Electronic Payment Systems – case studies –

Foundations of Secure e-Commerce
(bmevihim219)

Dr. Levente Buttyán
Associate Professor
BME Híradástechnikai Tanszék
Lab of Cryptography and System Security (CrySyS)
buttyan@hit.bme.hu, buttyan@crysys.hu

Thinking Inside the Box: System-Level Failures of Tamper Proofing

Saur Drimer    Steven J. Murdoch    Ross Anderson
University of Cambridge, Computer Laboratory
15 JJ Thomson Avenue, Cambridge CB3 0FD, United Kingdom
http://www.cl.cam.ac.uk/users/{sd440,jm217,ra14}
**Thinking Inside the Box**

- PIN entry devices (PEDs) are critical security components in smartcard payment systems
  - they receive a customer’s card and PIN
  - potential to make clone cards with associated PIN

- main findings:
  - the tamper proofing of PEDs used in practice is unsatisfactory
  - the certification process used to evaluate PEDs is also unsatisfactory

- the authors implemented practical low-cost attacks on two certified (evaluated), widely-deployed PEDs (the Ingenico i3300 and the Dione Xtreme)
  - by tapping inadequately protected smartcard communications, an attacker with basic technical skills can expose card details and PINs

- these attacks highlight problems throughout the entire process of specification, design, certification and deployment
  - critical vulnerabilities arise because of the poor integration of cryptographic, physical and procedural protection
  - incentive structures of the certification process are inadequate

**Background**

- smartcards are now replacing magnetic strip cards for point of sale and ATM payments in many countries
  - increased resistance to counterfeiting

- protocol overview
  - customers authorize a transaction by inserting a bank smartcard and entering a PIN into a PIN entry device (PED);
  - the PIN is verified by the smartcard, which is in turn authenticated to the PED by a public key certificate;
  - transactions may be further authenticated online by the card issuer;

- other features
  - for backwards compatibility, cards have both a chip and magnetic strip
    - the strip is used in ATMs without chip readers or when the chip is unreadable
  - a copy of the magnetic strip is stored on the chip in its public-key certificate and is sent to terminals (for backward-compatibility reasons)
    - PEDs must therefore protect not just PINs entered by cardholders but also card details
Threat model

- potential attackers
  - merchants have free access to PEDs (as do corrupt employees);
  - customers sometimes have access for long enough to tamper with PEDs;
  - fraudsters have impersonated service engineers to gain access

→ the PED must be assumed to operate in an uncontrolled environment

- most economically important threat:
  - if the PIN and card details are intercepted when they are sent, unencrypted, between the card and PED a fake magnetic strip card may be created

- misplaced incentives:
  - the PED is purchased by the merchant from a list of devices approved by the banks, yet its role is to protect the cardholder
  - PEDs must protect valuable secrets in an unsupervised environment, while being cheap enough to be affordable by all merchants

Devices used in the study

- Ingenico i3300 and Dione Xtreme
- each of these was obtained online for under $20
  - thus, in practice, the sale of PEDs is not restricted

- both terminals have passed the Visa PED evaluation, which requires that the terminal meet one of four alternative requirements
  - defeating the tamper-detection would cost over $25,000 per-PED; or
  - inserting a PIN-stealing bug would be
    - detected, or
    - take more than ten hours, or
    - cost over $25,000

- it turns out that neither terminal meets any of these requirements
Ingenico i3300

- The Ingenico PED’s enclosure is made from two plastic shells attached to each other by four ‘Torx 6’ star-head screws.
- A tamper-response switch is released upon opening the shell, and breaks a supervisory circuit.
- One entire internal circuit board layer is a dense sensor mesh that is intended to detect drilling from the rear of the PED.
  - This sensor mesh extends to a three-sided wall that protects the switch from drilling through a user-accessible compartment.
- Additionally, four contacts are pressed by the enclosure’s top shell, so as to alarm if the keypad panel is removed.
  - The contacts are surrounded by a conductive ring connected to the battery supply; this is presumably to prevent the attacker from defeating the mechanism by injecting a conductive liquid.
- The processing module is gift-wrapped with a coarse sensor mesh and then potted.

