Private Discovery of Common Social Contacts*

Emiliano De Cristofaro[1], Mark Manulis[2], Bertram Poettering[2]

[1] University of California, Irvine
edecrist@uci.edu

[2] Cryptographic Protocols Group, TU Darmstadt & CASED, Germany
mark@manulis.eu, bertram.poettering@cased.de

Abstract

The increasing use of computing devices for social interactions propels the proliferation of online social applications, yet, it prompts a number of privacy concerns. One common problem occurs when two unfamiliar users, in the process of establishing social relationships, want to assess their social proximity by discovering mutual contacts. In this paper, we introduce *Private Contact Discovery*, a novel cryptographic primitive that lets two users, on input their respective contact lists, learn their common contacts (if any), and nothing else. We present an efficient and provably secure construction, that (i) prevents arbitrary list manipulation by means of contact certification, and (ii) guarantees user authentication and revocability. Following a rigorous cryptographic treatment of the problem, we define the privacy-protecting *contact-hiding* property and prove it for our solution, under the RSA assumption in the Random Oracle Model (ROM). We also show that other related cryptographic techniques, such as Private Set Intersection and Secret Handshakes, are unsuitable in this context. Experimental analysis attests to the practicality of our technique, which achieves computational and communication overhead (almost) linear in the number of contacts.

1 Introduction

The increasing volume of electronic social interactions motivates the need for efficient privacy-enhancing techniques. One interesting problem occurs when two unfamiliar users want to *privately* discover their common contacts: in doing so, they would like to reveal to each other only the contacts that they share.

We focus, for instance, on the discovery of mutual social-network *friends*. Online social networks provide friends with services to share interests, activities, or pictures, and help them build and reflect social relations. Popular social network websites, such as Facebook, Linkedin, or MySpace, involve millions of active users, who access their services *ubiquitously* — e.g., 250 of 500 million Facebook users access it from their mobile devices [39]. Other projects, such as Nokia's Awarenet [1], aim at letting users in physical proximity interact using their mobile phones, without relying on a central server or using an Internet connection.

One of the first steps toward establishing social-network relationships is to verify the existence of *common* friends. Consider the following settings: (1) a social network user wants to extend her network and is willing to establish friendships with other users with whom she has mutual friends; (2) a mobile phone user would like to interact with other users in physical proximity (e.g., in a bar or on the subway), given that they have some common friends on a

*An earlier version of this paper appeared in the Proceedings of 9th International Conference on Applied Cryptography and Network Security (ACNS '11).

given social network, e.g., Facebook. One crucial problem, in these scenarios, is discovering mutual friends in a *privacy-preserving* manner.

A naïve solution would require users to reveal their friends to each other. Clearly, this would not preserve users' privacy, since their complete lists would be exposed. Another trivial solution would employ and trust a central server to find and output the common friends. However, a central server is not necessarily trusted and not always available. Also, such a server would learn not only users' friends, but also which users become friends, how, when, and where. For instance, in scenario (2) above, mobile phone users might be willing to discover their mutual friends on a social network (e.g., Facebook) but may not be connected to the Internet or they may want to operate outside the social network website.

In order to protect privacy, we are faced with a couple of fundamental issues. First, we need to prevent a malicious user from manipulating her list of friends, e.g., by populating it with her best guesses of other user's list to maximize the amount of information learned, or by "impersonating" unwarranted friendships. Then, as friend relationships may vary over time, we need an efficient mechanism allowing to *revoke* friendships.

In this paper, we introduce the concept of **Private Contact Discovery**, a novel general construct geared to preserve user privacy, not only in social network interactions, but also in any other application that uses personal contact lists. We design a cryptographic primitive involving two users, e.g., Alice and Bob, on input their contact lists, that outputs only the list of mutual contacts (if any). The protocol prevents users from claiming unwarranted friendships by introducing *contact certification*. For instance, in order to include Carol in her contact list, Alice needs to obtain a certificate from Carol attesting this friendship. Then, when Alice interacts with Bob, not only the entries in her contact list are hidden from Bob, but also the possession of corresponding certificates with respect to non-common friends (and vice-versa). Note that, in our solution, these certificates are specific to individual users, i.e. malicious transfer of certificates, e.g. to enable others to claim unwarranted friendship, is impossible. Our protocol does not require any trusted server nor is bound to a specific network infrastructure, and can be used in both centralized and distributed environments.

1.1 Private Contact Discovery with Available Tools?

The problem of Private Contact Discovery bears some resemblance with several cryptographic constructs. We review them below and discuss why they are inappropriate for the problem of Private Contact Discovery.

Private Set Intersection (PSI). PSI techniques, e.g. [2, 13–15, 20, 23, 24, 29, 30], allow two parties to compute the intersection of their input sets, such that they learn nothing beyond the intersection (and the set sizes). However, PSI does not prevent parties from manipulating their inputs, thus, in the context of contact discovery, it would not prevent users from claiming unwarranted friendships.

Authorized PSI (APSI). APSI [14,15] extends PSI by ensuring that inputs are authorized by an appropriate Certification Authority (CA). Thus, unless they hold authorizations on their inputs (typically, in the form of digital signatures), parties do not learn whether the corresponding input belongs to the set intersection. Similarly, Private Intersection of Certified Sets [10] allows a trusted CA to ensure that all protocol inputs in PSI are valid and bound to each protocol participant. Note, however, that these constructs involve one single CA, whereas, every user in the context of Contact Discovery would have to act as an independent "CA" for her contacts. Also, one may think that the social network provider could certify friendships, but such a solution would incur a fundamental problem. In fact, in APSI, authorizations are signatures: assuming that Alice and Bob are both friends with Carol, Alice would have a signature on a message in the form of "Carol→Alice", while Bob would have one on "Carol→Bob". Therefore, since (Authorized) Private Set Intersection techniques only output matching elements, we cannot use them

for privately discovering common contacts (as messages representing common friendships would not match).

Secure Two-Party Computation. Generic secure two-party computation, e.g. [21, 32, 42], allows two parties with respective private inputs x and y to compute a functionality $f(x, y) = (f_1(x, y), f_2(x, y))$ such that: one party obtains $f_1(x, y)$ and the other receives $f_2(x, y)$, while neither learns more than its own input and the output. The functionality underlying Private Contact Discovery would require malicious model and could, possibly, be expressed and solved using general secure two-party computation techniques. However, the expected computational and communication overhead would be too large (possibly thousands of rounds of interaction) and thus not be applicable in practice.

Anonymous Credentials (AC). AC schemes, e.g. [7,9], allow a provider to issue to a user an anonymous credential on various attributes. The user can then prove to a third party that she possesses valid credentials issued by that provider, yet without revealing further information about credentials and attributes. AC schemes do not seem to offer an immediate solution to Private Contact Discovery. Indeed, one could think that user's friends may act as providers issuing friendship credentials, however, AC proofs would disclose information about credential issuers. For the same reason, Credential-Authenticated Key Exchange [8] does not provide an immediate solution to the Private Contact Discovery problem.

Affiliation-Hiding Authentication (AHA). AHA protocols, also called *Secret Handshakes* (SH) [3, 11, 26, 28, 41], allow two parties with membership credentials issued by the same organization, called *Group Authority* (GA), to *privately* authenticate each other. Specifically, one party can prove to the other that it has a valid credential, yet this proof hides the identity of the issuing organization, unless the other party also has a valid credential from the same organization. Some protocols [6, 27, 33] efficiently support *multiple* credentials, i.e, multiple Group Authorities, and are more closely related to the Contact Discovery problem. Specifically, Jarecki and Liu [27] introduced a multiple-credential Affiliation-Hiding Authentication scheme, with overhead (almost) linear in the number of credentials, secure under GapDH assumption. Recently, Manulis, Pinkas, and Poettering [33] proposed another efficient multiple-credential AHA scheme, secure under RSA assumption. One could think that Private Contact Discovery can be solved using efficient multiple-credential AHAs. For instance, every user could act as a GA and issue credentials as a contact certification: whenever two users want to discover whether or not they have common contacts, they execute AHA on input their credentials. However, this approach would incur several problems. In fact, multiple-credential AHA schemes, such as [27,33], assume that GAs are unconditionally trusted and always follow protocol specification. While this assumption might be realistic in classic AHA scenarios (where GAs are courts or investigation agencies), it is not reasonable, in the context of Contact Discovery, to trust all users, e.g., of a social network. Consider the case of [27]: in the process of obtaining credentials from GAs, users need to surrender all their secret keys which are not dependent on the specific group but are valid for all of them. If Eve certifies Bob to be her friend, she would obtain Bob's secret keys, thus, she would be able to impersonate Bob and/or test Bob's friendship with other users. Although recent results in [35, 36] relax some of the trust assumptions on GAs in AHA protocols, it is not clear how to efficiently extend them to the multiple-credential setting.

