Be Prepared: The EMV Preplay Attack

Mike Bond | University of Cambridge
Marios O. Choudary | University Politehnica Bucharest
Steven J. Murdoch | University College London
Sergei Skorobogatov and Ross Anderson | University of Cambridge

More than 2.37 billion EMV smart card–based payment cards are in use worldwide, responsible for 30 percent of all card-present payments. Yet, a series of significant vulnerabilities makes EMV vulnerable to the preplay attack. Specifically, weak and defective random number generators and various protocol failures leave the EMV system open to fraud at scale.

The leading system for smart card–based payments worldwide is EMV (named after its original inventors—Europay, MasterCard, and Visa), also known as "chip and PIN." More than 96 percent of card-present transactions in Europe use EMV technology, and it is starting to be introduced in the US too. In fact, more than 2.37 billion cards are in use worldwide.

The EMV integrated circuit card technology supersedes the familiar magnetic strip on cards. A cryptographic protocol is executed between a chip card and bank servers based on a message authentication code (MAC) over transaction data, including an unpredictable number. Because they contain a smart-card chip, EMV cards are more difficult to clone than traditional magnetic-strip cards. Yet, in the decade since EMV's introduction, a series of significant vulnerabilities has emerged.

We discovered two EMV protocol flaws: there's no terminal ID to identify involved parties, and the nonce isn't generated by the relying party. Together, these make EMV vulnerable to the *preplay attack*, during which malicious users replay prerecorded transaction data from a target card at a later time. This powerful attack is possible because of a protocol flaw, possibly exploited in conjuction with weak random number generators (RNGs). These flaws can also be exploited using a man-in-the-middle device between the terminal and the acquirer, or malware installed at an ATM or point of sale (POS) terminal.

Our investigation started when we discovered that EMV implementers often use counters, timestamps, or homegrown algorithms to supply the nonce. Here, we describe the survey methodology we developed to chart the scope of this weakness, evidence from ATM and terminal experiments in the field, and our proof-of-concept attack implementation. Finally, we explore why these flaws evaded detection until now.

The Smoking Gun

The case that kicked off the research we report here began when a Maltese customer of HSBC, Alex Gambin, complained about ATM transactions that were wrongly billed to his card in Palma, Majorca, on 29

1540-7993/15/$31.00 © 2015 IEEE

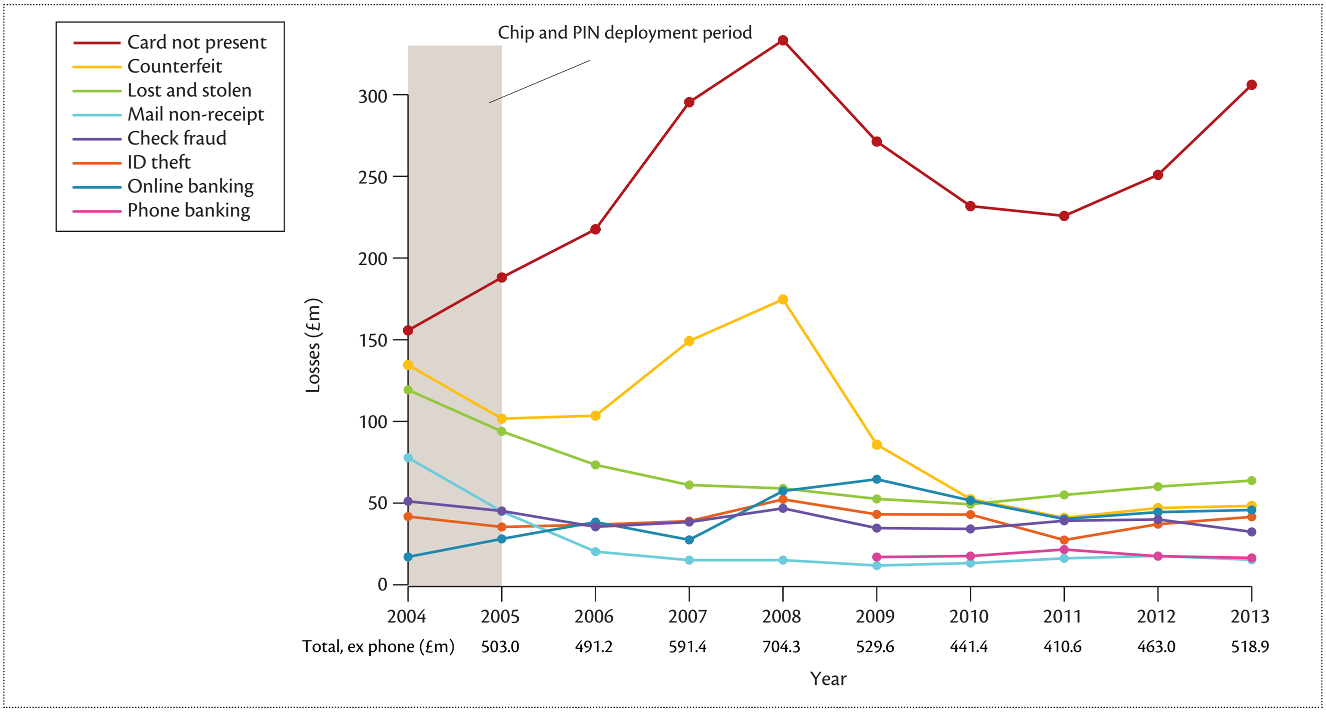

Figure 1. Fraud on UK payment cards. Card-not-present fraud increased since the introduction of EMV technology. (Financial Fraud Auction UK, 2014.)

