SBVLC: Secure Barcode-based Visible Light Communication for Smartphones (Latest Version)

Bingsheng Zhang, Kui Ren, *Senior Member, IEEE,* Guoliang Xing, *Senior Member, IEEE,* Xinwen Fu, *Senior Member, IEEE,* and Cong Wang, *Member, IEEE*

Abstract—2D barcodes have enjoyed a significant penetration rate in mobile applications. This is largely due to the extremely low barrier to adoption – almost every camera-enabled smartphone can scan 2D barcodes. As an alternative to NFC technology, 2D barcodes have been increasingly used for security-sensitive mobile applications including mobile payments and personal identification. However, the security of barcode-based communication in mobile applications has not been systematically studied. Due to the visual nature, 2D barcodes are subject to eavesdropping when they are displayed on the smartphone screens. On the other hand, the fundamental design principles of 2D barcodes make it difficult to add security features. In this paper, we propose SBVLC - a secure system for barcode-based visible light communication (VLC) between smartphones. We formally analyze the security of SBVLC based on geometric models and propose *physical* security enhancement mechanisms for barcode communication by manipulating screen view angles and leveraging user-induced motions. We then develop three secure data exchange schemes that encode information in barcode streams. These schemes are useful in many security-sensitive mobile applications including private information sharing, secure device pairing, and contactless payment. SBVLC is evaluated through extensive experiments on both Android and iOS smartphones.

Index Terms—Short-range smartphone communication, key exchange, secure VLC, 2D barcode streaming, QR codes.

1 Introduction

Short-range communication technologies including Near Field Communication (NFC) and 2D barcodes have enabled many popular smartphone applications such as contactless payments, mobile advertisements, and data sharing. Evolved from the *radio frequency identification* (RFID) technology, NFC can enable reliable low-power communication between RF tags and readers. However, NFC requires additional hardware and has been supported by only about a dozen of smartphone platforms on the market. Recent studies have shown that NFC is subject to security vulnerabilities such as eavesdropping and jamming. In addition, many types of active attacks, such as data corruption, relay attack [2] and man-in-the-middle attack [3] also have been exploited on NFC-enabled portable devices.

Compared with NFC, 2D barcodes have enjoyed a significantly higher penetration rate in mobile applications. This is largely due to the extremely low barrier to adoption – almost every camera-enabled smartphone can read and process 2D barcodes. As an alternative to NFC, 2D barcodes have been increasingly used for security-sensitive applications including payments and personal identification. For instance, PayPal recently rolled out a barcode-based payment service for retail customers [4]. As one of the handy features of iPhone series, the Passbook App stores tickets, coupons, and gift/loyalty cards using scannable barcodes.

However, the security of barcode-based communication in mobile applications has not been systematically studied. Due to the visual nature, 2D barcodes are subject to eavesdropping when they are displayed on the smartphone screens. The proliferation of smartphones in turn puts a portable camera in everyone's pocket, making eavesdropping significantly easier. This is exacerbated by wide spread use of surveillance cameras in public areas like shopping malls. On the other hand, the fundamental design principles of 2D barcodes make it difficult to add security features. First, a 2D barcode only contains a very limited amount of information and hence cannot adopt advanced encryption primitives. Moreover, most existing barcode applications are based on a single barcode exchange, which is insufficient to establish a secure communication channel. Recently, several systems are designed to *stream* a series of barcodes between a LCD screen and smartphone camera [5], [6]. These systems can enable high-throughput ad hoc communication between smartphones without relying on the Internet connectivity. However, they are designed based on highly customized barcodes which are not widely adopted in practice.

In this paper, we investigate secure barcode-based communication for smartphones. We design a new system that can stream QR codes between smartphones at a throughput comparable to that of state-of-art NFC systems. Due to the inherent directionality, the *visible light communication* (VLC) channel of barcode exchanges yields some interesting security properties. We formally analyze the security of VLC based on geometric models and propose *physical* security enhancement mechanisms such as manupilating

- *Bingsheng Zhang and Kui Ren are with the Department of Computer Science and Engineering, The State University of New York at Buffalo. E-mail: {bzhang26,kuiren}@buffalo.edu.*
- *Guoliang Xing is with the Department of Computer Science and Engineering, Michigan State University. E-mail: glxing@msu.edu.*
- *Xinwen Fu is with the Department of Computer Science, University of Massachusetts Lowell. E-mail: xinwenfu@cs.uml.edu.*
- *Cong Wang is with the Department of Computer Science, City University of Hong Kong, Hong Kong. E-mail: congwang@cityu.edu.hk.*

A preliminary version [1] of this paper was presented at the 33rd IEEE Conference on Computer Communications (INFOCOM'14).

view angles and leveraging user-induced motions. Based on our security analysis, we develop three secure data exchange protocols that encode information in barcode streams. We believe such protocols are useful in many mobile applications including private information sharing, secure device pairing, and contactless mobile payment, etc.

Contributions. We propose SBVLC (Secure Barcode-based Visible Light Communication) – a novel secure ad-hoc wireless communication system for smartphones. Unlike NFC, SBVLC can be widely adopted by most off-the-shelf smartphones. It works across various smartphone platforms equipped with a color screen and a front-facing camera. Our system can also be easily extended to support other mobile and portable devices such as laptops and tablets. We use rigorous 2D and 3D geometric models to thoroughly examine the security of the proposed system. To the best of our knowledge, this work is the first that focuses on modelling and analyzing the security of VLC channel and barcode-based communication between smartphones. Specifically, we first design a real-time duplex screen-camera VLC channel based on 2D barcode streaming. By embedding extra information into the color of *Quick Response* (QR) codes, we developed a fast QR filtering technique to quickly remove the non-QR and duplicate QR frame images. On top of the duplex VLC channel, we further propose three secure communication schemes.

1) *Two-phase message transfer scheme*. It is designed for smartphones to opportunistically exchange data such as contracts and photos. It is ultra lightweight and without using any complex cryptographic building blocks.
2) *Smartphone handshake scheme*. It is developed for the standard key-exchange-then-encryption paradigm. The scheme serves as an alternative key exchange protocol to the conventional DH key exchange protocol[1]. The established key can be used later for many security applications.
3) *All-or-nothing data streaming scheme*. It is tailored for secure temporary data transfer without the key exchange phase. The scheme utilizes all-or-nothing transformation to enhance the channel security — it preserves the confidentiality of all the transmitted data, if the eavesdropper misses at least one barcode frame during the entire communication.

All the proposed schemes are evaluated through extensive experiments on both Android and iOS smartphone platforms. The benchmark result shows that the SBVLC achieves high level security and NFC-comparable throughput.

Road Map. The rest of this paper is organized as follows. Section 2 introduces the system architecture and preliminaries. In Section 3, we give 2D and 3D geometric security models. In Section 4, we enable a real-time one-way screen-camera VLC channel based on color QR codes. In Section 5, we propose various physical protection approaches; we then develop and analyze three secure communication schemes: (a) two-phase message transfer scheme; (b) smartphone handshake scheme; (c) all-or-nothing data streaming scheme. In Section 6, we study the compatibility, usability and robustness of SBVLC system. Finally, Section 7 summaries related work, and a conclusion is given in Section 8.

1. The proposed key exchange protocol is post-quantum secure, while the conventional DH key exchange will be immediately broken once large enough quantum computers are available. NB: we do not claim the efficiency advantage of our key exchange protocol over the DH key exchange protocol. In fact, since their security assumptions are not comparable, it is hard to determine an equivalent security parameter of the DH protocol for a meaningful efficiency comparison.

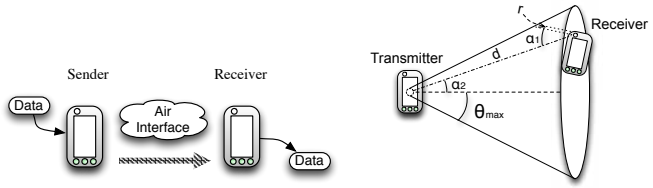
2 PRELIMINARIES

Barcode-based communication. 1D/2D barcodes are widely used to transfer information through optical machine-readable patterns. Nowadays, most off-the-shelf smartphones can read and display barcodes, such as UPC code [7], QR codes [8] and Data Matrix [9]. In particular, QR code was invented in 1994 and approved as ISO/IEC 18004 in 2000. The standard QR code has 40 different versions, ranging from 21×21 to 177×177 modules. QR codes have build-in *error correction code* (ECC), and there are 4 error correcting levels – L (7%), M (15%), Q (25%), H(30%), respectively. To ensure readability to legacy smartphones, only QR codes up to version 10 are mostly used in practice. A single QR code with version 10 can only store 271 characters using 'L' ECC level. For many emerging applications, one QR code is not enough, which could severely hinder its adoption in such applications. It is also the case that existing barcode-based communication systems are easily subject to attacks for its visual nature. We would like to address these issues in this work.

Design Goal and SBVLC architecture. Our goal is to enable secure barcode-based communication between smartphones. The focus is to achieve data confidentiality against eavesdropping. Designed for off-the-shelf smartphone platforms, SBVLC should be lightweight. For example, it is implausible to establish a secure channel for a single-barcode communication with overhead of multiple-round barcode exchange. In addition, we want to avoid any unnecessary cryptographic assumptions. We note that the security of NFC relies on Diffie-Hellman key exchange [10], [11], which is easy to break using quantum computers.

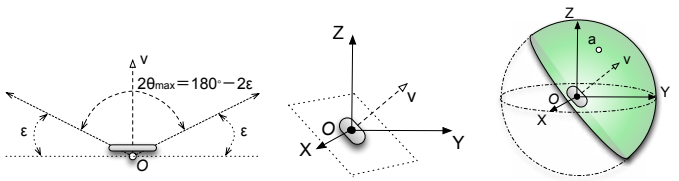
The communication mode of SBVLC is ad-hoc in that the sender and the receiver are not expected to have a common shared secret knowledge such as secret key in priori to the communication. Similar to NFC setting, there is an air interface between the sender and the receiver, and the typical reception distance is also a few inches. As shown in Fig. 1 (a) and (b), SBVLC supports secure data exchange for both smartphone-smartphone and smartphone-terminal scenarios. SBVLC works on top of a fully duplex VLC channel, and thus the smartphones must be equipped with a color screen and a front-facing camera as the sender and the receiver are required to 'talk' to each other simultaneously. SBVLC works among various mobile platforms without specific requirement on the screen size and camera resolution, but a better specification usually leads to higher communication throughput.

VLC channel model. We now give the formal definition of a smartphone VLC channel. Fig. 1 (b) illustrates the VLC channel model, and the parameters are defined as follows. The receiver distance to the source is d and the receiver aperture radius is r. The angle from the source-receiver line and the receiver normal is denoted α_1 and to the source beam axis is denoted α_2, which is also known as the viewing angle. In our context, a one-way smartphone VLC channel consists of a transmitter (realized by a smartphone screen) and a receiver (realized by a smartphone front-facing camera); barcodes are used as the channel coding schemes. In order to achieve real-time communication, the underlying coding scheme must be efficient.



