Security Protocols and Evidence: Where Many Payment Systems Fail

Steven J. Murdoch, Ross Anderson

Abstract. As security protocols are used to authenticate more transactions, they end up being relied on in legal proceedings. Designers often fail to anticipate this. Here we show how the EMV protocol – the dominant card payment system worldwide – does not produce adequate evidence for resolving disputes. We propose five principles for designing systems to produce robust evidence. We apply these to other systems such as Bitcoin, electronic banking and phone payment apps. We finally propose specific modifications to EMV that could allow disputes to be resolved more efficiently and fairly.

1 Introduction

Even if a security protocol design is sound, the implementation may be flawed; principals may be dishonest; or other principals may raise doubt about the integrity of humans or system components. Such issues frequently occur with financial transaction protocols, where real money is at stake.

In this paper, we use payment cards as a case study for developing principles for designing systems to produce robust evidence of their correct operation. These principles apply widely, but banking is a good place to start. Section 2 will summarize the EMV protocol and highlight its flaws from some case studies of disputes; Section 3 will introduce a set of principles for designing systems to produce reliable evidence; Section 4 will discuss some other systems and Section 5 will show how these principles can be applied to payment systems.

2 The EMV protocol and its flaws

The EMV protocol [11] promoted by EuroPay, MasterCard and Visa is now the world's dominant smart card payment system, with 1.55 billion cards (both credit and debit) in issue as of Q2 2012 [10]. The USA is a late adopter but has a target of 2015 for completing deployment [15].

EMV provides a standard toolkit to build security protocols which interoperate despite the details differing by brand and by country. In the UK and Canada, the system is known as "Chip and PIN" because most point-of-sale transactions are authenticated with a PIN; Singapore continued to use signatures to authenticate customers; and the USA will be somewhat similar to Singapore.

Financial Cryptography and Data Security 2014, 3–7 March 2014, Barbados (pre-proceedings draft)

An EMV transaction consists of three stages. The first is *card authentication* where a chip in the card proves to the terminal that it is authentic. Next, *cardholder verification* involves the customer either entering a PIN or signing for the transaction. Finally, in *transaction authorization*, the card produces one or more message authentication codes; as these use a key shared between the card and the issuing bank, they can only be verified if the terminal is online.

The EMV protocol has numerous vulnerabilities, some of which are the inevitable result of implementation choices. For example, banks can issue expensive cards that use public-key cryptography in the card authentication step, or cheap ones that merely present a certificate, signed by the issuing bank, on the card data; cards using this *static data authentication* option are easy to clone [4]. Other vulnerabilities were errors of design or implementation.

Insider attacks and blunders: Visa admitted that criminals have used brute-force attacks against the PIN-verification command of their hardware security modules (HSMs) to discover customer PINs, bypassing PIN-retry limits [21]. API attacks on HSMs have been known for a decade [8], and can also be used to steal the keys needed to forge cards. Call-centre operators can send PIN-readvice letters to an address controlled by an accomplice [17]; and many other bank employees have been prosecuted for abusing their access to commit fraud in various ways [1]. Blunders also happen; in one case, two identical cards were sent to a customer – a 'this should never happen' failure in the process of personalization.

PIN verification flaws: Where the customer PIN is verified by the card offline – the default in most countries for merchant terminals – a fraudster can often use a stolen card without knowing the PIN by inserting electronics between the card and the terminal that tells the terminal the PIN verified correctly, but tells the card that the transaction was authorised by signature [19]. Despite fraud losses since 2010 [20] and publicity since 2011, only a few banks cross-check the card and merchant records carefully enough to detect this 'No-PIN' attack.

Pre-play attack: In an EMV transaction, the terminal sends the card the transaction amount, the date, and a random challenge; these are authenticated by the card. However many terminals do not generate proper random numbers; some use a counter instead. So an attacker with a payment terminal can get an authentication code that will be accepted by a different terminal at some future date [7]. The communications from a terminal to a bank can also be manipulated to achieve the same effect: the attacker can insert a prerecorded nonce and authentication code to make a transaction work. So a correct authentication code does not automatically imply that the card was used in that terminal.

