Robbing the Bank with a Model Checker

David Basin
ETH Zurich

Nordic Fintech Week
Sept 2024

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Research on Tamarin & EMV — Collaborators

Tamarin Team

Simon Meier

Benedikt Schmidt

Cas Cremers

Ralf Sasse

Jannik Dreier

. . .

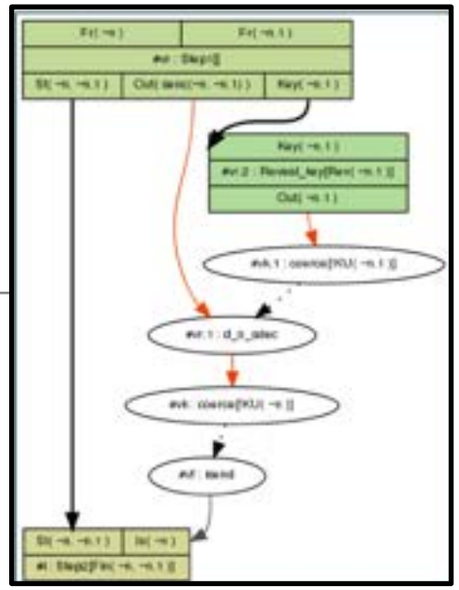
EMV

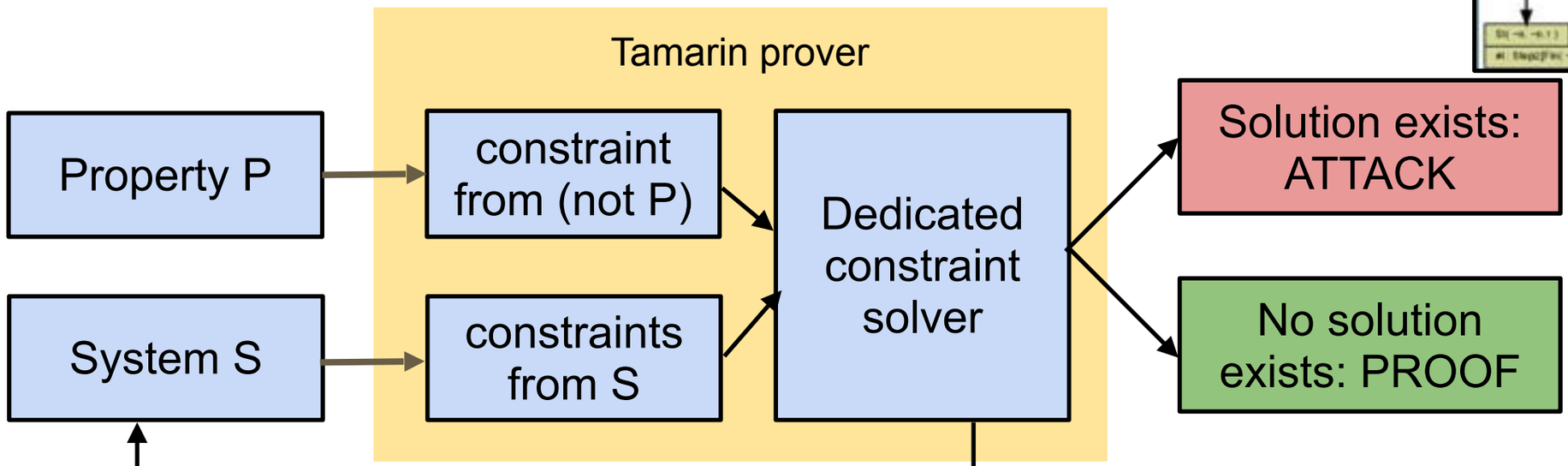
Ralf Sasse

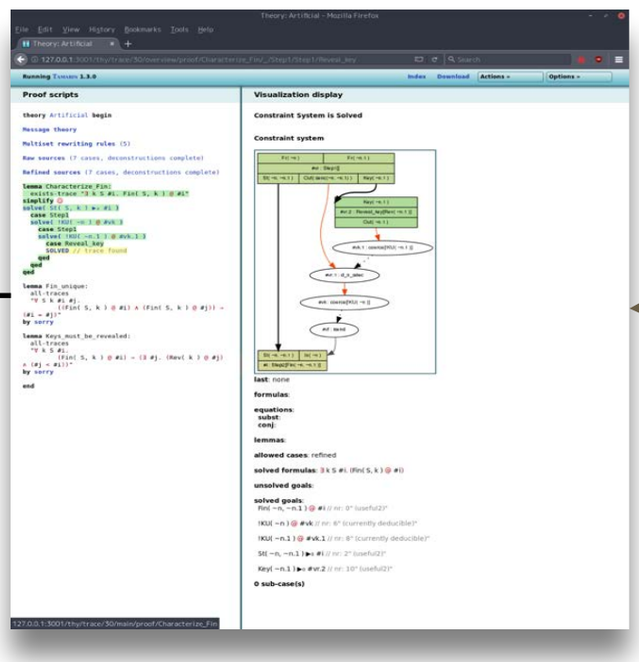
Jorge Toro Pozo

Xenia Hofmeier

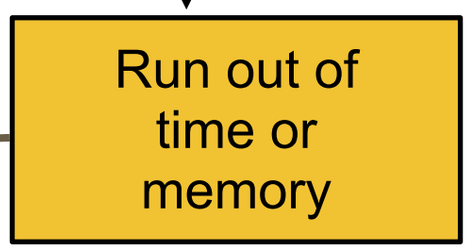
What is a Model Checker (Tamarin)?







