RZ 3736 (# 99746) 11/05/2009
Computer Science 11 pages

Research Report

NFC-CAP
Security Assessment

Diego A. Ortiz-Yepes

IBM Research GmbH
Zurich Research Laboratory
8803 Rüschlikon
Switzerland

Research
Almaden • Austin • Beijing • Delhi • Haifa • T.J. Watson • Tokyo • Zurich

NFC-CAP

SECURITY ASSESSMENT

DIEGO A. ORTIZ-YEPES

IBM ZURICH RESEARCH LABORATORY

1. INTRODUCTION

NFC-CAP is a mobile phone based authentication mechanism for eBanking developed at the IBM Zurich Research Laboratory in partnership with Nordea. At the core of this mechanism, NFC and CAP have been used[1]. The latter, Chip Authentication Program (CAP) [CAP07], is a specification developed by MasterCard that provides mechanisms for customer authentication based on EMV (Europay - Master Card - Visa) compliant smart cards [EMV04][2]. The former, Near Field Communication (NFC), is an emerging technology related to RFID that is already being incorporated into commercially available mobile phones allowing them to communicate over very short distances (in the order of a few centimeters) with other NFC-enabled devices. Interestingly, NFC is compatible with other short range communication technologies, particularly those used by proximity cards, i.e. contactless smart cards.

NFC-CAP uses a NFC enabled mobile phone and a contactless or dual interface card in order to implement a variant of unconnected mode CAP. The phone replaces the standalone Personal Card Reader (PCR) traditionally required by CAP, communicating with the card using the NFC interface.

The rest of this document will be focused on the security aspects of the NFC-CAP solution. It will, however, not focus on the entirety of the solution, i.e. the protocols themselves as established by CAP, but rather on the security impact of replacing the PCR with the NFC mobile phone and the contact-only smart card with a contactless or dual interface smart card. Particularly, Section 2 presents an overview of how the NFC-CAP solution works, Section 3 enumerates the threats to the solution, Section 4 the attacks, and Section 5 the countermeasures taken to mitigate these threats. Section 6 concludes. Appendix 1 presents the list of acronyms and abbreviations used throughout this document, and Appendix 2 describes in detail the PIN encipherment mechanism used to send the PIN from the NFC phone to the card.

2. NFC-CAP OVERVIEW

PRINCIPALS

The principals involved in the NFC CAP system are illustrated in Figure 1, and described below.

[1] For the complete list of acronyms and abbreviations used throughout this document, refer to Appendix 1 (page 10).

[2] DPA, Dynamic Passcode Authentication, is an equivalent specification developed by Visa.

