First Contact: New vulnerabilities in Contactless Payments

Leigh-Anne Galloway

Tim Yunusov

Aleksei Stennikov

December 4, 2019

Updated December 6, 2019

Document Version 1.1

ABSTRACT

In wide use since 2007, contactless (NFC) payments have been a part of commerce for more than a decade. Contactless transactions now account for more than 40 percent of transactions globally. With U.S. adoption set to grow significantly over the next two years, just how secure and safe are contactless payments?

The EMV protocol is inherently flawed. Here we show two vectors of attack. In the first, we show how it is possible to bypass cardholder verification limits for contactless payment cards. Circumvention also works against mobile wallets using locked cell phones. Weaknesses in implementation of these limits enable for criminals to bypass restrictions intended to cap fraud-related losses.

In the second, flaws in the values of generation keys, unpredictable number (UN), and application transaction counter (ATC) allow for reuse of transaction data. This makes it possible to carry out pre-play attacks against contactless cards using EMV modes. We examine why this is possible and perform a demonstration of a pre-play attack.

BACKGROUND

Contactless payments make use of payment protocols that have been around for much longer than the technology itself. Contactless supports two operating modes: EMV (chip-based cards) and magstripe. Magstripe is considered less secure, and for good reason. By comparison, EMV modes are more secure and more complex.

Our research focuses on EMV chip operating modes used for contactless payments. We demonstrate two classes of contactless attack vectors: limit circumvention (exceeding maximum transaction amounts) and pre-play/replay attacks.

In the first attack, we bypass hard limits for Visa cards. Limits are intended to prevent cards from making a payment over a specified value. Hard limits result from a combination of limits set on the terminal, forming an upper bound on the amount of a transaction. Where these limits are implemented, they vary based on geographic location. In the U.K the limit for physical cards is set to £30. By contrast, soft limits allow cardholders to pay for items over a specified value with the addition of cardholder verification. Soft limits are present in much of the rest of Europe and in the U.S. In the U.S. the limit for physical cards is set to $50. A combination of limits is implemented for mobile wallets, Google Pay, and Apple Pay; contactless payments up to $10,000/£5,500 can be made as long as cardholder verification is supplied by the device.

We show how to bypass limits for cards without cardholder verification. In addition to this, we will demonstrate how to circumvent limits on mobile wallets for locked devices. This is a significant concern because issuing banks rely on limits to cap fraudulent transactions. This attack vector permits a criminal to make fraudulent transactions for much larger amounts than the intended limit. In our testing, we were able to make single payments for up to £100 in the U.K.

For the second attack, we demonstrate how to carry out pre-play attacks without downgrading the payment mode using pure EMV kernels. Previous research on pre-play attacks has relied on downgrading the operating mode to a less secure mode such as MSD/magstripe. The method described here does not require the attacker to downgrade to a legacy mode.

Contactless technology

NFC is a short-range wireless technology based on ISO/IEC-14443. [1] There are many applications for NFC. We most commonly interact with NFC when boarding public transport, paying for goods with NFC cards and mobile phones, or using a smart passport or hotel key.

Figure 1. Example of NFC payment technology

NFC contactless cards are known as a proximity IC card (PICC). The terminal is known as a proximity coupling device (PCD). The card itself doesn't contain any on-board power. Instead, the terminal provides power to the card by means of inductive coupling.

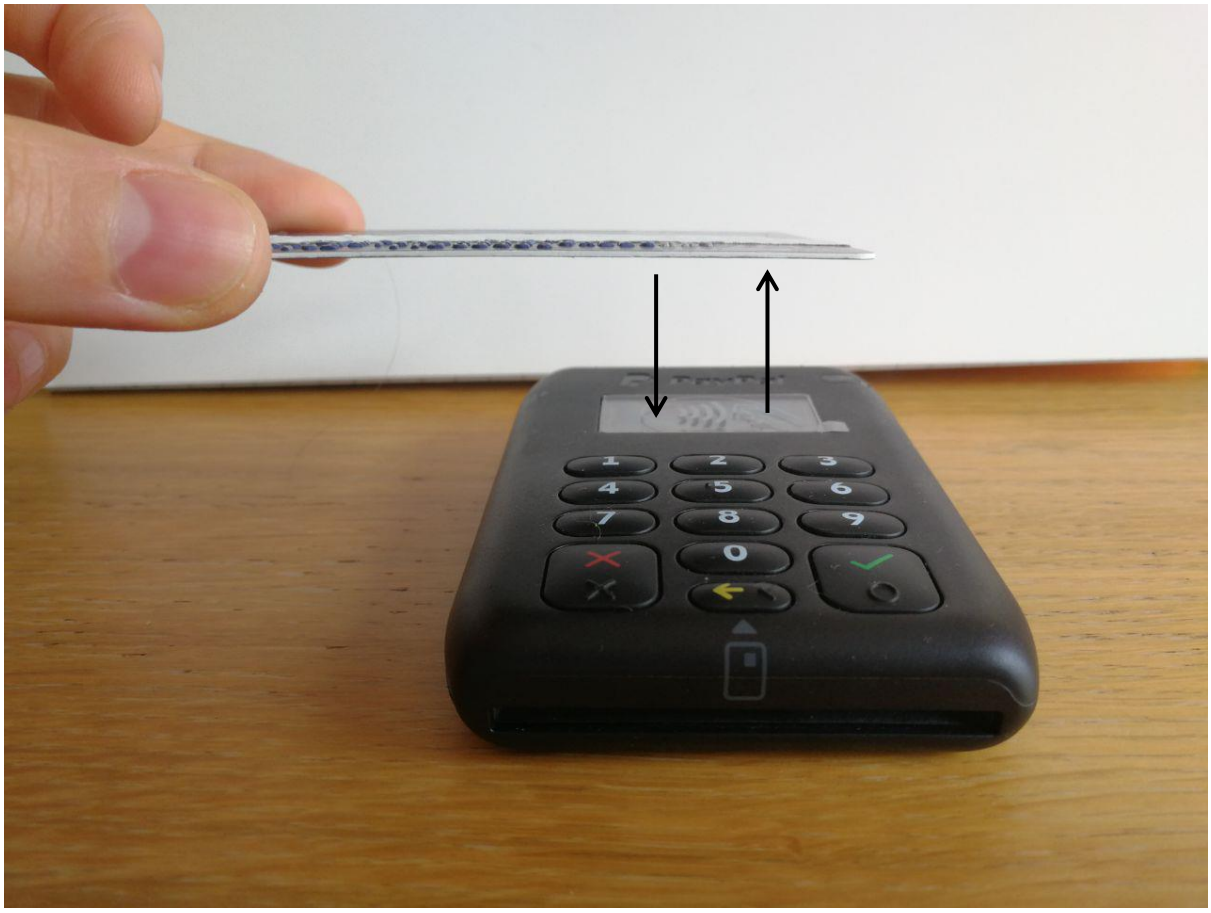

Figure 2. NFC uses inductive coupling. The terminal initiates the transaction and controls the exchange of data. It's a common misconception that the card initiates the transaction. In reality, the terminal begins the process by waking up the card.

EMV

EMV is a syndicate comprising the major card brands: Europay (merged with MasterCard), MasterCard, and Visa. EMV is the global standard for chip-enabled smart cards. NFC is the technology that allows smartcards to make contactless EMV transactions.

Figure 3. A smart card

The protocol itself consists of a subset of protocols, called kernels, which are used to transmit information. Visa describes EMV kernels as "a set of functions that provides the processing logic and data that is required to perform an EMV contact or contactless transaction." [2] The exact mechanisms in each kernel are described in specification documents available on the EMVCo website. [3] Much about EMV is undocumented and regarded as proprietary. Add to this that the documentation largely fails to provide any overview of the EMV operating logic. This significantly hinders payment research. For readers interested in a better understanding of EMV, we recommend "EMV in a nutshell." [4] This paper greatly helped us to fill in the gaps between the specifications and our own understanding. The vulnerabilities we describe in this document involve kernel 1 and kernel 3 used in Visa qVSDC, commonly referred to as Visa payWave. The exact mappings for each kernel are as follows [5]:

- Kernel 1 for some Visa and JCB cards
- Kernel 2 for MasterCard
- Kernel 3 for Visa
- Kernel 4 for American Express
- Kernel 5 for JCB
- Kernel 6 for Discover
- Kernel 7 for UnionPay

Contactless payment transaction

In order to understand where vulnerabilities exist in the payment process, it's important to know more about the flow of processes between the card and terminal. This flow is different for Visa than for MasterCard. Almost all communication between the terminal and card is sent using the Tag Length Value (TLV) encoding structure. Every data object is sent using this format. This is a hierarchical structure in which tags may be nested within other tags.

```
6f 47 84 0e 32 50 41 59  2e 53 59 53 2e 44 44 46
30 31 a5 35 bf 0c 32 61  30 4f 07 a0 00 00 00 03
10 10 50 10 56 69 73 61  20 20 20 20 20 20 20 20
20 20 20 20 87 01 03 9f  0a 08 00 01 05 01 00 00
00 00 bf 63 04 df 20 01  80 90 00
```

Figure 4. An example of the TLV structure: this is the card's response to the terminal, as described in Step 2

If we read the first line, we see that 6F [6] tells us that this is the File Control Information (FCI) Template. "47" shows us that this data object is 71 bytes long. "84" is the Dedicated File name (DF). Then we have the binary value "32 50 41 59 2e 53 59 53 2e 44 44 46 30 31", or "2PAY.SYS.DDF01" in ASCII. This indicates that the card works with the Visa payWave Proximity[1] Payment System Environment (PPSE).

