Private Handshakes*

Jaap-Henk Hoepman

TNO Information and Communication Technology
P.O. Box 1416, 9701 BK Groningen, The Netherlands
jaap-henk.hoepman@tno.nl
and
Institute for Computing and Information Sciences
Radboud University Nijmegen
P.O. Box 9010, 6500 GL Nijmegen, the Netherlands
jhh@cs.ru.nl

Abstract. Private handshaking allows pairs of users to determine which (secret) groups they are both a member of. Group membership is kept secret to everybody else. Private handshaking is a more private form of secret handshaking [BDS+03], because it does not allow the group administrator to trace users. We extend the original definition of a handshaking protocol to allow and test for membership of multiple groups simultaneously. We present simple and efficient protocols for both the single group and multiple group membership case.
Private handshaking is a useful tool for mutual authentication, demanded by many pervasive applications (including RFID) for privacy. Our implementations are efficient enough to support such usually resource constrained scenarios.

1 Introduction

A secret handshake allows members of a (secret) group to identify each other, without revealing their membership to potential eavesdroppers or malicious impostors. As an informal example taken from the real world, it would allow FBI agents attending a hacker convention to recognise each other without giving away their presence to the rest of the audience[1].

Several years ago, Balfanz *et al.* [BDS+03] revived interest (e.g., [CJT05]) in the development of secure (cryptographic) protocols to implement such secret handshakes. According to them, secret handshakes are fundamentally different from *one-way accumulators* [BM93] and *private matchmaking* [BG85, Mea86, ZN] (not to be confused with distributed match making [MV88]). We show that this distinction is only superficial (depending on a particular notion of traitor tracing), and

* This research was partially funded by Sentinels project JASON (NIT.6677).
Id: secret-handshakes.tex 12 2007-06-15 10:41:52Z jhh .

[1] This, off course, is not withstanding the use of any other distinctive features to 'spot' a typical FBI agent. Moreover, in this scenario, where all people present belong essentially to just two groups, non-membership of one group 'proves' membership of the other…

that much simpler protocols, derived from the literature on matchmaking (and pretty much equivalent to one-way accumulators) serve equally well as secret handshake protocols. We call these protocols *private* handshaking protocols.

Such private handshaking protocols (that, unlike secret handshaking, do not implement traceability) are quite suitable to resource constrained environments, like low-end smart card, RFID or NFC-based[2] systems [RE03, Fin03]. Moreover, they implement a form of mutual authentication that is sorely needed in many pervasive systems [WSRE03, HHJ+06]. For instance, the privacy of a holder of an RFID tag is better protected if the reader must authenticate to the tag before the tag releases any information. A private handshaking protocol could ensure that the tag would only grant access if the reader and the tag belong to the same group.

1.1 State of the art

Private matchmaking protocols, originally studied by Baldwin and Gramlich [BG85] (and followed up upon by Zhang and Needham [ZN]), allow users that share the same 'wish' to locate and identify each other securely and privately. The canonical example used in both papers is that of matching job openings at big corporations with high-ranked managers looking for their next job opportunity. In this example a corporation will not want to publicly announce availability of a position, and similarly, a high-ranked manager will not want to reveal his or her job aspirations to everybody. The protocol of Baldwin and Gramlich [BG85] requires the presence of an on-line trusted third party. Zhang and Needham [ZN] improve on this by not using a trusted third party at all, and not using public-key cryptography either (making their protocol very light-weight).

Secret handshaking protocols, as studied by Balfanz *et al.* [BDS+03] consider membership of a secret group instead, and allow members of such groups to reliably identify fellow group members without giving away their group membership to non-members and eavesdroppers. An example of this problem was given in the introduction. Balfanz *et al.* also pose the additional requirement that a group member can choose to authenticate to other group members that have a certain *role* within that group. Furthermore, they require that group membership is revocable, and that the protocols are forward repudiable, traceable and collusion resistant (see section 2.2 for details). Their protocols are secure under the Bilinear Diffie-Hellman assumption [**?**] and the random oracle model [BR93b]. They require that each user periodically obtains fresh pseudonyms from the group administrator, for use in a handshake protocol run.

