambda] = \delta^{q_{\mathrm{AM}}(k)+q_{\mathrm{Co}}(k)}\Pr[\mathsf{Imp}_3|\Lambda]$.

Game_5 In this game, $\mathcal{B}$ simulates the random oracle $\mathcal{O}_{\mathcal{H}}$. For each fresh bit string ID queried to $\mathcal{O}_{\mathcal{H}}$, $\mathcal{B}$ picks uniformly at random $m_{\mathsf{ID}} \in [\![1, n]\!]$. With probability p, it sets $h_{\mathsf{ID}} = m_{\mathsf{ID}}^e \cdot \alpha$ and $t_{\mathsf{ID}} = 0$; otherwise $\mathcal{B}$ picks $t_{\mathsf{ID}} \in [\![1, n]\!]$, coprime to e, and sets $h_{\mathsf{ID}} = g^{e \cdot t_{\mathsf{ID}}} - m_{\mathsf{ID}}^e$. $\mathcal{B}$ returns h_{ID} as the answer to the oracle call and stores the 4-tuple $(\mathsf{ID}, h_{\mathsf{ID}}, m_{\mathsf{ID}}, t_{\mathsf{ID}})$ in the H_k-List. In the random oracle model, this game is clearly identical to the previous one, therefore $\Pr[\mathsf{Imp}_5] = \Pr[\mathsf{Imp}_4]$.

Game_6 Now, $\mathcal{B}$ simulates the $\mathcal{O}_{\mathrm{AM}}$ and the $\mathcal{O}_{\mathrm{Co}}$ oracles.
For all $i \in [\![1, q_{\mathrm{CG}}(k)]\!] \setminus \{\ell\}$, thanks to the knowledge of d_i, $\mathcal{B}$ can perfectly simulate the member key generation protocol for the group G_i, and therefore can reply to $\mathcal{A}$'s queries involving members of G_i.
For queries $\mathcal{O}_{\mathrm{AM}}(G^\star)$, $\mathcal{B}$ picks uniformly at random a pseudonym ID and queries $\mathcal{O}_{\mathcal{H}}$ on ID, $\mathcal{B}$ act as follows:

- If $t_{\mathsf{ID}} = 0$, then $\mathcal{B}$ picks uniformly at random $s_{\mathsf{ID}} \in [\![1, n]\!]$ and queries $T_{\mathsf{ID}} = (-m_{\mathsf{ID}})^{-e} g^{e^2 s_{\mathsf{ID}}}$ to its oracle $(\alpha + (\cdot)^e)^d$. It gets the value U_{ID}, and sets $p_{\mathsf{ID}} = -U_{\mathsf{ID}} \times m_{\mathsf{ID}} \bmod n$. It outputs p_{ID} as the public key associated with $G^\star$ and ID and stores $(p_{\mathsf{ID}}, s_{\mathsf{ID}})$ in a list denoted by $\mathsf{KeyList}$.
- Otherwise, $\mathcal{B}$ outputs m_{ID} as the public key associated with $G^\star$ and ID and stores $(m_{\mathsf{ID}}, \perp)$ in a $\mathsf{KeyList}$.

The behaviour of $\mathcal{B}$ is identical for queries $\mathcal{O}_{\mathrm{AM}}(G^\star, \mathsf{ID})$ (with always $t_{\mathsf{ID}} = 0$ in this case).
Finally, for members of $G^\star$ with pseudonym ID queried to the $\mathcal{O}_{\mathrm{Co}}$ oracle, we know that $t_{\mathsf{ID}} = 0$. $\mathcal{B}$ can retrieve s_{ID} in the $\mathsf{KeyList}$. This perfectly simulates the oracles, therefore we have $\Pr[\mathsf{Imp}_6] = \Pr[\mathsf{Imp}_5]$.