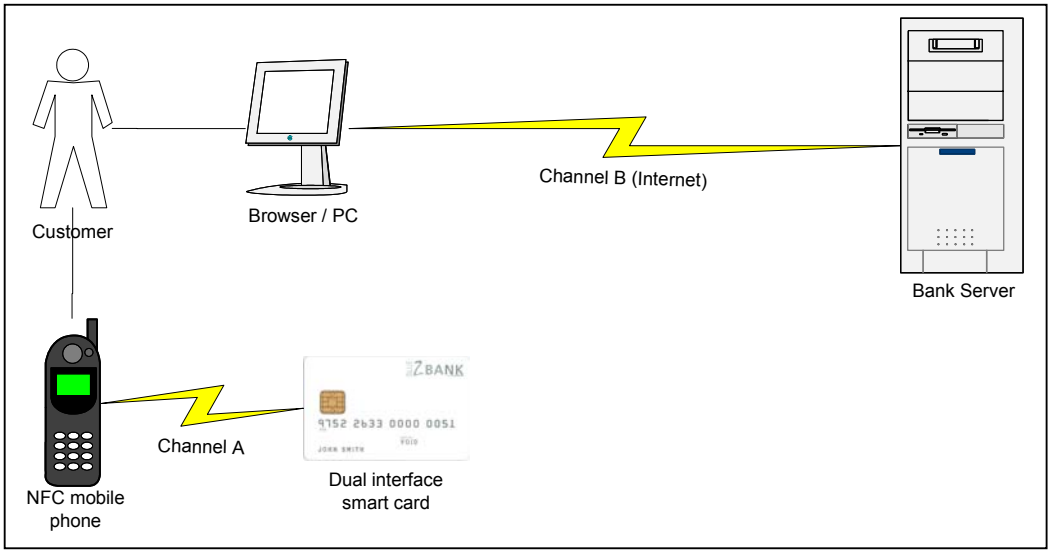

***Figure 1**. NFC CAP Principals.*

- **DUAL INTERFACE SMART CARD**

The smart card considered for this solution has both an ISO/IEC 7816 contact interface as well as an ISO/IEC 14443 contact-less interface. It should have at least one selectable EMV 4.1 [EMV04] compliant (payment) applet to be used for CAP purposes. Such an applet will be referred to as the *CAP cardlet*, or simply, the *cardlet*. The card itself may also be used for other (financially-related) purposes, e.g. as a debit or credit card.

The card's PIN is *known* by the card, the customer and the bank. Stringent procedures are assumed to be in place preventing the PIN from being disclosed or misused by bank personnel. The card stores and is able to provide the bank's public key certificate, i.e. its Issuer's Public Key Certificate [|B|], and its own PIN encipherment certificate [|C|]. Further, the private key associated to [|C|], i.e. K_C^{-1}, is securely stored in the card and cannot be used for signature generation. In fact, the sole purpose of this key is PIN decipherment (see Appendix 2, page 11 for details).

- **NFC MOBILE PHONE**

Minimally, it has a numeric keypad, a small display with graphic capabilities and NFC interface, and supports executing Java Platform Micro Edition (ME) applications[3]. It runs a CAP application capable of interacting with the card, which will be referred to as the *CAP midlet*, or simply, the *midlet*. Such a *midlet* contains two static signed lists: the first with the allowed issuers' public keys, and the second with the supported CAP *cardlet* AIDs.

- **CUSTOMER**

Also referred to as the *account holder* or *user*. It is assumed that she exercises due care with her PIN in order to keep it secret.

[3] http://java.sun.com/javame/index.jsp

- **BROWSER/PC**

It is the device from which the user accesses the Internet eBanking site. No particular assumptions are made about this equipment.

- **BANK SERVER**

Also referred to as *the server*. It corresponds to the computing equipment at the bank side that services the browsers' requests using SSL. It is assumed to be very unlikely to be compromised, both by outsiders and/or insiders. The mechanisms devised to ensure this condition fall outside the scope of this document. Finally, the *challenges* that it generates are assumed to be unpredictable.

- **CHANNEL A (CARD — PHONE)**

It is a wireless, i.e. radio frequency, communication channel operating at 13.56 MHz as specified in ISO/IEC 18092. This channel should carry information reliably over a few centimeters, after which the signal's power should decrease making the information unintelligible.

- **CHANNEL B (BROWSER/PC — BANK SERVER)**

Corresponds to the Internet.

USER AUTHENTICATION

The user authentication protocol (i.e. *Login* mode) is described below and illustrated in Figure 2.