Their results were later improved by Castelluccia *et al.* [CJT05] with protocols based on CA-Oblivious encryption secure under the random oracle model and either the Computational Diffie Hellman assumption or the RSA assumption [MOV96]. Like Balfanz *et al.* , unlinkability in their protocols is achieved at

[2] RFID stands for Radio Frequency IDentification. NFC stands for Near Field Communication. See the references for more information.

the cost of an ample supply of fresh pseudonyms used one by one in every protocol run. Also, both protocols assume the existence of a group administrator that distributes group secrets to group members, and that can discover any traitors. Unfortunately, this also implies that the administrator can discover all instances of a protocol run in which a particular user participated[3]. This is clearly a strong breach of privacy.

Tsudik and Xu [TX05] extend the secret handshaking problem to more than 2 participants (but still determining shared membership of a *single* group), and present protocols solving this generalisation with reusable credentials. This removes the main drawback found in previous protocols. Xu and Yung [XY04] previously achieved a similar reusability of credentials.

Meadows [Mea86] built a matchmaking protocol without relying on an on-line trusted third party (but using public key cryptography, cf. [ZN]). Interestingly, she studied the matchmaking problem in the secret handshake setting: i.e., she considered secret group membership instead of communicating wishes. The difference between both is subtle, but important (see [BDS⁺03]): if the wish can be guessed, then (by definition of the matchmaking problem that any pair of users sharing the same wish can identify each other) the owner of that wish can be identified. Similarly, if 'secret' group names are used as input to matchmaking protocols, then anybody able to guess the group name can locate the other, real, group members, and moreover can impersonate a group member.

In a similar vain, set intersection protocols [FNP04, KS05] are subtly different from private handshaking protocols as well. Typically, the domains of the sets over which the intersection has to be computed is much smaller, and in any case, any element in the domain is a possible member. For private handshaking protocols, however, group membership is encoded by a secret value from a much larger, sparsely occupied, domain. Moreover, not all set intersection protocols require the outcome of the computation to be secret. A more thorough discussion of the relationship between secret handshaking, oblivious encryption/signatures and hidden credentials can be found in [Hol05].

1.2 Our results

We define the *private handshaking* problem as a more private form of secret handshaking [BDS+03], that does not allow a group administrator (or anyone else) to trace users running the protocol. This makes private handshaking a more private form of secret handshaking. Our model and definitions are described in Sect. 2. The main contribution of this paper is the conclusion that, when dropping traceability, much more efficient implementations of handshaking are possible. This makes such protocols viable for resource constrained environments, like RFID or NFC-based systems.

[3] In the current implementations of these protocols, this is trivial because the parties exchange pseudonyms initially distributed by the group administrator. More fundamentally, this could be achieved in full generality by running the traitor tracing protocol on a normal protocol run. By definition, this this would reveal the parties involved (provided they were members of the group).

We extend the definition of handshake protocols to handle the (much more common) case where people are members of several groups. Using existing, single-group, handshaking protocols Alice and Bob (member of a and b groups respectively) can do no better than running $a \times b$ handshake protocols in parallel to determine all the groups that they share membership of. We show that, in fact, $O(a + b)$ type protocols exist.

We then present two protocols for private handshaking, one for the case where Alice and Bob are members of a single group (Sect. 3), and another where Alice and Bob are a member of any number of groups each (Sect. 4). Both use a single Diffie-Hellman key exchange [DH76] and exchange as many hashes as the largest number of allowed group membership per user[4]. Security of the protocols relies on the Diffie-Hellman assumption [MOV96] and the random oracle assumption [BR93b].

