Secure Device Pairing based on a Visual Channel (Short Paper)*

Nitesh Saxena†
University of California, Irvine, USA
nitesh@ics.uci.edu
Jan-Erik Ekberg, Kari Kostiainen, N. Asokan
Nokia Research Center, Helsinki, Finland
{jan-erik.ekberg, kari.ti.kostiainen, n.asokan}@nokia.com

Abstract

Recently several researchers and practitioners have begun to address the problem of how to set up secure communication between two devices without the assistance of a trusted third party. McCune, et al. [4] proposed that one device displays the hash of its public key in the form of a barcode, and the other device reads it using a camera. Mutual authentication requires switching the roles of the devices and repeating the above process in the reverse direction.

In this paper, we show how strong mutual authentication can be achieved even with a unidirectional visual channel, without having to switch device roles. By adopting recently proposed improved pairing protocols, we propose how visual channel authentication can be used even on devices that have very limited displaying capabilities.

1. Introduction

The popularity of short-range wireless technologies like Bluetooth and Wireless Local Area Networking is experiencing enormous growth. Newer technologies like Wireless Universal Serial Bus[1] are around the corner and promise to be as popular. This rise in popularity implies that an ever increasing proportion of the users of devices supporting short-range wireless communication are not technically savvy. Such users need very simple and intuitive methods for setting up their devices. Since wireless communication is easier to eavesdrop on and to manipulate, a common set up task is to initialize secure communication. In this paper, we will use the term *pairing* to refer to this operation.[2]

The pairing problem is to enable two devices, which share no prior context with each other, to agree upon a security association that they can use to protect their subsequent communication. Secure pairing must be resistant to a man-in-the-middle adversary who tries to impersonate one or both of these devices. The adversary is assumed to be capable of listening to or modifying messages on the communication channel between the devices. One approach to secure pairing is to use an additional physically authenticatable "out-of-band" (OOB) channel. The adversary is assumed to be incapable of modifying messages on the OOB channel.

There has been a significant amount of prior work on building secure pairing protocols using OOB channels [6, 1]. They consider different types of OOB channels including physical connections, infrared, etc. Recently, McCune, et al. proposed a scheme called "Seeing-is-Believing" (SiB), where the OOB channel is implemented as a visual channel. The SiB visual channel consists of a two-dimensional barcode displayed by (or affixed to) a device A, that represents security-relevant information unique to A. A user can point another camera-equipped device B at the barcode so that B can read the barcode visually, and use this information to set up an authenticated channel to A. If both devices are camera-equipped, they can mutually authenticate each other. "Authentication" in this case is based on demonstrative identification [1] rather than with respect to a claimed name.

In this paper, we propose several extensions to SiB. We start with a brief description of SiB in Section 2. In Section 3, we describe an alternative protocol that enables mutual authentication even when only one device has a camera. In Section 4, we show how visual channel authentication can be done even in highly constrained environments. We discuss the applicability and relevance of our extensions in Section 5.

*Full version of this paper is available at [5]

†Work done while visiting Nokia Research Center, Helsinki

[1] http://www.usb.org/developer/wusb

[2] The term *pairing* was introduced in the context of Bluetooth devices. Other roughly synonymous terms include "bonding", and "imprinting".

2. Seeing-is-Believing (SiB)

In SiB [4], a device A can authenticate to a device B, if B is equipped with a camera. The hash of A's public key is encoded in the form of a two-dimensional barcode. A typical barcode has dimensions approximately 2.5x2.5 cm^2. If A has a display, its public key can be ephemeral, and the barcode is shown on the display. Otherwise, A's public key needs to be permanent and the barcode is printed on a label and affixed to A. Authentication is done by the user pointing B's camera at A's barcode. The basic unidirectional authentication process is depicted in Figure 1.

1. A calculates h_A as $h(K_A)$
   $A \longrightarrow B$ (visual channel): h_A
2. $A \longrightarrow B$ (insecure channel): K_A
   B calculates h' as $h(K_A)$ using the K_A received. If h' does not match the h_A received in Step 1, B aborts.

