Självständigt arbete på avancerad nivå

Independent degree project – second cycle

M.Sc. Thesis

within Computer Engineering C, course, 30 points

Device Pairing Using Visible Light Communications

Shangyuan Guo
Abstract

Device authentication in ad hoc networks is becoming more and more important. Nowadays, there are many interesting applications which communicate via the short-rang wireless communication channel (such as Bluetooth or WiFi). In the communication, a great deal of sensitive information is required to be transmitted. Therefore, device authentication is significant. In order to build a secure authentication mechanism, protocols are proposed using human control visual channels. However, this method brings many challenges, the main one being the burden placed on humans. Therefore, in this thesis, these protocols are optimized using visible light communication techniques, which significantly reduce the work faced by humans.

Keywords: Ad hoc networks, Security, Visual channel.
Acknowledgements

First of all, I would like to express my sincere gratitude to my supervisor Xin Huang. He enabled me to take part in this meaningful project, and provided me with a great deal of useful guidance and suggestions for this project. Without his valuable support, it would have been impossible to accomplish this thesis.

I also would like to express heartfelt thanks to Professor Tingting Zhang. She also provided me with valuable comments and constructive suggestions. Most importantly, she helped me to overcome many significant drawbacks. Without her encouragement and suggestions, it would have been impossible to accomplish this thesis.

Finally, I am deepest grateful to my parents for all their love and support.
Table of Contents

Abstract .................................................................................................................. iii
Acknowledgements ............................................................................................... iv
Table of Contents .................................................................................................. v
Terminology ........................................................................................................... vii
Acronyms vii

1 Introduction ........................................................................................................ 1
  1.1 Background and problem motivation ......................................................... 1
  1.2 Overall aim .................................................................................................. 2
  1.3 Scope .......................................................................................................... 2
  1.4 Concrete and verifiable goals ...................................................................... 3
  1.5 Outline ....................................................................................................... 3
  1.6 Contributions ............................................................................................. 4

2 Related work ...................................................................................................... 5
  2.1 Body Area Network ................................................................................... 5
  2.2 BAN Characteristics ................................................................................ 6
  2.3 Visible Light Communication .................................................................... 6
  2.4 Manual method .......................................................................................... 7
  2.5 Key Distribution ......................................................................................... 7
  2.6 OOB-based security protocols .................................................................. 8
  2.7 HCBK protocol ......................................................................................... 8

3 Methodology ...................................................................................................... 10
  3.1 Channels and Attack Model ...................................................................... 11
  3.2 Diffie-Hellman Key Exchanges ................................................................ 12
  3.3 Quick Response code .............................................................................. 15
  3.4 Digest Function and Hash Function ............................................................ 18
  3.5 Human factors .......................................................................................... 19
  3.6 Technological factors ................................................................................ 20

4 Quick Response code based channel .............................................................. 21
  4.1 QR-based channel establishment ............................................................... 21
  4.2 SiB protocol using QR-based channel ..................................................... 22

5 Human controlled LED-Camera channel ...................................................... 25
  5.1 HLC Channel establishment .................................................................... 25
### Table of Contents

5.1.1 On the transmitter side ........................................ 26
5.1.2 On the receiver side ........................................ 26
5.2 HLC-HCBK Protocol ........................................ 32
5.3 System design ................................................ 34

6 Results ........................................................................ 36
6.1 Demonstration for SiB protocol ................................. 36
6.2 HLC Channel and Human Burden ............................... 39
6.3 Performance for the LED location detection process ........ 41

7 Security analysis ...................................................... 43
7.1 Man in the middle .................................................. 43
7.1.1 MITM is eliminated in SiB protocol ......................... 43
7.1.2 MITM is eliminated in HLC-HCBK protocol .......... 44
7.2 Denial of Service ................................................... 45
7.2.1 DoS against SiB protocol ................................... 45
7.2.2 DoS against HLC-HCBK protocol ......................... 45
7.3 Offline Pin Cracking .............................................. 46
7.4 Other security considerations .................................. 46

8 Conclusions ............................................................. 47

References .................................................................... 48
## Terminology

### Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>OOB</td>
<td>Out of Band</td>
</tr>
<tr>
<td>HLC</td>
<td>Human controlled LED-Camera</td>
</tr>
<tr>
<td>HCBK</td>
<td>Hash Commitment Before Knowledge</td>
</tr>
<tr>
<td>BAN</td>
<td>Body Area Network</td>
</tr>
<tr>
<td>WSN</td>
<td>Wireless Sensor Network</td>
</tr>
<tr>
<td>ECDH</td>
<td>Elliptic-Curve Diffie-Hellman</td>
</tr>
<tr>
<td>SHCBK</td>
<td>Symmetric Hash Commitment Before Knowledge</td>
</tr>
<tr>
<td>VLC</td>
<td>Visible Light Communication</td>
</tr>
<tr>
<td>USB</td>
<td>Universal Serial Bus</td>
</tr>
<tr>
<td>NS</td>
<td>No Spoofing</td>
</tr>
<tr>
<td>NB</td>
<td>No Blocking</td>
</tr>
<tr>
<td>QR code</td>
<td>Quick Response code</td>
</tr>
<tr>
<td>SiB</td>
<td>Seeing is Believing</td>
</tr>
<tr>
<td>OOK</td>
<td>On-Off Keying</td>
</tr>
<tr>
<td>MITM</td>
<td>Man In The Middle</td>
</tr>
<tr>
<td>DoS</td>
<td>Deny of Service</td>
</tr>
<tr>
<td>ECI</td>
<td>Extended Channel Interpretation</td>
</tr>
</tbody>
</table>
1 Introduction

As communications and information technology continue to evolve, short-range wireless communication technology is becoming more mature. The general sense is that as long as the communication transceiver transmits information via radio waves, both sides and the transmission distance are limited to the shorter range (tens of meters) or less, that this can be called short-range wireless communications. Short-range wireless communications, for example IrDA, Bluetooth and 802.11 (Wi-Fi), assist people to avoid physical connections, thus making people’s lives more convenient.

Currently, more and more short-range wireless communication products are appearing in daily life, such as transmitting the data between two smart phones via Bluetooth and, additionally, the information being transmitted among the body sensors.

Since sensitive private information is collected or transmitted in these products, their security becomes issues of concern. Many researches have focused on using an extra channel, with characteristics that are different, namely the Dolev-Yao channel. On this extra channel, the researchers proposed that attackers will be unable to intercept, modify or produce fake messages.

1.1 Background and problem motivation

In the past few years, many key distribution protocols for wireless sensor networks have been designed. The majority are based on pre-deployment key schemes. Generally, these pre-deployment key schemes are divided into two phases.

1. A secrets pre-deployment phase: before the sensor nodes are deployed, some secrets are stored within the nodes.

2. Symmetric keys establishment phase: after the nodes are deployed, nodes can use the secrets to establish the symmetric keys.

However, these key management schemes also have shortcomings. For example, when the sensor nodes are compromised, it is difficult to update pre-deployed secrets.
Recently, key distribution schemes using asymmetric cryptography and out of band (OOB) channels have been proposed.