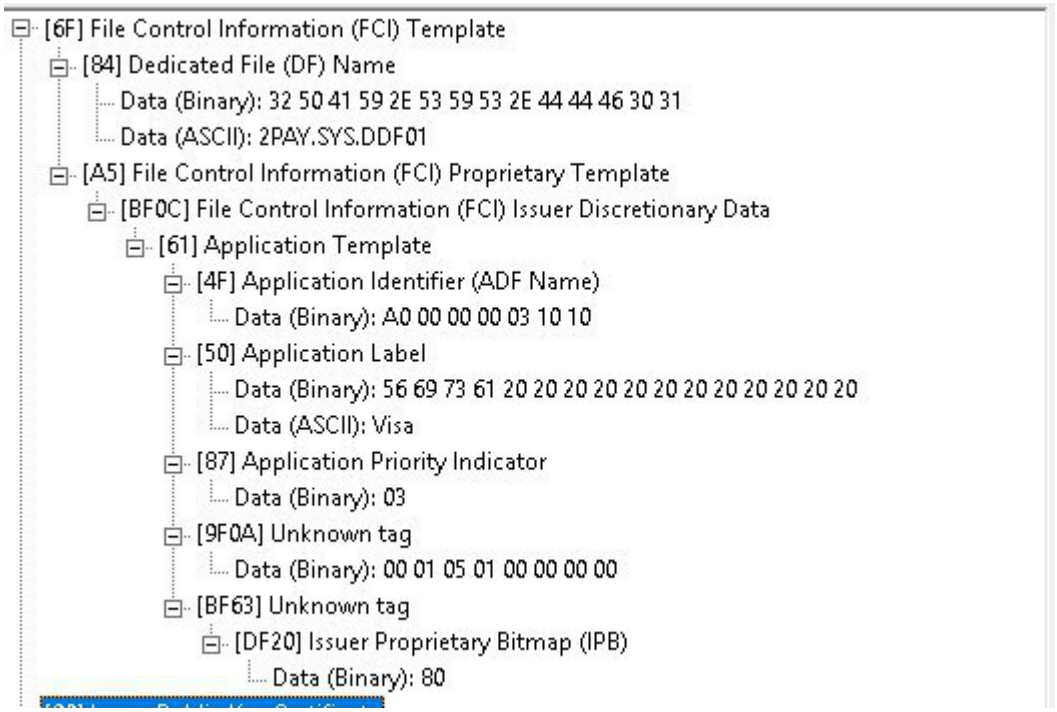

Figure 5. Logic structure of the TLV data shown in Figure 4

[1] So called because communication is done via NFC. For a transaction when the chip is inserted, this is called the Payment System Environment (PSE).

Here is the logic of interaction between the card and terminal for Visa cards:

1. The terminal initiates the transaction and asks the card "Which application and modes do you support? Visa? MasterCard? American Express?" This process involves reading the Proximity Payment System Environment (PPSE).

2. The card answers with the environment and modes that it supports. This information is returned in order of priority: "I work with the Visa debit/credit data structure. Also, I support quick Visa Smart Debit/Credit (qVSDC)." The card responds with its Application Identifier (AID).

3. The terminal then responds: "I want to work with your Visa debit/credit structure." The terminal selects Visa payWave AID.

4. The card responds: "I support qVSDC and I need input to the following data fields in order to complete the transaction." The card provides its Processing Options Data Object List (PDOL).

5. The transaction may fail at this point if the terminal does not support the same modes as the card. If it does support the same modes, it will select one of three contactless modes. The three modes are magstripe mode, qVSDC and Visa Smart Debit/Credit (VSDC). In this scenario, our card supports qVSDC, so the terminal selects qVSDC and sends all the requested data fields to the card. This information is sent as unstructured data. It includes the amount, an unpredictable number (UN), the currency, and a list of Terminal Transaction Qualifiers (TTQ). The terminal issues a command to the card to generate the cryptogram using the information it has provided. This command is GENERATE_AC (Generate Application Cryptogram). For qVSDC this is not a separate command but rather is returned by the card in the next step.

6. If data received by the terminal is incomplete or the card does not support certain aspects of the transaction,[2] the card terminates the transaction. If data is complete, the card provides the cardholder information and signs the transaction with a cryptogram. The card provides a response to the terminal that includes; the Application Cryptogram (AC), the Application Transaction Counter (ATC), and the Primary Account Number (PAN), Track2 Equivalent data, and the Card Transaction Qualifiers (CTQ).

7. The terminal performs risk analysis using the information provided in the TTQ and CTQ. It decides which steps to take next. If the terminal determines that the transaction does not need a Cardholder Verification Method (CVM), then it sends all information to the acquirer, card networks, issuer and awaits the response.

[2] For example: if the transaction amount exceeds the card's country limit, the transaction will fail.

Risk analysis

Risk analysis is carried out at several points in the transaction: the terminal, payment provider, card networks, and issuer. Here we are concerned only with the risk analysis performed by the terminal. This process is described by EMVCo in the Book-B Entry Point Specifications [7] and Book-A. The terminal determines which steps to take next based on a combination of data elements and flags. A critical component of the contactless user experience is that it does not require a CVM. When a chip is inserted into the terminal, a CVM (typically a PIN or signature) is required. This provides assurance that the cardholder has authorized the transaction. With contactless card transactions there may be no assurance that the cardholder was present during the transaction. This is because contactless transactions often don't require a CVM.

The risk of fraud is offset by limits and the TTQ. Contactless transactions are managed by three limits on the terminal. These are the Reader Contactless Floor Limit, the Reader Contactless Transaction Limit, and the Reader CVM Required Limit. In turn, the Reader Contactless Transaction Limit is separated into two parts: the Reader Contactless Transaction Limit without Consumer Device Cardholder Verification Method (CDCVM) and the Reader Contactless Transaction Limit with CDCVM.

```
# Application Settings: Mastercard - Purchase
APP=A000000004
9C=00
9F01=00
9F06=A0000000041010
9F09=0002
9F6D=0001
9F15=0001
9F35=22
9F40=0000000000
9F7E=00
DF810C=02
DF8117=00
DF8118=60
DF8119=08
DF811A=9F6A04
DF811B=20
DF811E=10
DF811F=08
DF8120=FC50808800
DF8121=0000000000
DF8122=FC50808800
DF8123=000000000000
DF8124=000000003000
DF8125=000000550000
DF8126=000000003000
DF812C=00
DF812D=000013
70=9F1D082CB8000000000000
```

Figure 6. The three limits defined on a U.K. terminal for MasterCard purchases

In the configuration file in Figure 6, there are several limits defined by tags. The names and functions of these tags are publicly known:

Tag	Name	Value
DF8123	Reader Contactless Floor Limit	£0
DF8124	Reader Contactless Transaction Limit (No CDCVM)	£30
DF8125	Reader Contactless Transaction Limit (CDCVM)	£5500
DF8126	Reader CVM Required Limit	£30

The transaction limit for Reader Contactless Limit with CDCVM (£5500) is significantly higher than without CDCVM (£30). This means that a cardholder can make a transaction with a payment wallet, such as Google Pay or Apple Pay, for up to £5500. But a contactless transaction made with the same physical card issued in the UK has a limit of £30.

```
# Application Settings: Visa
APP=A000000003
9F01=00
9F06=A000000003
9F09=0001
9F15=0001
9F35=22

# Visa Dynamic Reader Limit Settings: Set 1
DRL=A000000003
9F5A=01
DF01=01
DF02=01
DF03=000000550001
DF04=000000000001
DF05=000000003001

# Visa Dynamic Reader Limit Settings: Set 2
DRL=A000000003
9F5A=02
DF01=01
DF02=01
DF03=000000550001
DF04=000000000001
DF05=000000003001

# Visa Dynamic Reader Limit Settings: Set 3
DRL=A000000003
9F5A=03
DF01=00
DF02=00
DF03=000000550001
DF04=000000000001
DF05=000000003001

# Visa Dynamic Reader Limit Settings: Set 4
DRL=A000000003
9F5A=04
DF01=00
DF02=00
DF03=000000550001
DF04=000000000001
DF05=000000003001
                                                    #
```

Figure 7. Limits on a U.K. terminal for Visa purchases

In the configuration file in Figure 7, we can see several limits defined for Visa cards. These tags are proprietary to Visa. We have determined these tags to be:

Tag	Name	Value
DF04	Reader Contactless Floor Limit	£0.01
DF05	Reader Contactless Transaction Limit (No CDCVM)	£30.01
DF03	Reader Contactless Transaction Limit (CDCVM)	£5500.01

```
# Application Settings: Mastercard - Purchase
APP=A0000000041010
9C=00
9F01=00
9F06=A0000000041010
9F09=0002
9F6D=0001
9F15=0001
9F35=22
9F40=0000000000
9F7E=00
DF810C=02
DF8117=00
DF8118=20
DF8119=08
DF811A=9F6A04
DF811B=20
DF811E=10
DF811F=08
DF8120=FC50808800
DF8121=0000000000
DF8122=FC50808800
DF8123=000000000000
DF8124=000001000000
DF8125=000001000000
DF8126=000000005000
DF812C=00
DF812D=000013
70=9F1D082428000000000000
```

