Quantum Inspired Security on a Mobile Phone

Yu Qin\textsuperscript{1} and Wanjiaman Li\textsuperscript{2}

\textsuperscript{1} Chongqing University of Education, Chongqing 400065, China
isdemiqin@outlook.com

\textsuperscript{2} University of Bristol, Bristol BS8 1TH, UK
liwanjiaman.1993.2012@my.bristol.ac.uk

Abstract. The widespread use of mobile electronic devices increases the complexities of mobile security. This paper aims to provide a secure communication environment for smartphone users. Some research proves that the one-time pad is one of the securest encryption methods, and the key distribution problem can be solved by using the QKD (quantum key distribution).

The objective of this project is to design an Android APP (application) to exchange several random keys between mobile phones. Inspired by QKD, the developed APP uses the quick response (QR) code as a carrier to dispatch large amounts of one-time keys. After evaluating the performance of APP, it allows the mobile phone to capture and decode 1800 bytes of random data in 600ms. The continuous scanning mode of APP is designed to improve the overall transmission performance and user experience, and the maximum transmission rate of this mode is around 2200 bytes/s. The omni-directional readability and error correction capability of QR code gives it better real-life application, and the features of adequate storage capacity and quick response optimize overall transmission efficiency. The security of this APP is guaranteed since QR code is exchanged face-to-face, eliminating the risk of being eavesdropped. Also, the id of QR code is the only message that would be transmitted through the whole communication.

The experimental results show this project can achieve superior transmission performance, and the correlation between the transmission rate of the system and several parameters, such as the QR code size, has been analyzed. In addition, some existing technologies and the main findings in the context of the project are summarized and critically compared in detail.

Keywords: One-time pad, QKD (quantum key distribution), QR Code.

1 Introduction

Over the last few decades, with the development of internet technologies, information sharing has become more convenient for people. In 2014, Facebook users shared information was almost two million times every minute, and the number of emails sent per minute over 200 million [1]. Today, mobile internet access is more convenient for users, bringing a dramatic increase in the amount of shared information per minute. However, such data explosion has posed a new level of complexity to information
security. The main aim of this project is to provide a secure way to protect the personal information of mobile phone users.

Current public-key cryptography relies on computational complexity. Since the large numbers can be factorized in polynomial time by using the quantum computer, this public key encryption method cannot provide a guaranteed security. Today, the secret-key cryptography is the only provably secure way to achieve absolute security, and the one-time pad key is used as a secret key between sender and receiver. Each key of one-time pad can be used only once to encrypt text, and both key and message should be equal in length, and thus it cause the key distribution problem. Recent research studies indicate that this problem can be elegantly solved by using quantum key distribution (QKD), which uses quantum mechanics to establish an absolute secure channel, thus ensuring that the security cannot be violated by the quantum computer.

However, the QKD is a new technology and the implementation of it is still complicated. Inspired by QKD, the main purpose of this project is to develop an Android application (APP) to exchange the one-time keys between two mobile phones using Quick Response (QR) codes, with JAVA as the main programming language. The camera and screen of the smartphone are used as part of the system’s hardware, and the key is exchanged face-to-face between users to avoid access by a third party. The QR code is a good choice for distributing the one-time pad keys between smartphones, because it has omni-directional readability, error correction ability, and large capacity to store keys. Also, it will simplify the process of promoting this APP to real-life applications, for example, online payments or ATM transactions.

The paper can be approximately summarized as follows: Section 1 presents a brief introduction to give readers a basic understanding of the project. After the main objective of the project has been explained, sections 2 summaries and critically compare existing works and relevant methods. Section 3 explains the working principle of this project and illustrates how to operate each function of the APP. In section 4, the transmission performance of the system is evaluated based on different parameters, such as the QR code size and the quantity of QR code, after which the best transmission result is obtained. The last section reviews and concludes all the useful aspects of the project and proposes some feasible improvement for future works.

2 Preliminaries and Problem Statement

2.1 Quantum Key Distribution

Quantum key distribution (QKD) is based on the quantum mechanics to provide a guaranteed security, and it is suggested to be used to solve the one-time pad key distribution problem. QKD can be implemented by applying prepare and measure protocols and entanglement based protocols, such as the BB84 protocol, the six-state protocol and the EPR protocol. The BB84 is a common protocol proposed by Bennett and Brassard in 1984, and the six-state protocol is a derivative version of BB84 protocol. The ERP protocol is proposed by Einstein, Podosky, and Rosen. Despite the quantum key distribution can achieve an absolutely security and detect the presence
of eavesdropper, the complicated implementation of QKD is still a challenge in physics.

