Chip and PIN is Broken

To appear at the 2010 IEEE Symposium on Security and Privacy (draft)

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,rja14,mkb23}

***Abstract*—EMV is the dominant protocol used for smart card payments worldwide, with over 730 million cards in circulation. Known to bank customers as "Chip and PIN", it is used in Europe; it is being introduced in Canada; and there is pressure from banks to introduce it in the USA too. EMV secures credit and debit card transactions by authenticating both the card and the customer presenting it through a combination of cryptographic authentication codes, digital signatures, and the entry of a PIN. In this paper we 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. The fraudster performs a man-in-the-middle attack to trick the terminal into believing the PIN verified correctly, while telling the issuing bank that no PIN was entered at all. The paper considers how the flaws arose, why they remained unknown despite EMV's wide deployment for the best part of a decade, and how they might be fixed. Because we have found and validated a practical attack against the core functionality of EMV, we conclude that the protocol is broken. This failure is significant in the field of protocol design, and also has important public policy implications, in light of growing reports of fraud on stolen EMV cards. Frequently, banks deny such fraud victims a refund, asserting that a card cannot be used without the correct PIN, and concluding that the customer must be grossly negligent or lying. Our attack can explain a number of these cases, and exposes the need for further research to bridge the gap between the theoretical and practical security of bank payment systems.**

***Keywords*-EMV; Chip and PIN; card fraud; bank security; protocol failure; security economics; authentication**

I. Introduction

Smart cards have gradually replaced magnetic strip cards for point-of-sale and ATM transactions in many countries. The leading system, EMV [1], [2], [3], [4] (named after Europay, MasterCard, and Visa), has been deployed throughout most of Europe, and is currently being rolled out in Canada. As of early 2008, there were over 730 million EMV-compliant smart cards in circulation worldwide [5]. In EMV, customers authorize a credit or debit retail transaction by inserting their bank smart card and entering a PIN into a point-of-sale terminal; the PIN is verified by the smart card, 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 to the customer (the issuer).

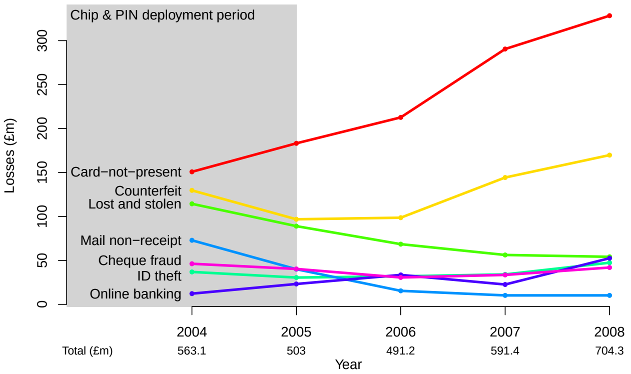


Figure 1. Fraud statistics on UK-issued cards [6]

EMV has been branded "Chip and PIN" in English-speaking countries, and was heavily promoted during its national rollout in the UK. The technology was advertised as a solution to increasing card fraud: a chip to prevent card counterfeiting, and a PIN to prevent abuse of stolen cards. Since its introduction in the UK the fraud landscape has changed significantly: lost and stolen card fraud is down, and counterfeit card fraud experienced a two year lull. But no type of fraud has been eliminated, and the overall fraud levels have actually risen (see Figure 1). The likely explanation for this is that EMV has simply moved fraud, not eliminated it.

One goal of EMV was to externalise the costs of dispute from the issuing bank, in that if a disputed transaction has been authorised by a manuscript signature, it would be charged to the merchant, while if it had been authorised by a PIN then it would be charged to the customer. The net effect is that the banking industry, which was responsible for the design of the system, carries less liability for the fraud. The industry describes this as a 'liability shift'.

Security economics teaches us that such arrangements create "moral hazard," by insulating banks from the risk of their poor system design, so it is no surprise when such plans go awry. A number of papers have documented technical attacks on EMV. However, it is now so deeply entrenched that changes can be very hard to make. Fundamental protocol changes may now require mutual agreement between banks, merchants, point-of-sale hardware manufacturers, and international card schemes (Visa, MasterCard, and American Express), all of which lobby hard to protect their interests. As with the Internet communications protocols, we are stuck with suboptimal design decisions made a decade ago. So few system changes have been made, and meanwhile the volume of customer complaints about disputed transactions continues to rise. A June 2009 survey revealed that one in five UK victims of fraud are left out of pocket [7].

In the past few years, the UK media has reported numerous cases where cardholders' complaints have been rejected by their bank and by government-approved mediators such as the Financial Ombudsman Service, using stock excuses such as 'Your card was CHIP read and a PIN was used so you must have been negligent.' Interestingly, 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.

In this paper, we describe a potential explanation. We have demonstrated how criminals can use stolen "Chip and PIN" (EMV) smart cards without knowing the PIN. Since "verified by PIN" – the essence of the system – does not work, we declare the Chip and PIN system to be broken.

II. Protocol Failure

EMV is both a protocol suite and a proprietary protocol framework: a general toolkit from which protocols can be built. In practice, it works as follows. 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 customisable options regarding authentication and risk management. Their selection must comply with card scheme rules as well as the EMV framework. Meanwhile 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).

Since we cannot enumerate the myriad of possible protocols, we predominantly describe the protocol as it is deployed within the UK, and as is implemented similarly in other countries. In particular, the attack we introduce in this paper results from a fundamental protocol failure of both the EMV framework, and a failure of the proprietary MAC protocols that are used by issuing banks (and approved by the card schemes). The attack we describe applies to most, if not all, current deployments.

As Figure 2 shows in detail, the EMV protocol can be split into three phases:

Card authentication:
  Assures the terminal which bank issued the card, and that the card's data has not been tampered with

Cardholder verification:
  Assures the terminal that the PIN entered by the customer matches the one stored on the card

Transaction authorization:
  Assures the terminal that the bank which issued the card authorizes the transaction to proceed

1) Card authentication: EMV smart cards may contain multiple separate applications with different cryptographic keys, such as a debit or credit card for use at shops, ATM functionality, and MasterCard Chip Authentication Programme (CAP) applications for online banking. Thus when a card is inserted into a point of sale terminal, it first requests a list of supported applications (by reading the file "1PAY.SYS.DDF01"), and selects one of them. The actual transaction is then initiated by sending the Get Processing Options command to the card.