Figure 8. The three limits defined on a U.S. terminal for MasterCard purchases

In the configuration file in Figure 8, there are several limits defined by tags. The names and functions of these tags are publicly known:

Tag	Name	Value
DF8123	Reader Contactless Floor Limit	$0
DF8124	Reader Contactless Transaction Limit (No CDCVM)	$10,000
DF8125	Reader Contactless Transaction Limit (CDCVM)	$10,000
DF8126	Reader CVM Required Limit	$50

```
# Application Settings: Visa
APP=A000000003
9F01=00
9F06=A000000003
9F09=0001
9F15=0001
9F35=22

# Visa Dynamic Reader Limit Settings: Set 1
DRL=A000000003
9F5A=01
DF01=01
DF02=01
DF03=000001000001
DF04=000000000001
DF05=000000005001

# Visa Dynamic Reader Limit Settings: Set 2
DRL=A000000003
9F5A=02
DF01=01
DF02=01
DF03=000001000001
DF04=000000000001
DF05=000000005001

# Visa Dynamic Reader Limit Settings: Set 3
DRL=A000000003
9F5A=03
DF01=00
DF02=00
DF03=000001000001
DF04=000000000001
DF05=000000005001

# Visa Dynamic Reader Limit Settings: Set 4
DRL=A000000003
9F5A=04
DF01=00
DF02=00
DF03=000001000001
DF04=000000000001
DF05=000000005001
                                                                    #
```

Figure 9. Limits on a U.S. terminal for Visa purchases

In the configuration file in Figure 9, we can see several limits defined for Visa cards. These tags are proprietary to Visa. We have determined these tags to be:

Tag	Name	Value
DF04	Reader Contactless Floor Limit	$0
DF05	Reader Contactless Transaction Limit (No CDCVM)	$50.01
DF03	Reader Contactless Transaction Limit (CDCVM)	$10,000.01

Again, the limits for contactless purchases made using a consumer device are much higher than those permitted with physical contactless cards.

In addition to these limits, the Terminal Transaction Qualifiers (TTQ) and Card Transaction Qualifiers (CTQ) are compared to make a risk-based decision on the terminal. The TTQ contains fields describing the terminal's requirements for proceeding with the transaction. The qualifiers include: support for magstripe, support for EMV, support for EMV contact chip mode, offline or online processing, online or offline PIN, signature support, offline or online data authentication, online cryptogram required, CVM required, and whether CDVM is supported. A full list of TTQ data fields is provided in the EMV Book-A Architecture and General Requirements, table 5-2. The CTQ is specified by the card issuer and includes the following information: is online PIN required, is signature required, and whether CDCVM has been completed.

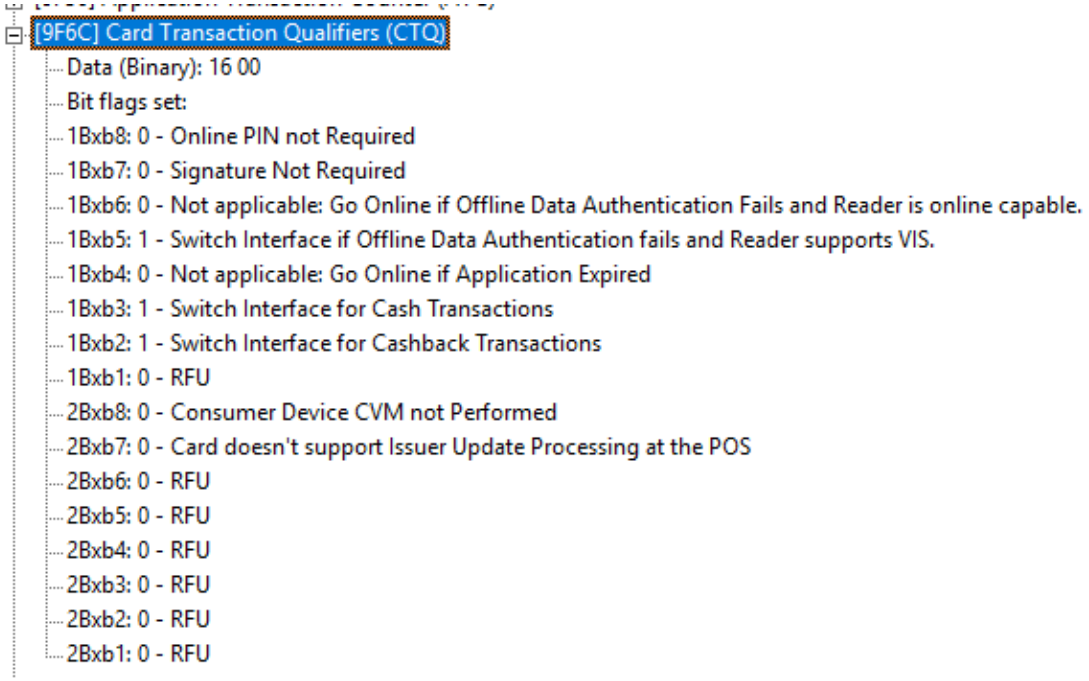

Figure 10. CTQ for a U.K.-issued Visa card making a contactless transaction of less than £30. Online PIN and signature are not required because this transaction is less than the U.K. limit.

Analyzing contactless transactions

During the card/terminal negotiation process, three data points are used to carry out risk analysis. These are the limits specified in the terminal configuration file, the TTQ, and the CTQ. If any one of these conditions is not met, then the transaction fails. In the following video, observe the card/terminal negotiation process for a U.K.- issued card and terminal for a transaction over £30. Communication between the card/terminal is directed through an NFC proxy so that the Application Protocol Data Units (APDU) can be read.

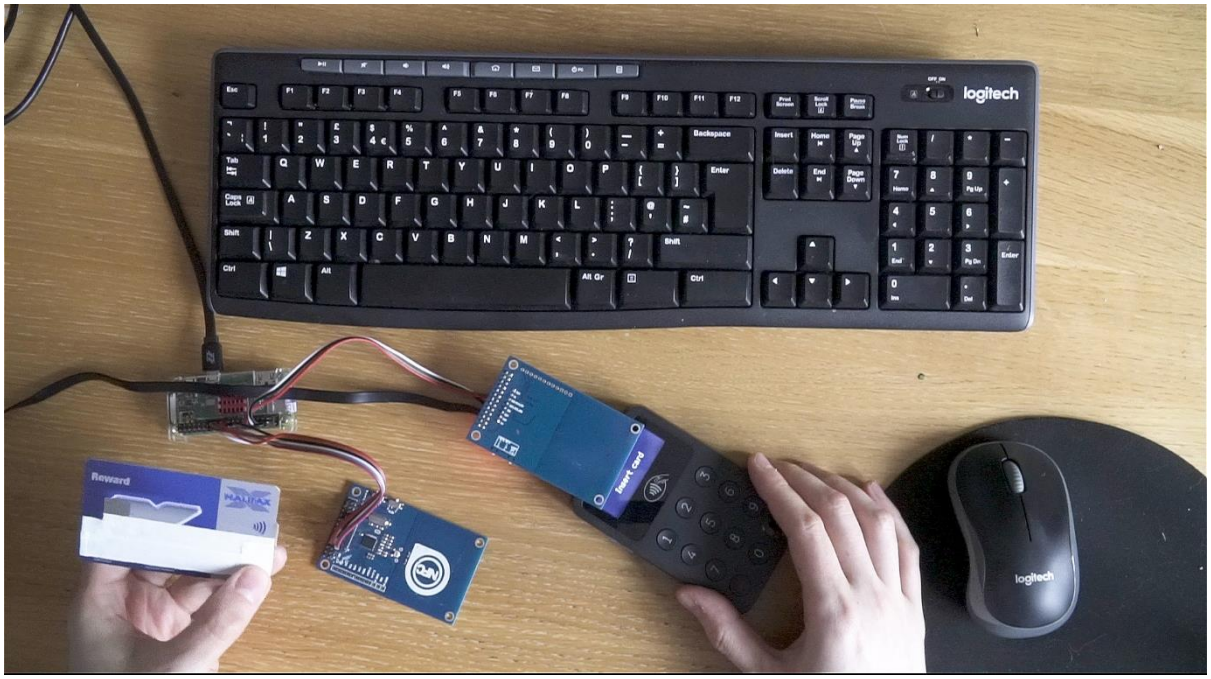

Figure 11. NFC proxy. The card (on the left) is a U.K.-issued card with a limit of £30 on contactless transactions. This is enforced by the terminal (on the right).

