Social Security: Combating Device Theft With Community-Based Video Notarization

Alana Libonati
University of North Carolina
Chapel Hill, NC, USA
alana@cs.unc.edu

Apu Kapadia
Indiana University
Bloomington, IN, USA
kapadia@indiana.edu

Michael K. Reiter
University of North Carolina
Chapel Hill, NC, USA
reiter@cs.unc.edu

ABSTRACT

People increasingly rely on mobile devices for storing sensitive information and credentials for access to services. Because these devices are vulnerable to theft, security of this data is put at higher risk — once the attacker is in physical possession of the device, recovering these credentials and impersonating the owner of the phone is usually straightforward and hard to defend by purely local means. We introduce the concept of 'notarization', a process by which a remote notary verifies the identity of the device user through video chat. We describe the design and implementation of a system that leverages notarization to protect cryptographic keys that the device uses to decrypt device data (e.g., website passwords) or perform signatures in support of client-side TLS, without trusting the notary with these keys. Through a lab-based study with 56 participants, we show that notarization even by strangers is effective for combating device theft.

INTRODUCTION

By the end of 2011, there were nearly 6 billion mobile cellular subscribers worldwide [14], and a Pew report found that around one in two American adults owned smartphones in early 2012 [28]. We believe it is inevitable that these devices will become the primary portals by which humans interact with services, including remote services (e.g., banking and healthcare web sites) and more local ones (e.g., point-of-sale terminals or automatic teller machines, where the device may replace a credit or debit card). Since many of these services will be security-critical for the user, it similarly seems inevitable that mobile devices will be the repository for credentials, such as signature or decryption keys, which earn the supplicant access to these services or to local information (e.g., sensitive information downloaded from those services, not to mention passwords, private text messages, and emails).

Because these devices are mobile and nearly constantly carried, they are a common target of theft; e.g., according to research firm Gartner, 113 mobile phones are lost or stolen in the United States every minute [12]. As such, it is critical that these devices, or the credentials they hold, be rendered unusable in the wrong hands. Numerous tools exist to track and remotely erase data on stolen phones, but a thief can interfere with these by simply powering off the phone or putting the phone in 'Airplane mode', for example [15]. In the absence of tamper-proof hardware on the device, authentication of the user in a purely local fashion will be unable to protect against reverse-engineering the device and extracting the corresponding credentials. Consequently, in this paper we explore means to authenticate the device user by a remote entity that is physically out-of-reach of the attacker as a precondition for the device using the credentials it holds (c.f., [19]).

There are many alternatives by which this remote entity might authenticate the device user. Passwords or PINs (i.e., 'what you know') are one option, but these secrets are often guessed or stolen. Other solutions involve biometric recognition by fingerprint or face recognition (i.e., 'what you are'). However, biometrics can require hardware on devices that is not ubiquitous (e.g., for scanning fingerprints) and some means to ensure that the biometric readings are collected from the live user, versus being replayed (e.g., in the case of face recognition, from a stored video).

In this paper we explore 'who you know' as a novel alternative to authentication based on 'what you know' or 'what you are'. In this scenario, a person in the device owner's social network (who we term the *notary*) confirms that the current device user (the *supplicant*) is, in fact, the device owner — what one might call 'social security'. To do so, the notary interacts with the supplicant by video chat, for example (see Steps 1 and 2 in Figure 1). If the notary assents (Step 3), then the use of the device's credentials can progress as usual. However, if the notary refuses, then the use of the credentials will be blocked even by an attacker with physical possession of the device and the skill to reverse engineer it. Our approach, which we call *notarization*, also ensures that the notary cannot impersonate the device owner without physical access to the device. We expect that notarization is suitable primarily for protecting high-value data or transactions, e.g., transferring or withdrawing bank funds past some limit or decrypting sensitive files (e.g., health documents) or signing emails on the device.

Based on this idea, we detail the design of SSN (Social Security through Notarization), a device-resident application that supports notarization to enable the use of credentials on the mobile device. SSN specifically protects the use of cryptographic keys to decrypt device-resident content or to perform digital signatures in support of a connection using client-side TLS [30]. Moreover, it can support client-side TLS connections to web sites from a display computer other than the device hosting it. This usage requires additional software to be installed on the display computer as well, which we also describe.

Beyond notarization by someone in the device owner's social network, SSN also supports notarization by a stranger

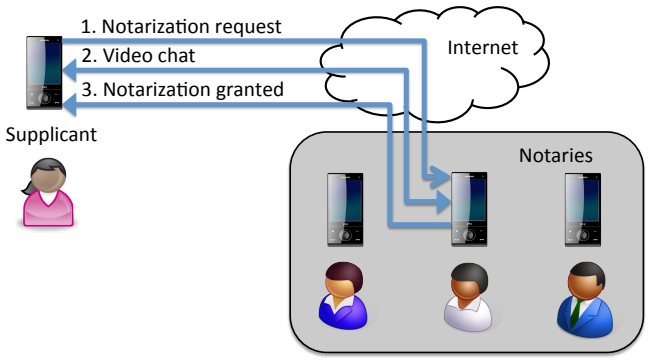

Figure 1. Notarizing a supplicant via video chat.

using a certified photograph of the device owner. Notarization by a stranger may be of use not only when the device owner's social contacts are unavailable for notarizing, but also in cases where specialized notaries (e.g., those working for a bank) may be required to notarize supplicants. While it is known that social contacts such as friends and colleagues would easily recognize supplicants [5], an open question is whether strangers can reliably notarize supplicants. Therefore, we present results from a detailed user study to shed light on this question and to evaluate user comfort with notarization through video chats in general. Our user study tested not only the ability of a notary to match interactive video of a stranger to her photograph, but also the frequency with which an attacker who has both a stolen device and a photograph of its legitimate owner — as he might obtain from the device itself, if the photos are not encrypted so that their decryption requires notarization — can, with the help of modern software for making that photograph appear alive, fool a stranger into believing she is interacting through video with the legitimate device owner. Following several rounds of notarization, participants answered a series of questions, which allowed us to evaluate their overall comfort with notarization.

To summarize, we make two contributions. First, we describe the design and implementation of SSN, which is novel in using notarization by video as a method for authenticating supplicants. We detail two use cases of SSN, one for decrypting passwords (for other services) stored locally on the device, and another to protect the establishment of client-side TLS connections. Moreover, this is accomplished without permitting the notary to impersonate the supplicant and without divulging sensitive information to the notary. Second, through a detailed user study, we shed light on the effectiveness of using strangers as notaries and users' overall comfort with notarization in practice. Our results show that even with impersonation attacks using sophisticated avatars, notarization by strangers can be a viable, if imperfect, method for protecting against impersonation attacks.