Misreporting by terminal: We have seen cases where the issuer's logs stated that a transaction was PIN-authenticated but the receipt showed it was signature-authenticated [12]. Merchants have an incentive to lie to their bank, as PIN transactions attract lower fees and are less likely to be charged back. In this case the issuer relied upon the (unauthenticated) merchant-reported value rather then the (authenticated) card-reported value, and denied the customer a refund.

Transaction reversal: The EMV transaction that authenticates a payment is separate from the later settlement transaction where the merchant actually gets paid. A UK gang noticed that while cardholders were authenticated to the bank, merchants were not. They would buy expensive goods from a merchant, then impersonate that merchant to the bank to do a transaction reversal, and spend the same money all over again. At trial, bank experts' and defence experts' estimates of the losses gang's takings differed by many millions of pounds, and the jury failed to agree.

Where it is clear which type of fraud has occurred, the card scheme rules will specify who must pay the costs. The hard cases are where it is not clear whether the correct PIN and card were used, and merchants or customers disagree with the banks' view of what happened. Many of the above cases led to fierce disputes – which is why they came to attention.

3 Designing for evidence

The above cases show that the evidence produced by EMV transactions is just not adequate for discriminating between attacks, and can lead to unfair treatment of both cardholders and merchants. Banks for their part fear that due to the lack of confidence that can be placed on the evidence, they may be forced to refund customers who are actually making fraudulent claims of fraud. It is in the interests of all honest parties to design a protocol that produces robust evidence. In this section, we will explore what principles might help.

First, evidence must be usable. In the case of Job v Halifax [13], the bank was unwilling to disclose the card's authentication keys because they were derived from a batch key, and other cards using keys derived from this were still in issue. In addition, key management procedures were considered commercially sensitive. So an outside expert witness could not have verified the authentication codes in the logs. This brings us to our first principle:

Principle 1: Retention and disclosure. Protocols designed for evidence should allow all protocol data and the keys needed to authenticate them to be publicly disclosed, together with full documentation and a chain of custody.

It follows that nothing in the calculations needed to check a protocol run should depend on any security sensitive, commercially confidential, or personal information. The processes used to generate, issue, use, store and recover both keys and data must be open to inspection by hostile litigants.

Second, evidence mechanisms must be tested end-to-end. Many cryptography papers have statements like 'so the judge raises Alice's signature s to the power e, finds it's equal to $h(m)$, and sends Bob to jail.' This is sadly unrealistic. Each party in legal proceedings presents their own evidence, and they can challenge the evidence presented by the other party. For example, the digital tachographs now used to monitor drivers' hours in Europe are designed to produce authenticated logs with digital signatures, but these are not yet used [2]. A vehicle inspector who stops a truck suspected of a violation simply uses the traditional procedure of printing out two separate copies of the log from the vehicle unit and sealing them in evidence bags. The cryptography although present is disregarded. This should have been expected: system functionality that isn't tested thoroughly before deployment isn't likely to work well, especially if the main stakeholders and their vendors don't think it matters. Our second principle is therefore

Principle 2: Test and debug evidential functionality. When a protocol is designed for use in evidence, the designers should also specify, test and debug the procedures to be followed by police officers, defence lawyers and expert witnesses.

With digital tachograph records, police officers had to improvise, and continued using ancient techniques, as did the organisations that received EMV fraud reports. This led to front-line dispute resolution being left to bank call centres and second-line resolution to bodies such as the Financial Ombudsman Service that do not have the technical expertise to challenge bank logs. The easiest way to deal with disputes was to fob off customers who were not particularly profitable, or perhaps who were not rich enough to fight the bank in court. In the tachograph case, the failure might be described perhaps as a missed business opportunity; in the bank case as a failure of regulation.

Third comes complexity. Systems incorporating a security protocol are usually much more complex than the protocol itself. For example, card payment systems incorporate EMV but also include backwards compatibility with legacy systems, data collection for marketing, interfaces with call centers, and settlement services. Bugs in, or insider attacks through, these other systems can lead to inaccurate logs – as in the fraudulent reversal case above. Systems that are complex and poorly documented are also more liable to have exploitable bugs – complexity was at fault for the No-PIN attack. Our third principle is therefore

Principle 3: Open description of TCB. Systems designed to produce evidence must have an open specification, including a concept of operations, a threat model, a security policy, a reference implementation and protection profiles for the evaluation of other implementations.