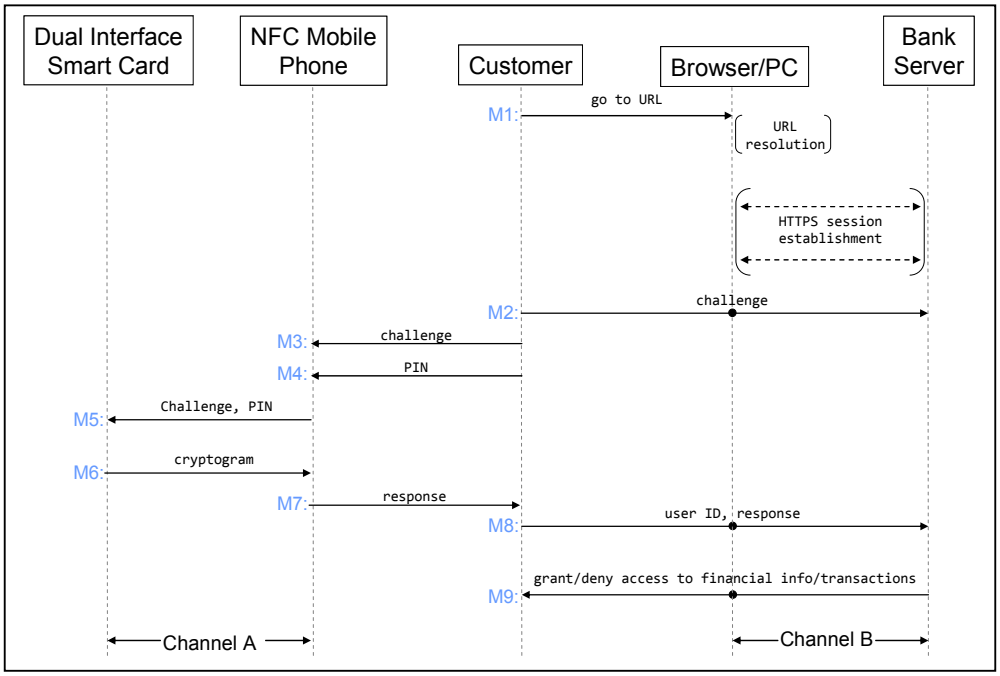

Figure 2*. User authentication protocol.*

1. The user enters the *url* of the eBanking site in the PC (M1).
2. The PC resolves the *url* and opens up the bank's site. A HTTPS session is established between the bank and the browser over channel B.
3. The server sends a form to the browser with a *customer-ID* field, and a *challenge*, which is a random number between 6 and 8 digits associated with the SSL connection (M2).
4. The user starts the *midlet* in the phone.
5. The user selects the *Log-in* mode in the *midlet* and types the challenge into the phone. (M3)
6. The customer types her PIN into the phone (M4).
7. The phone sends the *challenge* and the PIN to the card (M5). It obtains a cryptogram in return (M6). Using that cryptogram, the phone generates a code (*response*) that it displayed to the user (M7).
8. The user types in her *customer-ID* and the *response* in the appropriate fields of the web form in the PC. This form is then submitted to the server (M8).
9. The server checks that the received *response* corresponds to the issued *challenge*. If the *response* is valid, the bank presents the customer with her account(s) summary, as well as the appropriate transaction options (M9).
10. The user can perform a transaction by selecting the appropriate option and filling the necessary fields. No further user authentication takes place unless a (server defined) time out occurs.

The customer is authenticated by the bank by proving that she is able to produce an appropriate *response* to the random *challenge* sent by the bank, which is only possible by possessing a genuine bank issued smart card. Also, in order to use the keys stored in the smart card, she must provide her PIN to the card, which is in fact used to authenticate her to the card.

TRANSACTION DATA AUTHENTICATION

The transaction data authentication protocol (i.e. *Sign* mode) works as illustrated in Figure 3: After the user has been authenticated, she fills in the transaction information in her computer, which is then sent to the bank.

1. Upon receiving the transaction information (M1), the bank sends a *verification code* to the user through the browser (M2).
2. If the *midlet* has been closed after the user authentication step, the user starts it again in the phone.
3. The user selects the *Sign* mode (in the midlet) and types the *verification code* sent by the server into the phone. (M3)
4. The user types her PIN into the phone (M4).
5. The mobile phone sends the PIN and the *verification code* to the card (M5) obtaining a cryptogram in return (M6). Using that cryptogram, the mobile phone calculates a *numeric code*. The result of the calculation is displayed to the user in the mobile phone display (M7).
6. The user enters the code shown by the mobile phone in the web form, submitting it to the server (M8).