2 Model and notation

2.1 System and adversary model

We assume a distributed system of n nodes, connected by asynchronous message passing. Nodes can be members of zero, one or more groups $G \in \mathcal{G}$. There are m different groups. We write $i \in G$ if node i belongs to group G, and $\mathcal{G}_i$ for the set of all groups to which node i belongs. We assume group membership is fixed and part of the initialisation of the system. We will discuss the ramifications of this assumption later on in Sect. 5.

The system is controlled by a Dolev and Yao [DY81] style adversary $\mathcal{A}$ that may block, delay, relay, delete, insert or modify messages. This allows him to force nodes to participate in a protocol run together with other nodes specified by the adversary[5]. The adversary may also corrupt any number of nodes in the system, read all data stored by such nodes, and participate in protocol runs "being within" such nodes. Nodes and the adversary are modelled as probabilistic polynomial-time Turing machines. We write $\mathcal{A} \in G$ if the adversary corrupted a member of group G, and $\mathcal{G}_{\mathcal{A}}$ for the set of all groups for which the adversary corrupted a node. If a node i is corrupted we write $i \in \mathcal{A}$. In this case $\mathcal{G}_i$ is assumed to be a subset of $\mathcal{G}_{\mathcal{A}}$. Uncorrupted nodes are honest.

In other words, the adversary induces a sequence of message exchanges and protocol steps called a *run*. At the start of each run, all nodes are initialised. In this phase, nodes may be given long term secret data needed to securely run the protocol. However, the adversary may subvert any number of nodes and retrieve this secret information stored by them. Finally, the adversary may force any node

[4] Balfanz *et al.* [BDS+03] argue that a Diffie-Hellman key exchange cannot be used to implement secret handshaking. Their argument however depends on the requirement that individual members of a group need to be traceable, and hence does not apply to *private* handshaking protocols.

[5] Bellare *et al.* [BR93a, BPR00] model the same adversarial power by allowing the adversary to query an infinite supply of protocol oracles.

to reveal any secret information resulting from a particular protocol exchange. Typically, this involves a session key established by the protocol.

2.2 The private handshake problem

We have the following set of requirements (cf. [BDS+03, TX05]) for a private handshake protocol run between two nodes i and j, belonging to groups $\mathcal{G}_i$ and $\mathcal{G}_j$ that returns output $\mathcal{O}_i$ to i and $\mathcal{O}_j$ to j. All statements below hold with overwhelming probability, for arbitrary adversary $\mathcal{A}$, for an arbitrary group G and nodes i, j.

correctness/safety $\mathcal{O}_i, \mathcal{O}_j \subseteq \mathcal{G}_i \cap \mathcal{G}_j$.

progress If i and j are honest and all messages exchanged between them during the run are delivered unaltered, $\mathcal{O}_i = \mathcal{O}_j = \mathcal{G}_i \cap \mathcal{G}_j$.

resistance to detection Let $j \in \mathcal{A}$ but $\mathcal{A} \notin G$. Then the adversary $\mathcal{A}$ cannot distinguish a protocol run in which it interacts with a node $i \in G$ from a run involving a simulator[6].

indistinguishability to eavesdroppers Let $i, j \notin \mathcal{A}$. Then the adversary $\mathcal{A}$ cannot determine whether $i \in G$ or $i \notin G$. This holds even if $\mathcal{A} \in G$. Note that both participants in the run need to be uncorrupted, and that the adversary does not modify[7] messages exchanged between i and j.

unlinkability Adversary $\mathcal{A}$ is unable to distinguish a protocol run involving node i from a protocol run involving a node $j \neq i$ with $\mathcal{G}_j = \mathcal{G}_i$, even when $\mathcal{G}_{\mathcal{A}} = \mathcal{G}_i$ and $\mathcal{A}$ participates in the protocol runs[8].

forward repudiability After the run, node i cannot convince another node k whether $j \in G$ or not. In other words, a run between i and j is indistinguishable from a run between i and i, for anyone except i.

