Vis-à-vis Cryptography: Private and Trustworthy In-Person Certifications

Ian Miers* Matthew Green* Christoph U. Lehmann, MD† Aviel D. Rubin*

Abstract

The growing role of mobile devices in previously face to face interactions presents new domains for cryptographic applications. At the same, time the increased role of digital systems raises new security and privacy issues. With some thirty thousand notifications sent, inSPOT.org's electronic notification of exposure to sexually transmitted infections is one such concerning development. This paper explores those concerns, the features of an ideal service for both notification and certification, and outlines protocols for cryptographic solutions.

Keywords: medical records, applied cryptography, protocols, privacy

1 Introduction

Whether it is using LinkedIn to find jobs, WebMD to obtain medical advice, or Foursquare for social interaction, digital systems now play an important role in human interaction. At the same time, these technologies have introduced new security and privacy risks. These risks have accelerated with the increased popularity of mobile applications, which move these interactions out from behind monitors and into the public realm.

One interesting consequence of these trends is the emergence of online systems for notifying sexual partners of exposure to sexually transmitted infections (STIs). In 2004, the digital advocacy group ISIS introduced inSPOT, an Internet-based system for notifying partners via e-cards. inSPOT has received significant attention from medical researchers [19, 16]. Hogben *et al.* describe the medical importance of notification systems such as inSPOT, noting that "The high number of cases of gonorrhea and chlamydia... makes [paper-based] partner notification for all named partners impractical in many jurisdictions." [16]

inSPOT's very existence is stems from another change in human interactions: the prevalence of sites/apps for online dating and casual sex. Without these apps, inSPOT's target audience would be unlikely to have email addresses for their partners, or any contact information at all for the case of pseudonymous partners. Users of such networks already share information on disease status [4]. In the case of one site, MANHUNT, HIV status is actually part of every user's profile. Similarly, at least one business sells digital "STD free" certificates that can be included in online dating site profiles.

Although there is a fair amount of coverage in the popular press of the role of online dating in STI transmission, there have been relatively few scientific evaluations [4]; hence it is difficult to determine what role STI status plays. What we can say with confidence is that individuals are cognizant of the issue when finding partners online, that currently the filtering process relies on trust, and that it would be desirable to develop more reliable methods for screening partners.

This paper examines the security and privacy implications of STI notification and certification. We emphasize that the former is not a theoretical problem in computer security, nor is it limited to small scale public health experiments: these problems are present in real systems with large numbers of users. Moreover, given the extremely sensitive nature of the data involved – STI status, sexual partner history, and sexual orientation – and a potential motivation to cheat or game the system (*e.g.,* to appear healthy when not) – there is a need for serious anonymity and security guarantees.

Our Contribution. To address the concerns above, we propose TruSTI, a mobile application that allows partners to *certify* their STI status in a manner that preserves the privacy of all participants. The core of TruSTI is a novel cryptographic protocol by which a user Alice can prove her status to another partner, without revealing information that could later be used against her. The tech-

*Johns Hopkins University Department of Computer Science [imiers,mgreen,rubin]@cs.jhu.edu

†Johns Hopkins University School of Medicine clehmann@jhmi.edu

niques used in constructing this protocol borrow from the area of *anonymous credentials* [6] and *deniable signatures* [22, 12]. We also propose an extended protocol that allows users to securely notify their partners of a change in STI status. All of our protocols allow partners to interact via an untrusted connection (*e.g.,* the Internet).

Specifically, our contributions are:

1. A new *deniable* certification protocol that uses signatures with efficient protocols and non-interactive zero-knowledge proofs. Using this protocol, a user A may *certify* her status to a user *B* without revealing any information that could later be used as evidence that a transaction even took place.

2. An anonymous credential system that uses human-verifiable identifiers: specifically, user photographs. Many existing credential systems assume the existence of a PKI, or some arbitrary 'pseudonym'. By tying photographs to the credential, we address many of the usability challenges involved in deploying a "face-to-face" credential system.

3. A deniable *notification* protocol that a user can employ to securely notify partners of a change in STI status.

4. Finally, we consider the implementation and deployment challenges of our system, and propose a design specification for a TruSTI deployment on smartphones.

We stress that TruSTI is not a theoretical exercise. We have designed the system to work with real mobile hardware operated by users with no technical training. Although we employ advanced cryptographic techniques in our certification and notification protocols, the cryptography is kept "under the hood". Most importantly, TruSTI does not require users to exchange or manage cryptographic keys, something that has proven to be challenging for non-technical users (see *e.g.,* [25]). We assume only that the user is familiar with smartphone social applications such as Facebook and that she is capable of linking to a social network profile with a recognizable photograph. See Figures 1 and 3 for an example of what TruSTI looks like from the user's perspective.