Friend-of-Friend. Prior work has attempted to solve problems similar to the one considered in this paper. Von Arb et al. [40] present a mobile social networking platform which enables *Friend-of-Friend* (FoF) detection in physical proximity. Matching of friend lists is provided using PSI techniques [25, 29]. As discussed earlier, this approach fails to effectively guarantee privacy, as contact lists can be artificially expanded. Freedman and Nicolosi [19] propose two solutions for the FoF problem, in the context of trust establishment in email whitelisting. One solution is based on hash functions and symmetric encryptions, the other on bilinear maps. Both solutions leverage friendship attestation but do not support user revocation — a necessary

requirement in our context. Also note that, as opposed to our protocol: (1) their solution based on symmetric encryption allows users to maliciously transfer attestations to other users, and (2) their technique using bilinear maps is inefficient, as it involves a quadratic number of bilinear map operations.

Non-Cryptographic Techniques. Besides the work focusing on protecting privacy by means of cryptographic techniques, some solutions targeting the discovery of common contacts have been proposed in different and broader contexts. Some techniques address the Friend-of-Friend problem with none or unclear privacy properties [12, 31]. Other solutions analyze, to a higher extent, social relationships, without focusing on privacy. For instance, [38] uses random walks to discover *communities* in large social-network graphs, [43, Chapter 12] formalizes the problem of dynamically identifying core communities (i.e., sets of entities with frequent and consistent interactions), [44] builds a prediction model to identify certain social structures, e.g., friendship ties and family circles, while [17] attempts at identifying communications that substantiate social relationship types.

Remark 1. Private Contact Discovery can be used as an important building block for privacy-preserving social interactions. Indeed, although prior work has focused on privacy concerns in this context, we highlight the need for a cryptographic treatment of them to obtain clear guarantees. This includes formal definition of privacy goals and design of provably secure and practical solutions. Also, Günther, Manulis, and Strufe [22] recently proposed a cryptographic model and solutions for Private User Profiles, another building block for privacy in social interactions.

1.2 Contribution and Organization

Our contributions are manifold: First, we define Private Contact Discovery, a novel cryptographic tool that allows two users to discover their common contacts, without leaking information on any other contacts, and without relying on any (trusted) third parties. Second, we provide rigorous privacy definitions and security model for this new notion. In particular, we define its main privacy goal called *Contact-Hiding*. Finally, we propose a very efficient solution, secure under the RSA assumption in ROM, which also supports efficient revocation. Performance analysis attests to the practicality of our protocol, which incurs almost linear computational and communication complexities in the number of alleged contacts. This efficiency stems from the use of the recent Index-Hiding Message Encoding (IHME) scheme [33, 34].

Paper Organization. After preliminaries in Section 2, we introduce Private Contact Discovery and present our solution, alongside its performance analysis, in Section 3. Next, in Section 4, we formalize the security model for Private Contact Discovery and state Contact-Hiding security of our scheme. Section 5 concludes the paper and provides an outlook into further research directions. In Appendix, we present the proof of Contact-Hiding security of our solution.

2 Preliminaries: Assumptions and Building Blocks

Definition 1 (RSA Assumption on Safe Moduli) *Let* $\mathsf{RSA\text{-}G}(\kappa')$ *be a probabilistic algorithm that outputs pairs* (N, e)*, where* $N = PQ$ *for random* κ'*-bit primes* $P \neq Q$ *such that* $P = 2P' + 1, Q = 2Q' + 1$ *for primes* P', Q'*, and* $e \in \mathbb{Z}_{\varphi(N)}$ *is coprime to* $\varphi(N)$*. The RSA-success probability of a PPT solver* $\mathcal{A}$ *is defined as*

$$\mathsf{Succ}^{\mathsf{rsa}}_{\mathcal{A}}(\kappa') = \Pr\big[(N, e) \leftarrow \mathsf{RSA\text{-}G}(\kappa');\ z \xleftarrow{\$} \mathbb{Z}_N;\ m \leftarrow \mathcal{A}(N, e, z);\ m^e = z \ (\mathrm{mod}\ N)\big].$$

The RSA assumption on Safe Moduli *states that the maximum RSA-success probability* $\mathsf{Succ}^{\mathsf{rsa}}(\kappa')$ *(defined over all PPT solvers* $\mathcal{A}$*) is negligible in* κ'*.*

One important building block of our protocol is an Index-Hiding Message Encoding (IHME) scheme, recently introduced by Manulis, Pinkas, and Poettering [33], which we review below. Definition 2 recalls the underlying concept of Index-*Based* Message Encoding (IBME), also introduced in [33]. It is an encoding technique that combines a set of *indexed* input messages, $m_1, \ldots, m_n \in \mathcal{M}$ (where $\mathcal{M}$ is a message space), into a single data structure $\mathcal{S}$. Any message can be individually recovered from $\mathcal{S}$ using its index, which is arbitrarily chosen from an index set $\mathcal{I}$, and specified at encoding-time.

Definition 2 (Index-Based Message Encoding) *An* index-based message encoding *scheme* $(\mathsf{iEncode}, \mathsf{iDecode})$ *over an index space* $\mathcal{I}$ *and a message space* $\mathcal{M}$ *consists of two efficient algorithms:*

$\mathsf{iEncode}(\mathcal{P})$**:** *On input a tuple of index/message pairs* $\mathcal{P} = \{(i_1, m_1), \ldots, (i_n, m_n)\} \subseteq \mathcal{I} \times \mathcal{M}$*, with distinct indices* $i_1, \ldots, i_n$*, this algorithm outputs an encoding* $\mathcal{S}$*.*

$\mathsf{iDecode}(\mathcal{S}, i)$**:** *On input of an encoding* $\mathcal{S}$ *and an index* $i \in \mathcal{I}$ *this algorithm outputs a message* $m \in \mathcal{M}$*.*

An index-based message encoding scheme is correct *if* $\mathsf{iDecode}(\mathsf{iEncode}(\mathcal{P}), i_j) = m_j$ *for all* $j \in \{1, \ldots, n\}$ *and all tuples* $\mathcal{P} = \{(i_1, m_1), \ldots, (i_n, m_n)\} \subseteq \mathcal{I} \times \mathcal{M}$ *with distinct indices* i_j*.*

Further, [33] defines IBME schemes that guarantee *index-hiding* security as *'Index-Hiding Message Encoding'* (IHME) schemes. Informally, an IHME scheme guarantees that no adversary, by inspecting an IBME structure $\mathcal{S}$ that encodes random messages, can learn any useful information about the deployed indices, even if she knows some of the indices and/or messages. We refer to [33] for the formal definition of the index-hiding property. Here we only recall the polynomial-based construction of perfect IHME from [33]. It is defined over $\mathcal{I} = \mathcal{M} = \mathbb{F}$ for an arbitrary finite field $\mathbb{F}$ (e.g., $\mathbb{F} = GF(p)$ as in [33]) and provides the index-hiding property in an information-theoretic sense.

$\mathsf{iEncode}(\mathcal{P})$**:** On input of $\mathcal{P} = \{(i_1, m_1), \ldots, (i_n, m_n)\} \subseteq \mathcal{I} \times \mathcal{M} = \mathbb{F}^2$, the encoding is defined as the list $\mathcal{S} = (c_{n-1}, \ldots, c_0)$ of coefficients of the polynomial $f = \sum_{k=0}^{n-1} c_k x^k \in \mathbb{F}[x]$ that interpolates all points in $\mathcal{P}$, i.e. $f(i_j) = m_j$ for all $(i_j, m_j) \in \mathcal{P}$. Note that this polynomial exists uniquely, i.e., the $\mathsf{iEncode}$ algorithm is deterministic.

