Int. J. Inf. Secur. (2013) 12:49–65
DOI 10.1007/s10207-012-0183-4

SPECIAL ISSUE PAPER

Private discovery of common social contacts

Emiliano De Cristofaro · Mark Manulis · Bertram Poettering

Published online: 16 December 2012

Abstract Digital services that are offered, and consumed, on the basis of social relationships form the backbone of *social clouds*—an emerging new concept that finds its roots in online social networks. The latter have already taken an essential role in people's daily life, helping users to build and reflect their social relationships to other participants. A key step in establishing new links entails the reconciliation of shared contacts and friends. However, for many individuals, personal relationships belong to the private sphere, and, as such, should be concealed from potentially prying eyes of strangers. Consequently, the transition toward social clouds cannot set aside mechanisms to control the disclosure of social links. This paper motivates and introduces the concept of *Private Discovery of Common Social Contacts*, which allows two users to assess their social proximity through interaction and learn the set of contacts (e.g., friends) that are common to both users, while hiding contacts that they do not share. We realize private contact discovery using a new cryptographic primitive, called contact discovery scheme (CDS), whose functionality and privacy is formalized in this work. To this end, we define a novel privacy feature, called *contact-hiding*, that captures our strong privacy goals. We also propose the concept of *contact certification* and show that it is essential to thwart impersonation attacks on social relationships. We build provably private and realistically efficient CDS protocols for private discovery of mutual contacts. Our constructions do not rely on a trusted third party (TTP)—all contacts are managed independently by the users. The practicality of our proposals is confirmed both analytically and experimentally on different computing platforms. We show that they can be efficiently deployed on smartphones, thus allowing ad hoc and ubiquitous contact discovery outside of existing social networks. Our CDS constructions allow users to select their (certified) contacts to be included in individual protocol executions. That is, users may perform context-dependent contact discovery using any subset (circle) of their contacts.

Keywords Common social contacts · Social clouds · Friend-of-friend detection · Social PKI · Privacy

E. De Cristofaro
Palo Alto Research Center (PARC), 3333 Coyote Hill Road, Palo Alto, CA 94041, USA
e-mail: edc@parc.com

M. Manulis
Department of Computing, University of Surrey, Guildford, Surrey GU2 7XH, United Kingdom
e-mail: mark@manulis.eu

B. Poettering (✉)
ISG, Royal Holloway, University of London, Egham, Surrey TW20 0EX, United Kingdom
e-mail: bertram.poettering@rhul.ac.uk

1 Introduction

1.1 Social clouds, relationships, and PKIs

The concept of *social clouds* is an emerging paradigm that allows users to mutually offer and consume services, such as interest sharing, activity planing, organization of events, multimedia content exchange, and so on. Social clouds offer an appealing way for users to expand and reflect their social relationships, and then use them for social interaction or collaboration in business and leisure. In 2012, popular social networking sites, such as Facebook, Linkedin, or MySpace, already involved millions (possibly billions) of active users [13,30]. Moreover, there is a tremendous growth in the number of users accessing social network services *ubiquitously*—for instance, 500 of today's 900 million

Facebook users access it from their mobile devices [13]. Naturally, the increasing amount of users and interactions also prompts some privacy concerns, with respect to private information continuously disclosed to users and providers.

At the basis of any trustworthy social interaction among users lies the initial establishment of their social relationship. Just as in real life, users of social clouds can approach unknown users directly to exchange contacts or to become friends. However, a more promising way is often to verify the existence of *common* contacts or friends. The establishment of social relationship between unfamiliar users based on common social relationships is appealing to build a *social PKI*, where trust into users' public keys is established and verified based on users' social proximity. However, given its intrinsically social nature, this approach creates more severe privacy issues when compared to traditional PKIs. Indeed, personal social relationships of a user can be regarded as a sort of "social wealth", which many users would like to keep private. As a result, the initial establishment of social relationships between two unfamiliar users is, at the same time, a challenging task (w.r.t. user privacy) and a necessary building block for social PKIs and for any other interaction in the social cloud.

Consider the following scenario: Two unfamiliar users, connected to the same mobile ad hoc network, would like to assess their social proximity by discovering their mutual contacts or friends. A naïve solution would require them to reciprocally reveal their friends' identities. Clearly, this would completely expose their contact lists and would not achieve any privacy protection. Another intuitive solution would employ a central server (TTP) to collect their lists and output common friends. However, such a server would not only learn participants' identities and friends, but presumably also time and location of their interaction. Moreover, central servers are not necessarily reachable in an ad hoc environment, for example, if users meet in a place without an Internet connection or in case they want to operate outside an existing infrastructure.

1.2 Private contact discovery

The concept of *private contact discovery*, which the authors introduced in [8] and now extend in this paper, is a novel general construct geared to preserve user privacy, not only in existing social network interactions, but also in any other application that could be based on the social proximity among users. In particular, private contact discovery can be seen as a fundamental building block for designing social PKIs, where establishment of mutual trust between users (and their public keys) would depend on their social proximity.

We introduce a new cryptographic primitive, called *Contact Discovery Scheme* (CDS), which lets two users, on input their respective contact lists, learn their mutual contacts (if any), and nothing else. The essential privacy property behind CDS is the property of *contact-hiding*—it asserts that no information about existing social relationships of some participating user is leaked to another user who is not "sufficiently close" in the social graph. In this work, we approach contact-hiding CDS solutions in the context of mutual social contacts of the first degree. That is, with our CDS protocols, unfamiliar users can discover their mutual first-degree contacts or friends without leaking any information about other existing relationships.

From a system point of view, our CDS protocols do not rely on any third party nor are they bound to any specific network infrastructure. Thus, they are suitable for more general social cloud settings than those offered by existing social networking sites; for example, users may use our schemes to interact outside of a particular social network site and still be able to assess their social proximity and—depending on the outcome—establish their social relationship and mutual trust in the digital world. The efficiency of our protocols allows deployment on mobile devices, as demonstrated by our experiments. Thus, our schemes are also suitable for *ubiquitous* social interaction and establishment of ad hoc relationships.

An important design element of CDS is the notion of *contact certification*; for instance, in order for one user, Alice, to be able to claim existing social relationship to another user, Carol, it is not sufficient for Alice to include Carol's identity into her contact list. On the contrary, Alice must obtain Carol's authorization beforehand, in form of a contact certificate. Alice can then use this certificate within CDS protocol sessions with some third (unfamiliar) user, Bob, and learn whether Carol is their common first-degree contact, provided that also Bob has an appropriate certificate issued by Carol. We argue that contact certification is essential to prevent users from claiming unwarranted social relationships to other users in the digital world. In other words, contact certification is fundamental to prevent impersonation attacks in the social context and is an important building block for social PKIs based on the establishment and discovery of mutual social relationships. As we illustrate in Sect. 1.3, existing approaches to realize functionalities similar to private contact discovery do not base on contact certification. As a consequence, they offer only a limited privacy gain in the real world where adversarial users are not prevented from maliciously inflating their contact lists with the aim of maximizing the amount of information learned about other users' contacts.

In order to effectively preserve the privacy of existing social relationships, it is essential to consider contact certificates as private information themselves. As a result, one of the challenges in the design of suitable contact-hiding CDS protocols is to let two users determine whether their contact lists contain certificates issued by the same users (their shared contacts) without disclosing those lists and

without leaking any information about other contacts that they do not have in common. Moreover, the transcripts of the CDS protocol sessions between Alice and Bob should not reveal the identities of their social contacts to any third party eavesdropping the protocol execution or attempting to actively engage one of the participants in the protocol session.

1.3 Prior attempts toward private contact discovery

We now give an overview over some prior attempts to approach problems similar to private contact discovery and discuss their shortcomings. To the best of our knowledge, none of existing approaches offers adequate privacy protection of contacts, considering our strong contact-hiding requirements (which is formally modeled in Sect. 4.4). We also analyze whether the challenge behind the design of contact-hiding CDSs can be solved using existing cryptographic techniques.

Von Arb et al. [39] present a mobile social networking platform that enables *Friend-of-Friend* (FoF) detection in physical proximity. Their solution compares friend lists through methods of *private set intersection* (PSI) [1,5,7,9,16,20,21,24–26]. In PSI schemes, users run on input their individual sets of elements, and the goal of the protocol is to let them learn the intersection of these sets without disclosing any information about further elements. Roughly speaking, Von Arb's attempt to design a CDS protocol is to simply run a PSI protocol on participating users' friend lists and to output obtained matches. Unfortunately, it is trivial for an attacker to include the identities of arbitrary users in its input list and to learn peer's friend list from the output of the PSI protocol. We observe that a primitive related to PSI, called *Authorized PSI* (APSI) [10,9], does not prevent malicious users from arbitrarily manipulating their input lists either, since authorized elements can be passed on from user to user. The problem is that APSI assumes a single (trusted) global authority that, in the context of CDS, would have to certify contact relationship between the users. Within others, this contradicts the idea behind private contact discovery where users should be able to manage their social relationships on their own and keep their relationship undisclosed to third parties.

Freedman and Nicolosi [15] propose two additional solutions for the FoF problem, in the context of trust establishment in email white-listing. One solution is based on hash functions and symmetric encryption, the other on bilinear maps. Both solutions leverage friendship attestation, but basically implement the naïve matching approach where users jointly enumerate all $O(n^2)$ combinations of their friends and then run an equality check on each pair to identify matching contacts. The solution based on symmetric encryption opens ways to social impersonation attacks by allowing users to maliciously transfer attestations to other users, meaning that users can claim social relationships to other users without having their consent. The pairing-based technique suffers from similar problems and is furthermore inefficient as it involves a quadratic number of bilinear map operations. We observe that constructions from [15] lack rigorous security analysis.

Huang, Chapman, and Evans [3,22] recently described their ready-to-use contact discovery application for the Android platform. Using garbled circuits [40] to solve private contact discovery as a generic instance of secure multiparty computation, they report timing values of 150 s to match 128 contacts, which seems relatively inefficient for real-world deployment. Additionally, their construction suffers from the security issues discussed in the PSI context above, that is, adversaries are not prevented from arbitrarily populating their contact lists. Moreover, security is claimed only in a model with semi-honest adversaries—we argue that such weak model is inappropriate for Internet applications involving social relationships where the presence of users who arbitrarily misbehave just to lure other users to believe the existence of forged social relationships must be expected.

We observe that there is a range of Friend-of-Friend detection mechanisms that either do not aim at achieving any privacy goals or have very loose and unclear privacy requirements, for example, [4,27,28]. In the domain of non-private solutions, there are also some more general approaches for dealing with social relationships; for instance, [37] uses random walks to discover *communities* in large social network graphs, [41, Chapter 12] formalizes the problem of dynamically identifying core communities (i.e., sets of entities with frequent and consistent interactions), [42] builds a prediction model to identify certain social structures, for example, friendship ties and family circles, while [11] aims at identification of communications that substantiate social relationship types.

2 Contributions and technical roadmap

This section presents our contributions, proposed techniques, and the organization of this paper.

2.1 Formalizing private contact discovery

We formalize the concept of private contact discovery by defining a new cryptographic primitive called *contact discovery scheme* (CDS). The core functionality of CDS allows users to independently certify their contacts and any pair of participants to determine the set of their mutual first-degree contacts. The corresponding privacy goal is modeled through the notion of *contact-hiding*. We use a standard cryptographic modeling approach where security and privacy goals are defined through games played between an adversary and a challenger that simulates the honest users. The

contact discovery protocol of a contact-hiding CDS leaks no information about contacts that are not shared with the other protocol participant. We also discuss how to extend the CDS functionality to support revocation of contact certificates.

