INVERSE PATH

EMV Vulnerabilities

Andrea Barisani
<andrea@inversepath.com>

Daniele Bianco
<daniele@inversepath.com>

INVERSE PATH

Liability shift

- liability shifts away from the merchant to the bank in most cases (though if merchant does not roll EMV then liability explicitly shifts to it)
- however the cardholders are assumed to be liable unless they can unquestionably prove they were not present for the transaction, did not authorize the transaction, and did not inadvertently assist the transaction through PIN disclosure
- PIN verification, with the help of EMV, increasingly becomes "proof" of cardholder presence

Liability shift

Canadian Imperial Bank of Commerce (CIBC) spokesman Rob McLeod said in relation to a $81,276 fraud case: “our records show that this was a chip-and-PIN transaction. This means [the customer] personal card and personal PIN number were used in carrying out this transaction. As a result, [the customer] is liable for the transaction.”

The Globe and Mail, 14 Jun 2011

INVERSE PATH

EMV is broken

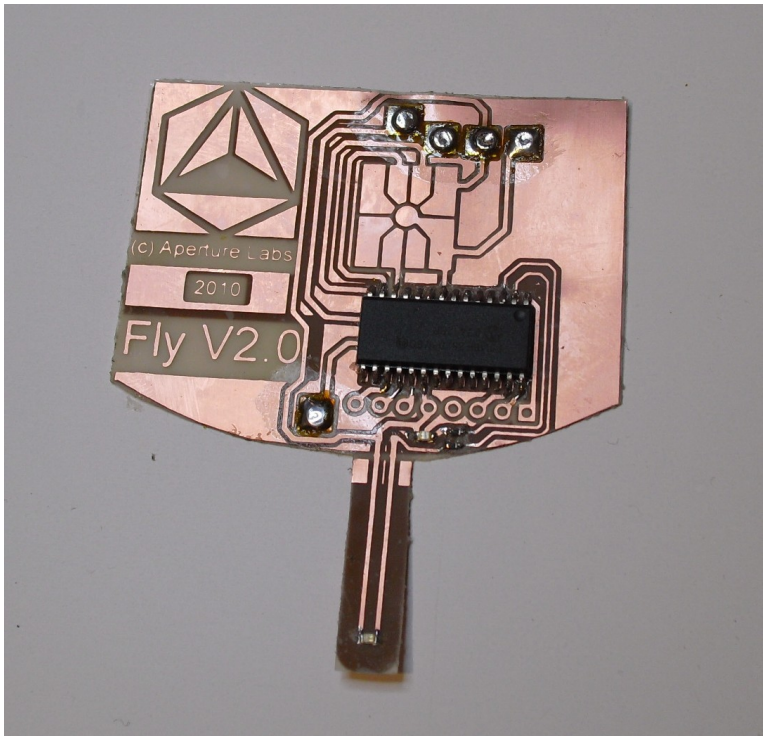
- S. J. Murdoch, S. Drimer, R. Anderson, M. Bond, "Chip and PIN is Broken" - University of Cambridge (stolen cards can be successfully used without knowing the PIN)
- A. Barisani, D. Bianco, A. Laurie, Z. Franken, "Chip & PIN is definitely broken" (PIN harvesting on all kind of EMV cards)
- M. Bond, O. Choudary, S. J. Murdoch, S. Skorobogatov, R. Anderson"Chip and Skim: cloning EMV cards with the pre-play attack" - University of Cambridge

