Seeing-Is-Believing: using camera phones for human-verifiable authentication

Jonathan M. McCune* and Adrian Perrig

CyLab, Electrical and Computer Engineering Department,
Carnegie Mellon University, Pittsburgh, PA, USA
E-mail: jonmccune@cmu.edu
E-mail: perrig@cmu.edu
*Corresponding author

Michael K. Reiter

Department of Computer Science,
University of North Carolina, Chapel Hill, NC, USA
E-mail: reiter@cs.unc.edu

Abstract: Current mechanisms for authenticating communication between devices that share no prior context are inconvenient for ordinary users, without the assistance of a trusted authority. We present and analyse Seeing-Is-Believing (SiB), a system that utilises 2D barcodes and camera-phones to implement a visual channel for authentication and demonstrative identification of devices. We apply this visual channel to several problems in computer security, including authenticated key exchange between devices that share no prior context, establishment of the identity of a TCG-compliant computing platform, and secure device configuration in the context of a smart home.

Keywords: device pairing; key establishment; 2D barcodes; MITM; man-in-the-middle attack; camera phones; wireless networks.

Reference to this paper should be made as follows: McCune, J.M., Perrig, A. and Reiter, M.K. (2009) 'Seeing-Is-Believing: using camera phones for human-verifiable authentication', *Int. J. Security and Networks*, Vol. 4, Nos. 1/2, pp.43–56.

Biographical notes: Jonathan M. McCune is a PhD candidate in the Electrical and Computer Engineering Department at Carnegie Mellon University. He earned an MS Degree from Carnegie Mellon University in 2005, and a BS Degree from the University of Virginia in 2003. His research covers both low-level system security and authentication in ad hoc networks. Other projects include VMM detection, distributed mandatory access control, access control with smart phones, and sensor network security.

Adrian Perrig is an Associate Professor in Electrical and Computer Engineering, Engineering and Public Policy, and Computer Science at Carnegie Mellon University. He earned the PhD Degree in Computer Science in 2002 from Carnegie Mellon University, and spent three years during his PhD Degree at the University of California, Berkeley. He received the MS Degree in Computer Science in 1999 from Carnegie Mellon University and the BSc Degree in Computer Engineering in 1997 from the Swiss Federal Institute of Technology in Lausanne (EPFL). His research interests revolve around building secure systems and include internet security, security for sensor networks and mobile applications.

Michael K. Reiter received the BS Degree in Mathematical Sciences from the University of North Carolina, Chapel Hill (UNC-CH) in 1989 and the MS and PhD Degrees in Computer Science from Cornell University in 1991 and 1993, respectively. He is currently the Lawrence M. Slifkin Distinguished Professor in the Department of Computer Science, UNC-CH. He has held technical leadership positions in both the industry and academe. From 1998 to 2001, he was the Director of Secure Systems Research, Bell Laboratories. From 2001 to 2007, he was a Professor and the Technical Director of CyLab at Carnegie Mellon University. His research interests include computer and communications security and distributed computing.

1 Introduction

Users often wish to configure two devices to communicate over a secret and authentic channel, e.g., to exchange sensitive documents or personal messages. This is attainable via a dedicated physical connection such as a cable. However, today's devices increasingly feature convenient, wireless communication interfaces (e.g., 802.11, Bluetooth, and WiMax). Unfortunately, wireless communication is invisible to humans, rendering it vulnerable to Man-In-The-Middle (MITM) attacks. A MITM attack takes place when Alice and Bob believe they are communicating with each other, when in fact they are both communicating with Charlie, who is able to monitor, modify, inject, suppress, or otherwise tamper with Alice and Bob's intended communication without their knowledge.

An out-of-band communication channel that provides authenticity suffices to defeat MITM attacks. Protocols based on public-key cryptography, such as Diffie and Hellman (1976), can bootstrap secret and authentic communication given authentic public keys. The challenge, then, is to construct an out-of-band channel that provides authenticity for the exchange of public keys using the interfaces present on current devices. Balfanz et al. (2002) refer to such a mechanism as providing *demonstrative identification* of the communication devices. We approach this problem with the premise that, in many situations, a user can *visually* identify the desired device.

We propose to use the camera on a mobile phone as a new *visual channel* to achieve demonstrative identification of communicating devices formerly unattainable with wireless communication. We term this approach Seeing-Is-Believing (SiB). In SiB, one device uses its camera to take a snapshot of a barcode encoding cryptographic material identifying, e.g., the public key of another device. We term this a *visual channel*. Barcodes can be pre-configured and printed on labels attached to devices, or they can be generated on-demand and shown on a device's display.

As camera-equipped mobile phones rapidly approach ubiquity, these devices become an excellent platform for security applications that can be deployed to millions of users. Today's mobile phones increasingly feature internet access, cameras, high-quality displays, and short-range Bluetooth wireless radios. They can perform public-key cryptographic operations in under one second.

We apply this visual channel to several problems in computer security. SiB can be used to bootstrap authenticated key exchange between devices that share no prior context, including such devices as mobile phones, wireless access points, and public printers. We also use SiB to aid in the establishment of the identity of a TCG-compliant[1] computing platform, and to secure device configuration in the context of a smart home.

1.1 Outline

We survey related work in Section 2 and provide an overview of SiB in Section 3. Section 4 presents the use of SiB for authenticated key exchange between mobile devices. Section 5 explains how to use SiB to achieve demonstrative identification of, and secure connection to, a particular wireless device in a unidirectional authentication scenario. We also show in Section 6 how this technology can be used to achieve slightly weaker – but still quite valuable – security properties in the context of, e.g., a smart home. Our implementation is detailed in Section 7, and we present applications of SiB in Section 8. We offer a security analysis in Section 9 and state our conclusions in Section 10.

2 Related work

SiB is closely related to work on authentication involving mobile devices, and barcode scanning with camera phones.

2.1 Authentication

In this section, we study authentication between two co-located entities with no prior trust relationships. This context rules out the use of a public key infrastructure or trusted third party to perform authentication.

A common mechanism to establish a secure channel between two entities is to use Diffie-Hellman key establishment (Diffie and Hellman, 1976). Unfortunately, a MITM attack is possible if the two entities do not share any established trusted information. Bellovin and Merrit propose the Encrypted Key Exchange (EKE) protocol, which prevents the MITM attack if both parties share a secret password (Bellovin and Merrit, 1993). Several researchers have refined this approach (Bellovin and Merrit, 1992; Boyko et al., 2000; MacKenzie et al., 2000; Wu, 1999), but they all require a shared secret password between the two entities, which may be cumbersome to establish in many mobile settings.

Another approach to defeat the MITM attack is to use a secondary channel to verify that the same key is shared by two parties. An approach that several researchers have considered is that a human can manually verify that the generated keys are identical (Čagalj et al., 2006; Laur and Nyberg, 2006; Vaudenay, 2005). Uzun et al. (2007) found usability issues with general classes of string comparison-based protocols. To avoid manual comparison, researchers have devised visual metaphours that represent the hash of a key to make it easier for people to perform the comparison (Dohrmann and Ellison, 2002; Goldberg, 1996; Levien, 1996; Perrig and Song, 1999). Though these schemes make key comparison easier for the user, they still rely on the user to diligently compare the resulting visual key representations. With SiB, visual device identification is an integral part of establishing a connection between devices, though in a far less overt way.

To defend against MITM attacks, Stajano and Anderson propose to set up keys through a link that is created through physical contact (Stajano and Anderson, 1999). However, in many settings, devices may not have interfaces that connect for this purpose, or they may be too bulky to carry around. Balfanz et al. (2002) extend this approach to use short-range wireless infrared communication. Of all these approaches, theirs is the most closely related to SiB, and we discuss it further in Section 3.2. Čapkun et al. (2003) have further extended this research direction. They make use of one-hop transitive trust to enable two nodes that have never met to establish a key. SiB could leverage this technique equally well.