$\mathsf{iDecode}(\mathcal{S}, i)$**:** On input of $\mathcal{S} = (c_{n-1}, \ldots, c_0)$ and index $i \in \mathcal{I}$, this algorithm outputs the evaluation $m = f(i) = \sum_{k=0}^{n-1} c_k i^k$ of f at position i.

Our protocol deploys the IHME scheme in a black-box way, thus proposing another application of this recent primitive. Furthermore, in our experimental analysis, we will also take into account recent optimizations regarding the improved polynomial interpolation algorithms and implementation of the IHME scheme presented in [34].

3 Private Contact Discovery

3.1 Contact Discovery: Syntax and Correctness

A Contact Discovery Scheme CDS is defined as a tuple of four algorithms and protocols:

$\mathsf{Init}(1^\kappa)$**:** This algorithm is executed once by each user U. On input of a security parameter 1^κ, it initializes internal parameters $U.\mathsf{params}$ and clears U's contact revocation list, i.e., $U.\mathsf{crl} = \emptyset$. $U.\mathsf{crl}$ is authenticated by U and will be distributed to other users (i.e., all contacts of U should have access to the up-to-date $U.\mathsf{crl}$). In contrast, $U.\mathsf{params}$ is private to U.

$\mathsf{AddContact}(U \leftrightarrow V)$**:** This is a protocol, executed between user U and user V, who wishes to become a contact of U. User U adds identity of V to her contact list. In addition, a corresponding contact certificate $\mathsf{cc}_{U \to V}$ is output to V. Note that we model contact establishment as a unidirectional process: If U should become a contact of V as well then they additionally will execute $\mathsf{AddContact}(V \leftrightarrow U)$.

$\mathsf{RevokeContact}(U, V)$**:** This algorithm is executed by user U. On input identity of V, the contact revocation list of U is updated to $U.\mathsf{crl} \leftarrow U.\mathsf{crl} \cup \{V\}$.

$\mathsf{Discover}(V \leftrightarrow V')$**:** This is an interactive algorithm (protocol), executed between users V and V', to discover common contacts. V's private input is $(\mathsf{role}, \mathsf{CL}, \mathsf{partner})$, where $\mathsf{role} \in \{\mathsf{init}, \mathsf{resp}\}$ specifies the role of the session as initializer or responder, contact list CL is a set of pairs of the form $(U, \mathsf{cc}_{U \to V})$, for some users U, and $\mathsf{partner}$ is the name/id of the supposed protocol partner. All values $\mathsf{cc}_{U \to V}$ are contact certificates previously obtained as output of $\mathsf{AddContact}(U \leftrightarrow V)$. V''s private input is $(\mathsf{role}', \mathsf{CL}', \mathsf{partner}')$, defined analogously. Further, users keep track of the state of created $\mathsf{Discover}(\mathsf{role}, \mathsf{CL}, \mathsf{partner})$ protocol sessions π through session variables that are initialized as follows: $(\pi.\mathsf{role}, \pi.\mathsf{CL}, \pi.\mathsf{partner}) \leftarrow (\mathsf{role}, \mathsf{CL}, \mathsf{partner})$, $\pi.\mathsf{state} \leftarrow \mathsf{running}$, $\pi.\mathsf{SCL} \leftarrow \emptyset$, and $\pi.\mathsf{id}$ is set to the own identity. After the protocol completes, $\pi.\mathsf{state}$ is updated to either $\mathsf{rejected}$ or $\mathsf{accepted}$. In the latter case, *shared contact list* $\pi.\mathsf{SCL}$ holds a non-empty set of user identifiers.

Definition 3 (Correctness of CDS**)** *Assume that users V and V' interact in a* $\mathsf{Discover}$ *protocol on input* $(\mathsf{role}, \mathsf{CL}, \mathsf{partner})$ *and* $(\mathsf{role}', \mathsf{CL}', \mathsf{partner}')$*, respectively. Let π and π' denote the corresponding sessions. Let* $\mathsf{CL}_\cap$ *denote the set of users (contacts) U that appear in both* CL *and* CL' *with the restriction that neither* $\mathsf{partner}$ *nor* $\mathsf{partner}'$ *are contained in the respective contact revocation lists.* CDS *scheme is* correct *if: (1) π and π' complete in the same state, which is* $\mathsf{accepted}$ *iff* $(\mathsf{role} \neq \mathsf{role}' \wedge \mathsf{CL}_\cap \neq \emptyset \wedge \mathsf{partner} = \pi'.\mathsf{id} \wedge \mathsf{partner}' = \pi.\mathsf{id})$*, and (2) if the sessions accept then* $\pi.\mathsf{SCL} = \pi'.\mathsf{SCL} = \mathsf{CL}_\cap$.

3.2 Protocol Specification

We now present our CDS construction and describe the instantiation of Init, $\mathsf{AddContact}$, $\mathsf{RevokeContact}$, and $\mathsf{Discover}$ algorithms. We assume that $\mathsf{AddContact}$ sessions among (honest) users of CDS are protected by secure channels, whereas, during $\mathsf{Discover}$, the channel does not need to be confidential. Note that many applications that would use CDS, such as social networks or further group applications, already provide an authentication infrastructure for their users, e.g., they deploy a PKI or use some password-based techniques. Such an authentication infrastructure can then be used for various types of communication, including the execution of CDS protocols. With this assumption in mind, we can now focus on the core functionality of the CDS scheme, namely the private discovery of shared contacts, for which potential attacks may be mounted by other application users, i.e., from the inside.

Let $\kappa, \kappa' \in \mathbb{N}$ denote security parameters, where κ' is polynomially dependent on κ. As a building block, our construction utilizes the $\mathsf{IHME} = (\mathsf{iEncode}, \mathsf{iDecode})$ scheme from [33] (see also Section 2), defined over the finite field $\mathbb{F} = GF(p) \cong \mathbb{Z}_p$, where p is the smallest prime number satisfying $p > 2^{2\kappa'+\kappa}$. In addition, the protocol makes use of two hash functions

$$H : \{0,1\}^* \to [0, p-1] \qquad \text{and} \qquad H^* : \{0,1\}^* \to [0, p-1],$$

modeled as random oracles. For convenience, for each $N \in \mathbb{N}$ of length $2\kappa'$, we define:

$$H_N : \{0,1\}^* \to \mathbb{Z}_N;\ x \mapsto H^*(N \,\|\, x) \bmod N.$$

The four algorithms and protocols of CDS are instantiated as follows:

$\mathsf{Init}(1^\kappa)$. The setup routine run by each user U mainly consists of the generation of safe RSA parameters. Given security parameter κ', two κ'-bit safe primes $P = 2P' + 1$ and $Q = 2Q' + 1$ are picked randomly. The RSA modulus is set to $N = PQ$, and a pair $e, d \in \mathbb{Z}_{\varphi(N)}$ is chosen s.t. $ed = 1 \pmod{\varphi(N)}$. Observe that $\varphi(N) = (P-1)(Q-1) = 4P'Q'$.
The largest element order in $\mathbb{Z}_N^\times$ is $\lambda(N) = \mathrm{lcm}(P-1, Q-1) = 2P'Q' = \varphi(N)/2$. [26] and [35] show that for half of the elements $g \in \mathbb{Z}_N^\times$ it holds that $\mathrm{ord}(g) = \lambda(N)$ and $-1 \notin \langle g \rangle$, i.e., $\mathbb{Z}_N^\times \cong \langle -1 \rangle \times \langle g \rangle$. Let $\mathsf{Init}()$ algorithm find such $g \in \mathbb{Z}_N^\times$ (e.g., by random sampling and testing) and assign $U.\mathsf{params} \leftarrow (N, e, d, g)$.
Finally, the algorithm initializes U's contact revocation list by setting $U.\mathsf{crl} \leftarrow \emptyset$.