Next, the terminal reads cardholder information from the card by sending a Read Record command with the appropriate file identifiers. These records include card details (e.g. primary account number, start and expiry date), backwards compatibility data (e.g. a copy of the magnetic strip), and control parameters for the protocol (e.g. the cardholder verification method list, and card data object lists, which will be discussed later).

The records also include an RSA digital signature over a subset of the records, together with a certificate chain linking the signing key to a card scheme root key known to the terminal. In one variant of EMV, known as SDA (static data authentication), the card itself is not capable of performing RSA operations, so it can only present the terminal with a static certificate. Cards employing the DDA (dynamic data authentication) variant additionally contain RSA private keys which are used to sign a nonce sent by the terminal and whose corresponding public keys are authenticated by the certificate chain.

SDA cards (which prior to 2009 all UK banks issued) are vulnerable to a trivial and well-known replay attack in which the certificate is read from a card and written to a counterfeit one (these are often called "yes cards" because they will respond "yes" to a PIN verification request, no matter what PIN is entered). The card is then used at a point-of-sale terminal which has no online connection to the banking network, and because there is no real-time interaction, the MAC produced during transaction authorization cannot be checked by the bank before the goods have already been handed over.

However, the vast majority of UK point of sale terminals maintain a permanent online connection, so such attacks

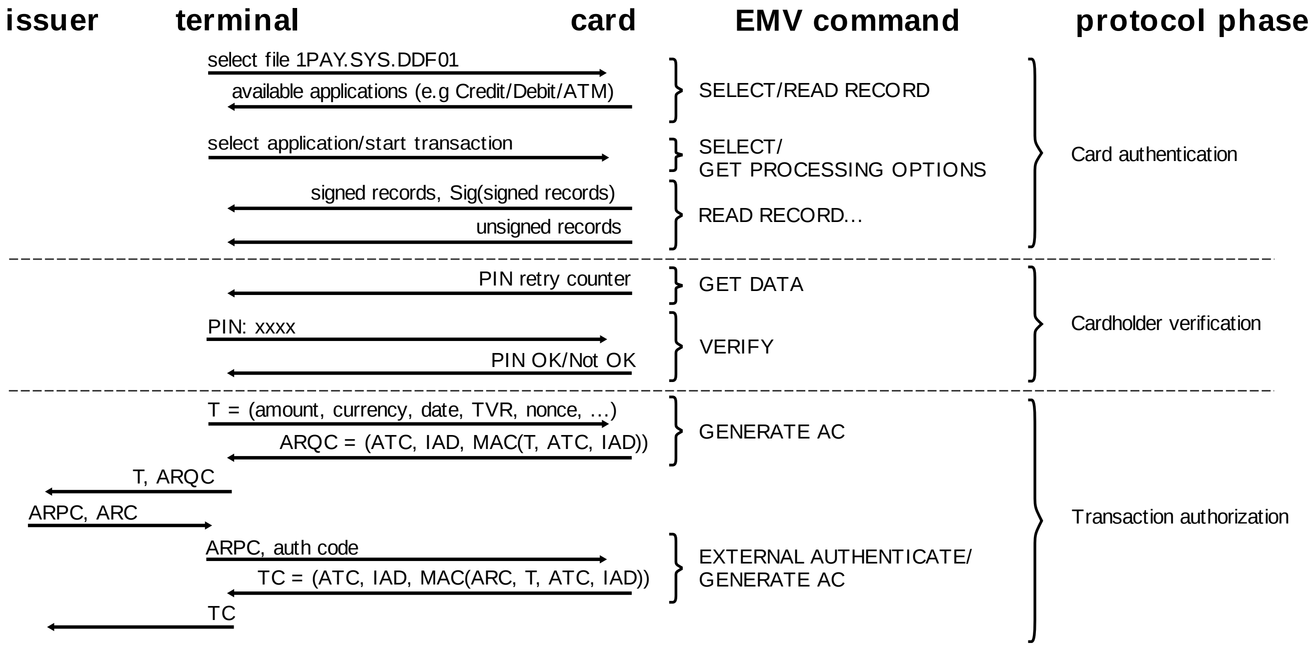


Figure 2. A complete run of a Chip and PIN protocol.

normally can be detected[1]. Since 2009, some UK banks have started issuing DDA cards, which resist counterfeiting even in offline transactions. However the attack presented in this paper does not rely upon the yes card attack; as we discuss in Section VI, it is entirely independent of card authentication, by SDA or DDA.

2) Cardholder verification: The cardholder verification step starts with a mechanism negotiation, performed between the card and the terminal, to establish what authentication method they can (or must) use. This is driven by a data element called the cardholder verification method (CVM) list. The CVM list specifies the card's policy for authenticating the cardholder.

Protocol negotiation for authentication mechanism selection is notoriously hard to get right. EMV specifies a complex negotiation algorithm by which the terminal can decide the appropriate method depending on the value of the transaction, the type (e.g. cash, purchase), and the terminal's capabilities. The CVM list also specifies what action should be taken if cardholder verification fails, i.e., whether the next method should be tried or the transaction rejected.

In practice, however, only a small subset of these capabilities is used. Cards we have examined specify, in descending order of preference, *PIN verification*, *signature verification*, and *no verification*. If a terminal is not capable of one of these, it may be skipped; for example, unattended terminals cannot do signature verification, and some vending machines are not equipped with PIN entry devices/keypads. The card itself may permit the terminal to attempt signature verification if PIN verification fails, but in practice merchants will normally reject the transaction.

The only exception is a type of card known as a "Chip & Signature" card, which does not support PIN verification. These are issued to customers who request them, normally because they are unable to remember a PIN or are visually impaired. However, 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 (optionally encrypted), and the card compares it to the PIN it stores. If they match, the card returns `0x9000`, and if it fails the card returns `0x63`*Cx*, where *x* is the number of further PIN verification attempts the card will permit before locking up. Note that the card's response is not directly authenticated.

ATM cardholder verification works differently, and uses a method known as "online PIN", as opposed to "offline PIN" described above. Here, the PIN is encrypted by the ATM, and sent to the issuer over a payment network. The issuer then compares the PIN to their own records, and sends the result back to the ATM. The attack we present in this paper only applies to offline PIN cardholder verification and therefore ATM transactions are outside of its scope.