1. Many studies have demonstrated that asymmetric cryptography is feasible in sensor networks.

2. The OOB channel is an auxiliary physically authenticated channel, which can provide data origin authenticity, data integrity and freshness.

If asymmetric cryptography and OOB channels are integrated, the keys can be securely distributed or updated after deployment and the problem, previously mentioned, can be solved.

Typically, the protocols based on OOB channels work as follows. The devices exchange public information, i.e., public key, ID, over the normal channel. When all the messages have been sent and received, the devices will compute a cryptographic fingerprint, independently. Finally, the OOB channel is used to transfer fingerprints that are used to verify the authenticity of the public key(s).

However, since most mobile devices have different interfaces, finding suitable OOB channels is important.

1.2 Overall aim
This thesis concentrates on bootstrapping ad hoc networks using visual channels (QR code-camera channel and LED-Camera channel) that are able to reduce the human burden in addition to achieving the desired level of security.

An attempt is also made, in this thesis, to implement the SiB protocol and HLC-HCBK protocol.

In particular, their usability and security are evaluated.

1.3 Scope
This thesis concentrates on bootstrapping ad hoc networks using visual channels. These protocols provide a symmetric key after being bootstrapped.
Device Pairing Using Visible Light Communications
Shangyuan Guo

1 Introduction

Generally, consideration is not given to the mobile device being compromised as, if this is the case, the entire protocol will be compromised. Fake mobile devices are also not considered as people are generally familiar with their equipment and thus fake mobile devices can be recognized.

1.4 Concrete and verifiable goals
The goals of this thesis are listed as follows:

- Study the OOB channels, and related protocols. Summarize the theories and technologies with regards to previous research. Analyze the strengths and weakness for each protocol and technology.
- Design the visual channels; make them easy to use.
- Integrate the visual channels into SiB protocol and HLC-HCBK protocol.
- Compare the performances of visual channels; and analyze their usability and security.

1.5 Outline
The structure of this thesis is briefly described as below:

Chapter 2 offers an overview of the theoretical aspects related to Body Area Network, OOB channels, visible light communication and key distribution.

Chapter 3 explains the notations and the preliminary knowledge. It also discusses the factors which have an effect within the protocol.

Chapter 4 describes how to implement the visual channel using a Quick Response code. It will also introduce a SiB protocol. Finally, there will be an introduction to the manner in which the protocol is to be implemented.

Chapter 5 describes how to implement the visual channel with LEDs. It will introduce how to establish human controlled LED-Camera channels and the HLC-HCBK protocol. Finally, an implementation of the protocol will be provided.
Chapter 6 shows the results, and discusses the experiences which occurred during the whole project.

Chapter 7 analyze the security of these protocols, and discusses some possible attacks on these protocols.

Chapter 8 draws some conclusions with regards to this project, provides a summary of the whole project and also makes some suggestions on the future work.

1.6 Contributions
The contributions are as follows.

Human controlled LED-Camera OOB channels are used instead of human interactive OOB channels. Since LEDs are difficult to be read by human users, image and video processing technologies are used. Using these techniques, computers and mobile phones can automatically recognize LEDs flashes and compare them. In this way, the human burden is reduced.
2 Related work

In this chapter, some related works in this project are introduced, including body area network, BAN characteristics, OOB-based security protocols, visible light communication, manual method, key distribution and the HCBK protocol.

2.1 Body Area Network

As shown in Figure 1. Body Area Network (BAN) is a network attached to a human and which consists of a set of sensor nodes and a central coordinator.

- Sensors: worn on the body.
- Central coordinator: gateway between the BAN and the external network.

BANs provide convenience for chronic patients. It can collect the sensor data of patients in real-time and can also save this information for further investigation [7].

Figure. 1 Body Area Network
2.2 BAN Characteristics

BANs are different to wireless sensor networks (WSNs). Some of the differences are shown below:

- **Group size:** BAN sensors are worn on the body or inside clothes, thus the BAN does not have redundant sensors to handle failures.

- **Density:** the density of BAN sensor nodes is relatively low. The WSN is usually deployed in unattended areas, it requires more nodes to be deployed in order to substitute for any failed node.

- **Data rate:** WSNs are used for the monitoring of events. The BANs are commonly used for recording physiological activity and behaviour, thus data streams exhibit a relatively steady rate.

- **Mobility:** BAN nodes are attached to the human body, thus it has mobility to some extent, while the WSN nodes are usually considered as being stationary [7].

2.3 Visible Light Communication

Visible light communication (VLC) is an emerging wireless communication technology.

In the VLC system, LED has the dual role of communication and lighting. This is because the modulation rate is very high, thus a human is unable to completely feel it. The advantages of LED are low power consumption, long lifetime, small size, environmental protection and high response sensitivity [21, 22]. It can be used for ultra-high-speed data communications.

Compared with the radio frequency communications, VLC has the following advantages [21, 22]:

- A visible light on the human body is relatively safe, thus causing no harm. VLC system is mainly used indoors with LED lights used to transmit data and with only a minor amount of radiation on the human body.

- LEDs are used everywhere. Almost every place in life has lighting, so the lighting for communication can be installed anywhere and it can be more convenient for wireless data transmission.
Transmit power can be high. Compared to the infrared communication, the infrared communication can cause greater damage to the human eye; it must suppress the transmission power to a low level and thus the system performance will be severely limited. For radio communications, the RF signal can cause relatively large damage to the human body, so it must also limit the power. VLC is the visible light transmission of information; therefore, the power can be relatively high.

- Does not require certification of the radio spectrum.

- No electromagnetic interference. This advantage allows visible light communication to be used in hospitals, aircraft, etc...

Thus VLC technology has great prospects. It has attracted many researchers.

### 2.4 Manual method

“Manual authentication” [9] was proposed by C. Gehrmann and K. Nyberg, in this method, people are required type the short stings (about 18 bits) display on one device onto the other device or compare short strings displayed on the devices. There are many applications using these methods, such as Mobile phone connections, ATM cash machines and some Bluetooth devices. For the majority of these the user has to remember a 4-digit number. Although this method can prevent MITM attack, it is vulnerable to offline guessing attacks, and the other weakness for this method is that the PIN has to be kept secret.

### 2.5 Key Distribution

In sensor network, both symmetric and asymmetric cryptography based key distribution techniques have been studied by many researchers. Most symmetric key distribution schemes in sensor networks require pre-deployment secrets. Keys could be deployed deterministically, such as, a single master key or a pair-wise key [19]. There are also many non-deterministic key pre-deployment techniques, such as, key distribution schemes in [3, 6, 8]. As has been mentioned in the previous section, the pre-deployment schemes are not convenient.

Asymmetric cryptography based key distribution schemes have also been developed [10, 16]. They create a pair of public and private keys,
and use them to initiate secure connections between nodes. However, there is no trusted third party and it is difficult to exchange public keys securely.

