Cloning Credit Cards: A combined pre-play and downgrade attack on EMV Contactless

Michael Roland

13 August 2013 • USENIX WOOT'13 • Washington, D.C., USA

This work is part of the project "High Speed RFID" within the EU program "Regionale Wettbewerbsfähigkeit OÖ 2007–2013 (Regio 13)" funded by the European regional development fund (ERDF) and the Province of Upper Austria (Land Oberösterreich).

NFC Research Lab Hagenberg • www.nfc-research.at
A research group of the University of Applied Sciences Upper Austria

Outline

- Introduction
  - EMV Contactless
  - MasterCard PayPass
- Pre-play and Downgrade Attack
  - How it works
  - Implementation
  - Results & Improvements
  - Workarounds
- Demo
- Conclusion

EMV Contactless

www.nfc-research.at

- Standard for credit/debit cards with contactless interface
- Based on ISO 14443
  - Inductive coupling
  - 13.56 MHz
  - Compatible to NFC
- Combines different payment systems
  - AmEx ExpressPay: Kernel 4
  - JCB J/Speedy: Kernel 1 & 5*
  - MasterCard PayPass: Kernel 2
  - Visa payWave: Kernel 1 & 3

Source: Infineon

*) since version 2.3, March 2013

Kernel 2: MasterCard PayPass

- 2 modes
  - EMV mode
  - Mag-Stripe mode
- EMV mode
  - Secure chip uses EMV protocol over contactless (“Chip & PIN”)
- Mag-Stripe mode
  - Secure chip emulates magnetic stripe system
    - Compatibility mode to magnetic stripe back-end systems
- Support in contactless cards and terminals
  - Mag-Stripe mode: mandatory
  - EMV mode: optional (Europe/SEPA: mandatory)

Kernel 2: EMV Mode

- Card contains
  - Static card data (e.g. account number, expiry date, etc.)
  - Issuer’s digital signature over static data
  - Public keys of card and issuer
  - Secret key of card for digital signature
- Transaction
  - Terminal reads card data
  - Terminal authenticates card data
    - using issuer’s digital signature
  - Card authenticates payment transaction
    - by generating digital signature over transaction data (amount, date, etc.)

Kernel 2: Mag-Stripe Mode

- Card contains
  - Static card data (e.g. account number, expiry date, etc.)
    - Format comparable to that on magnetic stripe
  - Secret key for generating dynamic card verification codes
- Transaction
  - Terminal reads card data
  - Terminal sends unpredictable number (UN) to card
  - Card generates dynamic card verification code (CVC3) for UN
    - Authenticates card (but not the contents of a transaction)
    - Can be verified by card issuer during online authorization
- Main differences to EMV mode
  - No offline authentication of static card data
  - No authentication of payment transaction data

Goal of our Attack

- Skimming of contactless credit cards
  - We want to create a clone of a credit card
  - We want to use this clone to pay at POS terminals
- Target of our attack: Kernel 2's Mag-Stripe mode
  - Supported by all cards and terminals
  - Most data is static and can be skimmed
  - Terminal cannot check integrity of static data (no signature, etc.)
  - Problem: Dynamic card verification code (CVC3)
    - Used as a proof that terminal communicates with original card
    - **Existing attacks simply skip CVC3** and use skimmed data with merchants that do not require a CVC (e.g. Amazon)

CVC3
Dynamic Card Verification Code

- CVC3 = function(unpredictable number,
  transaction counter,
  secret card key/card data)
  - Secret card key:
    - Securely stored on card and cannot be skimmed
    - Protects against generation of CVC3s without original card
  - Transaction counter (ATC):
    - Stored on card and incremented for every transaction
    - Protects against re-use of CVC3s (re-play)
    - Protects against out-of-sequence use of CVC3s
  - Unpredictable number (UN):
    - Challenge generated by terminal
    - Protects against pre-generation of CVC3s (pre-play)

Pre-play Attack despite CVC3

- Pre-play protection relies on unpredictable number
  - If UN is predictable an attacker can pre-generate CVC3s!
- UN in EMV Contactless Kernel 2:
  - UN is a 4-byte field
    - 2^{32} (~4.3 billion) possible values
    - Pre-generation unfeasible
  - *BUT:* field is limited to BCD-encoding
    - 100 million possible values
    - ~43 times less than field limit
    - Pre-generation still unfeasible
  - *BUT:* number of BCD digits is defined by issuer (**& stored on card**)
    - Typical limit: 2-3 digits
    - 3 digits: 1000 possible values
    - ~4.3 million times less than field limit
    - **Pre-generation is feasible!!! → Pre-play attack**

Limitations

- ATC sequence
  - Any transaction with a higher ATC invalidates CVC3s for lower ATCs
  - Attack is only possible until original card is used for another transaction
- Mag-Stripe mode only
  - Attack only works for Mag-Stripe mode transactions
  - *BUT:* EMV mode transaction is performed if both, card and terminal, support EMV mode (e.g. in Europe)
  - Attack does not work if card and terminal support EMV mode