Figure 1. SiB unidirectional authentication protocol (B authenticates A)

K_A is A's public key. $h()$ is a cryptographic hash function, which is resistant to second pre-image finding. K_A can be long-lived, in which case the output of $h()$ must be sufficiently large, e.g., at least 80-bits. If K_A is ephemeral, the output of $h()$ can be smaller, at least 48 bits [2]. SiB could accommodate 68 bits of hash into a single two-dimensional barcode, but requires a good quality display due to the typical size of the barcode[3]. Mutual authentication requires the protocol of Figure 1 being run in each direction. This has two implications for SiB. First, mutual authentication is possible only if **both** devices are equipped with cameras. A camera-less device can only achieve a property known as "presence" [4]. Presence is weaker than authentication because A has no means of knowing if B is really the device that the user of A intended to communicate with. We summarize the types of authentication achievable using SiB for given combinations of device types in Table 1. Second, in order to run the protocol in each direction, the roles of the devices have to be switched so that first A's camera can scan B's display and then B's camera can scan A's display. This increases the overall execution time. The average execution time in SiB was 8 seconds [4], even though time required to recognize a barcode is just about one second.

These implications limit the applicability of SiB in various practical settings. Many devices cannot have either cameras or high quality displays for different reasons. Commoditized devices like wireless access points are extremely cost-sensitive and the likelihood of adding new hardware for the purpose of authentication is very small. Devices like Bluetooth headsets are typically too small to have displays or even to affix static barcode stickers.

[3] SiB can encode the data into several barcodes displayed in sequence.

Y has $\rightarrow$ X has $\downarrow$	C & D	C only	D only	None
C & D	$X \leftrightarrow$ Y	$X \leftrightarrow$ Y$_s$	$X \leftarrow$ Y $X \xrightarrow{p} Y$	$X \leftarrow Y_s$
C only	$X_s \leftrightarrow Y$	$X_s \leftrightarrow Y_s$	$X \leftarrow Y$ $X \xrightarrow{p} Y$	$X \leftarrow Y_s$
D only	$X \rightarrow Y$ $X \xleftarrow{p} Y$	$X \rightarrow Y$ $X \xleftarrow{p} Y$	none	none
None	$X_s \rightarrow Y$	$X_s \rightarrow Y$	none	none

Notation:
C: Camera, D: Display
P_s: "Device P needs a static barcode label affixed to it."
$P \rightarrow Q$: "Device P can strongly authenticate to device Q."
$P \xrightarrow{p} Q$: "Device P can demonstrate its presence to device Q."

Table 1. Authentication levels in SiB

To summarize, SiB has the following drawbacks:

1. Mutual authentication is not possible unless both devices are equipped with cameras.
2. The need to switch device roles increases overall execution time.
3. Applicability of SiB is limited in situations where one device has limited capabilities (e.g., small size and limited display).

3. Seeing Better: Upgrading Presence to Authentication

We observe that the first two drawbacks stem from the fact that mutual authentication is done as two separate unidirectional authentication steps. Therefore, we propose to solve both problems by performing mutual authentication in a single step by having each of A and B compute a *common* checksum on public data, and compare their results via a unidirectional transfer using the visual channel. Let us call this protocol VIC, for "Visual authentication based on Integrity Checking." (See Figure 2.)

The security of the authentication of A to B in VIC depends on the attacker not being able to find two numbers $X1$ and $X2$ such that $h(K_A, X1) = h(X2, K_B)$. This implies that if the attacker can learn K_B ahead of time, $h()$ needs to be collision-resistant. If K_B is ephemeral (or a nonce picked by B is appended to K_B in message 2 and in the calculation of h_A and h_B), it is sufficient for $h()$ to be resistant to second pre-image finding, since the attacker can no longer use any pre-computed collisions. The security of the authenti-

1. $A \longrightarrow B$ (insecure channel): K_A
2. $A \longleftarrow B$ (insecure channel): K_B
   A calculates h_A as $h(K_A|K_B)$ and B calculates h_B as $h(K_A|K_B)$
3. $A \longrightarrow B$ (visual channel): h_A
   B compares h_A and h_B. If they match, B accepts and continues. Otherwise B rejects and aborts. In either case, B indicates accept/reject to the user.
4. A prompts user as to whether B accepted or rejected. A continues if the user answers affirmatively. Otherwise A rejects.