7. The server checks the validity of the code. If it is valid, the transaction is accepted and executed (M9).

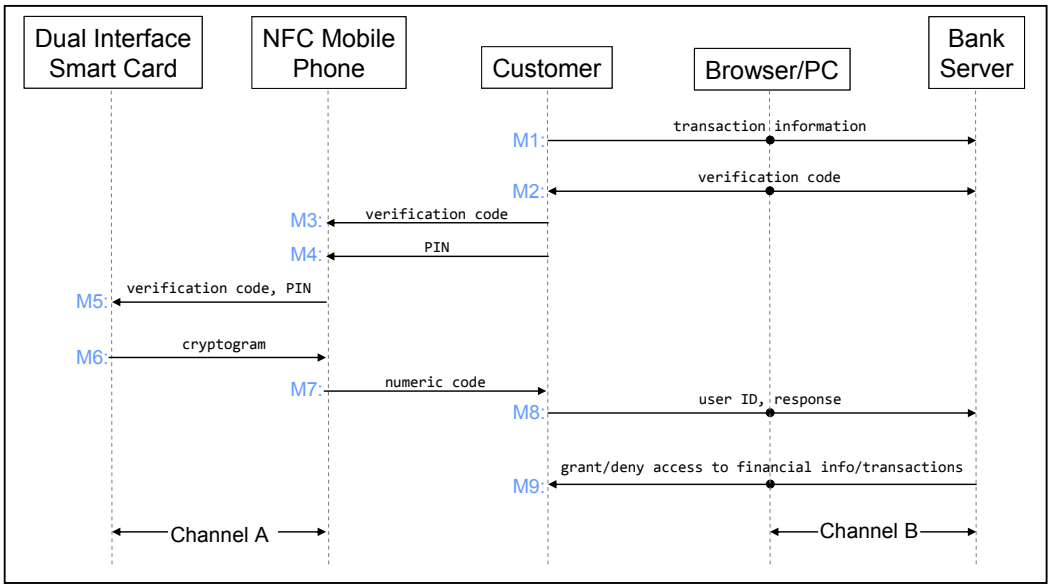

***Figure 3**. Transaction data authentication protocol.*

3. THREATS

The main threats faced by the NFC-CAP solution are:

- **PIN REVELATION.**

  The attacker manages to get hold of the PIN used to authenticate the user to the smart card. Once the attacker obtains this information, the effective level of security of the user and transaction authentication mechanisms is reduced, although not completely eliminated because the attacker still needs to get hold of the card in order to carry out a meaningful attack.

  This threat must be considered with especial attention when the card used for CAP authentication is also used as an ATM, or credit card. It is particularly relevant when a global PIN is used, and/or the card also has a magnetic stripe. This follows from the fact that by getting hold of both the card and the PIN the attacker may be able to attack using other channels, such as ATMs, POSs, etc.

- **USER IMPERSONATION**

  The attacker manages to impersonate the user to the bank, this is, fooling the bank into thinking that the user has authenticated, when in reality it is the attacker (impersonator) who has done so. This threat is relevant to the extent that it may allow the attacker to gain access to private customer information.

- **TRANSACTION INJECTION**

  The attacker manages to inject a transaction that the user did not intend to perform.

- **TRANSACTION DATA MODIFICATION**

  The attacker manages to modify the information of a transaction submitted by the user, for instance, changing its beneficiary and/or its amount.