3) Transaction authorization: In the third step, the terminal asks the card to generate a cryptographic MAC over the transaction details, to be sent to the issuing bank for

[1] There are viable criminal attack scenarios involving yes cards, and criminal business models, but these are beyond the scope of this paper.

approval. The terminal calls the Generate AC command, to request an ARQC (authorization request cryptogram) from the card. The payload of this command is a description of the transaction, created by concatenating data elements specified by the card in the CDOL (card data object lists). Typically this includes details like the transaction amount, currency, type, an unpredictable nonce generated by the terminal, and the TVR (terminal verifications results), which will be discussed later.

The cryptogram sent in response 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), which is calculated over the rest of the message including a description of the transaction. The MAC is computed, typically using 3DES, with a symmetric key shared between the card and issuer.

If the card permits the transaction, it returns an ARQC; otherwise, it returns an AAC (application authentication cryptogram) which aborts the transaction. The ARQC is then sent by the terminal to the issuing bank, via the acquirer and payment network. The issuer will then perform various cryptographic, anti-fraud and financial checks: such as check whether the card has not been listed as stolen, whether there are adequate funds in the account, and whether their risk analysis algorithm considers the transaction acceptable. 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 typically a MAC over ARQC $\oplus$ ARC. Both items are forwarded by the terminal to the card with the External Authenticate command.

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 calls Generate AC again so that the card issues a TC (transaction certificate) cryptogram, signifying 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 above description assumes that the terminal chose to perform an online transaction and contact the issuer. In event of an offline protocol, a TC is immediately produced; such offline flows are beyond the scope of this discussion, but our attack applies just as well to these.

III. The Attack

The central flaw in the protocol is that the proceedings of the PIN verification step are never explicitly authenticated. Whilst the authenticated data sent to the bank contains two fields which incorporate information about the result of the cardholder verification – 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. The TVR merely enumerates various possible failure conditions for the authentication, and in event of success does not indicate *which particular method was used* (see Figure 4).

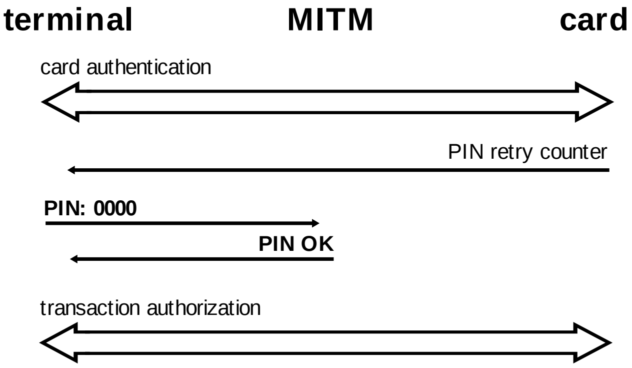


Figure 3. The man-in-the-middle suppresses the PIN Verify command to the card, and tells the terminal that the PIN has been verified correctly. A complete transaction is detailed in Appendix A.

- bit 8: 1 = Cardholder verification was not successful
- bit 7: 1 = Unrecognized CVM
- bit 6: 1 = PIN Try Limit exceeded
- bit 5: 1 = PIN entry required and PIN pad not present or not working
- bit 4: 1 = PIN entry required, PIN pad present, but PIN was not entered
- bit 3: 1 = Online PIN entered
- bit 2: Reserved for future use
- bit 1: Reserved for future use

Figure 4. Terminal verification results (TVR) byte 3.

Therefore a man-in-the-middle device, which can intercept and modify the communications between card and terminal, 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 one 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. Because the dummy PIN is never sent to the card, the PIN retry counter is not altered. The modified protocol flow is shown in Figure 3.

Neither the card nor terminal will spot this subterfuge 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

- bit 4: 1 = Issuer Authentication performed and failed
- bit 3: 1 = Offline PIN performed
- bit 2: 1 = Offline PIN verification failed
- bit 1: 1 = Unable to go online

Figure 5. IAD format, byte 5 (bits 4–1) from a Visa version 10 cryptogram [8, Appendix A-13, p222].

generates a zero byte), and the card believes it was not attempted (so will accept the zero byte).

The IAD (Figure 5) does often indicate whether PIN verification was attempted, however it is in an issuer-specific proprietary format, and not specified in EMV. Therefore the terminal (which knows the cardholder verfication method chosen), cannot decode it. The issuer, which can decode the IAD, does not know which cardholder verification method was used, and so cannot use it to prevent the attack. Because of the ambiguity in the TVR encoding, neither party can identify the inconsistency between the cardholder verification methods they each believe were used. The issuer will thus believe that the terminal was incapable of soliciting a PIN, which is an entirely plausible, yet inaccurate, conclusion.

In countries where EMV is deployed, PIN-based cardholder verification is normally mandatory for cards which support it. Although the CVM list permits merchants to fall back to signature, they rarely offer this. Therefore, unless a thief can somehow discover the PIN, using a stolen card is difficult without exploiting a vulnerability. In these scenarios, our attack could be used by criminals to carry out a point-of-sale transaction.

In fact, the authors are regularly contacted by bank customers who have had fraudulent transactions carried out shortly after their card has been stolen, and who state that they did not write down their PIN, but found that their bank accused them of negligence and refused to refund the losses. The attack we describe here may explain some of these cases.

IV. Attack Demonstration

We successfully executed the attack using several different Chip and PIN cards at a live terminal. The schematic and a photograph of the equipment used is shown in Figure 6. Stills from a video of us carrying this attack out are in Figure 7, and the full video will be available online after anonymous reviewing is complete. The hardware for the attack was made of cheap off-the-shelf components and required only elementary programming and engineering skills.

The man-in-the-middle circuit interfaces with the terminal through a fake card. This card has thin wires embedded in the plastic which connect the card's contact pads to an interface chip ($2 Maxim 1740 [9]) for voltage level-shifting. This is connected to a general-purpose FPGA board ($189 Spartan-3E Starter Kit [10]) that drives the card and converts between the card and PC interfaces. Through a serial link, the FPGA is connected to a laptop, which is in turn connected to a standard smart card reader from Alcor Micro ($8) into which the genuine card is inserted. A Python script running on the laptop relays the transaction while waiting for the Verify command being sent by the terminal; it then suppresses it to the card, and responds with `0x9000`. The rest of the communication is unaltered.