Figure 2. VIC mutual authentication protocol

cation of B to A depends, in addition, on the user correctly reporting the comparison result reported by B back to A.

Because VIC needs only a unidirectional visual channel, it is now possible to achieve mutual authentication in the cases where SiB could only achieve presence. In addition, the execution time for mutual authentication is shorter since no device role switching is required anymore. Thus, VIC addresses the first two drawbacks of SiB identified in Section 2.

In Table 2, we summarize the types of authentication achievable using VIC for given combinations of device types. Notice that since the checksum is different for each instance of VIC, at least one device must have a display and that the static barcode labels cannot be used with VIC.

Y has → X has ↓	C & D	C only	D only	None
C & D	X ↔ Y	X ↔ Y	X ↔ Y	none
C only	X ↔ Y	none	X ↔ Y	none
D only	X ↔ Y	X ↔ Y	none	none
None	none	none	none	none

Notation
C: Camera, D: Display
P ↔ Q: "Devices P and Q can mutually authenticate."

Table 2. Authentication levels in VIC

4. Seeing With Less: Visual Channel in Constrained Devices

Now we show how to enable visual channel authentication on devices with very limited displays. This is made possible by using key agreement protocols that require short authenticated integrity checksums. We begin by describing such protocols.

4.1. Authentication Using Short Integrity Checksums

The reason why SiB needs good displays is the high visual channel bandwidth required for the SiB protocol. Assuming that the attackers have access to today's state-of-the-art computing resources, the bandwidth needed is at least 48 bits in the case of ephemeral keys [2], rising to 80 bits in the case of long-lived keys. These numbers can only increase over time.

Fortunately, there is a family of authentication protocols that has very low bandwidth requirements. The first protocol in this family was proposed by Gehrmann et al. in [2]. Several subsequent variations on the same theme have been reported [7, 3]. We apply the variation called "MA-3" [3] to get VICsh (VIC with short checksum), as depicted in Figure 3.

1. A chooses a long random bit string R_A and calculates h_A as $h(R_A)$.
   $A \longrightarrow B$ (insecure channel): h_A, K_A
2. B chooses its own long random bit string R_B
   $A \longleftarrow B$ (insecure channel): R_B, K_B
3. $A \longrightarrow B$ (insecure channel): R_A
   B now computes h'_A as $h(R_A)$ and compares it with the h_A received in message 1. If they do not match, B aborts. Otherwise B continues.
4. A calculates hs_A as $hs(R_A, R_B, K_A, K_B)$ and B calculates hs_B as $hs(R_A, R_B, K_A, K_B)$
   $A \longrightarrow B$ (visual channel): hs_A
   B compares hs_A and hs_B. If they match, B accepts and continues. Otherwise B rejects and aborts. In either case, B indicates accept/reject to the user.
5. A prompts user as to whether B accepted or rejected. A continues if the user answers affirmatively. Otherwise A rejects.

Figure 3. VICsh mutual authentication protocol based on short integrity checksum

K_A, K_B are as in the case of SiB. $h()$ represents a commitment scheme and $hs()$ is a mixing function with a short n-bit output (e.g., n = 15 . . . 20) such that a change in any input bit will, with high probability, result in a change in the output. Refer to [3] for formal description of the requirements on $h()$ and $hs()$, and their instantiations, as well as for the proofs of security of the protocol.

4.2. Trimming Down the Display

Now, we describe visual channel authentication using the VICsh protocol on a display-less device containing a single

light source such as a light-emitting diode (LED).