RELATED WORK

There have been several designs by which a person leverages others in his social network to enable access to resources, either to prevent someone who has stolen his device from doing so (as we do here) [29,35] or to regain access after losing one or both of his authentication factors [4, 26]. The high-level difference between our work and these previous works is that we focus specifically on video notarization, both enabling it through a comprehensive system design and implementation and evaluating it through a user study; previous works offer neither in the context of video based authentication or notarization. Particular choices that we make in our design offer further differences, moreover. For example, these works address only scenarios in which the device is used to access a *remote* resource (e.g., a web site) and require coordination with (and changes to) that remote resource. Due to the cryptographic mechanisms on which it builds [18, 19], SSN can provide protection for on-device data and, when used to protect access to remote resources, is fully interoperable with standardized and widely implemented protocols. Aside from making deployment easier, this also enables a conceptually distinct usage model in which the device owner can choose to utilize SSN unilaterally, in a fashion analogous to writing "check id" in the signature field of a credit card. While our choice of cryptographic mechanisms facilitates these goals, we stress that our focus and contribution lie in designing and evaluating a system for video notarization, versus the underlying cryptographic mechanisms that it employs.

Existing studies in psychology have shown the relative ease with which participants can identify familiar faces and the difficulty they have identifying unfamiliar ones [5–7]. These results (and this intuition) suggest that people would have little trouble identifying members of their social network, but that using strangers as notaries deserves some careful thought. Pike et al. [23] show that motion appears to aid recognition. However, none of the previous studies evaluated the use of interactive video as we do here.

With the increased ubiquity of mobile phones, there have been a number of systems that rely on them to help secure web authentication. To add an additional layer of protection against password theft, some services provide support for two-factor authentication by sending a unique code via SMS which must be entered following input of the usual password [1, 10]. This provides no protection against phone theft, however; in that case, security is reduced to knowledge of the password only. Other systems utilize trusted mobile phones to access websites securely from untrusted machines [8, 33]. While SSN also utilizes a mobile device to support access to websites from an untrusted display computer, our focus here is at least as much on protecting against the misuse of a stolen mobile device (using notarization by others) as it is on defending against compromise of the display computer.

DESIGN OF A VIDEO-CHAT NOTARIZATION SYSTEM

In this section we provide an overview of the design and implementation details of SSN.

Overview

Presently, SSN is designed to protect the use of keys of two varieties: private decryption keys for decrypting either passwords for entry to remote web sites or other data stored locally on the mobile device (e.g., emails, SMS messages), and

private digital signing keys that can be used to access remote web sites via client-side TLS [9].

Encrypted password use case

Our SSN-based password manager application supports local encryption of passwords used for entry to remote web sites. The user initiates access to a protected web site from the mobile device by selecting the URL from a list of bookmarks in the SSN application. The SSN application first checks to see whether the user has recently been notarized using the technique that the device owner specified for this URL when it was entered into the bookmarks. The two available password notarization techniques are password/PIN (where the local password can be decrypted by supplying another password or PIN to our remote cloud service; see Section 'Implementation and User Experience') and video-chat (where the local password is decrypted following a successful video chat with a notary). If notarization is required, the application initiates the notarization process. If the method of notarization required for this URL is video chat, then the application prompts the user to select a notary to notarize him, from a list of allowable notaries previously configured for this URL by the device owner. Upon selection of the notary (Step 1 of Figure 2), the SSN application establishes a video chat with the SSN application on the notary's device (Step 2), after which the notary can indicate (or not) the authenticity of the device owner (Step 3). If the notary is satisfied with the authenticity of the device owner, the notary's device conveys to the supplicant's device a capability (Step 4) that is valid for a *notarization interval* of a preconfigured amount of time. If the required method is password/PIN, our remote cloud service (see Section 'Implementation and User Experience') takes the place of the notary, sending a capability to the supplicant's device upon successful entry of the user's PIN.

During the notarization interval, the supplicant's device can interact with the notary's device (without interrupting the notary herself) in order to perform cryptographic operations (Steps 5–6). Protocols to force the supplicant's device to interact with the notary's device to perform cryptographic operations, without permitting the notary's device to learn the supplicant's private key, are well known; we employ protocols due to MacKenzie and Reiter [18, 19]. Briefly, these protocols cryptographically share the private key between the supplicant's and notary's devices, and permit the notary's device to perform a partial decryption of the stored credentials, using its share of the key. The supplicant's device can then complete the decryption using its share. The notary's device cooperates in this protocol only if presented the capability it generated during the notarization process, and only during the notarization interval.[1]

[1] Alternative protocols exist that remove the need for an interaction per decryption (Steps 5–6 in Figure 2), by reconstructing the private decryption key at the supplicant's device for the duration of the notarization interval [19]. We employ protocols that never recreate the private decryption key on the device, since recreating the private decryption key would allow a reverse engineer who captures the device during the notarization interval to extract it. Moreover, the device owner can destroy its authorization proactively (e.g., because he is done with his sensitive task) by simply deleting the capability, if he so chooses, to prevent an attacker who then captures the device from making use of the authorization.

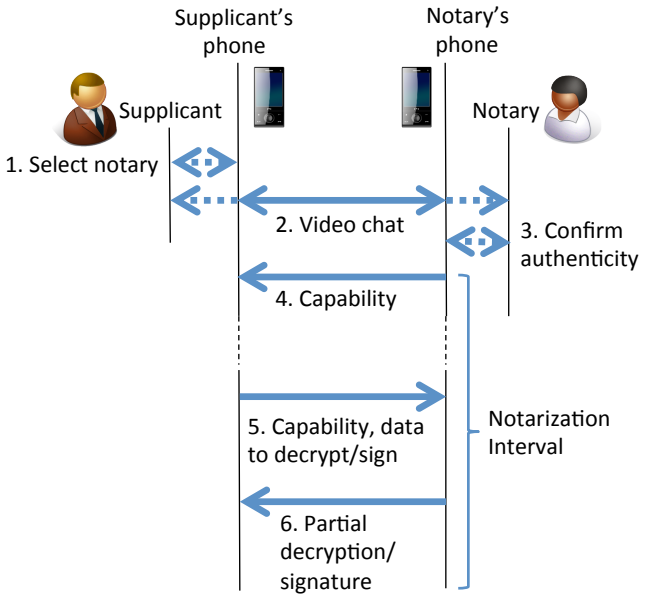

Figure 2. Overview of notarization process and control. Steps 1–4 are executed, if necessary, between steps 1–2 of Figure 3. Steps 5–6 are executed between steps 4–5 of Figure 3.

We assume the notary's device or cloud service is not compromised, though we stress that it (provably) has no capabilities to impersonate the devices it notarizes. Rather, its compromise can, at worst, reduce the supplicant's security to depending solely on the possession of her device, i.e., a single factor of authentication.

TLS use case

Using SSN to support client-side TLS is shown in Figure 3. Rather than encrypted credentials, client-side TLS certificates are stored when these URLs are bookmarked in the SSN application. After choosing a URL, the user then selects a computer to which this URL should be displayed (from a list of previously registered computers, which could include the phone itself); see Step 1 of Figure 3. To digitally sign for the client in a TLS exchange, SSN must gain access to the value to be signed. We obtain this value by routing TLS through a proxy local to the machine on which the browser is being run, which need not be the mobile device. This proxy exports the value to sign to the mobile device (e.g., over Bluetooth or TCP/IP), which signs the value (subject to the controls described below) and returns it to the proxy.