2.6 **OOB-based security protocols**

OOB channel based security protocols have recently been used for distributing keys and IDs in sensor networks. McCune et al. propose the “Seeing-is-Believing” (SiB) protocol [15]. SiB can build two unidirectional visual device-to-device channels, which use one device to encode the data into a barcode and use the other device with a photo camera to read it. Huang et al. [11, 12] proposed an ECDH-SHCBK protocol and an ECDH-HCBK protocol. Li et al. Proposed a secure sensor association and key management scheme for body area networks [14], which required users to compare synchronized LED flashing patterns. Saxena and Uddin [24] proposed a visual channel that utilizes LEDs on sensor nodes, in which they attempt to make a good quality interfaces as well as the corresponding receivers on devices. However, they have not optimized protocols for sensor nodes.

General OOB channel based security protocols are researched by Balfanz et al. [1], Creese et al. [4], Gehrmann et al [8], Vaudenay [25], Cagalj et al. [2], Wong and Stajano [26] and Roscoe and Nguyen[17,18]. In addition, there are more human interactive security protocols that can be found in survey paper [18].

2.7 **HCBK protocol**

Bill Roscoe designed The Hash Commitment Before Knowledge (HCBK) protocol [23] that aims to bootstrap security in ad hoc networks. This protocol mainly works on these mobile devices which have no pre-shared information. It can achieve a high level of authentication by means of the use of a combination of the Dolev-Yao channel and OOB channel. The Dolev-Yao channel can be used to transmit the public keys, ID or other information [5]. The OOB channel is used for authentication and message integrity, which means that messages exchanged in the OOB channel cannot be modified or otherwise tampered with. Therefore, this approach can secure against all possible MITM attacks [13].

The figure 2 shows the two-party graphical representation of the HCHK protocol. N means the normal Dolev-Yao channel and OOB means the authentication channel.
Figure 2. Graphical representation of the HCHK protocol
3 Methodology

In this chapter, some knowledge to be used in this project will be introduced, which includes network model and design requirements, channels and attack models, commitment scheme and digest function, human factors and technological factors. First of all, some frequently used notations are listed in Table 1.

**Table 1. Notations**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A, B</td>
<td>Device A, Device B</td>
</tr>
<tr>
<td>( H() )</td>
<td>Hash function</td>
</tr>
<tr>
<td>( H )</td>
<td>Hash value</td>
</tr>
<tr>
<td>( \mathcal{D}() )</td>
<td>Digest function</td>
</tr>
<tr>
<td>( \text{Digest} )</td>
<td>Digest value</td>
</tr>
<tr>
<td>→</td>
<td>Wireless channels</td>
</tr>
<tr>
<td>⇒</td>
<td>OOB channels</td>
</tr>
<tr>
<td>⇒HLC</td>
<td>Human controlled LED-Camera channels</td>
</tr>
<tr>
<td>( G )</td>
<td>Group size</td>
</tr>
<tr>
<td>( pk_i )</td>
<td>Public key</td>
</tr>
<tr>
<td>( ID_i )</td>
<td>Sensor ID</td>
</tr>
<tr>
<td>( \text{Nonce} )</td>
<td>A long random number</td>
</tr>
<tr>
<td>( \text{commit}_i )</td>
<td>( H(\text{nonce}, \text{Sink}) )</td>
</tr>
<tr>
<td>( E_{pk_i}(shk_i) )</td>
<td>( shk_i ) is encrypted using key ( pk_i )</td>
</tr>
<tr>
<td>( shk_i )</td>
<td>Authentic secret key</td>
</tr>
</tbody>
</table>
3.1 Channels and Attack Model

Apart from the wireless communication channels among nodes, there are some OOB channels in small scale sensor networks. Examples could include USB connections and irDA.

In this project, the Dolev-Yao threat model [5] was used as the attack model for wireless channels. In the Dolev-Yao model, the attacker controls the wireless communication channels: he/she can obtain and modify any messages which are transmitted over these channels; he/she can initiate a conversation with any other user. However, the attacker is computationally bounded.

However, it is inappropriate to use the Dolev-Yao model for some OOB channels. One example is the USB connection. If the nodes connected via USB connection are trustworthy, it is impossible for the attacker to obtain or modify the messages over the USB connection; the attacker also cannot initiate a conversation with the devices. As another example, if the user can setup the irDA channel carefully, the attacker cannot obtain messages, modify messages or initiate a fake conversation. From these examples, it can be clearly seen that the Dolev-Yao model is not always suitable for OOB channels. In order to model the OOB channels, the following attack models have been adopted [4].

- No Spoofing Channel (NS): the attacker cannot spoof messages on this channel.
- No Blocking Channel (NB): The attacker cannot block messages on the channel. Therefore, session hijacking, redirecting and deny-of-service attacks are impossible.

The channels and attack models are summarized in Table 2. The last four rows are the capabilities of the attackers.
Table 2. Channels and attack models

<table>
<thead>
<tr>
<th>Attack Models</th>
<th>Wireless channels</th>
<th>OOB channels</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Dolev-Yao model</td>
<td>NS</td>
</tr>
<tr>
<td>Over-hearing messages</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Initiate fake conversions</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Block messages</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Modify messages</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

3.2 Diffie-Hellman Key Exchanges

There are many insecurity factors during message transmission in a network. In order to make the network message transmission more secure, the Diffie-Hellman key exchange algorithm was designed for that purpose. The Diffie-Hellman key exchange algorithm is a secure protocol, which allows two parties that have no prior information of each other to create a shared secret key over an insecure communication channel and this secret key can be used to encrypt the message during the communications.

In 1976, Whifield Diffie and Martin Hellman published this secure protocol, which later became a very famous and typical theory in cryptography. This method, although initially only used as a communication protocol in order to establish a common key, lacks authentication capabilities. However, the theoretical basis caused a profound influence on later cryptographic methods. This includes the famous RSA, which implements public key cryptography using an asymmetric algorithm.

Diffie-Hellman key exchange theory is based on modular exponentiation and is difficult to reverse. The following are definitions of some parameters:
Device Pairing Using Visible Light Communications
Shangyuan Guo

Methodology

2014-03-19

\[ p : \text{large prime} \]
\[ g : \text{primitive root mod } p \]
\[ a : \text{a private key selected by A-side} \]
\[ b : \text{a private key selected by B-side} \]
\[ K : \text{secret key} \]
\[ mod : \text{modular} \]

Basic theory is as follows:

According to number theory:

\[ (g^a \mod p)^b \mod p = (g^b \mod p)^a \mod p \]

Therefore, communication can be achieved on both the A-side and B-side, respectively, to select a secret key \( a \) and a secret key \( b \), with public information \( p \) and \( g \), then:

A-side:

\[ A = g^a \mod p \]
\[ K = B^a \mod p \]

B-side:

\[ B = g^b \mod p \]
\[ K = A^b \mod p \]

Assuming that the A-side is an initiator of key exchange and that the communication channel can be modified by an attacker during the transmission of information, the entire key exchange processes are as follows:

<table>
<thead>
<tr>
<th>Step</th>
<th>A-side</th>
<th>Communication channel</th>
<th>B-side</th>
<th>Common information</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Select a private</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
key a.
Calculate:
\[ A = g^a \mod p \]

2
Send \( A, g, p \) to B-side

3
Select a private key a.
Calculate:
\[ B = g^b \mod p \]
Calculate:
\[ K = A^b \mod p \]

4
Send \( B, g, p \) to A-side

5
Calculate:
\[ K = B^a \mod p \]

In the step 5, both sides have the K that is the secret key after key exchange. This key is used to encrypt and decrypt at a later stage.