Another example comes from curfew tags, which are used in many countries to track offenders released early from prison, or given a community sentence instead of prison. The tag is typically a tamper-evident ankle bracelet that alarms if the offender tried to pull it off, or goes out of range of a base station at his home between 7pm and 7am. However one UK operator kept logs only at a back-end system that was notoriously buggy, and was thus unable to distinguish between tamper events and false alarms due to software bugs. As a result, tampering prosecutions that were subject to technical challenge had to be dropped [3]. The curfew enforcement contract has now gone to a different firm. A much better design would have been to get the base station to create and sign log entries for storage on the back-end server. The base station contains tamper-resistant cryptography in any case, and using this to sign the log would have removed the server software from the trusted computing base (TCB). A useful precedent may be the Google NFC wallet, where logs are generated in the secure element in the NFC chip and stored on Google's servers, thus removing both the Android handset and the merchant terminal from the TCB.

So if designing a system that is too complex or sensitive for a full open specification to be feasible, such as a smartphone incorporating a mobile wallet payment system, the prudent engineer will design the payment part of the system so that it has open mechanisms and independent logging, with a clear specification of the APIs or other interfaces by means of which an attacker might have fed malicious instructions to it. That way, expert witnesses can investigate how the overall system might have been tampered with.

Our fourth point is related, and concerns the effects of failure. In practice, the evidence for a disputed EMV transaction is simply a record that an EMV transaction happened. At best, there may be enough information in the logs to repeat the security checks; but if a fraud was carried out successfully, the attacker must have seen to it that the checks passed. This applies even to cards implementing the most secure EMV variant, Combined DDA/Application Cryptogram Generation (CDA), where the card signs a hash of the transaction. The transaction should only work if the CDA signature verifies – but, perversely, neither the signature nor the data needed to verify it are sent back to the bank.

This is quite the wrong way round. Compare what happens with an old-fashioned manuscript signature: frauds are easier to commit than with a PIN, but are also easier to investigate because criminals are likely to produce a signature which forensic inspection will reveal as a forgery. Similarly, banknotes are designed to support three levels of checking – by the public, by bank tellers and by central-bank examiners. The public know a few of the security features, the tellers a few more, while only the banknote issuer knows all of them.

It would therefore be beneficial if the system used for dispute resolution could make extra checks. Fraudsters would have to bypass the normal checks, but would have less incentive or opportunity to circumvent the secondary ones. Our fourth principle is therefore

Principle 4: Failure-evidentness. Transaction systems designed to produce evidence must be failure-evident. Thus they must not be designed so that any defeat of the system entails the defeat of the evidence mechanism.

This is a more subtle property than the classic case of a fail-stop system. Failure-evidentness might in some cases require independent mechanisms so it can detect a total system compromise, and these mechanisms might have to be based on random sampling. For example, the UK has had successive waves of ATM frauds that the banks initially believed were impossible, and tried to blame on customers, until a large enough number of complaints from respectable cardholders or merchants whose business was too valuable to alienate forced managers to take a second look. The same happened with transaction reversal frauds. In some overseas jurisdictions, ATM cameras are mandatory for other reasons (in New York to deter mugging) and these ensure that fraud patterns resulting from a new modus operandi cannot so easily be ignored. Regulators might consider requiring 5% of the ATM fleet to be equipped with cameras. This would reduce the incentive on middle managers to deny a problem for as long as possible and hope it will go away.

Finally, there is a governance issue. Even if digital evidence starts off being retained, open, tested and forensically efficacious, it is not trivial to ensure that it will remain so as the system evolves, or that failures will be fixed. Initial forensic procedures can be specified by the system designer, but if he retains control he may resist admitting that anything was overlooked. He may have long-term supply contracts with banks worth many millions and be very anxious to not increase his manufacturing costs. Banks similarly may be anxious not to shake confidence in the system, for fear of encouraging fraudulent claims of fraud. Our fifth suggested principle is therefore aimed at regulators.

Principle 5: Governance of forensic procedures. The forensic procedures for investigating disputed payments must be repeatable and be reviewed regularly by independent experts appointed by the regulator. They must have access to all security breach notifications and vulnerability disclosures.

