Payload Formats and Universal Data for Near Field Communication

L. N. Harnaningrum
Department of Computer Science & Electronics, Gadjah Mada University, Yogyakarta, Indonesia
STMIK AKAKOM Yogyakarta, Indonesia

Ahmad Ashari
Department of Computer Science & Electronics, Gadjah Mada University, Yogyakarta, Indonesia

Mardhani Riasetiawan
Department of Computer Science & Electronics, Gadjah Mada University, Yogyakarta, Indonesia

ABSTRACT
NFC has a variant where Type 1 with NFC-A, Type 2 with NFC-A, Type 3 with NFC F, and Type 4 with NFC-B. In its application, the use of NFC is based on its tag type. If we go further, each type has similarities and differences. We can look at the various parameters, but for now we will look at 4 parameters, namely Sequence format, Bit level coding, frame format and data & payload format. From the four parameters, compared to the parameter value and obtained the difference and the equation.

With that use, an mobile application only runs in one type, while for another type must be modified. To overcome this, it is proposed a framework that can handle the tags of different types. Proposals are made for universal data and payload formats, and the expectations can be recognized by all NFC tag reading tools.

Architectures are built with attention to the similarities and differences for each type. For above four parameters. Beginning with getting the data for the use of the most tag type, then made the proposed universal data format. Commonly used types are types 1 and 2. Of the two types, the differences of the four parameters are on the data and payload parameters. And more specifically on the CRC. Thus, a universal data and payload format with a specific CRC is used.

General Terms
Embedded system, proximity card technology.

Keywords
NFC, communication, multi-tag, universal, IoT.

1. INTRODUCTION
Near Field Communication (NFC) is a short-range communication technology. Due to the popularity of NFC, the NFC is widely applied in the field of Internet of Things (IoT) [1]. NFC technology reduces the use of many other devices, so only smartphones, which have embedded NFCs in them, are then widely used for various purposes. Smartphones are becoming quite popular nowadays, as smartphone users are become increasing over time. With that development, the embedded NFC inside the smartphone is widely used and quite popular [2].

Implementation of NFC has been done in various fields, such as health, business, transportation and others [2]. NFC applications in the Service Domain include healthcare, location, finance, social networking, entertainment, education etc. In the field of healthcare applications, NFC can facilitate communication between patients and paramedics, such as doctors, nurses, pharmacists, and other sections. In the field of healthcare management is used to communicate patient data carried by patients in the form of NFC cards and data on the server. NFC is also used to create healthcare information that can come from a smartphone. The NFC's Location Based Service (LBS) application is used for tracking applications, navigation and routing applications, identification and access control applications. In finance NFC is used for payment, E-Money and E-Wallet Application, ticketing application, coupon application and loyalty application. In other areas there is social networking application, entertainment application, education application, miscellaneous NFC application.

In terms of the development of his own tag, NFC can be discussed with two kinds approach. First from the side of his tag, the second in terms of communication technology. NFC tag type there are 4 kinds, type 1, type 2, type 3 and type 4. Each type has its own specification; there is the same part, there are different. The frame, data and payload formats have different specifications. As for the technology there are 3 kinds, NFC A, B and F. Each different type of delivery of the signal and the format of data that is sent.

The purpose of multi-tagging is to make one application accessible to multiple cards, even with different types. This will make it easier for developers to develop applications. Similarly, from the side of the device, will be cheaper and easier in implementation if one device can be used by many cards. The hope of creating a framework that can be multi-tagging applications and devices can be recognized, identifiable and ultimately workable even though its type is different. This paper will discuss about the proposed NFC tag framework based on the results of the paper review. The proposal is still in the form of architectural design, not yet until the test phase.

2. NFC AND MULTITAGING
NFC is based on Radio Frequency Identification (RFID) technology. NFC operates at 13.56 MHz and follows ISO14443 and ISO 18092 for low-level data exchange between two NFC devices. Specifically, these two ISO standards specify the operating frequency, modulation, coding scheme, routine anti-collision, and communication protocols. NFC data exchange format (NDEF) and NFC tag format is defined by Forum NFC.[3]

NFC devices can be divided into two categories: active and passive devices. An active device, such as a phone with NFC facility and an NFC card reader, is always connected to a power source or has a battery installed in the device. Furthermore, the device generates an electromagnetic field

when the device wants to communicate with the expected NFC peer. In contrast, a passive device often has no power source, except that an electromagnetic field is generated with active devices that are at close range with passive devices. Therefore, the active device must continually question the passive device to detect if a passive device is at its disposal. Examples of passive devices are NFC tags, contactless smart cards and NFCs that are part of the phone being in card emulation mode.