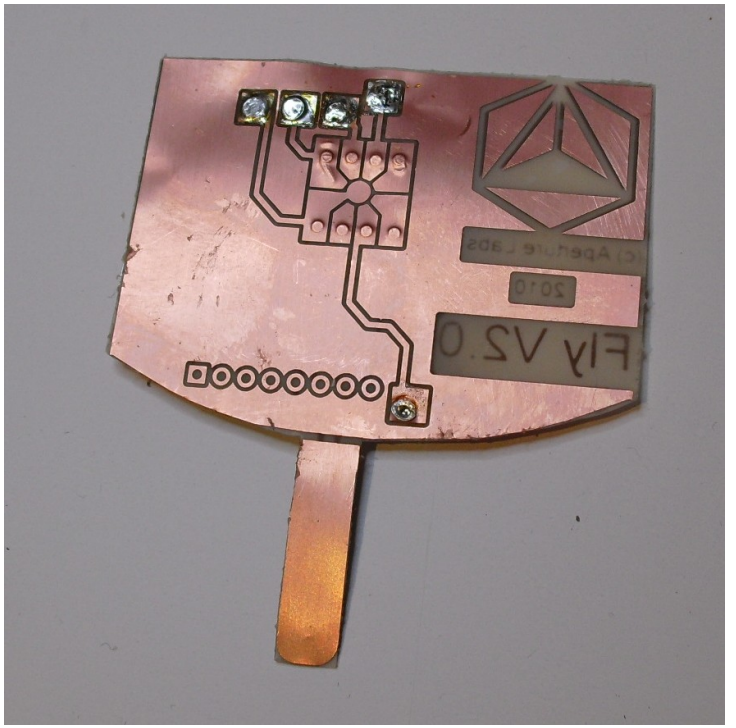
INVERSE PATH

ATM skimmers

INVERSE PATH

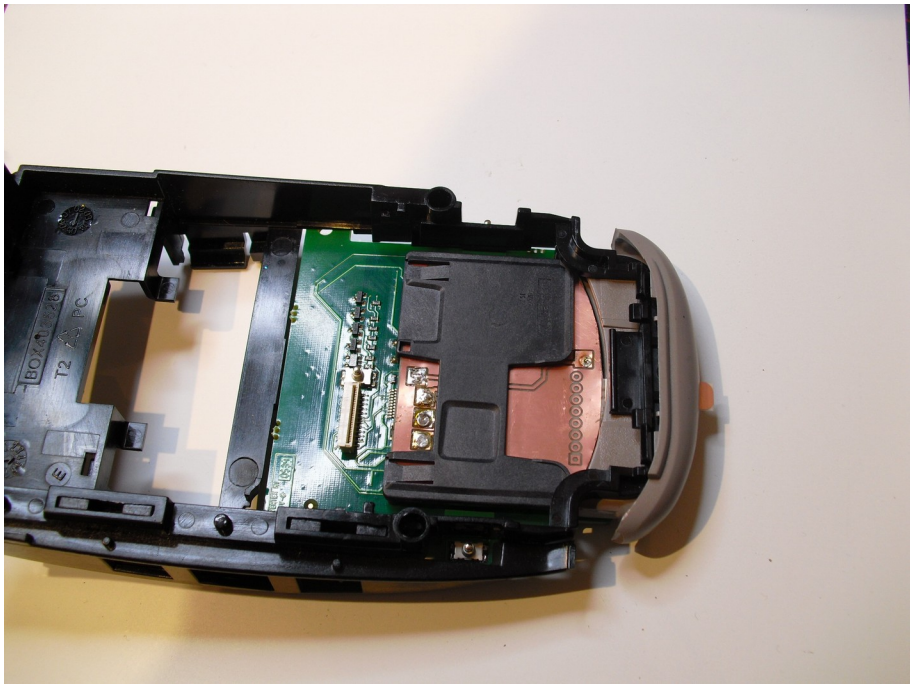
EMV skimmers (research)

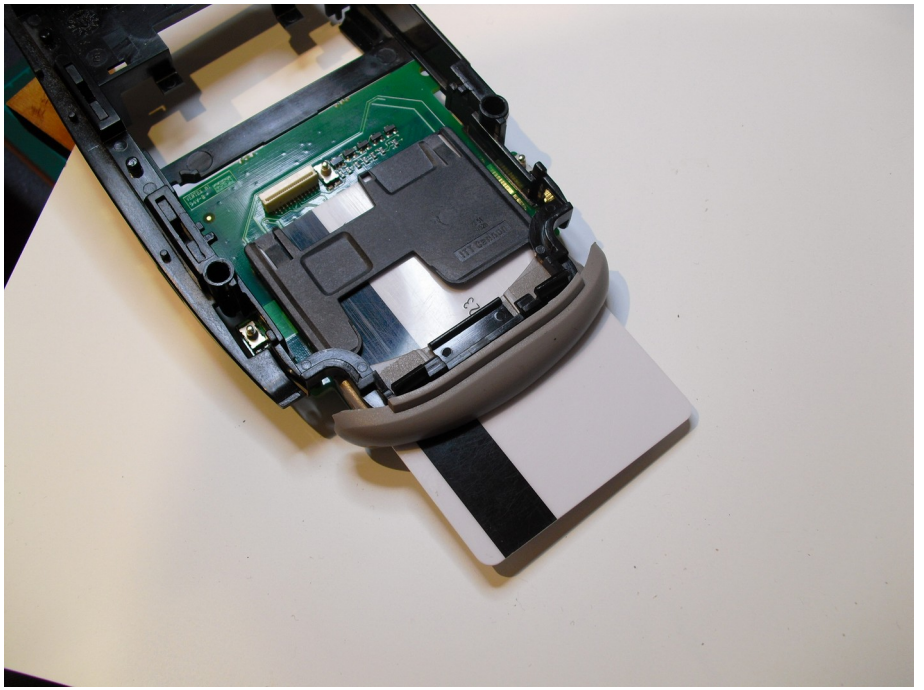




INVERSE PATH

EMV skimmers (research)