$\mathsf{AddContact}(U \leftrightarrow V)$. In this protocol user U, on input $U.\mathsf{params} = (N, e, d, g)$ and identifier id of a user V, computes contact certificate $\mathsf{cc}_{U \to V} = (N, e, g, \sigma_V)$ with $\sigma_V = (H_N(\mathsf{id}))^d \bmod N$, i.e., the Full-Domain-Hash RSA signature [4] on id, and confidentially hands it out to V.

$\mathsf{RevokeContact}(U, V)$. User U revokes given user V by inserting V into its contact revocation list: $U.\mathsf{crl} \leftarrow U.\mathsf{crl} \cup \{V\}$. It is assumed that an up-to-date version of this list is distributed authentically to all contacts of U.

$\mathsf{Discover}(V \leftrightarrow V')$. The contact discovery protocol is executed between two users V and V' with inputs $(\mathsf{role}, \mathsf{CL}, \mathsf{partner})$ and $(\mathsf{role}', \mathsf{CL}', \mathsf{partner}')$, respectively (see Section 3.1 for a description of parameters). The protocol is specified in detail in Figure 1. Each user obtains its contact list and, for each entry in it, parses the friendship certificate (lines 2–3). Next, our protocol combines Okamoto's technique [37] for RSA-based identity-based key agreement (lines 4–5 and 18) with a special padding scheme (introduced in [16]) to hide the size of deployed RSA-moduli (lines 6–7 and 17). Note that several instances of Okamoto's protocol are run in parallel — one for each contact in contact list CL — and all transferred messages are IHME-encoded into a single structure before transmission (lines 8 and 10). Upon receiving the IMHE-encoded structure, each user, for every unrevoked certificate, decodes the messages (line 16) and removes the probabilistic padding applied in lines 6–7 (line 17). As we demonstrate in Section 3.3, values r calculated in line 18 are equal for both protocol participants, when computed for the same common contact. Confirmation messages (c_0, c_1) are derived from this value (lines 19–20), IHME-encoded in lines 23–24, and verified after the last communication round (line 28). Each common contact is then added to the SCL list (line 29). Note that contact revocation is handled in lines 15 and 22. Finally, the protocol terminates with "$\mathsf{accepted}$", unless SCL is empty, in which case "$\mathsf{rejected}$" is returned (lines 31-34).

3.3 Protocol Correctness

Suppose that users V, V' have valid contact certificates $\mathsf{cc}_{U \to V}, \mathsf{cc}_{U \to V'}$, respectively, for a shared contact U. Then $\mathsf{cc}_{U \to V} = (N, e, g, \sigma_V)$ and $\mathsf{cc}_{U \to V'} = (N, e, g, \sigma_{V'})$ for common parameter set N, e, g. In a $\mathsf{Discover}(V \leftrightarrow V')$ protocol session, V would receive $\vartheta = (-1)^{b'} g^{t'} \sigma_{V'}$ from V' (lines 5 and 17) and compute $r = (\vartheta^e / H_N(\mathsf{partner}))^{2t}$ (line 18). From $\sigma_{V'} = H_N(\mathsf{partner})^d \bmod N$ (see $\mathsf{AddContact}$ protocol) it follows that $r = g^{2ett'}$. User V' obtains the same value r by executing analogous computations (with $\mathsf{partner}'$ and t'). The protocol's correctness is now implied by IHME's correctness, and verifiable by inspection of Figure 1. The security analysis of the protocol is postponed to Section 4.3, after the specification of the security model.

3.4 Protocol Efficiency and Performance Analysis

We now discuss the efficiency of our CDS construction. We focus on the protocol $\mathsf{Discover}$ since Init is run only once per user, while $\mathsf{AddContact}$ and $\mathsf{RevokeContact}$ are executed only once for each added or removed contact, respectively, and can be performed off-line.

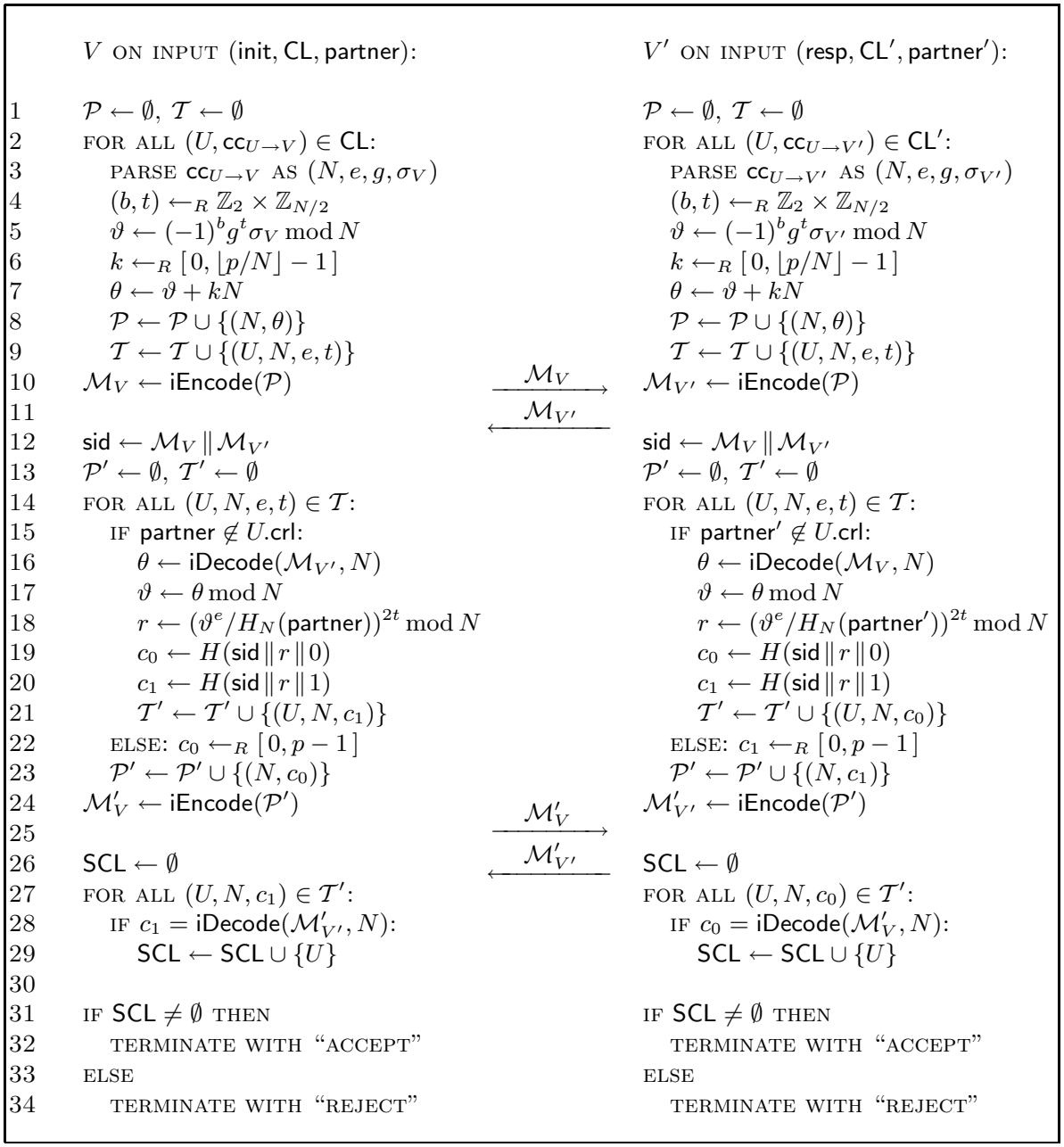
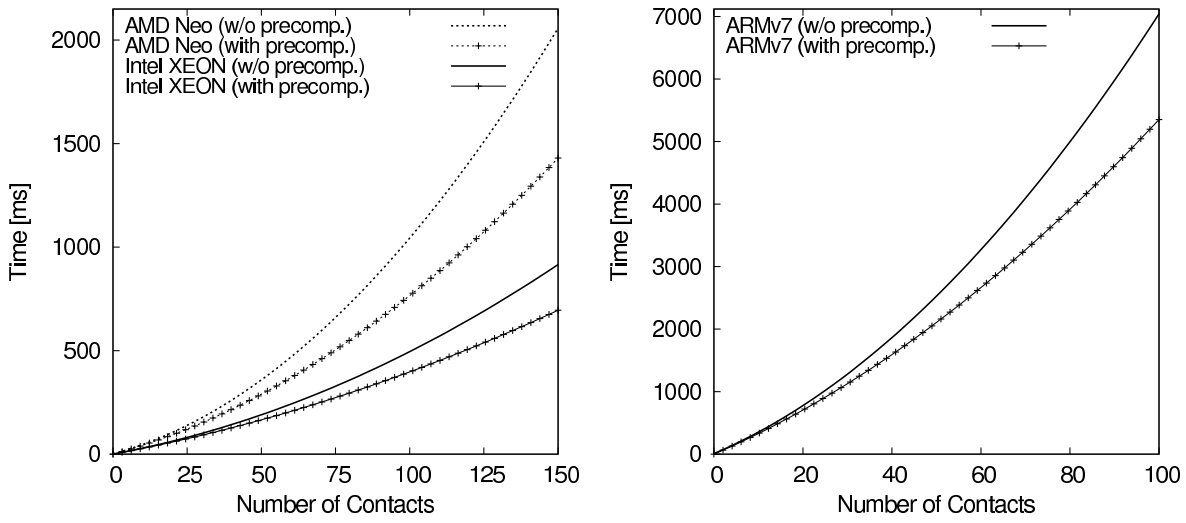