Anti-tampering illustrated

(a) Circuit diagram: areas around pad and switch are connected to a battery to prevent injection of a conductive liquid.
Signal eavesdropping attack

- the Ingenico PED’s rear has a user-accessible compartment for the insertion of optional SIM-sized cards to expand its functionality
- they found a 1mm diameter hole in the plastic surrounding the internal compartment, through which they could access the lines carrying the data signal by using a bent paperclip
- this does not leave any external marks
- having tested this attack in the laboratory, they repeated it in the field; they tapped a terminal from a London shop and, during a transaction, extracted the card and PIN details without triggering the tamper detection system

(a) A paperclip is shown inserted through a hole in the Ingenico’s concealed compartment wall to intercept the smartcard’s data. The front of the PED is shown on the top right.
Dione Xtreme

- the Dione PED is ultrasonically sealed at seven interlocking plastic joints, and has a simple pad shorting a contact to detect opening.

- unlike the Ingenico PED, it has no mechanisms to detect drilling from the rear (the designers even provide easily accessible circuit board pads to short the tamper detection mechanism).

- however, the main processing unit and the keypad are potted together, which makes it harder to capture PIN keystrokes between the keypad and the processor.

Signal eavesdropping attack

- the Dione PED does not provide a concealed compartment to hide the wiretap, but is still vulnerable.

- by drilling a 0.8mm hole from the rear, one can insert a 4 cm needle into a flat ribbon connector socket.

(b) A needle is shown inserted through the rear of the Dione PED for data interception, attaching to a ribbon cable connector shown on the bottom left; a mounted PED is in the top right.
The full attack

- the thin white wire is connected to an FPGA board that translates the data and sends it to a laptop
- the scope and laptop screen show an ‘answer to reset’ (ATR) initial exchange intercepted using the tap
- a small FPGA or microcontroller board with some non-volatile memory can easily fit inside the Ingenico PED’s compartment and record transaction details without the cardholder’s knowledge
- a wire routed from the back of a mounted Dione PED to a recorder under the counter will not be detected unless the cardholder conducts a very close inspection
  - it has been shown that PEDs can drive 1.5m cables placed between the card slot and a card

Some conclusions

- in both designs the secure storage for cryptographic keys is fairly well protected
- however, in each case it is possible to tap the data line of the PED-smartcard interface
  - the data exchanged on this line is not encrypted
  - it yields both the information one needs to create a fake magnetic-strip card and the PIN associated with it
Shim-in-the-middle attack

- A flexible circuit board is placed between the card and card-reader contacts that transmits transaction details to a nearby receiver.
- Low profile components in the reader are used to create a simple transmitter.
- A receiver is placed nearby to record card details and PINs.
- The fraudster can create an ‘inserter card’ with the shim attached to it so that, when inserted into a particular device, the shim locks into place as the carrier card is removed.
- This attack completely bypasses all tamper protections and does not even require the participation of anyone in the store.

Encrypted PIN

- The attack reads data as it passes between PED and card ‘in the clear’.
- If both the card and PED support it, the protocol permits the PIN to be encrypted under the card’s public key.
- However, cards currently issued (in the UK) do not support this, as banks chose the low-cost option where cards do not possess the capability to do asymmetric cryptography.
- Upgraded cards will prevent a passive eavesdropper from observing the PIN (though card details are still unencrypted).
Bypassing PIN encryption

- A card advertises its support of encrypted PIN verification by placing an appropriate entry in the cardholder verification method (CVM) list, which is sent at the start of the transaction.

- However, often, the CVM list is not signed, and so may be modified – causing the PIN to be sent unencrypted.
  - This attack can be done by an active tap that selectively alters the communication, forcing a HIGH bit to LOW so that the PED thinks that the card cannot process encrypted PINs, and sends the cardholder’s PIN in the clear.

- If a full middleman scenario can be implemented, a more sophisticated attack may defeat even signed CVM lists.
  - Here, the attack device impersonates an entirely different card to the PED at the start of the transaction, and presents a CVM list that allows clear PIN entry.
  - Once the customer has entered their PIN and it has been intercepted, the attack device causes the PED to restart the normal transaction.
  - At worst, this looks like an intermittent error; in some PED implementations it may be possible to avoid alerting the customer at all.

Some conclusions

- These attacks illustrate that evaluators should consider active attacks too.
  - All of the specifications appear to consider only passive taps.