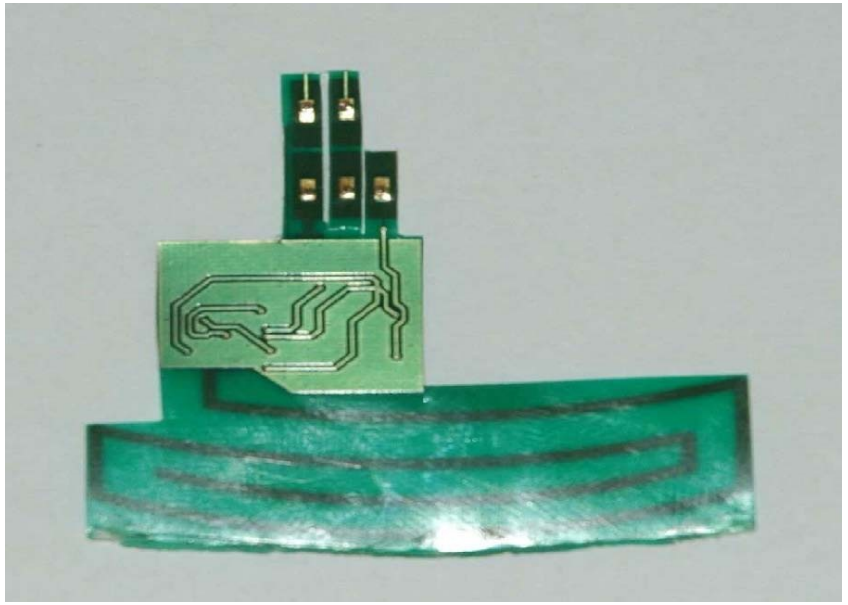


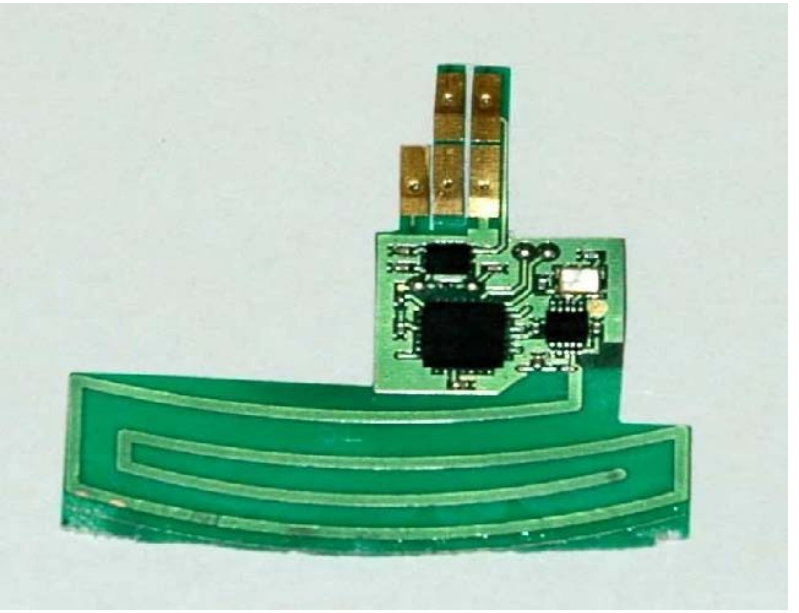
INVERSE PATH

chip skimmer installations dated 2008 have been reported in the wild by law enforcement authorities after our “Chip & PIN is definitely broken” presentation was made available

INVERSE PATH

EMV skimmers (practice)





INVERSE PATH

EMV skimmer

- trivial installation by “hooking” with a special card
- powered by the POS itself
- data can be downloaded with a special card recognized by the skimmer
- little development effort + cheap

INVERSE PATH

EMV smartcards

- information is stored on a filesystem organized in applications, files and records

