Authentication Challenges in NFC Technology: Standards, Vulnerabilities, and Proposed Countermeasures

Domenic Lo Lacono, Madisyn DeLozier, Owen Joslin, Sam Millman, and William Joslin

CSEC.472.01 - Authentication and Security Models

Rochester Institute of Technology

April 28, 2024

Abstract

Near Field Communication (NFC) technology, facilitating contactless communication between devices, has gained widespread popularity and adoption in various sectors, most notably in payments and data exchange. This paper provides a comprehensive overview of the authentication challenges in NFC, digging deeper into the underlying technology, authentication methods, technical architecture, security vulnerabilities, and proposed countermeasures. More minutely, it looks into NFC's operational modes and communication methods, detailing NFC specifications and functionality. It will explore NFC's practical applications, looking into tap-to-pay transactions and current countermeasures to enhance security, such as NFC-SEC and virtual cards. Additionally, this paper will look into potential NFC-related attacks like eavesdropping and Man-in-the-Middle (MitM) and will propose new advanced authentication methods such as Encryption Record Type Definition (ERTD) and Bilinear Pairing to mitigate against these attacks. Through a detailed analysis and looking at proposed solutions, this paper will contribute to a deeper understanding of NFC technology, its vulnerabilities, and mitigations, offering new insights for researchers and field practitioners.

Keywords: Near Field Communication, Tap-to-Pay, NFC-SEC, Man-in-the-Middle, Encryption Record Type Definition, Bilinear Pairing.

1 Introduction

NFC is a key development in short-range wireless technology that has greatly changed how we handle digital transactions and data exchanges. Derived from radio-frequency identification (RFID) technology, NFC allows easy communication between devices at close range, making it essential in areas like mobile payments, access control, data sharing, contactless ticketing, and more. This move towards cashless transactions using mobile payments is only expected to grow and one forecast suggests, "by 2025, 75% of all transactions will be made without cash" [1]. This projection shows the increasing reliance on technologies like NFC to shape the future of commerce and personal exchange.

To tackle the security issues that arise with this technology, mainly MitM and eavesdropping, this study explores two advanced authentication methods, Encryption Record Type Definition ERTD and Bilinear Pairing, designed to reduce these vulnerabilities.

As NFC technology continues to grow and become part of more technological interactions, understanding and solving its security challenges is crucial. This study provides a comprehensive overview of these issues, laying the groundwork for future research and development in NFC security.

The following research questions will be answered:

- What security risks does NFC authentication impose?
- What are alternative authentication methods?
- What are possible attacks against NFC Payments?
- What authentication features mitigate the risk of attacks?

This paper aims to enhance the knowledge available to researchers and practitioners in the field by detailed analysis and reviewing solutions, ensuring the secure use of NFC technology.

2 Background

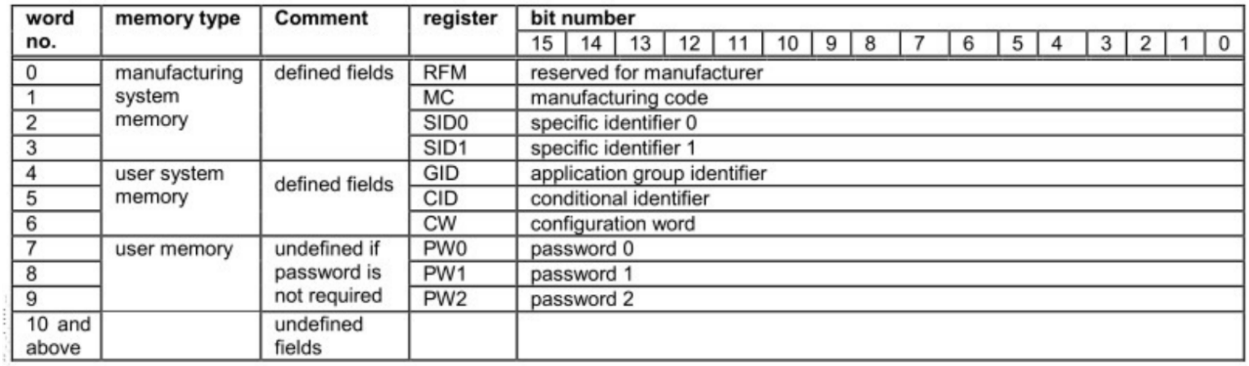
NFC is a short-range RFID technology that provides contactless communication between an NFC chip and a reader. Various NFC chip providers, such as Seritag, STMicroelectronics, Identiv Inc., and CardLogix Corporation, offer diverse authentication methods without a universal standard. Despite the availability of standards like PCI Contactless Payments on COTS (CPoC), International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC) 14443 and ISO/IEC 18092, their implementation is not mandatory. As described by Seritag, authentication relies on each tag being encoded with a secret key. This key generates a unique code with each scan, which is then verified on a server using a copy of the key. If the code matches, the tag is authenticated as the original, rendering the code invalid thereafter. If the “code is not what is expected, then the tag is assumed to be a copy,” and the authentication fails [2]. An NFC chip and an app enable transaction approval while segregating sensitive data from the phone’s operating system. NFC operates in two (2) modes: active and passive. In active mode, “both NFC devices generate their own radio frequency to carry data,” while in passive mode, “only one NFC device generates the radio frequency field” [3]. In addition, NFC also has three (3) communication modes:

- Card Emulation Mode: Allows “NFC-enabled devices to act like smart cards, [enabling] users to perform transactions such as purchases, ticketing, and transit access control" [3].
- Read/Write Mode: Allows an “NFC-enabled device. . . to read information stored on NFC tag embedded in smart posters and displays. A user can retrieve tag information that is stored in the tag for further uses” [3].
- Peer-To-Peer Mode: Allows “two (2) NFC-enabled devices to communicate with each other to exchange information and share files” [3].

This paper will further detail this basic authentication and NFC chip functions.

