Defending Against Device Theft with Human Notarization

(Invited Paper)

Alana Libonati*, Kelly Caine†, Apu Kapadia‡ and Michael K. Reiter*
*University of North Carolina, Chapel Hill, NC
Email: {alana,reiter}@cs.unc.edu
†Clemson University, Clemson, SC
Email: caine@clemson.edu
‡Indiana University, Bloomington, IN
Email: kapadia@indiana.edu

***Abstract*—People increasingly rely on mobile phones 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 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.**

I. Introduction

As of early 2013, 56 percent of American adults owned smartphones [1], and by the end of 2014, there will be nearly 7 billion mobile cellular subscribers worldwide [2]. 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 user 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).

These devices are mobile and nearly constantly carried, and so are often stolen; by one account, 113 mobile phones are lost or stolen in the U.S. per minute [3]. So, it is critical that these devices, or the credentials they hold, be rendered unusable in the wrong hands. 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 [4]. Without 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., [5]).

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. To do so, the notary interacts with the supplicant by video chat, for example (see Steps 1 and 2 in Fig. 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., withdrawing bank funds past some limit, decrypting sensitive files (e.g., health documents), or signing emails on the device.

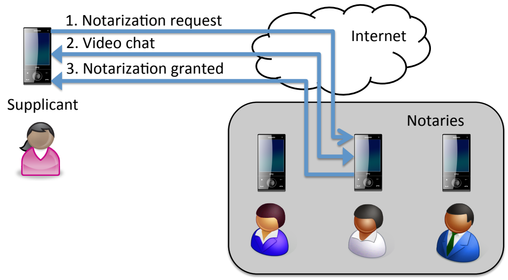


Fig. 1. Notarizing a supplicant via video chat.

COLLABORATECOM 2014, October 22-25, Miami, United States

DOI 10.4108/icst.collaboratecom.2014.257706

Based on this idea, we present DNo (Device Notarization), a device-resident application that supports notarization to enable the use of credentials on the mobile device. DNo 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 [6]. Moreover, it can support client-side TLS connections to web sites from a display computer other than the device hosting it. This usage requires 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, DNo also supports notarization by a stranger 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 [7], an open question is if strangers can reliably notarize supplicants. Thus, 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 can fool a stranger into believing she is interacting through video with the legitimate device owner. Our study tested a fairly sophisticated attacker in this regard, namely one who has both a stolen device and a photograph of its owner, from which he creates an animated avatar for the owner to display to the notary. Participants also answered a series of questions that allowed us to evaluate their overall comfort with notarization.

To summarize, we make two contributions. First, we describe the design and implementation of DNo, which is novel in using notarization by video as a method for authenticating supplicants. We detail two use cases of DNo, one for decrypting passwords (for other services) stored locally on the device, and another to protect the establishment of client-side TLS connections. Moreover, DNo does not permit the notary to impersonate the supplicant or divulge sensitive information to the notary. Second, through a detailed user study, we shed light on the effectiveness of using strangers as notaries and users' 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 defending a stolen device.

II. Related Work

There have been several designs by which a person leverages others in his social network to gain access to resources, either to prevent someone who has stolen his device from doing so (as we do here) [8], [9] or to regain access after losing one or both of his authentication factors [10], [11]. 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 [5], [12], DNo can protect 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 enables a conceptually distinct usage model in which the device owner can choose to utilize DNo unilaterally. 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 [7], [13], [14]. 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. [15] 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 use them to help secure web authentication. To add an extra 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 [16], [17]. 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 [18], [19]. While DNo 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.

III. Design of DNo

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

A. Overview

Presently, DNo 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 [6].

1) Encrypted password use case: Our DNo-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 DNo application. The DNo 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 Sec. III-B) 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 Fig. 2), the DNo application establishes a video chat with the DNo 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 Sec. III-B) 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 [5], [12]. 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]

We assume the notary's device and the cloud service are not compromised, though they (provably) cannot impersonate the devices they notarize. Rather, their compromise can, at worst, reduce the supplicant's security to depend solely on device possession, i.e., one-factor authentication.