2.2 Quick Response (QR) Code

QR codes can be divided into several different areas, each with their own specific purpose. The detailed structure is illustrated as explained below 10:

Finder Pattern: As illustrated in the figure 1, there are three identical structures shown in all the corners except the bottom right one. Each pattern consists of a 3*3 matrix of black modules surrounded by white modules that are again surrounded by black modules [9]. The finder pattern is used to enable the scanner can recognise the QR-code and promotes the decoder to locate the correct orientation.

Separators: The white separators are used to separate the finder patterns from the useful data, and makes the QR code can be recognised by decoder easily and quickly.

Timing Pattern: The alternating black and green modules belongs to the timing pattern. The decoder can determine the width of each module.

Alignment Pattern: The alignment patterns is used to compensate the image distortion, and this feature can help scanner to decode the data easier. For large QR code size, more alignment patterns would be required.

Format Information: Format information contains 15 bit data, and it is close to the separator. This part is used to store the error correction level of the QR code and the chosen masking pattern [9].

Data: This part is used to convert the data into a bit stream and stored in eight-bit long code-words.

Error Correction: The Error correction part works similar to part 6, and eight bit long code-words are utilised to store error correction codes.

Remainder Bits: If the data and error correction bits are not the integer multiple of 8 bits, this part is used to contain the empty bit.

QR-codes contain 40 different versions, and each version contains various size of modules. The smallest QR code has a size of 21*21 modules while the largest version consists of 177*177 modules.

Fig. 1. The Structure of QR-Code.
The QR code is used in this project. In theory, it has adequate capacity to store one-time keys, error correction ability, and can only be recognised by machine; thus, it has better extensibility in real-life applications. In addition, considering the widespread use of mobile phones and the fact that users can use their smart phone cameras to capture QR codes instead of having to buy new devices, the practical aspect of using the QR code in authentication schemes becomes more evident. Currently, QR codes are used mainly in authentication mechanisms; however, not many useful mobile applications exist that use QR codes as a tool to distribute one-time pad key. This project aims to design an APP that utilises QR-codes to solve the key distribution problems and enhance mobile phone security.

2.3 Problem Statement

The classical cryptographies include public key cryptography and secret key cryptography. While public-key cryptography is popular in the past, it is not recommended due to potential social and economic risks. Secret-key cryptography is proved to be absolutely secure in the sense of information theory; the security of this mechanism is enhanced by using the one-time pad as a secret key. The main problem of one-time pad is the key distribution, and the quantum key distribution method could hold promise for solving this issue.

3 The Developing Procedure of the APP

The APP is developed in the Android Studio software environment, and JAVA is used as the main programming language. To achieve the project’s objective, the structure of the source code has been divided into different parts; each part has its own contribution and collaborates with the other parts as well. The property of this APP is presented in figure 8; the target SDK version is API 23, while the minimum one is 16. The ‘appcliationId’ is a unique identifier for the APP; if this ID is modified, the Google application store will treat this APP as a completely different one, even if the content is unchanged. Dependency plays an important role since it is the link amongst different terminal elements (activity) of the whole project.

As shown above, the project is comprised of different folders (modules). In the project, the ‘QR_Code_demo’ works as a core module that controls all the user interfaces (UI). Section 4 will explain in detail the designing and working procedures of the APP, and the entire source code can be accessed from the CD attached at the back of the thesis.

3.1 Working principles

Figure 2 illustrates the working procedure of the APP. The client and server rely on Wi-Fi Direct to discover peers and connect with each other. The client acts as a receiver and uses the smartphone’s embedded camera to capture the QR code which is shown on the server screen. The server presents multiple QR codes per cycle, and the client decodes all of them continuously until all one-time keys are exchanged. Since the key can only be used once, the server deletes the corresponding QR code when the client informs the id of scanned QR code.
The structure of source code can be considered as a three-tier architecture which contains different modules:

**Tier 1: Devices discovery and connections**
Module: • Wi-Fi Direct connection and peer discovery

**Tier 2: The exchange of one-time keys**
Module: • Encoding and decoding of QR-code

**Tier 3: Communications**
Module: • Socket communications

![Diagram of APP Working Procedure](image)

**Fig. 2. APP Working Procedure.**

### 3.2 Devices discovery and Connections

In the first tier of the APP, the Wi-Fi Direct connection and peer discovery module establishes the connection between mobile phones.