Figure 1: Specification of $\mathsf{Discover}(V \leftrightarrow V')$.

3.4.1 Computational Complexity and Bandwidth Requirements

The computational complexity of the Discover protocol is essentially related to the number of (relatively more expensive) exponentiations, executed for each contact in lines 5 and 18. Any user V needs to compute $2|\mathsf{CL}_V|$ modular exponentiations with modulus size $2\kappa'$, where $|\mathsf{CL}_V|$ denotes the number of contacts of V. If the polynomial-based IHME constructions from [33] or [34] are used to encode messages, the polynomial interpolations and evaluations only require inexpensive operations, such as multiplications in $\mathbb{F}$. Specifically, the number of multiplications in $\mathbb{F}$ would amount to $O(|\mathsf{CL}_V|^2)$ and $O(|\mathsf{CL}_V| \cdot |\mathsf{CL}_{V'}|)$, respectively, where V' denotes the protocol partner. Nevertheless, the corresponding workload can be considered small in practice, as discussed in [33, 34].

The CRL-based check for revocation of partner's pseudonym in line 15 can be implemented in logarithmic complexity (assuming sorted crls). Note that users need to keep revocation lists of



(a) AMD Neo and Intel Xeon (b) ARMv7

Figure 2: Running times of our Discover protocol on different CPUs with an increasing number of contacts. For each CPU, we consider session-independent (off-line) precomputations from [34]. All measurements are performed for 80-bit (symmetric) security and 1024-bit RSA moduli.

their contacts up-to-date. In practice, this does not impose a significant overhead, as revocation lists grow incrementally and include only (short) identifiers of revoked contacts. Thus, the related communication overhead is negligible compared to that of an actual Discover session.

The overall communication complexity of the Discover protocol (including the IHME-encoded transmission) is linear in the number of contacts. More precisely, each user sends and receives in total approximately $2(|\mathsf{CL}_V| + |\mathsf{CL}_{V'}|)(2\kappa' + \kappa)$ bits. Observe that this value can be lowered to $2(|\mathsf{CL}_V| + |\mathsf{CL}_{V'}|)(\kappa' + \kappa)$ by shortening confirmation messages c_0, c_1 to κ bits, in lines 19 and 20 (see also [34]).

3.4.2 Experimental Analysis

In addition to our asymptotic analysis, we also measured the performance of our scheme, experimentally. To this end, we conducted several experiments involving laptops and mobile devices. Our prototypes use the recent optimizations to IHME scheme, proposed by Manulis and Poettering [34].

Following [34], for $|\mathsf{CL}| < 100$, the overall costs of running the protocol is dominated by the time consumed in the exponentiations, when related to IHME encoding. If certain IHME-related precomputations from [34] are possible, then this bound increases to $|\mathsf{CL}| < 250$. Therefore, the computational overhead of our CDS construction is, in practice, almost linear in the number of contacts.

Figure 2 presents running times of our Discover protocol, using different CPUs: a single core of an Intel XEON 2.6GHz CPU, an AMD NEO 1.6GHz processor (often found in Netbook computers), and an ARMv7 600MHz CPU (installed on many today's smartphones). All measurements were performed using the GMP library [18], thus, execution on smartphones can be even speeded up using different cryptographic libraries optimized for mobile environments.

We observe that our protocol for Private Contact Discovery scales fairly well. For security level $(\kappa, 2\kappa') = (80, 1024)$, i.e., 80-bit symmetric security and 1024-bit RSA moduli, on laptops and server machines, a full protocol execution requires less than a second, even for 100 or more contacts per user. On cores with smaller footprint, e.g., on recent smartphones like Nokia's N900 (equipped with the ARMv7 600MHz processor), protocol execution with 100 contacts requires about 5 seconds, which is an acceptable overhead. Note that smartphones' CPU speeds are

envisioned to increase rapidly in the near future (e.g., the iPhone 4G is already equipped with a 1GHz processor). Finally, we computed that each user sends and receives around 300 Bytes per user of his contact list, where we assume $|\mathsf{CL}_V| = |\mathsf{CL}_{V'}|$ for simplicity. That is, in the protocol execution with 100 contacts, a total of 30KB is transmitted.

We conclude that our Private Contact Discovery solution is efficient and practical enough for actual deployment, also on smartphones widely available *today*. Yet, our technique does not give up solid privacy guarantees, as we show in the next section.

4 Security Model for Contact Discovery Protocols

In this section, we introduce our security model for a Contact Discovery Scheme (CDS). We formalize this notion by describing adversarial capabilities and defining Contact-Hiding security. Finally, we analyze the properties of our scheme with respect to this model.

4.1 Adversary Model

The adversary $\mathcal{A}$ is modeled as a PPT machine interacting with protocol participants and having access to the following set of queries, where $\mathcal{U}$ denotes the set of honest users in the system.

$\mathsf{Discover}(U, \mathsf{role}, \mathsf{CL}, \mathsf{partner})$: This query results in initiating, on behalf of user $U \in \mathcal{U}$, a new session π of Discover. Query's input is a role identifier $\mathsf{role} \in \{\mathsf{init}, \mathsf{resp}\}$, a contact list $\mathsf{CL} \subseteq \mathcal{U}$ of users, and an identifier partner of the protocol partner. Query's output is a first protocol message M (if available).

$\mathsf{Send}(\pi, M)$: With this query, message M is delivered to session π. After processing M, the output (if any) is given to $\mathcal{A}$. The query is ignored if π is not waiting for input.

$\mathsf{Reveal}(\pi)$: This query is ignored if $\pi.\mathsf{state} = \mathsf{running}$. Otherwise, the query returns ($\pi.\mathsf{state}$, $\pi.\mathsf{SCL}$).

$\mathsf{RevealCC}(V, U)$: This query gives the adversary contact certificate $\mathsf{cc}_{U \to V}$ of user V for contact U. It models the possibility of selective contact corruptions.

$\mathsf{Revoke}(U, V)$: This query lets user U include user V in its contact revocation list $U.\mathsf{crl}$.

4.2 Contact-Hiding Security

Informally, the Contact-Hiding property protects users from disclosing non-matching contacts to other participants. We model CH-security with a game, following the indistinguishability approach. The goal of the adversary is to decide which of two contact lists, CL_0^* or CL_1^*, is used by some challenge session π^*. The adversary can also invoke any number of Discover sessions, and perform Reveal and RevealCC queries at will.