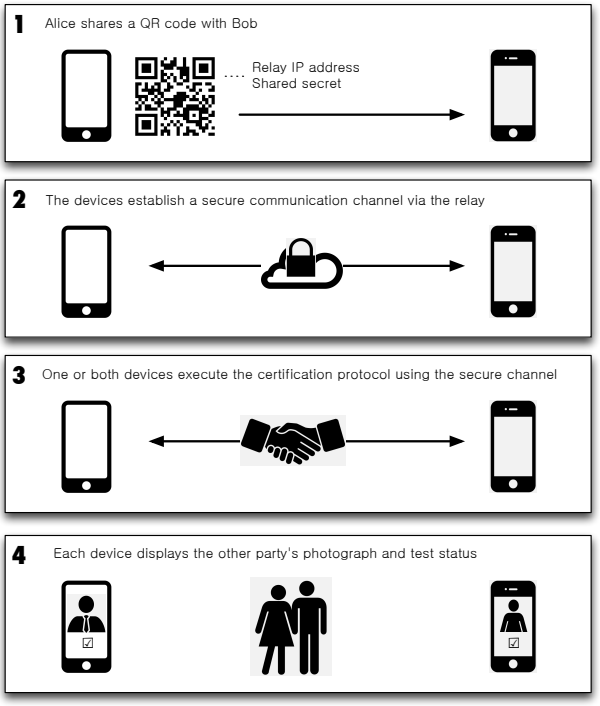

Figure 1: How TruSTI certification works in practice. Note that the devices exchange messages using a secure communications protocol, with the assistance of an untrusted Internet relay service. Only steps (1) and (4) are visible to the user.

2 Medical Background

This section gives a basic background on Sexually Transmitted Infections (STIs), human behavior, and the role of STIs in partner selection.

Sexually Transmitted Infections. Considering the prevalence of STIs and the fact that a number of these infections such as chlamydia or HIV may be spread when the affected patient is asymptomatic, detection and treatment are critical. For example, it is estimated that at least 1 in 6 Americans have herpes and that many are unaware of their infection [26]. This underscores both the importance, well understood in the public health community, of testing and notification of partners when they have been exposed.

STI testing, however, is not instantaneous. STIs have incubation periods ranging from days (Herpes Simplex) to months (Human Papillomavirus) [26]. Hence a negative STI test only reflects that the user had no STIs roughly six months prior. Even in high risk populations, tests are recommended only annually [26].

Testing involves a patient, a physician's office or clinic where the samples are taken, and a lab where the test is conducted. In the US, there are two major commercial medical and diagnostic laboratories: Quest Diagnostics and Laboratory Corporation of America (LabCorp) [17].

Notification. An important concept of STI prevention is partner notification. Notifying past partners of a new STI provides the opportunity to initiate early testing and may prevent further spread of the infection.

Traditional STI notification occurred by mail, by phone, or in person. Notification was either the patient's

responsibility, facilitated by a public health investigator, or done by the public health investigator at the patient's request [16]. These approaches have two shortcomings. First, as noted previously, the volume of cases makes this impractical. Second, the personal interactions involved may cause under reporting [14].

As far as we know, the first major system for online partner notification is inSPOT. inSPOT was designed to provide a "Web site that uses electronic postcards (e-cards) to assist people in disclosing an STI diagnosis to their sex partner(s)" [19]. Similar to an e-vite or an email, inSPOT users can create an e-card notifying their partner to get tested. They may include a personal message. The card is sent to a user provided email address either anonymously or from a named recipient.

The assumption is that users will have met their partner via an online personal ad or dating site and thus learned their partner's email address. This assumption does not hold, nor does inSPOT work, in the case of a meeting in a bar or via a geo-social networking app (see next section). Such limitations obviously reduce the effectiveness of the system.

inSPOT has been at least moderately successful in providing notifications: 9,916 e-cards were sent in the L.A area alone in 2006 [19]. Between the site's creation in 2004 and 2008, some 30,000 notifications were sent [19]. Considering the relatively small high risk communities that this program targeted and the small number of cities involved, these numbers are encouraging.

STIs and social networks. A new set of mobile applications have emerged for "geo-social networking." Using these apps to find sex partners in non-online situations is an increasingly prevalent practice in at least some target communities. One prominent app, Grindr boasts over 1 million gay and bisexual users worldwide. Rather than being used behind monitors at home, Grindr is used in bars and other locations in lieu of a physical approach [20].

Such users, accustomed to digital intermediaries for partner selection are the most likely initial users of TruSTI. At the same time, since ease of finding sexual partners increases the risk of acquiring STIs and since youth with partners found both online and offline are more likely to report STIs [3, 21], TruSTI could be of substantial use to this demographic.

Partner filtering by STI status in social networks. Even though there is evidence that sensation seeking and impulsive decision-making are associated with sexual risk behaviors, the impact of knowledge about a potential partner's STI status in the selection process has not been well described [10]. Indirect evidence points to the STI status as an important selection criterion: users in chat rooms and on dating sites do advertise that they are "disease and drug free" [4]. The popular gay dating site MANHUNT.net has a field for HIV status on all of its profiles.