(a) SBVLC System Architecture. (b) VLC Channel Model.

Fig. 1: SBVLC System Architecture and Channel Model.



(a) 2D screen model. (b) 3D screen model. (c) 3D visible zone.

Fig. 2: Screen Model and Visible Zone.

3 SECURITY MODEL

Successful defense against eavesdropping vastly depends on careful analysis of the attack scenarios and adopting suitable protection mechanisms based on the analysis. Before presenting our secure communication schemes, we would like to build formal 2D and 3D geometric security models and study several physical protection mechanisms in this section. The 3D model reflects the situation in reality, but the 2D model is also useful and intuitive, because we can always take a projection map $P : \mathbb{R}^3 \to \mathbb{R}^2$ and project all the objects onto a plane, e.g. by taking the projection matrix

$$P = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} ,$$

we can map any point (x, y, z) in the 3D space $\mathbb{R}^3$ to a point $(x, y, 0)$ as its projection on the x-y plane, which is the plane parallel to the ground.

We now present the 2D/3D geometric model of a smartphone screen. The typical screen size of a mainstream smartphone platform is between 3 and 6 inches. One important feature of a smartphone screen addressed in our model is its visible angle. A 2D screen model with visible angle $2\theta_{\mathsf{max}}$ is depicted in Fig. 2 (a), where the screen is represented as an interval, and the vertex of the screen visible angle is located at the origin $\mathcal{O}$. Let $\theta_{\mathsf{max}} = 90° - \varepsilon$, where $\varepsilon \in [0°, 90°]$. Notice that the smartphone screen visible angles become increasingly wide along with the development of display technology. Current record holder, Samsung super AMOLED screen can achieve $176°$ visible angle; namely, $\varepsilon = 90 - \frac{176}{2} = 2°$. Since ε is usually small, given a typical smartphone screen size, the distance between $\mathcal{O}$ and the screen center is less than 0.1 inch. Considering this distance is negligible to an adversary who is far away, we ignore the tiny difference between $\mathcal{O}$ and the screen center.

Similarly, the screen can be modelled as a plane that passes through the origin in the 3D model. We describe the screen orientation by quantifying its normal vector $\boldsymbol{v} \in \mathbb{R}^3$. As shown in Fig. 2 (b), such plane is uniquely determined by its normal vector $\boldsymbol{v}$, so we denote the screen plane as $\mathsf{pl}(\boldsymbol{v})$. In order to address the notion of visibility, we define the *visible zone* in the 2D/3D model as follows.

Definition Let $t \in \{2, 3\}$. Let $\boldsymbol{v} \in \mathbb{R}^t$ be a normal vector and $\varepsilon \in [0°, 90°]$ be an angle. The visible zone of the screen plane $\mathsf{pl}(\boldsymbol{v})$ is denoted as the set $\mathsf{Vis}_t(\boldsymbol{v}, \varepsilon) \subseteq \mathbb{R}^t$ such that

$$\mathsf{Vis}_t(\boldsymbol{v}, \varepsilon) = \left\{ \boldsymbol{u} \in \mathbb{R}^t \,\middle|\, \frac{\boldsymbol{v} \cdot \boldsymbol{u}}{||\boldsymbol{v}||_2 \cdot ||\boldsymbol{u}||_2} \geq \sin(\varepsilon) \right\} \quad .$$

According to this definition, if a receiver is at location $\boldsymbol{a} \in \mathsf{Vis}_t(\boldsymbol{v}, \varepsilon)$, then the receiver is able to capture information emitted by the screen. (c.f. Fig. 2 (c).) Hence, the distance factor is not taken into account in our notion of visibility. The transmission rate decreases along with the increase of the distance between the transmitter and the receiver for a typical VLC channel. However, similar to the distance factor in the case of NFC, it only offers a fuzzy security guarantee, because it is hard to make assumptions on the attackers' devices. For the sake of uniformity, we don't differentiate the visibility in terms of distance, which only increases the soundness of our security claim.

3.1 2D/3D screen geometric model

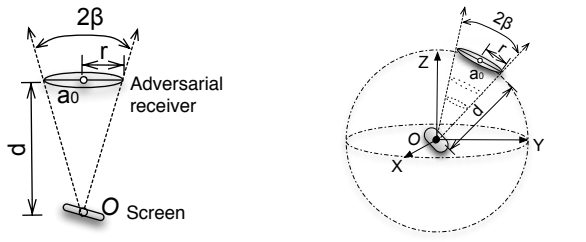
Single-receiver adversarial model. In the *single-receiver adversarial model*, the eavesdropper uses only one optical receiver during an attack event. This is the most common attack scenario in practice: a curious eavesdropper first occasionally discovers a VLC event, and he/she then tries to eavesdrop the communication with his/her carried optical receiver, e.g. a camera or a smartphone. Without loss of generality, the optical sensors of those receivers can be in arbitrarily sharp; in the t-D model, $t \in \{2, 3\}$, for a given optical sensor $\mathcal{D} \subseteq \mathbb{R}^t$, there exists a point $\boldsymbol{a_0} \in \mathbb{R}^t$ such that $\mathcal{D} \subseteq B(\boldsymbol{a_0}, r)$ with a minimum radius $r \in \mathbb{R}$, where $B(\cdot, \cdot)$ denotes a ball. The adversarial receiver is represented by the ball $B(\boldsymbol{a_0}, r)$ in our security analysis, and we note that the adversarial capability is (presumably) increased by this approximation. We assume that the shooting angle of the adversarial receiver can be optimized instantly during an attack; namely the angle $\alpha_1 = 0$ in Fig. 1 (b). Whereas, we don't consider the case that an adversary can physically move his/her receiver a long distance away from its initial position during a short period of time. Hence, position of the adversarial receiver is supposed to be fixed during eavesdropping.

As shown in Fig. 3 (a), the adversary's receiver can be represented as an interval with length $2r$ in the 2D model. Let the phone screen be at the origin $\mathcal{O}$, and the distance between the screen and the adversary's receiver is $d = ||\boldsymbol{a_0}||_2$. One can easily deduce the adversary's capture cone aperture as $2\beta = 2 \cdot \arctan\left(\frac{r}{d}\right)$. Recall that the distance d does not affect the eavesdropping successful rate in our security model. Therefore, in rest of this paper, we only quantify the adversary by the angle β and the position $\boldsymbol{a_0}$ when r and d parameters are not important in the context. Denote the single-receiver adversary as $\mathsf{Adv}_s(\boldsymbol{a_0}, \beta)$. We define the *adversarial capture cone* of $\mathsf{Adv}_s(\boldsymbol{a_0}, \beta)$ in 2D/3D model as:

Definition Let $t \in \{2, 3\}$. The adversarial capture cone of a single-receiver adversary $\mathsf{Adv}_s(\boldsymbol{a_0}, \beta)$ is

$$\mathsf{c}_t(\boldsymbol{a_0}, \beta) = \left\{ \boldsymbol{u} \in \mathbb{R}^t \,\middle|\, \frac{\boldsymbol{u} \cdot \boldsymbol{a_0}}{||\boldsymbol{u}||_2 \cdot ||\boldsymbol{a_0}||_2} \geq \cos(\beta) \right\} \quad .$$

Clearly, all the source beam emitted from the origin $\mathcal{O}$ that lies inside the adversarial capture cone $\mathsf{c}_t(\boldsymbol{a_0}, \beta)$ can be captured by the single-receiver adversary $\mathsf{Adv}_s(\boldsymbol{a_0}, \beta)$. Therefore, we can define 'visibility' as follows.



(a) 2D single-rcvr adv model. (b) 3D single-rcvr adv model.

Fig. 3: Single-receiver adversarial model.

Definition Let $t \in \{2, 3\}$. We say that the screen is visible to a single-receiver adversary $\mathsf{Adv}_s(\boldsymbol{a_0}, \beta)$, if and only if

$$\mathsf{Vis}_t(\boldsymbol{v}, \varepsilon) \cap \mathsf{c}_t(\boldsymbol{a_0}, \beta) \neq \emptyset \ .$$

3.2 2D/3D adversarial geometric model

The radius r is usually very small for a smartphone camera. Whenever r is negligible with respect to d, we have $\beta \approx 0$. We refer this special type of single-receiver adversary $\mathsf{Adv}_s(\boldsymbol{a_0}, 0)$ as *single-point adversary*.

Multi-receiver adversarial model. We now model a more powerful type of adversaries, who are able to control multiple optical receivers to launch an attack. We begin with *two-receiver adversary*, and Fig. 8 illustrates the situation in the 2D/3D model. There are a gap with angle γ (on the $\boldsymbol{a_0}$-$\boldsymbol{a_1}$ plane) between two adversarial capture cones $\mathsf{c}_t(\boldsymbol{a_0}, \beta_0)$ and $\mathsf{c}_t(\boldsymbol{a_1}, \beta_1)$. where $t \in \{2, 3\}$,

$$\gamma = \arccos\left(\frac{\boldsymbol{a_0} \cdot \boldsymbol{a_1}}{||\boldsymbol{a_0}||_2 \cdot ||\boldsymbol{a_1}||_2}\right) - \beta_0 - \beta_1 \ .$$

Denote $\mathsf{Adv}_m(\boldsymbol{a_0}, \beta_0, \boldsymbol{a_1}, \beta_1, \gamma)$ as the two-receiver adversary. In App. A, we reduce a two-receiver (or multi-receiver) adversary to a single-receiver adversary. (c.f. Thm. A.1.)

4 Enabling the SBVLC channel for smartphones

4.1 Channel coding scheme design

First of all, we need to enable a one-way real-time VLC channel between smartphones. We emphasis that all kinds of 1D and 2D barcodes can be the channel coding candidate. Our prototype adopts QR code due to its advantages over other conventional barcodes, including high information density per code and low sensitivity to varying lighting conditions and angles. As depicted in Fig. 4, the barcode streaming system runs between a sender and a receiver. At the beginning of a data transmission, the sender divides the data string into several data chunks. The size of each data chunk depends on the system parameters such as the maximum storage capacity of a single barcode and the error correcting rate of the employed error correcting coded (ECC) such as the classic Reed-Solomon (RS) codes.

Let ℓ_{max} be the maximum package size, which is the maximum raw string length that a single barcode can store before ECC encoding. The data chunk size is the payload size $\ell_p = \ell_{\mathsf{max}} - 16$ bits. The package is then encoded by ECC to a frame block, which is then processed to generate a barcode. The prepared barcodes are sequentially displayed on the sender's screen at a certain frame refresh rate. The receiver starts the decoding process as soon as the first barcode frame is captured by its front-facing camera. The successful barcode decoding process outputs a frame string, which is then decoded by ECC decoded to a package. Finally, the data string is assembled from those received data chunks.