Definition 4 (Contact-Hiding Security) *Let* $\mathsf{CDS} = \{\mathsf{Init}, \mathsf{AddContact}, \mathsf{RevokeContact}, \mathsf{Discover}\}$*, b be a randomly chosen bit, and* $\mathcal{Q} = \{\mathsf{Discover}, \mathsf{Send}, \mathsf{Reveal}, \mathsf{RevealCC}, \mathsf{Revoke}\}$ *denote the set of queries the adversary $\mathcal{A}$ has access to. We consider the following game between a challenger and the adversary $\mathcal{A}$:*

$\mathsf{Game}^{\mathsf{ch},b}_{\mathcal{A},\mathsf{CDS}}(\kappa, n)$:

- *The challenger creates n users, denoted by $\mathcal{U} = \{U_1, \ldots, U_n\}$. The adversary $\mathcal{A}$ specifies a set $\mathcal{U}^c \subseteq \mathcal{U}$ of initially corrupted users. Let $\mathcal{U}^h = \mathcal{U} \setminus \mathcal{U}^c$. $\mathsf{Init}(1^\kappa)$ is run for all $U \in \mathcal{U}^h$, and, for all combinations $(U, V) \in \mathcal{U}^h \times \mathcal{U}^h$, contact certificates $\mathsf{cc}_{U \to V}$ are created by respective user U and given to V, each time by running the $\mathsf{AddContact}(U \leftrightarrow V)$ protocol.*

  For all $U \in \mathcal{U}^c$, the adversary sets up all parameters himself, including $U.\mathsf{crl}$. He then specifies a list $\mathcal{L} \subseteq \mathcal{U}^h \times \mathcal{U}^c$, and for all $(U, V) \in \mathcal{L}$, protocol $\mathsf{AddContact}(U, V)$ is run, and the respective certificate $\mathsf{cc}_{U \to V}$ is given to $\mathcal{A}$;
- *$\mathcal{A}^{\mathcal{Q}}$ interacts with all (honest) users using the queries in $\mathcal{Q}$; at some point $\mathcal{A}^{\mathcal{Q}}$ outputs a tuple $(U^*, \mathsf{role}^*, \mathsf{CL}_0^*, \mathsf{CL}_1^*, \mathsf{partner}^*)$ where $U^* \in \mathcal{U}^h$, $\mathsf{role}^* \in \{\mathsf{init}, \mathsf{resp}\}$, $\mathsf{CL}_0^*, \mathsf{CL}_1^* \subseteq \mathcal{U}^h$ with $|\mathsf{CL}_0^*| = |\mathsf{CL}_1^*|$, and $\mathsf{partner}^*$ is any user id (in $\mathcal{U}$). Set $\mathcal{D}^* = (\mathsf{CL}_0^* \setminus \mathsf{CL}_1^*) \cup (\mathsf{CL}_1^* \setminus \mathsf{CL}_0^*) = (\mathsf{CL}_0^* \cup \mathsf{CL}_1^*) \setminus (\mathsf{CL}_0^* \cap \mathsf{CL}_1^*)$ is called the* distinguishing set*;*
- *the challenger invokes a $\mathsf{Discover}(U^*, \mathsf{role}^*, \mathsf{CL}_b^*, \mathsf{partner}^*)$ session π^* (and provides all needed credentials);*
- *$\mathcal{A}^{\mathcal{Q}}$ continues interacting via queries (including on session π^*) until it terminates and outputs bit b';*
- *the output of the game is b' if all of the following hold; else the output is 0:*
  - *(a) if there is a $\mathsf{Discover}$ session π' with $\mathcal{D}^* \cap \pi'.\mathsf{CL} \neq \emptyset$ and $(\pi'.\mathsf{id}, \pi'.\mathsf{partner}) = (\pi^*.\mathsf{partner}, \pi^*.\mathsf{id})$ which was in state $\mathsf{running}$ while π^* was in state $\mathsf{running}$, then neither $\mathsf{Reveal}(\pi^*)$ nor $\mathsf{Reveal}(\pi')$ was asked,*
  - *(b) for no $U \in \mathcal{D}^*$ a $\mathsf{RevealCC}(\mathsf{partner}^*, U)$ query has been posed or a pair $(U, \mathsf{partner}^*)$ is contained in $\mathcal{L}$, i.e. the adversary did not ask for a contact certificate for $\mathsf{partner}^*$ issued by any user in the distinguishing set.*

We define

$$\mathsf{Adv}^{\mathsf{ch}}_{\mathcal{A},\mathsf{CDS}}(\kappa, n) := \left| \Pr\left[\mathsf{Game}^{\mathsf{ch},0}_{\mathcal{A},\mathsf{CDS}}(\kappa, n) = 1\right] - \Pr\left[\mathsf{Game}^{\mathsf{ch},1}_{\mathcal{A},\mathsf{CDS}}(\kappa, n) = 1\right] \right|$$

and denote with $\mathsf{Adv}^{\mathsf{ch}}_{\mathsf{CDS}}(\kappa, n)$ the maximum advantage over all PPT adversaries $\mathcal{A}$. We say that CDS *is* CH-secure *if this advantage is negligible in κ (for all n polynomially dependent on κ).*

Conditions (a) and (b) exclude some trivial attacks on contact hiding. Condition (a) thwarts the attack where $\mathcal{A}$ starts a $\mathsf{Discover}(U', \mathsf{role}', \mathsf{CL}', \mathsf{partner}')$ session π' with $\mathsf{CL}' \cap \mathcal{D}^* \neq \emptyset$ and $(\pi'.\mathsf{id}, \pi'.\mathsf{partner}) = (\pi^*.\mathsf{partner}, \pi^*.\mathsf{id})$, relays all messages between π^* and π', and finally asks $\mathsf{Reveal}(\pi^*)$ or $\mathsf{Reveal}(\pi')$. By protocol correctness, $\pi^*.\mathsf{SCL} = \pi'.\mathsf{SCL}$ would contain elements from $\mathcal{D}^*$, and it would be trivial to correctly decide about b. Condition (b) prevents $\mathcal{A}$ to ask for a contact certificate issued by a user $U \in \mathcal{D}^*$ for a user $V \in \mathcal{U}$, to simulate a protocol session on behalf of V, to relay all messages between that session and π^*, and to decide about bit b from the results.

Remark 2. One may also define a stronger notion of contact-hiding by requiring that distinct sessions of the $\mathsf{Discover}$ protocol executed by the same user remain *unlinkable*. We observe that our $\mathsf{Discover}$ protocol in its plain form would not guarantee such unlinkability, since credentials (i.e., certified friendships) are re-used across multiple protocol executions, which also allows an efficient realization. Although linkable protocols may yield traceability concerns (with respect to eavesdropping adversaries) and leak sensitive information about users, we address this issue by executing the protocol over secure (encrypted and authenticated) channels. Nonetheless, it is an interesting open problem to design a $\mathsf{Discover}$ protocol, which would preserve the linear complexity of our solution and, simultaneously, achieve such stronger property of unlinkability.

4.3 Security Analysis of Our Protocol

Following the definition in Section 4.2, we argue that our CDS protocol, described in Section 3, offers Contact-Hiding security. We refer to Appendix A for the proof.

Theorem 1 (Contact-Hiding Security) *The* CDS *protocol in Section 3.2 is CH-secure under the RSA assumption on safe moduli, in Random Oracle Model (ROM).*

5 Conclusion

This paper motivated the importance, and introduced the concept of, Private Contact Discovery. Following a cryptographic treatment of the problem, we presented an efficient and provably secure construction. During protocol design, we overcame several challenges, such as the arbitrary expansion of contact lists, by using contact certification. Our solution relies on Full-Domain-Hash RSA signatures and on the recent IHME primitive [33]. We also showed, through experimental evaluation, that our solution is practical enough to be deployed in real-world applications, including those running on mobile devices.

Private Contact Discovery provides a valuable privacy-preserving tool that can serve as building block for many collaborative applications, including popular social networks. Since this work represents an initial foray into Private Contact Discovery, much remains to be done. First, we plan to extend our techniques to privately discover *communities*: consider, for example, two smartphone users in proximity willing to find out whether or not they are member of the same social community (e.g., a Facebook group or an Awarenet community [1]), in a privacy-preserving manner. Users may receive (from a community manager) credentials for community membership, and execute our CDS protocol to discover common memberships. Then, we intend to address the (privacy-preserving) discovery of i-th grade contacts and cliques [38], which currently seems impossible without relying on some trusted third party. Another interesting direction is to consider *fairness* [5] of the contact discovery process, a property that ensures a balanced gain of knowledge of protocol participants even against insider adversaries.