Transmission. We use frequency modulation to encode the data being transmitted (see Figure 4). The sender turns the light-source on and off repeatedly. The data is encoded in the time interval between each successive "on" or "off" event: a long gap represents a '1' and a short gap represents a '0'. Since the channel is unidirectional, the transmitter cannot know when the receiver starts reception. Therefore, the transmitter keeps repeating the sequence until either the user approves the key agreement, or a timeout occurs. The camera phones of today are limited to a frame rate of about 10 video frames/second. Nyquist-Shannon sampling theorem (sampling rate = 2 $\times$ bandwidth for no loss of information) limits the transfer speed with this algorithm to 5 bits/second.

Reception. The receiver processing is analogous: simplified, each received video frame is compressed into one value per frame (the sum of all the pixel values) , and the first-order difference between consecutive values (i.e., the derivative) is compared against a relative threshold based on maximum observed variation in the pixel sum. If the derivative is steep enough and in the right direction (alternating between positive and negative) a transition in lighting is registered. The time between two consecutive changes indicates the transfer of either a '1' or a '0' bit as depicted in Figure 4.

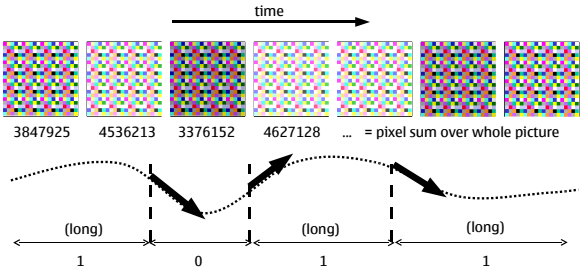

Figure 4. Data transmission via a single light-source visual channel

Trading Efficiency with Security. We designed two mechanisms that allow the possibility of a parameterizable trade-off between execution time and the level of security. First, we can reduce the execution time by exploiting the fact that the the transmitted data (i.e., the integrity checksum) is known to the receiver in advance. The receiver may start reception at any bit position, and records until the n-bit tail of the received bit-string matches against any of the rotated versions of the expected n-bit string. Therefore, the receiver accepts at most n possible matches for the transmitted value. For example, if the transmitted string is '1011', the receiver accepts if it receives any of the strings '1011', '0111', '1110', '1101'. Second, rather than doing error correction, we tolerate a certain number of errors in the n-bit transmission. With k accepted errors, the number of possible matches, based on a binomial distribution of errors, is $\sum_{i=0\ldots k} \binom{n}{i}$.

Using these mechanisms the probability p that the receiver will accept a random string as valid will increase from $\frac{1}{2^n}$ to an upper bound of $p = n \frac{\sum_{i=0}^{k} \binom{n}{i}}{2^n}$. If $k = 3$ bits are allowed to be wrong in an $n = 24$ bit sequence, p is 0.0064, whereas if only 1 bit error is allowed, p is 0.00004.

There are several ways to trade off security and execution time. The attack success probability p can be decreased by: (a) increasing the length of the checksum n, (b) reducing the number of acceptable errors k, (c) reducing the number of possible rotations that are acceptable as matches (say only every fourth), and (d) adding an external end marker (e.g., the light-source staying "on" for 0.5 seconds) to indicate the end of the checksum string, bringing p down to $\frac{\sum_{i=0}^{k} \binom{n}{i}}{2^n}$.

Implementation and Timings. We have developed a proof-of-concept implementation where a single blinking LED (connected to the parallel port of a PC) sends a signal that is received by a camera phone. Figures 5(a) and 5(b) illustrate our two demonstrator implementations. In 5(a), a Bluetooth pairing is established between a Symbian 8.0 camera phone and a Linux laptop with an LED (illustrating, e.g., a WLAN access point). In 5(b), two phones are paired using the display of one phone as the bi-state light.

Our algorithm makes bit reception quite tolerant. The data can be received at a distance of several tens of centimeters, the implementation is agnostic to camera focus problems and tolerates a fair bit of camera shaking, turning, etc.