While in terms of communication, NFC is divided into 3 modes, namely card emulation, peer to peer and read / write.

NFC is the development of RFID. If the RFID has a communication range of up to 10 meters, the NFC can only reach up to 10 cm. However, neither the NFC nor the RFID has any problems if the communication involved is more than one, both its tag and the reader. Several studies have tried to overcome the problem. Some of them are as follows.

In [4], in RFID systems, tag identification collision problems occur during data transmission because multiple tags respond at the same time when the reader sends a query. The system uses a multi-reader configuration method to minimize collision tags and to reduce tag recognition time. The results of the trials show a reduction in processing time to 40%. And the number of ideal reader is 2 pieces.

The introduction of RFID tags is also done by [5]. Implement a passive RFID system to identify multiple simultaneously marked objects, assuming that the number of tags is not known before. Then create a system to overcome this, then test with a variety of different tags and frame size. The results can be used to show how to efficiently identify a set of fixed RFID tags if the number of tags is unknown before and determine the tag reading parameters to achieve optimal run time. This work can be extended to the case of continuous tag readings, where tags enter and leave the area continue to run at high frequencies.

Radio Frequency Identification (RFID) system, with features such as non-contact, high precision and low cost, becomes an attractive technology in the field of indoor positioning [6]. This has caused widespread concern in recent years. dynamic analysis method, in which the Fisher information matrix is introduced into the RFID multi-tag distribution. paths and rates chosen have a direct impact on RFID performance in dynamic network environments.

About the stack architecture, proposed by [7]. Mobile device manufacturers expect the NFC stack to be an independent OS, independent hardware and adaptive OS application framework, but today's NFC mobile devices can barely meet all these requirements. Analyze OS services, ETSI standards and NFC Forum standards, Then propose a new NFC stack architecture. In this architecture, the NFC stack runtime environment encapsulates OS services associated with NFC stacks. The abstract layer of NFC hard-ware hides the distinction between different types of NFC controllers and offers an abstract NFC hardware service. This paper proposes a novel mobile device NFC stack architecture. The result is that the architecture stack works well on some mobile phones, compatible with NFCC chips, and supports third party applications.

The NFC adoption for IoT devices is done by [1]. Due to the popularity of NFC, the NFC technology is widely adopted by IoT devices. This creates a vulnerability to interference. A security-aware architecture is introduced in this paper, to protect NFC devices and related data from multiple attacks. The proposed NFC system has a secure protocol, connected to an NFC chip. First, checking the newly introduced NDEF message certificate and the sender's signature, and then if appropriate, will be accepted. If not, it will be discarded and simulated on MATLAB. The simulation results also show that the processing time remains almost the same for various recording sizes. Note that smaller signature sizes help to save processing time. Although the proposed system may require little additional time due to the addition of a secure protocol, it offers better security on various network attacks including modification certificates, data modifications, and jamming attacks. The proposed security protocols for the NFC architecture can be extended to improve the security of IoT applications and NFC devices.

With the development of NFC, issues are increasingly open, especially security and privacy authentication. Many NFC authentication protocols have been proposed for it, some of which only improve functionality and performance without considering security and privacy, and most protocols are heavy-weight [8]. This paper proposes an ultralight-weight mutual authentication protocol, which is named ULMAP. ULMAP only uses Bit and XOR operations to complete mutual authentication and prevent denial of service (DoS) attacks. In addition, it uses sub-key and sub-index numbers into its main update process to achieve security ahead. The most important thing is that the calculation and overhead of ULMAP storage are few. The proposed scheme has higher security and performance. Because the database stores the private key and the old and new session IDS, when the new session private key fails to update, the private key and the corresponding old IDS can also be used. The results show that the proposed scheme is lightweight, economical, practical, and easily protected from synchronization attacks.