June 2011, after he bought a meal at a restaurant there. He was refused a refund and asked us for advice. We observed that one of the fields in the log file, the unpredictable number (UN), looked rather predictable, as Table 1 shows. The UN appears to consist of a 17-bit fixed value followed by a 15-bit counter that cycles every three minutes.

If the UN generated by an ATM is in fact predictable, then attackers with temporary access to a card (say, in a Mafia-owned shop) could precompute the authentication codes needed to draw cash from that ATM at some time in the future. We discovered that many ATMs generate poor random numbers, making a preplay attack possible. What's worse, a flaw in the protocol also lets attackers substitute a precomputed transaction even when the RNG is sound.

We informed the industry about our findings in early 2012 so that ATM software could be patched. We are now publishing the details of our study to provide customers with evidence to pursue wrongly denied claims and to enable the crypto, security, and bank regulation communities to learn valuable lessons.

Fraud Using EMV Technology

With EMV technology, each bank card contains a smart-card chip that authenticates transaction data using a MAC calculated with a symmetric key shared between the card and the card-issuing bank. The chip protects against card counterfeiting. However, EMV cards haven't cut fraud as much as the industry hoped, as Figure 1 shows. Attackers have adapted in several ways. First, they moved from card cloning to card-not-present transactions, which include Internet, mail-order, and phone-based payments.

Table 1. Consecutive unpredictable numbers (UNs) from an ATM.

Date	Time	UN
2011-06-29	10:37:24	F1246E04
2011-06-29	10:37:59	F1241354
2011-06-29	10:38:34	F1244328
2011-06-29	10:39:08	F1247348

Second, attackers started making magnetic-strip clones of EMV cards. Instead of entering PINs only at ATMs, customers now also enter their PINs in POS terminals, which are much easier to tamper with.[1] Attackers steal card data and PINs and then use magnetic-strip clones in countries such as the US where ATMs still accept magnetic-strip cards.

Third, several technical vulnerabilities have emerged. For example, attackers can use a stolen EMV card at a POS device without knowing the PIN by using a man-in-the-middle device to trick a terminal into believing that the correct PIN was entered while the card

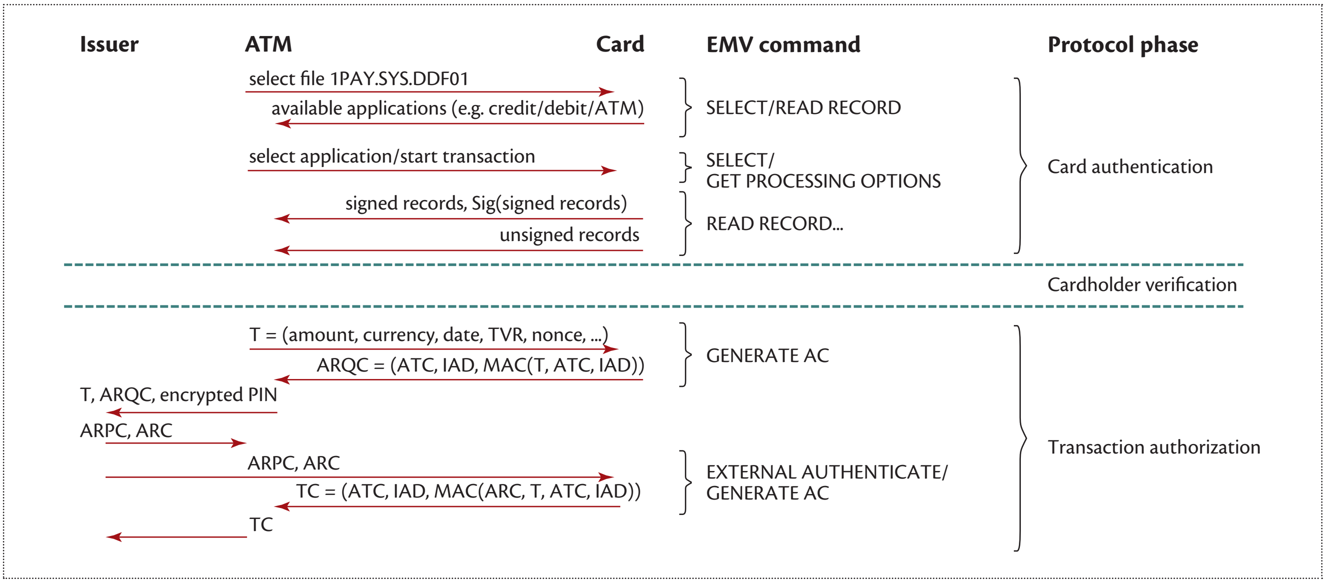
Issuer
ATM
Card
EMV command
Protocol phase
select file 1PAY.SYS.DDF01
available applications (e.g. credit/debit/ATM)
SELECT/READ RECORD
select application/start transaction
SELECT/
GET PROCESSING OPTIONS
Card authentication
signed records, Sig(signed records)
unsigned records
READ RECORD...
Cardholder verification
T = (amount, currency, date, TVR, nonce, ...)
ARQC = (ATC, IAD, MAC(T, ATC, IAD))
GENERATE AC
T, ARQC, encrypted PIN
ARPC, ARC
Transaction authorization
ARPC, ARC
EXTERNAL AUTHENTICATE/
GENERATE AC
TC = (ATC, IAD, MAC(ARC, T, ATC, IAD))
TC