Traditionally, the following two requirements are listed as well.

resistance to impersonation Let $j \in \mathcal{A}$ but $\mathcal{A} \notin G$. Then the adversary is not able to convince a node $i \in G$ that $\mathcal{A} \in G$.

non traceability The group administrator of group G is unable to link two different protocol runs involving the same node $i \in G$.

---

[6] Note how this requirement subtly circumvents the problem that the adversary *does* learn non-membership of i of the groups it is itself a member of (by corruption or otherwise).

[7] The powers of the adversary are limited to eavesdropping in this case. Clearly, an active adversary belonging to the same group as i can stage a man-in-the-middle attack and determine membership of G for i just like a legitimate node j could.

[8] The statement of this requirement is a bit involved because technically, an adversary can distinguish different nodes from the groups they are a member of, if the adversary itself is a member of those groups and if it participates in the runs. Intuitively, the requirement simply says that protocol runs do not carry node identifiers or similar.

However, resistance to impersonation is actually implied by correctness and the definition of $\mathcal{G}_i$ when i is corrupted. And non-traceability is equivalent to unlinkability if the group administrator is missing (or considered to be a normal, corruptible, node). We therefore omit these requirements from the list.

We refrain from imposing a fairness requirement (cf. [BDS+03]) which would require $\mathcal{O}_i = \mathcal{O}_j$ always. Fairness can be guaranteed, but at the expense of running a complex fair exchange type protocol.

Similarly, we do not require the protocol participants to set up a shared session key to be used whenever mutual authentication was successful. The protocols we present, however, do establish such a shared key.

Finally, we note that Meadows [Mea86] stipulates that an adversary that has stolen a secret from a group member cannot find out membership of the someone else without at least revealing group membership. This is similar to the resistance to impersonation requirement, when fairness is guaranteed. Otherwise, it will only hold when the adversary initiates the handshake.

3 Single membership protocols

We first present a protocol to determine shared membership of a single group. This protocol is basically a Diffie Hellman key exchange using a secret generator s as the group secret, and using the key validation phase as group membership test. The validated key can be discarded or used for secure communication between the authenticated parties. In fact, the protocol is very similar to SPEKE [Jab96], and Meadows [Mea86] basic protocol idea (but without exchanging the secret session key in the clear, instead using a key verification round as in [BPR00]).

3.1 Security proof

The following lemmas prove that protocol 1 implements private handshaking. We only sketch the proofs. Consider an arbitrary run between two nodes i and j, belonging to groups $\mathcal{G}_i = \{G_a\}$ and $\mathcal{G}_j = \{G_b\}$ where i returns output $\mathcal{O}_i$ and j returns output $\mathcal{O}_j$. Let $\mathcal{A}$ be an arbitrary adversary, and let G be an arbitrary group. A property holds with overwhelming probability if it holds with probability larger than $1 - 1/2^\sigma$, where σ is the security parameter. It holds with negligible probability if the probability is less than $1/2^\sigma$.

Lemma 3.1 (correctness/safety). *$\mathcal{O}_i, \mathcal{O}_j \subseteq \mathcal{G}_i \cap \mathcal{G}_j$ with overwhelming probability.*

Proof. Clearly the protocol ensures $\mathcal{O}_i \subseteq \mathcal{G}_i$. We have $G_a \in \mathcal{O}_i$ when $h_5(u^x) = h_5(v^y)$. This happens only, with overwhelming probability, when $u^x = v^y$, in other words $(s_b^y)^x = (s_a^x)^y$. This holds only with overwhelming probability when $s_a = s_b$. □

Lemma 3.2 (progress). *If i and j are honest and all messages exchanged between them during the run are delivered unaltered, then $\mathcal{O}_i = \mathcal{O}_j = \mathcal{G}_i \cap \mathcal{G}_j$.*