Following the initial publication of SiB (McCune et al., 2004, 2005), Saxena et al. (2006) further explored the visual channel. They consider the minimal device capabilities that can support SiB, and devise a video codec that can use a mobile phone's camera to decode data encoded in a severely constrained visual channel – in the limit a single flashing LED. This scheme is valuable in low-cost scenarios where the only output mechanism available may be an LED, though it requires a higher level of understanding from the user.

Also following our work, Goodrich et al. (2006) developed Loud-and-Clear, a system that uses an audio channel to establish authentic keys. In Loud-and-Clear, English phrases are derived from the hash of a device's public key. One device uses a text-to-speech engine to read a phrase aloud, while the other device displays a phrase on-screen. The human user is tasked with listening to one phrase and comparing it with the written phrase.

In some cases, the visual channel bandwidth available between two devices may be insufficient for standard cryptographic techniques. For example, a single barcode in our implementation has a data payload of only 68 bits. To address issues with low-bandwidth channels, Laur and Nyberg (2006) propose protocols based on *Manually Authenticated Strings* (MANA) conveyed across an out-of-band channel that may have low bandwidth. In their case, the low bandwidth channel is that of humans performing a manual comparison. MANA IV requires users to visually compare short ℓ-bit strings displayed on their devices and push a button on each device to indicate whether the strings match, where ℓ is presumed to be shorter than the output of a cryptographic hash function. Given SHA-1 as an acceptable hash function, $\ell < 160$.

SiB can be modified to use MANA-IV as the commitment protocol when the only available visual channel is exceptionally low-bandwidth. The visual channel is used to convey the ℓ-bit strings between devices, where they can be programmatically compared. This design is compelling because it removes the users' responsibility to carefully compare the ℓ-bit strings, thereby substantially reducing the opportunity for human error (Uzun et al., 2007). However, MANA IV requires the devices to exchange three messages before the visual channel exchange takes place. This necessitates an out-of-band mechanism for discovering the network identity of the other device. Traditional Bluetooth discovery mechanisms are one option in this scenario.

2.2 Barcode recognition with camera phones

SiB depends on a camera phone having the ability to use its camera to recognise two-dimensional (2D) barcodes. Several projects exist that seek to allow camera-equipped mobile phones to interact with physical objects through the use of 2D barcodes. Rohs and Gfeller (2004) develop their own 2D code explicitly for use with mobile phones, emphasising their ability to be read from electronic screens and printed paper. Woodside develops *semacodes*,[2] which is an implementation of the Data Matrix barcode standard for mobile phones (ISO/IEC, 2006). Woodside considers the primary application of semacodes as containers for a URL which contains information about the physical location where the barcode was installed. Madhavapeddy et al. (2004) use SpotCodes to enhance human-computer interaction by using a camera-phone as a pointing and selection device. Researchers working on the CoolTown[3] project at HP Labs propose tagging electronics around the house with barcodes to be read by camera phones or PDAs so that additional data about the tagged device can be easily retrieved.

Hanna (2002) considers devices with barcodes affixed to aid in the establishment of security parameters. His work considers a smart home, where a user may want to establish a security context for controlling appliances or other devices in a smart-home. In Hanna's work, the barcode contains a secret which is also stored inside the device. Hanna proposes using this secret to enable the secure transmission of commands to the device from a master controller over an untrusted network. We refer to the security property discussed by Hanna as *presence*, where it is desirable that only users or devices close to some device are able to control it. We discuss the notion of *presence* further in Section 6.

Today, recognition of 2D barcodes with mobile phones has become accepted practice. Phones are now available that include barcode recognition software, such as the QR Code reader on the Nokia N95.[4] Further, a Java standard has been published that specifies an API for barcode recognition on mobile phones (JSR-257, 2006).

3 Seeing-Is-Believing (SiB)

With SiB, a mobile phone's integrated camera serves as a visual channel to provide demonstrative identification of the communicating devices to the user while also providing an out-of-band mechanism for exchanging authentic information. By *demonstrative identification*, we mean the property that the user is sure her device is communicating with *that* other device. In SiB, the user identifies *that* other device visually. This serves to strongly authenticate data from the other device since the user knows precisely which devices are communicating. Thus, SiB can be used to bootstrap authentic and secret communication, thereby defeating MITM attacks while allowing the use of convenient wireless communication. SiB also captures user intentions in an intuitive way.

What better way for a user to tell device A that it should communicate securely with device B than to take a picture of device B using device A's integrated camera?

In the remainder of this section, we detail the physical realisation of the visual channel with 2D barcodes. The use of the visual channel to bootstrap secure communication is then illustrated with a specific example. We end this section with a discussion on using SiB with devices that may be lacking a display or a camera, or both. Sections 4 and 5 then provide detailed usage scenarios for the demonstrative identification provided by SiB. In Section 6, we move on to discuss a weaker – though still valuable – property that can be provided by the visual channel, which we term *presence*.

3.1 2D barcodes as a visual channel

We implement the visual channel with a 2D barcode (e.g., Data Matrix ISO/IEC, 2006), displayed by or affixed to one device and captured by another with its digital camera. When a user executes the SiB protocol, she must aim the camera of her mobile device at a barcode on another device (either displayed electronically or affixed to the device's housing). The act of aiming the camera at the desired device results in demonstrative identification of the targeted device. We say that the device displaying the barcode is in *Show* mode, and that the device whose camera is active is in *Find* mode.

We now present a more detailed example of the use of SiB. Suppose Alice and Bob want to set up a secure channel between their camera phones. Alice's phone generates a 2D barcode encoding appropriate public cryptographic material and *Shows* it on its screen, while Bob uses his phone's digital camera in *Find* mode to take a snapshot of Alice's screen displaying the barcode. Bob must watch his phone's LCD, acting as viewfinder, updating in real time in response to his positioning of his camera-phone. A barcode recognition algorithm processes each image in the viewfinder in real time and overlays a coloured rectangle around recognised barcodes. Once a barcode is successfully recognised, the view-finding process stops and the barcode recognition and error-correcting algorithms return the data represented by the barcode. Section 7 presents further details of our implementation.

3.2 Pre-authentication and the visual channel

We build on work by Balfanz et al. (2002), and Stajano and Anderson (1999), to secure wireless communication by leveraging an out-of-band channel for authentication. Our out-of-band channel is the visual channel. We adopt the term pre-authentication, as Balfanz et al. (2002) suggest to describe the authentic data exchanged on the visual channel. Pre-authentication data is later used to authenticate one or both of the communicating parties in almost any standard public-key communication protocol over the wireless link. Eavesdropping on the visual channel gives no advantage to an attacker, provided that the underlying cryptographic primitives are secure, and that the mobile devices themselves have not been compromised.

Balfanz et al. (2002) discuss the use of infrared communication as a 'secure side-channel' for pre-authentication between mobile devices. They focus on the property that infrared is a 'location-limited channel', emphasising the difficulty an attacker faces in trying to interfere with the channel, because he must be in close physical proximity to the communicating devices. The primary advantage of SiB is that it uses a visual channel instead of an invisible channel, thus adding a direct human factor. We acknowledge that attacks against infrared are difficult to perform, but we believe that the inability of the user to actually see which devices are communicating provides dangerous opportunities to an attacker.