Figure 2. Outline of an EMV transaction at an ATM. Although the messages between the card and ATM have been verified, messages between the issuer and ATM might vary depending on card scheme rules.

thinks it's authorizing a chip-and-signature transaction.[2] Attackers have already gone on trial in France for exploiting this no-PIN vulnerability.[3]

Still, a lot of fraud is unexplained and is often blamed on the cardholder. Because of this, there is a public interest in discovering new vulnerabilities, and the preplay attack explains several fraud cases reported to us by cardholders in Spain, Poland, and the Baltic states.

The Preplay Attack

EMV transactions consist of three phases:

- *card authentication*—the ATM or POS terminal reads and authenticates the card details;
- *cardholder verification*—the ATM or POS terminal verifies the person who presents the card, either by PIN or signature; and
- *transaction authorization*—the issuing bank decides whether the transaction should proceed.

Figure 2 illustrates these phases for an ATM transaction.

The principals are the card, the ATM or POS device, and the issuer. The third phase is of interest for the preplay attack. In transaction authentication, the ATM or POS terminal sends the card the amount, currency, date, terminal verification results (results of various checks the ATM or POS terminal performed), and a nonce (the unpredictable number in EMV terminology). The card responds with a MAC known as the authorization request cryptogram (ARQC), which is calculated over these records; the application transaction counter (ATC; a 16-bit number stored by the card and incremented for each transaction); and the issuer application data (IAD; a proprietary data field used to carry information from the card to its issuer).

The ARQC is sent to the issuer, which verifies it and checks whether funds are available, that the card hasn't been reported stolen, and that the transaction doesn't look suspicious. It then returns to the ATM or POS terminal an authorization response code (ARC) and an authorization response cryptogram (ARPC). The card verifies the ARPC (which is typically a MAC over the ARQC XORed with the ARC) and returns an authenticated settlement record known as a transaction certificate (TC), which can be sent to the issuer immediately or some time later as part of a settlement process. All these MACs are computed using a key shared between the card and the issuing bank, and so the ATM or POS terminal can't verify them.

In a normal EMV transaction, the card sends an ARQC to prove that it's alive, present, and engaged in the transaction. The ATM or POS device relies on the issuer to verify this and authorize the transaction. The UN ensures that transactions are unique and tied to a specific terminal. However, if attackers can predict UNs, they can mount preplay attacks that are indistinguishable from card cloning. That is, attackers collect authentication data at one terminal and play it to one or more other verifying parties later. For example, a tampered terminal in a store can collect card details and ARQCs from a victim for later use at an ATM or POS terminal whose UN can be predicted or manipulated.