Alice		Bob
Alice		**Bob**
group G_a		group G_b
group secret s_a		group secret s_b
(or random if none)		(or random if none)
Exchange		
pick random x		pick random y
	$\xrightarrow{(s_a)^x}$	receive v
receive u	$\xleftarrow{(s_b)^y}$	
Key validation		
	$\xrightarrow{h_4(u^x)}$	receive m
		if $m = h_4(v^y)$
		then $\mathcal{O}_b = \{G_b\}$
		else $\mathcal{O}_b = \emptyset$
receive m'	$\xleftarrow{h_5(v^y)}$	
if $m' = h_5(u^x)$		
then $\mathcal{O}_a = \{G_a\}$		
else $\mathcal{O}_a = \emptyset$		
$k := h_3(u^x)$		$k := h_3(v^y)$

Fig. 1. Message flow of the single membership private handshaking protocol.

Proof. This is easily verified by case analysis. □

Lemma 3.3 (resistance to detection). *Let $j \in \mathcal{A}$ but $\mathcal{A} \notin G$. Then the adversary $\mathcal{A}$ cannot distinguish a protocol run in which it interacts with a node $i \in G$ from a run involving a simulator with non-negligible probability.*

Proof. The adversary has to distinguish s_a^x from g^z given f^x for f known to the adversary, where x is fresh, random and unknown to the adversary. Moreover, s_a is unknown to the adversary (but it may know many s_a^y, for fresh and unknown y, from previous protocol runs). Distinguishing this would violate the Diffie-Hellman assumption. □

Lemma 3.4 (indistinguishability to eavesdroppers). *Let $i, j \notin \mathcal{A}$. Then the adversary $\mathcal{A}$ cannot determine whether $i \in G$ or $i \notin G$ with non-negligible probability. This holds even if $\mathcal{A} \in G$.*

Proof. Similar to the proof of the previous lemma. □

Lemma 3.5 (unlinkability). *Adversary $\mathcal{A}$ is unable to distinguish a protocol run involving node i from a protocol run involving a node $j \neq i$ with $\mathcal{G}_j = \mathcal{G}_i$, even when $\mathcal{G}_{\mathcal{A}} = \mathcal{G}_i$ and $\mathcal{A}$ participates in the protocol runs.*

Proof. Nodes i and j share the same state. Hence all messages sent by i could have been sent by j as well. □

Lemma 3.6 (forward repudiability). *After the run, node i cannot convince another node k whether $j \in G$ or not.*

Proof. Because i is a member of G, it can construct a valid protocol run between i and j all by himself, without j participating at all. □

4 Arbitrary membership protocols

It is possible to use the single membership protocol to determine all groups of which both Alice and Bob are a member, by running the previous protocol for all candidate pairs separately. However, if Alice is a member of a groups and Bob is a member of b groups, this requires $a \times b$ message exchanges (and more if the number of groups one is a member of should not be revealed). In this section we describe a more efficient protocol (see Protocol 2), which does *not* provide traitor tracing.

Suppose each user can be a member of at most m groups. Each group is identified by a group secret (which, essentially, is a random value). Each user A that is a member of a group stores its group secret in an array $s_A[]$. Any remaining cells in the array are filled with random values (not corresponding to groups). The array is randomly permuted after initialisation[9]. After establishing a shared secret session key k using a Diffie-Hellman key exchange, Alice and Bob exchange keyed hashes h_k and h'_k of each group secret. Real implementations should use HMAC [BCK96]. Alice stores the hashes it receives in a set H_B, looks for entries in $s_A[]$ whose hash occurs in H_B, and adds those as common group members to G_A.

Note that Alice needs to use a hash function different from the one used by Bob, in order to avoid detection of shared membership by eavesdroppers. If Alice wishes not to reveal membership of certain groups, she can replace the corresponding group secret with a random value. However, Bob cannot avoid revealing his membership of those groups (unless he decides to do so independently from Alice).