After checking whether the user has recently been notarized, the SSN application connects to a proxy on the designated computer and reports the URL indicated by the user and the client-side TLS certificate that the owner previously indicated for this URL (Step 2). The proxy initiates a client-side TLS connection to the web site (Step 3) and, at the appropriate time in that negotiation, forwards to the device (over the still-open connection) the TLS message requiring a digital signature with the private key corresponding to the public key in the client-side TLS certificate (Step 4). The device then

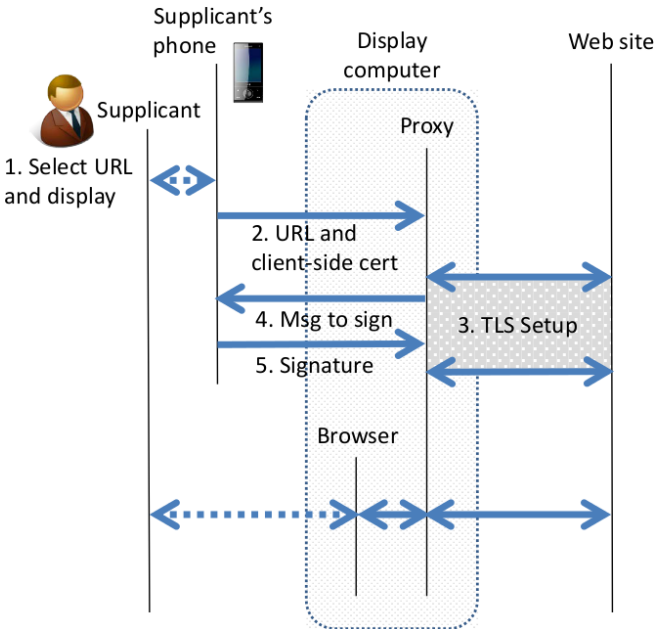

Figure 3. Use of SSN to establish TLS connection.

completes the notarization process and returns the signature (Step 5) allowing the proxy to complete the TLS exchange. Once the proxy has done so, it communicates to a browser extension to open the retrieved content in a new browser tab, and the user can interact with the page as normal.

Notarization by strangers

The set of possible notaries that the device owner can configure for notarizing the use of a URL includes, in addition to members of the device's address book, an 'Anyone' option. If a URL is configured so that the Anyone option is available for it, and if the supplicant selects this option in order to be notarized, then the supplicant's phone contacts a cloud-resident SSN service for notarizing the supplicant. In this case, the device must forward the selected encrypted password manager entry or client-side TLS certificate (but not its share of the private key, of course) to the service. Moreover, this certificate must have been created to include a photograph of the device owner. (We will discuss certificate creation in Section 'Initialization'.)

The role of the SSN service is to provide a portal for persons who are interested in notarizing others (presumably for pay, in a fashion similar to Amazon's Mechanical Turk) to be paired up with those needing notarization, or to otherwise implement a 'call center' for notarization of device users by trained notaries (in the case of a bank, for example). In this case, the notary is presented with the certified photograph of the device owner and a live video feed of the supplicant. The notary is then asked to confirm that the person in the video is pictured in the certified photograph and that the video feed is live, presumably by interacting with the supplicant. If the notary then indicates the authenticity of the supplicant, the SSN service sends a capability to the supplicant's device. During the notarization interval for that capability, the SSN service will respond to requests to sign messages or decrypt password manager entries by producing a partial signature or decryption using its share of the device's private key [18]. The process of notarization in the Anyone case is thus very similar to that in Figure 2, with the SSN service playing the role of the notary's phone.

Initialization

The process by which a device owner initializes his device for supporting notarization is not particularly complex. Below we describe the primary steps for initialization (aside from downloading the SSN application itself).

URLs URLs requiring authorization can be added to the SSN application by manual entry or by visiting the relevant URL in the phone's browser and selecting the option "Share Page" (Android) or clicking a custom bookmark (iPhone).

Notaries A list of possible notaries, which the user can assign to URLs manually, can be imported from the phone's address book. When a notary is first used, a new two-party sharing of the relevant private key is established with the notary's device through a delegation protocol [18]. Before a notary has been established for a key, it is important that the key is not stored in its entirety on the device. Thus, the initial two-party sharing of each private key is performed between the device and a cloud-resident SSN service — the same one that facilitates the Anyone option — immediately after the key is created. Delegating to a new notary therefore involves this service.

This delegation protocol requires a public key for the notary's device, which may be signed by a trusted certificate authority (CA), sent from the notary's device upon first use (i.e., a trust-on-first-use model, as is used in SSH), or obtained through an in-person key exchange [20]. The public key for the cloud-resident SSN service can be shipped with the SSN application or, again, established by trust-on-first-use. Note that decryption with the private key corresponding to the notary's public key (or, obviously, the cloud-resident SSN service's) should not itself require notarization. This key pair is used exclusively to support delegation.

Supplicants For the purposes of notarizing supplicants, a notary need not configure her SSN application except to import public keys with which to authenticate notarization requests from allowed supplicants. (Alternatively, the supplicant's device's phone number could be used to identify it, though obviously at a lower level of security.) Similar to the notary's key pair that supports the delegation protocol described above, these supplicant key pairs should be single-purpose and not require notarization to use. As above, a supplicant's public key may be signed by a trusted CA, follow a trust-on-first-use model, or be obtained by the notary's device through an in-person key exchange.

Display hosts A host to which the device owner plans to direct web pages will first need to have additional software installed on it beyond the web browser. This software will include the proxy to which the SSN device application will connect, the browser extension that permits the proxy to open tabs in the browser and provide content, and software for facilitating its 'pairing' with the device. The last of these dis-

plays the proxy's addressing information (presently we use the host's IP address and the port number on which the proxy listens, as well as the Bluetooth address of the host) in a 2-dimensional barcode on the host screen, permitting the SSN application on the device to import this information by photographing it [2, 20].

Client-side certificates Our SSN application supports the standard Certificate Signing Request (CSR) procedure [24] (also implemented by popular web browsers) for obtaining a client-side TLS certificate from a remote web site or from a CA that the remote site trusts. The primary addition that SSN requires for this process is the inclusion of a picture of the device owner in each certificate request for which notarization by Anyone is to be supported. Of course, since most smartphones and similar devices include a camera, obtaining a suitable picture should rarely pose a difficulty.

Privacy

Involving another person (the notary) in the process of notarizing a user raises the potential for privacy issues for both the notary and the supplicant. Here we briefly review the steps we have taken in our design to minimize those privacy risks.