The online service qpid.me allows users to reveal STI test results to a potential partners either via text message or on an online dating site. Unlike statements in chat rooms and on MANHUNT.net, which rely on trust, qpid.me verifies results from a testing center/lab. The service is relatively new, founded in 2010 and usage numbers do not appear to be publicly available [1].

3 Existing Approaches and Issues

We now briefly describe some existing technical solutions for partner notification and certification.

inSPOT.org inSPOT is used by a large number of users and offers a convenient way of notifying past sexual partners to get tested for an STI. Unfortunately, inSPOT addresses only the problem of post-encounter STI notification: it does not prevent encounters with STI infected partners. Moreover, inSPOT suffers from a number of security and privacy issues.

Although the inSPOT operators seem trustworthy, users' private information might be obtained by way of a server breach (something that has become increasingly common in recent years). Moreover, inSPOT uses standard non-encrypted electronic mail to notify partners, which further opens the user to an account breach on these services. The loss of this information is of particular concern for individuals who do not wish to publicly disclose their sexual preferences due to fear of discrimination [18]. Finally, a user's sexual preferences or infection status might "leak out" inadvertently due to machine-targeted advertising by web-based email providers.

Privacy issues associated with systems like inSPOT can lead to only partial partner notification. Individuals may under-report partners to avoid looking overly promiscuous, or they may leave off same-sex partners. Even when reporting is honest, some partners may not share the necessary email contact information for privacy reasons.

Finally, systems like inSPOT are always vulnerable to the problem of *false* notifications. Because these e-cards appear to come from a reputable site, they can cause emotional and financial harm (as, for example, one individual discovered [15]). inSPOT has no solution to this problem.

qpid.me qpid.me [1] provides certification of STI status via text message, a website, and badges on dating sites. Because certification involves both parties, over time qpid.me learns the sexual history of a user. Moreover, qpid.me has no strong binding between its user ids

Figure 2: Some example e-cards used by the inSPOT partner notification system [16].

and the user's medical records. Worse, since it is authenticated with a simple PIN, there is no binding whatsoever between qpid.me's user ids and a user's real world identity or identity on a dating site. Thus it is both reasonably easy for a malicious user to provide negative STI tests that are not hers and trivially easy to borrow a friend's clean qpid.me profile. In short, it has almost all of the privacy risk of inSPOT coupled with a far higher consequence for forgery.

4 TruSTI: Technical Approach

In this section we lay out some of the technical properties used in TruSTI.

4.1 Security Properties

TruSTI addresses two basic problems:1) How can a Prover demonstrate in-person to a Verifier that a Test Center has certified a set of results; and 2)How can the same Prover subsequently notify the Verifier if the results change. We now describe the privacy and security properties required for this transaction.

Unforgeability. Informally, the Prover should not be able to convince a Verifier to accept a set of results that were not certified by the Test Center.

Deniability The Verifier should not be able to convince a third party of the results, or even that an interaction took place. In practice, we require that any transcript of the cryptographic protocols should be completely forgeable by either party.

Authenticated notification. When notified of a later change in the Prover's test status, the Verifier should be assured that the notification came from the Prover.

Reciprocal anonymity. Preventing fraudulent notifications requires the Verifier to issue some credential that the Prover can use to prove interaction at some later point. However, neither party should be able to use this credential to prove a relationship.

4.2 Technical obstacles and solutions

The properties above closely map to the goals of an anonymous credential system. However, anonymous credentials are in and of themselves insufficient. We require additional properties:

Effective non-transferability. Digital objects are easily shared. This cannot be prevented or even detected in a truly anonymous credential system: by dint of their strong anonymity requirements, credentials cannot be tied to a physical owner. In practice, two approaches have been used to prevent sharing of credentials: one is to limit the usefulness of a credential (*e.g.,* by preventing double-spending); the other is to tie the credential to the user's public key.

Because both parties meet physically, we are not bound by the strong anonymity requirement and its prohibition against physically linkable credentials. We are free to use any techniques we want, including biometrics, to prove that Alice has a credential that was issued to her provided we don't further erode Alice's anonymity. We choose to include a photo in the credentials, effectively constructing a pseudonymous digital photo ID. Although the credential protocol may take place online, it is not fully verified until physical credential linkage is established by a face to face meet.

5 Cryptographic Preliminaries

The protocols in this work are based on several cryptographic building blocks which we describe in this section. The reader should refer to the cited references for details and security definitions. Let k be an adjustable security parameter for our protocols.

Zero knowledge proofs of knowledge. Zero knowledge proofs of knowledge allow a Prover to prove to a Verifier that she knows a witness to a statement, without revealing any information about the witness itself. There are a variety of techniques for proving common algebraic statements which have been used in the literature to construct statements over committed values [24, 7, 11, 9, 2]. The majority of these protocols can be made *non-interactive* using the Fiat-Shamir heuristic [13]. We denote such a proof as a $\mathsf{NIZKPoK}$.