4. ATTACKS

Attacks may be *passive* or *active*. The former occur when the attacker can only listen to, i.e. eavesdrop, the information handled by the principal. The latter, on the other hand, occur when the attacker can also can remove, modify and inject messages at will. Note that passive attacks are a subset of active attacks. Also, observe that if the adversary has to modify or alter a principal in any way in order to carry out an attack, then we will say that the attack requires a *compromise* of the principal. In our model we will consider the User to be trusted, this is, she will not collude purposefully with an attacker. Additionally, we will focus on the security impact of replacing the PCR with the mobile phone and the contact only smart card with a contactless/dual interface smart card. For this reason we shall leave out the protocols themselves as established by CAP, and shall not go in depth into PC and/or Channel B compromises[4].

The rest of this section will present the attacks associated to the threats presented in the previous section

PIN REVELATION

The PIN is only handled by four principals: The User, the Phone, Channel A, and the Card. The User is assumed to keep the secrecy of her PIN to the greatest extent possible. The Card is designed to protect and not leak the PIN. An attacker may attempt to ***replace the genuine card with a malicious one*** that captures the PIN for latter retrieval, but this is effectively thwarted by the PIN encipherment mechanism[5]. Any type of attack on Channel A is futile as a consequence of the encrypted PIN transfer mechanism outlined in Annex 2. Finally, without a ***mobile phone compromise***, an attack targeting the phone would not suffice to reveal the PIN. However, once the phone is compromised, e.g. by installing ***malicious software***, such as a key-logger, or replacing the *midlet* with a trojaned version, then a passive attack would suffice to reveal the user's PIN to the attacker.

USER IMPERSONATION

A passive attack does not suffice to impersonate the User to the Bank. This follows from the fact that the authentication credentials used for this purpose are dynamic. Consequently, even if an attacker manages to get hold of them, the chance that they can be maliciously reused is negligible. Of course, this statement is based on the assumption that the *challenges* issued by the Bank are unpredictable.

An active attack, on the other hand may allow the attacker to impersonate the user. Targeting the PC or Channel B is common to PCR based and phone based CAP, so we will not further address this avenue of attack. On the other hand, in NFC-CAP a ***phone compromise*** may

[4] For a more general view on the security of CAP see section 4 of [DRI09], and [HIL06].

[5] See the next section (Countermeasures, page 7) for details.

allow the attacker to get hold of a valid, fresh, and unused challenge/response pair by capturing the *response* as soon as the card has processed the challenge and sent back the respective reply. ***Malicious software*** installed on the phone could in principle communicate this information to any place in the world, e.g. by utilizing the GPRS network. Nevertheless, it must be noted that as the *challenge* is associated to the SSL session in place between the user's PC and the Server, the attacker must hijack this very session in order to be able to use the response generated by the phone. This means that the attacker must compromise ***both*** the phone and the PC (or **both** the phone and the network) in real time in order to impersonate the user using this attack. Although not infeasible, it must be noted that compromising two principals is harder than compromising a single one. Notwithstanding the above, it must be pointed out that just by compromising the PC or Channel B the attacker may impersonate the user regardless of the how the CAP authentication token is generated at the user side[6].

TRANSACTION INJECTION

Successfully managing to inject a new or unexpected transaction does not seem to be possible considering that the transaction authentication mechanism as presented in the previous section is in place. This follows from the fact that as soon as the Server receives the transaction data, the user will be asked to authenticate a transaction that she has not started herself, thus noticing that an attack is taking place.

TRANSACTION DATA MODIFICATION

Evidently, in order to modify the data of a transaction sent by the (legitimate) user, an active attack is required. Note also that a successful attack that manages to modify the transaction data on the PC would definitely exploit the lack of semantics of the *verification code* sent by the *Server* to be used as an input to the CAP token calculation[7], but this is an issue with the core *Sign* protocol and not the implementation, so we will not address it further.