2.1 NFC Hardware and Frames

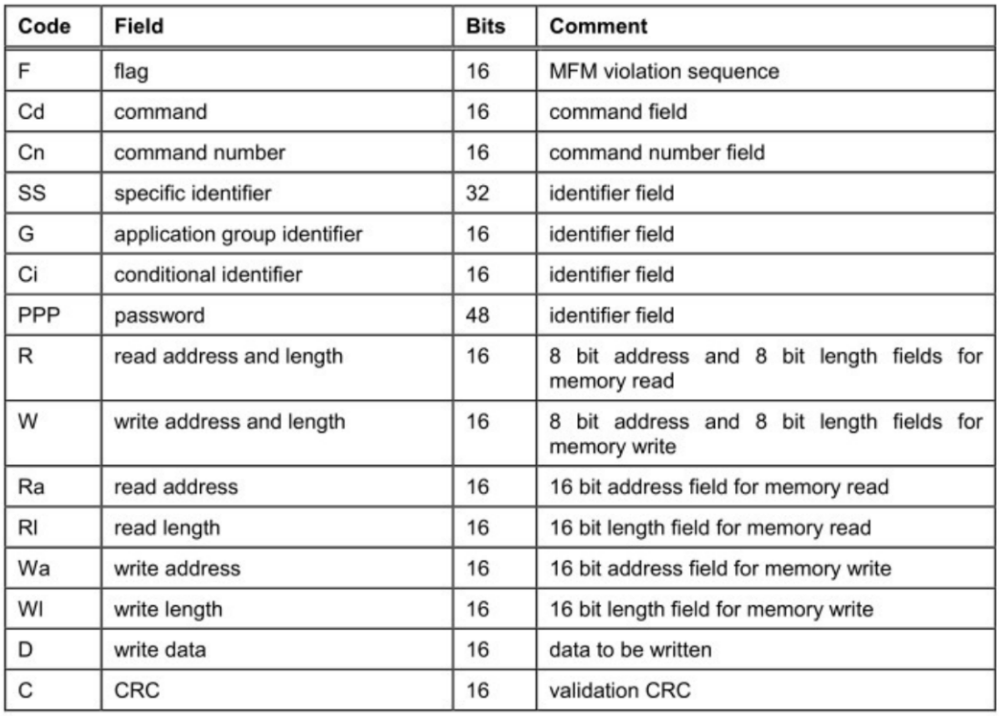
An NFC tag’s memory consists of three (3) parts: the manufacturing system memory area, the user system memory area, and the user memory area. Tags will either have four (4) kbit or less virtual memory or greater than four (4) kbit of virtual memory. Tags with four (4) kbit or less of virtual memory will use eight (8) bit addresses with eight (8) bit length fields. Tags with greater than four (4) kbit virtual memory will allow for eight (8) bit and sixteen (16) bit addresses and length fields [4]. The least significant bit is stored at lower virtual memory addresses and a lock pointer is utilized to prevent tag memory from being overwritten [4].



word no.	memory type	Comment	register	bit number															
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	manufacturing system memory	defined fields	RFM	reserved for manufacturer															
1			MC	manufacturing code															
2			SID0	specific identifier 0															
3			SID1	specific identifier 1															
4	user system memory	defined fields	GID	application group identifier															
5			CID	conditional identifier															
6			CW	configuration word															
7	user memory	undefined if password is not required	PW0	password 0															
8			PW1	password 1															
9			PW2	password 2															
10 and above		undefined fields																	

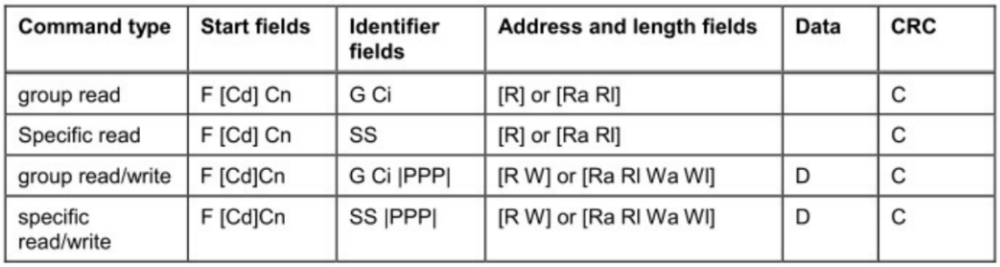
Figure 1: Virtual Memory Map Establishment [4]

NFC protocol commands have a specific format. All command fields are transmitted using the least significant bit first, as shown in Figure 2. Each command is time-stamped, and the tags store the first stamp received after entering an interrogator. The stored time defines when the tag first entered the interrogator and is utilized to determine tag order.



Code	Field	Bits	Comment
F	flag	16	MFM violation sequence
Cd	command	16	command field
Cn	command number	16	command number field
SS	specific identifier	32	identifier field
G	application group identifier	16	identifier field
Ci	conditional identifier	16	identifier field
PPP	password	48	identifier field
R	read address and length	16	8 bit address and 8 bit length fields for memory read
W	write address and length	16	8 bit address and 8 bit length fields for memory write
Ra	read address	16	16 bit address field for memory read
Rl	read length	16	16 bit length field for memory read
Wa	write address	16	16 bit address field for memory write
Wl	write length	16	16 bit length field for memory write
D	write data	16	data to be written
C	CRC	16	validation CRC

Figure 2: NFC Command Fields [4]



Command type	Start fields	Identifier fields	Address and length fields	Data	CRC
group read	F [Cd] Cn	G Ci	[R] or [Ra Rl]		C
Specific read	F [Cd] Cn	SS	[R] or [Ra Rl]		C
group read/write	F [Cd]Cn	G Ci |PPP|	[R W] or [Ra Rl Wa Wl]	D	C
specific read/write	F [Cd]Cn	SS |PPP|	[R W] or [Ra Rl Wa Wl]	D	C

Figure 3: NFC Valid Command Format [4]

Commands are generally utilized to identify tags, read, write, and lock memory and determine reply type and mode. The NFC tag will not reply to the NFC request if the Cyclic Redundancy Check (CRC) value is invalid. The request CRC and the reply CRC are calculated differently. The request CRC is a 16-bit IBM CRC defined in ISO/IEC 13239 as the following equation:

$$g(X) = X^{16} + X^{12} + X^5 + 1$$

The reply CRC is a thirty-two (32) bit ethernet CRC and is also defined in ISO/IEC 13239 as the following equation [4]:

$$g(X) = X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X^1 + 1$$

When operating the NFC tag, the baud rates needed are one hundred and six (106) kbps, two hundred and twelve (212) kbps, or four hundred and twenty-four (424) kbps. The initiator, the reader (NFC is conducted on a Reader-Talks-First or (RTF) basis), will broadcast a signal to see if a field exists at 13.56 MHz. If the field doesn't exist, it will attempt to detect any other NFC tags in the field by activating the target with its own generated radio field. This target will be initially listening. The initiator will choose the mode to communicate with the target, active or passive. The target will activate its own radio field if active mode is chosen. If the passive mode is chosen, the target will utilize the temporary transfer rate for communication. The initiator's frequency will determine the power of the target [5]. The target will confirm that the command transmitted is valid upon receiving the request. If invalid, the tag will drop the request and wait for another. The tag will randomly select one of eight channels to transmit its reply if valid. When the initiator receives the target's reply, the transfer rate will be adjusted with respect to the target's required speed [5]. This command exchange repeats every time the target receives the next valid command, randomly selecting and utilizing a new channel to reply [4]. For NFC payments, a secure radio within a smartphone will send a special code to the point-of-sale (POS) terminal, which, in return, sends the user transaction details. The user must enter a personal identification number (PIN) to approve the transaction [6].