To describe these proofs, we will use the notation of Camenisch and Stadler [8]. Consider for example: $\mathsf{NIZKPoK}\{(x) : h = g^x\}$. The variables inside the parenthesis indicate the secret values that are not revealed in the proof and the expression after the colon is a statement about those values that is proven. These proofs can also be used to prove *compound* statements. For example, to assert knowledge of *one* of two statements, we will use the expression $(\mathsf{Statement\ 1} \vee \mathsf{Statement\ 2})$. These proofs work over a wide variety of statements and conjunctions.

Signatures with efficient protocols. Our techniques require a signature scheme with the following special properties. First we require a scheme that can sign a vector of one or more messages. Second, we will need to conduct non-interactive proofs over statements related to these signatures. There are several signature schemes that meet these requirements, the most notable being the CL-signature of Camenisch and Lysyanskaya [6], which is secure under the Strong RSA assumption.[1] Notationally, we will describe the signature scheme as a tuple of algorithms $(\mathsf{KG}, \mathsf{Sign}, \mathsf{Verify})$. To describe a proof of knowledge of a signature σ on public message vector $\vec{m}$ and public key pk, we will use following notation: $\mathsf{NIZKPoK}\{(\sigma) : \mathsf{Verify}(pk, \sigma, \vec{m}) = 1\}$.

Commitments. Let $(\mathsf{CSetup}, \mathsf{Commit}, \mathsf{Decommit})$ be a secure commitment scheme where CSetup generates public parameters *params*; on input a message m and random "decommitment" r, $\mathsf{Commit}(params, m, r)$ outputs a commitment $\mathbf{C}$; and on input $\mathbf{C}, m, r$, $\mathsf{Decommit}$ outputs $\{0,1\}$.[2] Our subsequent constructions require an efficient protocol for proving knowledge of (m,r) with respect to a commitment, which we denote by $\mathsf{NIZKPoK}\{(m,r) : \mathbf{C} \in \mathsf{Commit}(params, m, r)\}$. We recommend using the Pedersen commitment scheme [23] based on the discrete logarithm assumption. Schnorr's technique [24] may be used to efficiently conduct the proof.

[1] See [6] for details of the proof protocols. We briefly note that these protocols can be made non-interactive by using the Fiat-Shamir heuristic; see *e.g.,* the protocols of [5] for an example.

[2] We will not be specific about the message space of the commitment scheme, and assume that it can handle arbitrary messages (possibly with the assistance of a collision-resistant hash function).

6 Certification

We now describe a cryptographic protocol by which one party may *certify* their status to another individual. Our protocol consists of three phases: *global setup*, conducted by the trusted Test Laboratory; *enrollment*, in which a user obtains a digital certification of their STI status and photo from the Lab; and *certification* in which a user proves this status to another individual. In each interaction, we assume a secure channel between the individuals communicating.

Setup. Prior to enrolling any patients, the test center generates a signature key pair $(pk_T, sk_T) \leftarrow \mathsf{KG}(1^k)$ as well as the global parameters for a commitment scheme. The test center publishes pk_T to all parties in the system.

Enrollment. When a patient obtains the results of an STI test, she enrolls in the system using the following steps. First, the user verifies her identity by producing a valid photograph, which the test center administrator should verify for accuracy.[3] As input to this process, the test center provides the user's STI status (which may consist of a vector of test results), the test date, and the test center's signing key sk_T.

1. **Credential generation.** The test center constructs a 'credential' $\sigma = \mathsf{Sign}(sk_T, \langle H, m_1, \ldots, m_t, T\rangle)$, where:
   (a) H is a collision-resistant hash of the patient's photograph.
   (b) $(m_1, \ldots, m_t)$ is a vector containing the patient's STI status (t is the number of distinct-conditions tested).[4]
   (c) T is a timestamp.

The test center provides the patient with the credential σ as well as the message elements above. The patient records these values securely on a device such as a smartphone.

Proving STI status. When an individual wishes to prove her status to another party, the two parties engage in the following protocol. We denote the first party as the "Prover", and her input is: her photograph, the STI status results $(m_1, \ldots, m_t)$, the timestamp T, and the credential σ generated during enrollment. The second party is the "Verifier", and provides no input.

[3] In practice, this photograph can be a public photograph: *e.g.,* a profile picture for a social network. It is preferable that the photo be publicly-available, to prevent a potential partner from using it as evidence that an interaction occurred.

[4] For simplicity, we will assume that the order and formatting of these results is fixed. In practice, these messages may be encoded as key-value pairs.

1. The Verifier first generates a random nonce $n \in \{0,1\}^k$ and a random decommitment value r, then generates a commitment $\mathbf{C} = \mathsf{Commit}(params, n, r)$, and transmits $\mathbf{C}$ to the prover. He stores (n, r) for later use.