3. ANALYSIS IN NFC TYPE TAG

NFC tag type there are 4 kinds, namely type 1 type 2, type 3 and type 4. While from the technology side, NFC is distinguished on the type of NFC-A, NFC-B and NFC-F. The types have their own specifications and can be viewed from various sides. Here we will discuss from 4 specifications, namely sequence format, bit level coding, frame format and data & payload format.

3.1 Sequence Format

From sequence format side, type 1 and type 2 uses the NFC-A synchronization mechanism, type 3 uses the NFC-F synchronization mechanism, and type 4 uses the NFC-B synchronization mechanism.

NFC-A and NFC-B distinguish polls and listen modulation. Type A NFC using Modified Miller coding with ASK 100% modulation for poll to listen modulation. While NFC type B in Poll Mode, the analog signal is modulated using NRZ-L coding based on the ASK 10% modulation principle . Using the modulation principle of NRZ-L coding with ASK modulation, the carrier amplitude is varied in order to define two particular patterns: L and H.

In type NFC-A and NFC-B conditions listen to the poll uses Manchester coding with OOK subcarrier modulation. NRZ-L with BPSK modulation uses a phase shift (180 °) of the subcarrier to define two particular patterns: N and M.

The different NFC types are NFC-F. NFC-F does not distinguish between poll-listen and the poll. Both use the same modulation. In both transmission directions, the analog signal is modulated using Manchester coding with ASK modulation.

In NFC-A the sync mechanism used does not require a sync signal, so the SoS is not required. And de-synchronization is indicated by EoS indicating the end of a sequence. While NFC-B is shown in Figures 1 and 2, illustrating the different parameters used for NFC-B technology signal synchronization and related signal timing parameters.

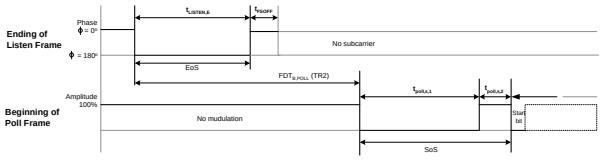

Figure 1. Synchronization and Timing Parameters between a Listen Frame and a Poll Frame[9]

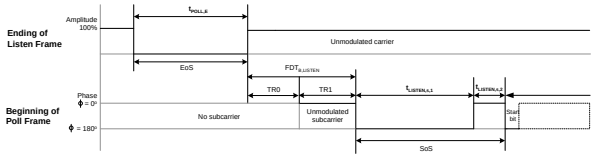

Figure 2. Synchronization and Timing Parameters between a Poll Frame and a Listen Frame [9]

Patterns are grouped in a pattern group that are 10 bit durations long. The separation between two pattern groups is defined as the Extra Guard Time (EGT). For the demodulator, the beginning of a signal is indicated by the SoS.

3.2 Bit Level coding

At bit level encoding types 1 and 2 use NFC-A bit level coding, type 3 uses NFC-F bit level coding, while type 4 uses NFC-B bit level coding.

Bit level coding for NFC-F type does not distinguish between poll and listen, both are the same. The patterns E and D are used to code the digital alphabet Logic "0" and Logic "1". For Poll and Listen Mode, the NFC Forum Device MUST code Logic "0" pattern E and logic "1" pattern D. And if the NFC Forum Device detect pattern E, then it MUST decode this as Logic "1", if detect pattern D, then it MUST decode this as Logic "0".

Bit level coding for type NFC-A and NFC-B distinguish between poll-listen and listen-poll. Poll-listen coding scheme for NFC-A as follow. The patterns X, Y, and Z are used to code the digital alphabet Logic "0" and Logic "1". Logic "0"s and Logic "1"s are the components of frames. For the poll Mode, the NFC Forum Device MUST code logic "1" pattern X and logic "0" pattern Y, with exception pattern Z MUST be used to code the first Logic "0" (SoF) and if there are two or more contiguous Logic "0"s, pattern Z MUST be used from the second Logic "0" on.

For the listen Mode, The NFC Forum Device MUST decode Logic "0" and Logic "1" as follows. The first pattern Z MUST be decoded as Logic "0"; If the NFC Forum Device detects pattern X, then it MUST decode this as Logic "1"; If the NFC Forum Device detects pattern Y after pattern X, then it MUST decode pattern Y as Logic "0"; If the NFC Forum Device detects pattern Z after pattern Y, then it MUST decode pattern Z as Logic "0"; If the NFC Forum Device detects pattern Z after pattern Z, then it MUST decode the last pattern Z as Logic "0".