NFC Forum Specifications are technology standards that tune and extend existing contactless standards. Protocol Technical Specifications include Logical Link Control Protocol (LLCP) Specification, Digital Protocol Specification, Activity Specification, Simple NFC Data Exchange Format (NDEF) Exchange Protocol (SNEP) Specification, and Analog Specification. LLCP defines two (2) service types essential for bi-directional communications: connectionless and connection-oriented. Digital Protocol Specification, with the implementation of ISO/IEC 18092 and ISO/IEC 14443 standards, can define the digital interface and half-duplex transmission protocol of the NFC-enabled device in all its roles, initiator, target, reader/writer, and card emulator. Activity Specification can be utilized to set up the communication protocol with another NFC device or NFC tag. When operating in peer-to-peer mode, SNEP allows an application on an NFC-enabled device to exchange NDEF messages with another NFC device. Analog Specifications are used to describe and specify the distinct features of external signals for NFC-enabled devices without specifying the design of the antenna for NFC-Enabled Devices [6].

2.2 NFC Tap-to-Pay Process

As seen on mobile devices, NFC utilizes tap-to-pay technologies to allow users to securely store credit or debit card information and easily utilize it for any NFC-enabled transaction, resulting in ease of use, saving time, and mobility. Card information is stored in an isolated location from the rest of the mobile device's operating system. NFC payments involve a secure radio within the

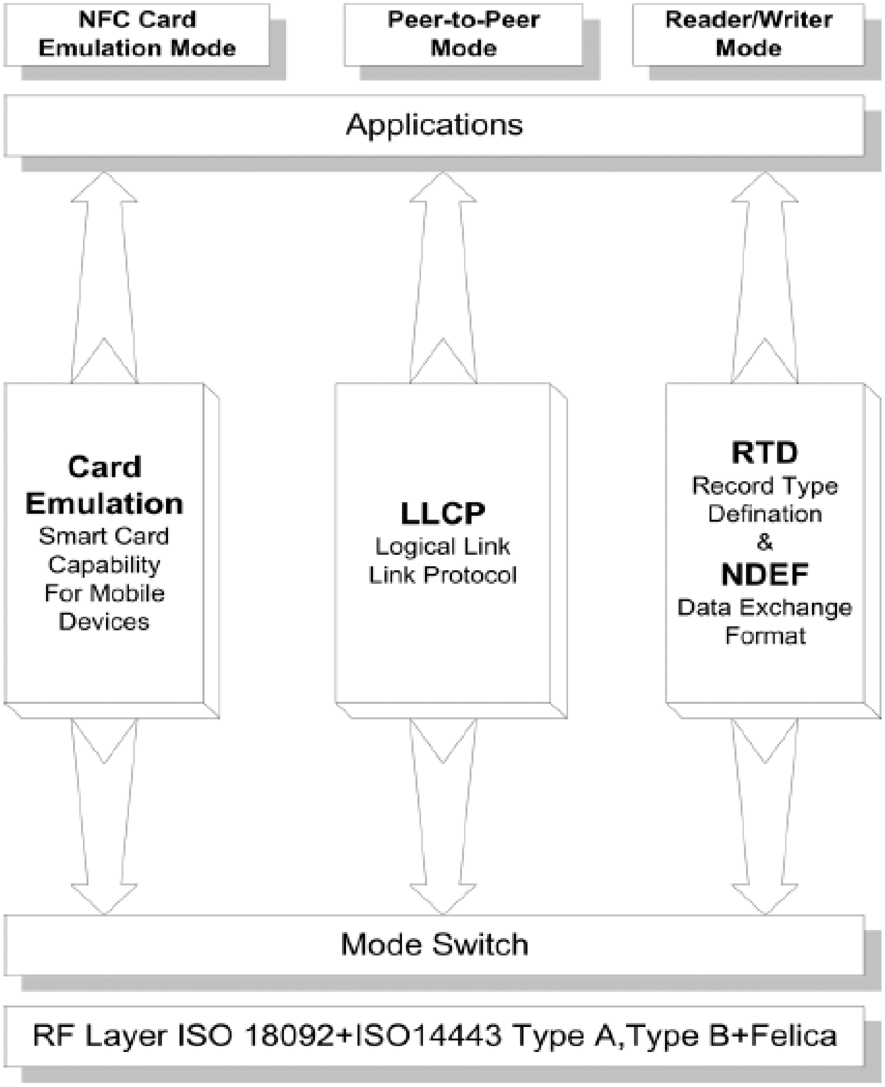


Figure 4: NFC Technical Architecture [6]

mobile device, sending a special code to a retailer's POS terminal [6]. As a result, the POS terminal will reply with transaction details to the mobile device. The transaction is then usually approved by a user's biometrics or typing a passcode for payment details to be provided to the POS terminal.

3 Attacks

3.1 Eavesdropping

Since NFC communication is wireless, eavesdropping attacks are a potential vulnerability. Eavesdropping attacks can occur when the NFC tag is set to either card emulation mode or peer-to-peer operation mode. In card emulation mode, the content of an NFC chip can be read by a threat actor even when the device is not in use [3]. This puts user privacy at risk. In peer-to-peer mode, if the data link is not using encrypted communication, a threat actor could perform an eavesdropping