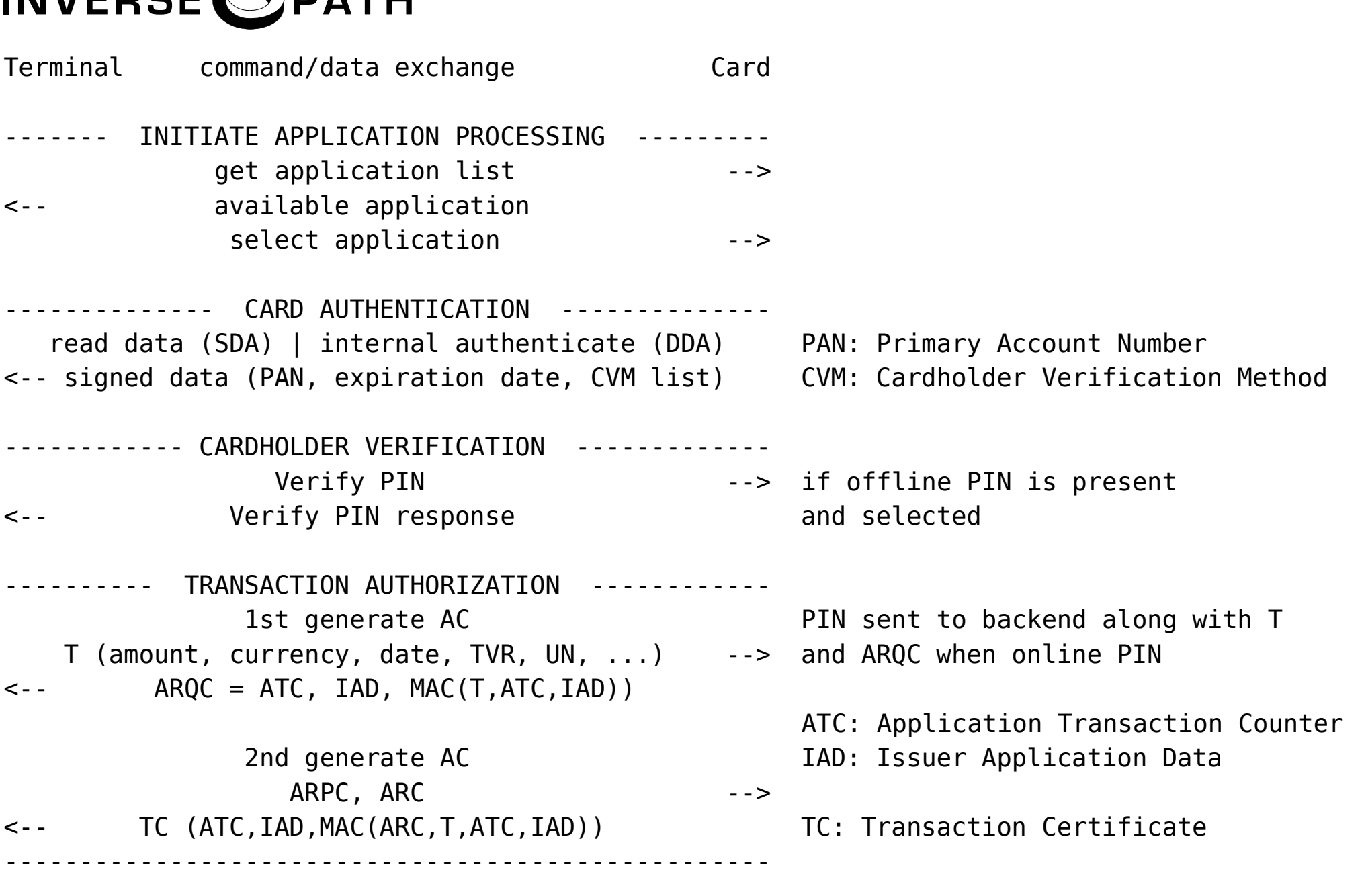
- the terminal talks to the card via APDU messages for reading records and issuing commands

```
Examples:

00A404000E315041592E5359532E4444463031 <- Select '1PAY.SYS.DDF01'
0020008008246666FFFFFFFFFF             <- Verify PIN ('6666')
```

- the EMV skimmer can intercept, read, man-in-the middle every part of the terminal <> ICC exchange

INVERSE PATH



ARQC: Authorization Request Cryptogram, ARPC: Authorization Response Cryptogram (generated by the card issuer), ARC: Authorization Response Code (authorization code for the terminal)

INVERSE PATH

The main design flaws of EMV

1. reading data, authenticating the card, authenticating the cardholder, authorizing a transaction are separate steps not securely tied together

2. the amount of unencrypted and unauthenticated data exchanged between the card and the terminal is excessive