For the Listen-poll coding scheme, the patterns E and D are used to code the digital alphabet Logic "0" and Logic "1". Logic "0"s and Logic "1"s, referred to as data bits, are the components of frames. For the Poll Mode, if the NFC Forum Device detects pattern D, then it MUST decode this as Logic "1" and if the NFC Forum Device detects pattern E, then it MUST decode this as Logic "0". For the Listen Mode, the NFC Forum Device MUST code Logic "1" for the pattern D and Logic "0" for the pattern E.

The patterns L and H are used to code the digital alphabet Logic "0" and Logic "1" for NFC-B poll-listen coding scheme. For the Poll Mode, the NFC Forum Device detect pattern L, it MUST decode this as Logic "0" and if detect pattern H MUST decode this as Logic "1". And then, if at Listen Mode, the NFC Forum Device MUST decode Logic "0" if detect pattern L and decode Logic "1" if detect pattern H. For the listen-poll coding scheme, use patterns N and M to code the digital alphabet Logic "0" and Logic "1". For the poll mode, if the NFC Forum Device detects pattern E, then it MUST decode this as Logic "0". And if the NFC Forum Device detects pattern D, then it MUST decode this as Logic "1". For the listen mode, the NFC Forum Device MUST code Logic "0" and Logic "1" if detect pattern D and logic "0" if detect pattern E.

3.3 Frame Format

NFC tag types from the frame format side can be distinguished as follows. Type 1 uses NFC-A frame format for Listen Mode and a specific frame format for Poll Mode. A Poll Frame consists of the following (Figure 3): NFC-A SoF, 8 data bits, transmitted lsb first, EoF. No parity is added. As for type 2 NFC is as follows. The Type 2 Tag platform transmits Commands and Responses in NFC-A standard frames, except for the ACK and NACK Response. A Listen Frame for the ACK and NACK Response consists of a short frame with 4 data bits. The Type 3 Tag platform transmits Commands and Responses in NFC-F frames. While type 4 is divided into 2 kinds, namely 4A and 4B. The Type 4A Tag platform transmits Commands and Responses in NFC-A standard frames. The Type 4B Tag platform transmits Commands and Responses in NFC-B frames.

	lsb			msb					
SoF	b1	b2	b3	b4	b5	b6	b7	b8	EoF

Figure 3. Frame format NFC tipe 1. [9]

When viewed from the technological side, Frame format for NFC-A is as follows. Data bits, when transmitted between NFC Forum Devices, are grouped within frames. The format of a frame is different for each Technology. NFC-A Technology groups the data bits together in a frame by adding an SoF and an EoF. A parity bit (P) is added at the end of each 8 data bits. Uses three types of frames: short frame, standard frame, and bit-oriented SDD frame. The short frame is used to initiate communication (wake-up). The standard frame is used for data exchange. The bit-oriented SDD frame is used for collision resolution. A short frame is used to initiate communication and consists of the following (Figure 4): SoF, up to 7 data bits transmitted lsb first, EoF. No parity is added. Following the SoF and preceding the EoF, the short frame MUST contain less than 8 data bits**.**

	lsb			msb				
SoF	b1	b2	b3	b4	b5	b6	b7	EoF

Figure 4. Short Frame[9]

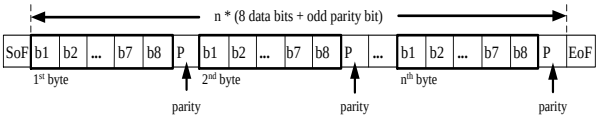

Figure 5. Standard Frame (Poll→Listen Communication) [9]

Standard frames are used for data exchange and consist of the following (Figure 5): SoF, n * (8 data bits + odd parity bit), with n ≥ 1, EoF (Poll→Listen communication only). Each 8 data bits in a standard frame MUST be followed by an odd parity bit. The parity bit P MUST be set such that the number of Logic "1"s is odd in the set consisting of b1 to b8 and P. Bit oriented SDD frames are used for collision resolution and result from a standard frame with a length of 7 bytes that is split into two parts. The split can occur after any data bit. Figure 6 shows an example with the split after the first bit of the second byte.