2.2 Contact-hiding CDS constructions

In this paper, we propose two different CDS protocols. Our first construction of contact-hiding CDS bases on the RSA-based, identity-based key exchange protocol by Okamoto and Tanaka [12]. However, in order to achieve the contact-hiding property, we had to slightly modify it. Specifically, we introduce techniques for blinding and padding of RSA group elements; these techniques were applied earlier in the context of some RSA-based affiliation-hiding protocols and secret handshakes [23,31,32]. The high-level idea behind our CDS protocol is to let each protocol participant interact with its peer and then compute a secret value for each (certified) contact in the input contact list. Then, in order to determine the set of mutual contacts, users need to identify which of the computed secret values match on both sides. The security property of the protocol guarantees that matching secret values occur only for mutual contacts and that, if some contact does not match, then no information about the corresponding secret value is leaked to the adversary. Our CDS protocol enjoys linear (in the number of alleged contacts) communication and computation complexity, when considering the number of public-key operations. This efficiency is achieved by using the *index-hiding message encoding (IHME)* technique developed in [31], where it was introduced to solve the then open problem of efficient group discovery (cf. [23]). In our CDS protocol, IHME is applied to encode messages of the aforementioned modified key exchange protocol. The index-hiding property of IHME is hereby essential to achieve the contact-hiding requirement of CDS.

We obtain our second CDS protocol by optimizing our first construction using an advanced IHME technique, called *interleaved IHME*, that was developed in the context of affiliation-hiding protocols in [33]. Its encoding operation splits messages into multiple chunks that are then individually encoded prior to their transmission. The shorter length of individual chunks results in a significant efficiency gain: currently known IHME constructions view messages as elements of a certain finite field, to which polynomial interpolation is applied in the encoding procedure, which in turn requires a quadratic number of ('symmetric') field operations. Optimized polynomial interpolation from [33], in combination with interleaved IHME, allows us to significantly decrease the computational overhead compared to our first CDS construction.

We analyze performance of our optimized CDS protocol by conducting measurements on three different computing platforms: (1) a server machine equipped with an Intel XEON CPU running at 2.6 GHz, (2) an AMD NEO at 1.6 GHz (often deployed in Netbook computers), and (3) an ARMv7 CPU running at 600 MHz, which is the de facto standard on smartphones produced in early 2010's. Our experiments attest to the practicality of our optimized CDS protocol and show that it remains efficient even for smartphones.

Our CDS protocols enjoy several other interesting properties. For instance, we show that users can independently choose (on a *per-session* basis) which of their certified contacts they want to deploy in a protocol run. This property improves flexibility of a CDS since users may decide not to reveal existence of some first-degree contacts even if these contacts are shared with the session partner. Contact discovery often occurs within some social context, put forth either by the application or the social environment within which the protocol is executed. For example, users who wish to discover shared business contacts, when using our protocol, can leave out certificates for members of their family, or other unrelated contacts, from the input list to that protocol session. Not only that this context-dependent choice of input contacts further speeds up the discovery process since not all existing contacts have to be processed, this feature also makes our protocol a perfect choice for *context-dependent* forms of contact management and discovery on the user's side, as reflected in the concept of 'circles' recently adopted in Google+ [19]. For example, when using our CDS protocols, users can freely arrange their contacts into various circles (e.g., family, close friends, business partners, etc.) and perform private contact discovery with respect to any of them.

Organization. Section 3 recalls some relevant mathematical background, reviews computational assumptions, and gives an overview over the Index-Hiding Message Encoding functionality from [31] and its optimized version from [33]. In Sect. 4 we formalize the concept of CDS schemes. Section 5 presents our first CDS construction and shows that it is provably contact-hiding. In Sect. 6, we present several optimizations and evaluate performance of proposed protocol, on three different computing platforms. Finally, Sect. 7 concludes the paper and discusses several directions for future research.

3 Prerequisites and building blocks

This section reviews computational assumptions and the concept of Index-Hiding Message Encoding.

3.1 Mathematical background

Although we generally assume that the reader is familiar with basic concepts of number theory and corresponding

computational assumptions, we review some important notions needed throughout this paper.

A *safe prime* p is a prime number such that $p = 2p' + 1$ holds, for a prime p'. For a safe prime p, the multiplicative group $\mathbb{Z}_p^\times$ of the finite field $\mathbb{Z}_p \cong GF(p)$ has order $p - 1 = 2p'$, and each of its subgroups has order 1, 2, p' or $2p'$ (by Lagrange's theorem). The subgroup of order p' consists exactly of all squares in $\mathbb{Z}_p^\times$; hence, it is called the subgroup of *quadratic residues* mod p, $QR(p)$ for short. Note that $QR(p)$ is generated by each square in $\mathbb{Z}_p^\times$, except by 1, and that $2|QR(p)| = |\mathbb{Z}_p^\times|$. Note also that about every second element g in $\mathbb{Z}_p^\times$ is primitive, that is, $\langle g \rangle_p = \mathbb{Z}_p^\times$.

For any prime p, the *Legendre symbol* $(\frac{\cdot}{p}) : \mathbb{Z}_p^\times \to \{-1, 1\}$ is defined by $(\frac{a}{p}) = 1 :\Leftrightarrow a \in QR(p)$. By considering $\{-1, 1\} = \mathbb{Z}_3^\times$, this mapping becomes a group homomorphism (with $\ker(\frac{\cdot}{p}) = QR(p)$). It can be proven that $(\frac{-1}{p}) = -1$ if and only if $p = 3 (\mathrm{mod}\, 4)$.

Euler's totient function $\varphi : \mathbb{N} \to \mathbb{N};\ m \mapsto \varphi(m)$ indicates the number of invertible elements in $\mathbb{Z}_m$, that is, $\varphi(m) = |\mathbb{Z}_m^\times|$. The related *Carmichael function* $\lambda : \mathbb{N} \to \mathbb{N};\ m \mapsto \lambda(m)$ indicates the order of the largest cyclic subgroup in $\mathbb{Z}_m^\times$. Both functions can easily be computed if the factorization of its argument is known. In particular, if $n = pq$ is an RSA modulus, that is, p, q are prime numbers, then $\varphi(n) = (p - 1)(q - 1)$ and $\lambda(n) = \mathrm{lcm}(p - 1, q - 1)$. If n is moreover a *safe RSA modulus*, that is, if $p = 2p' + 1$ and $q = 2q' + 1$ are safe primes, then we have $\lambda(n) = 2p'q' = \varphi(n)/2$. Observe that, for any given element $m \in \mathbb{Z}_n \backslash \mathbb{Z}_n^\times$, a non-trivial factor of n is given by $\gcd(m, n)$. Hence, picking elements at random from $\mathbb{Z}_n$ will yield non-invertible elements only with negligible probability, assuming hardness of the factorization problem.

The *Chinese Remainder Theorem* (CRT) states that rings $\mathbb{Z}_{pq}$ and $\mathbb{Z}_p \times \mathbb{Z}_q$ are isomorphic, for all primes p, q. We denote this by $\mathbb{Z}_{pq} \cong \mathbb{Z}_p \times \mathbb{Z}_q$. The corresponding (ring) isomorphism is given by $a_n \mapsto (a_n \mathrm{mod}\, p, a_n \mathrm{mod}\, q)$, and its inverse by $(a_p, a_q) \mapsto a_p + ph$ for $h = (a_q - a_p)/p (\mathrm{mod}\, q)$. Note that it follows that groups $\mathbb{Z}_{pq}^\times$ and $\mathbb{Z}_p^\times \times \mathbb{Z}_q^\times$ are isomorphic as well, that is, $\mathbb{Z}_{pq}^\times \cong \mathbb{Z}_p^\times \times \mathbb{Z}_q^\times$.

Let $n = pq$ be a safe RSA modulus. In particular, n is a Blum integer, that is, $p = q = 3 (\mathrm{mod}\, 4)$, and it follows that $-1 \notin QR(p)$ and $-1 \notin QR(q)$. Consider an element $g \in \mathbb{Z}_n^\times$ that is primitive in $\mathbb{Z}_p$ but is a quadratic residue mod q (or vice versa). Then, $\mathrm{ord}_n(g) = 2p'q' = \lambda(n)$ and $-1 \notin \langle g \rangle_n$. In this case, we have $\mathbb{Z}_n^\times \cong \langle -1 \rangle_n \times \langle g \rangle_n$. A simple combinatorial argument shows that this property holds for about a half of the elements in $\mathbb{Z}_n^\times$.

We formalize the RSA assumption in the setting of safe RSA moduli.

Definition 1 (*RSA assumption on safe moduli*) Let $\mathsf{SRSA-GEN}$ be an efficient algorithm that, on input security parameter 1^κ, outputs tuples (n, e, d) such that (a) $n = pq$ for primes p and q, (b) $p = 2p' + 1$ and $q = 2q' + 1$ for primes p' and q', and (c) $e, d \in \mathbb{Z}_{\varphi(n)}^\times$ with $ed = 1 \mathrm{mod}\, \varphi(n)$. The success probability of an adversary $\mathcal{A}$ with respect to $\mathsf{SRSA-GEN}$ is defined as

$$\mathsf{Succ}^{\mathsf{srsa}}_{\mathsf{SRSA-GEN},\mathcal{A}}(\kappa) = \Pr\left[\mathrm{Expt}^{\mathsf{srsa}}_{\mathsf{SRSA-GEN},\mathcal{A}}(\kappa) = 1\right]$$

where $\mathrm{Expt}^{\mathsf{srsa}}_{\mathsf{SRSA-GEN},\mathcal{A}}$ is defined in Fig. 1. The *RSA assumption on safe moduli* states that there exists an algorithm $\mathsf{SRSA-GEN}$ such that $\mathsf{Succ}^{\mathsf{srsa}}_{\mathsf{SRSA-GEN},\mathcal{A}}$ is negligible for all efficient adversaries $\mathcal{A}$.

$\mathrm{Expt}^{\mathsf{srsa}}_{\mathsf{SRSA\text{-}GEN},\mathcal{A}}(\kappa)$:
(a) $(n, e, d) \leftarrow_R \mathsf{SRSA\text{-}GEN}(1^\kappa)$
(b) $z \leftarrow_R \mathbb{Z}_n^\times$
(c) $m \leftarrow \mathcal{A}(n, e, z)$
(d) Return 1 iff $m^e = z$.

Fig. 1 SRSA experiment

3.2 Index-hiding message encoding

A tool that plays a central role in our privacy-preserving CDS protocols is a primitive called *index-hiding message encoding* (IHME), first proposed in [31]. The related concept of *index-based message encoding* is a technique that allows to encode a set of 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 of these messages can individually be recovered from $\mathcal{S}$ by addressing it via its *index*, which is arbitrarily chosen from an index set $\mathcal{I}$ and specified at encoding time. The scheme is *index-hiding* if it is impossible for an adversary to reveal information about the deployed indices by inspecting $\mathcal{S}$. These notions are now formalized, by first giving a syntactical definition of IBME, and then a game-based definition of IHME's index-hiding property.

Although the precise context in which we deploy IHME in our CDS protocols will become fully clear only in Sects. 5 and 6, we anticipate that we will let users specify as indices the identities of their individual contacts. Peers with matching contacts can recover and interpret the corresponding messages, while IHME's index-hiding property protects contacts' identities from adversarial observers.

Definition 2 (*Index-based message encoding*) An *index-based message encoding* scheme over an index space $\mathcal{I}$ and a message space $\mathcal{M}$ is a set IBME={iEncode,iDecode} of two efficient algorithms:

iEncode($\mathcal{P}$)

On input a set $\mathcal{P}$ of n index/message pairs, that is, $\mathcal{P} = \{(i_1, m_1), \ldots, (i_n, m_n)\} \subseteq \mathcal{I} \times \mathcal{M}$, with distinct indices $i_j,\ j \in [1, n]$, this algorithm outputs an encoding $\mathcal{S}$.