Game_7 Finally, $\mathcal{B}$ can easily simulates the handshake protocols for members with pseudonyms say ID_I and ID_M. Again, the only non trivial case occurs when $t_{\mathsf{ID}_I} \neq 0$ and $t_{\mathsf{ID}_M} \neq 0$. In this case, $\mathcal{B}$ sets $K = g^{t_{\mathsf{ID}_I} \cdot t_{\mathsf{ID}_M}}$.
If we denote $k_{\mathsf{ID}_I} = \log_{g^{e^2}}(p_{\mathsf{ID}_I}^e + h_{\mathsf{ID}_I})$ and $k_{\mathsf{ID}_M} = \log_{g^{e^2}}(p_{\mathsf{ID}_M}^e + h_{\mathsf{ID}_M})$, we get $e^2 k_{\mathsf{ID}_I} = e \cdot t_{\mathsf{ID}_I} \bmod \lambda(n)$ and $e^2 k_{\mathsf{ID}_M} = e \cdot t_{\mathsf{ID}_M} \bmod \lambda(n)$, and therefore $K = g^{e^2 k_I k_M}$. Querying this value with suitable nonces to the non-programmable random oracle $\mathcal{G}_r$, $\mathcal{B}$ can perfectly simulate the oracle $\mathcal{O}_{\mathrm{HS}}$, and we have $\Pr[\mathsf{Imp}_7] = \Pr[\mathsf{Imp}_6]$.

Finally, when Game_7 terminates, $\mathcal{A}$ outputs a triple $(G^\star, \mathsf{ID}^\star, d^\star)$ such that $G^\star$'s public key is (n, e, g^{e^2}), $\mathcal{O}_{\mathcal{H}}(\mathsf{ID}^\star) = g^{e t_{\mathsf{ID}}^\star} - m_{\mathsf{ID}^\star}^e$ and the public key associated with $G^\star$ and $\mathsf{ID}^\star$ is $m_{\mathsf{ID}^\star}$. Equipped with the pseudonym $\mathsf{ID}_{\mathcal{A}}$, $\mathcal{A}$ interacts with $\mathcal{B}$ emulating the member with pseudonym $\mathsf{ID}^\star$ to execute the handshake protocol. From the simulation, we know that $\mathcal{O}_{\mathcal{H}}(\mathsf{ID}_{\mathcal{A}}) = m_{\mathsf{ID}_{\mathcal{A}}}^e \alpha$. Whatever the bit $d^\star$ and the nonces are, if the protocol is successful, thanks to the EPK property of $\mathcal{G}_r$, $\mathcal{B}$ retrieves in its transcript $p_{\mathcal{A}}$ the public key of $\mathcal{A}$ and $K = g^{e^2 s^\star s_{\mathcal{A}}}$ where $e^2 s^\star = e \cdot t_{\mathsf{ID}^\star} \bmod \lambda(n)$ and $p_{\mathcal{A}}^e - g^{e^2 s_{\mathcal{A}}} = m_{\mathsf{ID}_{\mathcal{A}}}^e \alpha$. We have $K = g^{e t_{\mathsf{ID}^\star} s_{\mathcal{A}}} = (g^{e s_{\mathcal{A}}})^{t_{\mathsf{ID}^\star}}$, and using the extended Euclidean algorithm $\mathcal{B}$ finds integers u, v such that $t_{\mathsf{ID}^\star} u + ev = 1$, and sets $x = p_{\mathcal{A}} / m_{\mathsf{ID}_{\mathcal{A}}}$ and $y = K^u (p_{\mathcal{A}}^e + m_{\mathsf{ID}_{\mathcal{A}}}^e \alpha)^v / m_{\mathsf{ID}_{\mathcal{A}}}$ such that $x^e - y^e = \alpha \bmod n$. It remains to prove the bounds on $\mathbf{Succ}_{\mathsf{Gen},\mathcal{B}}^{\mathsf{diff-rsa}}$ and τ'. The computation follows readily the one supplied in details in the previous proof. □

5.3 Security of the scheme Gi×SH

Theorem 3. *Let Gen be an RSA-group generator, let $f : \mathbb{N} \to \mathbb{N}$ and let Gi×SH be the secret handshake protocol instantiated with Gen and f.*
Let $q_{\mathcal{H}}, q_{\mathcal{G}}, q_{\mathrm{CG}}, q_{\mathrm{AM}}, q_{\mathrm{HS}}, q_{\mathrm{RM}}, q_{\mathrm{Co}}, \tau \in \mathbb{N}^{\mathbb{N}}$.