In an open environment, within the whole key exchange processes and where information could be attacked, this can only occur in the network connection. Therefore, assuming that the E-side can eavesdrop the communication between the A-side and B-side, the information that can be intercepted by the E-side only involves \( A, B, g, \) and \( p \). In the calculation of the secret key \( K \), it must have knowledge of the private key \( a \) or private key \( b \). For the E-side, it can only be based on \( A, g, p \) to obtain \( a \), or can only be based on \( B, g, p \) to obtain \( b \). However, if \( p \) is sufficiently large, theoretically, it is not able to calculate \( a \) or \( b \) with the limited time. For the majority of applications, the transmission of information has a time limit. Therefore, if the E-side cannot crack the
key within a limited time, the communication between the A-side and B-side is safe.

### 3.3 Quick Response code

QR code is a square pattern using only black and white. In three corners, a small finder pattern is printed. The three finder patterns are those used to assist in decoding software to locate, the user can shoot at any angle and the content can still be read correctly. QR code is a widely used two-dimensional code, with a fast decoding speed. It can store various types of information.

![Figure 3. The structure of the QR code.](image)

- Finder pattern and separator: this is used to locate the two-dimensional codes. For each QR code, their positions are fixed, but the size for each QR code may be different, these black and white rectangles block can be easily detected and used in image processing.

- Alignment patterns: depending on different sizes, the number of alignment patterns is different. Alignment patterns are mainly used for aligning the shape of the QR code, especially when the QR code, printed on the surface, is not flat or if an aberration occurs when shooting.

- Timing patterns: these are small black and white plaid like axes, which define the grid on the two-dimensional code.
• Format information: they represent the error correction levels of the two-dimensional code, into L (Low), M (Medium), Q (Quartile) and H (High).

• Data zone: it uses a binary grid of black and white to encode data. 8 grids can encode one byte.

• Version information: it represents a two-dimensional code version, for which there are 40 kinds of matrix versions (usually is black and white), from 21 * 21(version 1) to 177 * 177 (version 40).

• Error correction codewords: it is used to correct errors due to the two-dimensional code being damaged.

Brief encoding processes:

1. Data Analysis: determines the type of encoding character, in accordance with the corresponding character set conversion, into a symbolic character. Select the error correction level, under the same conditions, the higher the level of error correction, the less is the storage capacity of the data.

2. Data encoding: the data is converted to a bit stream. Every 8 bits are one codeword and all the codewords consist of a codeword sequence. In fact, knowledge of the codeword sequence means that it is possible to have knowledge of all the contests in the two-dimensional code.

<table>
<thead>
<tr>
<th>Table 3. QR code capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Numeric</td>
</tr>
<tr>
<td>Alphanumeric</td>
</tr>
<tr>
<td>Binary</td>
</tr>
<tr>
<td>Kanji</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Meaning</th>
<th>Indicator</th>
</tr>
</thead>
</table>
Data can be encoded in accordance with a pattern, therefore, decoding can be more efficient.

3. Error correction: the data codewords are broken up into several Reed-Solomon code blocks, according to the code blocks and error correction capability level for generating error correction codewords. The error correction codewords are added to the back of the codeword sequence and become a new sequence.

<table>
<thead>
<tr>
<th>Table 4. Error correction capability levels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error lever</td>
</tr>
<tr>
<td>-----------------</td>
</tr>
<tr>
<td>L</td>
</tr>
<tr>
<td>M</td>
</tr>
<tr>
<td>Q</td>
</tr>
<tr>
<td>H</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Numeric encoding</th>
<th>0001</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alphanumeric encoding</td>
<td>0010</td>
</tr>
<tr>
<td>Byte encoding</td>
<td>0100</td>
</tr>
<tr>
<td>Kanji encoding</td>
<td>1000</td>
</tr>
<tr>
<td>Chinese encoding</td>
<td>1101</td>
</tr>
<tr>
<td>Structured append</td>
<td>0011</td>
</tr>
<tr>
<td>FNC1</td>
<td>0101</td>
</tr>
<tr>
<td>End of message</td>
<td>0000</td>
</tr>
</tbody>
</table>
If the two-dimensional code specification and error correction capability level have been determined, then their capability and error correction capability can also be determined.

4. **Structure Final Message**: Under the conditions of the QR, the code specification is determined and it is then necessary to interleave the data blocks and error correction codewords, after which a list of versions and the number of remainder bits that must be added to the back of the final message string.

5. **Module placement in matrix**: add the finder patterns, separators, alignment patterns, timing patterns and dark module into the matrix.

6. **Data masking**: put the data and error correction bits that have been obtained from the previous step into the QR code matrix together with the required function patterns.

7. **Format and version information**: generate the format and version strings and place them into the correct locations in the QR code.

### 3.4 Digest Function and Hash Function

A digest function \( \Phi() \) takes an input as message \( M \) of an arbitrary length and produce an output which is a short message digest \( D \) of fixed length. \( D \) is usually 16 bits or 32 bits. In other words, if \( K \) is a key domain,

\[
\Phi : K \times M \rightarrow D
\]

A hash function \( \mathcal{H}() \) takes an input as a message \( M \) of arbitrary length and produces, as an output, a long message digest \( H \) of fixed length. \( H \) is usually longer than 160 bits. In other words, if \( K \) is a key domain,

\[
\mathcal{H} : K \times M \rightarrow H
\]

The differences between the digest function and hash function are shown in Table 5. In pre-image attacks, attackers have knowledge of both \( H \) and \( D \) and they attempt to find the original message \( M \) such that

\[
H = \mathcal{H}(M), \quad D = \Phi(M)
\]

In combinatorial attacks, attackers attempt to find one input \( M_2 \) (\( M_1 \) is given) or two inputs \( M_1 \) and \( M_2 \) (\( M_1 \neq M_2 \) in both cases) such that
Device Pairing Using Visible Light Communications
Shangyuan Guo

3 Methodology

\[ H(M_1) = H(M_2), \; D(M_1) = D(M_2) \]

The digest function is vulnerable to combinatorial attack, while the hash function can resist both attacks.

<table>
<thead>
<tr>
<th></th>
<th>( H() )</th>
<th>( D() )</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Output Length</strong></td>
<td>Long (160 bits or longer)</td>
<td>Short (16 or 32 bits)</td>
</tr>
<tr>
<td><strong>Preimage attack</strong></td>
<td>Good</td>
<td>Good</td>
</tr>
<tr>
<td><strong>resistance</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Combinatorial attack</strong></td>
<td>Good</td>
<td>Vulnerable</td>
</tr>
<tr>
<td><strong>resistance</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### 3.5 Human factors

In 2000, Bruce Schneier pointed out “security is only as good as its weakest link, and people are the weakest link in the chain.” Security not only involves mathematical proofs, but also involves other factors such as human factors. If the assumption is that the mathematical proofs are reliable and certainly correct, but that human users are not able to be trusted to process large amounts of data, it becomes necessary to ensure the desired level of security.