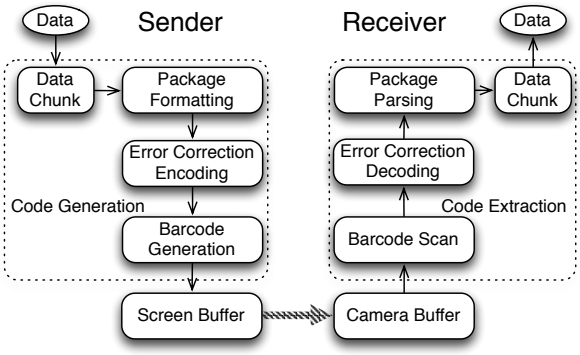


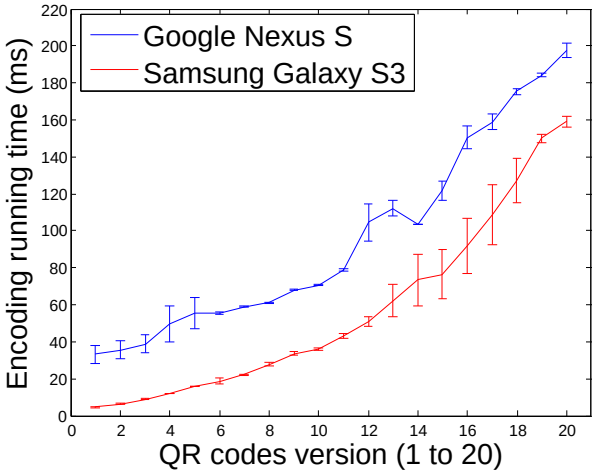
Fig. 4: 2D Barcode Streaming.

4.2 System Integration

Determining the optimal system parameters. SBVLC uses the 8-bit binary mode (mode indicator '0100') for QR code generation. The main system parameters that need to be decided includes the QR version, error correction level and frame refresh rate. In order to determine the proper ECC level, we did statistical test from QR version 1 to 20 on iPhone 4S, Google Nexus S and Samsung Galaxy S3. The result shows that low ('L') ECC level is sufficient in our usage scenario, and there is no correlation between the barcode decoding success rate and the error correction level even for high QR versions. Hence, we pick low ('L') ECC level for better storage capacity per barcode. Each data chunk is formatted to a package by adding a 16-bit sequence number in the header.

In order to achieve a real-time system, we must ensure that each barcode can be encoded and decoded on time. The charts in Fig. 5 show the performance evaluation of single-thread encoding and decoding running time tested on both Nexus S and Galaxy S3. Compared with the encoding running time, the decoding running time grows slower along with the increase of QR versions. This is because high quality QR frame image can be easily decoded with very few errors; subsequently, the ECC-decoding step becomes much faster than the ECC-encoding step.

In order to determine the proper frame refresh rate, we first tested the screen refresh rate and camera capture rate. Our experiment shows that the average time taken to refresh a QR frame screen is roughly the same on different platforms, ranging from 20 to 22 ms. Hence, displaying QR codes is not the system bottleneck unless the frame refresh rate is above 40 frames per second (FPS). In practice, the major challenges are brought by the low camera capture rate. Our system prototype fetches camera image preview using standard `callback` API on Android systems and `avcapturesession` API on iOS systems. The corresponding image capture rates of the front-facing cameras with image size 640×480 on Nexus S, Galaxy S3 and iPhone 4S are 8.3, 25.4 and 30.3 FPS, respectively. We observe that the camera capture rate of a legacy device might be very low, e.g. Nexus S. Since SBVLC requires a fully duplex two-way VLC communication between



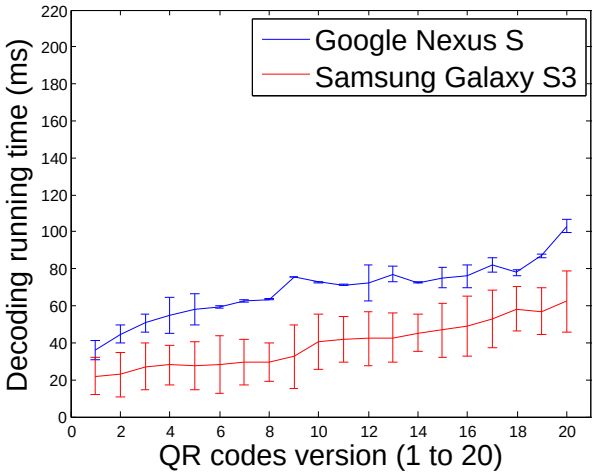
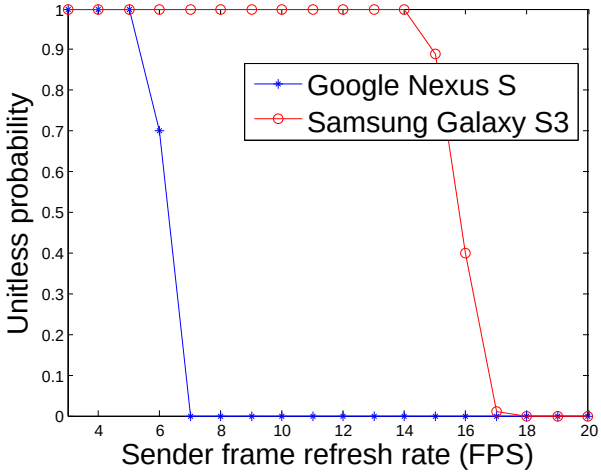


Fig. 5: QR barcode streaming performance.



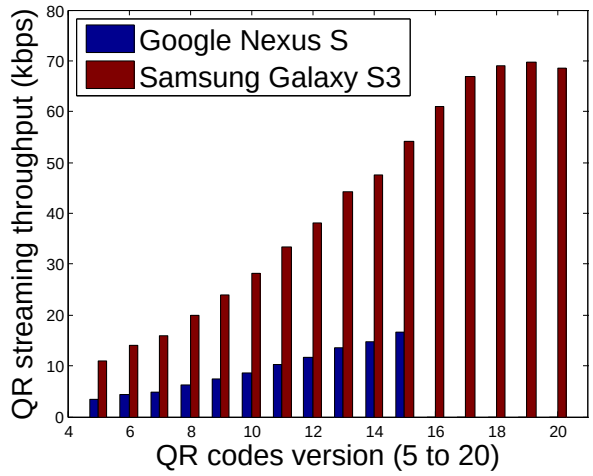


Fig. 6: Channel robustness and QR streaming throughput.

smartphones, the front-facing camera capture rate is crucial. We did channel robustness test to determine the frame refresh rate cap, and the left chart in Fig. 6 illustrates the probability that the receiver (front-facing camera) captures all the QR frames displayed by a sender under different frame refresh rates. The result confirms our conjecture that the ideal frame refresh rate cap τ_{max} should be half of the camera capture rate. Denote $t_{\mathsf{enc}}(i)$ and $t_{\mathsf{dec}}(i)$ be the average encoding and decoding running time (in seconds) of a version-i QR code. We can estimate the ideal frame refresh rate as

$$\tau_f(i) = \max\left(\tau_{\mathsf{max}}, \frac{1}{\max(t_{\mathsf{enc}}(i), t_{\mathsf{dec}}(i))}\right) .$$

Constructing fast QR filtering. Since the frame refresh rate cap is about half of the camera capture rate, it is expected to have multiple camera frame images for the same QR code. So we have to construct an efficient filter to remove duplicated QR frame images. Secondly, the filter should also be able to remove those images that does not contain a QR code before submitting them for decoding. In this section, we propose a novel fast QR filtering technique to remove those non-QR and duplicated QR frame images with only a few image pixel samples.

We utilize the color screen of a smartphone, and let the sender display the QR codes in blue and red alternating order such that any two consecutive QR codes are in different colors. Therefore, we can embed extra information into the colors of the QR codes while maintaining the traditional QR code functionality. Once the receiver captures a frame image, it randomly picks N pixel samples in the central area of the image. According to the RGB value of each pixel, the receiver then classifies the pixels into three bins: 'blue', 'red' and 'others'. The receiver

Algorithm 1: FrameClassifier$(\{p_i\}_{i=1}^{N}, \sigma)$

```
R = 0; B = 0;
for i ← 1 to N do
    if ||p_i − p_r||_1 < σ then
        R++;
    if ||p_i − p_b||_1 < σ then
        B++;
if R > 0 ∩ B = 0 then
    return 'Red';
else if B > 0 ∩ R = 0 then
    return 'Blue';
else
    return 'None';
```

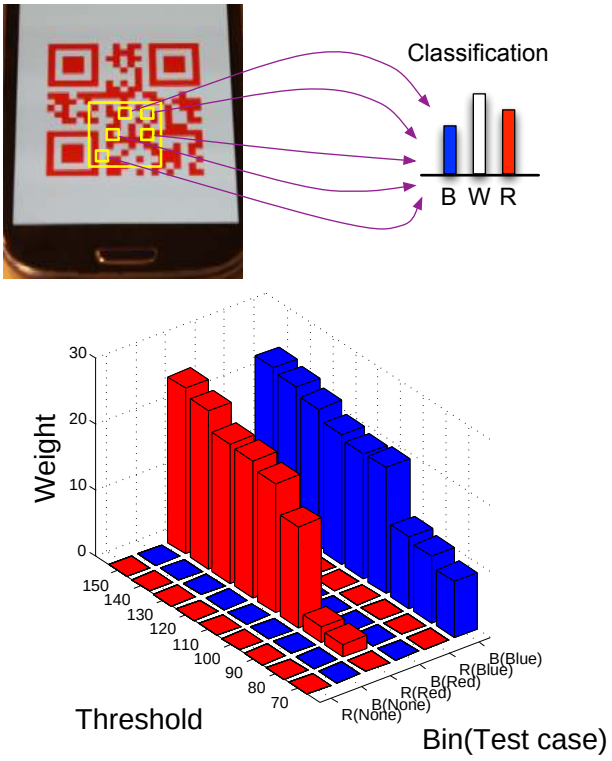
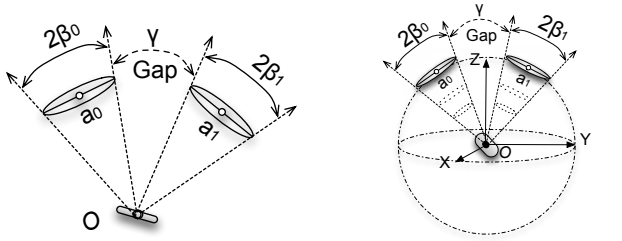


Fig. 7: Fast QR filtering.

will then make decisions based on the weight of those bins. Let $\boldsymbol{p_i} = [R_i, G_i, B_i]^T$ be the RGB vector of the i-th sampled pixel. Define the RGB vectors of red and blue as $\boldsymbol{p_r} = [255, 0, 0]^T$ and $\boldsymbol{p_b} = [0, 0, 255]^T$ respectively. Denote σ as a threshold value, and let $||\boldsymbol{x}||_1$ be the L1 norm of the vector $\boldsymbol{x}$. As described in Alg. 1, the classifier will return 'Red', 'Blue' or 'None', indicating that the image contains a red QR code, a blue QR code or no QR code, respectively. In the context of our system, no QR code means there is no red or blue QR code.