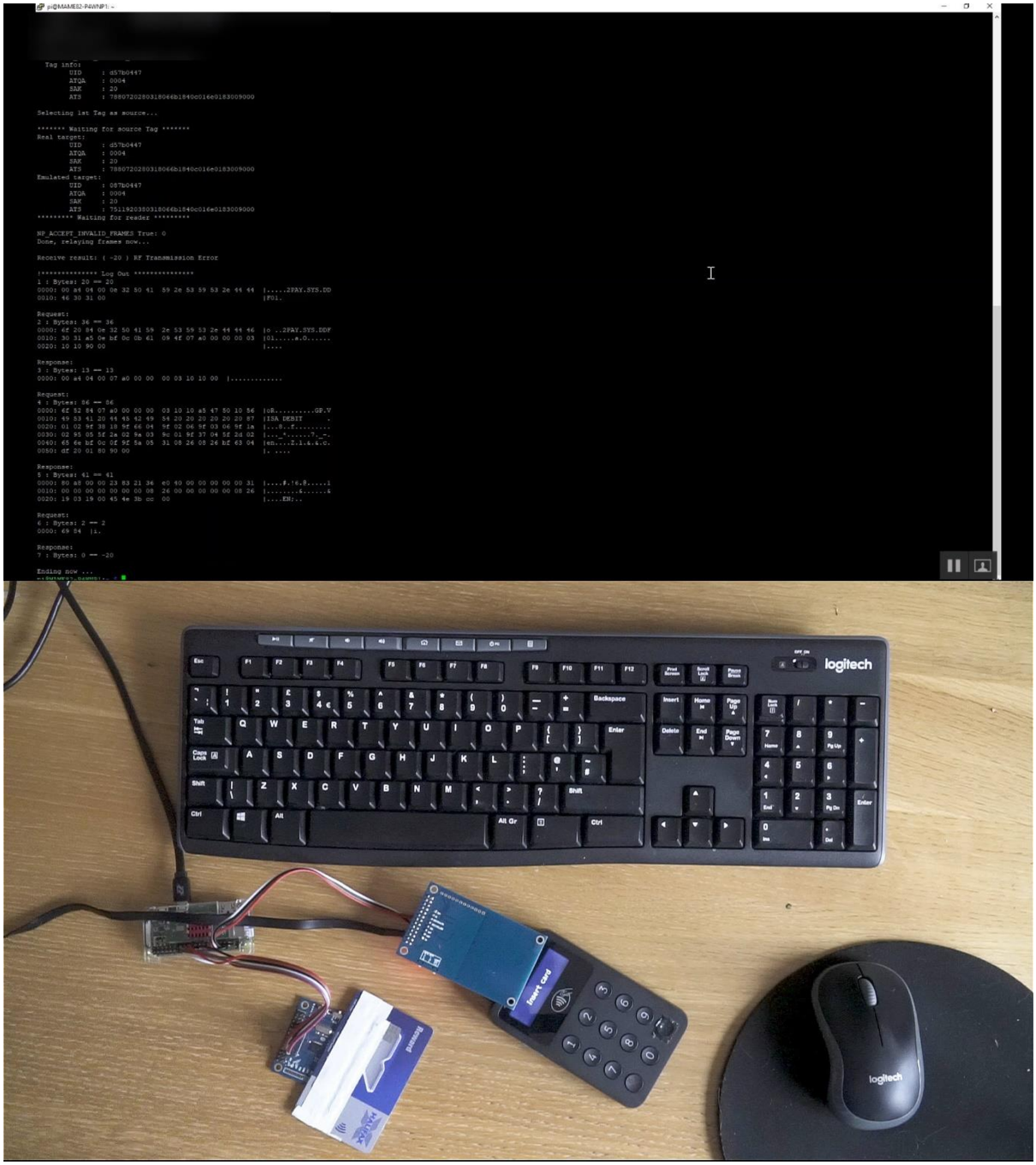

Figure 12. The terminal shows that in step six, the card does not send cardholder data and a cryptogram as expected. Instead the communication terminates and the terminal asks that the chip card be inserted instead.

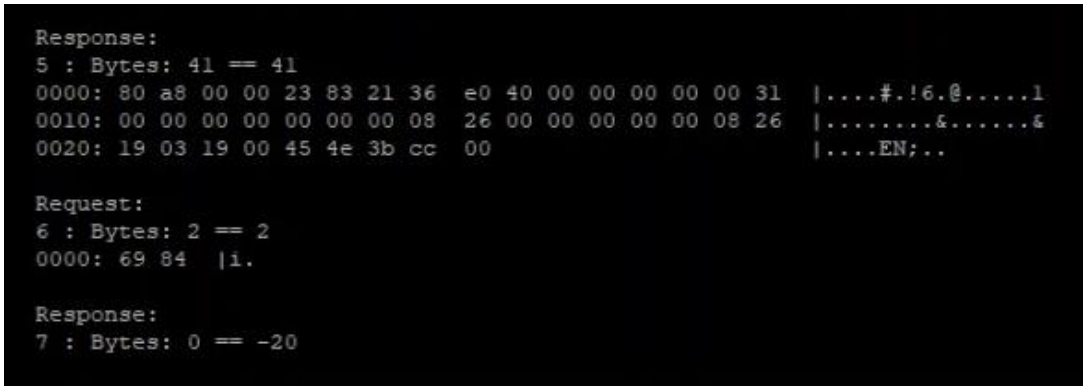

Figure 13. In request six, the card returns "69 84", which is the "End Application - parameter settings" response

The card returns the value "69 84". This is because the transaction exceeded the Visa card limit on contactless transactions (this limit is specified in the configuration file of the terminal). "69 84" is the end application parameter settings response (as defined in EMV Contactless Book-C, page 109). The outcome of this process is to "Insert, Swipe or Try Another Card."

If the transaction had been made using a consumer device, it would have passed the TTQ requirement and been sent to the acquirer. This is because limits for consumer devices are often much higher, and crucially, consumer devices can perform cardholder verification. CDCVM is performed entirely by the consumer device. Apple Pay uses Face ID, Touch ID, and passcodes to verify the cardholder. Google Pay requests that the cardholder unlock the device to perform CDCVM for higher amounts. For smaller transactions, the device simply needs to be "awake" with the screen active.

Authorization Request Cryptogram (ARQC)

Completion of CDCVM is indicated in the CTQ fields sent from the device to the terminal. The CDCVM flag is located in byte 2, bit 8. Changing this value from "0" to "1" instructs the terminal that CDCVM has been completed. How does the terminal know that CDCVM originates from a consumer device and not a physical card? Importantly, it doesn't. No other information sent to the terminal can verify that the CDCVM flag originates from a consumer device or that it originates from that specific device.

The EMV protocol is designed to prevent the re-use of transaction information and chip-based cardholder data. It does so by signing the transaction using a cryptographic function. The value of this is contained within the Authorization Request Cryptogram (ARQC). During the final stages of a payment, the terminal provides three fields to the card, which are used to generate the ARQC. These fields are the unpredictable number (UN), amount, and currency.

> *ARQC = cryptographic function (UN, amount, currency, Application Transaction Counter (ATC), date, other required fields)*

As part of this final stage, the card sends the ATC value in cleartext. The ATC increments for each transaction that is made, in order to prevent the transaction from being sent and authorized out of order. The ARQC is forwarded to the issuer in order to verify that the data is unique and untampered. The authorization server decrypts the cryptogram using a Hardware Security Module (HSM). The encrypted fields are compared to the unencrypted

fields to ensure that the data matches. If all the data matches, the ATC is sequential, and the cardholder has sufficient funds, then the issuer will authorize the request.

The UN generated by the terminal is 4 bytes in length. This is the main form of entropy for the cryptogram generated by the terminal. EMVco provides limited guidelines for UN generation, stating that "the Unpredictable Number could be generated by a dedicated hardware random number generator or could, for example, be a function of previous Application Cryptograms, the terminal Transaction Sequence Counter and other variable data (e.g. date/time)." [8]

In theory, each ARQC is unique because the UN is unique and the ATC increments. But in practice, issuers don't decline transactions with an ATC value that is lower than the current ATC. This seeming laxness is to allow for transactions that may have been made offline. Nor are there any restrictions to prevent the terminal from sending the same UN to the card each time. As such, the EMV protocol is inherently flawed. As we will proceed to show, weaknesses in the generation of the UN and validation of the ATC allow for pre-play and replay attacks.

Exceeding contactless payment limits by circumventing CDCVM

Given that the CDCVM source is not checked, and that consumer device transactions have much higher limits on terminals, it is possible to make a contactless transaction with a physical card that exceeds the limit implemented on the terminal. It also means that a large payment can be made with a consumer device without performing CDCVM on the device itself, with only the screen active.

This attack requires that CDCVM be supported by the terminal in the TTQ. Usually this isn't an issue because most terminals support CDCVM in their settings. In addition to this, two fields need to be modified:

1. In the TTQ sent to the card, the CVM required field (byte 2, bit 7) is changed from "1" to "0". This field indicates whether to require additional cardholder verification for the transaction. For transactions under the limit set on the terminal, this field is set to "0". By changing this bit to "0" the terminal tells the card it doesn't require cardholder verification.

2. In the response from the card to the terminal CTQ, CDCVM performed (byte 2, bit 8) is changed from "0" to "1". This tells the terminal that cardholder verification has already been performed on the device. Setting this bit to "1" enables the terminal to carry out its risk analysis and to determine which limit to impose. Because there is no PIN for consumer devices, the terminal cannot invoke a request for the PIN during risk management. In fact, CDCVM is always set to “1” when the CTQ is issued by a consumer device. This is because the consumer device determines which action should be taken in device to confirm CDCVM; should the screen be active or unlocked. After this, the terminal sends the transaction to the acquirer, card networks and issuer.