Figure 1 shows the pre-authentication phase of SiB, carried out over the visual channel. Device A *Shows* its public key by displaying a hash of its key as a barcode: $h_A \leftarrow hash(K_A)$. The user of device B then aims her device's camera at the display of device A, causing software (in *Find* mode) on device B to process the barcode from device A's display: $A \overset{\text{visual}}{\longrightarrow} B : h_A$. At this point, device B has an authentic copy of the hash of A's public key. We say that this hash was conveyed via the visual channel. Device A can then send A's full public key to device B via the untrusted wireless connection: $A \overset{\text{wireless}}{\longrightarrow} B : K_A$. After receiving K_A via the untrusted wireless connection, software on device B can recompute the hash of K_A ($h' \leftarrow hash(K_A)$) and compare the computed hash with the hash received via the visual channel: $h' \overset{?}{=} h_A$. If there is any discrepancy, device B aborts.

Provided that the mobile phone has not been compromised, and that the visual channel and relevant cryptographic primitives are secure against active adversaries (Section 9 presents a detailed security analysis), authentication in SiB requires merely that the user confirm her camera is pointed at the intended device.

3.3 Device configurations

The concepts of SiB can be applied in different ways to devices with different capabilities, each equipped with either a camera and display, a camera only, a display only, or neither. In some cases, these device configurations impose some limitations on the strength of the achievable security properties. Figure 2 summarises these properties.

The most flexible configuration for SiB is when both devices have both a camera and a display – these have a CD in their column or row heading in Figure 2. These devices can be mutually authenticated, since both possess cameras. Further, each device can make use of either a long-term public key or an ephemeral public key in each exchange, since barcodes containing keys are displayed on an electronic screen (as opposed to paper or some other fixed medium).

We refer to devices equipped with no display – devices without a D in their column or row heading in Figure 2 – as 'displayless' devices. These devices can be authenticated with a long-term public key. A barcode encoding a commitment to the key, or multiple barcodes encoding the key itself, must be affixed to the device's housing (e.g., in the form of a sticker). The issue of whether to use a commitment to a key, or the key itself, is addressed in Section 5.

Entries in Figure 2 marked *presence* indicate that demonstrative identification of communicating devices is unattainable, but a property we term *presence* is still achievable. Presence refers to the ability to demonstrate that a device is in view of someone. We describe this property in more detail in Section 6.

Figure 1 Pre-authentication over the visual channel. K_A is A's public key, which can be either long-term or ephemeral, depending on the protocol

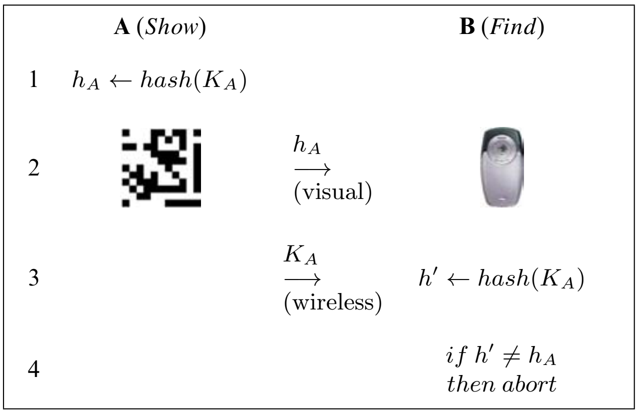

A (*Show*)
B (*Find*)
1 $h_A \leftarrow hash(K_A)$
2 $\xrightarrow{h_A}$ (visual)
3 $\xrightarrow{K_A}$ (wireless) $h' \leftarrow hash(K_A)$
4 $if\ h' \neq h_A$ $then\ abort$

Figure 2 Can a device of type X authenticate a device of type Y? We consider devices with cameras and displays (CD), cameras only (C), displays only (D), and neither (N)

		Y			
		CD	*C*	*D*	*N*
X	**CD**	✓	✓*	✓	✓*
	C	✓	✓*	✓	✓*
	D	presence	presence	×	×
	N	×	×	×	×

	Legend
✓	Strong authentication possible
✓*	Barcode label required on housing
presence	Confirm presence only
×	No authentication possible

4 Bidirectional authentication

Providing mutual authentication between mobile devices that share no prior context is a difficult problem. In this section, we show how SiB can be used to intuitively capture user intentions and establish a mutually authenticated security context between precisely the devices the user wants, without a trusted authority. Examples of the established security context include authenticated exchange of public keys, and an authenticated Diffie and Hellman (1976) key exchange to establish a shared secret. The device combinations we consider in this section are those where both devices have cameras.

We now walk through the use of SiB, beginning with device discovery and barcode generation. Next, we describe pre-authentication and bootstrapping a well-known public key protocol. Then, we describe options to satisfy different security requirements and project the likely performance of SiB on emerging mobile phones.

The SiB protocol begins when Alice and Bob decide they want to communicate securely. They must decide upon whose device will *Show* initially, and whose device will *Find*. The *Showing* device computes a commitment to its public key material and generates a barcode encoding this commitment, and any necessary network information to establish a wireless connection. The key material can take the form of a user's long-term public key, or it can be an ephemeral key for use in only one key exchange. One practical example of this key material is a self-signed public key certificate extended with additional information about the key owner (e.g., name, email address, etc., similar to a vCard Dawson and Howes, 1998; Howes and Smith, 1998). The decision regarding what form of public key material to use is orthogonal to the authentication provided by SiB.

The pre-authentication phase now begins. The users take turns *Showing* and *Finding* (displaying and taking snapshots of) their respective barcodes. The order is not important, but it is necessary that Alice's device capture the barcode commitment to Bob's public key, and that Bob's device capture the barcode commitment to Alice's public key. This pre-authentication protocol is secure as long as an attacker cannot find a second preimage for the commitment function, and is unable to perform an active attack on the visual channel.

After pre-authentication is complete, both devices now hold commitments to the other device's public key, and the devices can exchange public keys over the wireless link. The devices then perform the same commitment function over the other device's public key, ensuring that the result matches the commitment that was received over the visual channel. At this point, the devices have mutually authenticated one another's public keys, and Alice and Bob achieve demonstrative identification that the devices in their hands are the ones that are communicating. These authenticated public keys can then be used appropriately in any well-known public-key protocol on the wireless link (e.g., Diffie and Hellman, 1976, signed email, IKE Harkins and Carrel, 1998, SSL/TLS Dierks and Rescorla, 2006). It is imperative that in the chosen protocol, each party verifies that the other does in fact hold the private key corresponding to its authenticated public key.

A user may desire to protect their privacy by avoiding transmission of their public key on the wireless network. For example, public key transmission may allow eavesdroppers to ascertain which devices are communicating. The user's public key can be encoded in a barcode directly, or in a sequence of barcodes if a single barcode has insufficient data capacity. The key is thereby obtained by the other device without transmitting it on the wireless medium, while retaining the demonstrative identification property with respect to the device originating the key. It is then advisable that the public key protocol that is used with SiB authentication is key-private (Bellare et al., 2001).

As the processing and display capabilities of mobile phones improve, visual channel bandwidth will improve sufficiently for data transmitted over the visual channel to include network addresses for the relevant wireless interfaces (e.g., Bluetooth, 802.11) in addition to authentication data. This is more convenient for the user, since she never has to wait for discovery of neighbouring devices or select a device from a list. Madhavapeddy et al. (2005) use barcodes on camera phones to speed up the Bluetooth device discovery process in this way.