We set the parameter $N = 80$ and run experiments to determine the proper threshold σ. The 3D bar chart of Fig. 7 shows the weights of 'red' and 'blue' bins for different threshold values. The first pair of red and blue rows depict the weights of 'red' and 'blue' bins when test images contain no QR code; the second and third pairs of red and blue rows depict the weights of 'red' and 'blue' bins when test images contain red QR codes and blue QR codes, respectively. We found that both 'red' and 'blue' bins are constantly empty when the test images contain no QR code, even with threshold $\sigma = 150$. The classifier fails to correctly detect the 'red' color when $\sigma \leq 70$, but the weight of 'red' bin catches up quickly along with the increase of threshold. Aftermath, we select $\sigma = 110$ to tolerate the chromatic aberration caused by different smartphones' display screens and cameras. Our empirical result shows that our classifier can distinguish a image that contains no QR, a red QR or a blue QR with 100% accuracy. Its JAVA implementation on Android systems runs less than 0.1 ms on all tested smartphone platforms. Equipped with this classifier, the receiver is able to quickly filter the duplicated QR images with nearly zero computational overhead by removing the following QR images in the same color.

Channel realization We implemented the system on both Android and iOS, borrowing the some parts of the open source



(a) 2D multi-rcvr adv model. (b) 3D multi-rcvr adv model.

Fig. 8: Multi-receiver adversarial model.

QR library [12]. We set the frame refresh cap $\tau_{\max}$ as 5 FPS and 13 FPS for Google Nexus S and Samsung Galaxy S3, respectively. For single-thread encoding/decoding version, we found that the throughput bottleneck becomes the encoding time at the sender-end for higher QR versions in the Samsung Galaxy S3 case. Fortunately, most latest mainstream smartphones are equipped with multi-core CPUs, for instance, iPhone 4S is armed with a dual-core CPU and Galaxy S3 is armed with a quad-core CPU. To explore the benefit of multi-core CPUs, we deploy multiple encoding/decoding threads. On Galaxy S3, with 3 encoding threads, the amortized encoding time for QR version 20 is reduced under 90 ms, which is sufficient to send 10 QR codes per second. At the receiver end, once a frame image is captured by the camera, the receiver first uses our fast QR filter to remove the duplicated QR frames and non-QR frames. The filtered image will be pushed into the decoding queue to be decoded by multiple decoding threads.

Because small camera preview image size leads to higher camera capture rate and lower CPU usage, our system uses adaptive camera preview image resolutions ranging from 192×144 to 800×600 for different QR versions. We tested the QR streaming throughput on both Google Nexus S and Samsung Galaxy S3 from QR version 5 to 20. As illustrated in the right bar chart of Fig. 6, the channel throughput for Samsung Galaxy S3 reaches its peak at 70 kbps with QR version 19. The throughput bottleneck switches from the frame refresh cap to the limited computation resource after QR version 18, and that's why the throughput starts to drop after version 20. On Nexus S, it can only decode the QR codes up to version 15 due to its poor front-facing camera resolution; thus its maximum throughput is below 20 kbps.

5 The Proposed SBVLC Schemes

In this section, we first study the properties of various physical security techniques. We then specially tailor our SBVLC schemes to utilize those underlying security techniques and boost their effectiveness.

5.1 The underlying security techniques

Limiting visible angle. Here, we discuss some physical protection approaches based on limiting visible angle. One simple and effective security protection approach is visual angle blocking. In fact, with the designed working distance, the receiver (smartphone) already blocks about $2 \times 30^\circ$ viewing angle of the sender's screen during the communication. The users can also utilise the existing sheltering items/objects round the communication place such as walls, bodes or ground. (c.f. Fig. 9, left.) If the sender's screen is facing and close to a non-reflecting solid wall, it is easy to keep the screen from being seen by an eavesdropper.

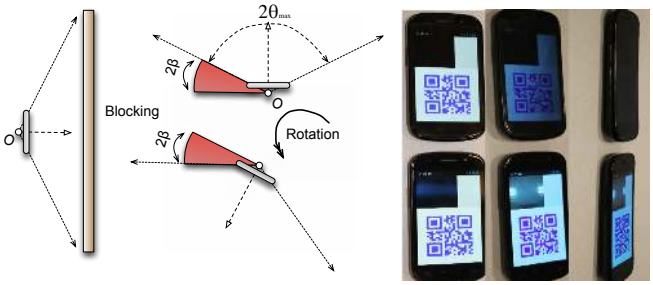


Fig. 9: Physical security approaches: visual angle blocking, rotation and privacy screen projector.

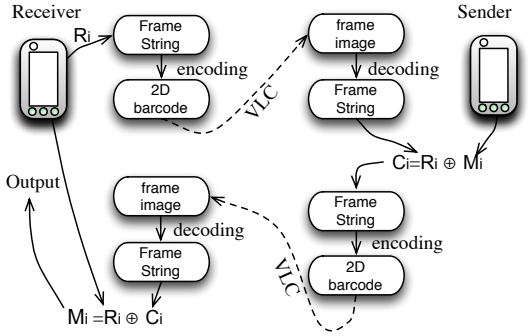


Fig. 10: Two-phase message transfer scheme.

As discussed above, the security of a VLC channel largely depends on the screen visible angle. Therefore, another effective protection approach is to minimize the screen visible angle. One opinion is to use *privacy screen projector* (a.k.a. *screen privacy filter*), e.g. [13], which is widely available in current market. According to [13], the contrast ratio drops to nearly 0 when the viewing angle is larger than $60°$. It means that the maximum visible angle is $2\theta = 120°$ for a screen equipped with a privacy screen projector. The right set of pictures in Fig. 9 shows our experiment results on Nexus S with 3M privacy screen projector. The top smartphones are equipped screen privacy projector, and the bottom ones are without privacy screen projector. From left to right, the pictures are taken with viewing angles $0°$, $30°$ and $60°$ respectively. Note that the usability of a legitimate receiver is not effected, for the privacy screen projector has negligible effect when the viewing angle is small. Our experimental validation confirms that the screen visible angle of a smartphone with privacy screen projector is around $120°$, which gives $\varepsilon \approx 30°$.

Proactive rotation mechanism. In the scenario where there is no proper sheltering objects, the users can still utilize the mobility of the smartphones to enhance the system security. Proactive rotation is a good user-induced motion to prevent the adversary from 'seeing' all the barcode frames. (We will later show how to amplify the security in Sec. 5.3 and Sec. 5.4 if the adversary misses at least one barcode frame.) Before that, we would like to provide some impossibility results in case that an eavesdropper with two of more optical receivers can predict the VLC event place and setup his/her receivers in optimal positions. It is easy to see that the optimal adversarial strategy against proactive rotation should always be distributing his/her receivers uniformly over the $360°$ cycle. In App. A, we show that if $\beta_0 + \beta_1 \geq 2\varepsilon$, there exists optimal receiver positions such that the screen is always visible to the adversary regardless the screen orientation. (c.f. Thm. A.2.)

Note that the visible angles of latest smartphone screens are close to $180°$. Therefore, if the adversary can predict the communicating smartphone screen position, he/she can easily eavesdrop the communication with two receivers. Therefore, the confidentiality cannot be preserved in the presence of an adversary who has two (or more) receivers at optimal positions. However, the communication place is hard to predict in most smartphone VLC case due to its mobility. So we assume that it is difficult for a the adversary to setup his/her devices at optimal positions in priori. We reduce any non-optimal multi-receiver adversary to a single-receiver adversary, whose device is modelled as the minimum ball that contains those receivers. Hence, we will only analyse security in the single-receiver adversarial model in our security analysis.

Let the adversarial capture cone aperture be 2β, and let the screen visible angle be $2\theta_{\mathsf{max}}$. As demonstrated in the middle of Fig. 9, the users have to rotate the screen $\omega > 2(\beta + \theta_{\mathsf{max}}) + \mu$ angle in order to ensure that the adversary cannot 'see' the screen for a moment, where μ is the additional rotation angle to guarantees that there is at least one barcode frame refreshed while rotating μ angle. Therefore, the rotation time for angle μ should be at least $\frac{2}{\tau_f}$, where τ_f is the system frame refresh rate. We can calculate the total rotation time for a given rotation speed ρ as follows:

$$t = \frac{2(\beta + \theta_{\mathsf{max}}) + \mu}{\rho} = \frac{2(\beta + \theta_{\mathsf{max}})}{\rho} + \frac{2}{\tau_f} \ .$$

Although the rotation speed does not effect the total rotation time, the higher speed leads to the larger rotation angle μ. In practice, there is a trade-off between the rotation time and rotation angle and one can derivative the optimal rotation speed based on his/her preference.

5.2 Two-phase message transfer scheme

After building a high-throughput real-time VLC channel, we are ready to focus on the security aspects. In particular, we are going to show that the communication system can achieve much higher security level once it has a duplex VLC channel. Consider the following scenario: Bob wants to share dozens of his contacts with his friend Alice. VLC seems to be an adequate tool to accomplish this task, because it is extremely simple to setup. However, an eavesdropper can shoulder sniff all the information if he/she can 'see' Bob's smartphone screen. To overcome this security issue, we propose the first scheme of SBVLC: two-phase message transfer scheme.

Protocol design. By combining two opposite-directional one-way screen-camera VLC channels, we can enable a fully duplex two-way VLC channel such that both smartphones are able to 'talk' to each other at the same time. We utilize this feature to construct a more secure message transfer protocol as follows. Let ℓ_p be the payload capacity of a single barcode. The sender first divides the data into n chunks with size ℓ_p. Fig. 10 shows the high-level data flow of our two-phase message transfer scheme, where the sender wants to send the receiver one data chunk $M_i \in \{0,1\}^{\ell_p}$. They do the following steps: (1) The receiver first randomly picks $R_i \leftarrow \{0,1\}^{\ell_p}$ and sends R_i to the sender through the receiver-sender VLC channel; (2) The sender fetches R_i and sends $C_i := M_i \oplus R_i$ to the receiver through the sender-receiver VLC channel; (3) The receiver fetches C_i and returns $M_i := C_i \oplus R_i$.

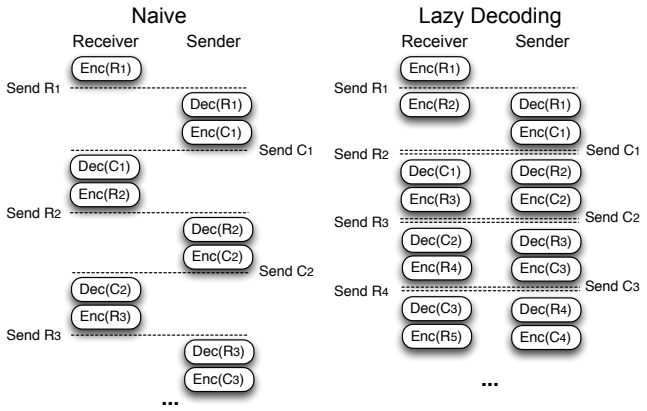


Fig. 11: Lazy decoding technique.