The following are some of the common errors which are easily made by humans. This includes people usually do not read the certificate, but giving authority to an untrusted Active X control, or that people will leave the ATM without getting back their card, when the money has come out. Some errors are also due to human physiology, including the special physiological structure of the human eye, where, if the frame is higher than about 10-12 frames per second, it will be thought of as being coherent [20], this phenomenon is called persistence of vision.
Therefore, minimizing the risk of these fatal errors by human users is also an important aspect. It can make the protocol more secure and more usable.

### 3.6 Technological factors

The technology can be used to reduce human workload if the machine is allowed to do more of the work instead of the human users. This is particularly true regarding work for which a desired level of security is required, as a machine is better able to provide this. This could involve encryption which is required to produce a large number of bits that makes the protocol more secure, because the attacker would thus require to spend more time to calculate a solution. However a larger number is not suitable for human users to read. If the machine is allowed to conduct this, then it becomes possible to avoid the risk of fatal errors which can be caused by human users.

In this thesis, large hash values are being made into QR code and making the digest value into an LED signal, then allowing the machine to read them and compare them, with human users only being required to push a button for confirmation. These actions are easy for human users.
4 Quick Response code based channel

In this chapter, the visual channel will be implemented using a Quick Response (QR) code. Firstly, how to establish QR-based channels will be introduced and secondly the SiB protocol using QR-based code channel will be dealt with. Finally, the system design and implementation will be shown.

4.1 QR-based channel establishment

QR-based channels (Figure 4) are one of the visual channels controlled by human users.

- Firstly, QR-based channels are established using QR-based code and cameras. These channels exist widely in mobile phones. Nowadays, more and more people use smart phones and almost all of these are equipped with cameras and screens. Thus it is very easy for people to use these functions to create QR-based channels.

- Secondly, the QR-based code channels are controlled by human users. Under the supervision of human users, it is extremely difficult for attackers to block or interference the messages transmitted via the QR-based channel. It is also difficult to initiate fake conversions. Thus the QR-based channels are NS and NB OOB channels.

The QR-based code channels are created as follows:
On the transmitter side, the data is encoded into the QR code and displayed on the mobile phone’s screen.

On the receiver side, the user uses the camera to take a snapshot of the transmit side’s screen, then the QR code recognition algorithm processes and decodes the image.

4.2 SiB protocol using QR-based channel

The protocol messages exchange procedure is listed as follows.

Protocol 1.

1. $A \rightarrow B$: $PK_a$

2. $B \rightarrow A$: $PK_b$

3. $A \Rightarrow B$: $H_a = \mathcal{H}(PK_a)$

4. $B \Rightarrow A$: $H_b = \mathcal{H}(PK_b)$

In this protocol 1, the message exchange procedure is explained as follows:

Round 1: device $A$ sends the Diffie-Hellman public key $PK_a$ to device $B$ over the wireless channel.

Round 2: device $B$ sends the Diffie-Hellman public key $PK_b$ to device $A$ over the wireless channel.

Round 3: device $A$ sends the hash value of $PK_a$, $H_a$, to device $B$ via the visual channel. Device $B$ verifies that $H_a = \mathcal{H}(PK_a)$. If the verification fails, the device aborts and informs the user.

Round 4: device $B$ sends the hash value of $PK_b$, $H_b$, to device $A$ via the visual channel. Device $A$ verifies that $H_b = \mathcal{H}(PK_b)$. If the verification fails, the device aborts and informs the user.

Round 5: if all the verifications are successful, the devices use the public keys to calculate the secret key.

System design This application has been built on the Galaxy Nexus, which is an Android smart phone. The operating system version is 4.2.1.
In this application, the QR code format and QR code reader are adapted from that of ZXing [27]. ZXing is an open source Java library for parsing multiple formats barcodes and 2D codes. For the hashing operation, the SHA-256 cryptographic hash function has been adapted. The ephemeral Diffie-Hellman key was also implemented in order to establish a secret key between the two mobile phones. Here is an example: Alice wants to communicate with Bob securely via the Bluetooth channel. As shown in the Figure. 5 The process is as follows:

1. Alice sends a request to Bob, asking for the establishment of a Bluetooth connection.

2. If Bob consents, the Bluetooth connection is established, but this is insecure. In order to establish a secure communication channel, Alice presses the “Send PKey” button to generate a key pair, and publishes the public key to Bob via the Bluetooth channel.

3. When Bob sees that he has received Alice’s public key from the mobile phone’s screen, Bob also presses the “Send PKey” button to generate a key pair. The public key is sent to Alice, the private key is retained.

After completing the above steps, public key exchange has been completed. The next steps are used to authenticate the public keys.

4. Alice presses the “QR” button and the public key received in step 3 will be hashed using SHA-256; hash output is then encoded into a QR code, and the QR code is shown on the Alice’s mobile phone’s screen.
5. Bob presses the “Scan” button, uses the digital camera to take a snapshot of Alice’s screen, and then decodes the value.

6. Meanwhile, Bob also uses the SHA-256 cryptographic hash function to encrypt his public key and then compares the two values. If the two values are the same, the verification is successful. Bob’s mobile phone will display “The verification is successful”. If the two values are not the same, the verification fails. Bob’s mobile phone will display “The verification fails, you should abort it!” Bob informs Alice to abort the bootstrapping.

7. If verification in step 4 is successful, Bob presses the “QR” button and the public key received in step 2 will be hashed using SHA-256; the hash output is then encoded into a QR code, and the QR code is shown on Bob’s mobile phone’s screen.

8. Alice presses the “Scan” button, uses the digital camera to take a snapshot of Bob’s screen, and decodes the value.

9. Meanwhile, Alice also uses the SHA-256 cryptographic hash function to encrypt his public key and compares the two values. If the two values are the same, the verification is successful, and Alice’s mobile phone will display “The verification is successful”. If the two values are not the same, the verification fails, Alice’s mobile phone will display “The verification fails, you should abort it!” Alice informs Bob to abort the bootstrapping.

Now, this security protocol is finished. Alice and Bob can use this authenticated channel to communicate securely.
5 Human controlled LED-Camera channel

In this chapter the visual channel will be implemented using LEDs. Firstly, how to establish Human controlled LED-Camera (HLC) channels will be stated after which the HLC-HCBK protocol will be introduced. Finally, the protocol will be implemented and the system design shown.

5.1 HLC Channel establishment

<table>
<thead>
<tr>
<th>Sender</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data → Encoding → Modulation → LED</td>
</tr>
</tbody>
</table>

\[ \downarrow \]

<table>
<thead>
<tr>
<th>Receiver</th>
</tr>
</thead>
<tbody>
<tr>
<td>Camera → Demodulation → Decoding → Data</td>
</tr>
</tbody>
</table>

**Figure 6.** HLC Channel

LED-camera channels exist widely in small scale sensor networks. On the one hand, many sensor nodes are equipped with LEDs, for example, the widely used sensor nodes: Tmote Sky nodes, MICAz nodes and MICA2 nodes all have several LEDs. On the other hand, almost all current smart phones and personal computers are equipped with cameras. Therefore, these components can be used to create HLC OOB channels.