Supplicant privacy Regardless of whether SSN is used to protect a device's signing key for client-side TLS sessions or to decrypt a ciphertext/password on the device, no cryptographic secrets are revealed to the notary's device that would permit it to impersonate the supplicant's (e.g., in the TLS session being established) or to recover the plaintext being computed. The URL or domain being accessed by the supplicant in a TLS establishment is also not directly revealed to the notary or his device. That said, in the TLS use case, a ciphertext created under the web site's public key is revealed to the notary's device. If the encryption algorithm used is not key-private [3], then this ciphertext can reveal statistical information about what web site is being accessed. SSN therefore cautions the user to select only notaries for a URL who he would be comfortable with learning that he had visited that site.

Notary privacy To protect the notary's privacy during notarization by Anyone, the video in this case is one-way: The notary can see the supplicant, but the supplicant can only hear the notary. Note that it is necessary for the notary to see the supplicant, to match him to the photograph displayed to the notary. Other notarization sessions, including those that involve a notary from the supplicant's social network, enable the notary to select per session whether the supplicant can see video of the notary.

Implementation and User Experience

Client software We have implemented SSN as an Adobe Air application in order to allow deployment to both Android and iOS smartphones. We wrote custom native extensions for Air to handle certain OS specific functionality, for example, device-to-device communication using Google Cloud Messaging (GCM) and Apple Push Notification service (APNs). The core cryptographic protocol in SSN is implemented using libgcrypt [17].

To support using SSN for setting up TLS sessions, the display computer runs a proxy that is an adaptation of mitmproxy [21], a Java-based SSL proxy that acts as a 'man in the middle.' We use a modified SSL implementation based on OpenJDK [22] to intervene in the SSL handshake as required by our protocol. Using the Google Web Toolkit [11], we developed a browser extension for receiving directions from the proxy to display content in a new tab. This extension also includes code from the open source ZXing multi-format 1D/2D barcode image processing library [36] to handle QR code generation. We employ the jWebSocket Java websocket server [13] to facilitate TLS-secured communication between the device and the proxy and between the proxy and the browser extension via our custom plugins.

Server software We have implemented the cloud service for initialization, delegation (sharing of keys), and managing notarization by strangers using a similar set of tools as our mobile device application and are currently hosting it on our own server. Since we do not expect users of SSN to maintain their own application in the cloud, this type of service is something we can imagine being offered by a service provider.

User interface The common-case use of the SSN application on a mobile device involves a simple menu-driven interface, e.g., to select a notary or a URL, and then a host display. The saved list of URLs contains both sites for which the user holds a TLS client-side certificate and sites that require a password-based login. Notarization conducted via video-chat by a member of the device owner's social network (vs. by a stranger) presents an interface as pictured in Figure 4 to the notary. The notary's interface asks her to respond to two questions during the video chat, specifically whether this supplicant appears to be the correct device owner and whether the supplicant video appears to be *live*, i.e., not a recording, which the notary ideally determines by interacting with the supplicant. We discuss this possibility further in Section 'Efficacy of Video-Chat Authentication'.

The notary interface for use by a stranger, i.e., one contacted by way of the SSN cloud service (see Section 'Notarization by strangers'), is similar to that pictured in Figure 4, except that rather than asking "Is this Alana?", the interface allows the notary to toggle between the supplicant video and a pane in which he can rotate through three different photos. (A similar interface is presented in our study in Section 'Efficacy of Video-Chat Authentication'.) One of these photos will be the certified photograph of the device owner, and the other two will be photographs of others who are of the same gender and race as the device owner (e.g., as specified in the device owner's certificate, along with his photograph). The notary is then asked to identify the photo corresponding to the person in the video, as well as to confirm that the video is live. Our use of a three-photo 'lineup' style interface for strangers who are notaries is motivated by studies indicating that lineups can improve performance in identification tasks [34], but it is not fundamental to our design.

There are several avenues for investigating the human aspects of using SSN, including the accuracy of video-based authentication and users' willingness to employ it or to notarize oth-

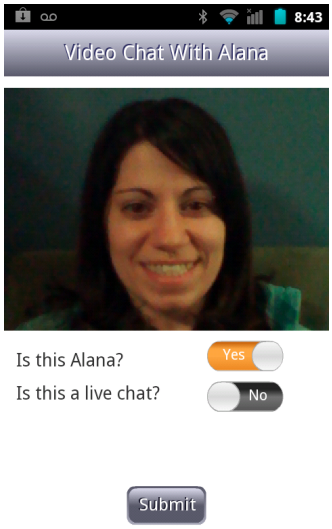

Figure 4. Prototype notary interface of SSN used by a named notary (vs. by a stranger). The video shows the supplicant. The notarization request came from Alana's device, as indicated by the "Is this Alana?" question in upper right.

ers to access critical resources. We provide an evaluation of these issues in the next section.

EFFICACY OF VIDEO-CHAT AUTHENTICATION

Recall that in addition to notarization by members of a device owner's social network, SSN also supports notarization by a stranger. While it is likely that notaries in a supplicant's social network can easily identify the supplicant [5], we wanted to know if strangers could perform this task in the event that notaries in one's social network are unavailable or, as mentioned earlier, in cases where trained notaries in a call-center may be used. In addition, we also wanted to evaluate how comfortable users might be with our approach in practice. We therefore conducted a user study with three goals: 1) to learn *how accurately notaries can identify supplicants* whom they do not know through a video conversation (by matching the person in the video against a set of photographs); 2) to learn *how reliably a notary can test the liveness of the supplicant*, i.e., how well the notary can distinguish a live video of a supplicant from a generated video, including one that is being manipulated to appear responsive to the notary's requests; and 3) to learn how comfortable users are with interacting over video chat both in general and specifically for seeking notarization, and to understand user perceptions and comfort with identification through video.

The threat model that gives rise to goal 2) is one in which an attacker both obtains a photograph of the owner and steals his device. After all, virtually all smartphones today include camera functionality, and so a device is likely to contain photographs of its owner. If these photographs are not encrypted so that notarization via SSN is required to decrypt them, then the attacker would be able to use a photograph of the device owner in his efforts to fool notaries. Commercially available software can enable the attacker to manipulate the photograph to appear dynamic (e.g., causing its eyes and mouth to move as needed), and so it is conceivable that to a stranger, this generated video of the device owner overlaid with the audio of the attacker (so that he can easily respond to notary questions, for example) would be convincing to a notary who is unfamiliar with the supplicant. A goal of our experiment was to evaluate how convincing such a generated supplicant is.

Method

Overview of the study

Our study comprised a set of lab sessions. In each session, each participant was randomly assigned to be either a supplicant or a notary. Those participants assigned to be notaries were always physically separated from those participants who were assigned to be supplicants so that they would not see each other or interact with each other. This was done to minimize any familiarity we might introduce extraneously and thereby influence the notarization process between strangers.

Each experiment then proceeded through multiple rounds in which notaries and supplicants were paired up for video chat. In each such pairing, the notary was instructed to interact with the supplicant and then make a decision about whether the supplicant was present in a set of three images, and if so, which one.