Wi-Fi is a wireless local area networking access technology with devices based on IEEE 802.11 standards [13]. It becomes a method of wireless communications similar to Bluetooth, and multiple connected devices can simultaneously transfer data at typical Wi-Fi speeds. Wi-Fi compatible devices can connect to the internet via a wireless access point (AP). Unlike the traditional method, the Wi-Fi Direct or Wi-Fi P2P can connect with other devices without using an access point [14]. The Wi-Fi Direct belongs to the Wi-Fi protocols standardised by the Wi-Fi Alliance, which supports Peer to Peer (P2P) communications between devices [15], thus allowing mobile devices to be connected even if they are from different manufacturers. The peer to peer connection can be established immediately when one of the Wi-Fi compatible devices meets the Wi-Fi Direct requirement [16], greatly reducing the setup procedures and improving the compatibility of devices. In addition, the file is still transferred at standard Wi-Fi speeds. Wi-Fi Direct also conserves the mobile devices’ power and enhances the overall QoS, since it is based on the infrastructure mode of IEEE 802.11. Wi-Fi Direct devices implement Wi-Fi Protected Setup (WPS) to guarantee security with minimal
user intervention. This project utilises these characteristics of Wi-Fi Direct to realise high speed and efficient communications between sender and receiver.

Devices in Wi-Fi Direct negotiate to establish one peer device to work as a temporary access point for all the other devices in the group, as shown in figure 3 below [17].

![Peer Devices in Wi-Fi Direct](image)

**Fig. 3.** Peer Devices in Wi-Fi Direct.

Figure 4 illustrates that the device acts as a Wi-Fi AP, also called the group owner (GO), and that other devices are clients of the group which are connected to the GO. The group owner and the clients are elected once the session is established, and the role of each device remains the same. If the GO decides to leave, the connection will be unavailable until a new GO is elected for the new session. During the session, the group owner advertises the P2P group periodically to help new devices to discover and join the group.

![P2P Group](image)

**Fig. 4.** P2P Group.
Wi-Fi Direct supports many topologies, as shown in the example in figure 5. The laptop in the middle is a client of group1 and acts as a group owner of P2P group 2. This feature of supporting multiple topologies extends the network and increases the effectiveness of communication.

![Diagram of Wi-Fi Direct Topology](image)

**Fig. 5.** Wi-Fi Direct Topology.

In this project, two android smartphones are used to comprise the P2P group. The sender acts as a group owner while the receiver is the client. First, these two phones are placed in a fixed position (cradle) to avoid errors in measurement, then the user clicks the search button at the top of the APP menu to discover peers via Wi-Fi Direct; the name of available peer phones will display on the screen as ‘Available Devices’. The status bar of the ‘Available Devices’ screen will show three types of situations: available, invited and connected. The ‘available’ status indicates that the peer phone is available to be connected. When the user plans to connect this phone, and invites it to the P2P group, the status will change to ‘invited’. Once the connection is established, the ‘connected’ status displays on the screen, indicating that the Wi-Fi Direct group is established. In addition, the APP will automatically switch from the home page to the QR-code transmission page, where the role of each device (server and client) is displayed on the screen.

### 3.3 The exchange of one-time keys

In the second tier, the client and server are both connected and in the P2P group. The exchange of the one-time key can be realised with the QR-code encoding and decoding module, and figure 17 shows the key files (JAVA class) of these two modules. When the client sends a request for one-time keys, about 200 to 3000 bytes of random numbers can be generated by ‘GenerateActivity’, then the ‘QRCodeEncoder’ is used to encode these numbers into pictures. Each generated picture is stored in the library of the server device first, then it will display on the smartphone screen. In the decoding module, the ‘ContinuousCaptureActivity’ is used to enable the smartphone of client to recognise and decode QR-codes. The ‘callback’ function allows the decoding procedure to run continuously and avoids the scanning of duplicate QR-codes.