In the scenario where the merchant colludes with the attacker for a cut of the profit, the hardware bulk is not a factor. When the merchant is unwitting, the security measures introduced to protect the customer from a corrupt merchant skimming the magnetic strip work in the attackers' favour. Cardholders are instructed not to hand their card to the merchant, and the merchant is under social pressure to look away during a transaction while the cardholder enters their PIN. The attack could easily be miniaturized: it can be ported to smaller hardware devices, and would not require a PC at all if the FPGA or microcontroller is programmed to parse the transaction and interface with the card. Miniaturized hardware could be entirely hidden in a coat sleeve and used immediately after the card is stolen.

Finally, we can envision a carrier card that hosts a cutout of the original card, which interfaces with a microcontroller that communicates with the terminal. This way, the attack is entirely encapsulated in a card form factor and can be moderately industrialized. Miniaturization is mostly a mechanical challenge, and well within the expertise of criminal gangs: such expertise has already been demonstrated in the miniaturised transaction interceptors that have been used to sabotage point of sale terminals and skim magnetic strip data. Miniaturization is not critical, though, as criminals can target businesses where a card can be used with wires running up the cashout operative's sleeve, and a laptop and FPGA board can be hidden easily in his backpack. There are firms from supermarkets and money changers whose terminals are located on the other side of a barrier from the checkout staff, who as a result cannot scrutinise the cards that customers use.

V. Causes

The failure we identify here can be patched in various ways, but at heart there is a fundamental protocol failure of EMV, in compartmentalising the issuer-specific MAC protocol too distinctly from the cardholder verification method negotiation. Both of the parties who rely on transaction authentication – the merchant and the issuing bank – need to have a full and trustworthy view of the result of the cardholder verification method negotiation, and because this data cannot be neatly collected by either party, the framework itself is flawed.

A key misconception of the designers was to think of the TVR and card verification results primarily as separate lists

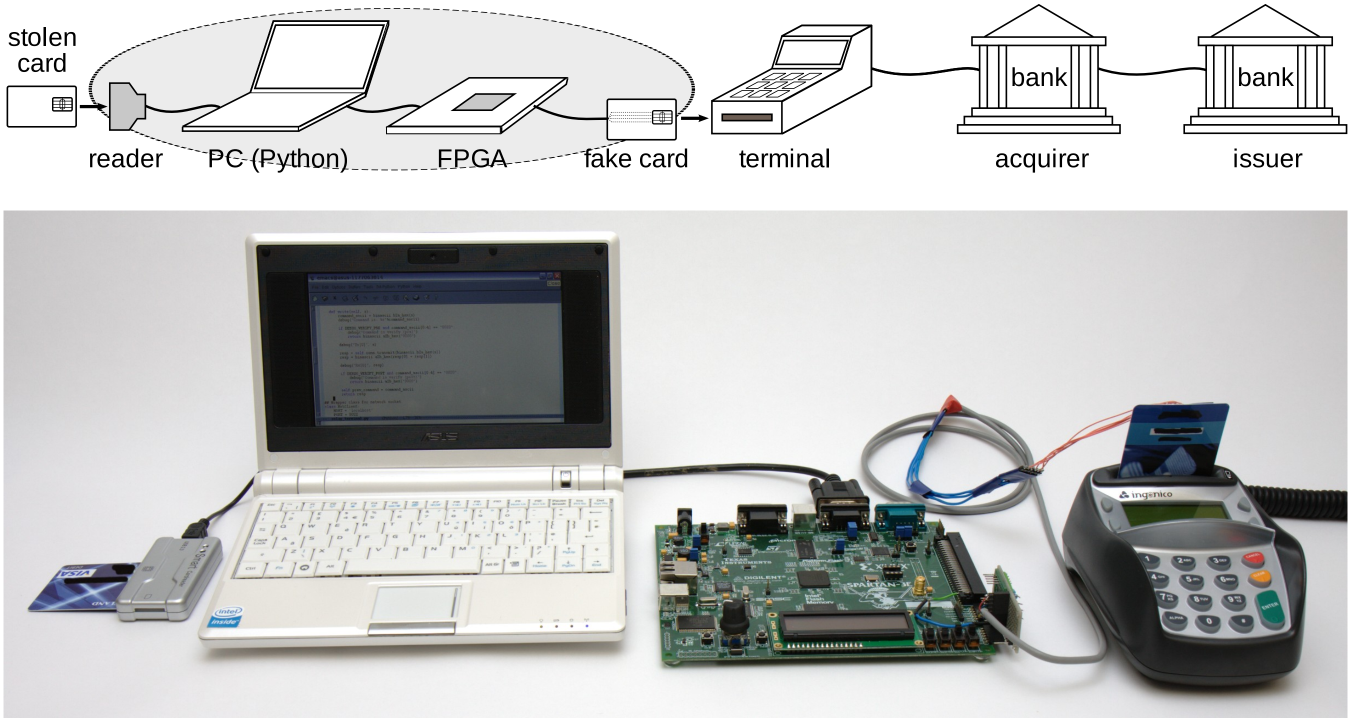


Figure 6. Components of the attack.