$\text{Expt}^{\text{ihide},b}_{\text{IHME},\mathcal{A}}(\kappa)$:
(a) $(I_0, I_1, M, state) \leftarrow \mathcal{A}_1(1^\kappa)$ such that $I_0, I_1 \subseteq \mathcal{I}$, $|I_0| = |I_1|$, and $M \in \mathcal{M}^k$, with $I_0 \cap I_1 = \{i_1, \ldots, i_k\}$ and $M = (m_1, \ldots, m_k)$, for some i_j, m_j, and k. Let $n = |I_0| = |I_1|$.
(the adversary chooses two sets of n indices each, as well as, for each index i_j in the intersection of these sets, a corresponding message m_j)
(b) let $I_b \setminus I_{1-b} =: \{i_{k+1}, \ldots, i_n\}$ and $m_{k+1}, \ldots, m_n \leftarrow_R \mathcal{M}$
($n - k$ further messages are chosen uniformly at random)
(c) $\mathcal{S} \leftarrow \mathsf{iEncode}(\mathcal{P})$ for $\mathcal{P} = \{(i_1, m_1), \ldots, (i_n, m_n)\} \subseteq \mathcal{I} \times \mathcal{M}$
(the messages are encoded for the indices in I_b)
(d) $b' \leftarrow \mathcal{A}_2(state, \mathcal{S})$
(e) return b'

Fig. 2 Index-hiding experiment

$\mathsf{iDecode}(\mathcal{S}, i)$

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

An IBME scheme is *correct* if $\mathsf{iDecode}(\mathsf{iEncode}(\mathcal{P}), i_j) = m_j$ for all $j \in [1, n]$, for all sets $\mathcal{P} = \{(i_1, m_1), \ldots, (i_n, m_n)\} \subseteq \mathcal{I} \times \mathcal{M}$ with distinct indices i_j.

Informally, an IBME scheme is *index-hiding* if it hides the indices in which the messages are encoded. That is, it ensures that an attacker who sees an encoding $\mathcal{S}$ and might even know some of the indices and corresponding messages cannot identify any other indices in which messages are encoded. We formalize this property in Definition 3.

Definition 3 (*Index-hiding message encoding*) Let IHME=$\{\mathsf{iEncode}, \mathsf{iDecode}\}$ denote an IBME scheme over index space $\mathcal{I}$ and message space $\mathcal{M}$. Let $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$ be an adversary that participates in the experiment of Fig. 2. The advantage of $\mathcal{A}$ is defined as

$$\text{Adv}^{\text{ihide}}_{\text{IHME},\mathcal{A}}(\kappa) = \left| \Pr\left[\text{Expt}^{\text{ihide},0}_{\text{IHME},\mathcal{A}}(\kappa) = 1\right] - \Pr\left[\text{Expt}^{\text{ihide},1}_{\text{IHME},\mathcal{A}}(\kappa) = 1\right] \right|.$$

We say that IHME is *index-hiding* if this advantage is negligible for all efficient adversaries $\mathcal{A}$. Moreover, IHME is *perfectly index-hiding* if $\text{Adv}^{\text{ihide}}_{\text{IHME},\mathcal{A}}(\kappa) = 0$ for all (unbounded) adversaries $\mathcal{A}$, for all κ.

3.2.1 A construction of IHME

We propose an efficient and perfectly index-hiding construction of IHME which is based on polynomial interpolation in finite fields. Let $\mathbb{F}$ denote an arbitrary finite field (e.g., $\mathbb{F} = GF(p)$ for a prime p), and let $\mathcal{I} = \mathcal{M} = \mathbb{F}$. An index-hiding message encoding scheme IHME=$\{\mathsf{iEncode}, \mathsf{iDecode}\}$ with index space $\mathcal{I}$ and message space $\mathcal{M}$ is given by the following algorithms:

$\mathsf{iEncode}(\mathcal{P})$

The encoding of $\mathcal{P} = \{(i_1, m_1), \ldots, (i_n, m_n)\} \subseteq \mathcal{I} \times \mathcal{M} = \mathbb{F}^2$ is defined as the list $\mathcal{S} = (c_{n-1}, \ldots, c_0)$ of coefficients of the polynomial $p(x) = \sum_{k=0}^{n-1} c_k x^k \in \mathbb{F}[x]$ that interpolates all points in $\mathcal{P}$,that is, $p(i_j) = m_j$ for all $(i_j, m_j) \in \mathcal{P}$. Note that this polynomial exists uniquely [34], that is, the $\mathsf{iEncode}$ algorithm is deterministic.

$\mathsf{iDecode}(\mathcal{S}, i)$

On input $\mathcal{S} = (c_{n-1}, \ldots, c_0) \in \mathbb{F}^n$ and index $i \in \mathcal{I}$, this algorithm outputs $m = \sum_{k=0}^{n-1} c_k i^k$,that is, evaluation $p(i)$ of the polynomial $p(x) \in \mathbb{F}[x]$ induced by the coefficients in $\mathcal{S}$.

Observe that our IHME construction is size-preserving: The total number of field elements needed to represent messages $\{m_1, \ldots, m_n\}$ on the one side, and the encoding $\mathcal{S}$ of these messages on the other side, is the same. While correctness of the construction is obvious, its index-hiding property is assured by the following theorem [31]:

Theorem 1 (Security of IHME construction) *The proposed* IHME *scheme provides perfect index-hiding.*

Proof In case $I_0 = I_1$, the distribution of $\mathcal{A}$'s output obviously cannot depend on bit b. Assume therefore that $I_0 \neq I_1$. Since the messages encoded for indices in $I_b \setminus I_{1-b}$ are chosen randomly, then, regardless of whether $b = 0$ or $b = 1$, the coefficients seen by $\mathcal{A}$ are of a polynomial which is random subject to the constraint that for the indices in $I_0 \cap I_1$ its values are equal to messages $m_1, \ldots, m_k$ provided by $\mathcal{A}$. The distribution of the coefficients seen by $\mathcal{A}$ is therefore independent of b, and $\mathcal{A}$'s advantage in $\text{Expt}^{\text{ihide}}$ is thus 0. □

3.2.2 Interleaved IHME

We propose a method to generically compose IHME schemes from other IHME schemes. This transformation is motivated by the efficiency gain that can be achieved by such constructions.

In Sect. 3.2.1, we have seen implementations of IHME's iEncodeand iDecoderoutines. Their computational complexity is $O(n^2)$ and $O(n)$, respectively, with regard to a fixed finite field $\mathbb{F}$. In Sect. 5, we will see that these fields may become rather large, for example, $|\mathbb{F}| \approx 2^{1108}$, and IHME

will perform accordingly slow (although still in $O(n^2)$). In this section, we present an *interleaving technique* which allows to (generically) speedup IHME computations. Note that the algorithms remain in $O(n^2)$ and $O(n)$, respectively; it is rather the constant that is considerably reduced.

Consider, for instance, an IHME setting with $\mathbb{F} = GF(2^{1024})$ and $\mathcal{M} = \mathcal{I} = \mathbb{F} \cong \{0,1\}^{1024}$. Instead of encoding messages $m_1, m_2, \ldots \in \mathcal{M}$ over this field, one could split all messages m_i into, say, 8 chunks $m_{i,1}, \ldots, m_{i,8}$, each of length $1024/8 = 128$. Now, using IHME over field $\mathbb{F}' = GF(2^{128})$, all $m_{i,1}$ can be IHME-encoded into a structure $\mathcal{S}_1$, all $m_{i,2}$ can be independently encoded into a structure $\mathcal{S}_2$, and so on. The overall encoding is then $\mathcal{S} = (\mathcal{S}_1, \ldots, \mathcal{S}_8)$. A gain in efficiency is caused by the trade of super-linear costs of finite field arithmetics for linear costs of splitting the field elements.

We formalize the ideas of the preceding paragraph in a more general setting: We show how to generically compose IHME schemes from IHME schemes with smaller message sets.

Definition 4 (*Interleaved* IHME) Let IHME'={iEncode', iDecode'} be an index-hiding message encoding scheme over index set $\mathcal{I}'$ and message set $\mathcal{M}'$. For any $\nu \in \mathbb{N}$, the *ν-interleaved* index-hiding message encoding scheme IHME={iEncode,iDecode} with index space $\mathcal{I} = \mathcal{I}'$ and message space $\mathcal{M} = (\mathcal{M}')^\nu$ is constructed from IHME' as follows:

iEncode$(\mathcal{P})$

On input of $\mathcal{P} = \{(i_1, (m_{1,1}, \ldots, m_{1,\nu})), \ldots, (i_n, (m_{n,1}, \ldots, m_{n,\nu}))\} \subseteq \mathcal{I} \times \mathcal{M} = \mathcal{I}' \times (\mathcal{M}')^\nu$, the resulting encoding is the list $\mathcal{S} = (\mathcal{S}_1, \ldots, \mathcal{S}_\nu)$ of IHME' encodings

$$\mathcal{S}_k = \mathsf{iEncode}'(\{(i_j, m_{j,k})\}_{1 \le j \le n}) \qquad \text{for } 1 \le k \le \nu.$$

iDecode$(\mathcal{S}, i)$

On input of $\mathcal{S} = (\mathcal{S}_1, \ldots, \mathcal{S}_\nu)$ and index $i \in \mathcal{I}$, this algorithm outputs $m = (m_1, \ldots, m_\nu)$, where

$$m_k = \mathsf{iDecode}'(\mathcal{S}_k, i) \qquad \text{for } 1 \le k \le \nu.$$

Index-hiding security of interleaved IHME is established in [33] via a standard hybrid argument (with $\nu - 1$ intermediate steps), where in the i-th hybrid experiment index set I_1 is used for structures $\mathcal{S}_1, \ldots, \mathcal{S}_i$, and index set I_0 is used for structures $\mathcal{S}_{i+1}, \ldots, \mathcal{S}_\nu$ (cf. experiment $\mathrm{Expt}^{\mathsf{ihide}}$ in Fig. 2). The tightness factor obtained in the corresponding reduction is ν.

Theorem 2 (Security of interleaved IHME) *For any given index-hiding* IHME*' scheme and any $\nu \in \mathbb{N}$, the ν-interleaved scheme* IHME *constructed in Definition 4 is index-hiding as well. If* IHME*' is perfectly index-hiding, then so is* IHME.

3.2.3 Interleaved IHME *over the integers*

In Sect. 5, we will require an IHME scheme where message space $\mathcal{M}$ has the form $\mathcal{M} = [0, T-1]$, for a large $T \in \mathbb{N}$. Instead of choosing T to be a prime number and deploying the IHME scheme from Sect. 3.2.1 over $GF(T)$, in order to improve efficiency of iEncode and iDecode operations, we can also choose $T = \Pi^\nu$ to be the ν-th power of a prime Π and apply ν-interleaved IHME over $GF(\Pi)$. This intuition is now formalized.

Let Π be a prime and $\nu \in \mathbb{N}$. By iEncode$(\mathcal{P}, \Pi, \nu)$ we denote an IHME encoding of $\mathcal{P}$ with index space $\mathcal{I} = [0, \Pi - 1]$ and message space $\mathcal{M} = [0, \Pi^\nu - 1]$, that is, $\mathcal{P} \subseteq \mathcal{I} \times \mathcal{M}$. This scheme is obtained by combining the interpolation-based construction from Sect. 3.2.1 with Definition 4, and by exploiting existence of finite field $\mathbb{F} = GF(\Pi)$ and the natural and efficient bijections $[0, \Pi - 1] \to \mathbb{F}$ and $[0, \Pi^\nu - 1] \to \mathbb{F}^\nu$ (e.g., for the latter, the representation to base Π, i.e., $a \mapsto (a_0, \ldots, a_{\nu-1})$ such that $a = \sum_{k=0}^{\nu-1} a_k \Pi^k$). Analogously, by iDecode$(\mathcal{S}, \Pi, \nu, i)$ we denote the corresponding IHME decoding at index $i \in [0, \Pi - 1]$.