2. The Prover now constructs the following non-interactive proof:

$$\Pi = \mathsf{NIZKPoK}\{(\sigma) : \mathsf{Verify}(pk_T, \sigma, \langle H, m_1, \ldots, m_t, T\rangle) = 1 \vee \mathbf{C} \in \mathsf{Commit}(params, n, r)\} \quad (1)$$

   Note that the Prover does not know the values (n, r) in the second equation. This is acceptable, since it is sufficient for the Prover to satisfy the first equation in the OR proof.

   Next, the Prover selects a random decommitment r_Π and computes a commitment on Π as:

$$\mathbf{C}_\Pi = \mathsf{Commit}(params, \Pi, r_\Pi)$$

   The Prover transmits the commitment $\mathbf{C}_\Pi$, her photograph, and the values $(m_1, \ldots, m_t, T)$ to the Verifier.

3. The Verifier reveals the nonce n and the decommitment value r (selected in step 1) to the Prover.

4. The Prover verifies that $\mathsf{Decommit}(params, \mathbf{C}, n, r) = 1$. If so, she transmits the proof Π and the decommitment r_Π to the Verifier.[5]

5. The Verifier checks that $\mathsf{Decommit}(params, \mathbf{C}_\Pi, \Pi, r_\Pi) = 1$, computes H by hashing the photograph, and verifies that Π is a valid proof of knowledge on all known messages. If the proof verifies, the Verifier outputs the Prover's photograph, timestamp T, and STI status vector $(m_1, \ldots, m_t)$. If not, it outputs Reject.

Commentary. The certification protocol will convince a Verifier that the Prover possesses a valid credential σ on her test results, *and* that the results are bound to the Prover's photograph. The main ingredient is the NIZK proof Π that the Prover constructs in step 2. Intuitively, this proof shows that *either* the Prover has a valid credential tied to the photograph, *or* that she knows the nonce n chosen by the Verifier in step 1. This is convincing to the Verifier, since at this point in the protocol the Verifier has not yet revealed the nonce n.

The rest of the protocol exists to ensure that the Verifier cannot convince some third party that the transaction has taken place. Having committed to a proof Π in step 2, the Prover now forces the Verifier to make (n, r) public. Only after receiving these values does the Prover reveal the actual proof Π. The nature of Π is such that *any party* with knowledge of (n, r) can forge a valid-looking proof (for any user at all, given only a photograph); hence, once these values are public, the proof will have limited ability to convince a third party.

[5]The Prover may optionally publish $(\mathbf{C}, n, r)$ to a public bulletin board via an anonymous communication network.

6.1 Detailed Security Analysis

A full security proof is beyond the scope of this paper, but in this section we briefly sketch the outline that such a proof would take. We begin with the following assumptions: (a) the signature scheme is existentially unforgeable, (b) the NIZK proof system is both sound and zero-knowledge, and (c) the commitment scheme is binding and hiding.

Forgery. We first consider the property of *credential unforgeability*: that an adversarial Prover cannot convince an honest Verifier to accept false data. For the purposes of our analysis, we assume that the Test Center is trusted. Therefore, if an adversarial Prover convinces the Verifier to accept invalid data, one of three events *must* have occurred: (1) the Prover has forged the signature σ;(2) the Prover has formulated an invalid NIZK for the credential; (3) the Prover has correctly formulated the NIZK, but has based the proof on knowledge of (n, r) rather than the credential σ.

Clearly the first two events directly contradict assumptions (a) and (b), and would allow us to create an adversary against those schemes. We therefore focus on the third event. In this case, there are two possibilities: that the Prover has *guessed* (n, r) at step 2, *before* the Verifier reveals these values; or that the Prover formulated the proof Π *subsequent* to step 2, and yet still produced a valid decommitment to C_Π. The former event should occur with only negligible probability, while the latter would indicate an attack on the binding property of the commitment scheme. Under our assumptions, the probability of such a credential forgery must be negligible in the scheme's security parameter k.

Deniability. We now consider the possibility that an adversarial Verifier will attempt to convince a third party that an interaction has taken place. Without loss of generality, we assume that the transaction actually *did* take place with the Prover, though – as we will show – this is not necessary. We also assume that the third party is able to communicate with the Verifier prior to and subsequent to the execution protocol, but does not communicate in real time (*i.e.,* it cannot use the Verifier as a real time proxy) during the actual protocol transaction (we will discuss this possibility in Section 8). Let $\mathscr{T}$ be a

transcript of the interaction between the two parties. We will proceed by showing that any legitimate transcript $\mathscr{T}$ is indistinguishable from a *simulated* transcript $\mathscr{T}'$ that could have been produced *without* the participation of the Prover.