2) TLS use case: Using DNo to support client-side TLS is shown in Fig. 3. Rather than encrypted credentials, client-side TLS certificates are stored when these URLs are bookmarked in the DNo 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 Fig. 3. To digitally sign for the client in a TLS exchange, DNo 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. 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.

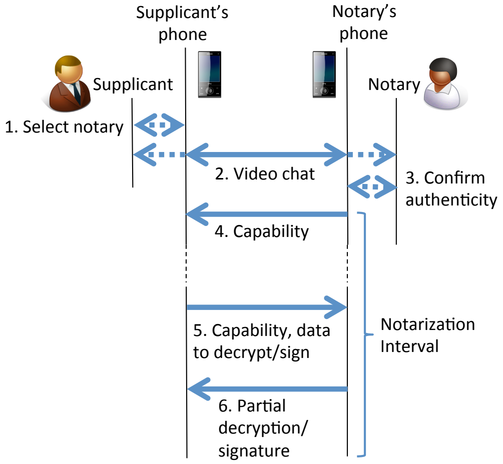


Fig. 2. Overview of notarization process and control. Steps 1–4 are executed, if necessary, between steps 1–2 of Fig. 3. Steps 5–6 are executed between steps 4–5 of Fig. 3.

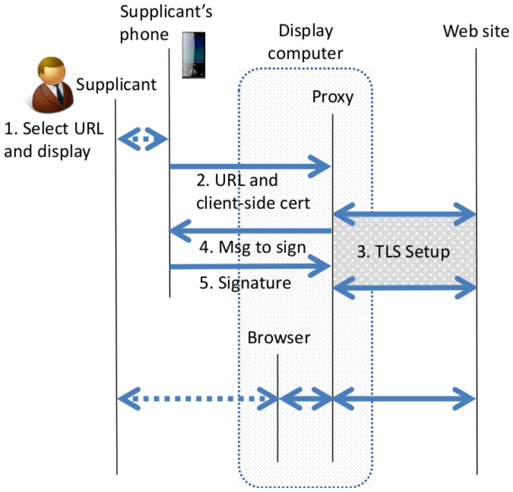


Fig. 3. Use of DNo to establish TLS connection.

After checking whether the user has recently been notarized, the DNo 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 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.

3) 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

[1] Alternative protocols exist that remove the need for an interaction per decryption (Steps 5–6 in Fig. 2), by reconstructing the private decryption key at the supplicant's device for the duration of the notarization interval [5]. 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. The device owner can destroy its authorization proactively (e.g., because he is done with his sensitive task) by simply deleting the capability to prevent an attacker who then captures the device from making use of the authorization.

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 DNo 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 Sec. III-A4.)

The role of the DNo 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 DNo service sends a capability to the supplicant's device. During the notarization interval for that capability, the DNo 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 [12]. The process of notarization in the Anyone case is thus very similar to that in Fig. 2, with the DNo service playing the role of the notary's phone.

4) Initialization: The process by which a device owner initializes his device for supporting notarization is summarized below.

URLs URLs requiring authorization can be added to the DNo 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 [12]. 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 DNo 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 DNo service can be shipped with the DNo 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 DNo 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 DNo application except to import public keys with which to authenticate notarization requests from allowed supplicants. 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 includes the proxy to which the DNo 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 displays the proxy's addressing information (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 2D barcode on the host screen, permitting the DNo application on the device to import this information by photographing it [20], [21].

Client-side certificates DNo supports the standard Certificate Signing Request (CSR) procedure [22] 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 DNo 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. Since most smartphones and similar devices include a camera, obtaining a suitable picture should rarely pose a difficulty.

5) Privacy: Involving another person (the notary) to notarize 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 DNo is used to protect a device's signing key for client-side TLS sessions or to decrypt a ciphertext 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 [23], then this ciphertext can reveal statistical information about what web site is being accessed. DNo 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.