This is a political hot potato in Europe at the moment. Security engineers and NGOs have pushed for breach-disclosure laws, while the European Commission has proposed a Network and Information Security directive that will compel all Member States to legislate for both breaches and vulnerabilities to be reported to a single government agency in each country. It is unclear that the designated agency is likely to have financial consumer protection as its first priority. Nonetheless, regulators must do what they can.

4 Other systems

The above principles can be illustrated by considering three different payment systems: phone banking apps, the overlay banking service Sofortüberweisung, and the cryptographic payment scheme Bitcoin.

4.1 Phone banking apps

Bank customers are increasingly making payments using phone banking apps. The security of these apps varies across platforms and suppliers, but the diversity of Android platforms has so far prevented significant use of protection mechanisms such as TrustZone [5], while mobile network operators have opposed the widespread use of secure elements in phone handsets themselves, instead promoting the SIMs they themselves control. As a result, apps provided by the handset vendors (such as the Google mobile wallet) are more or less limited to low-value payments, while high-value account payments are made using proprietary apps that run in user mode. In consequence, the vendors of banking trojan software like Zeus are starting to make versions available that target phone banking.

The typical phone banking app complies with none of our principles. First, the protocols and the embedded crypto are proprietary and may be covered by an NDA between the software vendor and the bank; the disclosure of technical details in one trial might expose vulnerabilities that could be exploited against other banks who bought banking apps from the same vendor. Next, we have seen no case of an open design or reference implementation, let alone support for dispute resolution or transparency to the regulator. The obscurity extends from the software design to the nature of the logs kept by the bank, or by the system house that operates its servers. And finally there is no reason to believe that such a system will be failure evident. A malware attack on the bank's customers that steals authentication keys, or simply modifies the app's user interface to make payments to the gang using the mechanisms described in Aurasium [22], could be catastrophic, and detected only when a mob of angry customers complain.

4.2 Sofortüberweisung

A payment service in Germany, Sofortüberweisung means 'instant payment'. This offers an service whereby a customer can make a payment to an online merchant using a Giro transfer from his bank account. A participating website might offer a shopper an option of a card payment with a fee or a Sofort payment with no charge. If she clicks on Sofort, it solicits her bank name and account number, then tries to log on to her bank account and asks for her password and authentication code when the bank demands it. It checks that funds are available and sends them to the merchant. In effect it does a man-in-the middle attack on the German banking system, and now has 3% of the online payment market.

For the merchant, it's cheaper than a card payment (the fee is .75% plus 10 cents versus 2.5% for a card); for the customer, it's more convenient than doing a Giro payment, as the interface is better, and the payment is tied to the merchant transaction automatically; but for the banks it's a nightmare. A third party is not only costing them money by arbitraging their services, but accumulating customer credentials and thus undermining their security. The German banks sued Sofort for inducing their customers to break their terms and conditions by disclosing passwords, but the case failed when the Federal competition authorities intervened and told the court that competition with the payment card cartel was welcome. Sofort now has a banking license.

The implications for our robustness principles are as follows. Principle 1, openness, is reinforced for all; bank attempts to make authentication processes obscure to thwart Sofort have failed. Principles 2 and 3 are disregarded by all players equally except insofar as openness is increased. Principle 4, of failure-evidentness, is seriously undermined. If a customer disputes a transaction with a bank, and has previously used Sofort for any transaction at all, then it's not obvious who is at fault, and in theory the bank could rely on its terms and conditions to void the customer guarantee. Principles 5 and 6 are essentially unaffected, although Sofort's very existence may in time drive regulators to acquire more technical nous.

4.3 Bitcoin

Bitcoin is a digital currency, or perhaps more correctly a digital resource designed to be scarce and electronically tradeable, in which coins are mined by principals who solve cryptographic puzzles ('miners') and can be transferred to other principals using digital signatures. Bitcoin miners find special hashes of all transactions seen to date, thereby guaranteeing consensus on the transaction history or 'blockchain' (unless a majority of miners were to start working on a different transaction history). Bitcoins are converted to and from real money by brokers, of which one firm (Mt. Gox) has most of the business. Principals are known only by one or more public signature verification keys, so anonymous transactions are possible (though coins can be traced through transactions, allowing traffic analysis of the Bitcoin economy [16]). Bitcoins have been used for both lawful and unlawful purposes, the latter including the 'Silk Road' auction market for illegal drugs and firearms, which was recently shut down by the FBI.