Any party can simulate a $\mathscr{T}'$ using only "forged" or public information: the alleged Prover's (publicly available) photograph, and an arbitrary set of values $(m_1,\ldots,m_t,T)$. The simulator selects (n,r) as in step 1, and constructs the NIZK using (n,r) as the witness. She then produces valid commitments $\mathbf{C}$ and $\mathbf{C}_\Pi$ to complete the protocol transcript. Clearly all aspects of the transcript $\mathscr{T}'$ *except* for the NIZK are distributed as in the real transcript $\mathscr{T}$. If a distinguisher succeeds in distinguishing $\mathscr{T}'$ from $\mathscr{T}$ with non-negligible advantage, we can use this distinguisher to violate the NIZK's witness-indistinguishability property, which in turn provides a distinguisher against the zero-knowledge property of the NIZK.

7 Notification

We next describe modifications to the *certify* protocol to enable the Prover to *notify* the Verifier of subsequent changes in her status. The new steps introduced in this section are: *notification grant* in which the Prover grants the Verifier tokens to later send a notification; *certify change* in which the Lab certifies that the Prover tested positive; and *notification send* in which the Prover proves to the Verifier that she tested positive.

To support notification, we make the following changes to the protocols described in §6:

Modified Enrollment protocol. During the Enrollment process, the Prover generates a keypair $(pk_P, sk_P) \leftarrow \mathsf{KG}(1^k)$. The Lab embeds pk_P into the credential σ. This public key will never be revealed, but instead used on in a NIZKPoKwhich proves knowledge of the corresponding secret key sk_P.

Modified Certification protocol.

1. In step 2 of the **Proving STI Status** protocol, the Prover generates a commitment $\mathbf{C}_P$ to her public key pk_P. She sends $\mathbf{C}_P$ to the Verifier, and modifies the NIZK to also prove (1) that she knows sk_P with respect to the committed value $\mathbf{C}_P$, and (2) that she knows the same sk_P with respect to $\mathbf{C}_{PL}$ in σ.

2. In step 3, the Verifier generates a random key pair $(pk_{rand}, sk_{rand}) \leftarrow \mathsf{KG}(1^k)$ and sends pk_{rand} to the Prover.

3. At the conclusion of the **Certification** protocol, the Prover interacts with the Verifier to extract a signature σ_{notify} under sk_{rand} on the Prover's public key pk_P which is embedded in $\mathbf{C}_P$.[6]

Notification send. When the Prover obtains a new test credential σ' with updated test status results $(m'_1,\ldots,m'_t,t')$, he can notify the Verifier by sending her $\langle pk_{rand}, (m'_1,\ldots,m'_t), t'\rangle$ and the following NIZK. Importantly he does not send his photograph,its hash H, or his public key. hence this prove contains no identifying information.

$$\Pi = \mathsf{NIZKPoK}\{\left(sk_P, pk_P, H, \sigma, \sigma_{notify}\right) : \mathsf{Verify}\left(pk_T, \sigma, \langle H, pk_P, m'_1,\ldots,m'_t,t',T\rangle\right) = 1 \wedge \mathsf{Verify}\left(pk_{rand}, \sigma_{notify}, \langle pk_U\rangle\right) = 1 \wedge (pk_P, sk_P) \in \mathsf{KG}\left(1^k\right)\ \} \quad (2)$$

The verifier checks this proof for correctness.

Discussion. The protocol above extend the basic Certification protocol to add the concept of *user identity* for the Prover. This is accomplished by adding a public key pk_U to the message signed by the Labl. In practice, the Prover never reveals his public key to the Verifier.[7] He does, however, use the extended Certification protocol to obtain a credential from the Verifier signing that public key. This credential allows him to later prove that an encounter took place and is tied to the Prover's public key. This does *not* violate the Verifier's privacy, since she generates this credential using a single-use random key pair.

In the event that the Prover later tests positive, he can prove that his new test credential σ' is bound to the same identity as his previous credential. Moreover, he can prove (to the Verifier) that he is the same individual with whom she interacted previously. Revealing this information does not compromise the Prover's anonymity, since the "notification" proof does not embed any information about the Prover's identity at all.

8 Implementation Sketch

In this section we describe an approach to implementing TruSTI using a mobile phone app platform, and discuss the various challenges that will be involved.

Application Overview. The end-user portion of the TruSTI is a mobile application. Figure 3 shows a rough mockup of the application UI after a user has enrolled

[6] This requires a protocol for extracting a signature on a committed value. Such protocols exist for the signature schemes suggested in §5.

[7] In principle, the Prover need not reveal pk_U to the Test Center either, although we do not explore the necessary extensions in this paper. The sole purpose of this key is to allow a Prover to *voluntarily* bind together test results taken at different times. It need never leave the application.