The supplicant's photo was always present in the set, but the notaries were not made aware of this fact. The notary was also instructed to test for liveness of the supplicant with which they were interacting . Because we were interested in understanding the methods participants would employ to determine liveness, we did not instruct notaries about how to determine if the video was of a live and present supplicant.

At the conclusion of each decision, the notary answered a brief questionnaire to indicate her degree of confidence that 1) the selected photo represented the supplicant and 2) that the video represented a live and present supplicant. Using the chosen photo and the liveness confidence we are able to determine the identification rate, as discussed below.

To measure the potential for tricking notaries into falsely believing they were interacting with live and present supplicants, we challenged notaries with custom avatars that were manipulated to be responsive to notary interaction. The avatars in these video feeds were created from photos of supplicants who were not part of the current lab session. We instructed the supplicants who were controlling these avatars to act naturally during these chats and to try to convince the notary that they were in fact the person in the video feed. The avatar chats were made to appear identical to the live chats, the only exception being that the notary (unbeknownst to them) was speaking to a supplicant who was not the person depicted in the video they were seeing. The photo set viewed by the notary included an image of the supplicant whose avatar appeared in the video feed, since this was meant to mimic an impersonation attack where fabricated video might be used in an attempt to match a device owner's certified photo.

Study implementation

Obtaining images As part of the recruitment process, participants were asked to submit three photos of themselves. From

this collection of images, a photo set was created for each participant containing one photo of this participant and photos of two other participants. Our goal was to create sets of images where all three people were similar in appearance in order to test the notary's ability to make a correct identification. An attempt was made to match gender, ethnicity, age, hair color, etc. whenever possible. This was sometimes challenging, for one because there were more than twice as many female participants than male, and also because we were limited in our choice of the two additional photos to participants who would not be present at the current lab session. For instance, if a notary viewed a photo set containing a photo of the person seated next to them, they could automatically eliminate that person from consideration when making their decision. Also note that a photo set had to be created for every participant since we had no way of knowing ahead of time who would be a notary or a supplicant during any given experiment (see below).

Lab setup and group assignment Upon arriving at our lab site, the participants were alternately sent to two different labs in order to divide them as evenly as possible into notaries and supplicants and to provide a randomized assignment to the supplicant or notary groups. The notary lab was equipped with five desktop computers, each of which had an attached headset with microphone. The supplicant lab also had five computers, each with an attached webcam, microphone, and speaker. (Supplicants were not provided headsets, since headsets would obscure the supplicants' physical appearance to the notaries.) Both rooms had group-specific FAQ sheets placed next to each computer, as well. Before the start of each lab session, members of our study team gave each group a brief introduction outlining the purpose of the study and detailing their role as notary or supplicant.

Minimizing extraneous participant interactions Participants were told to arrive in the lobby of our building where they would then be directed to the appropriate room by a member of our study team. To avoid accidental interaction between the groups, each participant was given a map with a highlighted path to their room, using separate hallways and stairwells for each group. As another precaution, each notary and supplicant was presented with a question immediately following each chat which asked whether she had ever interacted with the person they just chatted with before that day. We collected this data so that we could exclude any such chat pairings from our analysis in an effort to ensure that we were only looking at notarization between strangers. (For this reason, 2 out of 80 chats were excluded from consideration.)

Study interface At the start of each lab session, participants viewed a short walkthrough video detailing their role (either notary or supplicant) and the usage of their video-chat software. The supplicant's software sent both video and audio feeds to the notary with whom he was interacting, while the notary software sent audio only. The notary interface is shown in Figure 5. Both systems utilized a push-to-talk interface including an onscreen indicator to show which (if any) side was currently speaking; the reason for this choice is described below. Notaries were told to interact with the supplicant and compare their video feed to the provided photo set in order to verify the supplicant's identity and to verify that the video feed is of a live and present supplicant (versus a recording, for example). The first round was used for practice and could be repeated if desired. This was done to ensure that participants were comfortable using the software. The data from this first round was not analyzed.

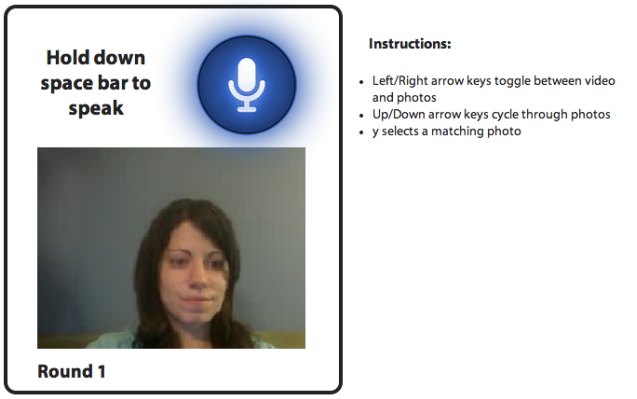

Figure 5. The notary user interface in the user study described in Section 'Efficacy of Video-Chat Authentication'. Video shows supplicant Microphone circle is red while supplicant presses a key to talk; blue when neither party is pressing a key; and green while (only) the notary presses a key to talk. As the instructions indicate, the notary can toggle between the live video stream and three photos of the same size as the video. The notary cycles through these three photos using the up/down arrow keys. The notary indicates her identification of the supplicant by pressing "y" while the intended photo is displayed.

The specific assertions presented to the notary after he selected a photo that he believed to be the supplicant were:

- "I am sure this photo matches the person in the video."
- "I am sure this was a live conversation and not a recording."

To each, the notary responded on a Likert-type scale with values "Strongly disagree", "Disagree", "Neutral", "Agree", and "Strongly agree".

The supplicant's user interface is similar to the notary's, with three important exceptions. First, the supplicant interface shows the video of the supplicant, not of the notary, so that the supplicant can see what the notary is seeing. (Recall that notarization by strangers involves video in only one direction but audio in both.) Second, the instructions on the right half of the screen were unnecessary for the supplicant, since the supplicant has no controls to manipulate during the notarization process. Third, of course the supplicant did not receive questions at the end of a round asking her confirm the identity or liveness of the other party.

The notary's interface was adapted to reflect technical limitations that would be typical of video-chatting over mobile devices. For example, the notary's video display was limited to a size approximately that of a modern smartphone screen. Moreover, we inserted randomly generated 'freezes' and 'skips' into the video to mimic glitches typical of live video chats. To produce these effects, we randomly applied

one of two custom filters to the video display. Both filters applied a slight pixelation to the video, and one inserted approximately half-second pauses every 12 seconds on average while the other inserted approximately one-second pauses every 8 seconds on average.

Avatar creation We used the SitePal service [27] to create avatars based on photos of participants from other lab sessions. A photo of the supplicant lab was used as the background image for the avatars so that they would not appear different from the live supplicant video feeds. The avatars were controlled by a supplicant whose real voice was heard by the notary even though the video feed was falsified. As described in Section 'Study implementation' supplicants were habituated to use a push-to-talk system for speech, and these inputs caused the lips of the avatar to move while the supplicant was speaking. We created both male and female avatars and ensured that the gender of the avatar matched that of the controlling supplicant.