Naively, both smartphones can invoke the above procedure n times to send n data chunks. For $i \in [n]$, both the sender and receiver set a counter $\mathsf{ctr} = i$ and put the counter in the frame header while transferring the i-th data chunk. The receiver first encodes the i-th random frame to a barcode and displays it on its screen. Meanwhile, the receiver keeps checking each frame image captured by the camera, trying to decode a new incoming barcode. Once C_i is received, the receiver extracts $M_i = C_i \oplus R_i$ and repeats the same procedure for data chunk M_{i+1}. Similarly, the sender tries to decode a incoming barcode for R_i. Upon success, the sender encodes $C_i = M_i \oplus R_i$ to a barcode and displays it on its screen. After that, the sender is waiting for the next incoming barcode. In such way, for QR version j, transferring each data chunk M_i takes

$$t(j) = 2 * (t_{\mathsf{enc}}(j) + t_{\mathsf{dec}}(j)) + t_{\mathsf{delay}} \ ,$$

where $t_{\mathsf{enc}}(j)$ and $t_{\mathsf{dec}}(j)$ are the running time of encoding and decoding for QR version j and t_{delay} is the system delay.

To improve the performance, we propose the *lazy decoding technique* as shown in Fig. 11. First of all, since the random frames are independent to the messages, the receiver can prepare the QR codes for random frames during any spare time or even offline. Secondly, we notice that the QR decoding success rate is very high; and thus the image can usually be decoded once it passes our fast QR filter. Therefore, upon receiving a NewBarcode, the receiver can first display the prepared QR code for the next random frame and then try to decode the NewBarcode. If decoding fails, the receiver can simply set the counter ctr of the next random frame to be the missing sequence number, and the sender will try to send the indicated data chunk again. After decoding, the receiver first recovers the message and then prepare the random QR for the next round. The simplified sender and receiver algorithms are described in Alg. 2 and Alg. 3. By applying our lazy decoding technique, the system takes

$$t(j) = t_{\mathsf{enc}}(j) + t_{\mathsf{dec}}(j) + t_{\mathsf{delay}}$$

to transfer each data chunk.

User interface design. We put a small camera preview window at the top of the screen to help the user to quickly align two smartphones such that both QR frame areas are captured by each others' front-facing cameras. Once the alignment is done, the preview window is shadowed at the beginning of the data transmission due to security concerns. (c.f. Sec. 5.2) Alternatively,

Algorithm 2: Sender(M)

$M_1, \ldots, M_n \leftarrow \mathsf{Split}(M)$;
for $i \leftarrow 1$ **to** n **do**
  while *No* NewBarcode *detected* **do**
    Obtain camera preview image;
  $R_i \leftarrow \mathsf{decode}(\mathsf{NewBarcode})$; $C_i = M_i \oplus R_i$;
  $F_i \leftarrow \mathsf{encode}(C_i)$; $\mathsf{display}(F_i)$;
return $\perp$;

Algorithm 3: Receiver$(\cdot)$

$R_1 \xleftarrow{\$} \{0,1\}^{\ell_p}$; $F_1 \leftarrow \mathsf{encode}(R_1)$; $\mathsf{display}(F_1)$;
$R_2 \xleftarrow{\$} \{0,1\}^{\ell_p}$; $F_2 \leftarrow \mathsf{encode}(R_2)$;
for $i \leftarrow 2$ **to** $n+1$ **do**
  while *No* NewBarcode *detected* **do**
    Obtain camera preview image;
  $\mathsf{display}(F_i)$;
  $C_{i-1} \leftarrow \mathsf{decode}(\mathsf{NewBarcode})$;
  $M_{i-1} = C_{i-1} \oplus R_{i-1}$;
  if $i \leq n$ **then**
    $R_{i+1} \xleftarrow{\$} \{0,1\}^{\ell_p}$; $F_{i+1} \leftarrow \mathsf{encode}(R_{i+1})$;
return $M = M_1 || \ldots || M_n$;

we can also blur the preview images on the fly such that the preview images can still assistant users for alignment but the blurred QR codes in the preview window can't be decoded. Hence, we can keep the blurred preview all the time during the whole process. However, we found that the real-time blurring process leads significant computational overhead, and subsequently it effects the performance in current smartphone environment. Therefore, we prefer the shadowing based solution. The screen layout is shown in Fig. 12, and two smartphones are expected to be in opposite direction during a communication. Our experiment shows this alignment can minimise the image distortion caused by the viewing angle, and thus the system is more robust.

Security analysis. It is easy to see that the distribution of R_i is independent to M_i; in addition, C_i itself does not reveal any information about M_i, that is

$$\forall c: \ \Pr[M_i = m | C_i = c] = \Pr[M_i = m] \ .$$

Therefore, the eavesdropper has to 'see' both screens in order to recover the message M_i; however, the time interval between sending R_i and C_i is only a few milliseconds. We can consider

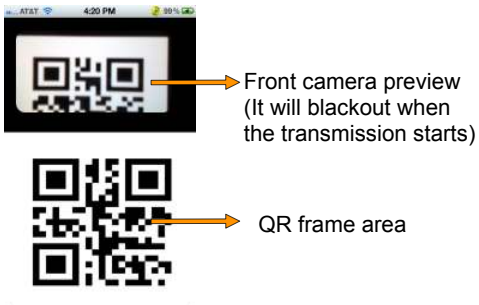


Fig. 12: User interface.

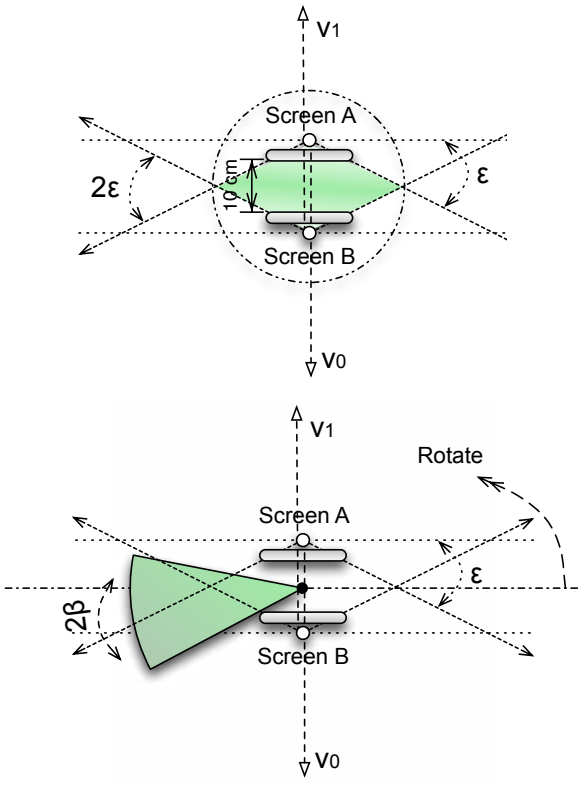


Fig. 13: Security of Two-phase Message Transfer.

both smartphones are sending the corresponding QR code at roughly the same time.

We can show that our two-phase message transfer scheme preserves the confidentiality of the transmitted data string against single-point adversaries in distance. Recall that 'visibility' is defined as the intersection between the adversarial capture cone and the visible zone(s). As depicted in the left of Fig. 13, the distance between two smartphones is around 10 cm, and we define the middle of two phones as the origin. An eavesdropper must be in the shadowed area in order to simultaneously 'see' both phone screens. We bound this shadowed area as a minimum ball $B(\mathcal{O}, d_{\mathsf{save}})$, where $d_{\mathsf{save}} = \frac{5}{\tan(\varepsilon)}$ cm. Hence, $\forall a_0 \notin B(\mathcal{O}, d_{\mathsf{save}})$, a_0 cannot be in both visual zones simultaneously, In other word, if the single-point adversary is more than d_{save}-distance away, then the data confidentiality is preserved. Plugging in the widest smartphone screen visible angle, $\varepsilon = 2°$, we have $d_{\mathsf{save}} \approx 143$ cm. It means that all the single-point adversaries who are more than 1.4 m away cannot eavesdrop the message regardless the quality of their optical devices. On the other hand, any adversary within the range can be easily detected by the user in most circumstances.

If the smartphones are equipped the privacy screen projectors as mentioned in Sec. 5.1, the system achieves much stronger security guarantees. When $\varepsilon = 30°$, we have $d_{\mathsf{save}} \approx 8.66$ cm. It is almost impossible for an adversary to be in this range without being noticed in practice. Fig. 14 illustrates our experiment validation of the security guarantees, and there exists no angle such that the camera can 'see' both screen simultaneously in about 1 foot distance. In general, SBVLC is secure against an adversary

Fig. 14: Eavesdropping experiment on smartphones equipped with privacy screen projectors.

with receiver radius r in distance d such that

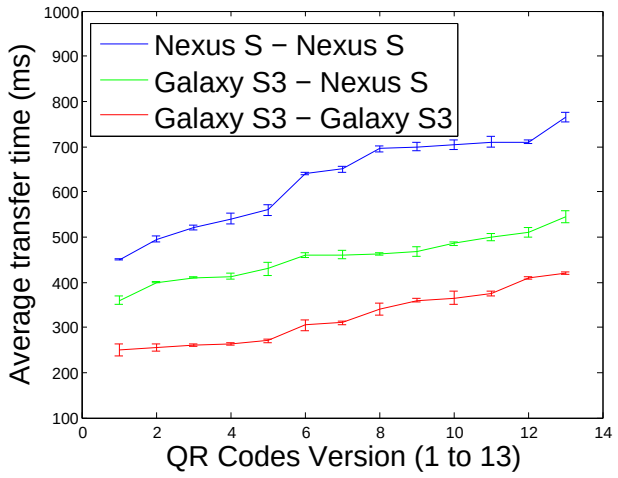
$$\arctan\left(\frac{r}{d - d_{\mathsf{save}}}\right) < \varepsilon \approx 30° \ .$$

When $d_{\mathsf{save}} << d$, we can approximate $d \approx d - d_{\mathsf{save}}$; thus the system can tolerate any single-receiver adversary with $\beta < 30°$. In App. A, we also generalize the result to the 3D case. (c.f. Thm. A.3, below.)

Implementation and performance. Using fully duplex VLC channel, our two-phase message transfer scheme naturally confirms message delivery, so that we don't need a frame refresh cap to avoid missing QR frames. The scheme requires that both the sender-receiver and receiver-sender VLC channels, so its computational requirement is nearly twice higher than the conventional one-way message transfer. The left chart in Fig. 15 shows the average time taken for one data chunk transfer on Galaxy S3 and Nexus S. In the Galaxy S3 case, the average time is between 150 and 200 ms for low QR versions, and it grows gradually as long with the increase of QR versions. The maximum communication throughputs is above 10 kbps in the Galaxy S3 case, and the reason why the throughput drops quickly for higher QR versions is due to the difficulty of stable alignment for both smartphones, which costs the decrease of decoding success rate. Due to lazy decoding technique, when the decoding success rate is low, the amortized transfer time gets longer quickly.

Remark. This scheme can be also used for mobile payment systems, where one party is a smartphone and another is a terminal equipped with a screen and camera, e.g. the users can securely 'show' their movie tickets to the terminal. In those usage scenarios, only a little bandwidth is required. Hence, the current system throughput is sufficient in practice. If we only add privacy screen projector at terminal side, we can compute the safe distance $d_{\mathsf{save}} \approx 17$ cm, which is promising to protect the tickets from shoulder sniffing.