B. Implementation and User Experience

Client software We have implemented DNo 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 DNo is implemented using libgcrypt.

To support using DNo for setting up TLS sessions, the display computer runs an adaptation of mitm-proxy, a Java SSL proxy that acts as a 'man in the middle.' We modified the OpenJDK SSL implementation to intervene in the handshake as required by our protocol. Using the Google Web Toolkit, 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 ZXing barcode library to handle QR code generation. We used the jWebSocket server to facilitate TLS-secured communication between the device and the proxy and between the proxy and the browser extension via our custom plugins.

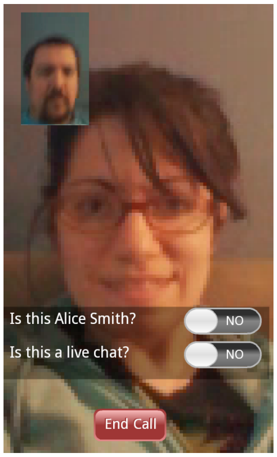


Fig. 4. Interface for named notary (vs. stranger). Video shows supplicant. The notarization request came from Alice's device, as indicated by the "Is this Alice Smith?" question.

Server software We have implemented the cloud service for initialization, delegation, 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 DNo to maintain their own application in the cloud, we envision this type of service being offered by a service provider.

User interface The common-case use of the DNo 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 Fig. 4 to the notary. The notary's interface asks him to respond to two questions during the video chat, specifically if this supplicant appears to be the correct device owner and if 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 Sec. IV.

The notary interface for use by a stranger, i.e., one contacted by way of the DNo cloud service (see Sec. III-A3), is similar to that pictured in Fig. 4, except that rather than asking "Is this Alice Smith?", 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 Sec. IV.) 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. We use a three-photo 'lineup' style interface for strangers who are notaries since studies indicate that lineups can improve performance in identification tasks [24], but it is not fundamental to our design.

IV. Efficacy of DNo

Recall that in addition to notarization by members of a device owner's social network, DNo also supports notarization by a stranger. To evaluate the effectiveness, feasibility, and usability of human-mediated crowdsourced authentication, we conducted a lab-based experiment. Specifically, we were interested in determining whether participants could be fooled by a sophisticated, customized video avatar. We chose to use a lab-based experiment instead of a field experiment to control factors such as participants' familiarity with one another. Our entire study was approved by UNC's Institutional Review Board (IRB).

A. Overview of the study

Our study comprised a set of lab sessions. For each session, participants were randomly assigned to act as either a supplicant or a notary. Notaries were always physically separated from supplicants so that they would not see 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. We used the chosen photo and the liveness confidence to determine the identification rate, which was one of our primary methods of system evaluation.

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. These avatars 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 identical to live chats, except that the notary (unbeknownst to him) 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.

B. Participants

In total, 62 people participated in the study. However, in one session our software malfunctioned and so our results are based on 56 participants (26 notaries and 30 supplicants). Participants were primarily younger (88% age 25 or under), Caucasian (53%) or African American (30%), female (73%) and had prior experience using video chat to have video conversations (see Table I and Fig. 6).

C. Procedure

Recruitment Study participants were recruited via flyers placed in several high-traffic areas on UNC's 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 showed up for a lab session.

Image set creation Using the images participants provided during recruitment, we created a customized photo set for each participant. Each set contained 1 photo of the participant and 2 photos of other participants. The 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.

Random assignment to groups There were two groups in our study: notaries and supplicants. Upon arriving at our lab site, participants were randomly assigned to either the supplicant or notary group. Notaries were sent to one lab space and supplicants were sent to another lab space to prevent participants from seeing one another.