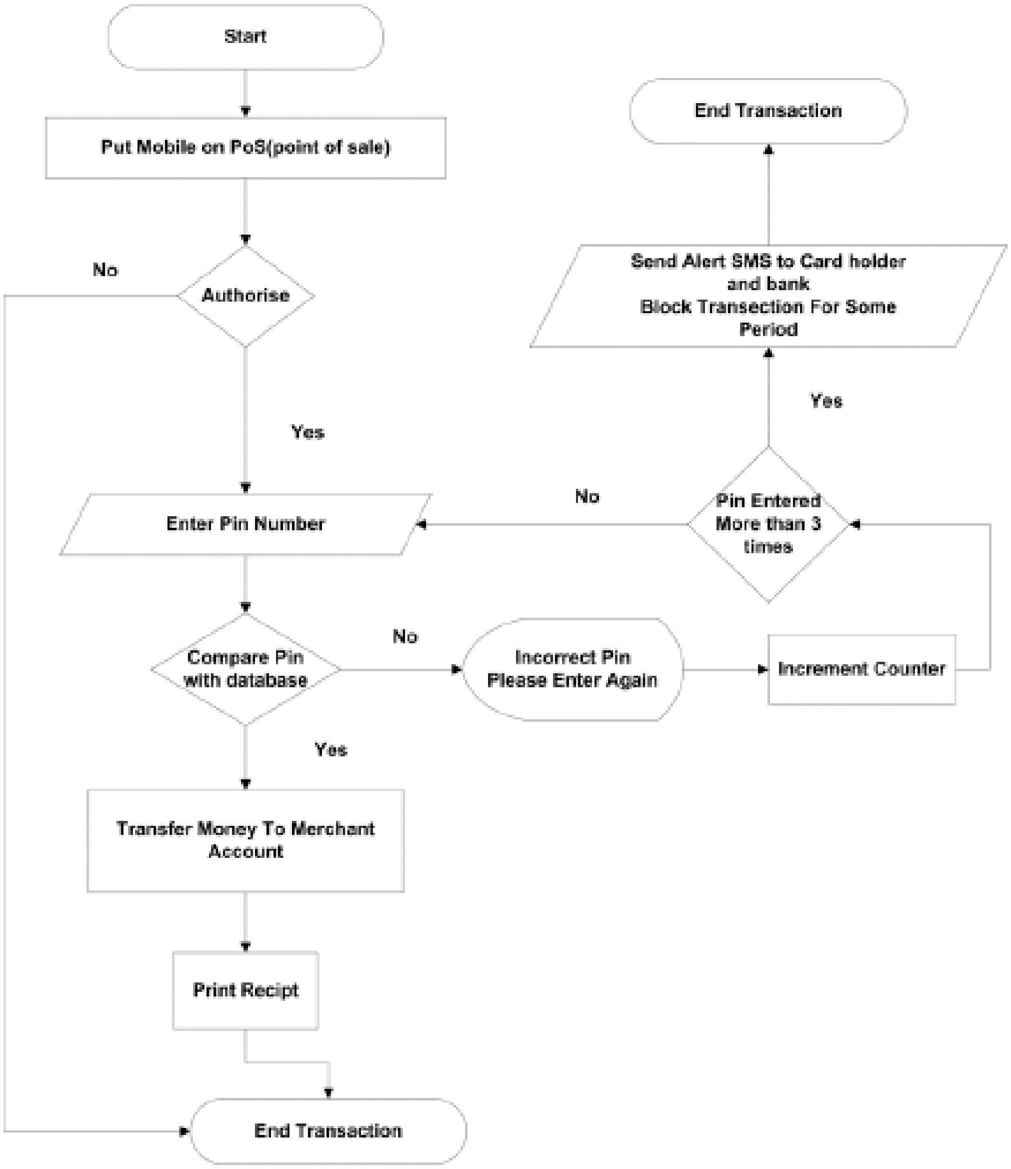

Start
Put Mobile on PoS(point of sale)
No
Authorise
Yes
Enter Pin Number
Compare Pin with database
No
Incorrect Pin Please Enter Again
Increment Counter
Pin Entered More than 3 times
No
Yes
Send Alert SMS to Card holder and bank Block Transection For Some Period
End Transaction
Yes
Transfer Money To Merchant Account
Print Recipt
End Transaction

Figure 5: NFC Payment Flow Chart [6]

attack [7]. This process involves utilizing a jammer to disturb communication or intercept packets when two NFC-enabled devices exchange statuses.

3.2 Man-in-the-Middle

MitM attacks necessitate the presence of a threat actor near the NFC-enabled device, equipped with an emulator card device and a reader device [8]. The attacker intercepts commands from the legitimate card reader using the emulator card device, relays them to the malicious reader device, and forwards responses from the victim back to the legitimate reader. This can result in a threat actor taking information received and modifying it to whatever degree is needed to achieve their goals; usually including but not limited to data corruption, data modification, and data insertion [3, 6].

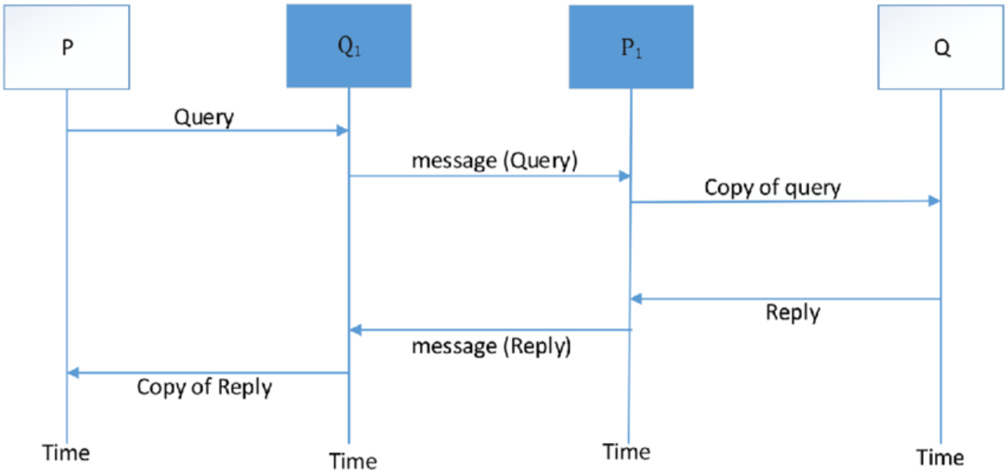


Figure 6: Diagram of MitM attack [9]

4 Countermeasures

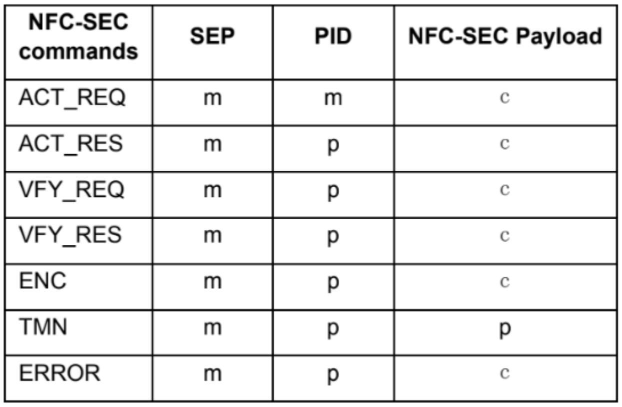
4.1 NFC-SEC

NFC-SEC's designated purpose is to provide confidentiality and integrity to NFC communications utilizing the Elliptic Curve Diffie-Hellman (ECDH) protocol for key agreement and Advanced Encryption Standard (AES) for data encryption and integrity [10]. NFC-SEC is defined in European Computer Manufacturers Association (ECMA)-385 and ECMA-386 published by Standards ECMA International. The peer-to-peer mode only operates at the link layer in the Open Systems Interconnection (OSI) reference model referenced by ISO/IEC 7498-1, leaving reader-writer and card emulation mode vulnerable to eavesdropping attacks. NFC-SEC works by utilizing three layers: NFC-SEC Users, NFC-SEC, and NFC [11]. NFC-SEC Users will request and access the NFC-SEC services through NFC-SEC Service Access Points (NFC-SEC-SAP). NFC-SEC will obtain an NFC-SEC Simple Data Exchange Protocol Data Unit (NFC-SEC-SDU) request from the NFC-SEC User and will reply with an NFC-SEC-SDU confirmation. When communicating with another NFC-SEC device, the NFC-SEC Secure Protocol Data Unit (NFC-SEC-PDU), consisting of NFC-SEC Protocol Control Information (NFC-SEC-PCI) and a single NFC-SEC-SDU, are exchanged through NFC Service Access Points (NFC-SA).