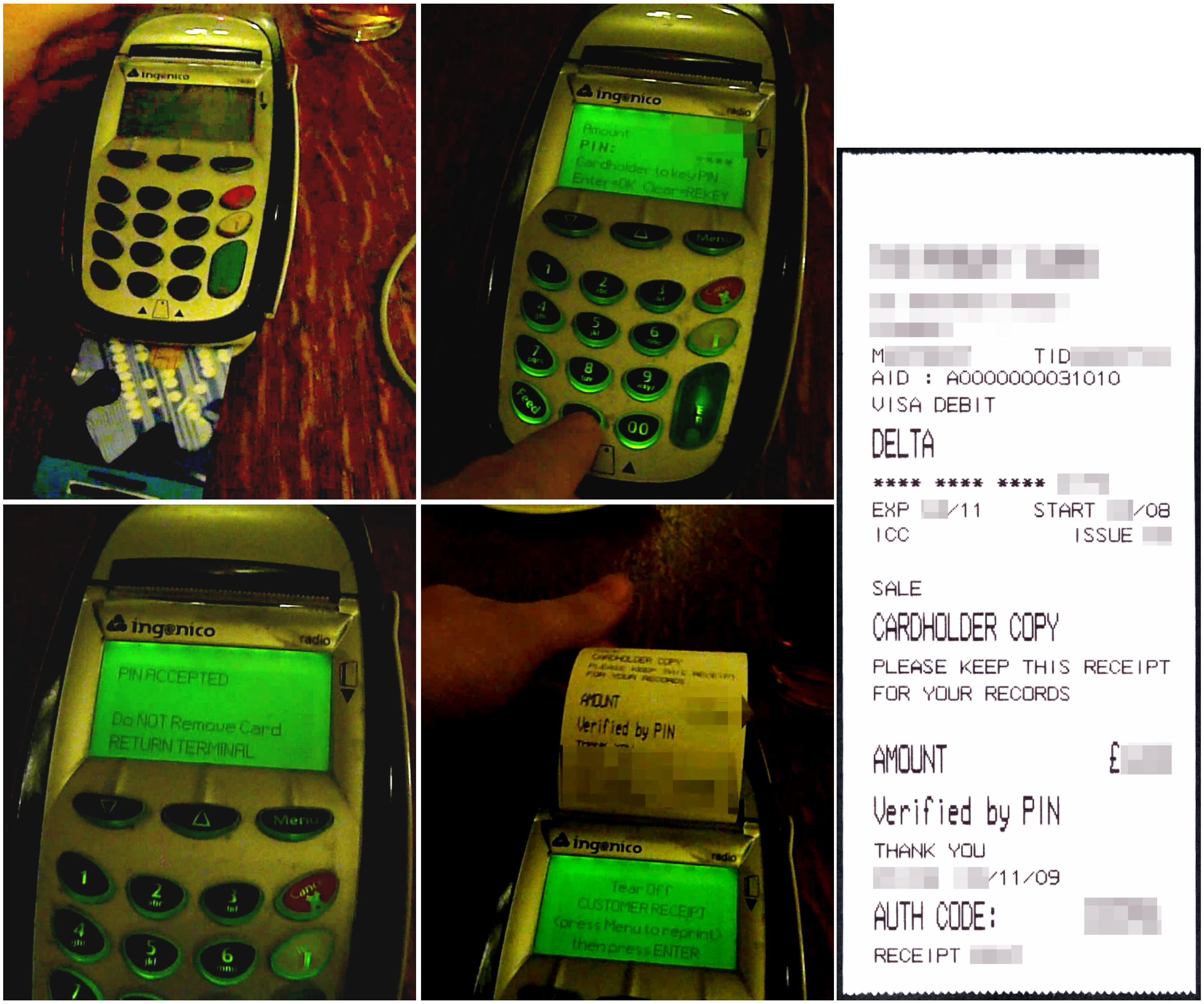
of possible failures represented by a bit mask, rather than as a report of the proceedings of the authentication protocol.

This is not to say that issuing banks cannot in future implement secure proprietary schemes within the EMV framework: because the internal protocols are proprietary anything is possible, and some potential options will be discussed in Section VI. But such schemes must make ever more complex and intricate analysis of the transaction data returned, driving up the complexity and fragility of the existing EMV card authorization systems. Essentially, they will have to ignore the framework, and without a change in the framework itself, the authorization calculations will remain so complex and dependent on external factors that further mistakes are inevitable. Also, as the protocol becomes more customized by the issuer, the introduction of new system-wide features sought for other purposes will become progressively more difficult and expensive.

The failure of EMV has many other aspects which will be familiar to security engineering. There was a closed design process, with no open external review of the architecture and its supporting protocols. The protocol documentation appeared eventually in the public domain – nothing implemented by 20,000 banks could have been kept secret – but too late for the research community to give useful feedback before commitments were made to implementation.

The economics of security also matter, and here they work out not just in the interaction between banks, customers and merchants (with the banks using their control of the system to dump liability, and thus undermining their own incentive to maintain it). There are also mismatches between acquirer and issuer banks, with only the latter feeling any real incentive to remediate security failures; between banks and suppliers, with the latter being squeezed on costs to the point that they have little incentive to innovate; and between banks and the facilities management firms to whom much of the business of card personalisation, network operation, and so on gets outsourced. The industry as a whole suffers from a significant collective action problem. It will be interesting to see which of the dozens of national bank regulators, or which of the three card schemes, will initiate action to deal with those aspects of the problems described here that cannot be tackled by issuer banks acting alone. It may be worth bearing in mind that the smart card industry spent some twenty years pitching its products to the banks before it managed to overcome the collective action problem and get the industry to move. In the absence of a catastrophe, changes that require everyone to act together are going to be slow at best.

A major contributing factor to the fact that these protocol flaws remained undiscovered is the size and complexity of the specification, and poor structure. The core EMV protocols are now 707 pages long, there are a further 2 126 pages of testing documentation, and card schemes also specify extensions (Visa publishes 810 pages of public



[redacted]
M[redacted] TID[redacted]
AID : A0000000031010
VISA DEBIT
DELTA
**** **** **** [redacted]
EXP [redacted]/11 START [redacted]/08
ICC ISSUE [redacted]

SALE
CARDHOLDER COPY
PLEASE KEEP THIS RECEIPT
FOR YOUR RECORDS

AMOUNT £[redacted]
Verified by PIN
THANK YOU
[redacted] [redacted]/11/09
AUTH CODE: [redacted]
RECEIPT [redacted]

Figure 7. Carrying out the attack. Although we entered the wrong PIN, the receipt indicates that the transaction was "Verified by PIN".

documentation, and there is more which is secret). Many options are given, and a typical implementation mixes some of the functionality from the published manuals with some issuer-specific enhancements. Security critical details are scattered throughout, and there is no one section which is sufficient to understand the protocol, overall threat model, or security policy. In fact, much detail is not specified at all, instead being left as an implementation decision by individual issuers.

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 [1])
- Encoding of Verify (EMV Book 3 [3, p71])
- Encoding of the TVR (EMV Book 3, Annex C [3, p171])
- Recommended generation algorithm for the ARPC (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, p58])
- Encoding of the CVM list (EMV Book 3, Annex C [3, p168])
- Algorithm for selecting cardholder verification method (EMV Book 3 [3, p103])
- Transaction flow (EMV Book 3 [3, p83])
- Values of the TVR for signature transaction (EMV Book 4 [4, p49])

- Whether the actual cardholder verification method used is included in the CDOL (unspecified, found by experimentation)
- Whether issuer checks value of IAD in online transactions (unspecified, found by experimentation)
- Whether terminal attempts to decode the IAD (unspecified, found by experimentation)
- Encoding of the IAD (proprietary, specified in Visa Integrated Circuit Card Specification, Appendix A [8, p222])

Ultimately EMV is a compatibility system and protocol toolkit; it allows interoperable protocols to be built but following the specification, even including the optional recommendations, does not ensure a secure protocol. This explains why there has been little analysis of EMV – the specification contains insufficient details to make any assertion about the security of implementations, as the latter relies on the details of proprietary, and often unpublished, elements. It is necessary to additionally perform experiments, as we have done. Researchers, and especially merchants who assist them, may be afraid of retribution from the banking industry, and thus make such experimentation difficult.