4 Contact discovery schemes

In this section, we introduce contact discovery schemes (CDS). We discuss the syntax and formalize our security model.

4.1 Execution model for CDS

We assume that CDS users communicate over secure channels [2,29] that are established independently of our CDS protocols. This assumption allows us to focus on the core functionality of CDS, namely on the private discovery of shared contacts, and ignore, for example, impersonation attacks on the communication channels.

Note that many anticipated application domains of CDS, including online social networks and collaborative group applications, already provide authentication mechanisms for their users that can be used to setup secure channels. In this case, the level of trust required in, for example, the social network provider, is nonetheless limited to the sole authentication of users. In other words, the provider is not trusted with users' private information. For the case that users prefer to communicate without any external setup, we propose

that they deploy key agreement and channel establishment protocols that do not rely on a PKI, but solely on a one-time authentic exchange of public keys (e.g., using [29]).

In addition to the existence of secure channels between users, we further assume that users have individual identities (i.e., strings $\mathsf{id} \in \{0,1\}^*$) under which they are known to each other. For instance, in the PKI-free model proposed above, these identities can be equal to users' public keys. If the secure channels are established by other means, then user identities have to be defined accordingly.

4.2 Syntax of CDS

Formally, a CDS consists of algorithms to initialize users, to issue contact certificates to other users, and to run the actual contact discovery protocol.

Definition 5 (*Contact discovery scheme*) A contact discovery scheme is defined as a set $\mathsf{CDS} = \{\mathsf{InitUser}, \mathsf{AddContact}, \mathsf{Discover}\}$ of three efficient algorithms and protocols:

$\mathsf{InitUser}(1^\kappa)$

This algorithm is executed once by each user U. On input of security parameter 1^κ, it initializes U's private certification key $U.\mathsf{sk}$.

$\mathsf{AddContact}(U, \mathsf{id})$

This algorithm is executed by user U, on input the identity id of a user V. User U uses its certification key $U.\mathsf{sk}$ to certify a given social relation to V, by issuing V a corresponding *contact certificate* $\mathsf{cc}_{U\to V}$.

Note that we model contact certification as a unidirectional process: A mutual certification requires two executions of AddContact algorithm.

$\mathsf{Discover}(U \leftrightarrow U')$

This protocol is executed between users $U \neq U'$ to discover their common contacts. User U's private input is $(\mathsf{CL}_U, \mathsf{partner}_U, r_U)$, where contact list CL_U is a set of pairs of the form $(V, \mathsf{cc}_{V\to U})$, for some users V, $\mathsf{partner}_U$ is the identity of the supposed protocol partner, and $r_U \in \{\mathsf{init}, \mathsf{resp}\}$ specifies the role of the session as initializer or responder. All values $\mathsf{cc}_{V\to U}$ are assumed to be contact certificates previously obtained as output of $\mathsf{AddContact}(V, U)$. Note that certification key $U.\mathsf{sk}$ is not input to Discover protocol: It is only needed for the certification of contacts in AddContact algorithm. Private input of user U' is $(\mathsf{CL}_{U'}, \mathsf{partner}_{U'}, r_{U'})$, defined analogously.

The protocol shall detect the set of users V for which both participants provide corresponding contact certificates, $\mathsf{cc}_{V\to U}$ and $\mathsf{cc}_{V\to U'}$, respectively. This *shared contact list* is denoted by SCL.

Users keep track of the state of created Discover protocol sessions π through session variables that are initialized by setting $\pi.\mathsf{state} \leftarrow \mathsf{running}$ and $\pi.\mathsf{SCL} \leftarrow \emptyset$, and by initializing $\pi.\mathsf{CL}$ and $\pi.\mathsf{partner}$ from the session parameters. In addition, $\pi.\mathsf{id}$ is set to the own identity. After the protocol completes, $\pi.\mathsf{state}$ is updated to accepted and $\pi.\mathsf{SCL}$ holds a (possibly empty) set of user identities.

Roughly speaking, CDS is correct if sessions executed between users (without interference of adversaries) reveal the set of common contacts:

Definition 6 (*Correctness of* CDS) Suppose that users U and U' interact in a Discover protocol on input $(\mathsf{CL}_U, U', \mathsf{init})$ and $(\mathsf{CL}_{U'}, U, \mathsf{resp})$, respectively. Let π and π' denote the corresponding sessions. Let $\mathsf{CL}_\cap$ denote the set of users (contacts) V that appear in both CL_U and $\mathsf{CL}_{U'}$. The CDS is *correct* if both sessions accept with $\pi.\mathsf{SCL} = \pi'.\mathsf{SCL} = \mathsf{CL}_\cap$ (with all but negligible probability).

4.3 A note on contact revocation

Our definition of CDS from Sect. 4.2 can be easily extended to include a mechanism for *contact revocation*, allowing any user V to revoke contact certificates $\mathsf{cc}_{V\to U}$ that V previously issued to a user U. That is, contact revocation allows users to indicate that they do no longer consider certain types of social relationships, for example, friendship to other users, as valid.

In our previous work [8], a contact revocation mechanism was explicitly defined as part of CDS and realized using *contact revocation lists (CRL)* as follows. Each user V is assumed to publicly maintain a list of its revoked contacts. In each execution of a $\mathsf{Discover}(U \leftrightarrow U')$ session, users U and U' take as additional input CRLs of all users in their respective contact lists CL_U and $\mathsf{CL}_{U'}$. This allows U to check whether U' is revoked by any of the users in CL_U and U' to check whether U is revoked by any of the users in $\mathsf{CL}_{U'}$. The protocol guarantees that any user V that has revoked U (or U') but still has social relationship with U' (or U) will remain hidden and not appear in the output shared contact list $\pi.\mathsf{SCL}$ of users U and U'.

The use of public contact revocation lists, however, reveals information about social relationships that revoking user V had in the past—this is something what, depending on the type of social relationship, user V may wish to keep private. On the other hand, social relationships are often influenced by the mutual behavior of the users. Therefore, if V could privately revoke U, the latter could still make the existence

of their relationship public, possibly to spite V, by disclosing certificates $\mathsf{cc}_{U\to V}$ and $\mathsf{cc}_{V\to U}$.

4.4 Security model for CDS

We introduce a security model for private contact discovery by describing the capabilities of the adversary and by defining an appropriate experiment for the security goal of *contact-hiding*.

4.4.1 Adversarial queries

We model adversary $\mathcal{A}$ as a probabilistic algorithm that runs in polynomial time and has the following queries at its disposal to interact with protocol participants. By $\mathcal{U}^h$, we denote the set of honest users in the system.

RequestCC(U, V)

Contact certificate $\mathsf{cc}_{U\to V}$ issued by user $U \in \mathcal{U}^h$ for user V is given to the adversary. Note that this query corresponds to AddContact algorithm and models the possibility of selective contact corruptions.

Discover(U, CL, partner, r)

User $U \in \mathcal{U}^h$ initiates a new session π of the Discover protocol, using all available certificates received from users listed in $\mathsf{CL} \subseteq \mathcal{U}^h$, and using $\mathsf{partner} \in \mathcal{U}\setminus\mathcal{U}^h$ and $r \in \{\mathsf{init}, \mathsf{resp}\}$ as further session parameters. Note that partner may not be an honest user; this restriction models assumed deployment of secure channels between (honest) users that execute the Discover protocol (cf. Sect. 4.1). This query returns a first protocol message M (if available).

Send(π, M)

Message M is delivered to session π. After processing M, the output (if any) is given to $\mathcal{A}$. This query is ignored if π is not waiting for input.

Reveal(π)

This query returns (π.state, π.SCL).

Note that we do not provide a query for user corruption. If defined, it would reveal user's secret key U.sk and the set of stored certificates $\mathsf{cc}_{V\to U}$. We argue that Corrupt queries are usually only needed to model forward secrecy in key establishment protocols. Indeed, in the CDS setting, as the corruption of users would reveal all their contacts anyway, there is close to nothing left to protect against, after a user corruption has taken place.

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

(a) the experiment creates a set of n users, denoted by $\mathcal{U} = \{U_1, \ldots, U_n\}$. Adversary $\mathcal{A}$ specifies a set $\mathcal{U}^c \subseteq \mathcal{U}$ of initially corrupted users. Let $\mathcal{U}^h = \mathcal{U}\setminus\mathcal{U}^c$. $\mathsf{InitUser}(1^\kappa)$ 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 $\mathsf{AddContact}(U, V)$ algorithm. For all $U \in \mathcal{U}^c$ the adversary sets up all parameters himself. He then specifies a list $\mathcal{L} \subseteq \mathcal{U}^h \times \mathcal{U}^c$, and for all $(U, V) \in \mathcal{L}$ algorithm $\mathsf{AddContact}(U, V)$ is run and the respective certificate $\mathsf{cc}_{U\to V}$ is given to $\mathcal{A}$.

(b) $\mathcal{A}(1^\kappa)$ interacts with all (honest) users using the queries from Section 4.4.1; at some point, $\mathcal{A}$ outputs a tuple $(U^*, \mathsf{CL}^*_0, \mathsf{CL}^*_1, \mathsf{partner}^*, r^*)$, where $U^* \in \mathcal{U}^h$, $\mathsf{CL}^*_0, \mathsf{CL}^*_1 \subseteq \mathcal{U}^h$ with $|\mathsf{CL}^*_0| = |\mathsf{CL}^*_1|$, $\mathsf{partner}^*$ is any user id (in $\mathcal{U}$), and $r^* \in \{\mathsf{init}, \mathsf{resp}\}$. Let $\mathcal{D}^* = \Delta(\mathsf{CL}^*_0, \mathsf{CL}^*_1)$ denote the symmetric difference of the sets CL^*_0 and CL^*_1, i.e., $\mathcal{D}^* = (\mathsf{CL}^*_0 \setminus \mathsf{CL}^*_1) \cup (\mathsf{CL}^*_1 \setminus \mathsf{CL}^*_0)$.

(c) the experiment invokes a $\mathsf{Discover}(U^*, \mathsf{CL}^*_b, \mathsf{partner}^*, r^*)$ session π^* (and provides all needed certificates)

(d) $\mathcal{A}$ continues to interact via queries (including on session π^*), until it terminates and outputs bit b'

(e) the output of the experiment is b' if the following condition holds; otherwise the output is 0:

(1) for no user $V \in \mathcal{D}^*$, a $\mathsf{RequestCC}(V, \pi^*.\mathsf{id})$ or $\mathsf{RequestCC}(V, \pi^*.\mathsf{partner})$ query has been posed, or a pair $(V, \pi^*.\mathsf{id})$ or $(V, \pi^*.\mathsf{partner})$ is contained in $\mathcal{L}$, i.e., the adversary did not request a contact certificate for id^* or $\mathsf{partner}^*$ issued by any user in set $\mathcal{D}^*$.

Fig. 3 Contact-hiding experiment

4.4.2 Privacy goal: Contact-Hiding (CH)

Informally, the property of Contact-Hiding (CH) protects users from disclosing non-matching contacts to other participants. We model CH security by means of an experiment, following the indistinguishability approach. The goal of the adversary is to decide which of two contact lists, CL^*_0 or CL^*_1, is used in a challenge Discover session π^*. The adversary can invoke any number of independent Discover sessions, and perform Reveal and RequestCC queries at will.