NFC-SEC utilizes Shared Secret Service (SSE) and Secure Channel Service (SCH) with the NFC-SEC User. SSE will establish a secret key through key agreement and key confirmation. Key agreement is established using `ACT_REQ` and `ACT_RES` commands. Key confirmation will be verified by `VFY_REQ` and `VFY_RES` commands. NFC-SEC-PDUs consist of 4 fields: NFC-SEC Commands, Secure Exchange protocol (SEP), Protocol Identifier (PID), and the NFC-SEC

Payload. Each field is either mandatory (m), prohibited (p), or conditional (c). SEP is a one (1) byte value determining whether the NFC-SEC-PDU is part of an SSE or SCH exchange. The NFC-SEC-PDU type and Reserved for Future Use (RFU) bits are set to zero (0) or are otherwise rejected by the receiver. SCH will establish a link key that is derived from the SSE's key agreement and key confirmation process. The sequence integrity process is as follows:

1. Each NFC-SEC entity maintains its Sequence Number Variable (SNV).
2. When SCH establishment occurs, the receiver will initialize its SNV with the same initial value as the sender's SNV, as specified in the NFC-SEC cryptography part.
3. The NFC-SEC cryptography part defines the PID and will specify a range of the SNV values.
4. When Encrypted Packet PDU (ENC) is sent, the NFC-SEC entity increases its SNV by one (1) and enters it into the Sequence Number (SN) field.
5. The SN field is protected by the NFC-SEC-PDU security mechanism.
6. When ENC is received, the NFC-SEC entity extracts the SN field and compares it with its SNV. If SN equals SNV, the NFC-SEC-PDU is discarded, and the state and SNV remain unchanged.
7. The NFC-SEC entity will increase the SNV by one (1).



NFC-SEC commands	SEP	PID	NFC-SEC Payload
ACT_REQ	m	m	c
ACT_RES	m	p	c
VFY_REQ	m	p	c
VFY_RES	m	p	c
ENC	m	p	c
TMN	m	p	p
ERROR	m	p	c

Figure 7: NFC-SEC-PDU Fields [11]

4.2 Virtual Cards

Virtual cards, such as the ones used by Google Pay, are a form of countermeasure to protect card data before even beginning an NFC communication. Google Wallet will be used as an example in this case. Google Wallet is a mobile payment application that connects to the NFC tag, which allows users to enter their credit or debit card information. The application creates a virtual card upon adding a credit or debit card. The virtual card will create a new card number for the existing card and replace it in all transactions done through Google Wallet [12]. Upon use, a credit or debit card will register any payments made with the Google Wallet as a Google Wallet payment instead of the merchant since the actual card is not being used.

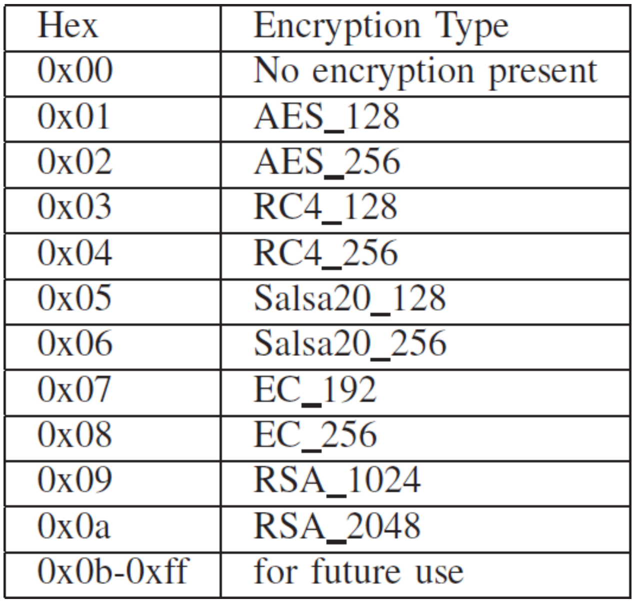
5 Proposed Authentication Methods

Due to the limited use of standards and vulnerabilities in NFC communication, alternative authentication methods have been proposed. The following section concisely reviews recent proposed solutions to ensure relevance to NFC communication today.

5.1 Encryption Record Type Definition

The primary goal of ERTD is to provide confidentiality to ndef messages, overall standardizing confidentiality-based operations in NFC [10]. This authentication method will encrypt and decrypt sensitive data in the following supported cryptography algorithms: AES, Rivest Cipher 4 (RC4), Salsa20, Elliptic Curve (EC), and Rivest-Shamir-Adleman (RSA). Each message will contain four (4) fields: Version, Key ID, Encryption Type, and Payload. The Version field is a one (1) byte field, set to 0x01, that specifies the ERTD version. The Key ID field is an eight (8) byte field that will be utilized to identify and uniquely fetch keys from the key store. This field will be used to perform encryption or decryption on the ndef record. The Encryption Type field is a one (1) byte field that specifies the cryptographic algorithm used. The payload field contains the sensitive data being sent.

The proposed ERTD system is deemed ERTD Middleware and is more lightweight regarding encrypted data size and processing time. This middleware consists of two modules: writing and reading. The writing module will ensure that ndef messages are encrypted in the NFC tag. The reading module will ensure ndef message is successfully decrypted and read before being presented to an application. These two modules comprise five (5) units: NFC Writer, Encryption Manager, Message Generator, and Decryption Manager. The NFC Writer is responsible for encrypting the ndef record and the ERTD. This process takes the Key ID, cryptographic algorithm, and format and generates an encryption record. This encryption record is appended to the encrypted ndef record, and the message is written on a tag. The Encryption Manager will fetch the encryption key from the key store using the Key ID. The encryption key and desired algorithm are used to encrypt the inputted text and then pass it onto the Message Generator. The Message Generator will have the type of ndef record, encryption algorithm, Key ID, and encrypted payload passed into it. It will then generate two records: an encrypted ndef and an ERTD. The encrypted ndef record is generated using the type and encrypted payload. The ERTD record is generated using the Key



Hex	Encryption Type
0x00	No encryption present
0x01	AES_128
0x02	AES_256
0x03	RC4_128
0x04	RC4_256
0x05	Salsa20_128
0x06	Salsa20_256
0x07	EC_192
0x08	EC_256
0x09	RSA_1024
0x0a	RSA_2048
0x0b-0xff	for future use

Figure 8: Proposed Values of Encryption Type Field [10]

ID and algorithm info. The ERTD record is appended to the encrypted ndef record and is written into the NFC tag. In the case of multiple records, a separate ndef record is generated for each of them. The Decryption Manager takes the encrypted payload, encryption type, and Key ID from the Message Parser. It will use the Key ID to fetch the decryption key from the Key Store. It will then decrypt the payload with the decryption key and encryption type information. The Decryption Manager will then construct a plaintext ndef message and forward it to the NFC application.