VI. Solutions and Non-Solutions

Core protocol failures are difficult to fix. None of the security improvements already planned by banks will help: moving from SDA to DDA will not have any effect, as these are both methods for card authentication, which occurs before the cardholder verification stage. Neither will a further proposed enhancement – CDA (combined data authentication) – in which the transaction authorization stage additionally has a digital signature under a private key held by the card. This is because the attack we present does not interfere with either the input or output of transaction authentication, so replacing a transaction MAC with a digital signature will not help.

One possible work-around is for the terminal to parse the IAD, which does include the result of PIN verification (Figure 5). This will only be effective for online transactions, and offline transactions where CDA is used, because otherwise the man-in-the-middle device could tamper with the IAD as it is returned by the card. It would also be difficult to implement because the IAD was intended only for the issuer, and there are several different formats, without any reliable method to establish which one is used by a particular card. However a solution along these lines would require the acquiring banks and the terminal vendors to act together, which for the incentive reasons discussed above would be both slow and difficult.

The realities of security economics mean that we have to look for a fix which requires changes only to customer cards and to the issuer's back-end systems. Such a repair may in fact be possible: the card can change its CDOL to request that the CVMR (cardholder verification method results) be included in the payload to the Generate AC command. This 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, we 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. Whether this fix works for a given bank will depend on their systems; we have not been able to test it, and given that it involves reissuing the issued card base it will take years to roll out.

These workarounds should resolve the particular flaw discussed in this paper, but there are likely more. A more prudent, but difficult to deploy, approach would be to follow established design principles for robust security protocols. For example, adopting the "Fail-stop" [11] principles would prevent this attack and likely others; it would also make the protocol easier to analyze. Alternatively a well-examined industry standard transport-layer confidentiality and authenticity standard, such as TLS [12], could be wrapped around the existing command set.

VII. Evidence in Chip and PIN Disputes

Even if it is infeasible to prevent the attack we present in this paper, it is important to detect whether it occurred when resolving cases where a customer disputes having made a transaction. While assisting fraud victims who have been refused a refund by their bank, we have requested the IAD so as to discover whether the card believes PIN verification succeeded, but have always been refused. This paper illustrates that while the IAD can be considered trustworthy (after its MAC has been verified), the TVR and merchant receipt must not.

In fact, dispute resolution processes we have seen in the UK are seriously flawed, even excluding the protocol failure described here. In one disputed transaction case we assisted in, the customer had his card stolen while on holiday, and then used in an EMV transaction. The issuer refused to refund this customer on the basis that their records show the PIN was used. Luckily, the customer managed to obtain the merchant receipts, and these contained the TVR. This indicated that the PIN was not used, and the merchant opted to fall back to signature.

We decoded the TVR and informed the customer, who was then able to get a refund. However, other customers are less fortunate: it is unusual for the TVR to be included on the receipt, and often the merchant receipt has been destroyed by the time the dispute is being considered. In these cases we have not been able to obtain the TVR, IAD, or even a statement by the bank as to how they established that the cardholder was verified through the correct PIN being entered.

Our demonstration therefore exposes a deeper flaw in EMV and the associated systems: they fail to produce adequate evidence to produce in dispute resolution or litigation.

Procedures are also a mess. For example, once a transaction is disputed a typical bank either destroys the card or asks the customer to do so, preventing information from being extracted which might show whether the card was actually used. Transaction logs are commonly only kept for 120 days, and by the time the dispute is being heard the bank may have destroyed most of the records. (This was the case in the well-known Job v. Halifax trial: even though the Halifax had been notified that the transaction was being disputed, the logs were then destroyed in defiance of Visa guidelines [13]).

These general issues are discussed by Murdoch [14], but with respect to the vulnerability described in this paper, such banks are faced with a difficult situation. If they have indeed destroyed all record of the IAD, they will be unable to show that disputed transactions actually used the correct PIN. So our findings should at least make banks understand that it is in their interest to retain evidence rather than destroy it.

Another evidential issue is that even if the issuer were able to establish whether the attack we present here had occurred, this may not help customers because the typical receipt still states that the PIN was verified. Although this may be false, many people evaluating evidence (adjudicators, judges, and jury members) will not know this. In one particular case, from 2009, the issuing bank, and goverment-approved adjudicator, explicitly relied upon the "Verified by PIN" indicator on the merchant receipt, in concluding that the transaction was PIN-verified and therefore the customer was liable. For this reason we propose that terminals no longer print "Verified by PIN" unless the protocol actually supports this assertion.

VIII. Related Work

EMV has been available for 14 years and is now widely deployed though little prior research exist on the security of the protocol. In 1999, Herreweghen and Wille [15] evaluated the suitability of EMV for Internet payment transactions and identified the problem of not being able to determine if the Verify command was ever executed because it is not authenticated. As a solution for their proposed Internet-based payment scheme, the authors suggest that the ARQC will only be generated if the Verify command has been successful, as a condition for a secure transaction.

More recently, interest in EMV has increased since it became widely deployed in 2005, but perhaps due to the complexity of the specification, the closed user community, and difficulties in carrying out experiments, researchers have not focused on this area. Anderson *et al.* [16] described how bank customers might have difficulty in obtaining refunds once transactions were authorized by PIN. That paper also outlines some potential attacks against Chip and PIN, such as cloning SDA cards for use in offline transactions, and the likelihood that criminals would migrate towards cross-border fraud if and when legacy magnetic strip transactions were disabled for domestic transactions.