The system throughput depends on 3 important factors: a) the smartphone front-facing camera capture rate, b) the encoding/decoding time and c) the storage capacity per single barcode. Hence a smartphone with high camera capture rate may lead to high throughput, for example it was said that iPhone 4S running on iOS 5 can capture up to 60 FPS; however, iOS 6 limits the maximum camera capture rate to be 30 FPS. In theory, we expect better throughput on iOS devices if we can remove this limitation. In terms of barcode scheme, since QR code is not specifically designed for smartphone environment, its encoding/decoding running time is relatively high for legacy devices. Considering that the frame refresh rate is limited by the camera capture rate, we have to increase the storage capacity per single barcode in order to improve the system throughput. As future work, we would like to replace QR codes with color barcodes [6] for shorter encoding/decoding time and higher storage capacity. It uses mul-



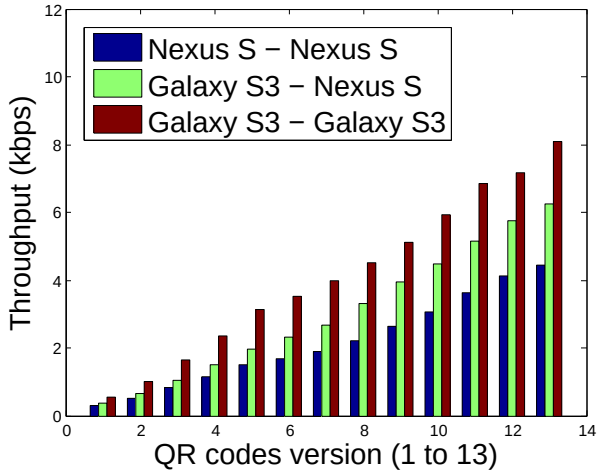


Fig. 15: Two-phase Transfer Performance.

tiple colors for each information block, so it can encode more information than the a mono-color QR code does. For instance, the storage capacity of a single color barcode for the 1280×720 resolution screen of Galaxy S3 with 7×7-pixel block size is about 34K bit. According to [6] the encoding/decoding time is less 20 ms, which is significantly faster than QR codes. We estimate that our system is able to reach above 200 kbps throughput if it adopts color barcode as its coding scheme.

5.3 Smartphone handshake scheme

In this section, we are going to deal with those adversaries whose $\beta \geq \varepsilon$. To preserve data confidentiality against such strong adversaries, we would like to use the standard key-exchange-then-encrypt paradigm. Namely, the sender and the receiver first negotiate a common secret key, and then they use the secret key to encrypt the communication channel with some stream cipher, say Salsa20. Note that the common secret key can be used in many other applications as a substitution of the conventional public-key based key exchange protocol.

Protocol design. We now present our key exchange protocol for smartphones, called *smartphone handshake scheme* that runs between two parties (smartphones) Alice and Bob, and they will establish a common secret key after the execution. We want to design a lightweight scheme that does not rely on any cryptographic

Alice		Bob
$A_0 \xleftarrow{\$} \{0,1\}^\ell$;	$\xrightarrow{A_0 \| 0^\ell}$	$B_0, K_0^b \xleftarrow{\$} \{0,1\}^\ell$;
	$\xleftarrow{C_0 \| B_0}$	$C_0 = A_0 \oplus K_0^b$;
$K_0^b = A_0 \oplus C_0$;		
For $i = 1, \ldots, n$ **do:**		
$A_i, K_i^a \xleftarrow{\$} \{0,1\}^\ell$;		
$D_i = B_{i-1} \oplus K_i^a$;	$\xrightarrow{A_i \| D_i}$	$B_i, K_i^b \xleftarrow{\$} \{0,1\}^\ell$;
	$\xleftarrow{C_i \| B_i}$	$C_i = A_i \oplus K_i^b$;
$K_i^b = A_i \oplus C_i$;		$K_i^a = B_{i-1} \oplus D_i$;
$h \xleftarrow{\$} \mathcal{H}$;	$\xrightarrow{h}$	
Return $sk = h(K_1^a, \ldots, K_n^a, K_0^b, \ldots, K_n^b)$;		

Fig. 16: Smartphone handshake scheme.

assumptions.[2] A typical key exchange scheme between Alice and Bob requires both parties to contribute key material, so we modify the package format to have two payload slots, as depicted above.

The high level protocol is described in Fig. 16. The main idea is as the follows. Bob utilizes the two-phase message transfer scheme to send his key material $\{K_i^b\}_{i=0}^n$ to Alice using payload-A; meanwhile, Alice is also sending her key material $\{K_i^a\}_{i=1}^n$ to Bob using payload-B. At the end, Alice picks a universal hash $h \xleftarrow{\$} \mathcal{H}$ and sends it to Bob. Both parties return their common secret key as

$$sk = h(K_1^a, \ldots, K_n^a, K_0^b, \ldots, K_n^b) \ .$$

After the common sk is established, Alice and Bob can use it to encrypt the one-way VLC channel. Alternatively, it can be used to pair two devices, e.g. Bluetooth.

Security analysis. The scheme should be combined with proactive rotation mechanism to enhance its security. First of all, we show that the rotation based protection approach is much more effective in our scheme, comparing with the standard one-way VLC case mentioned in Sec. 5.1. Recall we have to rotate $\omega > 2(\beta + \theta_{\max}) + \mu = 180^\circ + 2(\beta - \varepsilon) + \mu$ in the standard one-way VLC case. In our scheme, the adversary has to simultaneously 'see' both screens in order to extract the information. As illustrated in right of Fig. 13, since the adversarial capture cone must have intersection with both visible zones, when the rotation angle $\omega' > 2(\beta - \varepsilon) + \mu$ the adversary must lose vision of one of the screens at some moment. Therefore, the users are able to achieve the same security level as in the standard one-way VLC case with 180° less rotation.

Next, we show that if the established key has enough entropy to any eavesdroppers who fail to capture at least one frame from either side of the screen. The *min-entropy* of a random variable X is defined as $H_\infty(X) \stackrel{\text{def}}{=} -\log(\max_x \Pr[X = x])$. If an eavesdropper misses one frame, he/she cannot obtain a pair of (K_i^a, K_i^b) for some $i \in [n]$. The length of (K_i^a, K_i^b) is 2ℓ bits, so the min-entropy of the key materials is at least 2ℓ bits to the eavesdropper. The famous leftover hash lemma [14] states that universal hash functions are good randomness extractors to produce a $2 * \ell + \epsilon$ nearly bits with entropy 2ℓ from a long input string with min-entropy 2ℓ bits. According to leftover hash lemma, if the hash function h is randomly picked from the universal hash

2. Diffie-Hellman key exchange will be immediately broken once we have large enough quantum computer.

function family $\mathcal{H}$, then the entropy of sk is nearly 2ℓ bits to the eavesdropper.

Implementation and performance. In practice, we only need to establish 128- or 256-bit key, so the communication throughput is not crucial for the smartphone handshake scheme. Using QR versions 3 and 4, we can support maximum $\ell = 196$ and 296 bits, respectively. When ℓ is longer than the key length, say 128, we can use a simple algorithm to extract randomness instead of the universal hash, i.e.

$$sk = (\oplus_{i=1}^{n} K_i^a) \oplus (\oplus_{j=0}^{n} K_j^b) \ .$$

In terms of user-induced rotation motion, if the rotation speed is ρ, then in order to guarantee security, the users have to rotate $\omega > 2(\beta - \varepsilon) + \frac{2\rho}{\tau_f}$. For instance, let $\tau_f = 5$ FPS, $\varepsilon = 2^\circ$ and $\beta = 30^\circ$, we can obtain the angle and time trade-off chart as shown in Fig. 18. For example, if a user rotates at speed 40 degrees per second, he/she has to rotate 72°, which takes 1.8 seconds to finish; whereas, if a user rotates at speed 10 degrees per second, he/she only needs to rotate 60° but it takes 6 seconds. On the other hand, a user may always rotate a certain angle ω^* at certain speed ρ^* in practice, and we can deduce the system security level from ω^* and ρ^*.

5.4 All-or-nothing data streaming scheme

In some scenarios where the data string to be transferred is short, so it is not economical to setup a key first. However, one might still want to achieve higher security level. We need a scheme that allows the users to directly transmit the data without key exchange step while still offers high security guarantees. In this section, we propose the all-or-nothing data streaming scheme, which is specifically tailored for secure temporary data transfer without key exchange phase.

Protocol design. The aim of this scheme is to amplify the security such that the confidentiality of the entire transmitted data is guaranteed if the eavesdropper fails to capture at least one data frame. To achieve this goal, the sender first picks a random key and encrypts its data. Then the sender splits the key into many key shares and gradually sends those key shares together with the encrypted data chunks frame by frame. If the adversary miss one frame, then he/she cannot recover the key; subsequently, he/she cannot decrypt the captured data.

To achieve this spacial security feature, we would like to employ an all-or-nothing transformation. As usual, we split the data into n chucks of length ℓ-bit, denoted as $M_1, \ldots, M_n$. Let $\mathsf{PRF} : \{0,1\}^{\ell_k} \times \{0,1\}^{\lambda} \mapsto \{0,1\}^{\ell}$ be a pseudo-random function that takes input as a key $K \in \{0,1\}^{\ell_k}$ and an λ-bit string, and outputs an ℓ-bit pseudo-random string. As shown in Fig. 17, the sender first picks a random key $sk \xleftarrow{\$} \{0,1\}^{\ell_k}$, and then it masks the data chunks by computing

$$U_i = \mathsf{PRF}(sk, i) \oplus M_i$$

for $i \in \{1, \ldots, n\}$. It then compute $U_{n+1} = sk \oplus h(U_1, \ldots, U_n)$, where $h : \{0,1\}^* \mapsto \{0,1\}^{\ell_k}$ is a cryptographic hash and is viewed as a random oracle. The sender then invokes two-phase message transfer protocol to send $U_1, \ldots, U_{n+1}$ to the receiver. After receiving all the data, the receiver first recovers the secret key $sk = U_{n+1} \oplus h(U_1, \ldots, U_n)$ and then recovers the data as $M_i = U_i \oplus \mathsf{PRF}(sk, i)$ for $i \in \{1, \ldots, n\}$.

Security analysis. Analogously, we use proactive rotation based protection approach to ensure that the eavesdropper misses at least one frame. If the adversary misses the last frame, i.e. U_{n+1}, then she does not know sk. Since M_i is masked with a pseudo-random string, the adversary cannot learn anything from U_i. On the other hand, if the adversary misses U_j for some $j \in [n]$, then she cannot recover sk from U_{n+1} either. This is because h behaves as a random oracle and the adversary cannot guess $h(U_1, \ldots, U_n)$ without knowing U_j. To sum up, the confidentiality of all the data chunks is preserved.