Acknowledgments. We are grateful to Nokia for donating the Nokia N900 devices used in our experiments. Part of this research has been supported by the US Intelligence Advanced Research Projects Activity (IARPA) under grant number FA8750-09-2-0071. M. Manulis and B. Poettering acknowledge partial support from the German Science Foundation (DFG) project PRIMAKE (MA 4957), DAAD project PACU (PPP 50743263), and BMBF project POC (AUS 10/046).

References

[1] A. Ahtiainen, K. Kalliojarvi, M. Kasslin, K. Leppanen, A. Richter, P. Ruuska, and C. Wijting. Awareness Networking in Wireless Environments: Means of Exchanging Information. In *IEEE Vehicular Technology Magazine*, pages 48–54, 2009.

[2] G. Ateniese, E. De Cristofaro, and G. Tsudik. (If) Size Matters: Size-Hiding Private Set Intersection. In *PKC*, pages 156–173, 2011.

[3] D. Balfanz, G. Durfee, N. Shankar, D. K. Smetters, J. Staddon, and H.-C. Wong. Secret Handshakes from Pairing-Based Key Agreements. In *S&P*, pages 180–196, 2003.

[4] M. Bellare and P. Rogaway. Random Oracles are Practical: A Paradigm for Designing Efficient Protocols. In *CCS*, pages 62–73, 1993.

[5] F. Boudot, B. Schoenmakers, and J. Traoré. A Fair and Efficient Solution to the Socialist Millionaires' Problem. *Discrete Applied Mathematics*, 111(1-2):23–36, 2001.

[6] R. Bradshaw, J. Holt, and K. Seamons. Concealing Complex Policies with Hidden Credentials. In *CCS*, pages 146–157, 2004.

[7] S. Brands. *Rethinking Public Key Infrastructures and Digital Certificates: Building in Privacy.* MIT Press, 2000.

[8] J. Camenisch, N. Casati, T. Groß, and V. Shoup. Credential Authenticated Identification and Key Exchange. In *CRYPTO*, pages 255–276, 2010.

[9] J. Camenisch and A. Lysyanskaya. An Efficient System for Non-transferable Anonymous Credentials with Optional Anonymity Revocation. In *EUROCRYPT*, pages 93–118, 2001.

[10] J. Camenisch and G. M. Zaverucha. Private Intersection of Certified Sets. In *Financial Cryptography*, pages 108–127, 2009.

[11] C. Castelluccia, S. Jarecki, and G. Tsudik. Secret Handshakes from CA-Oblivious Encryption. In *ASIACRYPT*, pages 293–307, 2004.

[12] S. Chiou, S. Chang, and H. Sun. Common Friends Discovery with Privacy and Authenticity. In *IAS*, pages 337–340, 2009.

[13] D. Dachman-Soled, T. Malkin, M. Raykova, and M. Yung. Efficient Robust Private Set Intersection. In *ACNS*, pages 125–142, 2009.

[14] E. De Cristofaro, J. Kim, and G. Tsudik. Linear-Complexity Private Set Intersection Protocols Secure in Malicious Model. In *ASIACRYPT*, pages 213–231, 2010.

[15] E. De Cristofaro and G. Tsudik. Practical Private Set Intersection Protocols with Linear Complexity. In *Financial Cryptography*, pages 143–159, 2010.

[16] Y. Desmedt. Securing Traceability of Ciphertexts — Towards a Secure Software Key Escrow System. In *EUROCRYPT*, pages 147–157, 1995.

[17] C. Diehl, G. Namata, and L. Getoor. Relationship Identification for Social Network Discovery. *AAAI*, 22(1):546–552, 2007.

[18] Free Software Foundation. The GNU MP Bignum Library. `http://gmplib.org/`.

[19] M. J. Freedman and A. Nicolosi. Efficient Private Techniques for Verifying Social Proximity. In *IPTPS*, 2007.

[20] M. J. Freedman, K. Nissim, and B. Pinkas. Efficient Private Matching and Set Intersection. In *EUROCRYPT*, pages 1–19, 2004.

[21] O. Goldreich, S. Micali, and A. Wigderson. How to Play any Mental Game or A Completeness Theorem for Protocols with Honest Majority. In *STOC*, pages 218–229, 1987.

[22] F. Günther, M. Manulis, and T. Strufe. Cryptographic Treatment of Private User Profiles. In *Financial Cryptography Workshops*, 2011. Available from `http://eprint.iacr.org/2011/064`.

[23] C. Hazay and Y. Lindell. Efficient Protocols for Set Intersection and Pattern Matching with Security Against Malicious and Covert Adversaries. In *TCC*, pages 155–175, 2008.

[24] C. Hazay and K. Nissim. Efficient Set Operations in the Presence of Malicious Adversaries. In *PKC*, pages 312–331, 2010.

[25] B. Huberman, M. Franklin, and T. Hogg. Enhancing Privacy and Trust in Electronic Communities. In *EC*, pages 78–86, 1999.

[26] S. Jarecki, J. Kim, and G. Tsudik. Beyond Secret Handshakes: Affiliation-Hiding Authenticated Key Exchange. In *CT-RSA*, pages 352–369, 2008.

[27] S. Jarecki and X. Liu. Affiliation-Hiding Envelope and Authentication Schemes with Efficient Support for Multiple Credentials. In *ICALP (2)*, pages 715–726, 2008.

[28] S. Jarecki and X. Liu. Private Mutual Authentication and Conditional Oblivious Transfer. In *CRYPTO*, pages 90–107, 2009.

[29] S. Jarecki and X. Liu. Fast Secure Computation of Set Intersection. In *SCN*, pages 418–435, 2010.

[30] L. Kissner and D. X. Song. Privacy-Preserving Set Operations. In *CRYPTO*, pages 241–257, 2005.

[31] A. Korolova, R. Motwani, S. Nabar, and Y. Xu. Link Privacy in Social Networks. In *CIKM*, pages 289–298, 2008.

[32] Y. Lindell and B. Pinkas. An Efficient Protocol for Secure Two-Party Computation in the Presence of Malicious Adversaries. In *EUROCRYPT*, pages 52–78, 2007.

[33] M. Manulis, B. Pinkas, and B. Poettering. Privacy-Preserving Group Discovery with Linear Complexity. In *ACNS*, pages 420–437, 2010.

[34] M. Manulis and B. Poettering. Practical Affiliation-Hiding Authentication from Improved Polynomial Interpolation. In *ASIACCS*, pages 286–295, 2011. Available from `http://eprint.iacr.org/2010/659`.

[35] M. Manulis, B. Poettering, and G. Tsudik. Affiliation-Hiding Key Exchange with Untrusted Group Authorities. In *ACNS*, pages 402–419, 2010.

[36] M. Manulis, B. Poettering, and G. Tsudik. Taming Big Brother Ambitions: More Privacy for Secret Handshakes. In *PETS*, pages 149–165, 2010.

[37] E. Okamoto. Key Distribution Systems Based on Identification Information. In *CRYPTO*, pages 194–202, 1987.

[38] P. Pons and M. Latapy. Computing Communities in Large Networks using Random Walks. In *ISCIS*, pages 284–293, 2005.

[39] The Facebook, Inc. Facebook's statistics. `http://www.facebook.com/press/info.php?statistics`, 2010.

[40] M. Von Arb, M. Bader, M. Kuhn, and R. Wattenhofer. Veneta: Serverless Friend-of-Friend Detection in Mobile Social Networking. In *WiMob*, pages 184–189, 2008.

[41] S. Xu and M. Yung. k-Anonymous Secret Handshakes with Reusable Credentials. In *CCS*, pages 158–167, 2004.

[42] A. Yao. How to Generate and Exchange Secrets. In *FOCS*, pages 162–167, 1986.

[43] P. S. Yu, J. Han, and C. Faloutsos. *Link Mining: Models, Algorithms, and Applications.* Springer, 2010.

[44] E. Zheleva, L. Getoor, J. Golbeck, and U. Kuter. Using Friendship Ties and Family Circles for Link Prediction. In *SNA-KDD*, pages 97–113, 2008.