Definition 7 (*Contact-hiding security*) For a CDS, let $\mathrm{Expt}^{\mathsf{ch},0}$ and $\mathrm{Expt}^{\mathsf{ch},1}$ be the experiments specified in Fig. 3. The advantage of adversary $\mathcal{A}$ is defined as

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

We say that CDS is *contact-hiding* if $\mathrm{Adv}^{\mathsf{ch}}_{\mathsf{CDS},\mathcal{A}}$ is negligible in κ (for all n polynomially dependent on κ), for all efficient adversaries $\mathcal{A}$.

In Fig. 3, condition (1) excludes a trivial attack on contact-hiding: it prevents $\mathcal{A}$ from asking for contact certificates issued by users $V \in \mathcal{D}^*$ for a user $U' \in \mathcal{U}$, to simulate a protocol session on behalf of U' with challenge session π^*, and to decide about bit b from resulting SCL.

Remark 1 (Structure of experiment $\mathrm{Expt}^{\mathsf{ch}}$*)* The experiment in Fig. 3 is composed of two phases. The first is a setup phase (line (a)) where all (honest) users are initialized. In the second phase (lines (b)–(d)), sessions with these users are attacked. We admit that the model could be more general, for example, we could allow the adversary to setup fresh honest users *after* having communicated with other ones. We opted for the simpler model for clarity of illustration, but expect that our protocols stay secure even when analyzed in a more adaptive model.

Remark 2 (Variant of contact-hiding security) Observe that, in experiment $\mathrm{Expt}^{\mathsf{ch},b}$, we do not pose requirements on sets $\mathsf{CL}^*_0, \mathsf{CL}^*_1$, except that we demand $|\mathsf{CL}^*_0| = |\mathsf{CL}^*_1|$. It is easily seen by a hybrid argument that a modified definition of contact-hiding security with the additional constraint $|\mathsf{CL}^*_0 \backslash \mathsf{CL}^*_1| = 1 = |\mathsf{CL}^*_1 \backslash \mathsf{CL}^*_0|$ is equivalent to the one from Definition 7. In this case, we always have $|\mathcal{D}^*| = 2$.

5 An RSA-based contact discovery scheme

We present a concrete CDS construction as a first solution to the challenge of private contact discovery. We give a detailed proof of its security and privacy according to the model described in Sect. 4.4, in the random oracle model, assuming hardness of the RSA problem on safe moduli (cf. Definition 1). In addition, in Sect. 6, we analyze and optimize the performance of our scheme when considering its deployment in practice.

We start by giving a high-level overview over the protocol's design. Our scheme is built around a specific two-message identity-based key agreement protocol (IBKA) that is specially tweaked such that exchanged messages do not leak information about the corresponding certification authority. In our CDS, each user V runs its own CA, and the IBKA's credentials for users U are used as contact certificates $\mathsf{cc}_{V \to U}$. When executing the Discover protocol, users U, U' basically run several IBKA instances in parallel, one for each of their respective contacts. Exactly for the matching contacts, we expect the key agreements to succeed. The IHME primitive is used to bundle the individual messages of the IBKA instances into one large message, where the respective contacts are used as indices; note that their secrecy is ensured by IHME's index-hiding property. In a second protocol round, confirmation messages for the individual contacts are computed and transmitted (again via IHME) to allow the final assignment of the list of shared contacts SCL. A more detailed description of the protocol and its components follows in the next section.

5.1 Protocol specification

We assume that all interactions between users during AddContact and Discover sessions are protected by secure channels, as motivated in Sect. 4.1. However, we claim that our scheme would also be secure if Discover sessions would be run over a channel that guarantees authenticity but not privacy, since we mainly require that the corresponding channel mutually authenticates the identities U and U' of the two users involved in Discover session and prevents them from claiming different identities within the CDS scheme.

Let $\ell = \ell(\kappa)$ be polynomially dependent on security parameter κ. We use the perfect IHME scheme from Sect. 3.2.1 as a building block, where IHME is defined over finite field $\mathbb{F} = GF(T)$, for the smallest prime number T satisfying $T > 2^{\kappa+\ell}$. Moreover, let $H : \{0,1\}^* \to [0, T-1]$ and $H_n : \{0,1\}^* \to \mathbb{Z}_n$ be hash functions, for any (RSA modulus) $n \in \mathbb{N}$. The three algorithms of our CDS protocol are defined as follows:

InitUser

The setup routine run by each user U generates safe RSA parameters $(n, e, d) \leftarrow_R \mathsf{SRSA-GEN}(1^\kappa)$ (cf. Definition 1) and picks an element $g \in \mathbb{Z}_n^\times$ such that $\mathbb{Z}_n^\times = \langle -1 \rangle_n \times \langle g \rangle_n$ (i.e., $\mathrm{ord}_n(g) \approx n/2$, cf. Sect. 3.1). User U keeps certification key $U.\mathsf{sk} \leftarrow (n, g, e, d)$ secret. We assume in the following that the length in bits of RSA modulus n is κ.

AddContact

This algorithm is executed by user U. It takes as input certification key $U.\mathsf{sk} = (n, g, e, d)$ and identifier $\mathsf{id} \in \{0,1\}^*$ of a user V. User V receives contact certificate $\mathsf{cc}_{U \to V} = (n, g, e, \sigma_V)$, where σ_V is the RSA signature $\sigma_V = H_n(\mathsf{id})^d \bmod n$ on the full-domain hash of id.

Discover

The contact discovery protocol is executed between two users, U and U', on inputs $(\mathsf{CL}_U, \mathsf{partner}_U, r_U)$ and $(\mathsf{CL}_{U'}, \mathsf{partner}_{U'}, r_{U'})$, respectively. The protocol is specified in detail in Fig. 4. Note that the loop in line 2 goes over all contacts in the input lists CL_U and $\mathsf{CL}_{U'}$, respectively. We do not assume any specific order for CL_U and $\mathsf{CL}_{U'}$. That is, certificates (n, g, e, σ_V) drawn on both sides (by U and U') may refer to the same user

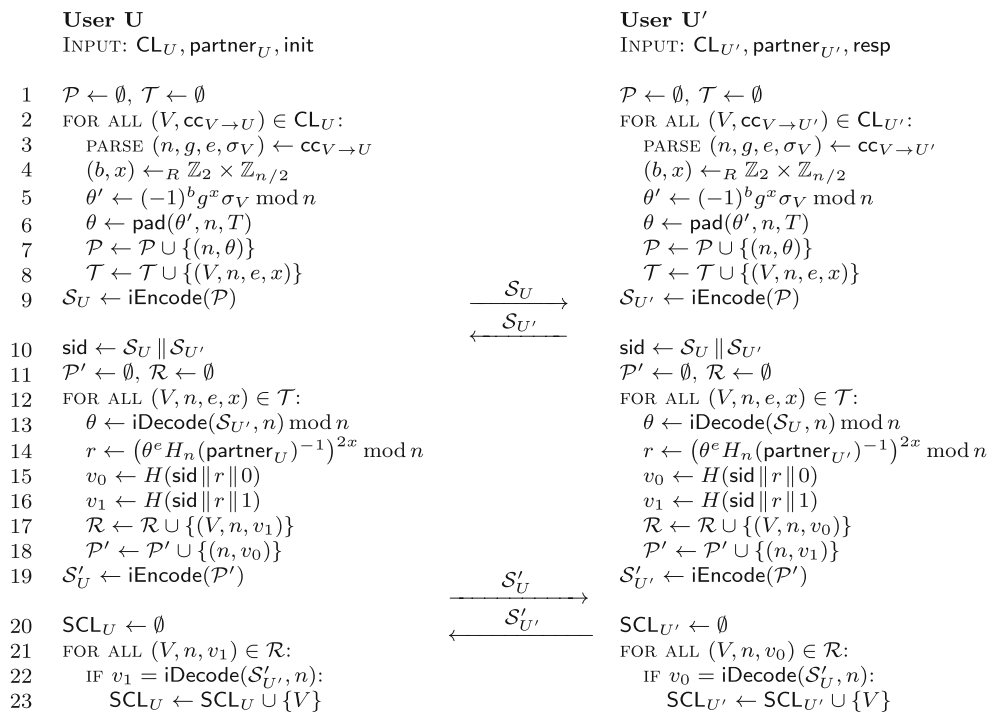

Fig. 4 RSA-based Discover protocol for private contact discovery

V or to two different users. By $\mathsf{pad}(\theta', n, T)$ (in line 6) we denote a probabilistic algorithm that maps its argument $\theta' \in [0, n-1]$ to a random element θ in interval $[0, T-1]$ such that $\theta = \theta' (\mathrm{mod}\, n)$. For concreteness, let pad map θ' to $\theta = \theta' + kn$, where $k \leftarrow_R [0, \lfloor T/n \rfloor - 1]$. Protocol's correctness follows from IHME's correctness and from $r_U = g^{2ex_U x_{U'}} = r_{U'}$ (cf. line 14), which holds for all contacts for which both participants present valid certificates, that is, deploy the same RSA parameters (n, g, e):

$$\begin{aligned} r_U &= \left((\theta_{U'})^{e_U} H_{n_U}(\mathsf{partner}_U)^{-1}\right)^{2x_U} \\ &= \left((\theta'_{U'})^{2e_U} H_{n_U}(\mathsf{partner}_U)^{-2}\right)^{x_U} \\ &= \left((g_{U'})^{2e_U x_{U'}} (\sigma_{U'})^{2e_U} H_{n_U}(\mathsf{partner}_U)^{-2}\right)^{x_U} \\ &= \left((g_{U'})^{2e_U x_{U'}} H_{n_{U'}}(\mathsf{id}_{U'})^2 H_{n_U}(\mathsf{partner}_U)^{-2}\right)^{x_U} \\ &= (g_{U'})^{2e_U x_U x_{U'}} \qquad (\mathrm{mod}\, n_U) \end{aligned} \tag{1}$$

Note that in this computation, we assume that the hash value $H_n(\mathsf{partner})$ is invertible (mod$n$) for all used pseudonyms partner. Indeed, as we argue in Sect. 3.1, the case that $H_n(\mathsf{partner})$ hits an element from $\mathbb{Z}_n \backslash \mathbb{Z}_n^\times$, in which the protocol would fail, occurs only with negligible probability.

Our protocol is loosely based on the identity-based key agreement scheme by Okamoto and Tanaka [36,12], its conversion by Jarecki et al. [23] to the setting of affiliation-hiding authentication, and the latter's variant by Manulis et al. [31] that detects not just a simple match of affiliations (i.e., of RSA parameter sets) but instead the set of all matching affiliations between two users.

Specifically, the principal idea behind our protocol is to compute for each contact a 'session key' r using Okamoto's technique from [36,12] in the RSA setting, that is, by exchanging values θ of the form $\theta = g^x \sigma_{\mathsf{id}}$ (where $\sigma_{\mathsf{id}} = H(\mathsf{id})^d$) and computing $r = (\theta^e / H(\mathsf{partner}))^y$ (lines 4, 5, and 14). Jarecki et al. [23] introduced to this setting the special choice of base element $g \in \mathbb{Z}_n^\times$ such that $\mathbb{Z}_n^\times \cong \langle -1 \rangle_n \times \langle g \rangle_n$. This property of g together with the blinding of g^x with $(-1)^b$ (line 5) makes all computed values θ' close to uniformly distributed in interval $[0, n-1]$. Jarecki et al. also introduced the padding function pad (line 6) which sends uniformly distributed elements in $[0, n-1]$ to elements (almost) uniformly distributed in $[0, T-1]$ (without changing residuosity mod n, i.e., without affecting Okamoto's protocol). This essentially hides RSA moduli n from (even active) observers, that is, the identities of contacts remain protected. The idea to execute several of these adapted key agreement protocols in parallel—one for each contact in contact list CL—and to transport the corresponding messages via IHME primitive (using RSA moduli n as indices, cf. lines 7, 9, 13, 18, 19, and 22) was first proposed in [31].