- There should be some anti-tampering measures that prevent the communication path from being broken.

- The card or PED should check if the data sent has been corrupted – a more feasible task than detecting passive taps.

- Displaying the cardholder name from the card’s certificate on the PIN entry prompt would allow some middleman attacks to be detected.
**iCVV**

- all the information needed to clone a working magnetic-strip card from a chip card is transmitted in clear during a normal transaction

- Visa has, therefore, proposed that the card verification value (CVV), a cryptographic checksum stored on the magnetic strip, be replaced with a different one – the ‘iCVV’ (CVV for Integrated Circuit Cards) – in the card’s certificate

- once iCVV would be universal a fraudster wanting to make a correct magnetic-strip card would need to recover the true CVV
  - using a ‘swipe and dock’ reader,
  - or by swiping the card in a separate reader

- despite Visa’s recommendation being made in 2002, and APACS stating that it is mandatory from January 2008, cards have still been issued in February 2008 that stored an exact copy of the magnetic strip on the chip

**Relay attacks**

- a bogus terminal in one location forwards transaction data to a bogus card in another
  - example:
    - the cardholder tries to pay two pounds to a parking-ticket machine in London, but the machine is run by a crook
    - when she gets her statement she sees a debit for twenty thousand dollars’ worth of casino chips in Macao
Countermeasures to relay attacks

- distance bounding protocols (not very practical yet)
- trustworthy user interface
  - personal card reader (pocket calculator like device) and the Chip Authentication Program (CAP) protocol:
    - the customer inserts their card into the calculator and types in their PIN, the transaction value and the recipient’s account number
    - the card then computes a code that is shown on the calculator screen and which the customer types into the POS terminal
    - the code is verified online by the bank
    - CAP keeps the card and PIN within the cardholder’s trust boundary, and provides strong authorization of the transaction, however, it may present usability problems
  - second communication channel
    - after a transaction has been placed, an SMS message is sent to the customer’s registered mobile phone, which contains the transaction amount, recipient and a code that the customer can release to authorize payment
  - RFID-based payment protocols
    - the customer’s credit card can become an application in her mobile phone
    - NFC gives a bidirectional high-bandwidth channel between phone and terminal

Lessons

- superficially, the ability to intercept PIN and card details is due to the inadequate anti-tampering mechanisms
- BUT:
  - the fact that the PIN is transferred unencrypted is due to a conscious choice by the card issuer to save money
  - even if PIN encryption were used, if the card personalization center did not sign the CVM list an attacker can still cause the PIN to be sent in the clear
  - finally, the fact that a cloned card is useful is due to magnetic-strip fallback being supported by all banks, and by issuers not implementing Visa’s iCVV recommendations
- parties may hope that the limitations of their choices will be mitigated by someone else’s work!
- the root cause of protection failure is the poor design and evaluation process
  - it’s just impossible to validate that each module enforces the security guarantees that the other parts of the system require of it, as these guarantees aren’t made explicit
Need for a security architecture document

- complex systems such as EMV should have a security architecture document that states clearly and succinctly the threats that the system is supposed to cope with and the mechanisms it will use to do so
- this document should be a public document
- even in classified systems, it must be available to implementers of every critical component, and short enough that they will read it
  - for example, the engineers implementing the Ingenico tamper mesh should have had a document that clarified the importance of protecting the serial data link from the card to the PED CPU, and make them think twice about how, where and why they route tracks and create holes/vias in the circuit board

High level abstractions

- a summary of the security boundaries should be given, showing how they are interconnected and how protected assets flow
  - example:

![High level abstractions diagram](image)

*Figure 5. Security boundaries of the payment system environment. The LCD, card reader, keypad and smartcard need to cross boundaries, shown by dotted lines, in order to complete a transaction.*
Security analysis

- given a security architecture that defines security boundaries, security analysis becomes more straightforward

- for each asset transmitted, it is easy to find all the boundaries it crosses, which provides a systematic way of working through the possibilities to establish whether those boundaries provide adequate protection

- for each container, it is possible to see if the anti-tampering mechanism properly protects both data within it, and any enclosed tamper proof boundaries