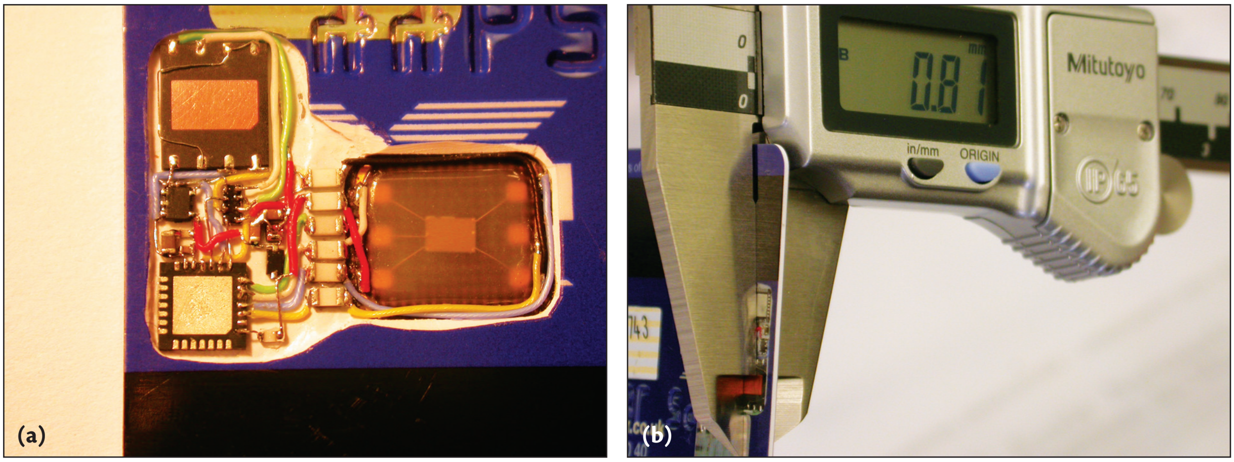

Figure 3. Passive monitoring card used to collect unpredictable number (UN) data. For the purposes of this data collection campaign, a standard debit card modified with an additional microcontroller and memory was used to observe the EMV system in practice.

EMV Protocol Flaws

The first protocol flaw is that the specification doesn't require the terminal's ID in authenticated messages, which is a classic mistake.[4] Although the EMV framework can support authenticating the terminal ID through a list of fields specifying what data should be sent to the card to generate the ARQC (the CDOL1), the standard cryptogram format developed by Visa (version 10) requires only the terminal country code in the computation of the MAC.

The second flaw is in the protocol architecture: the terminal generates the random number, but the issuing bank relies on it. Therefore, the issuer depends on the merchant for transaction freshness, but the merchant might not have the incentive to provide it and might be unable to deliver it correctly because of a lack of end-to-end authentication. In fact, the terminal could be collusive.

A formal analysis of EMV was published in 2011, but this flaw wasn't discovered.[5] The study made two errors. First, it modeled the UN as a fresh nonce, even though EMV doesn't require this. Second, it also modeled the issuer and terminal as the same principal, which they are not; the terminal communicates with an acquirer—that is, the merchant's bank—which in turn sends the transactions to a switch that finally relays the transactions to the issuer.

Preplay Attacks Based on a Weak RNG

The EMV protocol designers didn't think carefully about what was required for the UN to be truly unpredictable. The specifications and conformance testing procedures simply required that four consecutive transactions performed by the terminal should have unique UNs (see test 2CM.085.00[6]). Thus, a rational implementer in a hurry would simply use a counter.

Since we disclosed this flaw, the EMV 4.2 specification now offers guidance on generating the UN, but previous versions left the algorithm entirely up to implementers. Even the suggested construction (hash or XOR of previous ARQCs, transaction counter, and time) wouldn't be adequate if the ATM is rebooted and both the time and transaction counter are predictable.

UN Data Collection

A. Theodore Markettos and Simon Moore first showed that a preplay attack was possible against EMV if attackers could sabotage the RNG.[7] However, before our work, there was no empirical work on the quality of the RNGs used by actual ATMs or POS terminals. To address this, we set out to collect UNs generated by ATMs and POS terminals in our area, which together with log files from legal cases gave us an initial view of the EMV system in practice.

To obtain UN data and high-resolution timestamps from ATMs, we made a set of passive monitoring cards by adding our own ATM protocol analyzer circuitry, consisting of a microcontroller and memory, to a standard debit card. This process required carefully placing and connecting small components (down to a 0.4 mm pin pitch) and custom hardware to retrieve the transaction logs. Figure 3 shows the modified card. It remains a valid payment card—the transaction flow proceeds as normal—so it should always be accepted. It can also be inserted into a variety of ATMs and POS devices without arousing suspicion. (For ethical and prudential reasons, we informed the police that such experiments were underway, and we also went through our local ethics process.)

For each ATM investigated, we harvested between five and 50 UNs by performing repeated balance inqui-

Table 2. Categorized UNs from various ATMs.

Terminal no.	Weak random number generator (RNG)
ATM 1	690D4DF2
ATM 1	69053549
ATM 1	660341C7
ATM 1	5E0FC8F2
ATM 2	6F0C2D04
ATM 2	580FC7D6
ATM 2	4906E840
ATM 2	46099187
ATM 3	650155D7
ATM 3	7C0AF071
ATM 3	7B021D0E
ATM 3	1107CF7D
ATM 4	EB661DB4
ATM 4	2CB6339B
ATM 4	36A2963B
ATM 4	3D19CA14
ATM 5	F1246E04
ATM 5	F1241354
ATM 5	F1244328

Table 3. Categorized UNs from local point-of-sale (POS) terminals.

Terminal no.	Stronger RNGs
POS 1	013A8CE2
POS 1	01FB2C16
POS 1	2A26982F
POS 1	39EB1E19
POS 1	293FBA89
POS 1	49868033

ries and then making a small cash withdrawal. (All ATM transactions are authenticated by EMV protocol runs, but some have a zero withdrawal amount.) We used balance inquiries to minimize the number of withdrawals and avoid triggering any fraud-monitoring systems.