5.2 Bilinear Pairing

Like Hameed et al.'s proposed authentication process, Chen et al. based their bilinear pairing authentication method on an Elliptic Curve Cryptography (ECC) architecture. The motivation for using ECC in their case was that ECC provides the most efficient memory utilization and has the same strength as algorithms such as RSA but with a smaller key size [13]. NFC-enabled mobile devices have limited resources, making traditional public key cryptography algorithms difficult to compute and execute because of their heavy computation and long execution time. Chen et al.'s proposed solution offers an efficient key authentication scheme using bilinear pairing that is effective against eavesdropping or any kind of data modification or fabrication attacks. Their bilinear pairing is a computable map $\hat{e} : G1 \times G2 \rightarrow G3$, where $G1$ and $G2$ are additive cyclic groups of order n, and $G3$ is a multiplicative cyclic group of order n, satisfying properties of bilinearity, non-degeneracy, and computability.

- Bilinearity:

$$\hat{e}(aP1, P2) = \hat{e}(P1, P2)^a = e(P1, aP2)$$

$$\hat{e}(aP1, bP2) = \hat{e}(P1, P2)^{ab}$$

$$\hat{e}(P1 + P2, Q) = \hat{e}(P1, Q) \cdot e(P2, Q)$$

- Non-Degeneracy: There exists $P \in G1$, $Q \in G2$ such that $\hat{e}(P, Q) \neq IG3$ where $IG3$ is an identity element of $G3$.
- Computability: Must be efficient in computing $\hat{e}(P, Q) \forall P, Q \in G3$.

The proposed solution consists of four (4) stages consisting of eleven (11) steps, as seen in Figure 9. In Figure 9, C represents the Consumer, M represents the Merchant, and B represents a trusted Third Party, in this case, the Bank.

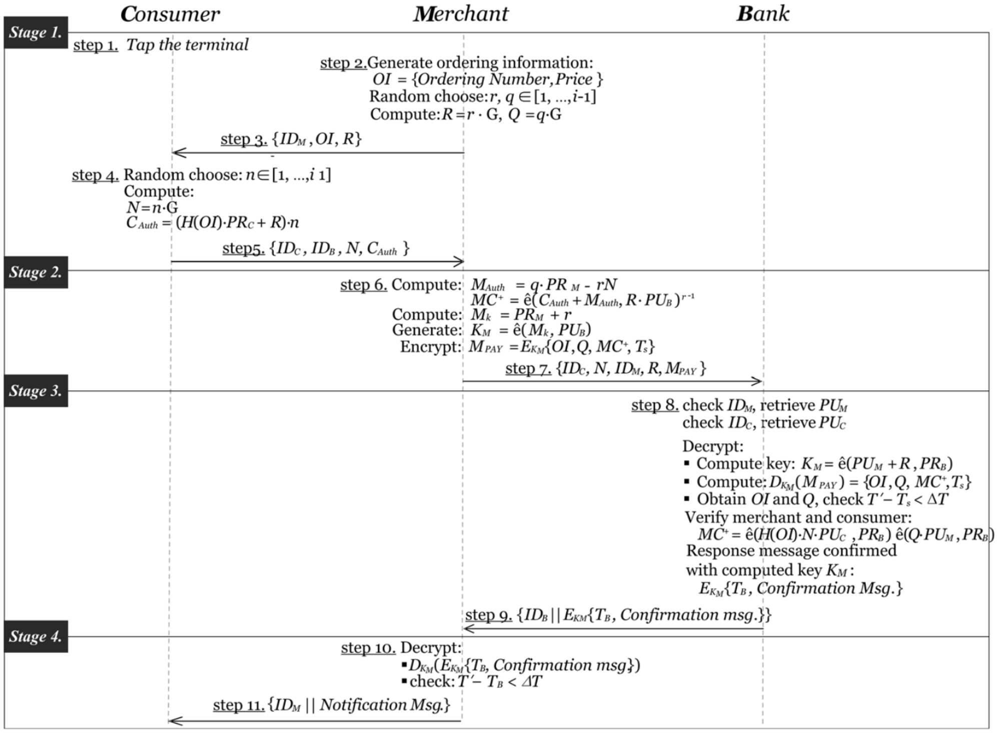

Consumer
Merchant
Bank
Stage 1.
step 1. Tap the terminal
step 2.Generate ordering information:
OI = {Ordering Number, Price }
Random choose: $r, q \in [1, ..., i\text{-}1]$
Compute: $R = r \cdot G$, $Q = q \cdot G$
step 3. $\{ID_M, OI, R\}$
step 4. Random choose: $n \in [1, ..., i\ 1]$
Compute:
$N = n \cdot G$
$C_{Auth} = (H(OI) \cdot PR_C + R) \cdot n$
step5. $\{ID_C, ID_B, N, C_{Auth}\}$
Stage 2.
step 6. Compute: $M_{Auth} = q \cdot PR_M - rN$
$MC^+ = \hat{e}(C_{Auth} + M_{Auth}, R \cdot PU_B)^{r^{-1}}$
Compute: $M_k = PR_M + r$
Generate: $K_M = \hat{e}(M_k, PU_B)$
Encrypt: $M_{PAY} = E_{K_M}\{OI, Q, MC^+, T_s\}$
step 7. $\{ID_C, N, ID_M, R, M_{PAY}\}$
Stage 3.
step 8. check ID_M, retrieve PU_M
check ID_C, retrieve PU_C
Decrypt:
▪ Compute key: $K_M = \hat{e}(PU_M + R, PR_B)$
▪ Compute: $D_{K_M}(M_{PAY}) = \{OI, Q, MC^+, T_s\}$
▪ Obtain OI and Q, check $T' - T_s < \Delta T$
Verify merchant and consumer:
$MC^+ = \hat{e}(H(OI) \cdot N \cdot PU_C, PR_B)\, \hat{e}(Q \cdot PU_M, PR_B)$
Response message confirmed
with computed key K_M:
$E_{K_M}\{T_B$, Confirmation Msg.$\}$
step 9. $\{ID_B \,||\, E_{K_M}\{T_B$, Confirmation msg.$\}\}$
Stage 4.
step 10. Decrypt:
▪$D_{K_M}(E_{K_M}\{T_B$, Confirmation msg$\})$
▪check: $T' - T_B < \Delta T$
step 11. $\{ID_M \,||$ Notification Msg.$\}$

Figure 9: Bilinear Pairing Authentication Process [13]

Stage One (1): The consumer, C, will make and send the contactless mobile payment with its authentication information to the merchant, M.