Example: PED security

"Given these definitions, it becomes clear where the flaws in EMV lie. The PIN is protected by the smartcard; given a tamper-resistant chip which will lock up after too many incorrect attempts, it cannot be directly retrieved from the card. But the PIN and card details are transmitted through the PED, which may or may not protect one or both of them. Card and PIN data may leak at the card–PED interface, and card data may also leak if the PED–bank link isn’t encrypted. Finally, since the customer cannot trust the terminal display, she is vulnerable to relay attacks."
Design constraints

- document length limit
  - if the documentation is more than a few pages long, the likelihood of a flawed system increases
  - hence, a length limit is a very reasonable requirement

- multiple protocol options
  - protocol options, and their negotiation procedures, are a significant factor behind system flaws
  - similar problems have been found in other security applications, such as the cipher-suite and version negotiation of SSL
  - where several protocol options affect the data items to be protected, all possible combinations should be set out
  - this will provide a list of scenarios that designers must check, and by making explicit the combinational complexity of options, help push back on their proliferation

Evaluation

- customers and merchants depend critically on the certification of terminals, PEDs, smartcards and other system components used in the EMV system

- some certification schemes merely ensure compatibility, such as EMV Level 1,

- there are also extensive security evaluations
  - e.g., Visa PED approval scheme
  - e.g., APACS (Association for Payment Clearing Services, UK) PED ‘Common Criteria’ Evaluation
Why evaluations fail?

- a security failure in an evaluated product can have a number of causes:
  - the evaluation framework might be defective
  - the protection profile might not specify adequate protection
  - the evaluator might miss attacks, or estimate their cost and complexity as too high

- economic reasons
  - misplaced incentives
    - often, evaluation is performed by a laboratory selected and paid by the device manufacturer
    - sometimes, the protection profile may also be defined by the device manufacturer
      - the device manufacturer will naturally select the lab that will give his product the easiest ride and charge the least money
  - market competition
    - market competition may help reduce evaluation costs, but it promotes a race to the bottom between the labs
  - regulation
    - to mitigate the race to the bottom problem, approved labs must be used
      - e.g., labs approved by Visa in case of PEDs
    - this might provide some quality control, but in practice the agencies appear never to have disapproved a licensed lab, for fear of undermining confidence in 'the system'
    - also, if the authority that approves labs is not the one that suffers from fraud, then it may not have incentives to select quality labs (moral hazard)
      - in case of EMV chip cards, it is the customer that is liable for fraudulent transactions

Example: PED evaluation

- for both of the devices used in the study, the proximate cause of evaluation failure was that the equipment just didn’t meet the protection goals set out in the Visa certification requirements

- a deeper cause was that these requirements were unrealistic:
  - given the shim attack, it’s just not clear that any compact low-cost device can be constructed that meets the requirements so the labs may have been faced with an impossible task

- the protection profile should not require that the card–PED interface be protected at all

- but the banks clearly had an incentive to pretend that it should be
  - by using cheap SDA cards rather than the more expensive DDA/CDA cards, they saved perhaps $1 per card over 70 million accounts
How to fix the problem?

- the certification process should be reengineered to take into account incentives and accountability
- ideally, evaluations would be conducted by representatives of the end users
  - but here, the cardholders and small merchants are not in a position to act collectively
- the next best option might be a hostile laboratory
  - the closest one can often get to this ideal is an evaluation by an academic lab
  - but the quantity and timeliness of these evaluations falls far short of the optimum
- vulnerability markets?
  - evaluations of equipment on which the public is forced to rely should in future come with a sufficient reward to motivate independent evaluation
    - e.g., for an evaluation at level EAL3, the authors propose a mandatory reward of $10,000
  - who would pay?
    - the authors propose that the rewards be paid not by the vendors, nor even by the evaluation labs, but by the certification bodies that license the labs

Chip and PIN is Broken

Steven J. Murdoch, Saar Drimer, Ross Anderson, Mike Bond
University of Cambridge
Computer Laboratory
Cambridge, UK
http://www.cl.cam.ac.uk/users/sjm217/sd410,rrs14,mkb23
**Chip and PIN is broken**

- smart cards have gradually replaced magnetic strip cards for point-of-sale and ATM transactions in many countries

- the authors describe and demonstrate a protocol flaw which allows criminals to use a genuine card to make a payment without knowing the card’s PIN, and to remain undetected even when the merchant has an online connection to the banking network
  - essentially a man-in-the-middle attack to trick the terminal into believing the PIN verified correctly, while telling the card that no PIN was entered at all