Downgrade Attack

www.nfc-research.at

- Limitation: Attack only works if either card or terminal support only Mag-Stripe mode
- Solution: Downgrade to Mag-Stripe mode
  - Make terminal believe it talks to a Mag-Stripe only card
  - Support for EMV mode is a flag in the Application Interchange Profile (one of the first data elements that the terminal reads from the card)
  - AIP has no integrity protection
  - **Change flag in AIP on card clone → Downgrade attack**

Mounting the Attack

www.nfc-research.at

- Collect data for pre-play and downgrade attack
  - Use app on NFC-enabled mobile phone (e.g. Galaxy Nexus)
  - Read static card data
  - Modify EMV mode flag
  - Pre-generate 1000 CVC3s
    - One code for each possible UN
    - At least one transaction can be performed
  - Performance
    - ~1000 codes/minute with Galaxy Nexus
    - *BUT:* not every card works well with every phone

- Create clone card
  - Use applet on Java Card
  - Applet contains data structures of credit card
    - Filled with static data from original card
  - Applet contains list of UN + ATC + CVC3 sets
    - Filled with pre-played CVC3s
    - Clone returns first set that matches given UN

Results

www.nfc-research.at

- Test
  - Read card data and pre-generate CVC3s using Galaxy Nexus
  - Copy data to clone card
  - Pay with clone card at POS
- Performed test using
  - 3 credit cards (from 2 different issuers)
  - 3 different terminals (all from same acquirer)
- **Payments were approved in all cases**

-C-U-S-T-O-M-E-R-
-R-E-C-E-I-P-T-

[redacted]

Terminal ID [redacted]83
TA No. 002056 RNo 0577

Card payment
MasterCard

EUR 0.01

PAN ############7993
EMV AID A0000000041010
VU no 158[redacted]
AIDPara 0100000002
Permission no. 391976
Date 23.07.13 15:15 Time

Approved

======================
AS-Proc-Code = 00 914
00
Capt.-Ref.= 0064
AID59: 781878
======================

PLEASE KEEP RECEIPT

Improvements

- Further reduce number of digits of UN
  - Number of digits is stored on card
  - Can be modified in clone card
  - Result: Faster pre-generation of CVC3s
  - *BUT:* Can be detected by issuer
    - Number of digits is sent to issuer during Mag-Stripe online authorization
    - 1 of 2 tested issuers detects **& rejects** such transactions
- Abuse terminal-specific weaknesses
  - Communication with 1 of 3 tested terminals can be forced to restart
    - Even after terminal sent unpredictable number to card
    - Upon restart terminal uses new unpredictable number
    - Works up to 6 times for one transaction
  - Clone card can restart transaction if no CVC3 is available for a given UN
    - Clone card can choose between 6 UNs
    - Card does not need to know a CVC3 for every UN
  - Result: Faster pre-generation of CVC3s

Workarounds

- Mag-Stripe mode vs. EMV mode
  - Issuer receives information if terminal supports EMV mode
  - Issuer receives information if transaction was performed using EMV mode or Mag-Stripe mode
  - Issuer knows if card supports EMV mode
  - Issuer **can detect** downgrade-case where EMV mode card is used at EMV mode terminal in Mag-Stripe mode
  - Our results show that issuers do **not** currently perform such checks

Workarounds (cont'd)

- Reduction of number of digits of UN
  - Number of digits used in transaction is sent to issuer
  - Issuer **can detect** if number of digits was tampered with
  - Our results show that some issuers have such checks in place
- Maximizing number of digits of UN
  - Adding one digit increases pre-generation time by factor 10
  - 4 digits: already 10 minutes → Pre-play infeasible!
  - Number of digits limited by Mag-Stripe back-end
  - Issuers should try to maximize size of UN

Video

http://youtu.be/VIAwxUs1ZFo

Conclusion

- Successful pre-play attack against Mag-Stripe mode
- Extended attack to EMV mode cards by downgrading to Mag-Stripe mode
- Protocols already contain countermeasures
- Many countermeasures are not implemented by issuers
- Reported our finding to MasterCard
  - Acknowledged vulnerabilities
  - Pointed out that their protocols and rules provide countermeasures
  - Left to the issuer to implement these measures

Michael Roland

Research Associate, NFC Research Lab Hagenberg
University of Applied Sciences Upper Austria

michael.roland (at) fh-hagenberg.at

This work is part of the project “High Speed RFID” within the EU program “Regionale Wettbewerbsfähigkeit OÖ 2007–2013 (Regio 13)” funded by the European regional development fund (ERDF) and the Province of Upper Austria (Land Oberösterreich).

NFC Research Lab Hagenberg • www.nfc-research.at
A research group of the University of Applied Sciences Upper Austria