Had the authorities not managed to identify the individuals behind Silk Road, legal coercion might conceivably have been used to shut Bitcoin down or bring it under regulatory control. There are several options. First, as pointed out by Böhme [18], law enforcement could have compelled the major brokers such as Mt. Gox to blacklist bitcoins that had been used on illegal markets such as Silk Road, thereby undermining Bitcoin's fungibility and causing loss of trust. A second possibility would be to coerce the Bitcoin developer community; this has been done in the Lavabit case, where a webmail provider shut his service rather than yield to an FBI demand that he hand over the service's SSL keys. A third possibility would be to coerce the miners: at present two mining companies produce over 50% of bitcoins, so could in theory tamper with the blockchain by, for example, not recognising a transaction made by a criminal suspect. A fourth would be for a government agency to acquire the computing power to produce over 50% of the mining activity and thus take over the blockchain directly.

From individual bitcoin holders' point of view, the main problem is that there is no issuing authority and thus no-one to turn to in the event that their bitcoins get stolen (or that they simply forget the password to their Bitcoin wallet, rendering their bitcoins unspendable). Thus Bitcoin fails to meet the consumer-protection provisions of the EU Payment Services Directive.

Bitcoin easily satisfies principle 1 (open data and checkability of authentication) and arguably 3 (open spec and implementation). It fails principles 2 (forensic and dispute procedures) and 5 (governance) because there is no dispute resolution mechanism. Principle 4 is also violated because a defeat of Bitcoin (for example, by legal coercion of the software) would be a catastrophic failure.

An interesting protocol design problem is if a court is contemplating ordering a break of Bitcoin – e.g. by coercing software developers, brokers, or miners – then is it feasible to move to a Bitcoin 2.0 that allowed selective transaction blacklisting in a robust way? Blacklisting all transactions with coins that were once used in Silk Road, for example, would lead to gross overblocking. Or is the only feasible outcome the total destruction of the Bitcoin ecosystem?

5 Improvements to EMV

It can be very hard to implement changes to any widely deployed protocol if that involves changing a lot of systems simultaneously. For example, the many bugs discovered in SSL/TLS over the past decade have mostly been fixed with server-side hacks, as it is simply too hard to change all the world's web servers and browsers at once. The same applies in spades to EMV, with 30,000 banks, millions of merchants, and over a billion cards in issue. We can therefore only consider changes that can be introduced piecemeal with changes to either cards or back-end systems.

Following principle 3, we propose performing the additional checks primarily on the card, because cards are far simpler than the back-end, are tamper resistant, and are in some sense under control of the customer. Therefore more information about their functionality can be disclosed and there are fewer opportunities for malicious modification.

5.1 Transaction counters

EMV cards maintain one or more counters that are incremented at the start of every transaction. This can already be quite useful for detecting cloned cards, because if a genuine card and its clone are used concurrently there will be sequence overlaps in attempted transactions.

The use of the transaction counter as an investigation tool does not require any changes to the card, but does require the development of procedures to extract it from the banks' logs and also from the legitimate card. Above all we need a regulatory change. For example, banks instruct their customers to cut up the card at once if there is a dispute, which is contrary to the customer's interest.

5.2 Transaction log

Optionally EMV cards can maintain a log of recent transactions. If the card is still in the customer's possession then the presence or absence of the disputed transaction in the card log is convincing evidence as to whether the legitimate card was used. However the transaction log is not commonly enabled, and there is a privacy impact of enabling the log as any merchant could then read it.

As with the transaction counter, no changes are needed to cards (other than enabling the feature) but there would need to be procedures developed for extracting and evaluating the results. Perhaps, with a bit more effort, a bank could arrange things so that its customers could read their card logs at its ATMs but still protect their privacy from merchants.

5.3 Forensics mode

An issue with the transaction counter and the transaction log is that gaining access to them requires initiating a transaction and therefore increasing the counter. For repeatability, it would be better if a card could be placed into a forensics mode where it is no longer able to carry out transactions but will disclose the transaction counter. The card could also unlock the transaction log so that it could be read, and allow access to internal risk analysis counters which could be correlated with bank logs.