- the paper considers how the flaws arose, why they remained unknown despite EMV’s wide deployment, and how they might be fixed

- the results have important public policy implications

**Background**

- Chip and PIN is advertised as a solution to increasing card fraud:
  - the chip prevents card counterfeiting, and
  - the PIN prevents abuse of stolen cards

- no type of fraud has been eliminated, and the overall fraud levels have actually risen
  - EMV has simply moved fraud, not eliminated it

- in recent years, an increasing number of complaints from believable witnesses indicate that their EMV cards were fraudulently used shortly after being stolen, despite there having been no possibility that the thief could have learned the PIN

- ???
Protocol overview

- preliminaries:
  - a bank that issues EMV cards selects a subset of the EMV protocols, choosing for instance between digital signature methods, selecting a MAC algorithm, and deciding on hundreds of customizable options regarding authentication and risk management
  - merchants and acquiring banks (who receive payments on behalf of merchants) simply procure EMV-compliant hardware and software and connect it to the payment networks (operated by card schemes)

- protocol:
  - customers authorize a credit or debit card transaction by inserting their card and entering a PIN into a point-of-sale terminal
  - the PIN is typically verified by the smart card chip, which is in turn authenticated to the terminal by a digital certificate
  - the transaction details are also authenticated by a cryptographic message authentication code (MAC), using a symmetric key shared between the payment card and the bank that issued the card

---

Protocol details

<table>
<thead>
<tr>
<th>issuer</th>
<th>terminal</th>
<th>card</th>
<th>protocol phase</th>
</tr>
</thead>
<tbody>
<tr>
<td>select 1f 1PAAY.SYS.CDF01</td>
<td>available applications (e.g. Credit/Debit/ATM)</td>
<td></td>
<td>Card authentication</td>
</tr>
<tr>
<td>select application</td>
<td>start transaction</td>
<td>signed records, Sig(signed records)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>unsigned records</td>
<td></td>
</tr>
<tr>
<td>PIN retry counter</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PIN, xxx</td>
<td>PIN OK/no OK</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cardholder verification</td>
<td></td>
</tr>
<tr>
<td>T = (amount, currency, date, TVR, nonce...)</td>
<td>ARQC = [ATC, IAD, MAC(T, ATC, IAD)]</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T, ARQC</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ARPC, ASC</td>
<td>TC = [ATC, IAD, MAC(ARC, T, ATC, IAD)]</td>
<td>Transaction authorization</td>
<td></td>
</tr>
</tbody>
</table>
Cardholder verification

- starts with a mechanism negotiation to establish what cardholder authentication method the card and the terminal can (or must) use
  - cardholder verification method (CVM) list states the card’s policy on when to use a PIN, or a signature, or nothing at all, to authenticate the cardholder
  - also specifies what action should be taken if cardholder verification fails (e.g., next method should be tried)
  - in practice, the CVM list may be short: (in descending order of preference) PIN verification, signature verification, and no verification
- the vast majority of transactions are ‘PIN verified’, which means the customer enters the PIN on a PIN entry device
- the PIN is sent to the card, and the card compares it to the PIN it stores
- if they match, the card returns 0x9000, and if it fails the card returns 0x63Cx,
  - where x is the number of further PIN verification attempts the card will permit before locking up
- important: the card’s response is not directly authenticated!

Transaction authorization

- the terminal request an ARQC (authorization request cryptogram) from the card
  - the payload of this command is a description of the transaction: transaction amount, currency, type, a nonce generated by the terminal, and the TVR (terminal verification results)
- the cryptogram sent to the bank includes a type code, a sequence counter identifying the transaction (ATC – application transaction counter), a variable length field containing data generated by the card (IAD – issuer application data), and a message authentication code (MAC), computed using 3DES with a symmetric key shared between the card and the issuing bank
- the ARQC is sent by the terminal to the issuing bank
- the issuer will then perform various cryptographic, anti-fraud and financial checks
- if the checks pass, the issuer returns a two byte ARC (authorization response code), indicating how the transaction should proceed, and the ARPC (authorization response cryptogram), which is a MAC
Transaction authorization