3. the backend relies on the correct and secure operation of the terminal (which cannot be guaranteed)

INVERSE PATH

EMV application data - online usage

- application data can be used to perform Card Not Present transactions (online, phone, ...) with parties that do not check Card Security Code (CVV, CVV2, ...) and do not employ 3-D secure (Verified by Visa, MasterCard SecureCode)
- the amount of websites that do not check the security code is not negligible (Amazon)
- annoying but customer has very high chances to always get refunded, fraudster theoretically cannot cash out (only goods & services, practice is a little different however)

INVERSE PATH

Offline data authentication

- depending on the chip technology three methods are available: Static Data Authentication (SDA), Dynamic Data Authentication (DDA), Combined Data Authentication (CDA)
- used by the terminal to validate the authenticity of the card
- enables offline transactions where supported
- never used by ATM (always online)
- Visa and MasterCard mandate all cards issued after 2011 to use DDA

INVERSE PATH

Forging Offline Transactions

- the EMV standard enables the possibility of performing offline transactions without the need of immediate online interaction with the backend
- SDA and DDA fail to provide any security for offline transactions as they can be either cloned (SDA) or tampered with (DDA)
- the CDA standard has been specifically created to address such flaws but it is assumed ineffective due to backward compatibility support

INVERSE PATH

The pre-play attack - theory

- discovered by the fine people at Cambridge.
- it highlights the poor design choices made in the EMV protocol
- the Terminal generates the Unpredictable Number (UN) and not the backend
- UN prediction or manipulation effectively results in a clone for a specific transaction

INVERSE PATH

The pre-play attack - implementation

- an attacker collects the transaction data (ARQC) from a genuine transaction
- data is re-played when the UN matches
- limitation: country, amount, currency and date must match the original transaction
- a valid PIN is, theoretically, only required for ATMs (forced online verification)

INVERSE PATH

The pre-play attack - detection

- transaction backends should validate the freshness of UNs and detect implementation errors and/or pre-plays
- the implementation of Application Transaction Counter (ATC) validation would raise an alarm in case of re-use
- the specific attack limitations ease, however not conclusively, its detection

INVERSE PATH

The PIN Verification “wedge” attack - theory

- an EMV skimmer and/or “shim” manipulates the PIN verification step
- Terminal: “is this a valid PIN ?” <-- “By all means!”
- the attack is anticipated by the EMV Common Payment Application Specification (15.5.3.4 - Terminal Erroneously Considers Offline PIN OK)

INVERSE PATH

	clean run	MITM run	
READER :	00 B2 02 0C 00	00 B2 02 0C 00	
CARD :	6C 54	6C 54	
READER :	00 B2 02 0C 54	00 B2 02 0C 54	
CARD :	70 52 5F 25 03 11 08 12	70 52 5F 25 03 11 08 12	
	5F 24 03 14 09 30 5A 08	5F 24 03 14 09 30 5A 08	
	■■ ■■ ■■ ■■ ■■ ■■ ■■ ■■	■■ ■■ ■■ ■■ ■■ ■■ ■■ ■■	
	5F 34 01 01 9F 07 02 FF	5F 34 01 01 9F 07 02 FF	
	00 8E 10 00 00 00 00 00	00 8E 10 00 00 00 00 00	
	00 00 00 01 03 02 03 1E	00 00 00 01 03 02 03 1E	CVM List preferred method
	03 1F 00 9F 0D 05 B8 78	03 1F 00 9F 0D 05 B8 78	01 = plaintext PIN
	AC 80 00 9F 0E 05 00 00	AC 80 00 9F 0E 05 00 00	
	...	...	
READER :	80 CA 9F 17 00	80 CA 9F 17 00	read PIN try counter
CARD :	6C 04	6C 04	
READER :	80 CA 9F 17 04	80 CA 9F 17 04	read PIN try counter
CARD :	CA 9F 17 01 04 90 00	CA 9F 17 01 04 90 00	
READER :	**00 20 00 80 08**	***intercepted***	**verify PIN**
CARD :	**20**	**--**	
READER :	**25 ■■ ■■ ■F FF FF FF FF**	***intercepted***	**APDU verify PIN command blocked**
CARD :	**90 00**	**90 00**	**ADPU response spoofed (0x9000 == PIN OK)**
READER :	80 AE 80 00 1D	80 AE 80 00 1D	generate AC (ARQC)
CARD :	AE	AE	
READER :	00 00 00 00 08 00 00 00	00 00 00 00 08 00 00 00	generate AC (data)
	00 00 00 00 03 80 00 00	00 00 00 00 03 80 00 00	
	00 80 00 09 78 14 05 22	00 80 00 09 78 14 05 22	
	00 FC 37 B5 CE	00 FC 37 B5 CE	
CARD :	61 20	61 20	
READER :	00 C0 00 00 20	00 C0 00 00 20	get response
CARD :	C0 77 1E 9F 27 01 80 9F	C0 77 1E 9F 27 01 80 9F	
	36 02 00 28 9F 26 08 7F	36 02 00 28 9F 26 08 7F	
	76 50 22 B4 E2 0A C9 9F	76 50 22 B4 E2 0A C9 9F	CVR in the IAD (tag 9F 10) indicates that the card
	10 07 06 01 0A 03 A4 A0	10 07 06 01 0A 03 A0 A0	did not perform any PIN verification in the MITM run
	04 90 00	04 90 00	b3 in byte 5 => 06 01 0A 03 **A0** A0 04
	...	...	**^^**