5.4 Cryptographic audit log

A weakness of all of the above approaches is that they still depend on the bank's logs for reliability and so do not meet the criterion of complete system disclosure. Past experience sadly suggests that banks in some countries will drag their feet over retaining logs and making them available; and that the regulators in these countries will be reluctant to force them. (The two properties are of course related.) So how can a bank in a well-regulated country protect its cardholders when they travel and transact in a poorly-regulated one? A forward secure audit log implemented by the card can provide a lot of protection while storing log records on the card issuer's server to avoid limitations on bank card memory.

The card would be initialized not just with a key used for authentication codes, but with an audit key that is also unique to each card (even if this card replaces a card which seemed to fail personalisation). The audit key is updated on each new transaction and a forward-secure MAC [6] is computed on the transaction (including the result of PIN verification). Even compromising the card's current audit key will not then be enough to produce fake log messages from the past. This construction also means that audit keys can always be produced in court to resolve disputes.

We want to prevent a forger working forwards as well as backwards, so that even if a card's original audit key is later compromised, the attacker still cannot go back and invent an entirely fake transaction history. So the bank should create a hash-chain over all online transactions, with the root being the audit key [14], and commit the audit records by including them in the customer's statement. Once put into forensics mode, the card would provide access to the final entry of the hash-chain. Then even with access to the original audit key, a criminal would not be able to insert a fake transaction without creating an inconsistency between the bank's log and the legitimate card's log.

6 Open questions

The adoption of the above proposals would substantially improve the quality of evidence which could be presented in EMV disputes. However, it would not resolve all cases. When there is no dispute that the correct card and PIN was used, liability depends on whether the PIN was discovered through customer negligence. Fraudulent requests for PIN-readvice letters or brute force attacks against bank HSMs cannot be stopped by changes to card software, but will require changes to back-end systems and operational procedures.

The relay attack [9] also poses a problem because a cryptographic audit log would only prove whether a card processed the transaction which was authenticated, not that the customer saw the transaction. Here too, operational changes can help: in Singapore, transactions are reported to the account holder by SMS, so any relay attacks should be rapidly detected. An alternative technology is a smartphone payment mechanism which can give a more trustworthy display.

ATM transactions are typically performed using online PIN verification and so the card is not able to know whether the PIN was verified correctly. This could be resolved by the ATM sending the PIN to the card for offline PIN verification in addition to the usual online PIN verification. This approach will produce a more valuable audit log as well as defeating attacks which rely on desynchronizing the version of the PIN on the card and the version on the issuer's back-end system.

7 Conclusions

We proposed five principles to guide designers of payment mechanisms and other systems that may have to be relied on to provide evidence.

We analysed a number of systems. Mobile phone banking apps are particularly bad as they typically abide by none of these; this may portend trouble for the industry, as the tagging systems used to monitor curfewees' parole also ignore the above principles, and have failed to stand up in court, with significant commercial consequences. Overlay payment systems such as Sofortüberweisung are less bad but still fall short; such systems may need carefully-designed logging systems to deal with frauds and disputes in the future. Bitcoin does not support any form of dispute resolution at all, and given that it is vulnerable to at least three forms of attack based on legal coercion and one based on brute-force, it may well be more fragile than most of its users realise. Our principles can also be used to expose and highlight design deficiencies in other monitoring systems, such as curfew tags and tachographs.

Our most detailed study was of EMV, 'Chip and PIN', the dominant card payment mechanism, which is used in Europe and Asia, and is now being deployed in the USA. This turns out to have a number of significant shortcomings. We argue that they can be mitigated by individual card-issuing banks, independent of any changes to the EMV protocol suite itself, by making transaction counters more accessible to forensic examination; by having logs of recent transactions on the card; and having key material on the card with which logs are authenticated, and which can be released to forensic examiners without compromising the security of the payment mechanism itself. These technical measures have to be complemented by changes in procedure – most notably telling customers to retain cards in transaction disputes rather than destroy them; and almost certainly by regulatory action too, which will ultimately be succesful only if card-issuing banks are less able than at present to externalise their fraud liability to their customers.