At POS terminals, sales assistants are often briefed to avoid handling or even looking at customer cards. So we used existing monitoring tools such as the Smart Card Detective,[8] which relies on a hidden wire running up the experimenter's sleeve.

During our UN collection campaign, we performed more than 1,000 transactions across 22 different ATMs and five POS terminals. Table 2 shows a selection of data collected from various ATMs exhibiting some ineffective algorithms. ATMs 1 and 2 contain a typical pattern, which we denote characteristic C, in which the high bit and the third nibble of each UN are always set to zero. This alone reduces the entropy of the UNs from 32 to 27 bits. Eleven of the 22 ATMs we looked at exhibited this characteristic. These included ATMs of wildly different ages running different operating systems, so we suspect it's an artifact of a particular EMV kernel postprocessing rather than of the RNG source itself.

Table 3 shows a list of consecutive UNs retrieved from a local POS terminal with a stronger RNG. Even in this case, the first bit appears to remain 0, which might suggest the use of a signed integer.

Based on our analysis of RNGs from logs, ATMs, and POS terminals, we distinguish three broad classes of ineffective RNGs:

- an obviously weak RNG algorithm, such as counters or clocks directly used as the UN, homegrown algorithms, or casting down to the wrong integer size;
- a simple RNG with little or no seeding, such as a linear congruential generator, combinations of fixed bits and bits that cycle, or using standard C library calls such as `time()` and `rand()`; and
- an RNG that can be put into a predictable state, such as a strong RNG fed by a weak source of randomness that's restarted on power-up or an RNG that relies only on data from previous transactions.

Harvesting the Data and Cashing Out

Given temporary access to an EMV card whose holder enters the PIN, attackers could program a terminal to read the static data from the card and call `GENERATE AC` to obtain an ARQC and TC for each possible UN. In this way, several dozen ARQCs can be harvested for each card. The only limitation is the time that the card can be left in a sabotaged POS while the customer believes the machine is waiting for authorization.

In the Majorca ATM that started this line of research, the counter rolled over every three minutes. In this case, an attacker might ask a card in a store for 20 ARQCs around a point in the 15-bit counter's cycle. On visiting the ATM, the attacker's card would first calibrate to the ATM's counter and then initiate transactions when the counter is expected to be in the range of the captured ARQCs. Figure 4a illustrates the preplay attack based on a weak RNG.

Preplay Attack Implementation and Evaluation

In our indistinguishability experiment, we used test cards with known ARQC-generation keys to prove the attack's viability. First, we took two test cards, A and B,

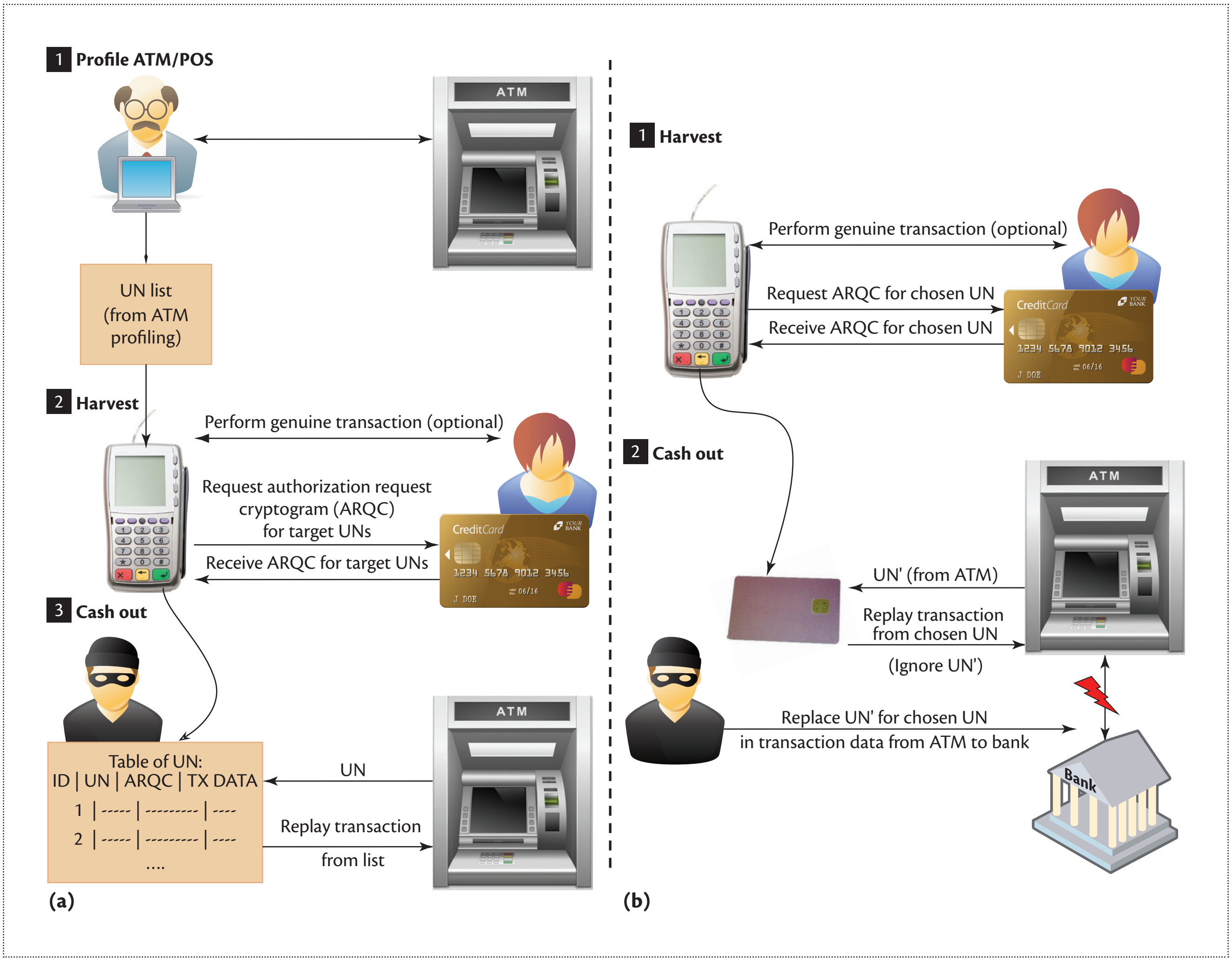
1 Profile ATM/POS
ATM
UN list (from ATM profiling)
2 Harvest
Perform genuine transaction (optional)
Request authorization request cryptogram (ARQC) for target UNs
Receive ARQC for target UNs
CreditCard
YOUR BANK
1234 5678 9012 3456
06/16
J DOE
3 Cash out
Table of UN:
ID | UN | ARQC | TX DATA
1 | ----- | --------- | ----
2 | ----- | --------- | ----
....
UN
Replay transaction from list
ATM
(a)
1 Harvest
Perform genuine transaction (optional)
Request ARQC for chosen UN
Receive ARQC for chosen UN
2 Cash out
ATM
UN' (from ATM)
Replay transaction from chosen UN
(Ignore UN')
Replace UN' for chosen UN in transaction data from ATM to bank
Bank
(b)