1. *Let $\mathcal{A}$ be a $(q_{\mathcal{H}}, q_{\mathrm{CG}}, q_{\mathrm{AM}}, q_{\mathrm{HS}}, q_{\mathrm{RM}}, q_{\mathrm{Co}}, \tau)$-IMP-ACT-adversary against Gi×SH, in the random oracle model. There exist a τ'-RSA adversary $\mathcal{B}$ such that*

$$\begin{cases} \mathbf{Succ}^{\mathsf{rsa}}_{\mathsf{Gen},\mathcal{B}}(k) \geq \mathbf{Succ}^{\mathsf{imp-act}}_{\mathsf{Gi\times SH},\mathcal{A}}(k)/[\exp(2) q_{\mathrm{CG}}(k)(q_{\mathrm{AM}}(k) + q_{\mathrm{Co}}(k) + 1)] \\ \tau'(k) \leq \tau(k) + (2q_{\mathcal{H}}(k) + 2q_{\mathrm{CG}}(k) + q_{\mathrm{AM}}(k) + q_{\mathrm{HS}}(k) + O(1)) T_{\mathrm{Exp}}(k) \end{cases}$$

   for all positive integers k.
2. *Let $\mathcal{A}$ be a $(q_{\mathcal{H}}, q_{\mathrm{CG}}, q_{\mathrm{AM}}, q_{\mathrm{HS}}, q_{\mathrm{RM}}, q_{\mathrm{Co}}, \tau)$-DET-ACT-adversary against Gi×SH, in the random oracle model. There exist a τ'-RSA adversary $\mathcal{B}$ such that*

$$\begin{cases} \mathbf{Succ}^{\mathsf{rsa}}_{\mathsf{Gen},\mathcal{B}}(k) \geq \mathbf{Adv}^{\mathsf{det-act}}_{\mathsf{Gi\times SH},\mathcal{A}}(k)/[2\exp(2) q_{\mathrm{CG}}(k)(q_{\mathrm{AM}}(k) + q_{\mathrm{Co}}(k) + 1)] \\ \tau'(k) \leq \tau(k) + (2q_{\mathcal{H}}(k) + 2q_{\mathrm{CG}}(k) + q_{\mathrm{AM}}(k) + q_{\mathrm{HS}}(k) + O(1)) T_{\mathrm{Exp}}(k) \end{cases}$$

   for all positive integers k.
3. *Let $\mathcal{A}$ be a $(q_{\mathcal{H}}, q_{\mathrm{CG}}, q_{\mathrm{AM}}, q_{\mathrm{HS}}, q_{\mathrm{RM}}, q_{\mathrm{Co}}, \tau)$-IND-ACT-adversary against Gi×SH, in the random oracle model, making at most $q_{\mathcal{G}}(k)$ queries to the non-programmable random oracle $\mathcal{O}_{\mathcal{G}}$ in the experiments $\mathbf{Exp}^{\mathsf{ind\text{-}act\text{-}b}}_{\mathsf{Gi\times SH},\mathcal{A}}(k)$ (for $b \in \{0,1\}$ and $k \in \mathbb{N}$). There exist a τ'-COMP-CDH adversary $\mathcal{B}$ such that*

$$\begin{cases} \mathbf{Succ}^{\mathsf{comp-cdh}}_{\mathsf{Gen},\mathcal{B}}(k) \geq \mathbf{Adv}^{\mathsf{ind-act}}_{\mathsf{Gi\times SH},\mathcal{A}}(k)/[2q_{\mathcal{G}}(k)] \\ \tau'(k) \leq \tau(k) + (q_{\mathcal{H}}(k) + O(1)) T_{\mathrm{Exp}}(k) \end{cases}$$

   for all positive integers k.