4.1 Security proof

The following lemmas prove that protocol 2 implements private handshaking for multiple group. We sketch the proofs of the lemmas. Consider an arbitrary run between two nodes i and j, belonging to groups $\mathcal{G}_i$ and $\mathcal{G}_j$ where i returns output $\mathcal{O}_i$ and j returns output $\mathcal{O}_j$ (where we treat the group secrets $s_i[x]$ to represent their respective groups). Let $\mathcal{A}$ be an arbitrary adversary, and let G be an arbitrary group. A property holds with overwhelming probability if it holds with probability larger than $1 - 1/2^{\sigma}$, where σ is the security parameter. It holds with negligible probability if the probability is less than $1/2^{\sigma}$.

[9] If not, Bob might be able to infer the number of groups of which Alice is a member from the fact that the x-th token happens to coincide with a group he himself is a member of.

Alice		**Bob**
generator g		generator g
group secrets $s_a[1,\ldots,m]$ (randomly permuted)		group secrets $s_b[1,\ldots,m]$ (randomly permuted)
Exchange		
pick random x		pick random y
	$\xrightarrow{g^x}$	receive v
receive u	$\xleftarrow{g^y}$	
$k := h(u^x)$		$k := h(v^y)$
Group membership		
	$\xrightarrow{h_k(s_a[0]),\ldots,h_k(s_a[m])}$	receive into H_a
receive into H_b	$\xleftarrow{h'_k(s_b[0]),\ldots,h'_k(s_b[m])}$	
$\mathcal{O}_a = \{s_a[i] \mid h'_k(s_a[i]) \in H_b\}$		$\mathcal{O}_b = \{s_b[i] \mid h_k(s_b[i]) \in H_a\}$

Fig. 2. Message flow of the generalised private handshaking protocol.

Lemma 4.1 (correctness/safety). *$\mathcal{O}_i, \mathcal{O}_j \subseteq \mathcal{G}_i \cap \mathcal{G}_j$ with overwhelming probability.*

Proof. Clearly $\mathcal{O}_i \subseteq \mathcal{G}_i$. If $x \in \mathcal{O}_i$ then also $h'_k[x] \in H_j$. Hence $h'_k(x) = z$ for some z received in the second phase of the protocol. If z is not sent by j, then k is unknown to the adversary. Hence the chances that $h'_k(x) = z$ are negligible. If z is sent by j then $z = h'_k(s_j[y])$ for some y. This happens with overwhelming probability if $x = s_j[y]$ and hence $x \in \mathcal{G}_j$. □

Lemma 4.2 (progress). *If i and j are honest and all messages exchanged between them during the run are delivered unaltered, then $\mathcal{O}_i = \mathcal{O}_j = \mathcal{G}_i \cap \mathcal{G}_j$.*

Proof. This is easily verified by case analysis. □

Lemma 4.3 (resistance to detection). *Let $j \in \mathcal{A}$ but $\mathcal{A} \notin G$. Then the adversary $\mathcal{A}$ cannot distinguish a protocol run in which it interacts with a node $i \in G$ from a run involving a simulator with non-negligible probability.*

Proof. Since $j \in \mathcal{A}$, the adversary does know the shared session key k derived using the Diffie-Hellman key exchange. However, since $\mathcal{A} \notin G$, it does not know the secret $s_i[x]$ for group G. Hence it cannot tell whether $h_k(s_i[x])$ and $h_{k'}(s_i[x])$ are hashes for the same group exchanged during different sessions, or if these hashes correspond to different groups. This holds even if the adversary knows k' for the other session as well. □

Lemma 4.4 (indistinguishability to eavesdroppers). *Let $i, j \notin \mathcal{A}$. Then the adversary $\mathcal{A}$ cannot determine whether $i \in G$ or $i \notin G$ with non-negligible probability. This holds even if $\mathcal{A} \in G$.*