Study orchestration After each participant viewed the walkthrough video, they entered their assigned participant ID number into our software's web interface to join the session. Once everyone had joined, one of the study team members would start the session via an administrative web interface. Starting the session in this manner was necessary in order to create the notary-supplicant pairings based on who actually showed up to the session. When making these pairings, the software also made sure that each notary would see one avatar during a random round (after the first round), and that each supplicant would act in the avatar role at most one time. The software also made it possible to repeat the first (practice) round if either side chose to do so, and also automatically advanced through the rounds once all the chats for the current round were completed.

Software implementation notes We implemented our study software as a Google Web Toolkit application with a MySQL backend. The video chat component was written in Actionscript and embedded in the web interface as a Flash movie. We used the open source Red5 Media Server [25] to relay the video and audio streams and various other inputs to control the push-to-talk interface, the round changes, and the avatar actions.

Study procedures

Recruitment Study participants were recruited via flyers placed in several high-traffic areas on the UNC Chapel Hill campus and email announcements sent to a campus listserv. To be eligible for the study, participants must have been born in the US, lived in the US at least through high school, and be at least 18 years of age. The US restriction was put in place to limit variation in speaking accents since supplicants would sometimes be required to impersonate others and we wanted these situations to appear as natural as possible. Interested participants were directed to our website where they were asked to submit three face images of themselves taken on three different occasions, and to sign up for a time when they could visit our lab to participate in a video chat session. Potential participants were offered $20 for completing one of these sessions, or a prorated amount if they terminated the study early. 97 people filled out this form and due to scheduling constraints we were able to invite 74 of them to come to one of our scheduled lab sessions. Of the 74 that we invited, 62 actually showed up for a lab session.

Participant demographics One of our sessions, with 6 participants, experienced a software malfunction and thus our results are based on experimental data gathered from 56 participants (26 notaries and 30 supplicants). At the end of each lab session, participants filled out a brief questionnaire that asked them to indicate their gender, race, and age. This data is summarized in Table 1. We also asked the participants how often they have video conversations. These responses are presented in Figure 6.

Table 1. Study Demographics

Variable	Notaries (n = 26)	Supplicants (n = 30)	Total (n = 56)
Male	7 (27%)	8 (27%)	15 (27%)
Age 25 or under	22 (85%)	27 (90%)	49 (88%)
Caucasian	14 (54%)	16 (53%)	30 (53%)
African American	8 (31%)	9 (30%)	17 (30%)
Asian	4 (15%)	4 (13%)	8 (14%)

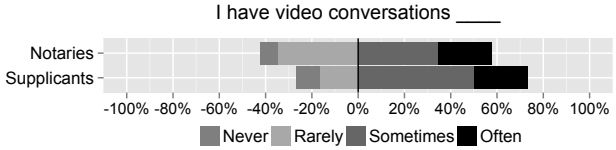

Figure 6. Net stacked distribution graph representing responses to the statement "I have video conversations ____". This statement was presented using a four-point Likert-type scale.

Ethical considerations Our user study was approved by UNC Chapel Hill's Institutional Review Board (IRB).

Findings

Identification accuracy

Our primary measures of participant performance were true and false identification rates. Recall that identification here involved two facets: selection of the correct supplicant photograph and confidence that the video session was a live representation of that supplicant. Since each aspect was given a confidence score by the notary at the end of the round, we needed some way to combine these scores to determine whether the notary's responses indicated sufficient confidence to declare the supplicant notarized. Specifically, we mapped responses on the Likert-type scale to numeric values ("Strongly disagree" $\rightarrow -2$, "Disagree" $\rightarrow -1$, "Neutral" $\rightarrow 0$, "Agree" $\rightarrow 1$, and "Strongly agree" $\rightarrow 2$) and defined the notary's *score* to be the minimum of his expressed photo confidence and his liveness confidence. We define the *true identification rate* (TIR) as the fraction of video chats *with live supplicants* after which the notary selected the supplicant's photograph and registered a score (as just defined) of at least a specified threshold t. The *false identification rate* (FIR) is then the fraction of video chats *with supplicant avatars* after which the notary selected the photograph matching the avatar and registered a score of at least t.

The One Notary ROC curve in Figure 7 then results by varying t in the range $[-2, 2]$. For example, setting $t = 2$ yields a TIR of over 50% and simultaneously an FIR of roughly 5%. On the other end of the spectrum, setting $t = -2$ yields a TIR of over 85% but also an FIR of roughly 80%. A balance point, i.e., at which $1 - \text{TIR} \approx \text{FIR}$, comes at around $t = 1$, in which case $1 - \text{TIR} \approx \text{FIR} \approx 24\%$.

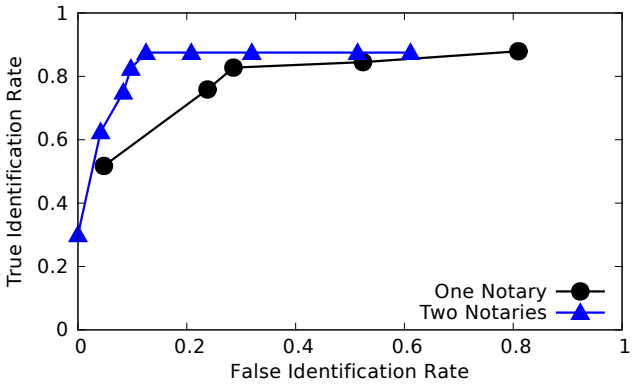

Figure 7. ROC curve illustrating true and false identification rates in the user study described in Section 'Efficacy of Video-Chat Authentication'.

We also show a Two Notaries ROC curve in Figure 7 that is constructed by combining the scores from each pair of video chats by two notaries with the same supplicant (or avatar based on the same human supplicant) in our study. Specifically, for each such pair of video chats, the scores of the two notaries were summed and compared to a threshold t, now ranged over $[-4, 4]$. As before, a combined score of at least t resulted in an identification for the purposes of computing a TIR and FIR. As Figure 7 shows, employing a pair of notaries in this way improves the ROC curve so that, e.g., its balance point at $t = 0$ yields $1 - \text{TIR} \approx \text{FIR} \approx 12\%$.

Liveness testing

One of the more interesting aspects of our study was learning how notaries would determine that they were speaking with a real person, i.e., that the supplicant was live and present. Note that we did not give participants any insight into the specific form of attack that our study attempted, i.e., one with live human audio overlaid on a manufactured video. Therefore, it is not surprising that some notaries adopted strategies that would be ineffective against this form of attack, as indicated in their responses to the post-study question, "What did you do to ensure that a live supplicant was present?" For example, most of the ineffective strategies tested (at best) only the liveness of the audio (but not of the video):

- "Ask what time it was, attempted to ask questions that would be difficult to give a stock answer to";
- "Asked questions about the present, like if they had a test etc.";
- "Ask questions that were not just yes or no answers."