□

6 Efficiency issues

In this section, we compare the performance of all the secret handshake schemes proposed up to now. For concreteness, we assume that our schemes are instantiated with 1024-bits RSA *moduli*, and that the Diffie-Hellman protocols from [5] are instantiated on a 160-bits prime order subgroup of a prime finite field of size 1024 bits. We denote by CJT1 the scheme based on the CA-oblivious encryption and by CJT2 the scheme using the additional proof of knowledge. We assume that the scheme from [1], denoted BDSSSW, is instantiated with the Tate pairing on an elliptic curve of MOV degree 6 on a ground base field of size 171 bits and that computing this bilinear map using Miller's algorithm [12] is 10 times more expensive than computing a discrete exponentiation in a 160 bits subgroup of a prime finite field of size 1024 bits (whose computation time is arbitrarily set to 1). In the table 1, we summarize the schemes' complexity in terms of bits exchanged during the protocol Handshake (in addition to the identification information $\mathsf{ID_I}$ and $\mathsf{ID_M}$) and the computational cost of the protocols AddMember and Handshake. The new schemes compare very favorably in performance with respect to the discrete log systems proposed so far [1, 5] and they can be used over a low bandwidth channel.

Scheme	BDSSSW [1]	CJT1 [5]	CJT2 [5]	*OT-SH*	*Gi+SH*	*Gi×SH*
Underlying Problem	CBDH	CDH	CDH	RSA	Diff-RSA	RSA
Number of rounds	3	4	3	4	3	3
Bits exchanged	640	8512	6304	1344	640	640
Computational cost						
AddMember	1.8	1	1	6.4	7.4	7.4
Handshake	22	8	6	4	2	2

Table 1. Efficiency comparison

7 Authenticated key exchange from secret handshakes

An authenticated key exchange protocol enables two parties to end up with a shared secret key in a secure and authenticated manner (*i.e.* no adversary can impersonate any party during the protocol or learn any information about the value of the exchanged secret). At the end of the execution of the protocol OT-SH.Handshake (*resp.* Gi⊕SH.Handshake), the parties can set up a temporary session key for securing further communication between them:

$$\begin{aligned} \mathsf{session} &= \mathcal{G}_r\left[(\mathcal{H}_k(\mathsf{ID_M}) \cdot c_{\mathsf{M}}^{e})^{k_{\mathsf{I}}} \bmod n \| \mathsf{ID_I} \| \mathsf{ID_M} \| c_{\mathsf{I}} \| c_{\mathsf{M}} \| 2\right] \\ &= \mathcal{G}_r\left[(\mathcal{H}_k(\mathsf{ID_I}) \cdot c_{\mathsf{I}}^{e})^{k_{\mathsf{I}}} \bmod n \| \mathsf{ID_I} \| \mathsf{ID_M} \| c_{\mathsf{I}} \| c_{\mathsf{M}} \| 2\right] \end{aligned}$$

$$\left(\begin{aligned} \textit{resp. } \mathsf{session} &= \mathcal{G}_r\left[(\mathbf{pk}_{\mathsf{ID_I}}{}^{e} \oplus \mathcal{H}_k(\mathsf{ID_I}))^{\mathbf{sk}_{\mathsf{ID_M}}} \bmod n \| \mathsf{ID_I} \| \mathsf{ID_M} \| n_{\mathsf{I}} \| n_{\mathsf{M}} \| 2\right] \\ &= \mathcal{G}_r\left[(\mathbf{pk}_{\mathsf{ID_M}}{}^{e} \oplus \mathcal{H}_k(\mathsf{ID_M}))^{\mathbf{sk}_{\mathsf{ID_I}}} \bmod n \| \mathsf{ID_I} \| \mathsf{ID_M} \| n_{\mathsf{I}} \| n_{\mathsf{M}} \| 2\right] \end{aligned} \right)$$