5 Unidirectional authentication

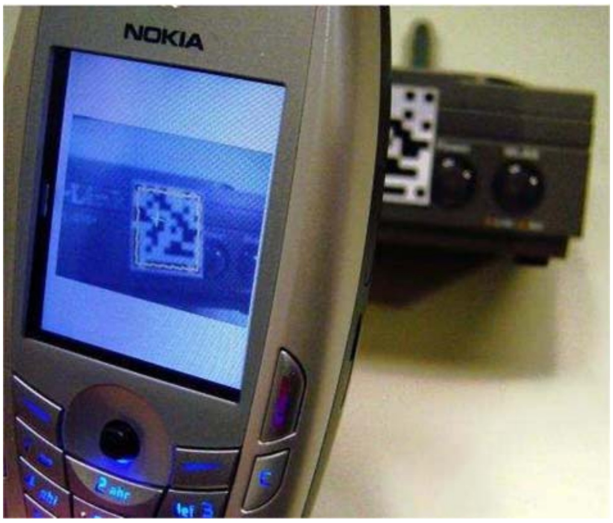
We now discuss entries from Figure 2 where the device of type X (the authenticator) is equipped with a camera, and the device of type Y (the device being authenticated) lacks a display and a camera. It is this presence of a camera on the authenticator, and lack of a display and a camera on the device being authenticated, that are responsible for the security properties of this particular device combination. We refer to a device of type X as camera-equipped, and a device of type Y as displayless.

Displayless devices do not have the ability to display newly generated values. Still, a camera-equipped device can authenticate displayless devices and establish secure communication channels. The displayless device must be equipped with a long-term public/private keypair, and a sticker containing a barcode of a commitment to its public key must be affixed to its housing. Since the displayless device is constrained to the use of a single public/private key pair for its entire lifetime, the option to generate per-interaction public keys no longer applies. Of course, devices can be reprogrammed and new stickers affixed, but we consider this to be a significant maintenance task. As in Section 4, there are privacy issues with using fixed public keys that might be of concern.

An 802.11 Access Point (AP) is one example of a class of devices where 'sticker-based' authentication may be desirable. Camera-enabled devices can authenticate the AP, enabling the establishment of a secure link-level connection between the camera-enabled device and the AP. This solution also enables deployment of wireless connectivity in environments where security policies require physical presence for network access. Figure 3 shows the SiB application on a mobile phone scanning a barcode installed on a wireless access point.

Another application where demonstrative identification of communicating devices is desirable is when using a printer in a public place. Similar to the wireless access point, the printer can have a barcode affixed to its housing so that a user can use SiB to authenticate wireless communication with the printer or print server and bootstrap the establishment of a secure connection. Secure communication is important here not only to ensure the secrecy of the printed document, but to prevent a MITM attack used to inject malicious software onto the user's computer by masquerading as a printer driver.

Figure 3 Phone running SiB scanning a barcode on an 802.11 access point (see online version for colours)



6 Presence confirmation

A display-only device (display-equipped and cameraless) is unable to strongly authenticate other devices using SiB. Equipped with no camera, it makes no difference whether the entity the cameraless device wants to authenticate has a display, or makes use of a barcode sticker – the cameraless device cannot 'see' them. However, display-only devices can obtain a property we refer to as *presence* (Figure 2). That is, it can confirm the presence of some other device in line-of-sight with its display.

To detect the presence of a nearby device, the display-only device generates a key K for a Message Authentication Code (MAC), encodes it in a barcode, and displays that barcode, noting the time when it was first displayed. Any nearby devices that are able to see the display and capture the barcode can send data to the display along with a MAC computed over that data: $\{data, MAC(K, data)\} \rightarrow display\text{-}only\ device$.

When the data and MAC arrive over the wireless channel, the display-only device knows that some device has been in line-of-sight during the time since K was first displayed. We emphasise that this *presence* property is quite weak – the display-only device has no way of knowing how many devices can see its display, or whether the radio signal is from the same device that is in line-of-sight with its display. It can only verify the MAC computed over the data received via the wireless channel, and it can measure the delay between displaying the barcode and receiving the MAC on the wireless channel.

Despite the weakness of the presence property, there are still practical applications for devices capable of determining presence. For instance, the presence property is useful in the context of a smart home. It can restrict remote control access of a television to users in the same room. In general, it can serve to limit authority to control a device to users located in view of that device.

Consider the establishment of a security context between a TV and a DVD player to secure wireless communication between the two. The user can use SiB to strongly authenticate the DVD player to her phone through a barcode attached to the DVD player's housing. She can then demonstrate the DVD player's presence to the TV by sending it the public key of the DVD player, along with a MAC over the DVD player's public key:

$$\{K_{DVD}, MAC(K, K_{DVD})\} \rightarrow TV.$$

The TV is then configured to establish a secure, authenticated connection to the DVD player whenever the user selects the DVD player as the active input source on the TV. Taken one step further, the TV can add the DVD player to its list of trusted devices, such that the TV will automatically accept input from the DVD player whenever the user inserts a DVD.

Following the initial publication of SiB (McCune et al., 2004, 2005), Saxena et al. (2006) extended the presence property to achieve authentication if users are willing to perform an integrity check. They devise Visual authentication based on Integrity Checking (VIC), where both devices compute a common checksum on exchanged public data, and compare their results via a unidirectional SiB session on the visual channel. VIC is applicable when both devices have an electronic display, and at least one device has a camera. VIC requires user A to prompt user B as to whether B's device accepted or rejected. User A must then press a button on her device to indicate whether B's device output accept or reject. While this step requires user diligence, it is a simple binary comparison, and may be a viable option when mutual authentication with SiB is not possible or prohibitively complex.

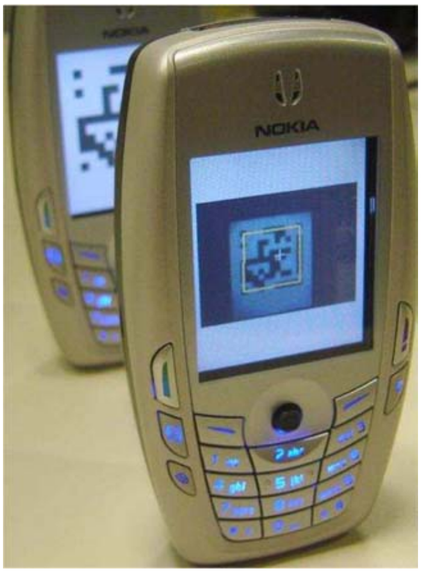
7 Implementation details

7.1 Series 60 phone application

We built SiB in C++ such that it will run on mobile phones running Symbian OS (tested with versions 6.1, 7.0s, and 8.1a) with the Nokia Series 60 User Interface. The size of the Symbian Installation System (SIS) file for SiB is only 52 KB, including a full implementation of RSA. This makes deployment feasible over even the most constrained channels, such as General Packet Radio Service (GPRS).

The Nokia N70 is our development platform today, though we initially developed SiB on the Nokia 6600 and 6620. To present a sense of the user experience with SiB, Figure 4 contains a photograph of SiB in action. Alice's Nokia 6620 (background), is displaying a barcode, while Bob's Nokia 6620 (foreground) is successfully decoding the data encoded in Alice's phone's barcode. In bidirectional authentication with SiB, Alice and Bob would then switch roles. Bob's phone would display a barcode, and Alice's phone would decode it.

Figure 4 SiB application on a Nokia 6620 with one phone scanning the barcode on the LCD of another (see online version for colours)



The barcode format and image processing algorithm in our system is adapted from *Visual Codes* (Rohs and Gfeller, 2004). The data contained in the barcodes for SiB is augmented with Reed-Solomon error correcting codes to provide better performance in the presence of errors in the image processing (Reed and Solomon, 1960). We ported Karn's implementation of Reed-Solomon codes to Symbian OS (Karn, 2002). The SHA-1 cryptographic hash function is used for all hashing operations (Jones, 2001), and all wireless communication occurs via Bluetooth Haartsen (2000).

It is worth pointing out that the last three years have seen a great deal of development in barcode processing on mobile phones. A Java standard for mobile devices has been published that specifies the use of 2D barcodes (JSR-257, 2006). While phones adhering to this Java specification are not yet available, phones are available today that include native barcode processing support, such as the Nokia N95. We plan to update our implementation to take advantage of this support.