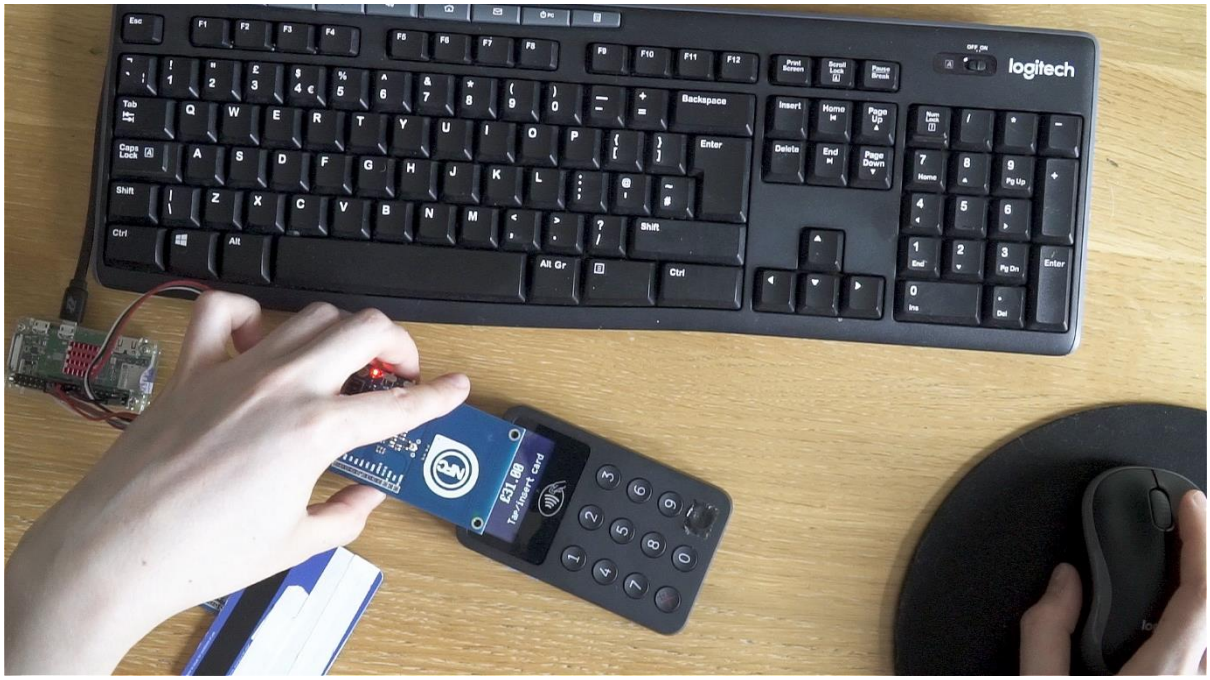

Figure 14. A contactless transaction of £31.00

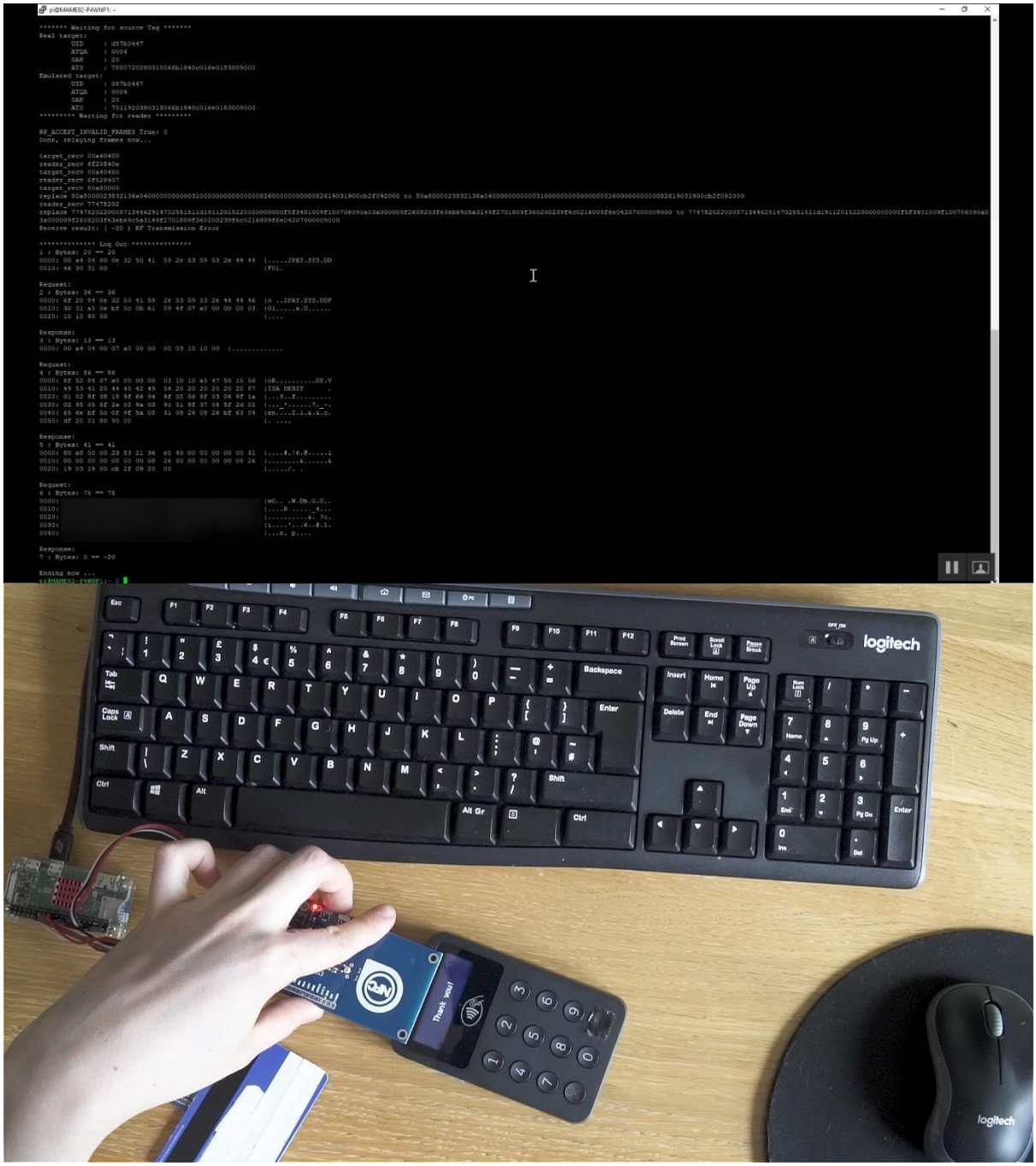

Figure 15. "CVM required" and "CDCVM performed" are substituted using the NFC proxy. Payment of £31.00 is successfully made without cardholder verification.