Proof. If $i, j \notin \mathcal{A}$, then the adversary does not know the shared session key k derived using the Diffie-Hellman key exchange. With a fresh, unknown, random key k, the keyed hash value $h_k(s_i[x])$ corresponding to the secret for group G is indistinguishable from a random value, even if the adversary knows $s_i[x]$. □

Lemma 4.5 (unlinkability). *Adversary $\mathcal{A}$ is unable to distinguish a protocol run involving node i from a protocol run involving a node $j \neq i$ with $\mathcal{G}_j = \mathcal{G}_i$, even when $\mathcal{G}_{\mathcal{A}} = \mathcal{G}_i$ and $\mathcal{A}$ participates in the protocol runs.*

Proof. Nodes i and j share the same state. Hence all messages sent by i could have been sent by j as well. □

Lemma 4.6 (forward repudiability). *After the run, node i cannot convince another node k whether $j \in G$ or not.*

Proof. Because i is a member of G, it can construct a valid protocol run between i and j all by himself, without j participating at all. □

5 Conclusions

We have presented two efficient protocols for secret handshaking. The second protocol efficiently supports membership of more than one group. The focus in this work is the efficiency of the protocols. They use only a few, quite simple, operations. This may allow the implementation of these protocols on resource constrained devices, like perhaps higher-end RFID tags. It is especially in these kinds of environments that a form of mutual authentication is required to provide a certain level of security and/or privacy.

Our protocols do not allow for easy revocation of group membership: all remaining members need to be given a new, fresh, group secret. More efficient ways to support group membership revocation are an interesting topic for further research, especially given the requirement that the resulting protocols should still be efficient and should not allow a group adminstrator to trace users. We also wish to develop more formal proofs for the security of our protocols.

Two other possible extensions of the basic pairwise private handshake are left for further investigation. First of all, one could consider a private group handshake where a subgroup of a secret group can recognise membership of the same group simultaneously (e.g., when setting up a meeting). Secondly, one could create password based private handshakes by using the original idea of Jablon [Jab96] based on a passkey shared by the members of the group.

We thank Flavio D. Garcia, David Galindo and Berry Schoenmakers for fruitful discussions on this topic, and the anonymous referees for their very insightful comments and suggestions.

References

[BG85] Baldwin, R. W., and Gramlich, W. C. Cryptographic protocol for trustable match making. In *IEEE Security & Privacy* (Oakland, CA, USA, 1985), IEEE, pp. 92–100.

[BDS+03] Balfanz, D., Durfee, G., Shankar, N., Smetters, D. K., Staddon, J., and Wong, H. C. Secret handshakes from pairing-based key agreements. In *IEEE Security & Privacy* (Oakland, CA, USA, 2003), IEEE, pp. 180–196.

[BCK96] Bellare, M., Canetti, R., and Krawczyk, H. Keying hash functions for message authentication. In *CRYPTO '96* (Santa Barbara, CA, USA, 1996), N. Koblitz (Ed.), LNCS 1109, Springer, pp. 1–15.

[BPR00] Bellare, M., Pointcheval, D., and Rogaway, P. Authenticated key exchange secure against dictionary attacks. In *EUROCRYPT 2000* (Bruges, Belgium, 2000), B. Preneel (Ed.), LNCS 1807, Springer, pp. 139–155.

[BR93a] Bellare, M., and Rogaway, P. Entity authentication and key distribution. In *CRYPTO '93* (Santa Barbara, CA, USA, 1993), D. R. Stinson (Ed.), LNCS 773, Springer, pp. 232–249.

[BR93b] Bellare, M., and Rogaway, P. Random oracles are practical: A paradigm for designing efficient protocols. In *1st CCS* (Fairfax, VA, USA, 1993), ACM, pp. 62–73.

[BM93] Benaloh, J., and Mare, M. de. One-way accumulators: A decentralized alternative to digital signatures. In *EUROCRYPT '93* (Lofthus, Norway, 1993), T. Helleseth (Ed.), LNCS 765, Springer, pp. 274–285.