To enable users to perform a key exchange between two camera phones, our application generates and maintains an RSA keypair representing the user's identity. We use the XySSL[5] library for RSA operations, including key generation, encryption, decryption, signing, and verification. Ephemeral Diffie and Hellman (1976) key exchange can also be used to establish a shared secret between the two devices, or users can upload their own key files from an existing application.

Bluetooth device discovery is a time consuming and error-prone process, since there is no user-friendly way to distinguish between two devices with the same Bluetooth Device Name. We eliminate the Bluetooth discovery process by including the Bluetooth MAC address in the barcode displayed by the first *Showing* device.

Thus, for a secure and usable SiB exchange, the device that *Shows* first needs to convey 48 bits of Bluetooth address and 160 bits of SHA-1 output (a total of 208 bits) in its barcode. Unfortunately, each *Visual Code* barcode has a useful data capacity of only 68 bits (Rohs and Gfeller, 2004), since 15 of the 83 total bits in the raw barcode format are reserved for Reed-Solomon codes. We now describe how we use multiple barcodes to increase the effective bandwidth of the visual channel.

7.2 Visual channel bandwidth

The visual channel bandwidth between two devices can be increased by choosing a barcode format with a higher data capacity or by using multiple barcodes of a given capacity. There are two basic approaches to using multiple barcodes: cycle through the barcodes one-at-time, or tile the barcodes side-by-side. Cycling is necessary on an electronic screen that is two small to display tiled barcodes, such as the screen on a mobile phone. Tiling is necessary when when cycling is not feasible, due to barcodes being printed on a label instead of displayed electronically. Tiling is also an option on larger electronic displays. In both cases, the Reed-Solomon codes embedded in each barcode indicate whether a processed code is valid or invalid, enabling fully automated scanning of multiple barcodes.

7.2.1 Cycling multiple barcodes

Barcodes can be cycled as fast as the camera and recognition algorithm on the other device can process them. On the Nokia N70, we achieve good results displaying each barcode for one seventh of a second on the *Showing* device, and configuring the camera on the *Finding* device to send bitmap images with a resolution of 160 × 120 to the recognition algorithm.

Encoding a 48-bit Bluetooth address and 160-bit SHA-1 output requires a total of four barcodes, including necessary sequencing information to allow the scanning device to properly reorder the scanned barcodes. The device that *Shows* first must include its Bluetooth address to enable the *Finding* device to initiate the Bluetooth connection between devices. Three barcodes suffice after the devices switch roles, since the Bluetooth connection is already established and only the output of SHA-1 and the sequencing information need to be encoded.

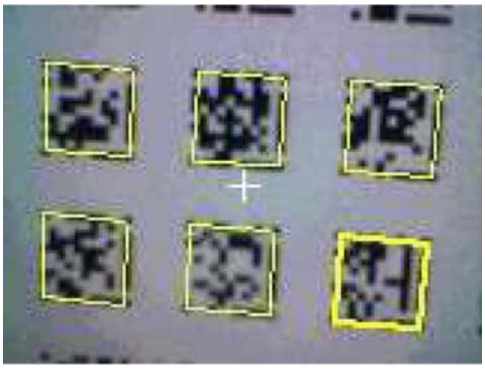
There is a limit to the corrective capability of the Reed-Solomon codes, and barcode scans with significant reading errors can cause the Reed-Solomon codes to report corrected errors when the data remains corrupted. We add an additional checksum to the data payload spread across multiple barcodes to detect and suppress these errors.

We performed timing analysis on our implementation of bidirectional authenticated RSA public key exchange between two Nokia N70s when operated by an experienced user (Table 1). We instrumented the application to track the length of time to generate the user's public key initially, though this is a one-time cost and may be unnecessary if the user has an existing public key on her desktop platform that can be copied to her mobile phone. We measured the length of time the user spends aiming her device before the four (first *Show*) or three (second *Show*) cycling barcodes are successfully recognised, as well as the length of time devices spend cycling their barcodes on-screen. Role-switch is automated via the Bluetooth connection between devices, so we present the latency involved in establishing the Bluetooth connection and causing each device to switch roles.

Table 1 Latency of mutual authenticated key exchange with SiB using our barcode-cycling implementation on Nokia N70s, including user-induced, computational, and Bluetooth overheads. The total is less than the sum of the components because operations may overlap, e.g., Alice's device may have completed barcode recognition and started establishing the Bluetooth connection while Bob's device is still displaying barcodes. RSA operations used 1024-bit keys

Operation	*Avg.* (s)	*St.dev.*
Generate public key	10.145	4.928
Recognise barcodes	5.609	2.079
Establish BT conn.	1.266	0.490
Display barcodes	6.456	1.598
Save public key	0.016	0.000
Total key exchange	*10.452*	*4.452*

Figure 5 Mobile phone screen shot showing the SiB application on a Nokia 6620 recognising multiple tiled barcodes displayed on an LCD screen (see online version for colours)



7.2.2 Tiling multiple barcodes

When scanning tiled barcodes, we configure the camera on the *Finding* device to return higher resolution 640 × 480 bitmaps. We have successfully scanned six tiled barcodes from a laptop's LCD in a single frame at this resolution on the Nokia 6620, as shown in Figure 5. The Nokia N70 supports image sizes up to 1600 × 1200, but the recognition algorithm imposes sufficient processing overhead to degrade the user's view-finding experience at higher resolutions. We conclude that scanning tiled barcodes for a single logical item is a viable implementation strategy.

8 Applications of Seeing-Is-Believing

We initially developed SiB in 2004 (McCune et al., 2004). Since then, we have gained some practical experience with its use in various circumstances, which we relate here.

8.1 Seeing-Is-Believing and the Grey Project

SiB has been in use at Carnegie Mellon for several years as part of the Grey Project (Bauer et al., 2005). Grey is an access control system with mobile phones as the primary development platform, and is currently in use by 25 people to access 35 office and laboratory doors. SiB is used in Grey to allow two users to exchange contact information, including users' public keys, in an authenticated manner. However, the implementation of SiB used by Grey is written in Java, and the performance impact of Java on the Nokia N70 is noticeable. We expect these problems to diminish with the next generation of mobile phones, where we hope to employ native barcode recognition in accordance with JSR-257 (JSR-257, 2006).

8.2 Group key establishment

Secure group communication requires the distribution of authentic information to group members' devices. We consider this problem in a context where members' devices share no prior context. SiB has proven to be quite usable for one-on-one exchange of information, such as between two people, or between one person and a device. However, as part of ongoing research on group key establishment, we have encountered some human-factors challenges when a large number of people try to perform SiB multiple times and in close proximity to one another. Recall that mutual authentication with SiB between two people requires a role switch, where the person whose device was initially in *Show* mode changes to *Find*, and vice versa.

In a group key establishment scenario, we have found that people often make one particular mistake. After performing the first half of a mutual exchange, they look for another person to exchange with, instead of performing the second half. For example, consider Alice, Bob, and Charlie, where all three would like to establish a group key. Alice may photograph Bob's device, and then, when her device switches to *Show* mode, she may allow Charlie to photograph it, instead of Bob. This opportunity for confusion has proved a major obstacle for the development of a usable group key-establishment protocol.

In Section 3, we introduced the property that SiB should either establish authentic communication, or fail. This property applied to SiB in a single direction only, and we achieve mutual authentication by repeating the unidirectional exchange in the other direction. In a group scenario, we require a binding between both unidirectional exchanges. That is, Alice authenticates Bob's key, and then Bob authenticates Alice's key, or else the exchange fails. To make the scheme usable, failure should be an infrequent occurrence. Given our experience with group key establishment, it is worth considering how to achieve strong mutual authentication between two people with only a single unidirectional SiB step. In Section 6, we showed that unidirectional SiB can only provide *presence* to the display-only device, but that an extension to use *Visual authentication based on Integrity Checking* (VIC Saxena et al., 2006) can provide mutual authentication if users are willing to press a button on one device based on whether the other device *accepts* or *rejects*. In particular, VIC reduces by a factor of two the number of SiB exchanges that must be done in a group scenario. It is the subject of future work to determine if this change results in fewer human errors.