References

1. Aldrick, P.: Former Lloyds head of fraud and security Jessica Harper charged over £2.5m fraud. The Telegraph (May 2012), `http://www.telegraph.co.uk/finance/financial-crime/9289673/Former-Lloyds-head-of-fraud-and-security-Jessica-Harper-charged-over-2.5m-fraud.html`
2. Anderson, R.: On the security of digital tachographs. In: ESORICS. LNCS, vol. 1485, pp. 111–125. Louvain-la-Neuve, Belgium (September 1998)

3. Anderson, R.: Offender tagging. Light Blue Touchpaper (September 2013), http://www.lightbluetouchpaper.org/2013/09/02/offender-tagging/
4. Anderson, R., Bond, M., Murdoch, S.J.: Chip and spin. Computer Security Journal 22(2) (2006), http://www.chipandspin.co.uk/spin.pdf
5. ARM: Building a secure system using TrustZone technology (April 2009), http://infocenter.arm.com/help/topic/com.arm.doc.prd29-genc-009492c/PRD29-GENC-009492C_trustzone_security_whitepaper.pdf
6. Bellare, M., Yee, B.: Forward-security in private-key cryptography. In: Topics in Cryptology – CT-RSA 2003, LNCS, vol. 2612, pp. 1–18. Springer (2003)
7. Bond, M., Choudary, O., Murdoch, S.J., Skorobogatov, S., Anderson, R.: Chip and skim: cloning EMV cards with the pre-play attack. In: Workshop on Cryptographic Hardware and Embedded Systems (CHES). Leuven, Belgium (September 2012), (invited talk) arXiv:1209.2531. http://arxiv.org/abs/1209.2531
8. Clayton, R., Bond, M.: Experience using a low-cost FPGA design to crack DES keys. In: Workshop on Cryptographic Hardware and Embedded Systems (CHES). LNCS, vol. 2523, pp. 579–592. Springer-Verlag, London, UK (2002), http://www.cl.cam.ac.uk/~rnc1/descrack/DEScracker.pdf
9. Drimer, S., Murdoch, S.J.: Keep your enemies close: Distance bounding against smartcard relay attacks. In: USENIX Security Symposium (August 2007)
10. EMVCo: About EMV, http://www.emvco.com/about_emv.aspx
11. EMVCo: EMV Specifications, http://www.emvco.com/specifications.aspx
12. Evans, T.: Barclays blamed me when £1,150 was stolen from my account – but its excuse was actually the bank's own blunder. Daily Mail (June 2012), http://www.dailymail.co.uk/money/saving/article-2162199/Barclays-blamed-1-150-stolen-account.html
13. Kelman, A.: Job v Halifax PLC (not reported) case number 7BQ00307. In: Mason, S. (ed.) Digital Evidence and Electronic Signature Law Review. vol. 6 (2009)
14. Ma, D., Tsudik, G.: A new approach to secure logging. ACM Trans. Storage 5(1), 2:1–2:21 (Mar 2009)
15. MasterCard Worldwide: Progress against roadmap, http://www.mastercard.us/_assets/docs/MasterCard_EMV_Timeline.pdf
16. Meiklejohn, S., Pomarole, M., Jordan, G., Levchenko, K., McCoy, D., Voelker, G.M., Savage, S.: A fistful of bitcoins: characterizing payments among men with no names. In: Internet Measurement Conference. pp. 127–140. ACM (2013)
17. Mitchell, A.: Indian call center fraud case highlights need for change. E-Commerce Times (April 2005), http://www.ecommercetimes.com/story/42112.html
18. Möser, M., Böhme, R., Breuker, D.: An inquiry into money laundering tools in the Bitcoin ecosystem. In: Proceedings of the APWG eCrime Researchers Summit (ECRIME 2013). San Francisco, USA (2013)
19. Murdoch, S.J., Drimer, S., Anderson, R., Bond, M.: Chip and PIN is broken. In: IEEE Symposium on Security and Privacy. pp. 433–446 (May 2010)
20. Sellami, S.: L'imparable escroquerie à la carte bancaire. Le Parisien (January 2012), http://www.leparisien.fr/faits-divers/l-imparable-escroquerie-a-la-carte-bancaire-24-01-2012-1826971.php
21. Visa: Presentation at ATM Security (October 2008), London, UK
22. Xu, R., Saïdi, H., Anderson, R.: Aurasium: Practical policy enforcement for Android applications. In: USENIX Security Symposium. Bellevue, WA, USA (August 2012)