As shown in figure 8, the client can select one mode from the menu, which includes ‘Automatic Single Scan Mode’, ‘Manual Continuous Scan Mode’, and ‘Auto-
matic Continuous Mode’. After selecting a mode, client places the QR-code within the scanning box to improve decoding efficiency, and the random yellow point shown in the scanning box is used to locate the finder pattern of the QR code. Once the QR code is scanned, the smartphone will vibrate and play a sound to indicate the QR code is scanned. A preview page of the decoded QR-code can be found at the bottom of screen. If the user selects the ‘Manual Continuous Scan Mode’, the decoded text of the one-time keys will display. The QR code size can also be found on the screen which helps the client monitor the transmission performance. The ‘Automatic Single Scan Mode’ is used to decode one QR code automatically only. On the other hand, the ‘Pause’ and ‘resume’ button can be used to stop the QR code decoding procedure temporarily.

Fig. 6. QR-Code Decoding Procedure.
3.4 Communications

In the third tier, the socket communications module enables the client and the server to securely communicate with others. The ‘IdSendActivity’ and ‘IdSendFragment’ options allow the id of the scanned QR code to be sent from client to server. Because the one-time key can only be used once, then the sender (server) will delete the corresponding QR code upon receiving the id. During the communication procedure, the system’s security is guaranteed by the socket communications module. In addition, the message transmitted between the client and the server is the id of the QR-code; users need not worry about their valuable information being stolen by a third party.

The communication procedure is presented in figure 9. If the ‘Manual Continuous Scan Mode’ is selected, the client must press the ‘send id now’ button manually to inform server. The ‘Automatic Continuous Mode’ automatically sends the id to tell the server which QR code has been scanned. When the server receives the message, the QR code which is stored in the local library of the phone will be deleted immediately.

![Fig.7. Communications between Client and Server.](image)

3.5 Payment mode

Since the project can achieve an absolute secure communication environment between mobile electronic devices, promoting the APP to real-life applications is easily done. As a real-life application, a novel payment mode of the APP was designed. The working procedure is illustrated in figure 10.

Firstly, when a buyer wants to pay the money for a cup of coffee or withdraw money from ATM, the seller or ATM can generate a QR-code which contains a payment link for the buyer. After the buyer scans the QR code, the APP will automatical-
ly jump to the ‘send payment’ page. If the buyer checks the amount of the payment and the name of the receiver are correct, they can press the ‘send payment’ button to complete the transaction. With this method, the buyer need not worry that their pin code and bank card information could be copied by malicious users. In addition, this mode includes offline and online verification to increase security, since smartphone access usually requires a password or fingerprint authentication to protect against malicious users and the buyer is also required to enter another password to confirm their payment. In addition, the bank or seller can add their own authentication method, such as SMS verification, to enhance the security level.

This section introduces the developing procedure of the APP, and the source code structure is based on a three-tier architecture which contains different modules: Devices discovery and connections, The exchange of one-time keys, and Communications. The Client and the server establish a connection via Wi-Fi Direct, which is convenient and secure. Users can choose a QR code decoding mode based on their needs. The ‘Manual Continuous Scan Mode’ allows the client to monitor the transmission progress, and the ‘Automatic Continuous Mode’ automates the scanning, communication, and deletion of QR codes, thereby achieving ideal transmission performance. The one-time pad keys distribution procedure of the system is completed when every QR code has been scanned and deleted. Since one-time keys are exchanged face to face, thus eliminating the possibility of copying by attackers, the security of the whole system is guaranteed. Finally, the introduction of a novel payment mode promotes the APP as a real-life application.
4 The Evaluation of Transmission Performance

Introducing the developing procedure of the APP reveals that the overall transmission performance is related to several parameters: the QR code size (m) in bytes, the display rate (r) / (the number of pictures presented per second), the quantity of QR-codes per cycle (n), and the time (t) between camera auto focus intervals in seconds. Therefore, about m*n bytes can be exchanged by using the APP, and the overall transmission rate (R) in bytes/s is calculated as: \[ R = \frac{m \times n}{t} \], and the total decoding time (T) can be found in the ‘Android monitor’ by connecting smartphones to a computer.

This Section discusses four experiments that are designed to analyse transmission performance in different scenarios and optimise the transmission rate of the system. Only one parameter can be changed in each experiment, and the mobile phones are placed in a cradle to avoid measurement errors. In addition, each set of data has been analysed at least 10 times to ensure accuracy.

4.1 The correlation between QR code size and transmission rate

Figure 24 shows how a larger QR code size brings a higher complexity to the pictures. This phenomenon may increase the amount of time the phone spends in analysing the decoded data. Since users expect to transfer more data at an ideal speed, the best transmission rate of the system will be discussed in this section.