Figure 4. Overview of the preplay attack: (a) using a weak random number generator and (b) tampering with the UN at an ATM or POS terminal.

loaded with the same ARQC-generation keys, initialized with the same ATC, and handled identically. Then, we harvested data from card A and programmed it onto a preplay card, implemented using the ZeitControl BasicCard platform (www.basiccard.com). Finally, we compared traces between the preplay card version of card A and the real card B and observed that they were identical. This means that, at the protocol level, it's impossible for an ATM to distinguish between the real and preplay cards.

UN Modification and Other Attacks

In real life, we cannot rely on communications between the merchant and the card-issuing bank to be protected by encryption or even authentication. This is a well-known problem from ATM networking.[9] In such cases, a man-in-the-middle device between the terminal and the bank can be used to attack systems even if their random number generation is sound. In this case, it's no longer necessary to profile an ATM or POS terminal. Attackers can simply choose an arbitrary UN and obtain the related transaction data, including the ARQC, from the victim's card. They then replay the transaction data at a terminal and replace the terminal's real UN with another one (see Figure 4b).

This could be an attractive way to attack merchants with high-value transactions, such as jewelers or investment firms. Even if merchants guard their premises and their POS equipment, attackers can go after the network link at a utility cabinet. And even if the UN generation algorithms are patched or the communication link between merchants and their issuer bank is secured, several other protocol attack variants exist.

Malware Infection

There have been numerous cases of malware-infected ATMs operating in Eastern Europe and of POS devices being infected in the US. Depending on the internal architecture, it might be easy for such malware to sabotage the choice of UN. In fact, one bank suggested to us that the Majorca ATM might have been infected with malware.[10] Alternatively, the malware might collude to ensure that the UN matches the presented ARQC.

Supply-Chain Attacks

Such attacks have already occurred on POS terminals in the wild and used to harvest magnetic-strip data. Thus, it's feasible that attackers, or even state-level adversaries, might sabotage the RNG deliberately, either to act predictably all the time or to enter a predictable mode when triggered via a covert channel. A suitably sabotaged RNG would probably be detected only via reverse engineering or by observing real-world attacks.

Collusive Merchants

In a transaction dispute, a customer claimed to have made a single small purchase at a merchant, yet the bank claimed the customer made 10 large purchases as well. These large purchases were filed via three different acquirers who reported different terminal characteristics despite each transaction coming from the same terminal, so the evidence of fraud is clear. The bank initially maintained that the transactions were the customer's fault. The customer hired a lawyer who engaged one of the authors as an expert witness, took the case to the Financial Ombudsman Service, and got a full refund. Banks' unwillingness to charge back such transactions to merchants, unless customers go to court, is an open incitement to merchant fraud.