8.3 Applications in trusted computing

The Trusted Computing Group (TCG) has specified a Trusted Platform Module (TPM), which is a dedicated security chip designed to increase the resilience of a computing platform to software-based attacks (Trusted Computing Group, 2007). The ability of SiB to demonstratively identify a computing device is useful in the context of trusted computing. We first discuss using SiB to establish TPM identity and configure an identified TPM. Then, we introduce the possibility that SiB can aid in the establishment of a trusted path to the human user of a TCG-compliant computing platform.

8.3.1 Establishing TPM identity

Today, many computing platforms are plagued by spyware that may capture users' actions, including keystrokes, potentially exposing sensitive information like passwords and credit card numbers (Saroiu et al., 2004). The presence of a TPM enables the construction of an Integrity Measurement Architecture (Sailer et al., 2004) that allows *attestations* to be generated by a computing platform, offering a mechanism that may be used to identify certain classes of malicious code such as spyware. In this context, an attestation is a signed list of all the software loaded for execution since boot. The information contained therein can be used by a remote party to make a trust decision about the attesting system, e.g., by examining the signed list for known instances of malware.

Attestations are signed by an asymmetric key that represents the identity of the attesting platform. The key used to sign attestations is an Attestation Identity Key (AIK), which is bound to a particular platform's Endorsement Key (EK) – the public EK is included in a certificate from the manufacturer stating that the TPM complies with the relevant specifications – by a third party called a Privacy CA.[6] The purpose of the Privacy CA is to protect the privacy of the TPM owner by allowing the use of multiple TPM identities, thereby thwarting malicious tracking of a particular computing platform. An attestation signed by an AIK conveys that a platform with a specification-compliant TPM and a particular AIK loaded a particular set of programs. However, it does not truly identify *which* platform loaded these programs. This enables a proxy attack, whereby a compromised machine that is challenged to attest its software state forwards the attestation request to a machine known to be in a benign state, and then forwards the resulting attestation back to the original challenger.

In many situations, it is imperative that the machine in front of the user is the one generating the attestation. For example, consider a bank that requires an attestation from an online banking client to ensure that the client is not running any known malware. An attacker can use a proxy attack to generate an attestation to satisfy the bank using one machine, while still capturing the user's banking credentials on the user's machine.

A solution to the above problem is to enable the user to definitively identify the TPM in her machine and provide that identity to the bank, so that the true origin of any attestations purported to be from the user's machine can be verified. We propose that TPM-equipped machines include a barcode commitment to their public EK somewhere on the case, so that SiB can be used to ascertain the true identity of the TPM in that machine.[7] A mechanism to initially convey the identity of the user's TPM from her mobile device to the bank is also needed. The Phoolproof Phishing Prevention system of Parno et al. (2006) offers one such mechanism.

This architecture also enables a scenario where a user verifies the software stack of their own machine, without involving a third party. The user's mobile device, equipped with the true identity of the TPM in the user's machine, can perform the necessary computation to process an attestation from the user's machine.

8.3.2 Establishing a trusted path for configuration of a TPM

In this section we motivate the establishment of a trusted path to configure the TPM in a TCG-compliant computing platform and then describe how SiB can be used to establish the trusted path. So far, we have assumed that users' devices are uncompromised. In this section, we relax this assumption with respect to the software running on a TCG-compliant computing platform, and discuss ways that SiB can aid in the establishment of a *trusted path* to configure a TPM.

In the previous section, we introduced the notion of an integrity measurement architecture and described how a mobile device can verify an attestation from a platform's TPM to check for malware. However, zero-day exploits and run-time vulnerabilities may remain undetected. Thus, attestation alone is not a solution to the problem of configuring a TPM in the presence of untrusted software.

One challenge in designing systems which incorporate a TPM is how a user can send commands to her TPM securely. The user has only a keyboard and display to communicate with her TPM-equipped platform, with untrusted operating system and window manager software between the I/O devices and the TPM. One solution is to perform TPM configuration in a controlled environment, immediately after initial software installation and before network connectivity. However, this is not viable in practice, as most users lack the necessary expertise and motivation to perform such configuration, or it may be desirable to configure a TPM after a platform has already been in use for some time.

A TPM is configured – typically by a user or vendor – with a secret, the Owner Authorisation Data (OAD), which can be used to exercise control over the TPM. A malicious party that captures the OAD (using, e.g., spyware) can change the OAD, delete (and in some instances change) application secrets in secure storage, and disable or enable TPM features at undesirable times. Unfortunately, if certain TPM features are disabled, a user has no way of knowing if malicious software is running and, e.g., logging all keystrokes. This is a serious problem if the user types the OAD while trying to configure the TPM – the malicious software has just captured the OAD.

Thus, it is undesirable to use the keyboard and display of a computing platform to configure the TPM, since malicious software running on the computing platform may steal the OAD. In the remainder of this section, we show how authentication achievable with SiB enables the user to send commands to the TPM using her camera phone, achieving secrecy so that a malicious application is unable to capture the OAD even if it has subverted the keyboard and display.

We propose the use of a camera phone to securely configure the TPM, where the user enters the OAD *only* on the camera phone. Using SiB, the camera phone can authenticate a TPM's public Endorsement Key, and bootstrap secure communication with the TPM through which the user can enter the OAD and reconfigure the TPM as desired. The mobile device encrypts the OAD for the TPM using the TPM's public Endorsement Key. Inside the TPM, the OAD is decrypted by the private Endorsement Key. The private EK is used exclusively for decryption (it is never used to, e.g., compute a digital signature).

To enable TPM reconfiguration from a camera phone, a sticker must be installed on the housing of the computing platform which contains either a barcode encoding a commitment to the public Endorsement Key, or several barcodes encoding the entire public Endorsement Key. In the case of a commitment, the full Endorsement Key can be obtained from the computing platform in an authenticated way analogous to the wireless access point example in Section 5.

Note that an attacker with direct access to the computing platform can subvert the TPM by physical means. Thus, the use of SiB enhances security under the assumption of software-only attacks, which represent the majority of threats, and requires an attacker to have physical access to the computing platform, ruling out all remote attacks.

9 Security analysis

In addition to the security of the underlying cryptographic primitives, the security of SiB is based on the assumption that an attacker is unable to perform an active attack on the visual channel, and is unable to compromise the mobile device itself. We first discuss the employed cryptographic primitives, then the security properties of various side-channels for authentication. Finally, we discuss attacks against the visual channel.

9.1 Cryptography

Our implementation uses cycling barcodes that provide sufficient bandwidth to convey a full 160 bit SHA-1 hash. As discussed in Section 4, the hash transmitted in the barcode needs to be secure against active attacks, which we achieve through the properties of the visual channel. However, if an adversary can find a second pre-image of the value encoded in the barcode, then a passive attack on the barcode coupled with an active attack against the wireless network connection can be successful. For particularly cautious users, and as mobile phone cameras and displays increase in fidelity, the key itself can be encoded in the barcode, eliminating this dependence on a cryptographic hash function.

9.2 Selecting an authentication channel

Mutual authentication between two parties without the assistance of a trusted authority requires a channel that is secure against active attacks, such as a MITM attack. We analyse potential channels based on the degree to which the user's intentions are captured, and the amount of feedback that the channel provides to the user. Figure 6 contains a summary of proposed channels and their characteristics.