Despite such cases, the majority of answers to this question indicated that notaries recognized the need to determine the liveness of both the audio and the video either initially or once something about the video alerted them. For example:

- "Had a conversation, told jokes to see if they laughed. Maybe my jokes are just bad?";
- "I asked the time and I asked them to make a funny face. My thought was that it tested both the 'live-ness' of the audio and the video.";
- "Ask them simple questions and ask them to do things like wave their hand over their head"

User comfort

In a questionnaire at the end of their participation in the study, the majority of participants indicated that they were comfortable interacting through video chat. Figure 8 presents the responses to this question as a net stacked distribution graph. The total width of each bar is equal to the percentage of non-neutral responses. More specifically, the overwhelming majority of supplicants indicated that they were comfortable seeking identification from another person through video chat (see Figure 9). When asked to rate identification through video, both notaries and supplicants were very positive (see Figure 10). These responses point to the overall comfort with using video chat for notarization.

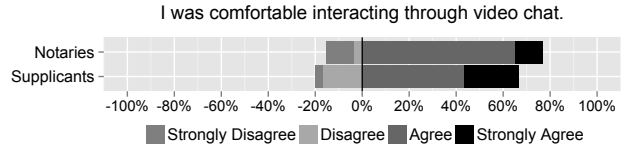

Figure 8. Net stacked distribution graph representing non-neutral responses to the statement "I was comfortable interacting through video chat." This statement was presented using a five-point Likert-type scale.

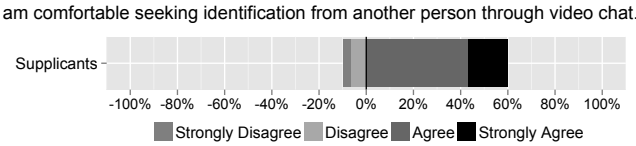

Figure 9. Net stacked distribution graph representing non-neutral responses to the statement "I am comfortable seeking identification from another person through video chat". This statement was presented using a five-point Likert-type scale.

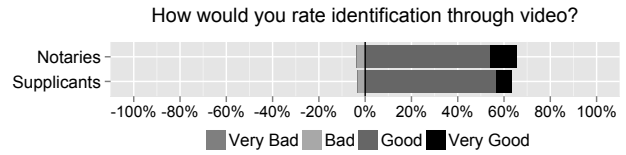

Figure 10. Net stacked distribution graph representing non-neutral responses to the question "How would you rate identification through video?". This question was presented using a five-point Likert-type scale.

Written responses from the participants indicated a degree of awkwardness in many cases, though this seemed to differ somewhat between notaries and supplicants. In part, this may have been due to the one-wayness of the video stream. For example, one notary wrote, "It (oddly) was more comfortable knowing that I could see them, but they couldn't see me." In contrast, one supplicant noted, "It was just a little odd because I couldn't see the other person," and another said, "I

usually feel uncomfortable chatting where someone can see me, but I can't see them."

A number of other useful insights came from the participant responses. For example, one notary indicated that it would have been helpful to have more photos to which to compare:

> I think it is easy to identify someone through a video, it may just be hard to know is they match one certain photograph. If I was given ten pictures of a person I could definitely tell which set belonged to which video chat person.

Another notary pointed out that ethnicity impacted his ability to correctly identify supplicants (though we presume he meant race, not ethnicity): "It's harder to identify those of other ethnicities than my own." In fact, it is well-known that people better recognize faces of people from their own races than from other races [16, 31, 32]. When using strangers as notaries, it may thus be advisable to utilize strategies that maximize supplicant and notary similarity.

Implications

The contribution of our user study is threefold. First, notarization by strangers is an imperfect defense, though it clearly provides a more useful middle ground than disallowing notarization entirely (the equivalent of a zero TIR) when no notary in a supplicant's social network is available. Our study suggests that when using a single stranger as a notary, careful thought should be given to selecting an appropriate confidence threshold. A threshold can be chosen to strike a balance between TIR and FIR, though for many practical uses it may be acceptable to decrease this threshold to improve the TIR with a corresponding detriment to the FIR. This tradeoff may be particularly attractive if the threat model under which we evaluated the FIR is considered more advanced than would be common. Our results also suggest the alternative of using two notaries, which generally yields better results than one notary but also comes with increased inconvenience. Other possible improvements suggested by participants include utilizing more photos per supplicant and utilizing notaries who are physically similar to the supplicants.

Though the true identification rates are not as high as we would like, recall that reliance on strangers for video notarization would generally be a last resort for when no notary from within the supplicant's social network is available. The false identification rates also fall short of the ideal, but recall that this measure represents the most difficult case for SSN: an attacker who steals a device, uses a photograph of the owner and state-of-the-art software to create a life-like avatar for the owner, and then accesses a resource for which Anyone is an allowable notary. Moreover, to prevent the attacker from trying strangers repeatedly until one assents, the SSN service can suspend the device after some number of consecutive notary rejections.

A second takeaway message is that while several notaries figured out effective measures to test the liveness of both the audio and video, some did not. It may be that when we provide training to the notary regarding methods to test the liveness of both video and audio that recognition accuracy would increase.

Third, we found that many of the participants in our study were generally comfortable with authentication through video chat. We believe this bodes well for the potential for a system like SSN to be accepted by users. Finally, the comments participants made about their use of the system helped us to identify ways to improve the system in the future.

Limitations and Future Work

There are, of course, several limitations to our study. Like most studies, our participants are not representative of the general population; ours were younger, better educated, and presumably mostly affiliated with our university in some fashion, for example. The extent to which our results generalize to the broader population is unclear, though since the duties of a notary rely on interpersonal interaction skills that people of all walks of life exercise on a daily basis, we would expect that our study might generalize quite well.

A natural concern about using strangers as notaries is the possibility that notaries will not take their responsibility seriously. Our study did not address this issue, and we did observe varying levels of commitment on the part of the notaries. We leave as future work the design of incentive schemes to motivate notaries to do a good job.

A third limitation of our study is that the avatars we constructed, though reasonably effective, were not perfect and presumably were well below the state-of-the-art of modern video and audio production. It seems likely that with access to state-of-the-art tools and expertise in special effects and animation, and with enough patience and motivation, an attacker could construct a video representation of nearly anyone that would fool a stranger (though perhaps not a friend). Nevertheless, we believe that notarization substantially raises the bar for all but very targeted attackers.

Finally, while participants in our study indicated a generally high level of comfort with video notarization by/of strangers, we would need to further study users' acceptance and behaviors in practice. We plan such a deployment and associated field trial after we have matured our prototype through limited field trials and pilot studies and improved it based on the findings from these studies. The field trial will shed light on various social and behavioral questions related to notary preference, privacy, and motivations of users for example. We expect to be able to deploy and collect results from such a field trial over the course of the next 12–18 months.