Merchants might maliciously modify their EMV stack to be vulnerable or inject replayed card data into the authorization/settlement system. They could take a cut from attackers using cloned cards at the store or just preplay transactions directly. We also have evidence of merchants tampering with transaction data to falsely represent transactions as PIN verified to shift liability and cut transaction fees. In the UK, there was a string of card-cloning attacks at petrol stations in which gang members bribed store managers to look the other way when they tampered with PIN pads and inserted monitoring devices into network connections—exactly what's needed for a preplay attack.

Limitations and Defenses

Limits on the preplay attack's effectiveness relate to the data fields included in the MAC and the quality of the issuer's cryptographic checks. When issuers follow card scheme standards, the attacker must choose the transaction country, date, and amount in advance. The card's PIN must be harvested at the same time as the preplay data (or already be known). Also, subsequent use of the real card might advance the ATC kept by the card and invalidate recorded data.

Defenses against Random Number Attacks

The simplest fix for random number attacks is a cryptographically secure RNG, but this isn't necessarily practical. RNG design is a matter for the acquiring banks, ATM vendors, merchants, and POS terminal suppliers, whereas the cost of fraud falls on card-issuing banks and customers. Issuers might unilaterally try to detect evidence of harvesting, such as by identifying large gaps in ATC. They should reject online transactions with out-of-order ATCs, but this is easier said than done because transaction reordering can occur in offline payments. We've witnessed banks processing duplicated transactions without checking ATCs.

> **"Banks' unwillingness to charge back such transactions to merchants, unless customers go to court, is an open incitement to merchant fraud."**

Defenses against Protocol Attacks

In the short to medium term, issuers would do better to meticulously verify the TC, which the card sends to the terminal as the transaction completes and should be submitted to the issuer when the transaction is presented for settlement. The TC states whether the card verified the ARPC, which in turn was computed by the card-issuing bank after it verified the ARQC. A preplay attack can still yield a TC, but its IAD will show that issuer authentication didn't complete successfully.

Still, TC checking is rare. Section 5.12 of Visa's "Transaction Acceptance Device Guide" states:[11]

> Devices operating in a single-message or host-capture environment should ensure a TC is generated for approved transactions. Although not needed for clearing, generating a TC ensures that cards do not request unnecessary online approvals on subsequent transactions and also provides liability protection for acquirers.

Mitigating acquirer liability in the event of stand-in processing is all very well, but our concern here is cardholder liability.

In the event of a court having to decide whether a series of disputed transactions from a single terminal was made with the cardholder's collusion or via a preplay attack, the first forensic test should be to examine the TC. If a valid TC is generated by a card following a correct ARPC that in turn followed a correct ARQC, then the card was present and active when the ARPC was generated. This doesn't exclude all possibilities of fraud, as relay attacks might occur,[12] but these are unlikely.

Discussion

In previous work, Steven J. Murdoch discussed the potential vulnerability of EMV given a poor RNG.[13] Markettos and Moore also explored how otherwise secure true RNGs could be manipulated to produce more deterministic output and how to exploit a weak RNG in an EMV transaction.[7] This article is the first work to show that EMV ATMs with poor RNGs exist in the wild, that they have been implicated in fraud, how they can be exploited, and the protocol flaws in the EMV specification that make this so hard to counter.

It's interesting to compare the preplay attack to full cloning, in which the ARQC generation keys are extracted. Full cloning might appear to be much more powerful, but because each card has its own ATC, these will diverge in due course and become detectable.

In fact, preplay attacks could be more powerful than full cloning attacks for reasons of scale. If keys could be extracted from cards at no cost—say, using a power analysis attack conducted by a rogue terminal—then cloning attacks might be done on an industrial scale. However, this is unlikely because the industry has spent 15 years and millions of dollars on countermeasures. It's more likely that cloning attacks would involve card capture followed by destructive reverse engineering and perhaps a semi-invasive attack costing tens to hundreds of dollars per card.

In contrast, preplay attacks could reach a massive scale. If an attacker succeeds in compromising several terminals (which was done in the UK by three separate gangs in the mid-2000s[14]) or compromising the communications to several high-value stores (which was done to jewelry stores in Hatton Garden, London, in the 1980s), the cards can have ARQCs harvested in one location and presented in another. The same holds if a number of terminals are compromised by malware.

EMV has been around for more than 10 years, yet attackers continue to identify serious new attack strategies. It's shocking that many ATMs and POS terminals have seriously defective RNGs that leave the system open to fraud. Even worse, the associated protocol failures are a growing real-world problem that leaves technology open to fraud at scale using malware in POS terminals.

This flaw challenges current thinking about authentication. Existing verification models don't easily apply to a complex multistakeholder environment. In fact, EMV was verified to be secure. As we explained, that verification didn't work, and mechanisms for rolling out fixes across networks with huge installed bases of cards and terminals and strong externalities are nowhere near serviceable.