In respect to the security of established 'session key' r (line 14), Jarecki et al. claim that their protocol offers key

indistinguishability with forward secrecy under the safe RSA assumption, if hash function H_n can be modeled as random oracle. However, the corresponding proof is flawed [38, Section 2.5.1]. Independently of Jarecki et al., Gennaro et al. [18,17] analyze Okamoto's protocol in a slightly different setting: In their variant, group element g is chosen to be a generator of $QR(n)$. Under this condition, however, $\mathbb{Z}_n^\times = \langle -1 \rangle_n \times \langle g \rangle_n$ does not hold (recall that this property is essential in our protocol). Here, we abstain from giving an adaption of the proof found in [18,17] to the setting where g is chosen according to our requirements. Nevertheless, we have verified that the proof is convertible to our setting in a sound way; all necessary modifications are worked out in [38, Appendix A, see also Section 2.5.1].

5.2 Security and privacy analysis

Our CDS construction guarantees contact-hiding, as formalized in Sect. 4.4.

Theorem 3 *Our RSA-based* CDS *scheme is contact-hiding under the RSA assumption on safe moduli, in the random oracle model.*

Proof Besides to the experiments $\text{Expt}^{\text{ch},b}$ from Fig. 3 (including the modification proposed in Remark 2), we will refer to a set of auxiliary games (experiments) that will help us to prove that our CDS scheme is contact-hiding. For each of these games $\mathbf{G}$, let $W = \Pr[\mathbf{G}(\kappa, n) = 1]$ denote the probability that $\mathbf{G}$'s execution results in the output of 1. We will parametrize these games with a bit b and denote this with a superscript, for example, $\mathbf{G}^b$.

Fix adversary $\mathcal{A}$ and parameters $\kappa, n = n(\kappa)$. We assume that, for any protocol session π, session variable $\pi.\mathsf{sid}$ holds the value computed in line 10 in Fig. 4, after receiving first protocol message $\mathcal{S}$. Consider the following games:

$\mathbf{G}_0^b$. This game is identical to $\text{Expt}^{\text{ch},b}_{\text{CDS},\mathcal{A}}(\kappa, n)$.

Our goal is to show that $|W_0^0 - W_0^1|$ is bounded by a negligible function. This holds trivially if the adversary violates condition (1) in Fig. 3, as this would directly imply $W_0^0 = W_0^1 = 0$. We hence assume in the following that adversary complies with the named condition.

$\mathbf{G}_1^b$. Game $\mathbf{G}_1^b$ is like Game $\mathbf{G}_0^b$, except that the simulation is aborted if, for any user $U \in \mathcal{U}$ and any two sessions run by U, a collision of session ids occurs, that is, if there exist sessions $\pi \neq \pi'$ with $(\pi.\mathsf{id}, \pi.\mathsf{sid}) = (\pi'.\mathsf{id}, \pi'.\mathsf{sid})$.

Observe that session ids, as assigned in line 10 of the protocol, contain values θ that are freshly and independently picked for each session and carry about $\log_2 T > \kappa + \ell$ bits of entropy each. By the birthday paradox, the probability of collisions of session ids to occur is bounded by $q_s^2/T < q_s^2/2^{\kappa+\ell}$ (which is negligible), where q_s denotes the total number of posed Discover queries. Hence, $|W_0^b - W_1^b|$ is negligible.

$\mathbf{G}_2^b$. Recall that for the challenge session π^*, we have that $\pi^*.\mathsf{id}$ and $\pi^*.\mathsf{partner}$ identify users in $\mathcal{U} = \{U_1, \ldots, U_n\}$. Game $\mathbf{G}_2^b$ is like Game $\mathbf{G}_1^b$, except that the simulator makes a priori guesses on the pseudonyms $(\mathsf{id}^*, \mathsf{id}') \in \mathcal{U}^h \times \mathcal{U}^c$ that will be $\pi^*.\mathsf{id}$ and $\pi^*.\mathsf{partner}$, respectively. If one of these guesses later turns out to be incorrect, i.e., if adversary demands challenge session be run for other users, then the experiment outputs a random bit (i.e., the simulation aborts).

Note that the probability that the simulator makes a correct guess is lower-bounded by $1/n^2$. Under the condition that the guess is wrong, we have $W_2^0 = W_2^1 = 1/2$. Otherwise, we have $W_1^b = W_2^b$. All in all, we obtain $|W_1^0 - W_1^1| \leq n^2 |W_2^0 - W_2^1|$.

$\mathbf{G}_3^b$. Game $\mathbf{G}_3^b$ is like Game $\mathbf{G}_2^b$, except that the simulator makes an a priori guess on user U^b such that $\{U^b\} = \mathsf{CL}_b^* \backslash \mathsf{CL}_{1-b}^*$, out of a set of size $|\mathcal{U}^h| \leq n$. Note that we assume the modification to experiment $\text{Expt}^{\text{ch},b}$ that is proposed in Remark 2. If the guess on U^b later turns out to be incorrect, then the experiment outputs a random bit (i.e., the simulation aborts).

Note that the probability that the simulator makes a correct guess is lower-bounded by $1/n$. Under the condition that the guess is wrong we have $W_3^0 = W_3^1 = 1/2$. Otherwise, we have $W_2^b = W_3^b$. All in all we obtain $|W_2^0 - W_2^1| \leq n |W_3^0 - W_3^1|$.

$\mathbf{G}_4^b$. Let r^* be the value r computed in challenge session π^* for contact U^b (line 14). Game $\mathbf{G}_4^b$ is like Game $\mathbf{G}_3^b$, except that all confirmation messages v (lines 15 and 16) that are computed in session π^* and all sessions π' with $\pi^*.\mathsf{sid} = \pi'.\mathsf{sid}$ in dependence on r^* are consistently replaced by random values in the range $[0, T-1]$.

Observe that named confirmation tags are computed from r^* by hashing this value, using hash function H. By the random oracle model, adversary can detect the difference between Games $\mathbf{G}_3^b$ and $\mathbf{G}_4^b$ only by querying (a string that contains) r^* to this oracle. However, the probability of this to happen can be bounded by $\text{Succ}^{\text{srsa}}_{\text{SRSA}-\text{GEN}}$ (cf. Definition 1), as proven in [18,17] and discussed in the following:

By embedding an SRSA challenge (n, e, z) into parameters n, g, e of user U^b and into (the hash value of) pseudonym id', a solution to the challenge can be computed from any hash query on r^*. Moreover, the actions of all (honest) users continue to be simulatable, with the exception that for user id' a $\mathsf{RequestCC}(U^b, \mathsf{id}')$ query cannot be processed. This behavior, however, is compliant with rule (1) in experiment $\mathrm{Expt}^{\mathsf{ch},b}$. For further details on the reduction, we refer to the analysis by Gennaro et al. [17] and to [38, Appendix A]. We conclude that, for a constant c,

$$|\Pr[W_4^b] - \Pr[W_3^b]| \leq c \cdot \mathsf{Succ}^{\mathsf{srsa}}_{\mathsf{SRSA-GEN},\mathcal{A}'}(\kappa)$$

for an adversary $\mathcal{A}'$.

$\mathbf{G}_5^b$. Game $\mathbf{G}_5^b$ is like Game $\mathbf{G}_4^b$, except that value θ for contact U^b, as computed by session π^* in line 6, is replaced by a random element: $\theta \leftarrow_R [0, T-1]$.

Observe that, in the protocol, θ is exclusively used to compute r^* in line 14 (and, correspondingly, in sessions π' with $\pi^*.\mathsf{sid} = \pi'.\mathsf{sid}$). As we decoupled this value from the remaining simulation in Game $\mathbf{G}_4^b$, the difference between W_4^b and W_5^b is bounded by the statistical difference of the two methods to generate θ. As discussed in Sect. 5.1 and in [23], this difference is negligible.

$\mathbf{G}_6^b$. Game $\mathbf{G}_6^b$ is like Game $\mathbf{G}_5^b$, except that, in session π^*, we replace index n, used for IHME encoding value θ for contact U^b, by a fixed (unused) index, for example, $n = 0$ (cf. lines 7 and 9).

The change introduced in Game $\mathbf{G}_6^b$ corresponds to the security experiment of IHME's index-hiding property (cf. Fig. 2): As θ is chosen uniformly from $[0, T-1]$, which coincides with IHME's message space $\mathcal{M}$, we can readily construct an IHME adversary $\mathcal{A}'$ from any distinguisher between Games$\mathbf{G}_5^b$ and $\mathbf{G}_6^b$. In the reduction, the set of moduli of the contacts in CL_b^* is assigned to index set I_0, while the set of moduli of the contacts in $\mathsf{CL}_b^* \backslash \{U^b\}$ together with index $n = 0$ is assigned to I_1. As messages M corresponding to the indices in $I_0 \cap I_1$, the θ-values for the contacts in $\mathsf{CL}_b^* \backslash \{U^b\}$ are taken without modification. We conclude that

$$|\Pr[W_6^b] - \Pr[W_5^b]| \leq \mathrm{Adv}^{\mathsf{ihide}}_{\mathsf{IHME},\mathcal{A}'}(\kappa)$$

for an adversary $\mathcal{A}'$.

In particular, as we deploy the perfect IHME scheme from Sect. 3.2.1, we have $\Pr[W_6^b] = \Pr[W_5^b]$.

As, in Game $\mathbf{G}_6^b$, the existence of a session π' such that $(\pi'.\mathsf{id}, \pi'.\mathsf{partner}) = (\pi^*.\mathsf{partner}, \pi^*.\mathsf{id})$ is impossible (due to restriction '$\mathsf{partner} \notin \mathcal{U}^h$' in $\mathsf{Discover}$ query in Sect. 4.4.1), verification tags v assigned by session π^* for contact U^b are random and completely independent from U^b and the rest of the simulation (recall the changes introduced in Game $\mathbf{G}_4^b$). In particular, (a) the tag v that π^* sends in lines 18 and 19 contains no information about contact U^b, (b) IHME structure $\mathcal{S}'$ that π^* sends in line 19 leaks no information about G^b (by an argument similar to the one in the hop to Game $\mathbf{G}_6^b$), and (c) a $\mathsf{Reveal}(\pi^*)$ query unveils no information about U^b, as the test in line 22 corresponding to contact U^b will pass only with negligible probability $1/T < 2^{-(\kappa+\ell)}$. Recall that the protocol's first message, $\mathcal{S}$, sent in line 9, does not leak information about group G^b since the hop to Game $\mathbf{G}_6^b$

We observe that the adversary cannot efficiently distinguish experiments $\mathbf{G}_6^0$ and $\mathbf{G}_6^1$, that is, we have $W_6^0 \approx W_6^1$. Putting everything together, we note that the advantage $\mathrm{Adv}^{\mathsf{ch}}_{\mathsf{CDS},\mathcal{A}}(\kappa, n) = |W_0^0 - W_0^1|$ is bounded by a negligible function, provided that the required assumptions hold. □

6 An optimized CDS construction

In this section, we analyze practical performance of our CDS scheme from Sect. 5. In addition, we propose several modifications that speedup the most important operations. In particular, we not only replace the IHME scheme deployed in the original $\mathsf{Discover}$ protocol by the optimized version from Sect. 3.2.3, but our improvements also cover unrelated aspects, for example, optimize bandwidth consumption and key sizes as well.