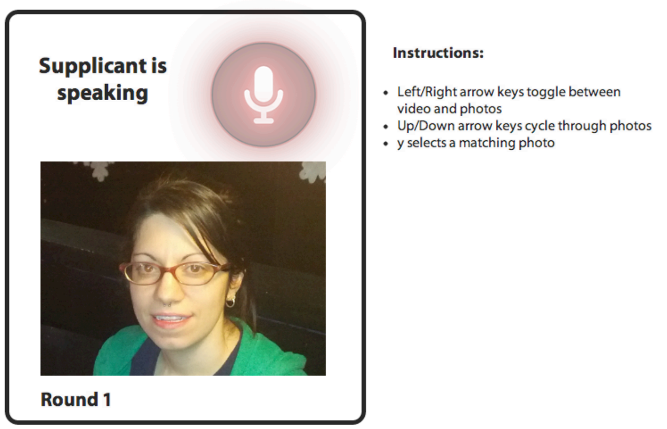
Minimizing extraneous participant interactions Participants were instructed to arrive in the lobby of our building where they would then be directed to the appropriate room by a member of our study team. To avoid accidental interaction between the groups, each participant was given a map with a highlighted 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 video-chatted with before that day. We collected this data to exclude any such chat pairings from our analysis in an effort to ensure that we were only looking at notarization between strangers. We excluded 2 out of 80 chats for this reason.

Experimental setup and participant instructions 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 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.

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. 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 ensured that each notary would see at most one avatar in a random round after the first, and that each supplicant would act in the avatar role at most once. 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.

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 Fig. 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 instructed to interact with the supplicant and compare their video feed to the provided photo set to verify the supplicant's identity and 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 discarded.

Evaluating identity and liveness Notaries were asked to evaluate the identity and liveness of their video chat partner. The specific assertions presented to the notary after he selected a photo that he believed to be the supplicant were: (i) "I am sure this photo matches the person in the video," and (ii) "I am sure this was a live conversation and not a recording."



Supplicant is speaking

Round 1

Instructions:

- Left/Right arrow keys toggle between video and photos
- Up/Down arrow keys cycle through photos
- y selects a matching photo

Fig. 5. The notary interface in the study described in Sec. IV. 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. The notary toggles between the live video and three photos of the same size as the video. The notary cycles through these photos using the up/down arrow keys. The notary indicates her identification of the supplicant by pressing "y" while the intended photo is displayed.

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 typical of video-chatting over mobile devices. For example, the size of the notary's video display was roughly 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.

Demographics and other participant characteristics At the end of each lab session, participants filled out a brief questionnaire that asked them to indicate their gender, race, and age; see Table I. We also asked the participants how often they have video conversations; see Fig. 6.

Avatar creation and use during experiment We used SitePal's 3D Photoface technology (www.sitepal.com) to produce realistic, lifelike avatars for the participants to interact with. The avatars were 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. 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.

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 (www.red5.org) 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.

TABLE I. 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%)

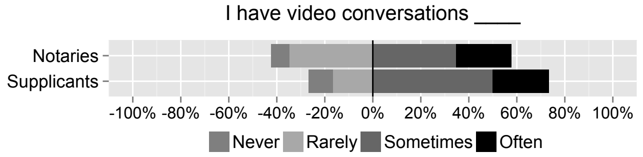


Fig. 6. Responses to the statement "I have video conversations ____".

D. Results

Chat length We measured the length of each chat in seconds; the distribution is shown in Fig. 7. Overall, chats lasted between 1 and 340 seconds. Chats with avatars lasted longer (mean of 100.6 seconds) than chats with real people (mean of 78.2 seconds), perhaps indicating notaries' further investigation of avatar cases, sensing something unusual. In both cases, however, it generally did not take long for the notary to draw her conclusion about the supplicant.

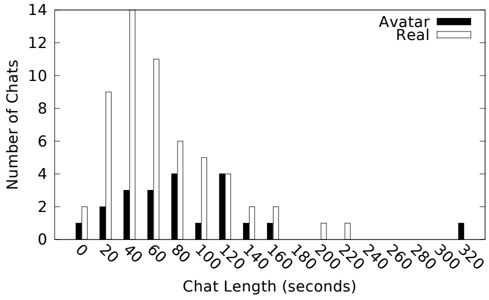


Fig. 7. Histogram of chat lengths, binned into 20-second intervals.