1. C brings the mobile phone close to M's POS terminal.
2. M generates the ordering information ($OI = OrderingNumber, Price$) for C's payment. M selects a random int r, q from field $[1, ..., i - 1]$. M uses randomly generated point, G, to calculate the value $R = r \cdot G$, $Q = q \cdot G$.
3. M sends payment information OI with M's ID_M and random number R to C.

4. C randomly chooses an integer n, and randomly generated point G is used to calculate $N = n \cdot G$. C calculates its authentication information C_{AUTH} as "proof", which it will send to B through M.
$$C_{AUTH} = (H(OI) \cdot PR_C + R) \cdot n$$

5. C will send its ID_C with its account B's ID_B and random number N with authentication C_{AUTH} to M.

Stage Two (2): M generates and adds its own authentication data to C's authentication data and forwards the encrypted payment information to B with the ordering information.

6. M receives the message sent by C, and the following calculation will be done:
$$M_{AUTH} = q \cdot PR_M - rN$$
$$MC^+ = \hat{e}(C_{AUTH} + M_{AUTH}, R \cdot PU_B)^{r-1}$$
M_{AUTH} is authentication information constructed by M, which will be verified by Bank B later. M will generate an encryption key K_M and use it to encrypt the payment message $\{OI, Q, MC^+, T_S\}$ as M_{PAY}
$$M_K = PR_M + r$$
$$K_M = \hat{e}(M_K, PU_B)$$
$$M_{\text{PAY}} = E_{KM}\{OI, Q, MC^+, T_S\}$$

7. M forwards C's authentication info C_{AUTH} after adding its own identification $\{ID_C, N, ID_M, R, MC^+, M_{PAY}\}$, which is used to validate communication and encrypt messages to B.

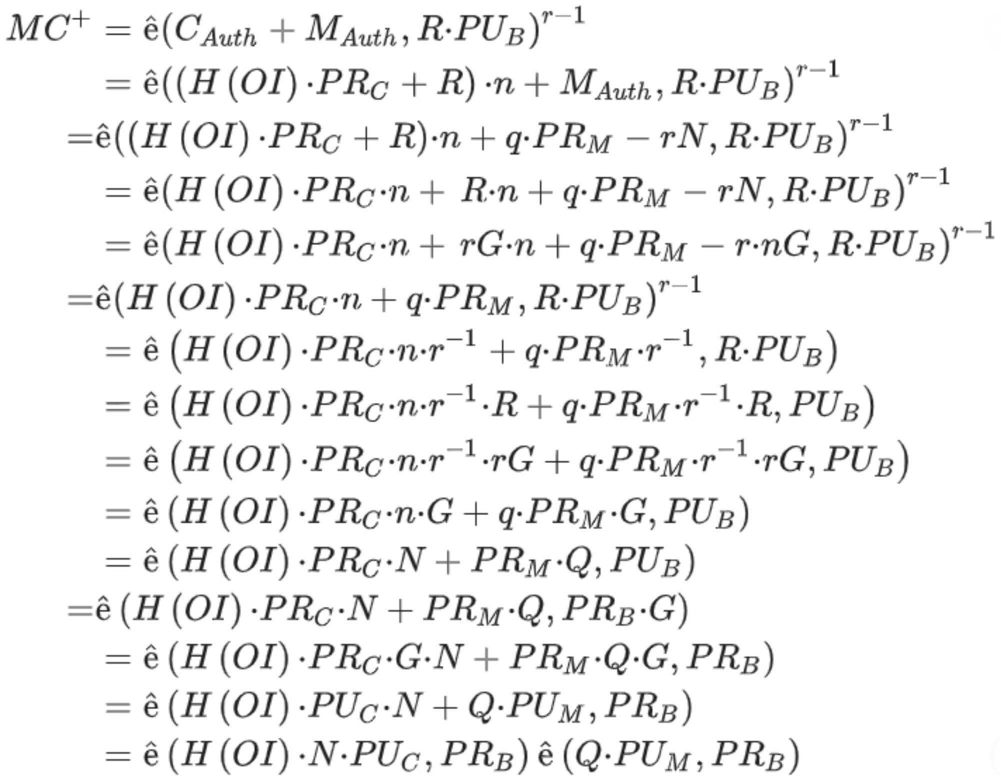
Stage Three (3): B receives authentication information sent by M and executes bilinear pairing-based exponentiation procedure to confirm the validity of the proof contained in the authentication information. If verification succeeds, a payment confirmation message is sent back to M.

8. B receives the message from M and obtains C and M's public keys using ID_C and ID_M. $\{OI, Q, T_S\}$ is obtained from decrypting message:
$$KM = \hat{e}(PU_M + R, PR_B)$$
$$D_{KM}(M_{PAY}) = \{OI, Q, MC^+, T_S\}$$
$$check : T' - T_S < \Delta T$$

B then obtains $\{OI, Q\}$ from the decrypted M_{PAY} and then uses $\{OI, Q\}$ to get the message $\{N, R, MC^{+}\}$ in order to verify C and M:

$$MC^{+} = \hat{e}(H(OI)NPU_{C}, PR_{B})\hat{e}(Q \cdot PU_{M}, PR_{B})$$

B verifies whether MC^{+} is equal to the result of the pairing using the following proof:



$$\begin{aligned}
MC^{+} &= \hat{e}(C_{Auth} + M_{Auth}, R{\cdot}PU_{B})^{r-1} \\
&= \hat{e}((H(OI){\cdot}PR_{C} + R){\cdot}n + M_{Auth}, R{\cdot}PU_{B})^{r-1} \\
&= \hat{e}((H(OI){\cdot}PR_{C} + R){\cdot}n + q{\cdot}PR_{M} - rN, R{\cdot}PU_{B})^{r-1} \\
&= \hat{e}(H(OI){\cdot}PR_{C}{\cdot}n + R{\cdot}n + q{\cdot}PR_{M} - rN, R{\cdot}PU_{B})^{r-1} \\
&= \hat{e}(H(OI){\cdot}PR_{C}{\cdot}n + rG{\cdot}n + q{\cdot}PR_{M} - r{\cdot}nG, R{\cdot}PU_{B})^{r-1} \\
&= \hat{e}(H(OI){\cdot}PR_{C}{\cdot}n + q{\cdot}PR_{M}, R{\cdot}PU_{B})^{r-1} \\
&= \hat{e}\left(H(OI){\cdot}PR_{C}{\cdot}n{\cdot}r^{-1} + q{\cdot}PR_{M}{\cdot}r^{-1}, R{\cdot}PU_{B}\right) \\
&= \hat{e}\left(H(OI){\cdot}PR_{C}{\cdot}n{\cdot}r^{-1}{\cdot}R + q{\cdot}PR_{M}{\cdot}r^{-1}{\cdot}R, PU_{B}\right) \\
&= \hat{e}\left(H(OI){\cdot}PR_{C}{\cdot}n{\cdot}r^{-1}{\cdot}rG + q{\cdot}PR_{M}{\cdot}r^{-1}{\cdot}rG, PU_{B}\right) \\
&= \hat{e}\left(H(OI){\cdot}PR_{C}{\cdot}n{\cdot}G + q{\cdot}PR_{M}{\cdot}G, PU_{B}\right) \\
&= \hat{e}\left(H(OI){\cdot}PR_{C}{\cdot}N + PR_{M}{\cdot}Q, PU_{B}\right) \\
&= \hat{e}\left(H(OI){\cdot}PR_{C}{\cdot}N + PR_{M}{\cdot}Q, PR_{B}{\cdot}G\right) \\
&= \hat{e}\left(H(OI){\cdot}PR_{C}{\cdot}G{\cdot}N + PR_{M}{\cdot}Q{\cdot}G, PR_{B}\right) \\
&= \hat{e}\left(H(OI){\cdot}PU_{C}{\cdot}N + Q{\cdot}PU_{M}, PR_{B}\right) \\
&= \hat{e}\left(H(OI){\cdot}N{\cdot}PU_{C}, PR_{B}\right)\hat{e}\left(Q{\cdot}PU_{M}, PR_{B}\right)
\end{aligned}$$

Figure 10: MC^{+} proof [13]

The proof results in the following:

$$MC^{+} = \hat{e}(C_{AUTH} + M_{AUTH}, R \cdot PU_{B})^{r-1}$$

$$= \hat{e}(H(OI) \cdot N \cdot PU_{C}, PR_{B})\hat{e}(Q \cdot PU_{M}, PR_{B})$$

If the values match, B successfully completed the verification phase. B then sends an encrypted message that contains a “confirmation message” to M. This message will be added to the payment confirmation time T_{B} and encrypted with key K_{M}:

$$D_{KM}(E_{KM}\{T_{B}, ConfirmationMsg.\})$$

9. B sends ID_{B} integrated with the encrypted message, $\{ID_{B}||E_{KM}\{T_{B}, ConfirmationMsg.\}\}$

to M.

Stage Four (4): The payment communication process is completed by sending notification information to the user.

10. M receives the confirmation message sent by B, decrypts it using K_M, and then checks T_B

$$D_{KM}(E_{KM}\{T_B, ConfirmationMsg.\})$$

$$check : T' - T_B < \Delta T$$

11. M checks the payment confirmation message, and a notice message is sent to C:

$$\{ID_M || NotificationMsg.\}$$

To prevent these attacks, this proposed solution highlights unlinkability and unforgeability. If a merchant transmits a false payment to the trusted third party using the proposed authentication method, the third party would be unable to verify the correctness of the payment amount, resulting in unforgeability. Additionally, if a threat actor intercepts data, they cannot obtain any useful or linkable information, as the data between the two devices must be unidentifiable. Despite the increased security of this proposed authentication method, ECC allows for efficiency. As detailed above, the four stages utilize a relatively small amount of computational power, each taking 112 ms, 123 ms, 227 ms, and 56 ms, respectively.

6 Conclusion

While NFC has its fair share of shortcomings for a technology of its scale, its growth has fundamentally changed contactless communication, specifically in payments and data exchange. The paper sheds light on NFC authentication challenges, predominantly due to the lack of universal authentication standards, which leaves the technology's infrastructure vulnerable. Furthermore, the paper looked into NFC specifications and functionality, exploring practical applications of the technology in areas such as tap-to-pay and current security enhancements like NFC-SEC and virtual cards. Moreover, it investigates NFC-related attacks, such as replay and MitM, and the proposed solutions, such as ERTD and bilinear pairing. It explored how ERTD aims to use cryptography to protect data in transit and how bilinear pairing utilizes ECC to provide unlinkability and unforgeability, protecting users' data. Though research has made necessary strides in the right direction, NFC technology is still in adolescence and will require much more analysis and examination to harden its security and authentication standards. By offering insights into NFC technology, this paper contributes to a deeper understanding among researchers and field practitioners, forging a path for enhanced security measures and secure implementation in NFC technology.

References

[1] P. Newswire, "Consumers demand nfc payment platforms for cryptocurrency rising at a rapid rate," *PR Newswire US*, 2019, accessed: May 21, 2019.

[2] Seritag, "NFC tag authentication explained," July 2023, seritag. https://seritag.com/learn/using-nfc/nfc-tag-authentication-explained.

[3] M. M. Singh, K. a. a. K. Adzman, and R. Hassan, "Near field communication (NFC) technology security vulnerabilities and countermeasures," *ResearchGate*, 2018, https://doi.org/10.14419/ijet.v7i4.31.23384.

[4] "Information technology - radio frequency identification for item management (ISO/IEC 18000-3:2004)," International Organization for Standardization, n.d.

[5] "ISO 14443," International Organization for Standardization, n.d.

[6] "NFC and NFC payments: A review," IEEE Conference Publication | IEEE Xplore, n.d., https://ieeexplore.ieee.org/abstract/document/7892683/authors#authors.

[7] C. Chen, I. Lin, and C. Yang, "NFC attacks analysis and survey," 2014. [Online]. Available: https://www.semanticscholar.org/paper/NFC-Attacks-Analysis-and-Survey-Chen-Lin/df8692cb5853dbc25345940963dce873ade9a643

[8] I. L. Churaev, K. Dakhkilgova, and K. Chaplaev, "NFC payment security," in *AIP Conference Proceedings*, 2021, https://doi.org/10.1063/5.0075551.

[9] A. Kumar, A. K. Jain, and M. Dua, "A comprehensive taxonomy of security and privacy issues in RFID," *Complex & Intelligent Systems*, vol. 7, no. 3, p. 1327–1347, 2021, https://doi.org/10.1007/s40747-021-00280-6.

[10] "Protecting NFC data exchange against eavesdropping with encryption record type definition," IEEE Conference Publication | IEEE Xplore, n.d., https://ieeexplore.ieee.org/document/7502861.

[11] "ECMA-385 - ECMA International," Ecma International, January 2021, https://ecma-international.org/publications-and-standards/standards/ecma-385/.

[12] "Use virtual card numbers to pay online or in apps - android - google pay help," Google Pay Help, n.d. [Online]. Available: https://support.google.com/googlepay/answer/11234179

[13] X. Chen, K. H. Choi, and K. Chae, "A secure and efficient key authentication using bilinear pairing for NFC mobile payment service," *Wireless Personal Communications*, vol. 97, no. 1, p. 1–17, 2017, https://doi.org/10.1007/s11277-017-4261-9.