Figure 3: A UI mockup for TruSTI. From the left: a Prover selects "Share" on the home screen; the parties bump their phones or scan a QR code to handshake; the Prover's STI results are displayed on the Verifying user's smartphone. The Verifier must inspect the prover's photograph manually. This is executed symmetrically to exchange STI results.

with a test lab. Upon opening the app and selecting the share feature, users will be prompted to either (a) bump their phones together, or (b) scan a displayed QR code. In the former case, the phones will exchange information via near field communication (NFC) or Bump – a cross platform API for data exchange triggered by the accelerometer event of bumping phones. In the later case, this exchange is accomplished by embedding the information in the QR code.

In either case, the data exchanged will be used to establish a secure channel via an Internet relay service, then execute the STI prove and notification grant protocols. If the protocols complete successfully, the results are displayed as an "ID card" showing the user's photo, last test date, and color-coded STI status.

Trusted Parties. Testing laboratories are already trusted by patients to keep medical information. Our architecture assumes that the test laboratories can be trusted to accurately sign user credentials, and to generate global parameters. This seems a reasonable assumption given that test labs already interact with the user physically and have access to test results. Test labs in our system do *not* possess the ability to trace or de-anonymize credentials, nor are they required to store results once a credential has been issued.

The fact that a trusted authority already exists and is well regulated is a benefit. Many cryptographic protocols simply assume *a priori* the existence of trusted authorities, public key infrastructure, and strong notions of identity.

8.1 Data Access

HIPAA requires explicit, potentially written, consent to disclose medical data such as STI status. The disclosure burden can be negated by including a QR code in test reports that contains a signed message with the patient's test information which the patient herself will scan. This alleviates HIPAA-covered entities from managing release requests. Given that there are a small number of major medical labs, it seems feasible to standardize this process.

Enrollment In addition to a certified set of test results, enrollment requires a trusted party to confirm the user's photo. After downloading the app, the user will scan the test result QR code and then select or upload a profile picture. After scanning a QR code at the clinic, the clinic's computer will be prompt an employee to confirm that the user's profile picture is of them. Once this is done, the user is fully enrolled.

Data Security Because the mobile app contains sensitive data such as the user's STI status and her number of partners (by way of the number of notification tokens), this data should be encrypted. The app will provide data encryption via a passphrase, using a key derivation function such as PBKDF2, and AES or built in equivalents(e.g, the iOS keychain) when possible and secure.

Scenarios It is likely that TruSTI users will meet online; hence it may be useful to provide access to the protocols via a separate web interface. However, this raises additional security and privacy issues. We hope to address these in future work.

Notification Delivery The notification protocol of Section 7 does not handle the practical issue of *delivering* notifications. Clearly, sending them from or to an email address would break the anonymity requirements. The user could send notifications via an anonymous networking system, or even post them *publicly* with the one-time public key pk_{rand} used to address the message. This type of notification might be acceptable, given that notifications do not reveal user identities.

9 Future Work

There is a great deal of additional work to be done in the area of status certification and notification. While the protocols in sections 6 and 7 constitute the bulk of the cryptographic work for the TruSTI system, a number of additional components will be needed for a complete working implementation. In particular, we will need a design that protects confidential user secrets, as well as an interface for extracting signed test credentials from a testing laboratory.

More generally, the data provided by TruSTI is somewhat coarse: an infection status and a test date. A significantly better approach would be to calculate an individual's probability of exposure by evaluating their partners' test statuses and modeling the infection spread rates. This can be thought of as a problem in secure multi-party computation (specifically, a distributed problem in graph-theory) that could be solved *inefficiently* using general techniques, *e.g.,* [27]. Developing a practical approach would likely require new cryptographic techniques.

In addition to notifying partners, users might be willing to share anonymized test data with a public health department. The current iteration of TruSTI does not allow such data collection. Future work would include a system for anonymizing and aggregating test results, while preserving individual users' privacy. Lastly, there may be several other applications for the protocols we present in this work. For example, users might use TruSTI to share genetic test results.

10 Conclusion

Current electronic STI notification and certification systems have serious security and privacy issues. We present cryptographic protocols for securely and privately achieving both goals along with the ground work towards a real implementation. We stress that unlike most cryptographic solutions, this one is practical: regulated trusted parties already exist and the cryptographic components can be made transparent to the end user.

References

[1] Qpid.me: For health professionals. Available at `https://qpid.me/about/health`.

[2] BRANDS, S. Rapid demonstration of linear relations connected by boolean operators. In *EUROCRYPT '97* (1997), vol. 1233 of LNCS, pp. 318–333.

[3] BUHI, E. R., COOK, R. L., MARHEFKA, S. L., BLUNT, H. D., WHELDON, C., OBERNE, A. B., MULLINS, J. C., AND DAGNE, G. A. Does the Internet represent a sexual health risk environment for young people? *Sexually transmitted diseases 39*, 1 (Jan. 2012), 55–8.

[4] BULL, S. S., AND MCFARLANE, M. Soliciting sex on the Internet: what are the risks for sexually transmitted diseases and HIV? *Sexually transmitted diseases 27*, 9 (Oct. 2000), 545–50.