Notarization analysis Recall that identification and liveness confidences were queried from notaries separately. To analyze the notarization results, we treated notaries' ordinal responses on the Likert-type scale for identification and liveness confidences numerically: “Strongly disagree” → -2, “Disagree” → -1, “Neutral” → 0, “Agree” → 1, and “Strongly agree” → 2. The mean identification confidence was 1.6 for real supplicants and 1.1 for avatars, and the mean liveness confidence was 1.5 for real supplicants and -0.8 for avatars. That is, while the identification confidence scores were roughly in agreement in the real and avatar cases—which is expected since in both cases, a photo of the person shown in the video was included in the options available to the notary—notaries' confidences for the liveness of the avatars was typically less than for real supplicants.

In practice, the two confidence scores input by the notary would need to be combined to determine whether the notary's responses indicated sufficient confidence to declare the supplicant notarized. Thus, we defined the combined notarization score to be the minimum of his photo confidence and liveness confidence. For example, if a notary reports that she “Strongly agrees” that this photo is of the person she chatted with, and she “Agrees” it was a live chat, the notarization score would be min(2,1) = 1. 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) then is 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.

We created a ROC curve by setting the threshold (t) equal to each possible notarization score (described above) in the range $[-2, 2]$ and then calculating the true and false identification rates that would result. For example, if we look at $t = 2$ and the above example where “Strongly Agree” and “Agree” were selected (and the photo choice really is correct), then $\min(2, 1) = 1 < t$, and so this trial would not count as a true identification. However, if the threshold had been 1, then this would have counted as a true identification.

The One Notary ROC curve in Fig. 8 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 $1-\text{TIR} \approx \text{FIR} \approx 24\%$. We should note that at $t = 1$, two notaries accounted for two false identifications and only two true identifications across their seven notary-supplicant chats, leading us to believe that they were not really trying.

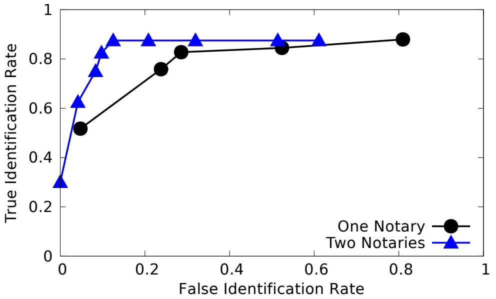


Fig. 8. ROC curve showing true and false identification rates in the user study described in Sec. IV.

We also show a Two Notaries ROC curve in Fig. 8 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 Fig. 8 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: qualitative results We did not give participants any insight into the specific form of attack that our study attempted. Thus, notaries were not aware that some of the supplicants they were interacting with were not humans but rather avatars with live human audio overlaid on a manufactured video. In addition to understanding the quantitative confidence notaries expressed in liveness determination, we were also interested in understanding how notaries would determine that they were speaking with a real person (i.e., that the supplicant was live and present), so we asked participants in a post-study questionnaire: “What did you do to ensure that a live supplicant was present?”

Notaries adopted a variety of strategies for determining liveness. About half of the 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”

On the other hand, the other half of participants adopted strategies that would be ineffective against an avatar attack. For example, one of the least effective 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.”

Note that we did not give participants any insight or instructions that would have helped them determine liveness. It is likely that with short instructions and/or training notaries would be able to dramatically improve their liveness identification rate. For example, if we instructed notaries to use strategies that involved testing both the liveness of the audio and video, we would anticipate better performance.

Comfort with chat In a questionnaire at the end of their lab sessions, most participants indicated that they were comfortable interacting through video chat. Fig. 9 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. A large majority of supplicants indicated that they were comfortable seeking identification from another person through video chat (see Fig. 10). When asked to rate identification through video, both notaries and supplicants were very positive (see Fig. 11).