HLC channels are controlled by human users. Firstly, with the assistance of human users, it is extremely difficult for attackers to block the messages transmitted over the HLC channel. Secondly, the human users can perceive any interference when the messages are transmitting via the HLC channel, thus it is also difficult to change these messages. Thirdly, since the communication parties are under control and any communication interference can be perceived, the attackers cannot
5 Human controlled LED-Camera channel
initiate fake conversions. Therefore, the HLC channels are NS and NB OOB channels.

5.1.1 On the transmitter side
The visual channels are created as in Figure 6. Data is firstly encoded into a message frame. In the modulation block, the encoded frame is used to modulate the optical output. The optical output will be transmitted to the camera of the receiver. The demodulation block in the receiver demodulates the optical signal. The demodulated signal will be decoded by the data decoding block.

<table>
<thead>
<tr>
<th>Header</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preamble</td>
<td>Delimiter</td>
</tr>
</tbody>
</table>

*Figure 7. Message frame structure*

The message frame consists of a header block and a data block. The frame structure is shown in Figure 7. The first part of the header is a preamble and this is 1010. It is used by the recipient to synchronize the clock. A delimiter is 11, which is used to denote the end of the preamble. After the delimiter, the sender ID (5 bits) and the packet type PktType (1 bit) are transmitted. After the header, 16-bit data is transmitted.

The modulation of the optical link calls On-Off Keying (OOK). More specifically, the light of a LED can be used as a carrier wave and the digital data can be represented as the presence or absence of the carrier wave.

5.1.2 On the receiver side
LED signals can be automatically recognized by a fixed camera. This camera can continue to take the image with the LED lights and the images are processed to determine the locations of the LEDs.

In this project, three methods to recognize the locations of LEDs will be introduced.

**Recognition based on Frame**
This method is based on the property that the background of frames remains unchanged but the brightness of LEDs in the frames are
different. Therefore, if two images are used, one image when the LEDs are off and the other when the LEDs are on, it is easy to determine all the locations of the LEDs.

Firstly, the camera takes a picture when all LEDs are off, say $I_f$; and it takes a picture when all LEDs are on, say $I_o$.

Secondly, the locations $L$ of the LEDs can be detected using $I_o - I_f$.

The LED recognition example is shown figure 8.

Figure 8. (A) Original picture; (B) LEDs are on (C) LEDs are off (D) After subtraction; (E) Locations of LEDs
The algorithm is shown in Algorithm 1.

**Algorithm 1 LED location detection process**

INPUT: $I_o$ and $I_f$  
OUTPUT: $L$  

1. $I = I_o - I_f$  
2. Convert image $I$ to binary image $I_{bw}$  
3. Applies morphological operations to the binary image $I_{bw}$  
4. Label connected components in $I_{bw}$  
5. Find the centre of mass of the regions in each connected component in $I_{bw}$  
6. Store the locations of the centres in $L$  
7. Return $L$

**Recognition based on Colour**

This method is based on the property that colour images of an LED have high absolute values in the red component of the RGB coordinates. A simple threshold-based method can distinguish the locations of the LEDs.

The first step is the separation of the red component. There are three components in an image; in this method only the red component is required, because LED has high values in the red component.

The second step: set a threshold in the red component, for example 240.

The third step: create a disk-shaped structuring element. Then this element is used to erode the image obtained in the second step.

Finally, the locations of the LEDs in the image can be determined.

The LED recognition example is shown figure 9.
The algorithm is shown in Algorithm 2.

**Algorithm 2 LED location detection process**

**INPUT:** $I$

**OUTPUT:** $L$

1. Get the red component $R$ from the RGB image $I$
2. Set the threshold value in $R$, get $R_{th}$
3. Applies morphological operations to the binary image $R_{th}$
4. Label connected components in $R_{th}$
5. Find the centre of mass of the regions in each connected component in $R_{th}$
6. Store the locations of the centres in $L$
7. Return $L$
Matching by correlation

This method is based on the property that the LEDs in the image have the same shape. Using this property, a template can be used to find all the places that match this given template. This is called the correlation problem.

In this method, each image must be processed, say \( f(x,y) \), the template \( T \) of LED, say \( t(s,w) \). The normalized correlation coefficient is used to calculate the correlation.

**Formula 1.**

\[
\delta(x,y) = \frac{\sum_s \sum_w [t(s,w) - \bar{t}] \sum_s \sum_w [f(x+s,y+w) - \bar{f}(x+s,y+w)]}{\left( \sum_s \sum_w [t(s,w) - \bar{t}]^2 \sum_s \sum_w [f(x+s,y+w) - \bar{f}(x+s,y+w)]^2 \right)^{\frac{1}{2}}}
\]

In formula 1, \( t \) means the average value of the template, \( f() \) means the average value that at \( f \) region coincides with \( t \). By using this formula it is possible to obtain the value \( \delta \), which is between -1 and 1, and when \( t \) matched the LED in the \( f \), the value is close to 1. A threshold is set as the matching criteria for success. Finally, the locations \( L \) of the LEDs in the image can be obtained.

The LED recognition example is shown figure 10.
The algorithm is shown in Algorithm 3.

**Algorithm 3 LED location detection process**

INPUT: $T$ and $I$

OUTPUT: $L$

1. Convert template $T$ and image $I$ to gray image $T_{gy}$ and $I_{gy}$
2. Use $T_{gy}$ to match the gray image $I_{gy}$ get a correlation image $I_{cor}$
3. Scales the intensities of image $I_{cor}$ to full 8-bit image $I_{cor'}$
4. Label connected components in $I_{cor'}$
5. Find the centre of mass of the regions in each connected component in $I_{cor'}$
6. Store the locations of the centres in $L$
7. Return $L$

When the locations of the LEDs have been determined, a value is set to determine whether the LEDs are on or off in the images. Then this value is used to compare the values of the pixels located at the locations of LEDs. If the value is greater than the values used as the comparison, then the LED is on, otherwise, it LED is off.

The characteristics of the HLC channels are summarized in Table 6. Compared to those of wireless channels, their transmission capabilities are limited. The maximum data rate of the visual channel is highly dependent on the maximum flashing speed of the LEDs and the maximum sample rate of the cameras. In addition, the senders and receivers are required to be in a certain spatial relationship (Line-of-Sight) in order to establish communication. In addition, the HLC
channels are controlled by human users. Furthermore, NS and NB are the attack models of HLC channels.

Table 6. HLC channel and IEEE 802.15.4 wireless channel

<table>
<thead>
<tr>
<th></th>
<th>Wireless channels</th>
<th>HLC channel</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data Rate</strong></td>
<td>Fast</td>
<td>Slow</td>
</tr>
<tr>
<td><strong>Line-of-Sight</strong></td>
<td>Not require</td>
<td>Require</td>
</tr>
<tr>
<td><strong>Human Control</strong></td>
<td>Not require</td>
<td>Require</td>
</tr>
<tr>
<td><strong>Attack Models</strong></td>
<td>Dolev-Yao</td>
<td>NS and NB</td>
</tr>
</tbody>
</table>

5.2 HLC-HCBK Protocol

The protocol messages exchange procedure is listed as follows.

Protocol 2.