INVERSE PATH

The PIN Verification “wedge” attack - detection

- the attack can be detected on the backed by correlating the Cardholder Verification Method Results (CVMR) generated by the Terminal, and the Issuer Application Data (IAD) generated by the card

- the IAD includes the Card Verification Results (CVR) that provides the card view of the transaction, unfortunately this field is vendor dependent

CVM Downgrade attack - theory

- described in our presentation *"Chip & PIN is definitely broken"*
- the CVM List is used by the Card to announce to the Terminal its cardholder verification support and preferences
- the CVM List is signed and is part of the Offline Authentication Data
- a failure in signature verification still results in Terminal application of the passed CVM List

INVERSE PATH

Cardholder verification

- the card advertises to the terminal the cardholder verification method preference via the CVM List (tag 8E)

Cardholder Verification Method (CVM) Condition Codes

Bits	Meaning	Value
8 7 6 5 4 3 2 1		
0	RFU	N/A
0	Fail cardholder verification if this CVM is unsuccessful	N/A
1	Apply succeeding CV rule if this CVM is unsuccessful	N/A
0 0 0 0 0 0	Fail CVM processing	00 or 40
0 0 0 0 0 1	Plaintext PIN verification performed by ICC	01 or 41
0 0 0 0 1 0	Enciphered PIN verified online	02 or 42
0 0 0 0 1 1	Plaintext PIN verification by ICC and signature (paper)	03 or 43
0 0 0 1 0 0	Enciphered PIN verification by ICC	04 or 44
0 0 0 1 0 1	Enciphered PIN verification by ICC and signature (paper)	05 or 45
0 0 0 1 0 1	Enciphered PIN verification by ICC and signature (paper)	05 or 45
0 x x x x x	Values in range 000110 – 011101 reserved for future use	06-1D/16-5D
0 1 1 1 1 0	Signature (paper)	1E or 5E
0 1 1 1 1 1	No CVM required	1F or 5F
1 0 x x x x	Values in range 100000 – 101111 reserved for future use	20-2F/60-6F
1 1 x x x x	Values in range 110000 – 111110 reserved for future use	30-3E/70-7E
1 1 1 1 1 1	Not available	3F or 7F

INVERSE PATH

Action Codes

- assuming a scenario with DDA only cards and a “secure” CVM List can we still harvest the PIN ?
- Issuer Action Codes (card) and Terminal Action Codes (terminal) specify policies for accepting or rejecting transactions (using TVR specifications)
- Issuer Action Codes and Terminal Action Codes are OR'ed
- three kinds: Denial, Online, Default; the Online Action Codes specify which failure conditions trigger online transactions

INVERSE PATH

Action Codes Example

```
9f0e Issuer Action Code - Denial  (5 bytes): 00 00 00 00 00
9f0f Issuer Action Code - Online  (5 bytes): f0 78 fc f8 00
9f0d Issuer Action Code - Default (5 bytes): f0 78 fc a0 00
```

- translation: “do not deny a transaction without attempting to go online, if offline SDA fails transmit the transaction online”
- in all tested terminals / cards we were able to manipulate the action codes (when necessary) so that tampering with the CVM List would not result in offline rejection