![QR Codes](image)

200 bytes 3000 bytes

Fig. 9. QR-Code in 200 and 3000 bytes.

The first experiment analyses the performance of the APP by changing QR code sizes from 200 bytes to 3000 bytes in different situations. For example, ‘r=1, n=10, t=2 s’ means 1 picture is presented per second, 10 QR codes are produced in the first cycle, and the time between camera auto focus intervals is 2 s.
As seen in figure 23, the overall transmission rate will increase until the QR code size reaches 2400 bytes, after which time, the rate will decrease rapidly. The best transmission rate is equal to 2178 bytes/s, when \( r=1 \) picture/s, \( n=10 \) QR codes, \( t=2 \) s, \( m=2400 \) bytes per QR code. The maximum data size encoded in one picture is up to 3000 bytes. Despite the larger size of QR code can deliver more data, it can negatively influence decoding efficiency. These changes of the figure 25 can be explained in this way: the larger the amount of data encoded in a QR code, the longer the client’s processing time will be. Figure 26 proves this explanation by showing that the average decoding time for one QR code increases as the QR code size becomes larger. Figure 13 shows a drastically increasing trend seen when the QR code contains data more than 2600 bytes.

4.2 The correlation between display rate and transmission rate

One of the major functions of this project is the continuous scanning mode, which is designed to improve transmission efficiency and user experience. In this mode, multiple QR codes can be provided continuously at a display rate by server and the client can scan all the QR codes continuously without press the scan button. The display rate
(r) is the number of QR-codes that can be presented every second, and it should be used as another parameter to test the system’s performance.

In the second experiment, the display rate is set from 1 to 6 pictures per second, while the other factors remain the same in every set. As shown in the figure 14, since the phone can decode 1800 bytes in around 600ms, when the QR-code size is smaller than 2400 bytes, the transmission rate can be improved by setting the display rate to 2 pictures/s. However, when the server presents the QR-code at 6 pictures per second, the client will need to capture and decode information in 0.166 (\(\frac{1}{6}\)) s. The smartphone is more likely to fail to distinguish one QR-code from another, possibly causing unstable decoding performance and producing a worse result. In addition, users are advised not to set the display rate faster than 1 pictures/s when they plan to transfer QR code sizes larger than 2400 bytes, as the performance will be degraded dramatically.

4.3 The correlation between quantity of QR codes and transmission rate

While the client expects to transmit as many QR-codes as possible every time, this could be a challenge for the APP. Therefore, the third experiment is designed to evaluate the correlation between the quantity of QR codes and the transmission rate.

In figure 15, the total quantity of presented QR-codes varies between 10 and 25. When the QR code size is smaller than 1000 bytes, the average decoding time for one QR code is 500ms, suggesting that the quantity does not significantly influence the transmission rate. However, the transmission rate dramatically decreases when the QR-code size gets larger than 1600 bytes. The reason might be that, when a client attempts to capture more QR-codes, the APP is less likely to decode all the data in one cycle, thus increasing the overall decoding time and achieving a worse transmission rate. Considering the efficiency of the system, the ideal total quantity of QR-codes presented by the server is 10 pictures.
4.4 The correlation between camera auto-focus interval and transmission rate

As discussed in section 4.4, the embedded camera is the hardware of the smartphone and the camera is expected to be utilised as efficiently as possible. The ‘AutoFocusManager’ of the source code initially sets the auto-focus interval to be 1 s.

As shown in figure 16, when the camera is trying to capture pictures every 0.1 s, the smartphone will have a harder time recognising and distinguishing each QR code, resulting in an undesirable rate. Since the transmission rate and autofocus interval show a positive relationship, it is suggested to set the camera auto-focus interval higher than 0.1 s to stabilize the decoding system. In addition, when the QR code size is larger than 1200 bytes, the transmission rate can be enhanced dramatically by changing the auto-focus interval from 0.1 s to 2 s. When the QR code size is smaller than 400 bytes, the decoding time for one QR code is around 400 ms, and the camera auto-focus may not significantly influence the system’s transmission rate.