As mentioned in [3], a number of desirable attributes of key agreement protocols have been identified:

1. **known session keys:** a protocol still achieves its goal in the face of an adversary who has learned some previous session keys.
2. **(perfect) forward secrecy:** if long-term secrets of one or more entities are compromised, the secrecy of previous session keys is not affected.
3. **unknown key-share:** entity i cannot be coerced into sharing a key with entity j without i's knowledge.
4. **key-compromise impersonation:** the knowledge of i's secret value does not enable an adversary to impersonate other entities to i.
5. **loss of information:** compromise of information that would not ordinarily be available to an adversary does not affect the security of the protocol.
6. **key control:** neither entity should be able to force the session key to a preselected value.

In these *scenarii*, the adversary controls all communication between entities, and can at any time ask an entity to reveal its long-term secret key. Furthermore she may initiate sessions between any two entities, engage in multiple sessions with the same entity at the same time, and in some cases ask an entity to enter a session with itself. In the random oracle model, the *key control* security

Property	OT-SH	Gi+SH	Gi×SH
known session keys	✔	✔	✔
forward secrecy	✔	✘	✘
unknown key-share	✔	✔	✔
key compromise impersonation	✔	✔	✔
loss of information	✔	✘	✘

Table 2. Security properties of the new AKE-SH

requirement of the privacy-preserving authenticated key exchange derived from OT-SH, Gi+SH and Gi×SH is unconditional (in the standard model, forcing a session key to a preselected value is at least as hard as breaking the one-wayness of $\mathcal{G}_r$). The other security requirements against active adversaries are summarized in table 2 (where the symbol ✘ means that the scheme does not reach this security requirement, whereas the symbol ✔ means that in the random oracle model the security requirement reduces to the composite Diffie Hellman problem). The proofs are straightforward adaptations of the proofs of security of the secret handshake protocols and therefore they are left to the reader.

8 Conclusion

A secret handshake protocol is a cryptographic primitive that allow members of a group to authenticate each other secretly. In this paper, we designed three efficient constructions for secret handshake based on the RSA assumption. These constructs are the first fully anonymous protocols relying on the RSA primitive. The new secret handshake protocols can handle roles just as easily as the one in [1, 5]. An interesting open issue is to design simple and efficient fully anonymous secret handshake schemes, relying as little as possible on random oracles (ideally without any).

Ackowledgements The author expresses his gratitude to Benoît Libert for his suggestions on improving the exposition and for pointing out some errors in an earlier draft of the paper.

References

1. D. Balfanz, G. Durfee, N. Shankar, D. K. Smetters, J. Staddon, and H. C. Wong, *Secret Handshakes from Pairing-Based Key Agreements.*, 2003 IEEE Symposium on Security and Privacy (S&P 2003), IEEE Computer Society, 2003, pp. 180–196.
2. M. Bellare and P. Rogaway, *Random Oracles are Practical: A Paradigm for Designing Efficient Protocols.*, Proceedings of the First ACM Conference on Computer and Communications Security, 1993, pp. 62–73.
3. S. Blake-Wilson, D. Johnson, and A. Menezes, *Key Agreement Protocols and their Security Analysis.*, 6th IMA International Conference on Cryptography and Coding (M. Darnell, ed.), Lect. Notes Comput. Sci., vol. 1355, Springer, 1997, pp. 30–45.