[CJT05] Castelluccia, C., Jarecki, S., and Tsudik, G. Secret handshakes from ca-oblivious encryption. In *10th ASIACRYPT* (Jeju Island, Korea, 2005), P. Lee (Ed.), pp. 293–307.

[DH76] Diffie, W., and Hellman, M. E. New directions in cryptography. *IEEE Trans. Inf. Theory* **IT-11** (1976), 644–654.

[DY81] Dolev, D., and Yao, A. On the security of public-key protocols. In *22nd FOCS* (Nashville, TN, USA, 1981), IEEE Comp. Soc. Press, pp. 350–357.

[Fin03] Finkenzeller, K. *RFID-Handbook*, 2nd ed. Wiley & Sons, 2003.

[FNP04] Freedman, M., Nissim, K., and Pinkas, B. Efficient private matching and set intersection. In *EUROCRYPT 2004* (Interlaken, Switzerland, 2004), C. Cachin and J. Camenisch (Eds.), LNCS 3027, Springer, pp. 1–19.

[HHJ+06] Hoepman, J.-H., Hubbers, E., Jacobs, B., Oostdijk, M., and Wichers Schreur, R. Crossing borders: Security and privacy issues of the european e-passport. In *1st IWSEC* (Kyoto, Japan, 2006), H. Yoshiura, K. Sakurai, K. Rannenberg, Y. Murayama, and S. Kawamura (Eds.), LNCS 4266, Springer, pp. 152–167.

[Hol05] Holt, J. E. Reconciling ca-oblivious encryption, hidden credentials, osbe and secret handshakes. Cryptology ePrint Archive, Report 2005/215, 2005. `http://eprint.iacr.org/`.

[Jab96] Jablon, D. P. Strong password-only authenticated key exchange. *Comput. Comm. Rev.* (1996). `www.integritysciences.com` and `http://www.std.com/~dpj`.

[KS05] Kissner, L., and Song, D. Privacy-preserving set operations. In *CRYPTO 2005* (Santa Barbara, CA, USA, 2005), V. Shoup (Ed.), LNCS 3612, Springer, pp. 241–257.

[Mea86] Meadows, C. A more efficient cryptographic matchmaking protocol for use in the absence of a continuously available third party. In *IEEE Security & Privacy* (Oakland, CA, USA, 1986), IEEE, pp. 134–137.

[MOV96] Menezes, A. J., Oorschot, P. C. van, and Vanstone, S. A. *Handbook of Applied Cryptography*. CRC Press, 1996.

[MV88] Mullender, S. J., and Vitányi, P. M. B. Distributed match-making. *Algorithmica* **3** (1988), 367–391.

[RE03] Rankl, W., and Effing, W. *Smart Card Handbook*, 3rd ed. Wiley & Sons, 2003.

[TX05] Tsudik, G., and Xu, S. Flexible framework for secret handshakes (multi-party anonymous and un-observable authentication). Cryptology ePrint Archive, Report 2005/034, 2005. http://eprint.iacr.org/.

[WSRE03] Weis, S. A., Sarma, S. E., Rivest, R. L., and Engels, D. W. Security and privacy aspects of low-cost radio frequency identification systems. In *1st SPC* (Boppard, Germany, 2003), D. Hutter, G. Müller, W. Stephan, and M. Ullmann (Eds.), LNCS 2802, Springer, pp. 201–212.

[XY04] Xu, S., and Yung, M. k-anonymous secret handshakes with reusable credentials. In *11th CCS* (Washington D.C., USA, 2004), V. Atluri, B. Pfitzmann, and P. D. McDaniel (Eds.), ACM, pp. 158–167.

[ZN] Zhang, K., and Needham, R. A private matchmaking protocol. http://citeseer.nj.nec.com/71955.html.