[5] CAMENISCH, J., HOHENBERGER, S., AND LYSYANSKAYA, A. Compact e-cash. In *EUROCRYPT '05* (2005), vol. 3494 of LNCS, pp. 302–321.

[6] CAMENISCH, J., AND LYSYANSKAYA, A. A signature scheme with efficient protocols. In *SCN '02* (2002), vol. 2576 of LNCS, pp. 268–289.

[7] CAMENISCH, J., AND MICHELS, M. Proving in zero-knowledge that a number n is the product of two safe primes. In *EUROCRYPT '99* (1999), vol. 1592 of LNCS, pp. 107–122.

[8] CAMENISCH, J., AND STADLER, M. Proof Systems for General Statements about Discrete Logarithms. *Science 75*, 260 (1997), 1–13.

[9] CAMENISCH, J. L. *Group Signature Schemes and Payment Systems Based on the Discrete Logarithm Problem*. PhD thesis, ETH Zürich, 1998.

[10] CHARNIGO, R., NOAR, S. M., GARNETT, C., CROSBY, R., PALMGREEN, P., AND ZIMMERMAN, R. S. Sensation Seeking and Impulsivity: Combined Associations with Risky Sexual Behavior in a Large Sample of Young Adults. *Journal of sex research* (Mar. 2012).

[11] CHAUM, D., AND PEDERSEN, T. P. Wallet databases with observers. In *CRYPTO* (1992), vol. 740 of LNCS, pp. 89–105.

[12] DWORK, C., NAOR, M., AND SAHAI, A. Concurrent zero-knowledge. *Journal of the ACM 51*, 6 (2004), 851–898.

[13] FIAT, A., AND SHAMIR, A. How to prove yourself: Practical solutions to identification and signature problems. In *CRYPTO '86* (1986), vol. 263 of LNCS, pp. 186–194.

[14] GHANEM, K. G., HUTTON, H. E., ZENILMAN, J. M., ZIMBA, R., AND ERBELDING, E. J. Audio computer assisted self interview and face to face interview modes in assessing response bias among STD clinic patients. *Sexually transmitted infections 81*, 5 (Oct. 2005), 421–5.

[15] GREATWEBGUY. inSpot.org a public service or a bad joke. From `http://greatwebguy.com/recommended/humor/inspotorg-a-public-service-or-a-bad-joke/`, October 2010.

[16] HOGBEN, M., AND KISSINGER, P. A review of partner notification for sex partners of men infected with Chlamydia. *Sexually Transmitted Diseases 35*, 11 Suppl (2008), S34–S39.

[17] IBISWORLD. Diagnostic and medical laboratories in the us: Market research report. From `http://www.ibisworld.com/industry/default.aspx?indid=1575`, February.

[18] LEHMANN, J. B., LEHMANN, C. U., AND KELLY, P. J. Development and health care needs of lesbians. *Journal of womens health the official publication of the Society for the Advancement of Womens Health Research 7*, 3 (1998), 379–387.

[19] LEVINE, D., WOODRUFF, A. J., MOCELLO, A. R., LEBRIJA, J., AND KLAUSNER, J. D. inSPOT: The First Online STD Partner Notification System Using Electronic Postcards. *PLoS Medicine 5*, 10 (2008), 4.

[20] MATT KAPP. Grindr: Welcome to the World's Biggest, Scariest Gay Bar, 2011.

[21] MCFARLANE, M., BULL, S. S., AND RIETMEIJER, C. A. Young adults on the Internet: risk behaviors for sexually transmitted diseases and HIV(1). *The Journal of adolescent health : official publication of the Society for Adolescent Medicine 31*, 1 (July 2002), 11–6.

[22] NAOR, M., AND YUNG, M. Deniable Ring Authentication. *Advances in Cryptology CRYPTO 2002 2442* (Sept. 2002), 481–498.

[23] PEDERSEN, T. Non-interactive and information-theoretic secure verifiable secret sharing. In *Advances in Cryptology—CRYPTO'91* (1992), J. Feigenbaum, Ed., vol. 576 of *Lecture Notes in Computer Science*, International Association for Cryptologic Research, Springer, pp. 129–140.

[24] SCHNORR, C. P. Efficient signature generation by smart cards. *Journal of Cryptology 4*, 3 (1991), 1–22.

[25] WHITTEN, A., AND TYGAR, J. D. Why Johnny can't encrypt: a usability evaluation of PGP 5.0. In *Proceedings of the 8th conference on USENIX Security Symposium - Volume 8* (Berkeley, CA, USA, 1999), SSYM'99, USENIX Association, pp. 14–14.

[26] WORKOWSKI, K. A., AND BERMAN, S. 2010 STD Treatment Guidelines. *Morbidity and Mortality Weekly Report 59*, RR-12 (2010).

[27] YAO, A. How to generate and exchange secrets. In *FOCS '86* (1986), pp. 162–167.