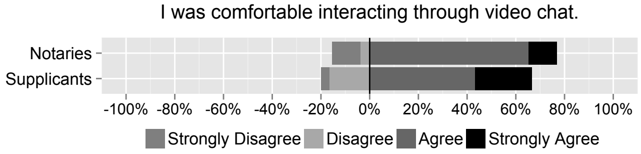


Fig. 9. Non-neutral responses to the statement “I was comfortable interacting through video chat.”

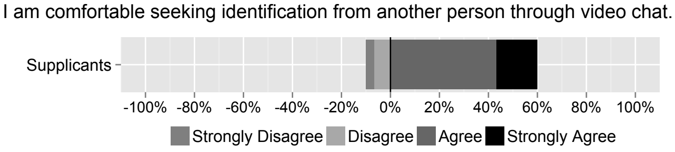


Fig. 10. Non-neutral responses to the statement “I am comfortable seeking identification from another person through video chat”.

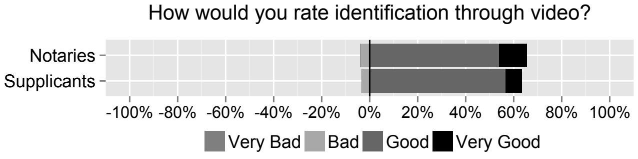


Fig. 11. Non-neutral responses to the question “How would you rate identification through video?”.

We also collected qualitative data about user comfort. Participants indicated awkwardness in some cases, though this seemed to differ somewhat between notaries and supplicants; notaries were more comfortable than 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.”

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 [25]–[27]. When using strangers as notaries, it may thus be advisable to utilize strategies that maximize supplicant and notary similarity.

E. Implications

Our study suggests that using a stranger as a notary is a feasible option for authentication as a last line of defense. In a short period of time, strangers were able to comfortably and fairly accurately identify a person and determine whether they were live. We also discovered that careful thought should be given to selecting an appropriate confidence threshold for identification accuracy and liveness. 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; recall that as tested, the FIR represents a very difficult case for DNo, namely an attacker who steals a device and uses a photograph of the owner and automated tools to create a life-like avatar for the owner.

Our results also suggest that using two notaries would yield better results than one (at a cost to convenience). Other improvements suggested by our study include utilizing more photos per supplicant and utilizing notaries of the same race as supplicants. To prevent the attacker from trying strangers repeatedly until one assents, the DNo service could suspend the device after too many notary rejections.

Another takeaway message is that while several notaries intuited effective measures to test the liveness of both the audio and video, some did not. It is likely that providing training to the notary regarding methods to test the liveness of both video and audio would improve recognition accuracy.

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 DNo to be accepted by users.

F. 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 the crowd 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.

V. Conclusion

We have introduced the concept of 'notarization', in which a remote entity (the notary) must verify via video chat who is in physical possession of a mobile device for the device to make use of its cryptographic credentials. We implemented DNo, an Android application using notarization to protect cryptographic keys used for decrypting on-device data or signing in support of client-side TLS. Since DNo 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.

Acknowledgments

This work was supported in part by NSF grants 1228364 and 1228471.

References

[1] A. Smith, "Smartphone ownership 2013," Pew Research Center, Tech. Rep., 2013.

[2] "The World in 2014: ICT facts and figures," http://www.itu.int/en/ITU-D/Statistics/Documents/facts/ICTFactsFigures2014-e.pdf.

[3] "Hackers, IT units focusing on smartphone security," http://www.reuters.com/article/2011/12/30/us-mobile-security-idUSTRE7BT0GV20111230.

[4] K. Komando, "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.

[5] P. MacKenzie and M. K. Reiter, "Networked cryptographic devices resilient to capture," *Int. J. Information Security*, vol. 2, no. 1, 2003.

[6] T. Dirks and E. Rescorla, "The transport layer security (TLS) protocol, version 1.2," IETC RFC 5246, 2008.