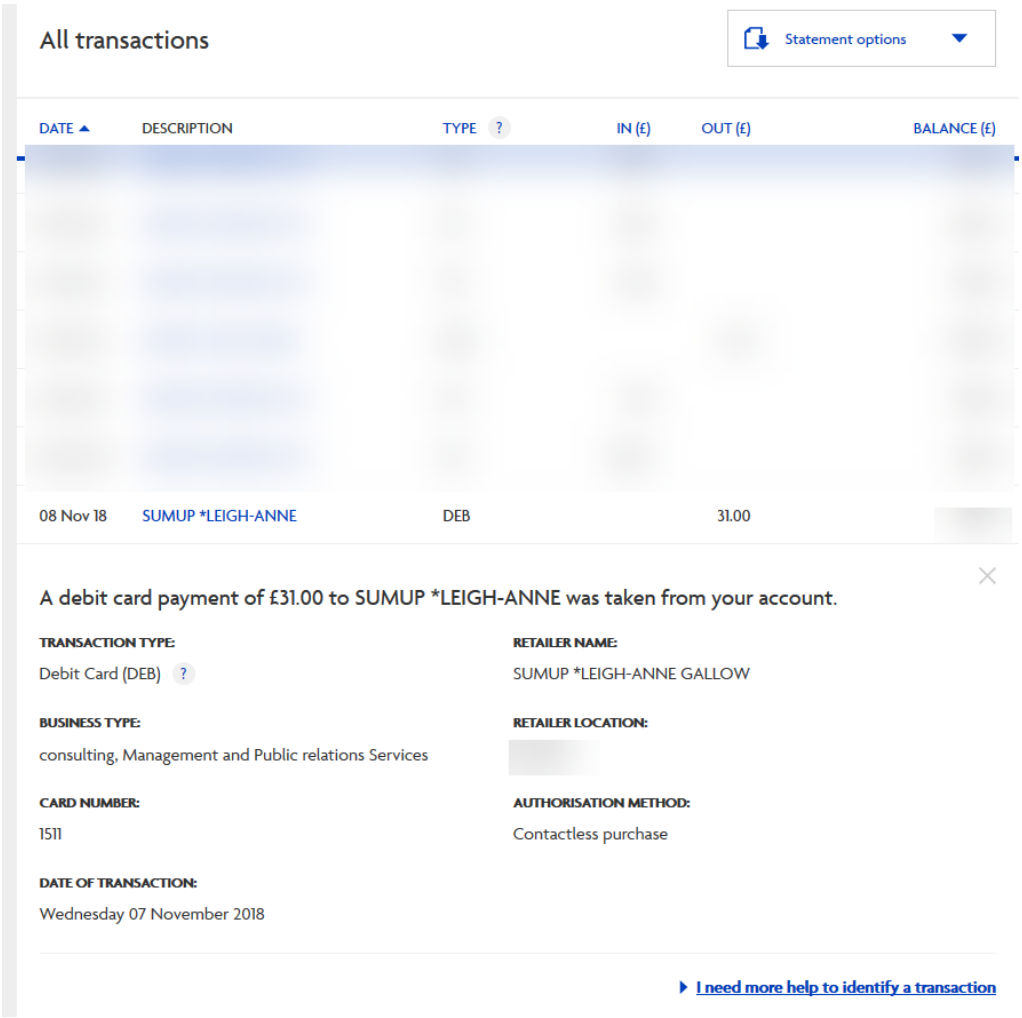

Figure 16. A contactless transaction exceeding the card limit

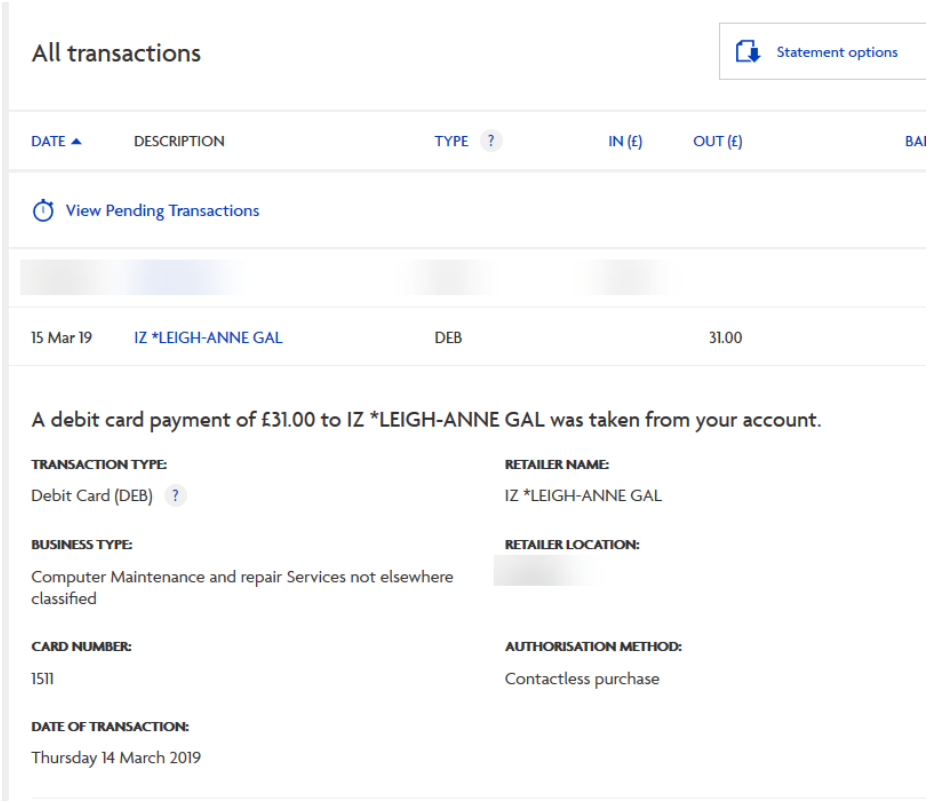

Figure 17. Another contactless transaction exceeding the card limit

Figures 16 and 17 show contactless transactions made for amounts above the card limit. Both transactions were successfully debited from the cardholder account. We are also able to make contactless transactions without cardholder verification for much larger values. In this example that follows, we were able to make a payment of £100.

Table of results: Exceeding contactless payment limits by circumventing CDCVM

Of the twelve Visa cards we tested, ten permitted us to bypass contactless limits using the method described above.

Card #	Type of Issuer	Region	Brand	Limit Bypass Successful
#1	Challenger bank	UK	Visa	Yes
#2	High-street bank	Asia	Visa	Yes
#3	High-street bank	Asia	Visa	Yes
#4	Challenger bank	USA	Visa	Yes
#5	High-street bank	USA	Visa	Yes
#6	Challenger bank	Asia	Visa	Yes
#7	Challenger bank	UK	Visa	Yes
#8	High-street bank	UK	Visa	No
#9	High-street bank	Asia	Visa	Yes
#10	High-street bank	Asia	Visa	Yes
#11	High-street bank	UK	Visa	No
#12	High-street bank	Asia	Visa	Yes