- the card validates the MAC contained within the ARPC, and if successful updates its internal state to note that the issuer authorized the transaction
- the terminal then requests that the card issues a TC (transaction certificate) cryptogram, indicating that it is authorizing the transaction to proceed

finally, the terminal sends the TC to the issuer, and stores a copy in its own records in case there is a dispute
at this point it will typically print a receipt, which may contain the legend ‘Verified by PIN’ if the response to Verify indicated success
one copy of the receipt is given to the cardholder and a second copy is retained

The flaw

- the central flaw in the protocol is that the PIN verification step is never explicitly authenticated
- while the authenticated data sent to the bank contains the Terminal Verification Results (TVR) and the Issuer Application Data (IAD), they do not together provide an unambiguous encoding of the events which took place during the protocol run
  - the TVR mainly enumerates various possible failure conditions for the authentication, and in the event of success does not indicate which particular method was used
- therefore, a man-in-the-middle device can trick the terminal into believing that PIN verification succeeded by responding with 0x9000 to Verify, without actually sending the PIN to the card
  - a dummy PIN must be entered, but the attack allows any PIN to be accepted
- the card will then believe that the terminal did not support PIN verification, and has either skipped cardholder verification or used a signature instead
The attack

- neither the card nor the terminal will spot the attack because the cardholder verification byte of the TVR is only set if PIN verification has been attempted and failed
  - the terminal believes that PIN verification succeeded (and so generates a zero byte)
  - the card believes it was not attempted (so will accept the zero byte)

Attack demonstration

- the hardware for the attack was made of cheap off-the-shelf components and required only elementary programming and engineering skills
Attack demonstration

- the authors successfully executed the attack using several different Chip and PIN cards at a live terminal
  - a film by BBC Newsnight of them carrying out the attack is also available

Causes

- protocol design principles were not followed
  - not everything is encoded in a message that is essential to its interpretation (TVR value does not tell which cardholder verification method was used)

- but how could this flaw remain in the deployed system?
  - there was a closed design process, with no open external review of the architecture and its supporting protocols
  - large size and complexity of the specification, and its poor structure
    - core EMV protocols are now 707 pages long
    - there are a further 2126 pages of testing documentation
    - card schemes also specify extensions (Visa publishes 810 pages of public documentation, and there is more which is secret)
    - security critical details are scattered throughout, and there is no one section which is sufficient to understand the protocol, the threat model, or the security policy
  - much detail is not specified at all, being left to implementation decisions by individual issuers
    - specification does not contain enough detail to support any claims about the security of implementations, as they depend on proprietary, and often unpublished, details
Example

For example, to confirm the existence of the security vulnerability discussed in this paper, we needed to establish:

- Lack of authentication in transport layer (EMV Book 1 [11])
- Encoding of Verify (EMV Book 3 [5, p71])
- Encoding of the TVR (EMV Book 3, Annex C [3, p71])
- Recommended generation algorithm for the ARQC (EMV Book 2 [2, p89])
- Recommended transaction data items to be included in the ARQC and TC (EMV Book 2 [2, p88])
- Absence of cardholder verification result in ARQC and TC requests (EMV Book 2 [2, p73], EMV Book 3 [3, p83])
- Encoding of the CVV list (EMV Book 3, Annex C [3, p168])
- Algorithm for selecting cardholder verification method (EMV Book 3 [3, p105])
- Transaction flow (EMV Book 3 [3, p83])
- Values of the TVR (EMV Book 4 [4, p10])
- Whether the actual cardholder verification method used is included in the CDOL (unspecified, found by experiment)
- Whether the issuer checks value of IAD in online transactions (unspecified, found by experiment)
- Whether the terminal attempts to decode the IAD (unspecified, found by experiment)

A possible fix

- realities of security economics mean that one has to look for a fix which requires changes only to customer cards or to the issuer’s back-end systems

- such a repair may in fact be possible:
  - the card can change its CDOL (card data object list) to request that the CVMR (cardholder verification method results) be included in the payload to the Generate AC command that is sent by the terminal to request the ARQC
  - the CVMR specifies which cardholder verification method the terminal believes was used, and so should allow the card and issuer to identify the inconsistency

- out of many, they have only seen one EMV card which requests this field, and it is not clear that the issuer actually validates the CVMR against the IAD