[7] V. Bruce, Z. Henderson, C. Newman, and A. Burton, "Matching identities of familiar and unfamiliar faces caught on CCTV images," *J. Exp. Psychol.-Appl.*, vol. 7, 2001.

[8] B. Soleymani and M. Maheswaran, "Social authentication protocol for mobile phones," in *2009 International Conference on Computational Science and Engineering*, Aug. 2009.

[9] J. Zhan and X. Fang, "Authentication using multi-level social networks," in *Knowledge Discovery, Knowlege Engineering and Knowledge Management, First International Joint Conference*, Oct. 2009.

[10] J. Brainard, A. Juels, R. Rivest, M. Szydlo, and M. Yung, "Fourth factor authentication: Somebody you know," in *13th ACM Conference on Computer and Communications Security*, 2006.

[11] S. Schechter, S. Egelman, and R. Reeder, "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.

[12] P. MacKenzie and M. K. Reiter, "Delegation of cryptographic servers for capture-resilient devices," *Distrib. Comput.*, vol. 16, no. 4, 2003.

[13] A. M. Burton, S. Wilson, M. Cowan, and V. Bruce, "Face recognition in poor-quality video: Evidence from security surveillance." *Psychol. Sci.*, vol. 10, no. 3, 1999.

[14] V. Bruce, Z. Henderson, C. Newman, and A. M. Burton, "Verification of face identities from images captured on video," *J. Exp. Psychol.-Appl.*, vol. 5, 1999.

[15] G. E. Pike, R. I. Kemp, N. A. Towell, and K. C. Phillips, "Recognizing moving faces: The relative contribution of motion and perspective view information," *Vis. Cogn.*, vol. 4, no. 4, 1997. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/713756769

[16] "Bank of America SafePass," http://www.bankofamerica.com/privacy/index.cfm?template=learn_about_safepass.

[17] "Google 2-step verification," https://support.google.com/accounts/bin/topic.py?hl=en&topic=28786&parent=2373945&ctx=topic.

[18] R. C. Jammalamadaka, T. W. van der Horst, S. Mehrotra, K. E. Seamons, and N. Venkasubramanian, "Delegate: A proxy based architecture for secure website access from an untrusted machine," in *22nd Annual Computer Security Applications Conference*, 2006.

[19] M. Wu, S. Garfinkel, and R. Miller, "Secure web authentication with mobile phones," in *DIMACS Workshop on Usable Privacy and Security Software*, 2004.

[20] J. M. McCune, A. Perrig, and M. K. Reiter, "Seeing-Is-Believing: Using camera-phones for human-verifiable authentication," *Int. J. Security and Networks*, vol. 4, no. 1–2, 2009.

[21] L. Bauer, S. Garriss, J. M. McCune, M. K. Reiter, J. Rouse, and P. Rutenbar, "Device-enabled authorization in the Grey system," in *Information Security: 8th International Conference, ISC*, 2005.

[22] "PKCS #10: Certification request syntax standard," http://www.rsa.com/rsalabs/node.asp?id=2132.

[23] M. Bellare, A. Boldyreva, A. Desai, and D. Pointcheval, "Key-privacy in public-key encryption," in *Advances in Cryptology – Asiacrypt 2001*, 2001.

[24] A. D. Yarmey, A. L. Yarmey, and M. J. Yarmey, "Face and voice identifications in showups and lineups," *Appl. Cognitive Psych.*, vol. 8, no. 5, Oct. 1994.

[25] T. Valentine and M. Endo, "Towards and examplar model of face processing: The effects of race and distinctiveness," *Q. J. Exp. Psychol. A*, vol. 44, no. 4, 1992.

[26] D. Levin, "Race as a visual feature: Using visual search and perceptual discrimination tasks to understand face categories and the cross race recognition deficit," *J. Exp. Psychol. Gen.*, vol. 129, no. 4, 2000.

[27] P. Walker and W. Tanaka, "An encoding advantage for own-race versus other-race faces," *Perception*, vol. 23, 2003.