Receiver		**Sender**
		$sk \xleftarrow{\$} \{0,1\}^{\ell_k}$;
For $i = 1, \ldots, n$ **do:**		
$R_i \xleftarrow{\$} \{0,1\}^{\ell}$;	$\xrightarrow{R_i}$	$U_i = \mathsf{PRF}(sk, i) \oplus M_i$;
	$\xleftarrow{F_i}$	$F_i = R_i \oplus U_i$;
$U_i = R_i \oplus F_i$;		
$R_{n+1} \xleftarrow{\$} \{0,1\}^{\ell_k}$;	$\xrightarrow{R_{n+1}}$	$U_{n+1} = sk \oplus h(U_1 \ldots U_n)$
	$\xleftarrow{F_{n+1}}$	$F_{n+1} = R_{n+1} \oplus U_{n+1}$;
$U_{n+1} = R_{n+1} \oplus F_{n+1}$		
$sk = U_{n+1} \oplus h(U_1 \ldots U_n)$;		
For $i \in [n]$, **return** $M_i = U_i \oplus \mathsf{PRF}(sk, i)$;		

Fig. 17: All-or-nothing data streaming scheme.

Implementation and performance. The performance of our all-or-nothing data streaming scheme is very similar to the two-phase message transfer scheme. We use AES-128 as the PRF and truncated SHA-1 as the hash function h. Since all the underlying cryptographic primitives are light-weight, the entire scheme is highly efficient. Compared with the aforementioned standard two-phase message transfer protocol, the communication overhead of this scheme is just one additional frame transmission.

6 Compatibility, Usability and Robustness

We tested the compatibility of SBVLC on iPhone 4/4S/5 and many Android smartphone platforms in various environments such as indoor, outdoor. The experiment shows that SBVLC works seamlessly across platforms under different lighting conditions. In terms of usability, we found that the rotation task is hard if two users hold their own phones and try to accomplish the rotation in a collaborative manor. The challenge is brought by maintaining the alignment of those two smartphones such that they are able to 'see' each other's barcode during the rotation. However, it is easy for a person to accomplish the rotation task if he/she holds these two smartphones in his/her both hands respectively. For instance, one can easily keep his/her upper body still and rotate his/her waist for a 90°-rotation task. We tested our system on 40 candidates randomly selected from the campus. Tbl. 1 shows the average time taken by a user to align two smartphones such that both phones can 'see' the other's barcode at the first attempt and after 5-min training. Typically, it takes longer for a user to align two smartphones that are in different size and sharp such as the iPhone

TABLE 1: The average time to align two smartphones for SBVLC communication.

Study case	First attempt	After training
Galaxy S3 - Galaxy S3	$5''$	$2''$
iPhone 4S - iPhone 4S	$6''$	$2''$
iPhone 4S - Galaxy S3	$14''$	$3''$

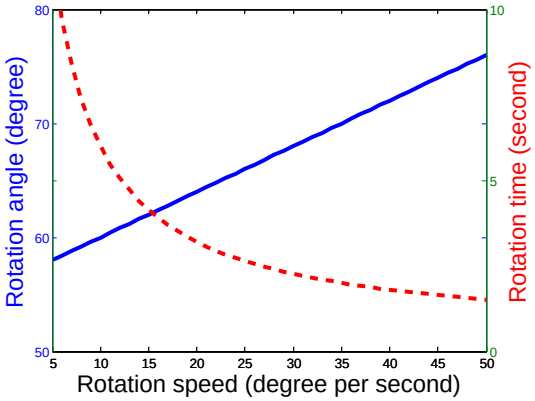


Fig. 18: Rotation angle and time trade-off.

4S-Galaxy S3 pair, but it becomes easier once the users get used to it. Given our single-person rotation instructions, 97.5% candidates can accomplish the 90°-rotation task within the first 2 attempts.

In terms of system robustness, since our focus is data confidentiality against eavesdropping, the scenarios where a barcode itself contains malicious information, e.g. URL, are orthogonal to this work. Many other active attacks, e.g. data modification and injection can be easily detected if the attack devices are near or in between the victims' smartphones; on the other hand, it is hard to implant a fake barcode from distance, for majority of the receiver's camera view is occupied/blocked by the sender's screen. We performed various jamming attacks to test the robustness of SBVLC. For instance, we use a laser pointer to shoot the receiver's camera at different angles. As shown in Fig. 19, the laser beam does not effect our system when the shooting angle is $\geq 60°$. On the flip side, the shooting angle can't be $\leq 30°$ in practice, because of the angle blocking by the other smartphone. In general, due to the usage of *visible light*, the jamming attacks can be easily detected and avoided, utilising the mobility of smartphones or physical blocking.

7 Related Work

Smartphones are widely used to scan 1D or 2D barcodes, such as UPC code, QR codes and Data Matrix. QR Droid [15] is a smartphone App related to this work. In QR Droid, the sender phone encodes a short message into a QR code and displays on its screen; the receiver uses its camera to capture the QR code and decodes it back to the message. The message can be encrypted with DES algorithm under a common secret key configured by both parties. However, there is no automatic key exchange step in the implementation of QR Droid. In terms of barcode design, by taking advantage of more colors, some new color barcodes are proposed to increase the capacity, e.g., *high capacity color barcode* (HCCB) [16]. 4D barcode [5] is recently proposed for robust data transmission between smartphones. However, its throughput on smartphone platforms is as little as 100 bits/s due to the heavy computational overhead of operations like border detection and barcode rectification. PixNet [17] can build a wireless link using LCDs and cameras. The system can achieve high throughput over a long distance based on *orthogonal frequency division multiplexing* (OFDM) and complex computer vision algorithms. Unfortunately, PixNet is not suitable for smartphones due to its high computation overhead. COBRA system [6] can achieve

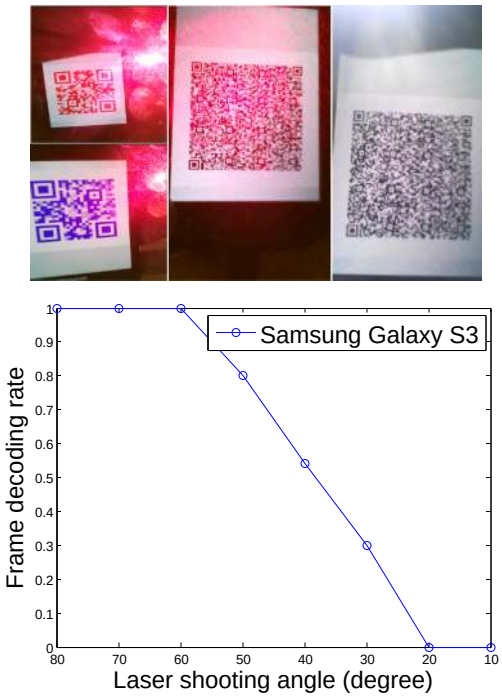


Fig. 19: Jamming experiment. (Tested on Galaxy S3.)

high speed barcode streaming between smartphones based on lightweight image processing techniques. But it improves system throughput by using highly customized barcodes, which are not widely adopted in practice. Moreover, the security of barcodes-based communication is not studied in [6]. Several recent studies have utilized barcode based out-of-band channels as security enhancement primitives. For example, McCune *et al.* proposed the Seeing-is-believing (SiB) system [18] for human authentication. It also can be used for secure device pairing [19]. Kainda *et al.* [20] also formally studied the usability and security of human involved out-of-band channels for device pairing. Similar, QR-TAN [21] was proposed to use QR codes as a VLC channel for transaction authentication. However, these studies only employ barcode-based VLC channels to as some building blocks, and they do not address the security of the barcode-based VLC channels themselves.

8 Conclusion

We proposed SBVLC, utilizing a fully duplex smartphone VLC channel based on 2D barcode. On top of the duplex VLC channel, we further propose three secure communication schemes. All SBVLC schemes are evaluated through extensive experiments on Android smartphones, and the results show that our system achieves high level security and NFC-comparable throughput. The system can be used for private information sharing, secure device pairing and secure mobile payment, etc. To our best knowledge, this work is the first one that formally defines and studies the security of a smartphone VLC system. It serves as a milestone for further development in secure VLC systems for smartphones. In future work, we would like to increase the system throughput, using color barcode streaming [6] as discussed in Sec. 5.2. We will also extend our system to support other mobile and portable devices, e.g. laptops and tablets.

ACKNOWLEDGEMENT

The first author was supported in part by University at Buffalo foundation, project FINER (Greek Secretariat of Research and Technology funded under "ARISTEIA 1."), and ERC project CODAMODA. The second author was supported in part by US NSF grants CNS-1421903, CNS-1318948, and CNS-1262275. The third author was supported in part by US NSF grant CNS-1423102. The fourth author was supported in part by NSF grant CNS-1116644. The last author was partially supported by Research Grants Council of Hong Kong (Project no. CityU 138513).

REFERENCES

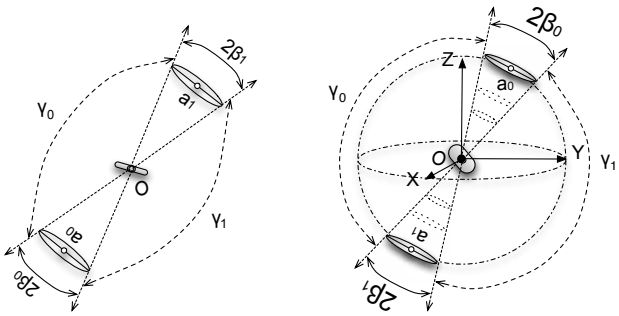
[1] B. Zhang, K. Ren, G. Xing, X. Fu, and C. Wang, "SBVLC: secure barcode-based visible light communication for smartphones," in *IEEE Conference INFOCOM 2014*, 2014, pp. 2661–2669.
[2] L. Francis, G. Hancke, K. Mayes, and K. Markantonakis, "Practical relay attack on contactless transactions by using nfc mobile phones," ePrint Archive, Report 2011/618, 2011.
[3] M. Allah, "Strengths and weaknesses of near field communication (nfc) technology," *GJCST*, vol. 11, no. 3, 2011.
[4] Barcode payment service, http://gigaom.com/2012/05/30/paypal-rolls-out-barcode-payments-in-the-uk/.
[5] T. Langlotz and O. Bimber, "Unsynchronized 4d barcodes: coding and decoding time-multiplexed 2d colorcodes," in *ISVC*, 2007.
[6] T. Hao, R. Zhou, and G. Xing, "Cobra: color barcode streaming for smartphone systems," in *MobiSys*, 2012.
[7] ISO/IEC 15420:2009, Information technology - Automatic identification and data capture techniques - EAN/UPC bar code symbology specification.
[8] ISO/IEC 18004:2006, Information technology - Automatic identification and data capture techniques - QR Code 2005 bar code symbology specification.
[9] ISO/IEC 16022:2006, Information technology - Automatic identification and data capture techniques - Data Matrix bar code symbology specification.
[10] "Norm ECMA-385. NFC-SEC: NFCIP-1 Security and Protocol," 2010.
[11] "Norm ECMA-386. NFC-SEC-01: NFC-SEC Cryptography Standard using ECDH and AES Reference," 2010.
[12] "Zxing (open source qr library)," 2012, http://code.google.com/p/zxing.
[13] R. R. Austin, "Privacy filter for a display Device," June 1996, US Patent No. US5528319.
[14] J. Håstad, R. Impagliazzo, L. Levin, and M. Luby, "Construction of a pseudo-random generator from any one-way function," *SIAM Journal on Computing*, vol. 28, pp. 12–24, 1993.
[15] QRDroid, 2012, http://qrdroid.com/.
[16] D. Parikh and G. Jancke, "Localization and segmentation of a 2d high capacity color barcode," in *WACV*, 2008.
[17] S. Perli, N. Ahmed, and D. Katabi, "Pixnet: interference-free wireless links using lcd-camera pairs," in *MobiCom*, 2010.
[18] J. McCune, A. Perrig, and M. Reiter, "Seeing-Is-Believing: using camera phones for human-verifiable authentication," *Int. J. Secur. Netw.*, vol. 4, no. 1/2, pp. 43–56, 2009.
[19] N. Saxena, J. erik Ekberg, K. Kostiainen, and N. Asokan, "Secure device pairing based on a visual channel," in *S & P*, 2006.
[20] R. Kainda, I. Flechais, and A. W. Roscoe, "Usability and security of out-of-band channels in secure device pairing protocols," in *Symposium on Usable Privacy and Security (SOUPS '09)*. ACM, 2009, pp. 11:1–11:12.
[21] G. Starnberger, L. Froihofer, and K. M. Goeschka, "QR-TAN: Secure Mobile Transaction Authentication," in *Availability, Reliability and Security (ARES '09)*. IEEE, Mar. 2009, pp. 578–583.