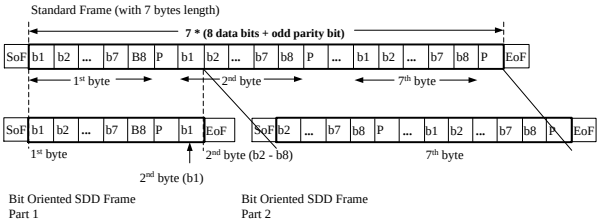

Figure 6. Bit Oriented SDD Frame (with Split after the First Bit of the Second Byte) [9]

The format frames for the NFC-B technology type are as follows. To transmit data, the NFC Forum Device configured for NFC-B Technology uses frames that are built from characters, who defines the format of characters and frames. A character consists of a Logic "0" start bit, a Logic "1" stop bit and eight data bits. The stop bit, start bit, and each data bit has a length of one bit duration (bd). Character are sent as frame. See Figure 7 and 8.

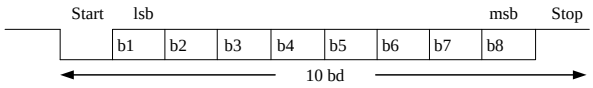

Figure 7. NFC-B – Character Format[9]

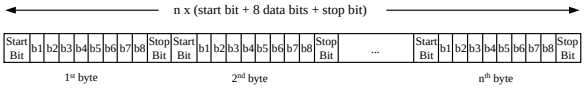

Figure 8. NFC-B – Frame Format[9]

While the frame format for NFC-F is as follows. A character consists of 8 data bits without start, stop, and parity bits, as shown in Figure 9. Characters are transmitted as a continuous string, with no separation in time between characters. A frame starts with the SoF followed by the data, as illustrated in Figure 10. Data consists of characters.

msb							lsb
b8	b7	b6	b5	b4	b3	b2	b1

Figure 9. NFC-F – Character Format[9]

SoF	Data (consisting of Characters)

Figure 10. NFC-F – Frame Format[9]

3.4 Data and Payload Format

For data and payload formats, each type has the following specifications. Type 1 Tag data has the following substructure, and consists of the payload and the EoD. The payload consists of the Commands and Responses. The Type 1 Tag data and payload format is illustrated in Figure 11. The EoD contains a two-byte checksum referred to as CRC_B. Input for CRC_B calculation is the payload.

Data					
Payload				EoD	
Byte 1	Byte 2	...	Byte n	CRC_B1	CRC_B2

Figure 11. Data and payload format type 1 NFC tag. [9]

Data					
Payload				EoD	
Byte 1	Byte 2	...	Byte n	CRC_A1	CRC_A2

Figure 12. Data and payload format type 2 NFC tag. [9]

Type 2 Tag data transmitted in an NFC-A standard frame, i.e., the bytes following the SoF, have the following substructure. They consist of the payload and, depending on the payload, of the EoD. The payload consists of the Commands and Responses. If present, the EoD contains a 2-byte checksum referred to as CRC_A. Input for CRC_A calculation is the payload. If the payload consists of the ACK or NACK Response, then EoD is not present. Type 2 Tag data and payload format is illustrated in Figure 12. Type 3 Tag data follow the data and payload format for NFC-F Technology. During the Device Activation Activity, Type 4A Tag data has the following substructure, which consists of the payload and of an EoD. The payload consists of the Commands and Responses. The EoD contains a two-byte checksum referred to as CRC_A. Input for CRC_A calculation is the payload.

Data					
Payload (Command or Response)				[EoD]	
Byte 1	Byte 2	...	Byte n	[CRC_A1]	[CRC_A2]

Figure 13. Data and Payload Format – NFC-A Standard Frame[9]

Data and payload format for NFC-A as follow (Figure 13). Data embedded in NFC-A short frames or NFC-A bit oriented SDD frames do not have SoD and EoD. Data embedded in NFC-A short frames or NFC-A bit oriented SDD frames do not have SoD and EoD. Payload exchanged between NFC Forum Devices consists of Commands and Responses.