1. $\forall_{Sensor_i} \rightarrow Sink : ID_i, pk_i$

2. $Sink \rightarrow \forall_{Sensor_i} : Sink, E_{pk_i}(shk_i), commit_i = H(nonce, Sink)$

3. $Sink \rightarrow \forall_{Sensor_i} : nonce$

4. $\forall_{Sensor_i} \Rightarrow_{HLC} Sink : ID_i, digest_i = D(nonce, Sink, ID_i, E_{pk_i}(shk_i), pk_i)$

In the HLC-HCBK, The message exchange procedure is explained as follows:

Round 1:

Step 1: Each sensor node $Sensor_i$ sends its $ID_i$ and public key $pk_i$ to sink via the wireless channels.
Device Pairing Using Visible Light Communications

Shangyuan Guo

5 Human controlled LED-Camera channel

2014-03-19

Action 1: (1) User enters the group size $G$ into Sink through the keypad on the computers or cell phones that are physically connected to the Sink. Sink verifies whether the number of communication devices is equal to $G$. If the number of communication devices does not equal $G$, then Sink aborts the protocol and sends a failure notification to the user. (2) Sink generates two values that are a commitment/opening pair: one is a long random number $nonce$, and another one is commitment $commit_i = H(nonce, Sink)$.

Round 2:

Step 2: Sink sends each sensor node its ID $Sink$, $E_{pk_i}(shk_i)$ and the commitment $commit_i$ via wireless channels.

Action 2: When each sensor node has finished step 1 and step 2, they flash LEDs. The user verifies that each sensor node has flashed and the number of communication devices is $G$. Then the user presses the sink button to release the opening of the message in the step 3 when the verification is successful.

Round 3:

Step 3: Sink sends the opening message $nonce$ to each sensor node via wireless channels.

Action 3: (1) each sensor node calculates $H(nonce, Sink)$, and verifies that $H(nonce, Sink) = commit_i$. If the verification fails, the sensor aborts and informs the user. (2) Sensor calculates $digest_i = D(nonce, Sink, ID_i, E_{pk_i}(shk_i), pk_i)$

Round 4:

Step 4: each sensor node sends its ID $ID_i$ and digest value $digest_i$ to Sink via the HLC channel.

Action 4: Sink verifies that $digest_i = D(nonce, Sink, ID_i, E_{pk_i}(shk_i), pk_i)$. If the verification fails, Sink aborts the protocol and informs the user. If all the verifications are successful from the step 1 to step 4, then each sensor node holds an authentic secret key $shk_i$ with Sink.
5.3 System design

![Diagram of system design](image)

**Figure 11. Example of HLC-HCBK protocol**

A Laptop with a Microsoft LifeCam VX-1000 is used to simulate the mobile phone and red LEDs in Tmote sky nodes as the sender. In order to distribute the symmetric keys securely between the laptop and the Tmote sky nodes, as shown in the figure 11, the system design is as follows:

1. The user presses the button on the Tmote sky nodes and then the nodes send their public keys to the laptop. When the user sees that all keys have been received, the user confirms the number of nodes and inputs the number of nodes into the laptop. If the user discovers that the number shown in the laptop does not match the number of nodes, the user must reset the bootstrapping.

2. The laptop sends Encrypted secret keys and commitments to the nodes. When the nodes receive the message, they flash the LEDs, at
this time, the user is required to verify whether the number of nodes is correct. If the number is correct, it will continue to step 3.

3. The user presses the button on the laptop to release the opening of the message. The nodes will verify the message, if the verification is successful, it will continue to step 4 and, if not, the nodes send a notification to the user; the user must reset the bootstrapping.

4. Nodes flash the LEDs, the user must use the camera to shoot with the nodes. If the verification is successful, and then the laptop displays the result, the verification is complete and if not the verification has failed. The user must reset the bootstrapping.

After completing all the steps, this security bootstrapping is finished. Laptop and sensor nodes can use this authenticated channel to communicate securely.
6 Results

In this chapter the demonstrations will be displayed and the performance of three LED location detection process methods will be compared.

6.1 Demonstration for SiB protocol

The SiB protocol demonstration is shown as follows:

Figure 12 displays the main user interface of the application.

![Main user interface](image_url)

**Figure 12.** Main user interface

Figure 13: when the connection has been built, users exchange the public key with each other. In the screen, it will display “public key has been sent”, when the public key is sent. The screen will display “Received the public key.”, when the public key has been received.
Figure 13 Key exchange

Figure 14: after the public key exchange steps, user A presses the “QR” button, and user B presses the “Scan” button on another mobile phone. User B can use the camera to scan the QR code.
Figure 14. Scan QR code

Figure 15: if the verification is successful, the screen displays “the verification is successful”.
6.2 HLC Channel and Human Burden

The Microsoft LifeCam VX-1000 is used as the camera and red LEDs in three Tmote sky nodes as the sender. The LED recognition example is shown in Figure 16. The example recognition result is 101011000111010101010101010.

Figure 15. Verification success

Figure 16. Demo
Several recognition tests have been tested using Matlab version 7.3 installed in a laptop (Intel Core 2 Duo processor T7500 2.2GHz, 2GB DDR2 RAM, Windows 7 Home Premium). These tests are as follows. The signals from the red LEDs are recognized in three Tmote sky nodes. The signals from the three LEDs are asynchronous and different. The blinking frequency is 2 blinks per second; all 10 tests can be successfully recognized. The successful recognition rate of signals from one LED is tested and it is determined whether the flashing frequency is 2 blinks per second; all the 50 random 28 bits string can be successfully recognized. However, the blinking frequency is 4 flashes per second; the application cannot correctly recognize the strings due to the sensor nodes’ clock drift. The recognition speed of the signals from one LED is tested as to whether the blinking frequency is 2 flashes per second; the average running time for recognition is less than 10 seconds. In this case, the LED blinking time is about 14 seconds. The distance between the camera and LEDs is tested for a range we tested from 30 cm to 60 cm and within this range, all LEDs can be recognized. The size of the images is tested and, the smaller the size, the faster it computes.

**HLC VS Blink-Blink**

Blink-Blink: In 2008, Saxena and Prasad [19] display their “Blink-Blink” (BB) pairing method; they use two devices to run a synchronized LED blinking pattern. In 2012, Ming L, Shucheng Y, Joshua.D.G, Wenjing L and Kui R, also use the synchronized LED blinking method, the blinking frequency is 400 milliseconds (around 2 flashes per second).

HLC: LED blinking is not synchronized, the number of sensor nodes can be random, and the blinking frequency is 2 flashes per second.

The human burden using HLC channels are shown in Table 7. Human users are not required to compare the status of LEDs in m nodes for 16 times. However, in all these protocols, the users are required to handle the group sizes and prevent physical attacks.

**Table 7. Human burden**

<table>
<thead>
<tr>
<th>( \text{HLC-HCBK} )</th>
<th>( \text{ECDH-SHCBK} )</th>
<th>( \text{ECDH-HCBK} )</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Digest comparison</strong></td>
<td>No</td>
<td>16( \times m ) times</td>
</tr>
</tbody>
</table>
### Table 1: Group size constraint

<table>
<thead>
<tr>
<th>Input once</th>
<th>Count Once</th>
<th>Count Once</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Prevent physical attacks</td>
</tr>
</tbody>
</table>

6.3 **Performance for the LED location detection process**

All these three methods can successfully recognize the LEDs.