6.1 Optimized Discover protocol

Let $H : \{0,1\}^* \to \{0,1\}^\ell$ denote a hash function (in contrast to $H : \{0,1\}^* \to [0, T-1]$ in Sect. 5.1), where $\ell = \ell(\kappa)$ is polynomially dependent on security parameter κ. Let Π be a prime slightly greater than 2^ℓ and let $\nu \in \mathbb{N}$ be minimal such that $\Pi^\nu > 2^{\kappa+\ell}$. A typical configuration in practice would be $(\kappa, \ell, \nu) = (1024, 80, 14)$, where $\kappa = 1024$ means that RSA moduli should be of length κ. IHME's $\mathsf{iEncode}$ and $\mathsf{iDecode}$ routines are now defined in respect to index set $\mathcal{I} = GF(\Pi)$ and message set $\mathcal{M} = GF(\Pi)^\nu$. Leaving $\mathsf{InitUser}$ and $\mathsf{AddContact}$ algorithms and the specification of hash function family $H_n : \{0,1\}^* \to \mathbb{Z}_n$ as in Sect. 5.1, our optimized version of the CDS protocol from Fig. 4 is depicted in Fig. 5.

The principal enhancement of the new design over the scheme from Sect. 5.1 is the deployment of the more efficient interleaved IHME scheme (in lines 10, 14, 20, and 23). In the original protocol, all messages that are exchanged in the two communication rounds are, when IHME-encoded, considered elements of finite field $\mathbb{F} = GF(T)$, where $T > 2^{\kappa+\ell}$. As we typically have $(\kappa, \ell) = (1024, 80)$, field arithmetic performs rather slow. In contrast, in Fig. 5, first-round messages $\theta \in [0, \Pi^\nu - 1]$ are encoded over a (much smaller)

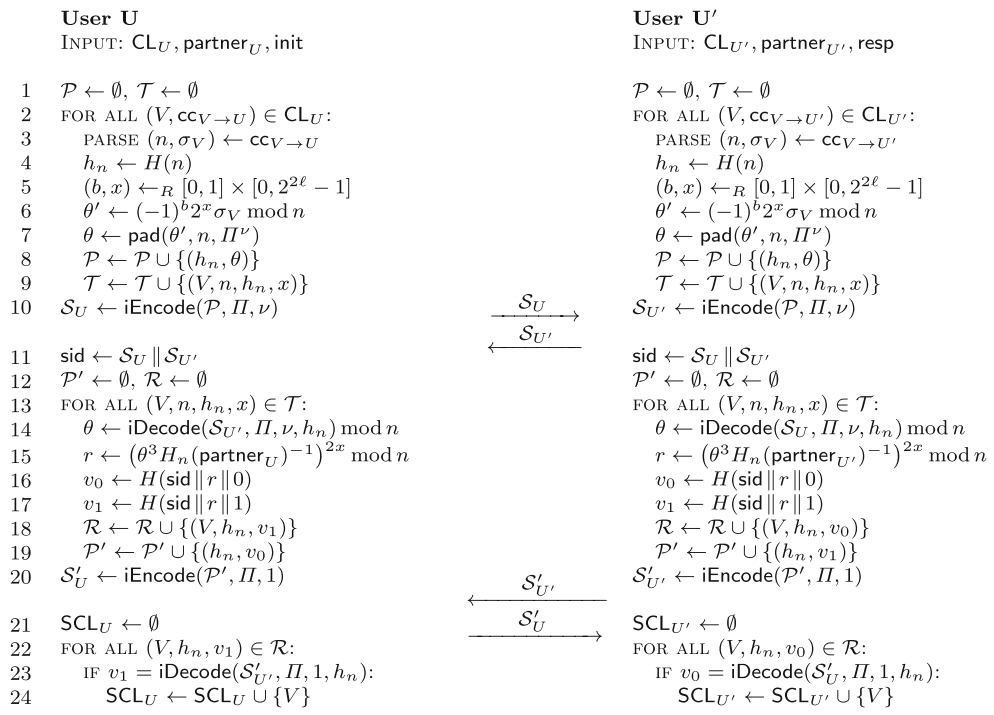

Fig. 5 Optimized RSA-based Discover protocol

field of $\Pi \approx 2^\ell$ elements, using the ν-interleaved technique. Note that careful choice of Π, for example, of low Hamming weight, allows impressively fast implementations of field arithmetics [6, Section 2.2.6]. Considering the second round messages, in the protocol from Sect. 5.1, the per-contact key confirmation messages are also of length $\kappa + \ell$, but actually ℓ bits would suffice for a secure scheme. In our new protocol, confirmation messages are shortened to this more reasonable level and encoded using IHME, again over the field of $\Pi \approx 2^\ell$ elements. Both these optimizations are expected to lead to a considerable boost of computational efficiency and bandwidth consumption, when compared to a naïve implementation of the protocol.

A consequence of the switch to a smaller field is that also deployed IHME indices have to be chosen from a smaller set (see Sect. 3.2.3). While, in Sect. 5.1, RSA moduli n serve directly as contact indices, in our new protocol the set of possible indices is reduced to the elements of $[0, \Pi - 1]$, which is much too small for allowing a direct embedding of moduli n. We solve this problem by hashing public contact parameters into this smaller set, using collision-resistant hash function H (line 4).

Another optimization is the fixed choice of RSA parameters g and e. While assigning $e = 3$ is a well-established technique to make the costs of the exponentiation by e vanish, it is shown in [33, Lemma 2] that $g = 2$ is a valid base element in our setting (i.e., $\mathbb{Z}_n^\times = \langle -1 \rangle_n \times \langle g \rangle_n$) whenever safe RSA modulus $n = pq$ is such that $p = 3 \bmod 8$ and $q = 7 \bmod 8$.

In contrast to Sect. 5, in our optimized protocol, ephemeral exponents x (see lines 5, 6 and 15) are not chosen from $\mathbb{Z}_{n/2}$ (where n is an RSA modulus), but from much smaller range $[0, 2^{2\ell} - 1]$. This, again, leads to a notable gain in efficiency in the modular exponentiations. Under the common assumption [18,35] that discrete logarithm problem (DLP) in $\mathbb{Z}_n^\times$ is hard even when exponents are short, distributions of ephemeral keys with short and long exponents, respectively, are computationally indistinguishable from each other (see Lemma 3.6 in [35]). Hence, shortening ephemeral keys in the described way does not result in a considerably weaker security of the protocol.

Note that even though the protocol in Fig. 5 is displayed as four-message protocol, by concatenating messages $\mathcal{S}_{U'}$ and $\mathcal{S}'_{U'}$ into a single message, the protocol is trivially turned into a three-message protocol.

6.2 Performance analysis

The computational complexity of the Discover protocol is essentially related to the number of exponentiations, executed for each contact in lines 6 and 15. Any user U needs to compute $2|\mathsf{CL}_U|$ modular exponentiations, where $|\mathsf{CL}_U|$ denotes the number of contacts of U. If the polynomial-based IHME constructions from Sects. 3.2.1–3.2.3 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 $O(|\mathsf{CL}_U|^2)$ and $O(|\mathsf{CL}_U| \cdot |\mathsf{CL}_{U'}|)$,

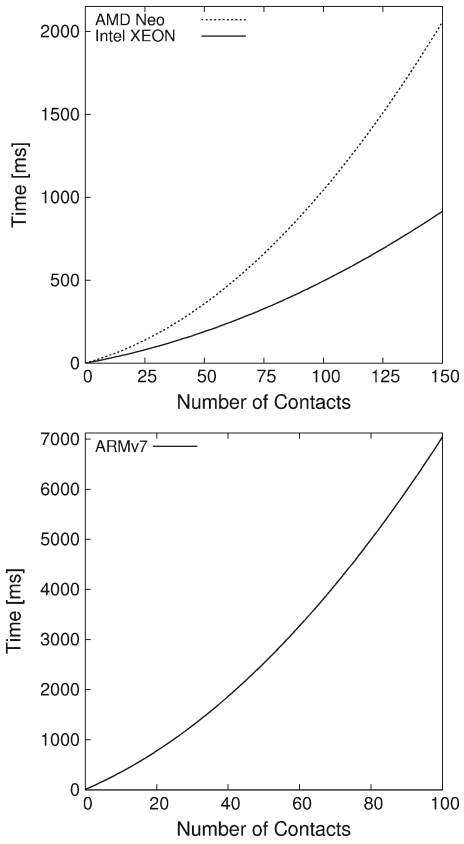

Fig. 6 Running times of our optimized Discover protocol on different CPUs, for an increasing number of contacts. All measurements are taken for 80 bit (symmetric) security and 1024 bit RSA moduli

respectively. The overall communication complexity of the Discover protocol (including the IHME-encoded transmission) is linear in the number of contacts.

We discuss performance results obtained from a concrete implementation of our optimized Discover protocol from Fig. 5. In particular, we present performance measurements and investigate the scalability of our Discover protocol for the security level $(\kappa, \ell) = (1024, 80)$, that is, 1,024 bit RSA combined with 80 bit 'symmetric security', and varying numbers of contacts $n = |\mathsf{CL}|$. These measurements are based on implementation where ν-interleaved IHME was used with $\nu = 14$.

Since privacy-preserving contact discovery is relevant particularly on mobile devices, our timing measurements are based CPUs with different computing powers. Figure 6 presents running times of our Discover protocol on a single core of an Intel XEON 2.6 GHz CPU, an AMD NEO 1.6 GHz processor (often found in Netbook computers), and an ARMv7 600 MHz CPU (installed on many of today's smartphones). All measurements were taken using the GMP library [14]; thus, execution on smartphones presumably can even be made more faster by choosing a different cryptographic library, optimized for mobile environments.

Observe that our protocol for private contact discovery scales fairly well in practice. For security level $(\kappa, \ell) = (1024, 80)$, on laptops and server machines, a full protocol execution requires less than a second, even for more than 100 contacts per user. On cores with smaller footprint, for example, on smartphones like Nokia's N900 (equipped with the ARMv7 600 MHz processor), protocol execution with 100 contacts requires about 7 s, which is still an acceptable overhead considered that the contact discovery protocol is executed only once, to establish social proximity. Also note that smartphones' CPU speeds are envisioned to increase rapidly in the near future (e.g., the iPhone 4G is equipped with a 1 GHz processor, the Samsung Galaxy Nexus with a dual-core 1.2 GHz CPU).

Finally, our measurements show that each user sends and receives around 300 bytes per user in his/her contact list (we assume $|\mathsf{CL}_U| = |\mathsf{CL}_{U'}|$ for simplicity). That is, in a protocol execution with 100 contacts per use, a total of 30 KB are transmitted.

7 Conclusion and future research

This paper provides a cryptographic treatment of *private contact discovery*. Our setting assumes that, in a social network environment, participants individually manage lists of their respective friends ('contacts'). If two users jointly execute a contact-discovering protocol, on input their contact lists, the protocol identifies the set of contacts they have in common. This matching is performed in a privacy-preserving way, that is, without disclosing non-matching contacts to the respective peer. After showing that available solutions to similar problems fail suffer from severe privacy shortcomings, we formally define a strong security model and construct a provably secure solution, based on the RSA assumption on safe moduli (in the random oracle model). The complexity of our protocols is linear in the number of contacts per user.

During protocol design, we overcome several challenges. In order to prevent adversaries from arbitrarily expanding their contact lists, we introduce the concept of *contact certification*. We also show, through an experimental evaluation, that our solutions are practical enough to be deployed in real-world applications, including those running on mobile devices.