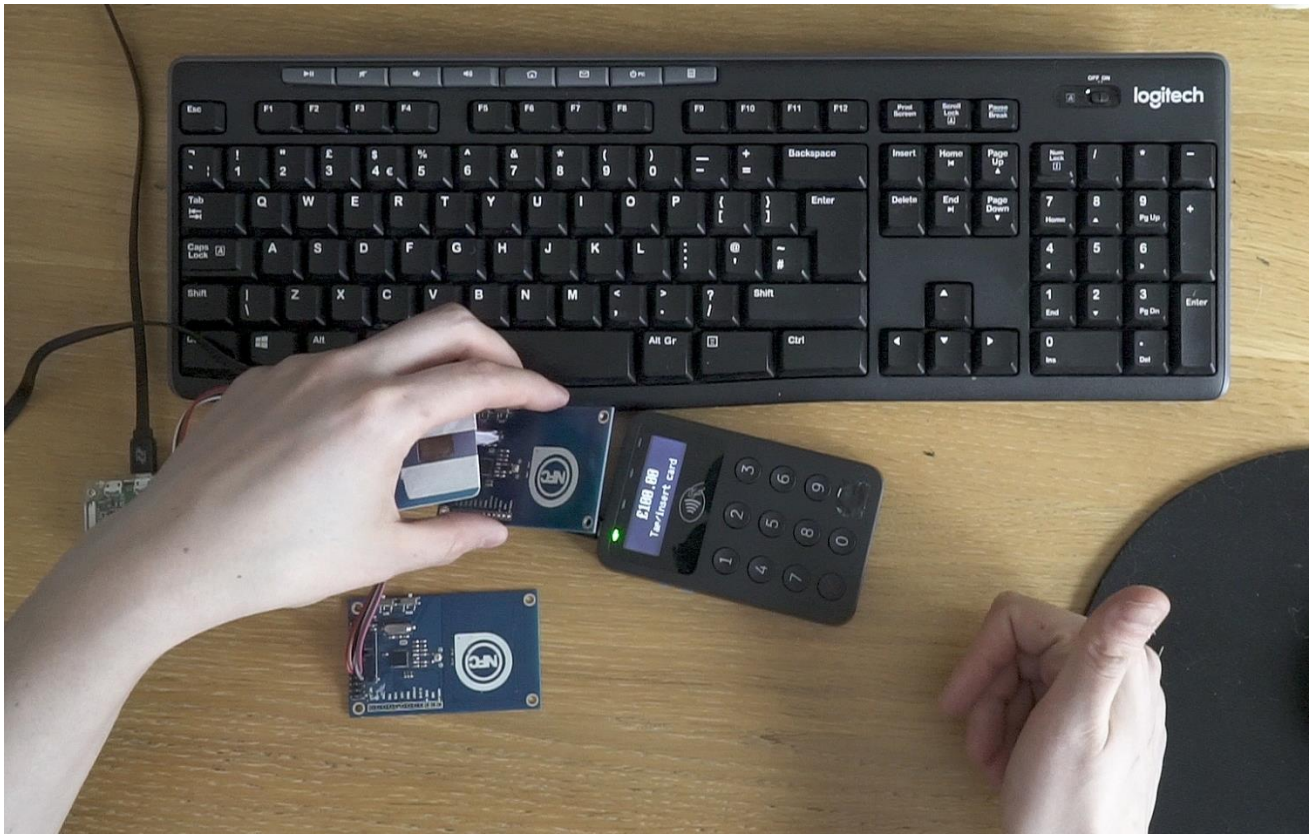

Figure 18. A contactless transaction of £100.00

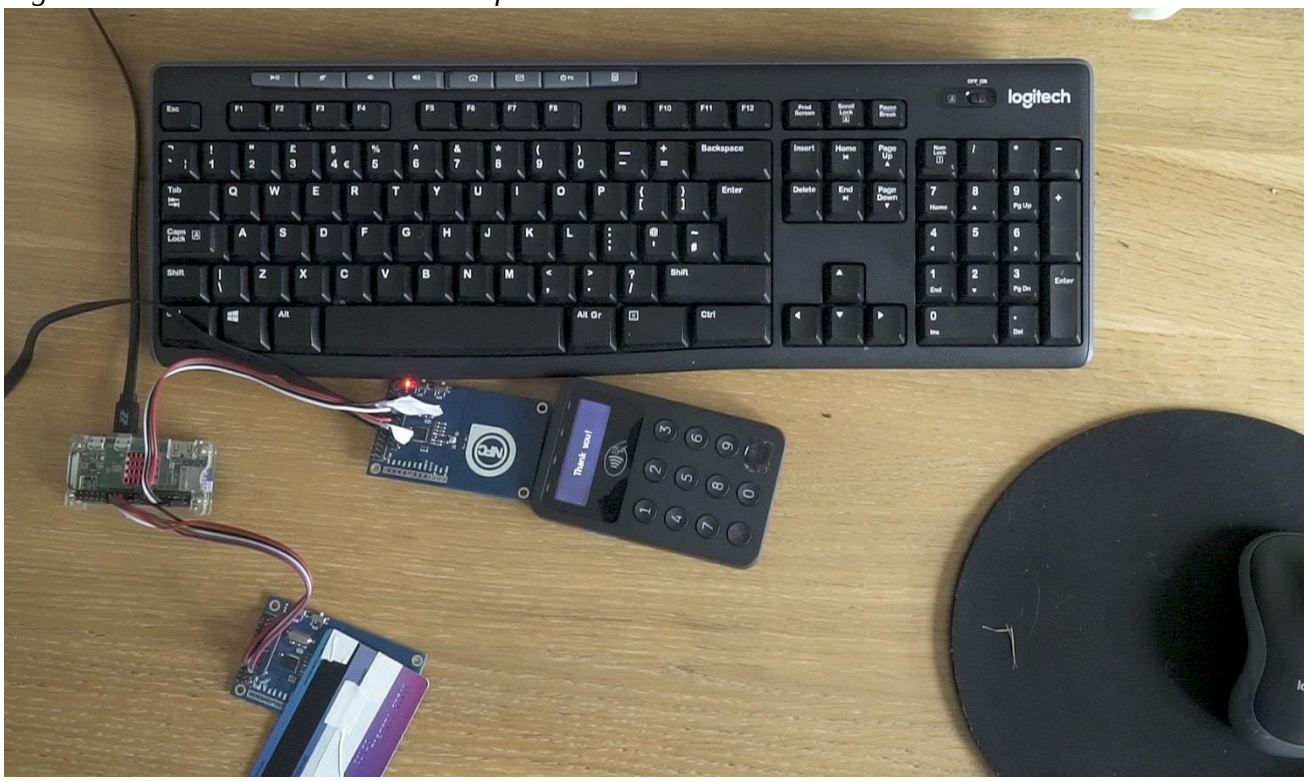

Figure 19. Payment of £100.00 successfully made without the need for cardholder verification

Bypassing CDCVM limits on Google Pay with Visa cards

For higher amounts (up to the terminal limit of $10,000/£5,500), Google Pay only requires CDCVM. For small transactions the Android device simply needs to have an active screen. This makes it possible to carry out electronic theft from a victim's wallet as long as NFC is active on the device and the screen is active. The screen can be activated by any number of methods. These methods can include requesting Bluetooth pairing, calling the phone, or pressing the volume keys. By proxying data between the Android device and the terminal, a transaction exceeding the CDCVM limit for Google Pay can go through.

This attack requires that CDCVM be supported by the terminal in the TTQ. Only one field needs to be modified.

1. In the TTQ sent to the device, the CVM required field (byte 2, bit 7) is changed from "1" to "0". This field indicates whether to require additional cardholder verification for the transaction. For transactions under the limit set on the terminal, this field is set to "0". By changing this bit to "0" the terminal tells the card it doesn't require cardholder verification.

Unlike with a card, the response from the device to the terminal CTQ, CDCVM performed (byte 2, bit 8) is always set to "1". This is because the consumer device determines which action should be taken in device to confirm CDCVM; should the screen be active or unlocked.

Figure 20. A contactless transaction of £31.00 with a consumer device. Screen must be active

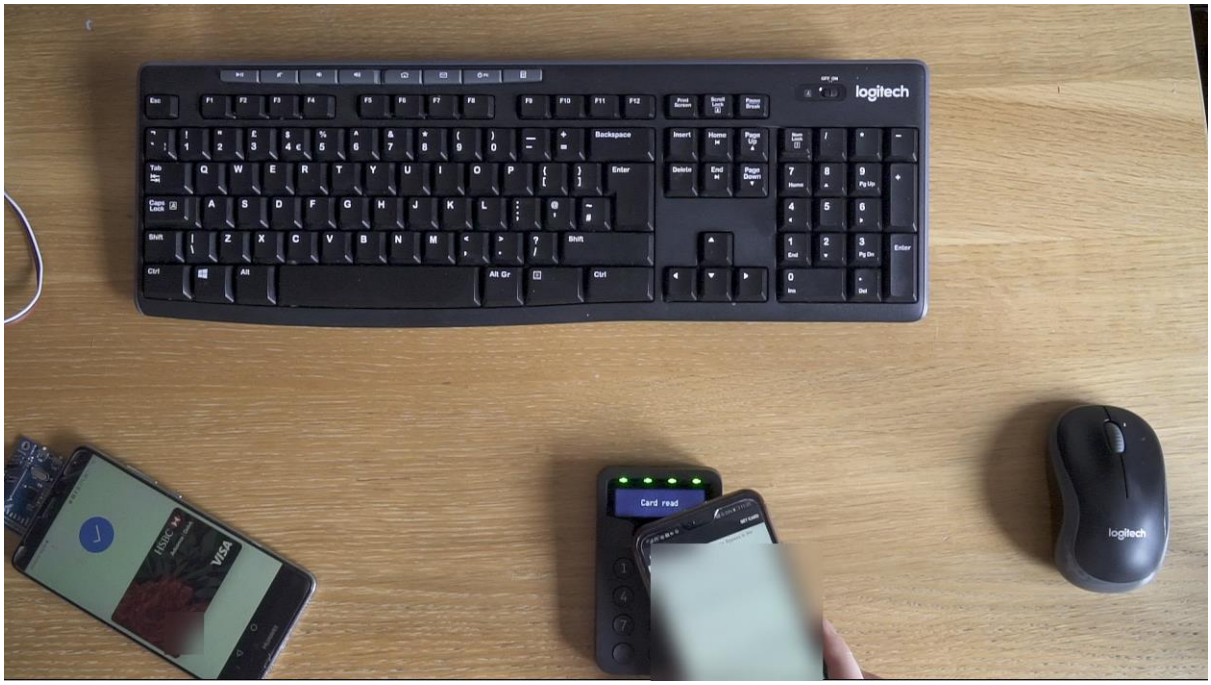

Figure 21. Card is successfully read from the proxy device

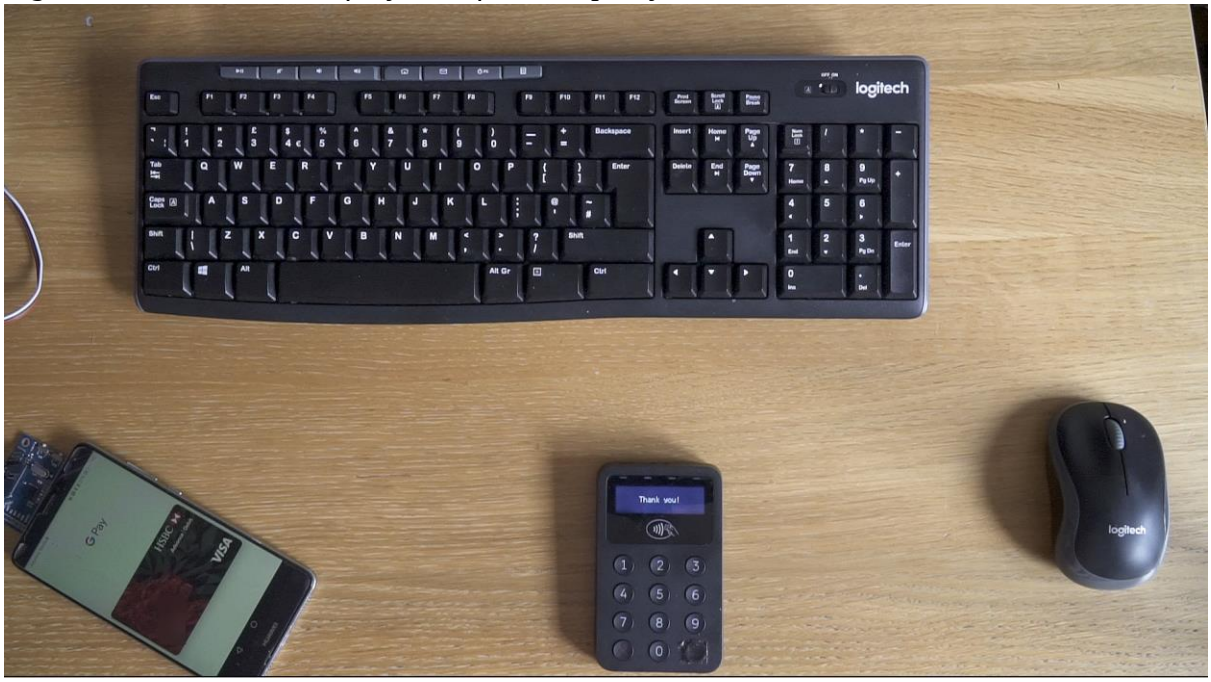

Figure 22. Payment of £31.00 successfully made without the need for the device to be unlocked

Cryptogram Versions

Our findings apply to all cryptogram versions of Visa qVSDC. The three versions of cryptogram are Cryptogram Version Number (CVN) 10, CVN17 and CVN18. Mastercard use the AIP to transmit CDCVM. As such it is not possible to bypass CVM limits on Mastercard cards. Visa includes the AIP in CVN10 and CVN18. However, unlike Mastercard, none of the elements contained within the CTQ and TTQ are included in the AIP and transmitted in the cryptogram.