![Graph showing performance comparison](image)

**Figure 17.** Comparison of the LED location detection process methods

A comparison of the LED location detection process methods can be seen in Figure 17.

Frame-based, its processing speed is moderate, no restrictions on background, but the preprocessing step is more than for the other methods.

Color-based is the fastest, but it requires a very strict background.

Correlation-based provides the best results and does not require a strict background, but it is the slowest.
7 Security analysis

In this chapter, the security tests and the other security considerations will be discussed.

7.1 Man in the middle

Man In The Middle (MITM) attack refers to an attacker who can independently contact both ends of the communication. The attacker can exchange data it receives, so that both ends of the communication think they are connected via a direct dialogue with each other, but in fact the attacker controls the entire session. In the MITM attack, the attacker can intercept communications and insert new content.

7.1.1 MITM is eliminated in SiB protocol

SiB protocol adopts a Diffie-Hellman key Exchange algorithm and, as known, the Diffie-Hellman key Exchange algorithm can be attacked by MITM attack. The method is as follows:

Fistly, \( p \) is set as a large prime, \( g \) is the primitive element of \( p \), \( a \) is the private key of device A, \( b \) is the private key of device B, \( m \) a is the private key of attacker. \( g^a(\text{mod } p) \) is the public key of device A, \( g^b(\text{mod } p) \) is the public key of device B, \( g^m(\text{mod } p) \) is the public key of attacker.

The MITM adversary must catch \( g^a(\text{mod } p) \), which is the public key of device A that is sent to device B. and then the adversary sends its own public key \( g^m(\text{mod } p) \) to device B instead of \( g^a(\text{mod } p) \).

Likewise, the MITM adversary must also catch \( g^b(\text{mod } p) \), which is the public key of device B that is sent to device A. The adversary then sends its own public key \( g^m(\text{mod } p) \) to device A instead of \( g^b(\text{mod } p) \).

After that, the adversary can obtain two keys: one is \( DH_{\text{keyA}} = (g^m)^a(\text{mod } p) \) and the other is \( DH_{\text{keyB}} = (g^a)^m(\text{mod } p) \). Device A can obtain the key: \( DH_{\text{keyA}} = (g^m)^a(\text{mod } p) \) and device B can obtain the key: \( DH_{\text{keyB}} = (g^a)^m(\text{mod } p) \).

In this way, the adversary can eavesdrop or modify the messages. Note that the adversary must inject its own public key in the key exchange.
If it is assumed that there is a MITM attack in the SiB protocol:

In round 1 and round 2, the adversary has successfully injected its own public key \( PK_m \).

In round 3 and round 4, the protocol uses the QR-based channel, as mentioned previously, the QR-based channel is the NS and the NB channel. Therefore the adversary cannot modify the messages in these rounds.

In round 3, device A sends \( H_a = H(PK_a) \) to device B, then device B will calculate and find \( H_a \neq H(PK_a) \).

In round 4, device B sends \( H_b = H(PK_b) \) to device A, then device A will calculate and find \( H_b \neq H(PK_a) \).

Therefore, the user is able to notice the attack and thus abort the bootstrapping in round 3.

7.1.2 MITM is eliminated in HLC-HCBK protocol

Assume that there is a MITM attack. In the MITM attack, the security analysis is as follows.

1. Because the HLC channels are NS and NB channels, the message in step 4 cannot thus be changed by the attacker.

2. If any message \( nonce \) in step 3 is changed by the attacker. Assume that the message has been changed into \( nonce' \). There are two situations.

   If the \( commit \) in step 2 is not changed, in action 3, sensor will find the \( H(nonce', Sink) \neq commit \): \( H(nonce, Sink) \), therefore this attack is negligible.

   If the \( commit \) in step 2 is changed, but the attacker does not know the final digest, it is therefore unable to find a suitable \( nonce' \) to match the digest value in step 2. In e step 4, the attacker cannot thus change the message, therefore, in action 4, Sink will discover \( D(nonce', Sink, ID_v, E_{pk_i}(shk_i'), pk_i) \neq D(nonce, Sink, ID_v, E_{pk_i}(shk_i), pk_i) \), therefore this attack is negligible.
3. If the attacker changes messages in step 1 or step 2 without changing the message in step 3. There are two situations.

If the \textit{commit} is changed by the attacker, in action 3, the sensor will discover the $H(\text{nonce}', \text{Sink}) \neq H(\text{nonce}, \text{Sink})$, therefore this attack is negligible.

If the \textit{commit} is not changed by the attacker, since the attacker does not know the final digest, it cannot find a suitable $shk'$ to match the digest value. Thus in step 4, the attacker cannot change the message, therefore, in the action 4, Sink will discover $\mathcal{D}(\text{nonce}, \text{Sink}, ID_i, pk_i, E_{pk_i'}(shk_i')) \neq \mathcal{D}(\text{nonce}, \text{Sink}, ID_i, pk_i, E_{pk_i'}(shk_i'))$, therefore this attack is negligible.

### 7.2 Denial of Service

Deny of Service (Dos) is an attack that makes the devices or networks unable to provide normal services.

#### 7.2.1 DoS against SiB protocol

This protocol can detect DoS attacks. When the device is attacked by a DoS, the user can see there are many messages on the smart phone screen. In this situation, user should stop the bootstrapping.

If the attacker blocked the wireless channels; the device cannot receive a normal message. In order to prevent these attacks, timeout mechanisms should be used. If the bootstrapping is not finished in the pre-defined time period, the user should stop the bootstrapping.

Meanwhile, in rounds 3 and 4, it is difficult to block the QR-based channel.

#### 7.2.2 DoS against HLC-HCBK protocol

The protocol can prevent or detect DoS attacks. The protocol can detect the wireless channel DoS attacks. If messages transferred over wireless channels are blocked, the nodes cannot pass the checks of steps 1 and 2 due to a time out. If the user notices the channel has been blocked, it is then possible to stop the bootstrapping. Meanwhile, it is difficult to block the HLC channel.
7.3 Offline Pin Cracking
This attack uses some high-tech tools. At first, attackers use tools to scan nearby devices that are communicating. It then connects to them, using the PIN code from 0000 to 9999 to discover the correct PIN. However, both the SiB protocol and the HLC-HCBK protocol use asymmetric cryptography and the OOB channel together, thus public keys do not suffer from this PIN-based attack.

7.4 Other security considerations
Although these protocols can prevent many attacks, there are no absolutely secure protocols. Compromised sensor nodes can obtain the secret key, but cannot influence the authenticity of keys among other nodes. The sink node must be well protected; otherwise, the whole protocol is compromised.
8 Conclusions

To conclude, QR code-camera channels have been implemented as have HLC channels. There are several advantages of using visual channels (QR code-camera and LED-camera). The main advantage is that the human burden is reduced.

In this thesis, the SiB protocol and HLC-HCBK protocol have been designed and implemented. The main advantage is that the MITM attack is eliminated. The strengths and weaknesses of these protocols have been discussed.

In future work, the plan is to improve the LEDs blinking frequency and to make the HLC channel more usable.
References