5. COUNTERMEASURES

In order to prevent an attacker from getting hold of the User's PIN when it is transmitted between the Phone and the Card over Channel A, a standard PIN Encryption mechanism defined in [EMV04] is used. The details of this mechanism are presented in Annex 2 (page 11). Also, this mechanism prevents an attacker from getting hold of the user's PIN by replacing the genuine card with a malicious one[8]. This follows from the fact that such a card would not be able to decrypt the PIN sent by the phone. The underlying reason for this is that in order to decrypt the PIN the malicious card would require having a private key associated to a certificate issued by the trusted bank authority. Naturally, assuming issuance of authentic cards is securely managed, a malicious card would not be able to obtain such a certificate, nor read the private key from the genuine card (as such a key is effectively protected against disclosure by the ICC).

[6] The vulnerability of CAP (in general) and many other two-factor authentication systems to MITM and MITB attacks is also pointed out in [HIL06], and [SCH05].

[7] For instance, if the customer issues a transaction such as: "transfer Fr. 100 to account number 123", she will receive a verification code that may look like, for example: "54634547". This number is meaningless to her, and even though it may possibly correspond to a digest of the transaction data, she has no way to validate it. Consequently, an active attacker can mount an attack replacing the original transaction for "transfer Fr. 1000 to account number 666", which will pass unnoticed by the user, as she would get a code (relayed by the attacker) such as, for example: "38472354".

[8] Which may be a rather far fetched and improbable attack, anyway.

The smart card is also used to store the URL from which the *midlet* can be downloaded. This way, when users touch the card for the first time to their phones, they are directed to the URL from which they can get the most current *midlet*. As bank cards are sent to their users by surface mail, an attacker could in principle overwrite the download link by getting hold of the envelope containing the card at any point before it reaches its recipient. Interestingly, she would not even need to tamper with the envelope due to the fact that this can be done via the contactless interface. However, the memory area where the download URL is written is protected using standard MIFARE access control mechanisms, offering a reasonable level of protection against this attack.

It could be argued that once the customer uses the card to start the download, she could be redirected to a rogue website using—for instance—a DNS poisoning attack. Such an attack is prevented by signing the application and asking the user to confirm the signer identity prior to the application download. Naturally, if the user fails to check this information, a trojaned application could be installed instead. Furthermore, if the user leaves her phone unlocked and unattended, nothing precludes an attacker from replacing the *midlet* with such a trojaned version from an arbitrary source without user intervention.

6. CONCLUSIONS

In terms of security, NFC-CAP only seems to be worse than regular PCR-based CAP when it comes to the protection of the card PIN, which can be obtained when an attacker manages to compromise the mobile phone. Without a doubt, the complexity of the software stack running on contemporary high end phones—particularly the operating system—is increasing dramatically, and it remains a rule of thumb that complexity is the worst enemy of security. At the bottom line, compromising a phone will always be more feasible than compromising a stand-alone PCR. Nevertheless, it must be noted—for the sake of completeness—that this security limitation seems to be outweighed by the usability and cost efficiency benefits brought on by replacing the PCR with the mobile phone, which have not been discussed here due to the security-related focus of this document (see. [ORT09]).

Also, Nordea has incorporated advanced dynamic behavior in the Transaction data authentication mechanism, i.e. *Sign*. By using this mechanism in tandem with the improved UI capabilities of phones, the lack of semantics of the *verification code* sent by the Server during the transaction data authentication mechanism is eliminated, and the user can be made aware of the semantics of each of the fields that are being used into the token calculation, thus making a transaction data modification attack more difficult than with standard PCR based CAP. This follows from the fact that in order to successfully execute the attack and keep it hidden from the user, the attacker would need to compromise **both** the PC (or Channel B), and the phone.

REFERENCES

[CAP07] Master Card Worldwide. *Chip Authentication Program. Functional Architecture.* February 2007.

[DRI09] S. Drimer, S. J. Murdoch, and R. Anderson, *Optimised to fail: Card readers for online banking*, in Financial Cryptography and Data Security, February 2009.