Another potential EMV weakness outlined in [16] was the relay attack, which was refined and demonstrated by Drimer and Murdoch [17]. Here, the criminal places a tampered Chip and PIN terminal in a store, which the victim uses to make a small transaction (e.g. buying a meal at a restaurant). Rather than placing the transaction, the terminal relays the communications between the victim's card and a fake card which is being used for a far larger transaction elsewhere (e.g. buying diamonds at a jewelery shop). The authors also describe a defence against this attack, in which the terminal and card engage in a cryptographic exchange which not only establishes authenticity but also a maximum distance bound, either eliminating or greatly limiting the applicability of the attack.

The relay attack requires that the victim's card is in the tampered terminal for the duration of the fraudulent transaction, which puts constraints on timing. Alternatively, an attacker could tamper a terminal to merely record card details, and then use them for a fraudulent transaction later. Drimer *et al.* [18] demonstrated that current Chip and PIN payment terminal have inadequate tamper resistance, and a tapping device can be surreptitiously added to record the customer's PIN and enough details to allow a cloned magnetic strip card to be created. Criminals are now known to have carried out variants of this attack, so banks are now taking action: the chip no longer has a copy of the magnetic strip (one data field is replaced), and magnetic strip fallback transactions are gradually being phased out.

The work presented in this paper is a significant advance in our understanding of attacks against EMV because it is applicable to online transactions (unlike cloned SDA "yes cards"); it does not require criminals to synchronise their fraudulent purchase with that of an unwitting customer (unlike the relay attack); and it does not depend on magnetic strip fallback (unlike the payment terminal tampering attacks). As a consequence, it is one of the most realistic and attractive attacks to be carried out by criminals, if and when magnetic strip transactions are no longer permitted. It could even be used at the moment, by criminals who wish to make purchases within countries which now mandate EMV transactions at point-of-sale. We believe it may explain a number of the transaction dispute cases reported to us.

If this attack becomes more widely used, its net effect will be that criminals can use stolen cards in shops without the cardholder being negligent – exactly as was the case with magnetic strip cards before the introduction of EMV. However, so long as the public is not aware of the attack, the banks will be able to get away with blaming cardholders for fraud. We have therefore decided on a policy of responsible disclosure, of publishing this paper some time after informing bank regulators in the UK, Europe and North America of the vulnerability.

At present, we understand that there is a lot of pressure on the US Federal Reserve from the banks it regulates to countenance a move from magnetic strip cards to EMV. This paper shows that such a move may be premature. It's not reasonable for the smart card industry to foist a broken framework on the US banking industry and then leave it to individual issuer banks to come up with patches. The EMV consortium should first publish its plans for fixing the framework, and the Fed should satisfy itself of three things.

First, will the fix work technically? For this, only open peer review will do. Second, will the high level of consumer protection so far enjoyed by US cardholders be preserved? Third, will the introduction of the remediated system introduce any systemic risks, because of moral hazard effects? For these last two questions to be answered in the affirmative, we believe that there must be no associated 'liability shift' as there has been in Europe and as is proposed in Canada.

IX. Response

We will include any responses from bank regulators, brands and trade associations in the final version of the paper.

X. Conclusion

In this paper we have shown how the PIN verification feature of the EMV protocol is flawed. A lack of authentication on the PIN verification response, coupled with an ambiguity in the encoding of the result of cardholder verification as included in the TVR, allows an attacker with a simple man-in-the-middle to use a card without knowing the correct PIN. This attack can be used to make fraudulent purchases on a stolen card. We have demonstrated that the live banking network is vulnerable by successfully placing a transaction using the wrong PIN. The records indeed falsely show that the PIN was verified successfully, and the money was actually withdrawn from an account.

Attacks such as this could help explain the many cases in which a card has supposedly been used with the PIN, despite the customer being adamant that they have not divulged it. So far, banks have refused to refund such victims, because they assert that a card cannot be used without the correct PIN; this paper shows that their claim is false.

We have discussed how this protocol flaw has remained undetected, due to the public specifications being not only complex, but also failing to specify security-critical details. Finally, we have described one way in which this vulnerability may be fixed by issuer banks, while maintaining backwards compatibility with existing systems. However, it is clear that the EMV framework is seriously flawed. Rather than leaving its member banks to patch each successive vulnerability, the EMV consortium should start planning a redesign and an orderly migration to the next version. In the meantime, the EMV protocol should be considered broken.

References

[1] *EMV – Integrated Circuit Card Specifications for Payment Systems, Book 1: Application Independent ICC to Terminal Interface Requirements*, Version 4.2 ed., EMVCo, LLC, June 2008.

[2] *EMV – Integrated Circuit Card Specifications for Payment Systems, Book 2: Security and Key Management*, Version 4.2 ed., EMVCo, LLC, June 2008.

[3] *EMV – Integrated Circuit Card Specifications for Payment Systems, Book 3: Application Specification*, Version 4.2 ed., EMVCo, LLC, June 2008.

[4] *EMV – Integrated Circuit Card Specifications for Payment Systems, Book 4: Cardholder, Attendant, and Acquirer Interface Requirements*, Version 4.2 ed., EMVCo, LLC, June 2008.

[5] EMVCo, "About EMV," November 2009, http://www.emvco.com/about_emv.aspx.

[6] APACS, "2008 fraud figures announced by APACS," March 2009, http://www.ukpayments.org.uk/media_centre/press_releases/-/page/685/.

[7] Which?, "Fraud victims struggle to get money back," June 2009, http://www.which.co.uk/news/2009/06/fraud-victims-struggle-to-get-money-back-179150.jsp.

[8] *Visa Integrated Circuit Card – Card Specification*, Version 1.4.0 ed., Visa International, October 2001.

[9] Maxim Integrated Products, Inc., *MAX1740, MAX1741 SIM/smart-card level translators in μMAX*, January 2001, http://datasheets.maxim-ic.com/en/ds/MAX1740-MAX1741.pdf. [Online]. Available: http://datasheets.maxim-ic.com/en/ds/MAX1740-MAX1741.pdf

[10] Xilinx Inc., "Spartan-3E starter kit," November 2009, http://www.xilinx.com/products/devkits/HW-SPAR3E-SK-US-G.htm.

[11] L. Gong and P. Syverson, "Fail-stop protocols: An approach to designing secure protocols," in *International Working Conference on Dependable Computing for Critical Applications*, September 1995, pp. 44–55.

[12] T. Dierks and C. Allen, "The TLS protocol," IETF, RFC 2246, January 1999.

[13] *Job v Halifax PLC case number 7BQ00307*, May 2009, http://www.stephenmason.eu/wp-content/uploads/2009/06/job-v-halifax-plc-judgment-5_june-2009.pdf.