INVERSE PATH

	clean run	MITM run	
READER :	00 B2 02 0C 00	00 B2 02 0C 00	
CARD :	6C 54	6C 54	
READER :	00 B2 02 0C 54	00 B2 02 0C 54	
CARD :	70 52 5F 25 03 11 08 12	70 52 5F 25 03 11 08 12	
	5F 24 03 14 09 30 5A 08	5F 24 03 14 09 30 5A 08	
	■■ ■■ ■■ ■■ ■■ ■■ ■■ ■■	■■ ■■ ■■ ■■ ■■ ■■ ■■ ■■	
	5F 34 01 01 9F 07 02 FF	5F 34 01 01 9F 07 02 FF	
	00 8E 10 00 00 00 00 00	00 8E 10 00 00 00 00 00	
	00 00 00 **02** 03 01 03 1E	00 00 00 **01** 03 01 03 1E	**CVM List modification (01 > plaintext, 02 > enciphered)**
	03 1F 00 9F 0D 05 B8 78	03 1F 00 9F 0D 05 B8 78	
	AC 80 00 9F 0E 05 00 00	AC 80 00 9F 0E 05 00 00	
	00 00 00 9F 0F 05 B8 78	00 00 00 9F 0F 05 B8 78	
	BC 98 00 5F 28 02 03 80	BC 98 00 5F 28 02 03 80	
	9F 4A 01 82 90 00	9F 4A 01 82 90 00	
	...	...	
READER :		**80 CA 9F 17 00**	**read PIN try counter**
CARD :		**6C 04**	
READER :		**80 CA 9F 17 04**	**read PIN try counter**
CARD :		**CA 9F 17 01 04 90 00**	
READER :		**00 20 00 80 08**	**verify PIN**
CARD :		**20**	
READER :		**25 ■■ ■■ XF FF FF FF FF**	**forced plaintext PIN interception**
CARD :		**90 00**	**PIN OK**
READER :	80 AE 80 00 1D	80 AE 80 00 1D	generate AC (ARQC)
CARD :	AE	AE	
READER :	00 00 00 00 08 00 00 00	00 00 00 00 08 00 00 00	generate AC (data)
	00 00 00 00 03 80 00 00	00 00 00 00 03 80 08 00	TVR indicates a DDA failure in the MITM run
	00 80 00 09 78 14 05 22	00 80 00 09 78 14 05 22	b4 in byte 1 => 03 80 **08** 00
	00 FC 37 B5 CE	00 FC 37 B5 CE	^^
CARD :	61 20	61 20	
	...	...	

INVERSE PATH

CVM Downgrade attack - detection

- the Terminal Verification Results (TVR) indicate offline data authentication failure
- every backend we have tested accepts failures in offline data authentication without raising an alarm, additionally it is quite common to witness the lack of offline data processing
- such markers however are not conclusive evidence, as the terminal can be reset before any communication to the backend. POS firmwares must be updated to refuse offline and plaintext PIN verification in case of signature invalidation.

INVERSE PATH

The PIN Verification “wedge” attack

"We have observed variations between countries. While cards from Belgium and Estonia work like British cards, we have tested cards from Switzerland and Germany whose CVM lists specify either chip and signature or online PIN, at least while used abroad. The attack described here is not applicable to them."

Turns out it actually is...

INVERSE PATH

CVM Downgrade + PIN Verification "wedge"

- the combination of the two attacks allows using stolen cards regardless of their configuration
- this has been successfully tested on European production systems in the summer of 2014.
- it works regardless of card configuration, it has been even tested with cards that do not store any PIN on the smartcard (online only)