Table 1. NFC-A – Command Set[9]

Command	Response	EoD Present	Frame Type
ALL_REQ. SENS_REQ		No	Short Frame
	SENS_RES	No	Standard Frame
SDD_REQ		No	Bit Oriented SDD Frame
	SDD_RES	No	Bit Oriented SDD Frame
SEL_REQ		Yes	Standard Frame
	SEL_RES	Yes	Standard Frame
SLP_REQ		Yes	Standard Frame

Data transmitted in an NFC-B frame have the following substructure. They consist of the payload and of an EoD (SoD is not used).

Data					
Payload (Command or Response)				EoD	
Byte 1	Byte 2	...	Byte n	CRC_B1	CRC_B2

Figure 14. Data and Payload Format – NFC-B [9]

The payload consists of the Commands and Responses as at Table 2. The EoD contains a 2-byte checksum referred to as CRC_B. The CRC_B is a function of k data bits, which is a multiple of 8. Input for CRC_B calculation is the payload. Figure 14 illustrates the NFC-B data and payload format.

Table 2. NFC-B – Command Set[9]

Command	Response
ALLB_REQ, SENSB_REQ	SENSB_RES
SLOT_MARKER (optional for Poll Mode)	SENSB_RES
SLPB_REQ	SLPB_RES

Data transmitted in an NFC-F frame has the following substructure. They consist of an SoD, the payload, and an EoD. The SoD contains a length byte LEN, indicating the length of the payload + 1. The payload consists of the Commands and Responses, as in Table 3. The EoD contains a two-byte checksum, referred to as CRC_F.

Table 3. NFC-F – Command Set [9]

Poll Mode (Command)	Listen Mode (Response)
SENSF_REQ	SENSF_RES

The CRC_F is a function of k data bits. The number of bits k is a multiple of eight. Input for CRC_F calculation is the SoD and the payload. The NFC-F data and payload format is illustrated in Figure 15.

Data						
SoD	Payload				EoD	
LEN	Byte 1	Byte 2	...	Byte n	CRC_F1	CRC_F2

Figure 15. Data and Payload Format – NFC-F [9]

If summarized, then the four parameters of the above NFC type can be seen in table 4.

Table 4. Summary of 4 NFC parameters

	Tipe 1	Tipe 2	Tipe 3	Tipe 4
Sequence Format	NFC-A	NFC-A	NFC-F	NFC-B
Bit level coding	NFC-A	NFC-A	NFC-F	NFC-B
Frame format	NFC-A	NFC-A	NFC-F	NFC-A or NFC-B
Data and Payload Format	Refer Figure 17	Refer Figure 18	Refer Figure 21	Refer Figure 18

4. UNIVERSAL DATA AND PAYLOAD FORMAT

4.1 Description of Universal Format

The widely used NFC applications are Type 2 and then Type 1. Type 2 is widely used in various applications is Mifare (https:// en.wikipedia.org/wiki/MIFARE), and Type 1 is Topaz. As for NFC-embeded mobile phone many use NFC Controller NXP.

Based on the usage and architecture of each type of NFC, the Universal Data and Payload Format of NFC tag as follows.

Table 5. Universal Data and Payload Format for Multitaging

Description	Universal Format
Sequence Format	NFC-A
Bit level coding	NFC-A
Frame format	NFC-A
Data and Payload Format	DATA / PayLoad (Byte 1, Byte 2, ..., Byte n) / EoD (CRC)

4.2 Specification on CRC

Universal Data and Payload Format for NFC frequently used.

At NFC-A and NFC-B, data consists of Payload and EoD. In the EoD section there is a Cyclic Redundancy Check (CRC). At NFC-A there is CRC_A and at NFC-B there is CRC_B.

The CRC_A encoding and checking process is defined in ITU-T. The generator polynomial used to generate the check bits is $x16 + x12 + x5 + 1$. The initial value shall be '6363'. The CRC_A shall be appended to the data bytes and shall be transmitted via standard frames. [10]

Data bytes (n Bytes)	CRC_B (2 Bytes)

Figure 16. Position of a CRC_B within a frame[10]

A frame shall only be considered correct if it is received with a valid CRC_B value. The frame CRC_B is a function of k data bits, which consist of all the data bits in the frame, excluding start bits, stop bits, delays between bytes, SOF and EOF, and the CRC_B itself. Since data is encoded in bytes, the number of bits k is a multiple of 8. For error checking, the two CRC_B bytes are included in the frame, after the data bytes and before the EOF. The CRC_B is as defined in ISO/IEC 3309. The initial register content shall be all ones: 'FFFF'. These two CRC_B bytes occur after the k/8 data bytes and before the EOF.