APPENDIX

Theorem A.1. In the 2D model, if $\gamma < 2\theta_{\mathsf{max}}$, for a screen with visible angle $2\theta_{\mathsf{max}}$, there exists $\boldsymbol{a}^*$ such that

$$\mathsf{Adv}_s(\boldsymbol{a}^*, \beta_0+\beta_1+\gamma) \equiv \mathsf{Adv}_m(\boldsymbol{a_0}, \beta_0, \boldsymbol{a_1}, \beta_1, \gamma) \ .$$

Proof We want to show that a two-receiver adversary $\mathsf{Adv}_m(\boldsymbol{a_0}, \beta_0, \boldsymbol{a_1}, \beta_1, \gamma)$ is equivalent to a single-receiver adversary $\mathsf{Adv}_s(\boldsymbol{a}^*, \beta_0+\beta_1+\gamma)$ for some $\boldsymbol{a}^*$. Consider an adversary who uses additional devices to fill the blind spot between those two adversarial capture cone, so that he/her can also capture the source beam from the screen that falls into the gap. This modified adversary has a continuous capture aperture $\beta_0+\beta_1+\gamma$, so he/she can be considered as a single-receiver adversary $\mathsf{Adv}_s(\boldsymbol{a}^*, \beta_0+\beta_1+\gamma)$, where $\boldsymbol{a}^*$ lies on the angle bisector. We need to show that this modified adversary has the same capture capability as the original two-receiver adversary. Indeed, they are different if and only if there exists $\boldsymbol{v}$ such that the visible zone $\mathsf{Vis}_2(\boldsymbol{v}, \varepsilon)$ has intersection with the gap but has no intersection with either capture cones $\mathsf{c}_2(\boldsymbol{a_0}, \beta_0)$ or $\mathsf{c}_2(\boldsymbol{a_1}, \beta_1)$. Since $\gamma < 2\theta_{\mathsf{max}}$, such $\boldsymbol{v}$ does not exist. Hence, $\mathsf{Adv}_s(\boldsymbol{a}^*, \beta_0+\beta_1+\gamma) \equiv \mathsf{Adv}_m(\boldsymbol{a_0}, \beta_0, \boldsymbol{a_1}, \beta_1, \gamma)$ as claimed.



(a) 2D optimal positions. (b) 3D optimal positions.

Fig. 20: Optimal positions for two-receiver adversary.

Theorem A.2. In 2D model, if $\beta_0+\beta_1 > 2\varepsilon$, for all $\boldsymbol{v}$, the screen $\mathsf{pl}(\boldsymbol{v})$ with visible angle $2\theta_{\mathsf{max}} = 180^\circ - 2\varepsilon$ is visible by the two-receiver adversary $\mathsf{Adv}_m(\boldsymbol{a_0}, \beta_0, \boldsymbol{a_1}, \beta_1, \gamma)$, where the line a_0-a_1 passes through the origin $\mathcal{O}$.

Proof As shown in Fig. 20 (a), when the line a_0-a_1 passes through the origin $\mathcal{O}$, we have $\gamma_0 = \gamma_1$ due to its symmetry. Given $\beta_0+\beta_1 > 2\varepsilon$, we can deduce that

$$\gamma_0 = \gamma_1 = \frac{360^\circ - 2(\beta_0+\beta_1)}{2} < 180^\circ - 2\varepsilon = 2\theta_{\mathsf{max}} \ .$$

According to Thm. A.1, we can reduce both cases $\mathsf{Adv}_m(\boldsymbol{a_0}, \beta_0, \boldsymbol{a_1}, \beta_1, \gamma_0)$ and $\mathsf{Adv}_m(\boldsymbol{a_0}, \beta_0, \boldsymbol{a_1}, \beta_1, \gamma_1)$ to the single-receiver adversaries. Subsequently, the adversary can cover the entire 360° cycle, so the screen $\mathsf{pl}(\boldsymbol{v})$ is always visible to the adversary for all $\boldsymbol{v}$.

We now show that similar result holds in the 3D model as well. (cf. Fig. 20 (b).) Recall that the visibility is defined as the intersection between the screen visible zone $\mathsf{Vis}_3(\boldsymbol{v})$ and the adversarial capture cones. It is easy to see that

$$\forall \boldsymbol{v} \in \mathbb{R}^3 : \ \mathsf{Vis}_3(\boldsymbol{v}, \varepsilon) \cap (\mathsf{c}_3(\boldsymbol{a_0}, \beta_0) \cup \mathsf{c}_3(\boldsymbol{a_1}, \beta_1)) \neq \emptyset \ .$$

Hence, the screen is always visible by the adversary.

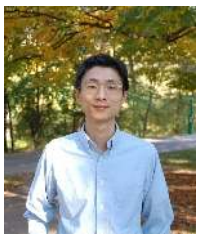
Theorem A.3. $\forall \boldsymbol{a} \in \mathbb{R}^3$, the $(\boldsymbol{a}, \beta)$-single-receiver adversary with $\beta < \varepsilon$ is not capable of eavesdropping any information about the data transmitted by the two-phase message transfer scheme (c.f. Alg. 2 and Alg. 3).

Proof Since the visibility is defined as the intersection between screen visual zones and the adversarial capture cone. It is easy to see that when $\beta < \varepsilon$ the adversarial capture cone $\mathsf{c}_3(\boldsymbol{a}, \beta)$ cannot simultaneously intersect with both screen visible zones

$\mathsf{Vis}_3(\boldsymbol{v_0}, \varepsilon)$ and $\mathsf{Vis}_3(\boldsymbol{v_1}, \varepsilon)$, where $\boldsymbol{v_0} = -\boldsymbol{v_1}$. Therefore, at least one of the two phone screens is invisible to the adversary at any given time, so the claim holds for all $\boldsymbol{a} \in \mathbb{R}^3$.

B ingsheng Zhang is a postdoctoral researcher at Department of Informatics and Telecommunications, National and Kapodistrian University of Athens, Greece. He received his B. Eng. in computer science from Zhejiang University of Technology in 2007, his M. Sc. in information security from University College London in 2008, and his Ph.D. degrees in computer science from University of Tartu in 2011. Before his current appointment, he was a postdoctoral researcher at Department of Computer Science and Engineering, University at Buffalo, SUNY, USA, and before that he was a part-time research associate at University College London and a full-time researcher at Cybernetica AS.

K ui Ren is an associate professor of Computer Science and Engineering and the director the Ubiquitous Security and Privacy Research Lab at State University of New York at Buffalo. He received his PhD degree from Worcester Polytechnic Institute. Kui's current research interest spans Cloud & Outsourcing Security, Wireless & Wearable System Security, and Human-centered Computing. His research has been supported by NSF, DoE, AFRL, MSR, and Amazon. He is a recipient of NSF CAREER Award in 2011 and Sigma Xi/IIT Research Excellence Award in 2012. Kui has published 135 peer-review journal and conference papers and received several Best Paper Awards including IEEE ICNP 2011. He currently serves as an associate editor for IEEE Transactions on Information Forensics and Security, IEEE Wireless Communications, IEEE Internet of Things Journal, IEEE Transactions on Smart Grid, Elsevier Pervasive and Mobile Computing, and Oxford The Computer Journal. Kui is a senior member of IEEE, a member of ACM, a Distinguished Lecturer of IEEE Vehicular Technology Society, and a past board member of Internet Privacy Task Force, State of Illinois.



G uoliang Xing is an Associate Professor in the Department of Computer Science and Engineering at Michigan State University. He received the B.S. degree in Electrical Engineering from Xian Jiao Tong University, China, in 1998, and the M.S. and D.Sc. degrees in Computer Science and Engineering from Washington University in St. Louis, in 2003 and 2006, respectively. From 2006 to 2008, he was an Assistant Professor of Computer Science at City University of Hong Kong. He is an Associate Editor of ACM Transactions on Sensor Networks. He received the Best Paper Awards at the 18th IEEE International Conference on Network Protocols (ICNP) in 2010 and the 12th ACM/IEEE Conference on Information Processing in Sensor Networks (IPSN) SPOTS track in 2012. He is an NSF CAREER Award recipient in 2010. His research interests include Cyber-Physical Systems for sustainability, mobile health, smartphone systems, and wireless networking.

X inwen Fu is an associate professor in the Department of Computer Science, University of Massachusetts Lowell. He received B.S. (1995) and M.S. (1998) in Electrical Engineering from Xi'an Jiaotong University, China and University of Science and Technology of China respectively. He obtained Ph.D. (2005) in Computer Engineering from Texas A&M University. Dr. Fu's current research interests are in network security and privacy, digital forensics and networking QoS. He has been publishing papers in conferences such as IEEE S&P, ACM CCS, ACM MobiHoc, journals such as ACM/IEEE ToN, IEEE TPDS, IEEE TC, and IEEE TMC, book and book chapters. Dr. Fu spoke at various technical security conferences including Black Hat. His research was aired on CNN and reported by Wired, Huffington Post, Forbes, Yahoo, MIT Technology Review, China Central Television (CCTV). His research is supported by NSF.

C ong Wang is an Assistant Professor in the Computer Science Department at City University of Hong Kong. He received his B.E and M.E degrees from Wuhan University in 2004 and 2007, and PhD degree from Illinois Institute of Technology in 2012, all in Electrical and Computer Engineering. He has worked at Palo Alto Research Center in the summer of 2011. His research interests are in the areas of cloud computing and security, with current focus on secure data services in cloud computing, and secure computation outsourcing. He is a Member of the IEEE and a Member of the ACM.