A Proof of Contact-Hiding Security of CDS from Fig. 1

We now prove the Contact-Hiding (CH) security (Definition 4) of our CDS protocol (illustrated in Fig. 1), relying on the RSA assumption on safe moduli (Definition 1) and on the security of the IHME scheme from [33]. By $\mathsf{Adv}^{\mathsf{ihide}}_{\mathsf{IHME}}(\kappa)$ we denote the advantage probability for breaking the index-hiding property of the IHME scheme, which (since the utilized IHME scheme is perfect) is equal to 0. Note that our security arguments have similarities with those of the multi-credential AHA protocol in [33], which proves the "affiliation hiding" property of the proposed AHA under the same assumptions. We only sketch the proof steps that mirror those in [33], as the reader can find more details in [33] and its predecessor [26]. In contrast, we detail arguments regarding the use of the IHME scheme. We prove CH-Security of CDS by presenting a sequence of games $\mathbf{G}_0, \ldots, \mathbf{G}_5$.

Game $\mathbf{G}_0$. We start with $\mathbf{G}_0 = \mathsf{Game}^{\mathsf{ch},b}_{\mathcal{A},\mathsf{CDS}}(\kappa, n)$, in which $\mathcal{A}$ interacts with simulator $\mathcal{C}$, which answers all queries honestly according to the specification of the game (see Definition 4).

Game $\mathbf{G}_1$. Game $\mathbf{G}_1$ is like $\mathbf{G}_0$, except that the simulation is aborted (the game outputs 0) if there exists a Discover session $\pi' \neq \pi^*$ that sends out the same $\mathcal{M}$ structure as π^* (see line 10 in Figure 1).

Note that $\mathcal{M}$ sent by π^* contains for each contact in CL^*_b a specific θ value. These are almost uniformly distributed in a set of size $p \approx 2^{2\kappa'+\kappa}$. Thus, the probability of finding a collision in the $\mathcal{M}$'s is upper-bounded by $q_q/2^{2\kappa'+\kappa}$ (where q_q denotes the number of Discover queries), and thus negligible in κ.

Game $\mathbf{G}_2$. Let $R = (r_1, \ldots, r_k)$ denote the list of r-values for the contacts in $\mathsf{CL}^*_b \setminus \mathsf{CL}^*_{1-b} = \mathsf{CL}^*_b \cap \mathcal{D}^*$ of session π^*, as computed in line 18 of the protocol. Game $\mathbf{G}_2$ is like $\mathbf{G}_1$, except that all confirmation messages c_0, c_1 of π^* (see lines 19 and 20), computed based on the values in R, are replaced by random elements in the respective range.

By the Random Oracle Model (ROM), the modification introduced in Game $\mathbf{G}_2$ can only be detected by adversaries that can compute and query the H oracle on at least one of the r-values in R. Let $r_t \in R$ be such a value (and assume that the simulator guesses t correctly). By embedding an RSA challenge into user identifiers (by programming the H_N oracle) and public user parameters (by choosing N and g appropriately), the problem of computing r_t can be reduced to the hardness of the RSA problem (see [26, Section 3] for more details). We conclude that the computational distance between $\mathbf{G}_2$ and $\mathbf{G}_1$ is polynomially dependent on $\mathsf{Succ}^{\mathsf{rsa}}(\kappa')$, and is thus negligible in κ.

Remark 3. Note that for all contacts in $\mathsf{CL}^*_b \cap \mathcal{D}^*$, the adversary $\mathcal{A}$ cannot distinguish correct confirmation messages for π^* from random ones. Therefore, the output set $\pi^*.\mathsf{SCL}$ is disjoint with $\mathcal{D}^*$.

Game $\mathbf{G}_3$. This game is like Game $\mathbf{G}_2$, except that, for session π^*, the θ-values for all contacts in $\mathsf{CL}^*_b \cap \mathcal{D}^*$ (as computed in line 7) are replaced by values uniformly random in $[0, p-1]$.

Observe from the protocol definition that the θ-values replaced in this game only affect the computation of the $r_t \in R$ (in Game $\mathbf{G}_2$), which cannot be computed and checked by the adversary anyway by a result of Game $\mathbf{G}_2$. Hence, the only detectable difference between $\mathbf{G}_2$ and $\mathbf{G}_3$ may arise from different distributions of the original and the modified θ-values. Although the original values are *not* uniformly distributed in $[0, p-1]$, their distribution is statistically indistinguishable from the uniform distribution. In [26], the corresponding statistical difference is proven to be bounded by $2^{-\kappa}$, what is negligible in κ.

Game $\mathbf{G}_4$. Game $\mathbf{G}_4$ is like $\mathbf{G}_3$, except that, for session π^*, in the IHME-encoding step in line 10, for all contacts in $\mathsf{CL}^*_b \cap \mathcal{D}^*$, the indices N are replaced by the indices N that correspond to the contacts in $\mathsf{CL}^*_0 \cap D^*$. For all contacts in $\mathsf{CL}^*_b \cap \mathsf{CL}^*_{1-b}$ the indices remain unchanged.

As Games $\mathbf{G}_3$ and $\mathbf{G}_4$ are exactly the same in case $b = 0$ (as nothing has changed), in the following we assume $b = 1$. We show that, if an efficient distinguisher $\mathcal{D}^{3,4}$ that distinguishes between Game $\mathbf{G}_3$ and $\mathbf{G}_4$ exists, then we can use it to construct an adversary $\mathcal{A}^{\mathsf{ihide}}$ against index-hiding of IHME as follows. Adversary $\mathcal{A}^{\mathsf{ihide}}$ acts as a challenger for $\mathcal{D}^{3,4}$, i.e., $\mathcal{A}^{\mathsf{ihide}}$ sets up all users, and answers all protocol queries honestly (but following the rules of Game $\mathbf{G}_3$), with the only exception that the encoding step in line 10 for session π^* is performed by the IHME challenger (i.e., the latter receives (I_0, I_1, M') where I_0 and I_1 are the sets of indices N of CL_0^* and CL_1^*, respectively, and M' is the list of the θ-values honestly computed for the contacts in $\mathsf{CL}_0^* \cap \mathsf{CL}_1^*$), which returns an encoding $\mathcal{M}$ using either the indices corresponding to CL_0^* or CL_1^*. The bit output by $\mathcal{D}^{3,4}$ serves as output b' for $\mathcal{A}^{\mathsf{ihide}}$. We see that the success probability of $\mathcal{D}^{3,4}$ is bounded by $\mathsf{Adv}^{\mathsf{ihide}}_{\mathsf{IHME}}(\kappa)$, which is 0 (since the IHME construction is perfect). Hence, the computational difference between Games $\mathbf{G}_3$ and $\mathbf{G}_4$ is 0.

Game $\mathbf{G}_5$. The step between Game $\mathbf{G}_4$ and $\mathbf{G}_5$ is very similar to the previous transition: this time, it is the IHME-encoding in line 24 for which the indices N of all contacts in $\mathsf{CL}_b^* \cap D^*$ are replaced by the indices N that correspond to the contacts in $\mathsf{CL}_0^* \cap D^*$.

The difference between $\mathbf{G}_5$ and $\mathbf{G}_4$ is bounded by $\mathsf{Adv}^{\mathsf{ihide}}_{\mathsf{IHME}}(\kappa) = 0$, exactly for the same reason above.

Conclusion. We conclude that the computational difference between $\mathbf{G}_0$ and $\mathbf{G}_5$ is negligible in κ. Therefore, a CH-adversary cannot distinguish between $\mathsf{Game}^{\mathsf{ch},b}_{\mathcal{A},\mathsf{CDS}}$ and $\mathsf{Game}^{\mathsf{ch},0}_{\mathcal{A},\mathsf{CDS}}$ with non-negligible probability, neither by analyzing the exchanged messages, nor by interpreting the results of Reveal queries. As the latter game contains no information about bit b, it follows that $\mathsf{Adv}^{\mathsf{ch}}_{\mathsf{CDS}}(\kappa, n)$ is negligible in κ (for all n polynomially dependent on κ).