(a) Pairing phone & laptop (b) Pairing two phones

Figure 5. Pairing Scenarios

With our setup, a 24-bit checksum signaled (1 error accepted) with the laptop is received and matched by the cam-

era phone. The execution times for a positive indication (match) is typically in the range of 5 to 8 seconds. The increased execution time is the price we pay for achieving visual channel authentication with devices that can not afford a full display.

4.3. Extending the Bandwidth on Better Displays

As we saw in Section 4.2, using VICsh with a single light source, and limiting the attack success probability to 2^{-20}, the execution time cannot be smaller than about 5 seconds.

A natural question is whether any speedup in the execution time is possible if there were multiple light sources or in other words, a better display. In the full paper [5], we describe the design and analysis of a new video codec that can be used to set up a visual channel between a device with a small display and a device with a video camera. The essential idea is that the data is encoded for error correction and then represented by multiple black-or-white rectangular slots in each screen frame. The frames are then displayed in sequence at a certain rate to be read by a video camera on the other device.

Our motivation was to investigate two different questions: whether the video codec can significantly improve the transfer time of a short checksum (15-20 bits), so that it can be used to reduce the execution time of secure pairing, and whether the video codec can enable applications other than secure pairing. We show that even with naive image recognition techniques, such a video codec performs reasonably efficiently. We refer the interested reader to [5].

We implemented the preliminary video codec using Python Imaging Library[4] on Linux. In the current implementation, our decoding algorithm is given as input the video frames captured from a camera phone. Overall, it takes approximately $5-7$ seconds for the whole process. We anticipate the performance to improve when the python implementation is ported to a native C++ implementation on the Symbian platform.

5. Discussion

5.1. Comparison of Different Protocols

Table 3 summarizes our recommendations on how mutual authentication can be achieved with different device type combinations. If both devices have camera and display, mutual authentication can be achieved either using SiB or VIC. SiB can be used with camera-only devices which can have static barcodes affixed to them. The case of two display-only devices is out of scope for this paper, and the basic MANA techniques which require the user to visually compare two short strings [2] can be used. In all the other cases, VIC could be the best choice since it provides mutual authentication and potentially better usability.

Y has $\rightarrow$ X has $\downarrow$	C & D	C only	D only
C & D	SiB/VIC	VIC	VIC
C only	VIC	SiB[a]	VIC
D only	VIC	VIC	MANA

Notation:
C: Camera, D: Display
[a]Both devices need static barcode labels affixed to them.

Table 3. Achieving mutual authentication

Table 4 summarizes when to use the two different flavours of VIC: If either one of the devices has a full display, then plain VIC as described in Section 3 can be used. Otherwise VIC combined with MA-3 (which we called VICsh) can be used. Table 4 also summarizes the execution time measurements for the two cases. The execution times for the constrained display case or for the limited display is substantially longer than in full display case. Despite this, we stress that this case is extremely relevant, since not all devices have full displays to support the display of barcodes.

Display type	Recorder type	Protocol	Execution time
Full	Still camera	VIC	1 second[a]
Limited	Video camera	VICsh	5-7 seconds[b]
Constrained	Video camera	VICsh	5-8 seconds[c]

[a]Symbian OS implementation on Nokia 6600 [4]
[b]Python implementation on PC
[c]Symbian OS implementation on Nokia 6630

Table 4. Applicability of proposed protocols

5.2. Device Discovery Strategies

It is often argued [6, 1] that one of the main benefits of using an OOB channel for security initialization is the ease of device discovery. For example in [1] the devices exchange complete addresses over infrared, and thus no in-band device discovery is needed.

We argue that in many scenarios an in-band device discovery is actually needed before the OOB message exchange. The increasing number of different OOB channels (such as infrared, camera and full display, camera and single LED etc.) results in situations where the user might not

[4]http://www.pythonware.com/products/pil/

always know which OOB to use with the two particular devices at hand. It should not be the user's burden to figure out which OOB to use (and how), but instead an in-band device discovery should take place and the best mutually supported OOB channel should be negotiated in-band and the user should be guided to use this OOB.