The structural governance failure we've exposed gives rise to systemic risk. In a multiparty world, where not even the largest card-issuing bank or acquirer or scheme operator has the power to fix a problem unilaterally, we cannot continue to rely on a slow and complex negotiation process among merchants, banks, and vendors. Regulators have been credulous in accepting industry assurances about operational risk management, and it's time for them to take an interest. The US Federal Reserve is now paying attention, but it is time for European and other regulators to follow suit. ■

Acknowledgments

Steven J. Murdoch is funded by the Royal Society. Marios O. Choudary was a recipient of the Google Europe Fellowship in Mobile Security, and this research was supported in part by that fellowship. The opinions expressed in this article do not represent the views of Google.

References

1. S. Drimer, S.J. Murdoch, and R. Anderson, "Thinking inside the Box: System-Level Failures of Tamper Proofing," *Proc. IEEE Symp. Security and Privacy*, 2008, pp. 281–295.
2. S.J. Murdoch et al., "Chip and PIN Is Broken," *Proc. IEEE Symp. Security and Privacy*, 2010, pp. 433–446.
3. S. Sellami, "L'imparable Escroquerie a la Carte Bancaire" [The Unstoppable Scam at Carte Bancaire], *Le Parisien*, 24 Jan. 2012; www.leparisien.fr/faits-divers/l-imparable-escroquerie-a-la-carte-bancaire-24-01-2012-1826971.php.
4. R. Anderson and R. Needham, "Programming Satan's Computer," *Computer Science Today*, LNCS 1000, Springer, 1995, pp. 426–441.
5. J. de Ruiter and E. Poll, "Formal Analysis of the EMV Protocol Suite," *Proc. Theory of Security and Applications* (TOSCA 2011), S. Moedersheim and C. Palamidessi, eds., LNCS 6693, Springer, 2011, pp. 113–129.
6. "EMVCo Type Approval Terminal Level 2 Test Cases," version 4.3a, EMVCo, Nov. 2011.
7. A.T. Markettos and S.W. Moore, "Frequency Injection Attack on Ring-Oscillator-Based True Random Number Generators," *Proc. Workshop Cryptographic Hardware and Embedded Systems*, 2009, pp. 317–331.
8. O. Choudary, "The Smart Card Detective: A Hand-Held EMV Interceptor," tech. report UCAM-CL-TR-827, Univ. of Cambridge, Computer Laboratory, Dec. 2012.

9. R. Anderson, *Security Engineering: A Guide to Building Dependable Distributed Systems*, Wiley, 2003, p. 336.
10. J. Kirk, "ATM Malware May Spread from Mexico to English-Speaking World," *PC Magazine*, 28 Oct. 2013; www.pcworld.com/article/2058360/atm-malware-may-spread-from-mexico-to-englishspeaking-world.html.
11. "Transaction Acceptance Device Guide (TADG)," version 2.2, Visa, July 2014; http://technologypartner.visa.com/download.aspx?id=32.
12. S. Drimer and S.J. Murdoch, "Keep Your Enemies Close: Distance Bounding against Smartcard Relay Attacks," *Proc. 16th USENIX Security Symp.*, 2007, article 7.
13. S.J. Murdoch, "Reliability of Chip and PIN Evidence in Banking Disputes," *Digital Evidence and Electronic Signature Law Rev.*, vol. 6, Nov. 2009, pp. 98–1155.
14. "Petrol Firm Suspends Chip-and-Pin," BBC News, 6 May 2006; http://news.bbc.co.uk/2/hi/uk_news/england/4980190.stm.

Mike Bond is a visiting industrial research fellow at the University of Cambridge Computer Laboratory. His research interests include payment systems security and API security, in particular of hardware security modules. Bond received a PhD in computer security from the University of Cambridge. Contact him at mike.bond@cl.cam.ac.uk.

Marios O. Choudary is a junior lecturer at the University Politehnica of Bucharest. His research interests include side-channel analysis, banking security, and communication protocols. Choudary received a PhD in computer science from the University of Cambridge, where he was funded by the Google Europe Fellowship in Mobile Security. Contact him at marios.choudary@cs.pub.ro.

Steven J. Murdoch is a Royal Society University Research Fellow at University College London, where he conducts research on banking security, anonymous communications, privacy, and usable authentication. Murdoch received a PhD in computer science from the University of Cambridge. Contact him at s.murdoch@ucl.ac.uk.

Sergei Skorobogatov is a senior research associate at the University of Cambridge Computer Laboratory and a member of the Security Group. His research interests include hardware security analysis of smart cards, microcontrollers, FPGAs, SoCs, and ASICs. Skorobogatov received a PhD in computer security from the University of Cambridge. Contact him at sergei.skorobogatov@cl.cam.ac.uk.

Ross Anderson is a professor of security engineering at the University of Cambridge, where he manages and conducts research on security protocols, mobile platforms, payment systems, and the psychology and economics of security. Anderson received a PhD in robust computer security from the University of Cambridge. He's a fellow of the Royal Society and the Royal Academy of Engineering. Contact him at ross.anderson@cl.cam.ac.uk.