CONCLUSION

We have introduced the concept of 'notarization', a process where a remote entity (the notary) can verify via video chat who is in physical possession of a mobile device as a necessary condition for the device to make use of its cryptographic credentials. We implemented SSN, an Android application using notarization to protect cryptographic keys used for decrypting on-device data or signing in support of client-side TLS. Since SSN decrypts standard ciphertexts and produces standard digital signatures with the private keys it protects, it is interoperable with existing protocols (e.g., client-side TLS)

and so users can unilaterally decide which services and data they wish to protect using it. Through a detailed user study, we evaluated the accuracy and user comfort with video-chat based notarization and the possibility of extending the notary role to users outside of one's social network. In particular, our user study allowed for sophisticated adversaries that use modern photo animation software to synthesize an interactive video of the legitimate device owner from a photo of that owner. We showed that while strangers do not make perfect notaries, they are still viable as a last resort when no notary in a supplicant's social network is available, especially considering that the threat model in our evaluation is likely more advanced than would be common.

Acknowledgment

This material is based upon work supported by the National Science Foundation under Award Nos. CNS-1228471 and CNS-1228364.

REFERENCES

1. Bank of America SafePass. **http://www.bankofamerica.com/privacy/index.cfm?template=learn_about_safepass**.
2. Bauer, L., Garriss, S., McCune, J. M., Reiter, M. K., Rouse, J., and Rutenbar, P. Device-enabled authorization in the Grey system. In *Information Security: 8th International Conference, ISC 2005*, vol. 3650 of *Lecture Notes in Computer Science* (2005), 431–445.
3. Bellare, M., Boldyreva, A., Desai, A., and Pointcheval, D. Key-privacy in public-key encryption. In *Advances in Cryptology – Asiacrypt 2001 Proceedings*, vol. 2248 of *Lecture Notes in Computer Science* (2001).
4. Brainard, J., Juels, A., Rivest, R., Szydlo, M., and Yung, M. Fourth factor authentication: Somebody you know. In *13th ACM Conference on Computer and Communications Security* (2006), 168–178.
5. Bruce, V., Henderson, Z., Newman, C., and Burton, A. Matching identities of familiar and unfamiliar faces caught on CCTV images. *Journal of Experimental Psychology-applied 7* (2001), 207–218.
6. Bruce, V., Henderson, Z., Newman, C., and Burton, A. M. Verification of face identities from images captured on video. *Journal of Experimental Psychology-applied 5* (1999), 339–360.
7. Burton, A. M., Wilson, S., Cowan, M., and Bruce, V. Face recognition in poor-quality video: Evidence from security surveillance. *Psychological Science 10*, 3 (1999), 243 – 248.
8. Ch, R., Jammalamadaka, R., Horst, T. W. V. D., and Mehrotra, S. Delegate: A proxy based architecture for secure website access from an untrusted machine. In *Proceedings of 22nd Annual Computer Security Applications Conference (ACSAC* (2006).
9. Dirks, T., and Rescorla, E. The transport layer security (TLS) protocol, version 1.2. IETC RFC 5246, Aug. 2008.
10. Google 2-step verification. **https://support.google.com/accounts/bin/topic.py?hl=en&topic=28786&parent=2373945&ctx=topic**.
11. Google web toolkit. **https://developers.google.com/web-toolkit/**.
12. Hackers, IT units focusing on smartphone security. **http://www.reuters.com/article/2011/12/30/us-mobile-security-idUSTRE7BT0GV20111230**.
13. jwebsocket. **http://jwebsocket.org/**.
14. Key statistical highlights: ITU data release June 2012. **http://www.itu.int/ITU-D/ict/statistics/material/pdf/2011%20Statistical%20highlights_June_2012.pdf**.
15. Komando, K. Lost or stolen smartphone? Find and erase it remotely. *USA Today* (November 12 2009). Available at **http://www.usatoday.com/tech/columnist/kimkomando/2009-11-12-lost-smartphones_N.htm**.
16. Levin, D. Race as a visual feature: Using visual search and perceptual discrimination tasks to understand face categories and the cross race recognition deficit. *Quarterly Journal of Experimental Psychology: General 129*, 4 (2000).
17. Libgcrypt. **http://www.gnu.org/software/libgcrypt/**.
18. MacKenzie, P., and Reiter, M. K. Delegation of cryptographic servers for capture-resilient devices. *Distributed Computing 16*, 4 (2003), 307–327.
19. MacKenzie, P., and Reiter, M. K. Networked cryptographic devices resilient to capture. *International Journal of Information Security 2*, 1 (2003), 1–20.
20. McCune, J. M., Perrig, A., and Reiter, M. K. Seeing-Is-Believing: Using camera-phones for human-verifiable authentication. *International Journal on Security and Networks 4*, 1–2 (2009), 43–56.
21. mitm-proxy. **http://crypto.stanford.edu/ssl-mitm/**.
22. Openjdk. **http://openjdk.java.net/**.
23. Pike, G. E., Kemp, R. I., Towell, N. A., and Phillips, K. C. Recognizing moving faces: The relative contribution of motion and perspective view information. *Visual Cognition 4*, 4 (1997), 409–438.
24. PKCS #10: Certification request syntax standard. **http://www.rsa.com/rsalabs/node.asp?id=2132**.
25. Red5 media server. **http://www.red5.org/**.
26. Schechter, S., Egelman, S., and Reeder, R. It's not what you know, but who you know — a social approach to last-resort authentication. In *27th ACM Conference on Human Factors in Computing Systems* (Apr. 2009).
27. SitePal. **http://www.sitepal.com**.

28. Smith, A. Nearly half of American adults are smartphone owners. Tech. rep., Pew Research Center, 2012.

29. Soleymani, B., and Maheswaran, M. Social authentication protocol for mobile phones. In *2009 International Conference on Computational Science and Engineering* (Aug. 2009), 436–441.

30. The transport layer security (tls) protocol version 1.2. `http://tools.ietf.org/html/rfc5246`.

31. Valentine, T., and Endo, M. Towards and examplar model of face processing: The effects of race and distinctiveness. *Quarterly Journal of Experimental Psychology 44* (1992).

32. Walker, P., and Tanaka, W. An encoding advantage for own-race versus other-race faces. *Perception 23* (2003).

33. Wu, M., Garfinkel, S., and Miller, R. Secure web authentication with mobile phones. In *DIMACS Workshop on Usable Privacy and Security Software* (2004).

34. Yarmey, A. D., Yarmey, A. L., and Yarmey, M. J. Face and voice identifications in showups and lineups. *Applied Cognitive Psychology 8*, 5 (Oct. 1994), 453–464.

35. Zhan, J., and Fang, X. Authentication using multi-level social networks. In *Knowledge Discovery, Knowlege Engineering and Knowledge Management, First International Joint Conference* (Oct. 2009), 35–49.

36. Zxing. `http://code.google.com/p/zxing/`.