In order to conveniently discover the desired device in-band, the user must put one of the devices into a temporary special discoverable mode so that the user does not have to select the correct device from a long list of device names. We call this action *user conditioning*. From the user's point of view this action can be performed, e.g., by pressing a button on the device or by selecting a menu option.

5.3. Usability Considerations

The security of VIC and VICsh relies on the user answering affirmatively in the last step (e.g., in Figure 2). If device B rejects the key agreement and indicates failure to the user, but the user inadvertently answers affirmatively in the last step, device A would conclude that the key agreement was authenticated even though B does not. One way to reduce the likelihood of accidental (or out of habit) confirmation is to use a specific confirmation button only for the purpose of secure device pairing. The downside is the cost of adding such a button.

Whether this accidental confirmation is a real concern can only be determined by extensive usability testing. To date, none of the research papers dealing with the problem of secure device pairing have reported substantial *comparative* usability testing. Given the level of recent interest in this area which has resulted in several pairing approaches, a comprehensive comparative usability testing will be a very valuable research contribution. We are addressing this in our current work.

5.4. Denial-of-Service

Another concern is the possibility of a denial-of-service attack. An attacker can disrupt a pairing attempt between two devices by simultaneously initiating pairing with one or both of the same devices. Accidental simultaneous pairing is likely to be very rare because of the user conditioning described in Section 5.2. Thus, if a device detects multiple pairing attempts, the best strategy may be to ask the user to try again later, rather than ask the user to choose the correct device. In addition, part of the device identifier sent via the visual channel can serve as a hint to picking the correct device in case of multiple parallel device pairing attempts. Note that in wireless networks, elaborate attempts to protect the pairing protocol against malicious attempts of denial-of-service are not cost effective because an attacker can always mount denial-of-service by simply disrupting the radio channel.

6. Conclusions

We proposed several extensions to the SiB approach of secure device pairing using a visual channel. We showed how strong mutual authentication can be achieved using just a unidirectional visual channel, and how visual channel authentication can be used even on devices that have very limited displaying capabilities, such as a single LED. Commoditized devices like wireless access points, and devices with form factor limitations like headsets, cannot afford to have full displays. Our contribution makes it possible to use visual channel authentication on such devices.

It would be feasible to trim down the camera to a simple light sensor, resulting in a channel somewhat similar to a unidirectional infrared channel. However, the former has usability and cost advantages: LEDs are typically already available on commodity devices, and an LED light source is easier for the user to visually identify.

Finally, we proposed a *video-based* codec which may help improve the speed of secure pairing in devices with less constrained, but not full, displays, as well as may lead to applications other than secure device pairing.

Acknowledgements: We thank Niklas Ahlgren, Aurélien Francillon, Stanisław Jarecki, Markku Kylänpää, Jonathan McCune, Valtteri Niemi, Kaisa Nyberg, Adrian Perrig, Marie Selenius, and the anonymous reviewers for their valuable comments.

References

[1] D. Balfanz et al. Talking to strangers: Authentication in ad-hoc wireless networks. In *Network and Distributed System Security Symposium, (NDSS)*, February 2002.

[2] C. Gehrmann et al. Manual authentication for wireless devices. *RSA CryptoBytes*, 7(1):29 – 37, Spring 2004.

[3] S. Laur et al. Efficient mutual data authentication based on short authenticated strings. IACR Cryptology ePrint Archive: Report 2005/424 available at `http://eprint.iacr.org/2005/424`, November 2005.

[4] J. M. McCune et al. Seeing-is-believing: Using camera phones for human-verifiable authentication. In *IEEE Symposium on Security and Privacy*, May 2005.

[5] N. Saxena et al. Secure device pairing based on a visual channel. IACR Cryptology ePrint Archive: Report 2006/050 available at `http://eprint.iacr.org/2006/050`, February 2006.

[6] F. Stajano and R. J. Anderson. The resurrecting duckling: Security issues for ad-hoc wireless networks. In *Security Protocols Workshop*, 1999.

[7] S. Vaudenay. Secure communications over insecure channels based on short authenticated strings. In *Advances in Cryptology - CRYPTO*, 2005.