Figure 6 Characteristics of various channels proposed for authentication. We acknowledge that rating the convenience of a channel is subjective; however, we believe it is useful to compare various channels in this way. Section 2 contains a discussion of many of these alternatives. COTS indicates that the necessary hardware is already present in Commercial Off-The-Shelf products. Symbols: yes (●), partial (◐), no (○)

		Resists	
Channel	*COTS*	*MITM*	*Convenient*
Ultrasound	○	○	●
Audible ("beeps")	●	◐	●
Radio	●	○	●
Physical Contact	○	●	●
Near Field Comm.	◐	◐	●
Wired Link	●	●	○
Spoken Passwords	●	●	○
Written Passwords	●	●	○
Visual Hash Verif.	●	●	◐
Infrared	●	◐	◐
Loud-and-Clear	●	●	◐
Seeing-Is-Believing	●	●	●

Activity on channels such as infrared, ultrasound, or radio is undetectable to humans without specialised equipment. Therefore, if Alice believes her device is communicating with Bob's device via infrared, the only assurance she has that it is actually doing so is through status indicators on the two devices. She cannot see infrared radiation leaving her device and entering Bob's, and she certainly cannot see an attacker's device outputting interference patterns and affecting the data stream. Similarly, in case of ultrasound and radio, Alice and Bob need to rely on status indicators of their devices, but they are not sure that Alice's device is indeed setting up a key with Bob's device. Thus, the users' intentions are not captured well, and feedback is indirect and prone to error. Using an audible signal (marked 'beeps' in Figure 6) for data exchange is more intuitive, but this would not work well in noisy environments and is still prone to a MITM attack since it can be difficult for people to tell where 'beeps' originate and how many devices are 'beeping'.

Physical contact between devices is much more intuitive for people and captures the intentions of the users – identifying the devices between which they want to establish a secure communication link (Stajano and Anderson, 1999). Unfortunately, most current devices are not equipped with an interface for this purpose. This may change in the future, however, as Near Field Communication (NFC) interfaces have been standardised for use in mobile phones (JSR-257, 2006). It is necessary to analyse the difficulty of performing an attack against an NFC device from a distance of several meters or more. An alternative approach is to use a wired link, for example connect both devices with a USB cable, however, this approach is not convenient to use and people would need to carry a wire with them.

Another approach is for Alice and Bob to establish a secret password, either by speaking the password aloud, or by writing passwords on paper and passing them to each other. Both Alice and Bob would then need to type in the password correctly, which the devices use to perform a secure password protocol, e.g., EKE (Bellovin and Merrit, 1993). We believe this approach is cumbersome in comparison with SiB, particularly on devices with a limited keyboard.

Finally, both devices could present a visual representation of the hash of the exchanged key material to detect a MITM attack (Dohrmann and Ellison, 2002; Goldberg, 1996; Levien, 1996; Perrig and Song, 1999). Each user must then press a button to indicate whether the images on their devices are the same. These approaches, however, are not secure unless people carefully compare the output of the visual hash function. We believe SiB has an advantage here not just in ease-of-use but because strong authentication is intrinsically linked with device identification.

9.3 Attacks against Seeing-Is-Believing

Active attacks are extremely difficult to perform against the visual channel without being detected by the user. The user has in mind the device at which she is aiming her camera, and will be conscious of a mistake if she takes a snapshot of anything else. We believe the act of taking a picture of *that* device – the one with which the user wants to communicate securely – is intuitive, and should therefore enjoy a low rate of operator error. Thus, the visual channel has the property of being resilient against active attacks (e.g., a MITM attack), and the property that active attacks are easily detected by the user, who can then terminate wireless communication. It is ideal for authentication, providing the user with demonstrative identification of the communicating devices without burdening the user with device names or certificate management.

In Section 6, we discuss a presence property which requires the user to demonstrate that her device can see a display. Kuhn details some attacks which enable a malicious party to read the contents of a CRT screen without actually being in line-of-sight with it. For example, a sophisticated adversary may be able to measure emitted electromagnetic radiation (Kuhn and Anderson, 1998), or to assemble the contents of the CRT by looking at reflected light from the CRT (Kuhn, 2002) Defense against this form of attack is outside the scope of SiB.

An attacker can disrupt the lighting conditions around Alice and Bob in an attempt to disrupt SiB. However, changes of sufficient magnitude to impair SiB are easily observed by Alice, Bob, and any people in the vicinity, alerting them to some kind of unusual behaviour. A more sophisticated, and subtle, attack is to use infrared radiation or a carefully aimed laser to overwhelm the CCD[8] in a phone's camera. If an attacker is able to flood an environment with sufficient infrared radiation or aim a laser directly at the camera's CCD, the CCD in a phone's camera can begin to saturate, and all attempts to take pictures will yield a picture with all pixels set at or above the intensity of the legitimate image, up to the maximum value for each pixel. Essentially, the image becomes noise. Alice will see that the image in her viewfinder is not the picture of Bob's phone that she expects, and can abort the protocol. We have experimented with an off-the-shelf red laser pointer and confirmed these claims.

Even without a user monitoring the process, the electronic-warfare-esque techniques necessary to cause the CCD to output a meaningful image other than the scene in front of the camera are beyond the reach of all but the most sophisticated adversaries with current technology. We are unaware of any attacks feasible today which result in anything but noise from the camera under attack.

10 Conclusion

We propose SIB, a system that uses barcodes and camera phones as a visual channel for human-verifiable authentication. This channel rules out MITM attacks against public-key-based key establishment protocols. The visual channel has the desirable property that it provides demonstrative identification of the communicating parties, providing the user assurance that her device is communicating with *that* other device. SiB enables establishment of a trusted path for configuration of the TPM in a TCG-compliant computing platform. We have also analysed the establishment of secure, authenticated sessions between SiB-enabled devices and devices missing either a camera, a display, or both, and found that secure communication is possible in many situations.

Acknowledgement

We are indebted to the following people for their insightful comments and helpful discussions: Michael Abd-El-Malek, Doug Baker, Lujo Bauer, Leendert van Doorn, Simson Garfinkel, Jason Rouse, and Jesse Walker. Chia-Hsin (Owen) Chen substantially improved the performance and robustness of our implementation on the Nokia N70. Ahren Studer and Cynthia Kuo contributed many ideas on SiB in a group scenario.

This research was supported in part by CyLab at Carnegie Mellon under grants DAAD19-02-1-0389 and MURI W 911 NF 0710287 from the Army Research Office, and grants CNS-0433540, and CNS-0627357 from the National Science Foundation, and by the iCAST project, National Science Council, Taiwan under the Grants No. (NSC95-main) and No. (NSC95-org). The views and conclusions contained here are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either express or implied, of ARO, CMU, iCast, NSF, or the US Government or any of its agencies.

References

Balfanz, D., Smetters, D., Stewart, P. and Wong, H.C. (2002) 'Talking to strangers: authentication in ad-hoc wireless networks', *Proceedings of the Symposium on Network and Distributed Systems Security (NDSS)*, pp.23–35.

Bauer, L., Garriss, S., McCune, J.M., Reiter, M.K., Rouse, J. and Rutenbar, P. (2005) 'Device-enabled authorization in the Grey system', *Information Security: 8th International Conference, ISC 2005*, *Lecture Notes in Computer Science*, Vol. 3650, pp.431–445.

Bellare, M., Boldyreva, A., Desai, A. and Pointcheval, D. (2001) 'Key-privacy in public-key encryption', *Proceedings of Advances in Cryptology (ASIACRYPT)*, pp.568–584.

Bellovin, S. and Merrit, M. (1993) 'Augmented encrypted key exchange: a password-based protocol secure against dictionary atttacks and password file compromise', *Proceedings of the ACM Conference on Computer and Communications Security (CCS)*, pp.244–250.