[14] S. J. Murdoch, "Reliability of chip & PIN evidence in banking disputes," in *Digital Evidence and Electronic Signature Law Review*, vol. 6. Pario Communications, November 2009, pp. 98–115, ISBN 0-9543245-9-5.

[15] E. V. Herreweghen and U. Wille, "Risks and potentials of using EMV for internet payments," *USENIX Workshop on Smartcard Technology*, May 1999.

[16] R. Anderson, M. Bond, and S. J. Murdoch, “Chip and spin,” March 2005, http://www.chipandspin.co.uk/spin.pdf.

[17] S. Drimer and S. J. Murdoch, “Keep your enemies close: Distance bounding against smartcard relay attacks,” in *USENIX Security Symposium*, August 2007. [Online]. Available: http://www.usenix.org/events/sec07/tech/drimer/drimer.pdf

[18] S. Drimer, S. J. Murdoch, and R. Anderson, “Thinking inside the box: system-level failures of tamper proofing,” in *IEEE Symposium on Security and Privacy (Oakland)*, May 2008, pp. 281–295. [Online]. Available: http://www.cl.cam.ac.uk/~sd410/papers/ped_attacks.pdf

Appendix

A. Transaction Log of MITM Attack

The following log was collected during one of our man-in-the-middle experiments, where we used one of our own cards to purchases goods in a online Chip and PIN transaction, while using the incorrect PIN. Data items which could be used to identify the merchant who assisted us with the experiments has been redacted (`xx`), and unnecessary detail has been removed for brevity (. . .). Principals are Terminal (T), Card (C), and man-in-the-middle (M).

```
T → C   00 a4 04 00 0e 31 50 41 59 2e 53 59 53 2e 44 44 -   Select file “1PAY.SYS.DDF01”
        46 30 31
C → T   6f 1a 84 0e 31 50 41 59 2e 53 59 53 2e 44 44 46 -   Opened “1PAY.SYS.DDF01” (language EN)
        30 31 a5 08 88 01 02 5f 2d 02 65 6e 90 00
T → C   00 b2 01 14 00                                      Read Record
C → T   70 40 61 1e 4f 07 a0 00 00 00 29 10 10 50 10 4c -   Available applications: “LINK” and “VISA
        49 4e 4b 20 20 20 20 20 20 20 20 20 20 20 20 87 -   DEBIT”
        01 01 61 1e 4f 07 a0 00 00 00 03 10 10 50 10 56 -
        49 53 41 20 44 45 42 49 54 20 20 20 20 20 20 87 -
        01 02 90 00
T → C   00 a4 04 00 07 a0 00 00 00 03 10 10                 Select file “VISA DEBIT”
C → T   6f 25 84 07 a0 00 00 00 03 10 10 a5 1a 50 10 56 -   Opened “VISA DEBIT” (language EN)
        49 53 41 20 44 45 42 49 54 20 20 20 20 20 20 87 -
        01 02 5f 2d 02 65 6e 90 00
T → C   80 a8 00 00 02 83 00                                Get Processing Options
C → T   80 0a 5c 00 08 01 01 00 10 01 04 01 90 00           Transaction started, 5 records available

T → C   00 b2 01 0c 00                                      Read Record
C → T   70 3e 57...5f 20...9f 1f...90 00                    Record (Track 2 Equivalent Data, Cardholder
                                                            Name, Track 1 Discretionary Data)

T → C   00 b2 01 14 00                                      Read Record
C → T   70 49 5f 25...5f 24...9f 07...5a...5f -             Signed record (Application Effective Date,
        34...9f 0d...9f 0e...9f 0f...8e 10 00 00 -          Application Expiration Date, Application Usage
        00 00 00 00 00 00 41 03 1e 03 02 03 1f 03 90 00     Control, Application Primary Account Number,
                                                            Application Primary Account Number Sequence
                                                            Number, Issuer Action Code – Default, Issuer
                                                            Action Code – Denial, Issuer Action Code –
                                                            Online, Cardholder Verification Method List)

T → C   00 b2 02 14 00                                      Read Record
C → T   70 81 93 93...90 00                                 Record (Signed Static Application Data)

T → C   00 b2 03 14 00                                      Read Record
C → T   70 81 c0 8f...9f 32...92...90 00                    Record (Certification Authority Public Key
                                                            Index, Issuer Public Key Certificate, Issuer
                                                            Public Key Exponent, Issuer Public Key
                                                            Remainder)

T → C   00 b2 04 14 00                                      Read Record
C → T   70 48 8c 15 9f 02 06 9f 03 06 9f 1a 02 95 05 5f -   Record (Card Risk Management Data Object
        2a 02 9a 03 9c 01 9f 37 04 8d 17 8a 02 9f 02 06 -   List 1 (CDOL1), Card Risk Management Data
        9f 03 06 9f 1a 02 95 05 5f 2a 02 9a 03 9c 01 9f -   Object List 2 (CDOL2), Application Version
        37 04 9f 08...5f 30...5f 28...9f 42...9f -          Number, Service Code, Issuer Country Code,
        44...90 00                                          Application Currency Code, Application
                                                            Currency Exponent)
```

T → C	80 ca 9f 17 00	Get Data (PIN try counter)
C → T	9f 17 01 03 90 00	Remaining PIN tries = 3
T → M	**00 20 00 80 08 24 00 00 ff ff ff ff ff**	Verify PIN “0000”
M → T	**90 00**	PIN correct
T → C	80 ae 80 00 1d xx xx xx xx xx xx 00 00 00 00 00 – 00 08 26 00 80 00 80 00 08 26 xx 11 09 00 xx xx – xx xx	Generate AC (ARQC)
C → T	80 12 80 xx xx xx xx xx xx xx xx xx xx 06 01 0a – 03 a0 00 10 90 00	ARQC
T → C	00 82 00 00 0a xx xx xx xx xx xx xx xx 30 30	External Authenticate
C → T	90 00	External authenticate successful
T → C	80 ae 40 00 1f 30 30 xx xx xx xx xx xx 00 00 00 – 00 00 00 08 26 00 80 00 80 00 08 26 xx 11 09 00 – xx xx xx xx	Generate AC (TC)
C → T	80 12 40 xx xx xx xx xx xx xx xx xx xx 06 01 0a – 03 60 00 10 90 00	TC