Table D-1. Data input for TC, AAC, ARQC With CVN 10 ('0A')

Data Element
Amount, Authorized
Amount, Other
Terminal Country Code
Terminal Verification Results (TVR)
Transaction Currency Code
Transaction Date
Transaction Type
Unpredictable Number
Application Interchange Profile
ATC
Card Verification Results

Figure 23. Data elements included in Cryptogram Version Number 10. Data collected from Visa Contactless Payment Specification (VCPS)

C.2 Cryptogram Version Number 17('11')

Table C-1: Data Elements included in Cryptogram Version Number 17

Tag	Data Element
'9F02'	Amount, Authorized
'9F37'	Unpredictable Number
'9F36'	Application Transaction Counter (ATC)
'9F10'	Issuer Application Data (IAD) Byte 5

Figure 24. Data elements included in Cryptogram Version Number 17. Collected from VCPS

Table D-2. Data input for TC, AAC, ARQC With CVN 18
Data Element
Amount, Authorized
Amount, Other
Terminal Country Code
Terminal Verification Results (TVR)
Transaction Currency Code
Transaction Date
Transaction Type
Unpredictable Number
Application Interchange Profile
ATC
Issuer Application Data

Figure 25. Data elements included in Cryptogram Version Number 18. Collected from VCPS

Recommendations:
1. Visa to add CTQ and TTQ fields in the cryptogram creation/verification process.
2. Issuers need to check that CTQ field CDCVM performed (byte 2, bit 8) is set to "0" for physical cards. The reason: it's impossible for a physical card to perform CDCVM.

EMV contactless pre-play attack
Pre-play attacks against chip cards were first described by the University of Cambridge in 2011. [9] We have found that pre-play attacks are possible for contactless cards as well. Unlike the attack described by Bond et al., this vector does not require any physical interaction or special skimming devices inserted into an ATM or terminal. This attack is truly "contactless": a criminal can read information from a card without physically touching it. This also works for mobile wallets on Android devices. No physical device interaction is required.

Integrity of the transaction is ensured by the UN, which is unique and the ATC, which increments. But in practice, issuers don't decline transactions with an ATC value that is lower than the current ATC. This seeming laxness is to allow for transactions that may have been made offline. Nor are there any restrictions to prevent the terminal from sending the same UN to the card each time. As such, the EMV protocol is inherently flawed

This attack takes advantage of weak random number generation. We need the terminal to generate a predictable UN. The simplest way to do this is to compromise the terminal and to force it to use the same value for the UN each time. After patching the library on our terminal, the UN always equals "aa aa aa aa".

```
Response:
5 : Bytes: 41 == 41
0000: 80 a8 00 00 23 83 21 36  e0 40 00 00 00 00 00 31  |....#.!6.@.....1
0010: 00 00 00 00 00 00 00 08  26 00 00 00 00 00 08 26  |........&......&
0020: 18 11 01 00 aa aa aa aa  00                       |....;.w..
```

Figure 26. Response from the terminal to the card contains the UN value "aa aa aa aa".

All that is needed is to send the cryptogram to the acquirer for the amount specified. The transaction will be authorized. It can even be reused several times. Remember that each time a transaction is made, the ATC increments. This allows the issuer to check for transactions that are made out of order. If an ATC is the same as a previous value, this provides a strong indication of replay/pre-play fraud. Why can we reuse the same cryptogram many times? Because the ATC is not validated by many issuers. In practice, of the thirty one cards we tested, eighteen of them allowed us to make a transaction with an ATC value that is equal to or less than the previous ATC.

Table of results: EMV contactless pre-play

Card #	Type of Issuer	Region	Brand	Replay Successful
#1	Challenger bank	UK	Mastercard	Yes
#2	Challenger bank	UK	Mastercard	Yes
#3	Challenger bank	UK	Mastercard	Yes
#4	Challenger bank	UK	Mastercard	Yes
#5	Challenger bank	UK	Mastercard	No
#6	Challenger bank	UK	Mastercard	Yes
#7	Challenger bank	UK	Visa	Yes
#8	Challenger bank	Asia	Mastercard	No
#9	High-street bank	Asia	Visa	No
#10	High-street bank	Asia	Mastercard	No
#11	Challenger bank	Asia	Mastercard	No
#12	High-street bank	Asia	Visa	Yes
#13	High-street bank	Asia	Mastercard	Yes
#14	High-street bank	UK	Mastercard	No
#15	High-street bank	Asia	Mastercard	Yes
#16	High-street bank	EU	Visa	Yes
#17	High-street bank	EU	Mastercard	No
#18	High-street bank	EU	Mastercard	Yes
#19	Challenger bank	USA	Mastercard	No
#20	High-street bank	EU	Mastercard	No
#21	High-street bank	EU	Mastercard	No
#22	Challenger bank	UK	Visa	Yes
#23	Challenger bank	USA	Visa	Yes
#24	Challenger bank	EU	Mastercard	Yes
#25	High-street bank	USA	Visa	Yes
#26	High-street bank	USA	Mastercard	Yes
#27	Challenger bank	Asia	Visa	No
#28	Challenger bank	Asia	Mastercard	Yes
#29	Challenger bank	UK	Mastercard	No
#30	Challenger bank	UK	Visa	No
#31	High-street bank	UK	Visa	Yes

Potential gains for an attacker are limited by the currency and amount. However, this attack may be combined with the vector we have already described to bypass limits for Visa cards. This allows an attacker to pre-generate transaction values over the CDCVM limit for Visa cards. If the attacker has access to a compromised terminal, then CDCVM limits for MasterCard can also be circumvented by changing contactless limits in the terminal configuration file. Finally, this attack may also be combined with the technique we describe to read cards from Google Pay wallets.

Recommendations:
1. Issuers need to check the ATC value. If this value is the same as a previous ATC value, the transaction should be considered fraudulent.

PSD 2.0
First introduced in 2007 by the EU, PSD2 is the second iteration of the 'Payment Services Directive' (PSD). The first directive was introduced in 2007 to regulate payment services and payment service providers. PSD2 had to be transposed into local law by the 13th of January 2018. One of the security conditions of this directive is a requirement for Strong Customer Authentication (SCA). In UK legislation this is described in The Payment Services Regulations 2017 [10] as "authentication based on the use of two or more elements that are independent, in that the breach of one element does not compromise the reliability of any other element, and designed in such a way as to protect the confidentiality of the authentication data".

In practice this means that two-factor authentication needs to be applied to contactless payments "every now and again". UK banks implemented this requirement in September 2019 by applying cumulative limits to contactless payments. Two-factor authentication is required for every five transactions or £150 spent, whichever occurs sooner. At this point the operation is declined and the cardholder is asked to insert the card to complete a CHIP and PIN transaction.

Much of our research was completed prior to September 2019, as such our exploration of SCA is in its infancy. We can confirm that SCA limits do not apply to consumer devices. Therefore, CDCVM limits on Google Pay with Visa cards can be bypassed without being affected by PSD2. We will publish data in 2020 on the exploration of bypassing SCA.

Special thanks
Our work would not be possible without the help of Artem Ivachev. Artem reversed the implementation of EMV core on a Muira terminal. This allowed us to set a predictable UN on the terminal and to better understand implementation of EMV.

References

[1] ISO-14443 is described over four parts at https://www.iso.org/standard. Accessed 03/18/2019.

[2] Visa chip management technology for merchants. https://www.visa.com/chip/merchants/grow-your-business/payment-technologies/credit-card-chip/docs/kernal-management-guidelines.pdf. Accessed 03/20/2019.

[3] EMVco. https://www.emvco.com/. Accessed 03/16/2019.

[4] Jordi van den Breekel, Diego A. Ortiz-Yepes, Erik Poll, and Joeri de Ruiter. EMV in a Nutshell. https://www.cs.ru.nl/E.Poll/papers/EMVtechreport.pdf. Accessed 03/16/2019.

[5] EMV Book-A Architecture and General Requirements. Ahttps://www.emvco.com/wp-content/uploads/2017/05/Book_A_Architecture_and_General_Rqmts_v2_6_Final_20160422011856105.pdf. Accessed 03/22/2019.

[6] Complete list of EMV tags. https://www.eftlab.com/index.php/site-map/knowledge-base/211-emv-aid-rid-pix. Accessed 03/22/2019.

[7] EMV Book-B Entry Point Specifications. https://www.emvco.com/wp-content/uploads/2017/05/BookB_Entry_Point_Specification_v2_6_20160809023257319.pdf. Accessed 03/22/2019.

[8] EMV Book 4 – Other Interfaces (page 57). https://www.emvco.com/wp-content/uploads/2017/05/EMV_v4.3_Book_4_Other_Interfaces_20120607062305603.pdf. Accessed 04/03/2019

[9] Chip and Skim: cloning EMV cards with the pre-play attack. https://www.cl.cam.ac.uk/~osc22/docs/preplay_oakland14.pdf. Accessed 04/03/2019.

[10] Chip and Skim: cloning EMV cards with the pre-play attack. http://www.legislation.gov.uk/uksi/2017/752/made. Accessed 02/12/2019.

Additional reading

Thomas Bocek, Christian Killer, Christos Tsiaras, Burkhard Stiller. An NFC Relay Attack with Off-the-shelf Hardware and Software. Rémi Badonnel; Robert Koch; Aiko Pras; Martin Drašar; Burkhard Stiller. 10th IFIP International Conference on Autonomous Infrastructure, Management and Security (AIMS), June 2016, Munich, Germany. https://hal.inria.fr/hal-01632735/document. Accessed 03/18/2019.

Peter Fillmore. An Overview of Contactless Payment Cards. https://www.blackhat.com/docs/us-15/materials/us-15-Fillmore-Crash-Pay-How-To-Own-And-Clone-Contactless-Payment-Devices-wp.pdf. Accessed 03/15/2019.

Michael Roland, Josef Langer. Cloning Credit Cards: A combined pre-play and downgrade attack on EMV Contactless. https://www.usenix.org/system/files/conference/woot13/woot13-roland.pdf. Accessed 03/15/2019.