Bellovin, S.M. and Merrit, M. (1992) 'Encrypted key exchange: password-based protocols secure against dictionary attacks', *Proceedings of the IEEE Symposium on Security and Privacy*, pp.72–84.

Boyko, V., MacKenzie, P. and Patel, S. (2000) 'Provably secure password authentication and key exchange using Diffie-Hellman', *Proceedings of Advances in Cryptology (EUROCRYPT)*, pp.156–171.

Čagalj, M., Čapkun, S. and Hubaux, J-P. (2006) 'Key agreement in peer-to-peer wireless networks', *Proceedings of the IEEE (Special Issue on Cryptography and Security)*, Vol. 94, pp.467–478.

Čapkun, S., Hubaux, J. and Buttyán, L. (2003) 'Mobility helps security in ad hoc networks', *Proceedings of the ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pp.46–56.

Dawson, F. and Howes, T. (1998) *vCard MIME Directory Profile*, RFC 2426.

Dierks, T. and Rescorla, E. (2006) *The Transport Layer Security (TLS) Protocol: Version 1.1*, RFC 4346.

Diffie, W. and Hellman, M.E. (1976) 'New directions in cryptography', *IEEE Trans. Inform. Theory*, IT-22:644–654.

Dohrmann, S. and Ellison, C. (2002) 'Public key support for collaborative groups', *Proceedings of the PKI Research Workshop*, pp.139–148.

Goldberg, I. (1996) *Visual Key Fingerprint Code*, http://www.cs.berkeley.edu/iang/visprint.c.

Goodrich, M.T., Sirivianos, M., Solis, J., Tsudik, G. and Uzun, E. (2006) 'Loud and clear: human-verifiable authentication based on audio', *Proceedings of the IEEE International Conference on Distributed Computing Systems (ICDCS)*, pp.1–10.

Haartsen, J.C. (2000) 'The Bluetooth radio system', *IEEE Personal Communications Magazine*, pp.28–36.

Hanna, S.R. (2002) *Configuring Security Parameters in Small Devices*, draft-hanna-zeroconf-seccfg-00.txt.

Harkins, D. and Carrel, D. (1998) *The Internet Key Exchange (IKE)*, RFC 2409.

Howes, T. and Smith, M. (1998) *MIME Content-Type for Directory Information*, RFC 2425.

ISO/IEC (2006) IS 16022:2006: *Information Technology – Automatic Identification and Data Capture Techniques – Data Matrix Bar Code Symbology Specification*, For review, International Organization for Standardization, Geneva, Switzerland.

Jones, P. (2001) *US Secure Hash Algorithm 1 (SHA-1)*, RFC 3174.

JSR-257 (2006) *JSR-257: Contactless Communication API*, Java Community Process.

Karn, P. (2002) *Reed-Solomon Encoding/Decoding*, http://www.ka9q.net/code/fec/

Kuhn, M.G. (2002) 'Optical time-domain eavesdropping risks of CRT displays', *Proceedings of the IEEE Symposium on Security and Privacy*, pp.3–18.

Kuhn, M.G. and Anderson, R.J. (1998) 'Soft tempest: hidden data transmission using electromagnetic emanations', *Proceedings of the Information Hiding Workshop (IHW)*, pp.124–142.

Laur, S. and Nyberg, K. (2006) 'Efficient mutual data authentication using manually authenticated strings', *Proceedings of Cryptology and Network Security (CANS)*, pp.90–107.

Levien, R. (1996) *PGP Snowflake*, Personal communication.

MacKenzie, P., Patel, S. and Swaminathan, R. (2000) 'Password authenticated key exchange based on RSA', *Proceedings of Advances in Cryptology (ASIACRYPT)*, pp.599–613.

Madhavapeddy, A., Scott, D., Sharp, R. and Upton, E. (2004) 'Using camera-phones to enhance human-computer interaction', *Proceedings of Ubiquitous Computing (Adjunct Proceedings: Demos)*, pp.1–2.

Madhavapeddy, A., Scott, D., Sharp, R. and Upton, E. (2005) 'Using visual tags to bypass Bluetooth device discovery', *Proceedings of the ACM Mobile Computing and Communications Review (MC2R)*, pp.41–53.

McCune, J.M., Perrig, A. and Reiter, M.K. (2004) *Seeing-is-Believing: Using Camera Phones for Human-Verifiable Authentication*, Technical Report CMU-CS-04-174, Carnegie Mellon University, pp.1–22.

McCune, J.M., Perrig, A. and Reiter, M.K. (2005) 'Seeing-is-believing: using camera phones for human-verifiable authentication', *Proceedings of the IEEE Symposium on Security and Privacy*, pp.110–124.

Parno, B., Kuo, C. and Perrig, A. (2006). 'Phoolproof phishing prevention', *Proceedings of the Financial Cryptography and Data Security 10th International Conference*, pp.1–19.

Perrig, A. and Song, D. (1999) 'Hash visualization: a new technique to improve real-world security', *Proceedings of the Workshop on Cryptographic Techniques and E-Commerce (CrypTEC)*, pp.131–138.

Reed, I.S. and Solomon, G. (1960) 'Polynomial codes over certain finite fields', *J. Society for Industrial and Applied Mathematics*, pp.300–304.

Rohs, M. and Gfeller, B. (2004) 'Using camera-equipped mobile phones for interacting with real-world objects', *Proceedings of Advances in Pervasive Computing*, pp.265–271.

Sailer, R., Zhang, X., Jaeger, T. and van Doorn, L. (2004) 'Design and implementation of a TCG-based integrity measurement architecture', *Proceedings of the USENIX Security Symposium*, pp.223–238.

Saroiu, S., Gribble, S.D. and Levy, H.M. (2004). 'Measurement and analysis of spyware in a university environment', *Proceedings of the Symposium on Networked Systems Design and Implementation (NSDI)*, pp.141–153.

Saxena, N., Ekberg, J-E., Kostiainen, K. and Asokan, N. (2006) 'Secure device pairing based on a visual channel (short paper)', *Proceedings of the IEEE Symposium on Security and Privacy*, pp.306–313.

Stajano, F. and Anderson, R. (1999) 'The resurrecting duckling: security issues for ad-hoc wireless networks', *Proceedings of the Security Protocols Workshop*, pp.172–194.

Trusted Computing Group (2007) *Trusted Platform Module Main Specification*, Parts 1–3, Version 1.2, Revision 103.

Uzun, E., Karvonen, K. and Asokan, N. (2007) 'Usability analysis of secure pairing methods', *Proceedings of the Usable Security Workshop*, pp.307–324.

Vaudenay, S. (2005) 'Secure communications over insecure channels based on short authenticated strings', *Advances in Cryptology (CRYPTO)*, Lecture Notes in Computer Science, Vol. 3621.

Wu, T. (1999) 'The secure remote password protocol', *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, pp.97–111.

Notes

[1] The Trusted Computing Group (TCG) is an organisation that promotes open standards to strengthen computing platforms against software-based attacks.

[2] http://www.semacode.com/

[3] http://www.cooltown.com/

[4] http://mobilecodes.nokia.com/

[5] http://xyssl.org/

[6] Version 1.2 of the TPM specification defines Direct Anonymous Attestation (DAA), which eliminates the need for Privacy CAs. However, the TPMs widely deployed today do not offer DAA support.

[7] The Endorsement Key (EK) is an encryption key. The mobile device can act as a Privacy CA and allow the platform to generate a new Attestation Identity Key, or the barcode on the case can encode a commitment to an Attestation Identity Key instead of, or as well as, the Endorsement Key.

[8] Charge Coupled Devices (CCDs) are the prevalent type of image sensor used in today's digital cameras.