**Fig. 13.** The Correlation between Quantity of QR-Codes and Transmission Rate.

**Fig. 14.** The Correlation between Camera Auto-focus Interval and Transmission Rate.
This section analyses the trade-off between transmission rate (R) and QR-code size (m), display rate (r), quantity of QR-code (n), and camera auto-focus interval (t). The system’s performance can be improved by encoding more data in each QR-code; however, transmitting a QR code that is larger than 2400 bytes would negatively influence the transmission rate. The ideal display rate depends on the QR code size. Since the APP can process 1800-2000 bytes in 500-600 ms, when the QR-code size is smaller than 2000 bytes, the display rate can be set to 2 pictures/s. Otherwise, setting the rate to 1 picture/s is suggested. In addition, the quantity of QR code and the camera auto-focus interval are recommended to be set as 10 pictures and 2 seconds, respectively. Last, the maximum transmission rate is around 2200 bytes/s when m=2400 bytes, r=1 picture/s, n=10 pictures, and t=2 s.

5 Conclusion and future work

This paper develops an Android APP to provide a secure communication channel for smartphone users, and the one-time pad key distribution problem is solved by using QR codes as a carrier. After introducing the relevant background and some existing technologies of the project, the developing and working procedures of the APP are analysed to help the reader understand each function clearly.

The large storage capacity and quick response of QR codes improve the performance of the system. The correlation between transmission rate and several parameters have been evaluated in order to find the optimised result. About 1800 bytes of random keys can be decoded in 600 ms, and the maximum transmission rate is around 2200 bytes/s. The continuous scanning mode improves the user’s experience by improving the working efficiency, and the payment mode makes this APP be more applicable to real life. The project achieves a guaranteed security since the QR code can be exchanged face to face to avoid eavesdropping, and the Wi-Fi direct connection can also protect users’ privacy.

For feasible future works, the transmission rate for QR codes larger than 3000 bytes could be enhanced by improving the hardware of smartphone, such as the screen resolution. Since the versions of Android vary, solving the API compatibility problem is recommended. Further, adding an exchange bank pin code function is suggested, which could result in increased APP applicability.

References
1. Gunelius, S. (2014, July 12). The data explosion in 2014 minute by minute – infographic. ACI.
2. Gisin, N., Ribordy, G., et al. (2002). Quantum cryptography. Reviews of Modern Physics, 74, 145.
3. Shannon, C. (1949). Communication theory of secrecy systems. Bell System Technical Journal, 28 (4), 656–715.
4. Molotkov, S. N. (2006). Quantum cryptography and V A Kotel’nikov’s one-time key and sampling theorems.
5. Liao, K. (2010). A novel user authentication scheme based on QR-Code. vol. 5.
6. Lamport, L. (1981). Password authentication with insecure communication. vol. 24, pp 772.
7. Bennett, C. H. and Brassard, G. (1984). Quantum cryptography: Public key distribution and coin tossing. In Proceedings of IEEE International Conference on Computers, Systems and Signal Processing
8. Bruss, D. (1998). Optimal eavesdropping in quantum cryptography with six states. Phys. Rev. Lett., 81, 3018–3021.
9. P. Kieseberg, M. Leithener, et al. “QR Code Security”, presented at the Proceedings of the 8th International Conference on Advances in Mobile computing and Multimedia, Paris, France, 2010
10. Qrcode.meetheed.com. (2017). “QR Codes - What are the different sections in a QR code?” Available at: http://qrcode.meetheed.com/question14.php [Accessed 27 April. 2017].
11. N. Harini, T. Padmanabhan, “2CAuth: A new two factor authentication scheme using QR-code”, International Journal of Engineering and Technology, Volume 5, Number 2, p.1087-1094 (2013)
12. A. Sen, Y. Jou, “Secure Data Transmission Technique for iPhone using Quick Response (QR) code”, 2015
13. “What is Wi-Fi (IEEE 802.11x)? A Webopedia Definition”. Webopedia.com.
14. “Wi-Fi Direct | Wi-Fi Alliance”. Wi-fi.org. Retrieved 2013-07-30.
15. L. VERMA and I. J. Na, “Method and device for investigating Wi-Fi display service in a Wi-Fi direct network,” ed: Google Patents, 2012
16. “Can a network based on devices certified under the Wi-Fi Direct program cross connect to an infrastructure network for Internet connectivity? | Wi-Fi Alliance”. Wi-fi.org. Archived from the original on April 3, 2013. Retrieved 2013-07-30.
17. Nuno Santos, “Wi-Fi Direct in Android Mobile and Ubiquitous Computing”, 2015-2016