INVERSE PATH

	clean run	MITM run	
READER :	00 B2 02 0C 00	00 B2 02 0C 00	
CARD :	6C 54	6C 54	
READER :	00 B2 02 0C 54	00 B2 02 0C 54	
CARD :	70 52 5F 25 03 11 08 12	70 52 5F 25 03 11 08 12	
	5F 24 03 14 09 30 5A 08	5F 24 03 14 09 30 5A 08	
	■■ ■■ ■■ ■■ ■■ ■■ ■■ ■■	■■ ■■ ■■ ■■ ■■ ■■ ■■ ■■	
	5F 34 01 01 9F 07 02 FF	5F 34 01 01 9F 07 02 FF	
	00 8E 10 00 00 00 00 00	00 8E 10 00 00 00 00 00	
	00 00 00 **02** 03 01 03 1E	00 00 00 **01** 03 01 03 1E	**CVM List modification (01 > plaintext, 02 > enciphered)**
	03 1F 00 9F 0D 05 B8 78	03 1F 00 9F 0D 05 B8 78	
	...	...	
READER :		**80 CA 9F 17 00**	**read PIN try counter**
CARD :		**6C 04**	
READER :		**80 CA 9F 17 04**	**read PIN try counter**
CARD :		**CA 9F 17 01 04 90 00**	
READER :		**00 20 00 80 08**	**verify PIN**
CARD :		**20**	
READER :		**25 12 34 5F FF FF FF FF**	**APDU verify PIN command blocked**
CARD :		**90 00**	**ADPU response spoofed (0x9000 == PIN OK)**
READER :	80 AE 80 00 1D	80 AE 80 00 1D	generate AC (ARQC)
CARD :	AE	AE	
READER :	00 00 00 00 08 00 00 00	00 00 00 00 08 00 00 00	generate AC (data)
	00 00 00 00 03 80 00 00	00 00 00 00 03 80 08 00	TVR indicates DDA failed in the MITM run
	00 80 00 09 78 14 05 22	00 80 00 09 78 14 05 22	b4 in byte 1 => 03 80 **08** 00
	00 FC 37 B5 CE	00 FC 37 B5 CE	^^
CARD :	61 20	61 20	
READER :	00 C0 00 00 20	00 C0 00 00 20	get response
CARD :	C0 77 1E 9F 27 01 80 9F	C0 77 1E 9F 27 01 80 9F	
	36 02 00 28 9F 26 08 7F	36 02 00 28 9F 26 08 7F	
	76 50 22 B4 E2 0A C9 9F	76 50 22 B4 E2 0A C9 9F	CVR in the IAD (tag 9F 10) indicates that the card
	10 07 06 01 0A 03 A0 A0	10 07 06 01 0A 03 A0 A0	did not perform any PIN verification in the MITM run
	04 90 00	04 90 00	b3 in byte 5 => 06 01 0A 03 **A0** A0 04
	...	...	^^

INVERSE PATH

CVM Downgrade + PIN Verification “wedge”

DEMO

INVERSE PATH

Transaction Certificate & Receipts

- the Transaction Certificate (TC) represents a "proof of transaction" signed by the card, in the very last phase (2nd GENERATE AC)
- the TC is not immediately sent to the backend and, in our experience, never parsed until a dispute arises
- tampering with the TC does not invalidate the transaction
- in fact the TC, IAD and ATC of the last phase can all be tampered with, receipts cannot be trusted

INVERSE PATH

Fraud & Liability

Inverse Path consulted several fraud victims in the past 4 years.

To defend their claims against cardholders, banks typically leverage on the following arguments:

- PIN verification cannot be compromised with EMV
- enciphered PIN guarantees security
- on-line PIN cannot be intercepted
- plaintext PIN is kept for backwards compatibility and can only be forced with “terminal tampering” and on “specific configurations”

INVERSE PATH

Fraud & Liability

The CVM downgrade attack invalidates claims concerning PIN protection. The PIN can always be covertly intercepted leaving no traces in the backend logs.

The PIN verification “wedge” attack, and its combination with the CVM downgrade one, require careful terminal/backend log correlation to prove the absence of malicious EMV protocol manipulations.

In case of fraud and disputes the bank should provide comprehensive transaction logs, including the UN, TVR, IAD and ATC.

INVERSE PATH

What to ask for?

Attack	Effect	Marker
pre-play	transaction cloning	Unpredictable Number (UN) Application Transaction Counter (ATC)
CVM downgrade	PIN interception	Terminal Verification Results (TVR)
PIN verification “wedge”	PIN spoofing on offline PIN selected	Cardholder Verification Method (CVM) Results Issuer Application Data (IAD)
CVM downgrade + PIN verification “wedge”	PIN spoofing	Terminal Verification Results (TVR) Cardholder Verification Method (CVM) Results Issuer Application Data (IAD)

It is essential to acquire backend data opposed to solely the terminal one.

INVERSE PATH

What to ask for?

Marker	Information
Unpredictable Number	The freshness of affected terminal UNs must be proved, additionally no matching UNs from comparable transaction within the same day must be present.
Application Transaction Counter	Gaps in the ATC must be accounted for.
Terminal Verification Results	TVR must not show offline data authentication failures.
Cardholder Verification Method Results	CVM Results must agree with the CVR, included in the IAD, in relation to PIN verification.
Issuer Application Data	The CVR, included in the IAD, must agree with the CVM Results in relation to PIN verification.

INVERSE PATH

Typical dispute outcomes

1. Arbiter rules that “The burden of proof in fraud claims shall fall in the owner of the payment infrastructure”. In addition it is ruled that the PIN “can be acquired in many ways that are outside the cardholder control”.

2. Refund of fraudulent charges is processed as soon as detailed logs are requested.

INVERSE PATH

Q&A

Thanks!

www.inversepath.com