4. R. Canetti, O. Goldreich, and S. Halevi, *The Random Oracle Methodology, Revisited.*, J. Assoc. Comput. Mach. **51** (2004), no. 4, 557–594.
5. C. Castelluccia, S. Jarecki, and G. Tsudik, *Secret Handshakes from CA-Oblivious Encryption.*, Advances in Cryptology - ASIACRYPT 2004 (P. J. Lee, ed.), Lect. Notes Comput. Sci., vol. 3329, Springer, 2004, pp. 293–307.
6. J.-S. Coron, *On the Exact Security of Full Domain Hash.*, Advances in Cryptology - CRYPTO 2000 (M. Bellare, ed.), Lect. Notes Comput. Sci., vol. 1880, Springer, 2000, pp. 229–235.
7. Y. Desmedt, *Securing Traceability of Ciphertexts - Towards a Secure Software Key Escrow System.*, Advances in Cryptology - EUROCRYPT'95 (L. C. Guillou and J.-J. Quisquater, eds.), Lect. Notes Comput. Sci., vol. 921, Springer, 1995, pp. 147–157.
8. W. Diffie, P. C. van Oorschot, and M. J. Wiener, *Authentication and Authenticated Key Exchanges.*, Des. Codes Cryptography **2** (1992), no. 2, 107–125.
9. M. Girault, *Self-Certified Public Keys.*, Advances in Cryptology - EUROCRYPT'91 (D. Davies, ed.), Lect. Notes Comput. Sci., vol. 547, Springer, 1991, pp. 490–497.
10. M. Jakobsson, K. Sako, and R. Impagliazzo, *Designated Verifier Proofs and Their Applications.*, Advances in Cryptology - EUROCRYPT'96 (U. M. Maurer, ed.), Lect. Notes Comput. Sci., vol. 1070, Springer, 1996, pp. 143–154.
11. M. Mambo and H. Shizuya, *A Note on the Complexity of Breaking Okamoto-Tanaka ID-Based Key Exchange Scheme.*, First International Workshop on Practice and Theory in Public Key Cryptography, PKC '98 (H. Imai and Y. Zheng, eds.), Lect. Notes Comput. Sci., vol. 1431, Springer, 1998, pp. 258–262.
12. V. S. Miller, *The Weil Pairing, and Its Efficient Calculation.*, J. Cryptology **17** (2004), no. 4, 235–261.
13. M. Naor, *On Cryptographic Assumptions and Challenges.*, Advances in Cryptology - CRYPTO 2003 (D. Boneh, ed.), Lect. Notes Comput. Sci., vol. 2729, Springer, 2003, pp. 96–109.
14. J. B. Nielsen, *Separating Random Oracle Proofs from Complexity Theoretic Proofs: the Non-committing Encryption Case*, Advances in Cryptology - CRYPTO 2002 (M. Yung, ed.), Lect. Notes Comput. Sci., vol. 2442, Springer, 2002, pp. 111–126.
15. S.-H. Oh, M. Mambo, H. Shizuya, and D.-H. Won, *On the Security of Girault Key Agreement Protocols against Active Attacks.*, IEICE Trans. Fundamentals **E86-A** (2003), no. 5, 1181–1189.
16. E. Okamoto and K. Tanaka, *Key Distribution System Based on Identification Information*, IEEE J. Selected Areas in Communications **7** (1989), 481–485.
17. J. Rosser and L. Schoenfeld, *Approximate formulas for some functions of prime numbers*, Ill. J. Math. **6** (1962), 64–94.
18. S. Saeednia, *A Note on Girault's Self-Certified Model.*, Inf. Process. Lett. **86** (2003), no. 3, 323–327.
19. R. Sakai, K. Ohgishi, and M. Kasahara, *Cryptosystems Based on Pairings.*, Proceedings of the Symposium on Cryptography and Information Security (SCIS 2000), 2000.
20. A. Shamir, *Identity-Based Cryptosystems and Signature Schemes.*, Advances in Cryptology - CRYPTO'84 (G. R. Blakley and D. Chaum, eds.), Lect. Notes Comput. Sci., vol. 196, Springer, 1985, pp. 47–53.
21. V. Shoup, *OAEP Reconsidered.*, J. Cryptology **15** (2002), no. 4, 223–249.
22. S. Xu and M. Yung, *k-Anonymous Secret Handshakes with Reusable Credentials.*, Proceedings of the 11th ACM Conference on Computer and Communications Security (V. Atluri, B. Pfitzmann, and P. McDaniel, eds.), ACM, 2004, pp. 158–167.