[EMV04] EMVCo LLC. *EMV Circuit Card Specifications for Payment Systems.* Version 4.1, 2004

[HIL06] A. Hiltgen, T. Kramp, T. Weigold, *Secure Internet Banking Authentication* in IEEE Security and Privacy, v. 4, n. 2, pages 21-29. IEEE Computer Society, 2006

[ORT09] D. Ortiz-Yepes. *Enhancing Authentication in eBanking with NFC-Enabled Mobile Phones* in ERCIM News, n. 76, January 2009.

[SCH05] B. Schneier. *Two-factor authentication: Too little, Too late*, in Communications of the ACM, v. 48, n. 4, p. 136. ACM, 2005.

APPENDIX 1. ACRONYMS AND ABBREVIATIONS

AID	Application Identifier
ATM	Automatic Teller Machine
CAP	Chip Authentication Program
DNS	Domain Name Server
DPA	Dynamic Passcode Authentication
EMV	Europay - Master Card - Visa
GPRS	General Packet Radio Service
HTTP	Hypertext Transfer Protocol
HTTPS	HTTP Secure
ICC	Integrated Circuit Card
MAC	Message Authentication Code
MITM	Man In The Middle
MITB	Man In The Browser
NFC	Near Field Communication
PC	Personal Computer
PCR	Personal Card Reader
PIN	Personal Identification Number
POS	Point Of Sale
RFID	Radio Frequency Identification
RSA	Rivest-Shamir-Adleman public key cryptosystem
SSL	Secure Socket Layer/Transport Layer Security
UI	User Interface
URL	Universal Resource Locator

APPENDIX 2. PIN ENCYPHERMENT

Data exchanged over Channel A (Card — Phone) is protected against eavesdropping by the intrinsic characteristics of the radio signal used to carry the information across this channel. More precisely, the low power of the electromagnetic field generated by the phone makes it very hard to recover any information from distances greater than a few centimeters. Nevertheless, due to the fact that the PIN is transmitted to the card using this channel, there exists a risk that using specialized equipment a determined attacker might be able to compromise the PIN over this channel across a longer distance, e.g. in the order of a few meters. For this reason, the PIN is sent encrypted from the phone to the card using the PIN Encipherment mechanism defined in [EMV04]. This mechanism works as follows:

1. The phone obtains the issuer certificate [|B|] and the card's PIN encipherment certificate [|C|], checking their validity.
2. A PIN block b is constructed as shown in Figure 4, where:
   - Each square represents a nibble.
   - T is a static control field with value `0x2`.
   - N is the PIN length. It ranges from `0x4` to `0xC`.
   - P_i is the i^{th} PIN digit. Its value may range between `0x0` up to an including `0x9`.
   - F is a static filler with value `0xF`.

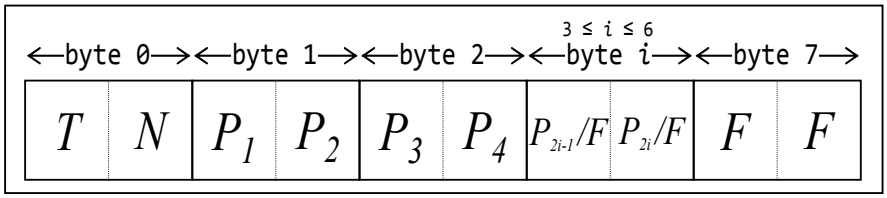

***Figure 4**. PIN block b*

3. An 8 byte random number r_C is obtained from the smart card.
4. The phone generates a $N - 17$ bytes long random bit string r_P, where N corresponds to the length in bytes of K_C, i.e. the public key associated to [|C|].
5. Let b be a static one byte data header of value `0x7F`. Then, the enciphered PIN c is calculated as:

$$c \leftarrow E[\, b \,||\, b \,||\, r_C \,||\, r_P]K_C.$$

Once the card receives c, it can recover the PIN by decrypting it using K_C^{-1}, checking b, and extracting the appropriate PIN digits using the value of N in the PIN Block b.