Based on the CRC, CRCs are created that can be used for both data formats and payloads. Here are the byte formats for CRC_A and CRC_B. The encoding and decoding process can be carried out easily through a 16-stage cyclic shear register with an appropriate feedback gateway. Flip-flop registers are numbered from FF0 to FF15. FF0 is the left-most flip-flop where data is shifted in. FF15 will be the right-most flip-flop where data is shifted out.

Table 5. Initial content of 16-stage shift register according to value '6363' for CRC_A [10]

FF0	FF1	FF2	FF3	FF4	FF5	FF6	FF7	FF8	FF9	FF10	FF11	FF12	FF13	FF14	FF15
0	1	1	0	0	0	1	1	0	1	1	0	0	0	1	1

CRC_B has a similar way, but with different initial values. The two CRC architectures will be collaborated to achieve more optimal and universal results.

5. CONCLUSION

NFC with many types have their own characteristics. For that, it needs to be made a method that can be used to make NFC can be recognized by any reader. Then a universal data and payload format that can identify the NFC. The universal data and payload format, changing the sides of data formats and payloads, in particular the CRC section made universally.

The universal data and payload format specification will have an effect on the performance of the card for multitaging as a whole. These effects can occur in the elaboration of payload change implicate heavier, or the distance of the distant communications, or the bit rate, maybe even the quality. If implemented against a dynamic system, both in terms of number of cards and from the side of the card flow coming out into the reading area, then it will likely also give a different effect.

If there is already universal data and payload format, it will be useful for built applications. NFC tag users can use any type for public applications, such as e-ticketing, e-payment and others. Thus, app developers can also create one app and can be used by all NFC tag owners.

The result of universal data and payload format can make the tags in different type usable and acknowledge by a device. Thus the introduction to multitagging can be done and developed.

6. REFERENCES

[1] A. Asaduzzaman, S. Mazumder, And S. Salinas. 2017. A Security-Aware Near Field Communication Architecture. Int. Conf. Networking, Syst. Secur.

[2] V. Coskun, B. Ozdenizci, And K. Ok, "The Survey On Near Field Communication. 2015. Sensors (Basel)., Vol. 15, No. 6, Pp. 13348–405.

[3] R. Minihold. 2011. Near Field Communication (Nfc) Technology And Measurements White Paper. P. 26.

[4] C. S. Kim, B. I. Jang, And H. K. Jung. 2014. Performance Analysis Of Anti-Collision Algorithm For Tag Identification Time Improvement. Int. J. Softw. Eng. Its Appl., Vol. 8, No. 3, Pp. 1–10.

[5] H. Vogt. 2002. Multiple Object Identification With Passive Rfid Tags. Ieee Int. Conf. Syst. Man Cybern., Vol. Vol.3, No. October, P. 6.

[6] Y. Xiao-Lei, Y. Yin-Shan, Z. Zhi-Min, And W. Dong-Hua. 2014. Geometric Pattern Of Rfid Multi-Tag Distribution In Dynamic Iot Environment. Icist 2014 - Proc. 2014 4th Ieee Int. Conf. Inf. Sci. Technol., Vol. 1, No. 2, Pp. 809–812.

[7] K. Xiao And L. Luo. 2013. A Novel Mobile Device Nfc Stack Architecture. Ieee 11th Int. Conf. Dependable, Auton. Secur. Comput. Dasc 2013, Pp. 169–173.

[8] K. Fan, P. Song, And Y. Yang. 2017. Ulmap: Ultralightweight Nfc Mutual Authentication Protocol With Pseudonyms In The Tag For Iot In 5g. Mob. Inf. Syst., Vol. 2017.

[9] Nfc Forum Inc. 2010. Nfc Digital Protocol Technical Specification Nfc Forum Tm. Technology.

[10] Iso/Iec 18092. 2013. International Standard Iso/Iec Telecommunications And Information Communication — Interface And Protocol," Int. Stand. Ref. Number Iso/Iec 180922013(E), Vol. Second Edi.