Items for future research include the problem of privately identifying i-th degree contacts (for $i \geq 2$). Consider two users, Alice and David, willing to privately discover whether there exists a 'chain of contacts' (of length $\leq i$) between them. That is, the main challenge is to let them efficiently discover whether there exist other users, for example, Bob and Charlie, such that Alice is friend with Bob, Bob is friend with Charlie, and Charlie is friend with David. We leave it as an interesting open problem to instantiate an efficient pri-

vate discovery protocol for such i-th grade contacts, without relying on trusted third parties. Finally, an additional item for future work is hiding the size of contact lists. While this extra privacy feature has been recently realized in related problems (such as, Private Set Intersection [1]), it remains a challenging open problem for the CDS functionality.

Acknowledgments Mark Manulis was supported by the German Research Foundation (DFG) through grant MA 4096. He and Bertram Poettering also acknowledge support from the Center of Advanced Security Research Darmstadt (CASED) and the European Center for Security and Privacy by Design (EC SPRIDE). Work has been partially done while Emiliano De Cristofaro was at UC Irvine, and Mark Manulis and Bertram Poettering were at CASED & TU Darmstadt. This is an extended version of the paper with the same title that appeared in the proceedings of ACNS 2011 in LNCS 6715, Springer.

References

1. Ateniese, G., De Cristofaro, E., Tsudik, G.: (If) size matters: Size-hiding private set intersection. In: D. Catalano, N. Fazio, R. Gennaro, and A. Nicolosi (eds.) PKC 2011: 14th International Workshop on Theory and Practice in Public Key Cryptography, vol. 6571 of Lecture Notes in Computer Science, pp. 156–173. Taormina, Italy, March 6–9. Springer, Germany, Berlin (2011)
2. Boyd, C., Mathuria, A.: Protocols for Authentication and Key Establishment. Springer, Berlin (2003)
3. Chapman, P., Evans, D., Huang, Y., Koo, S.: Common Contacts–Privacy-preserving shared contact computation. http://www.mightbeevil.com/contacts/
4. Chiou, S.-Y., Chang, S.-Y., Sun, H.-M.: Common friends discovery with privacy and authenticity. In: IAS, pp. 337–340. IEEE Computer Society (2009)
5. Dachman-Soled, D., Malkin, T., Raykova, M., Yung, M.: Efficient robust private set intersection. In: Abdalla, M., Pointcheval, D., Fouque, P.-A., Vergnaud, D. (eds.) ACNS 09: 7th International Conference on Applied Cryptography and Network Security, vol. 5536 of Lecture Notes in Computer Science, pp. 125–142. Paris-Rocquencourt, France, June 2–5. Springer, Germany, Berlin (2009)
6. Hankerson, D., Menezes, A., Vanstone, S.: Guide to Elliptic Curve Cryptography. Springer, Berlin (2004)
7. De Cristofaro, E., Kim, J., Tsudik, G.: Linear-complexity private set intersection protocols secure in malicious model. In: Abe, M. (ed.) Advances in Cryptology–ASIACRYPT, vol. 6477 of Lecture Notes in Computer Science, pp. 213–231. Singapore, December 5–9. Springer, Germany, Berlin (2010)
8. De Cristofaro, E., Manulis, M., Poettering, B.: Private discovery of common social contacts. In: Lopez, J., Tsudik, G. (eds.) ACNS 11: 9th International Conference on Applied Cryptography and Network Security, vol. 6715 of Lecture Notes in Computer Science, pp. 147–165, Nerja, Spain, June 7–10. Springer, Germany, Berlin (2011)
9. De Cristofaro, E., Tsudik, G.: Practical private set intersection protocols with linear complexity. In: Sion, R. (ed.), FC 2010: 14th International Conference on Financial Cryptography and Data Security, vol. 6052 of Lecture Notes in Computer Science, pp. 143–159. Tenerife, Canary Islands, Spain, January 25–28. Springer, Germany, Berlin (2010)
10. De Cristofaro, E., Jarecki, S., Kim, J., Tsudik, G.: Privacy-preserving policy-based information transfer. In: Goldberg, I., Atallah, M.J. (eds.) Privacy Enhancing Technologies, vol. 5672 of Lecture Notes in Computer Science, pp. 164–184. Springer, Berlin (2009)
11. Diehl, C.P., Namata, G., Getoor, L.: Relationship identification for social network discovery. In: AAAI, pp. 546–552. AAAI Press (2007)
12. Okamoto, E., Tanaka, K.: Key distribution system based on identification information. IEEE J. Sel. Areas Commun. **7**(4), 481–485 (1989)
13. Emerson, R.: Huffingtonpost: Facebook Users Expected To Pass 1 Billion In August 2012. http://www.huffingtonpost.com/2012/01/13/facebook-users-1-billion-icrossing_n_1204948.html, July 2012
14. Free Software Foundation. The GNU MP Bignum Library. http://gmplib.org/
15. Freedman, M.J., Nicolosi, A.: Efficient private techniques for verifying social proximity. IPTPS, In (2007)
16. Freedman, M.J., Nissim, K., Pinkas, B.: Efficient private matching and set intersection. In: Cachin, C., Camenisch, J. (eds.) Advances in Cryptology–EUROCRYPT 2004, vol. 3027 of Lecture Notes in Computer Science, pp. 1–19. Interlaken, Switzerland, May 2–6. Springer, Germany, Berlin (2004)
17. Gennaro, R., Krawczyk, H., Rabin, T.: Okamoto-Tanaka revisited: Fully authenticated Diffie-Hellman with minimal overhead. Cryptology ePrint Archive, Report 2010/068, 2010. http://eprint.iacr.org/2010/068.pdf
18. Gennaro, R., Krawczyk, H., Rabin, T.: Okamoto-Tanaka revisited: Fully authenticated Diffie-Hellman with minimal overhead. In: Zhou, J., Yung, M. (eds.) ACNS 10: 8th International Conference on Applied Cryptography and Network Security, vol. 6123 of Lecture Notes in Computer Science, pp. 309–328, Beijing, China, June 22–25. Springer, Germany, Berlin (2010)
19. Google Inc. Google+. http://plus.google.com
20. Hazay, C., Lindell, Y.: Efficient protocols for set intersection and pattern matching with security against malicious and covert adversaries. In: Canetti, R. (ed.) TCC 2008: 5th Theory of Cryptography Conference, vol. 4948 of Lecture Notes in Computer Science, pp. 155–175. San Francisco, CA, USA, March 19–21. Springer, Germany, Berlin (2008)
21. Hazay, C., Nissim, K.: Efficient set operations in the presence of malicious adversaries. In: Nguyen, P.Q., Pointcheval, D. (eds.) PKC 2010: 13th International Conference on Theory and Practice of Public Key Cryptography, vol. 6056 of Lecture Notes in Computer Science, pp. 312–331. Paris, France, May 26–28. Springer, Germany, Berlin (2010)
22. Huang, Y., Chapman, P., Evans, D.: Privacy-preserving applications on smartphones. In: 6th USENIX Workshop on Hot Topics in, Security (2011)
23. Jarecki, S., Kim, J., Tsudik, G.: Beyond secret handshakes: Affiliation-hiding authenticated key exchange. In: Tal M. (ed.), Topics in Cryptology–CT-RSA 2008, vol. 4964 of Lecture Notes in Computer Science, pp. 352–369. San Francisco, CA, USA, April 7–11. Springer, Germany, Berlin (2008)
24. Jarecki, S., Liu, X.: Efficient oblivious pseudorandom function with applications to adaptive OT and secure computation of set intersection. In: Reingold, O. (ed.) TCC 2009: 6th Theory of Cryptography Conference, vol. 5444 of Lecture Notes in Computer Science, pp. 577–594. Springer, Berlin, Germany, March 15–17, 2009
25. Jarecki, S., Liu, X.: Fast secure computation of set intersection. In: Garay, J.A., De Prisco, R. (eds.) SCN 10: 7th International Conference on Security in Communication Networks, vol. 6280 of Lecture Notes in Computer Science, pp. 418–435. Amalfi, Italy, September 13–15. Springer, Germany, Berlin (2010)
26. Kissner, L., Song, D.: Privacy-preserving set operations. In: Shoup, V. (ed.) Advances in Cryptology–CRYPTO 2005, vol. 3621 of Lecture Notes in Computer Science, pp. 241–257, Santa Barbara, CA, USA, August 14–18. Springer, Germany, Berlin (2005)

27. Korolova, A., Motwani, R., Nabar, S.U., Xu, Y.: Link privacy in social networks. In: ICDE, pp. 1355–1357. IEEE (2008)
28. Korolova, A., Motwani, R., Nabar, S.U., Xu, Y.: Link privacy in social networks. In: Shanahan, J.G., Amer-Yahia, S., Manolescu, I., Zhang, Y., Evans, D.A., Kolcz, A., Choi, K.-S., Chowdhury, A. (eds.) CIKM, pp. 289–298. ACM (2008)
29. Krawczyk, H.: SIGMA: The "SIGn-and-MAc" approach to authenticated Diffie-Hellman and its use in the IKE protocols. In: Boneh, D. (ed.), Advances in Cryptology—CRYPTO 2003, vol. 2729 of Lecture Notes in Computer Science, pp. 400–425, Santa Barbara, CA, USA, August 17–21. Springer. Germany, Berlin (2003)
30. LinkedIn. Press center - about us. http://press.linkedin.com/about, July 2012
31. Manulis, M., Pinkas, B., Poettering, B.: Privacy-preserving group discovery with linear complexity. In: Zhou, J., Yung, M. (eds.) ACNS 10: 8th International Conference on Applied Cryptography and Network Security, vol. 6123 of Lecture Notes in Computer Science, pp. 420–437, Beijing, China, June 22–25. Springer, Germany, Berlin (2010)
32. Manulis, M., Poettering, B., Tsudik, G.: Taming big brother ambitions: More privacy for secret handshakes. In Atallah, M.J., Hopper, N.J. (eds) Privacy Enhancing Technologies, vol. 6205 of Lecture Notes in Computer Science, pp. 149–165. Springer (2010)
33. Manulis, M., Poettering, B.: Practical affiliation-hiding authentication from improved polynomial interpolation. In: ASIACCS, pp. 286–295 (2011)
34. Schatzman, M.: Numerical Analysis: A Mathematical Introduction. Clarendon Press, Oxford (2002)
35. Goldreich, O., Rosen, V.: On the security of modular exponentiation with application to the construction of pseudorandom generators. J. Cryptol. **16**(2), 71–93 (2003)
36. Okamoto, E.: Key distribution systems based on identification information. In: Pomerance, C. (ed.) Advances in Cryptology—CRYPTO '87, vol. 293 of Lecture Notes in Computer Science, pp. 194–202, Santa Barbara, CA, USA, August 16–20. Springer, Germany, Berlin (1988)
37. Pons, P., Latapy, M.: Computing communities in large networks using random walks. J. Graph Algorithms Appl. **10**(2), 191–218 (2006)
38. Poettering, B.: Privacy protection for authentication protocols. PhD thesis 2012. http://tuprints.ulb.tu-darmstadt.de/2867
39. von Arb, M., Bader, M., Kuhn, M., Wattenhofer, R.: Veneta: Serverless friend-of-friend detection in mobile social networking. In: WiMob, pp. 184–189. IEEE (2008)
40. Yao, A.C.-C.: How to generate and exchange secrets (extended abstract). In: FOCS, pp. 162–167. IEEE Computer Society (1986)
41. Yu, P.S., Han, J., Faloutsos, C.: Link Mining: Models, Algorithms, and Applications. Springer, Berlin (2010)
42. Zhelevam, E., Getoor, L., Golbeck, J., Kuter, U.: Using friendship ties and family circles for link prediction. In: Giles, C.L., Smith, M., Yen, J., Zhang, H. (eds.) SNAKDD, vol. 5498 of Lecture Notes in Computer Science, pp. 97–113. Springer (2008)