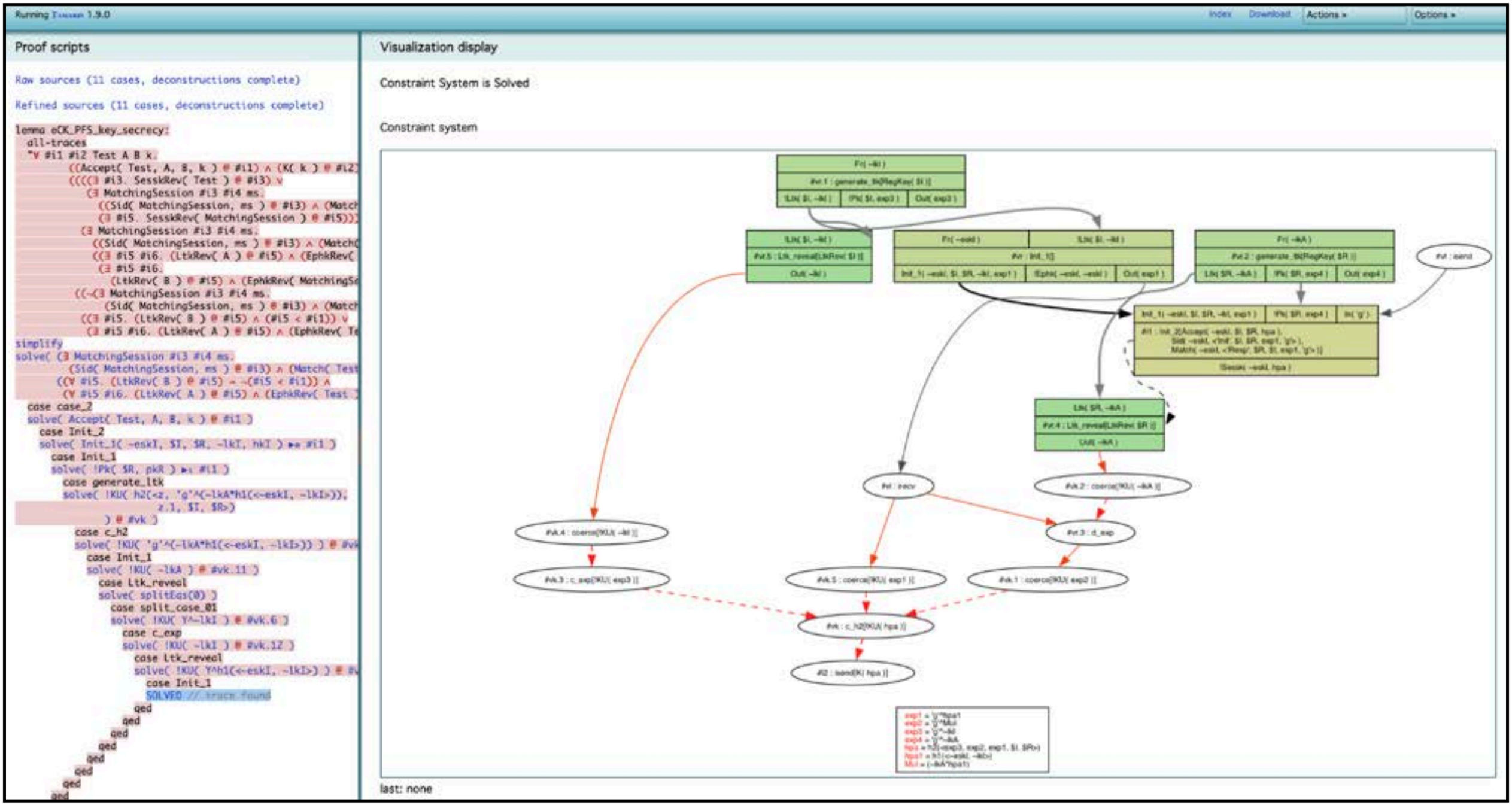
Provide **hints** for the prover (e.g. invariants)



Interactive mode
Inspect partial proof

Does Protocol Satisfy Property?

Or can the adversary attack it?



Prover combines backward search and constraint solving with user interaction, tactics, and lemmas

Real world Applications

A Formal Analysis of Apple's iMessage PQ3 Protocol

Felix Linker
Department of Computer Science, ETH Zurich

Ralf Sasse
Department of Computer Science, ETH Zurich

David Basin
Department of Computer Science, ETH Zurich

Abstract

We present the formal verification of Apple's iMessage PQ3, a highly performant, device-to-device messaging protocol offering strong security guarantees even against an adversary with quantum computing capabilities. PQ3 leverages Apple's identity services together with a custom, post-quantum secure initialization phase and afterwards it employs a double ratchet construction in the style of Signal, extended to provide post-quantum, post-compromise security.

intercept and to decrypt it in the future when quantum computers become sufficiently powerful [6].

In this paper, we present our formal analysis of Apple's advanced, widely deployed iMessage PQ3 Messaging Protocol, or PQ3 for short. PQ3 is used across all of Apple's devices for device-to-device messaging and underlies many other Apple services, e.g., iMessage, FaceTime, HomeKit, and HomePod hand-off. PQ3 is designed to be performant and to offer strong guarantees against powerful adversaries, including those who later possess quantum computers.

A Comprehensive Symbolic Analysis of TLS 1.3

Cas Cremers
University of Oxford, UK

Marko Horvat
MPI-SWS, Germany

Jonathan Hoyland
Royal Holloway, University of London, UK

Sam Scott
Royal Holloway, University of London, UK

Thyla van der Merwe
Royal Holloway, University of London, UK

ABSTRACT

The TLS protocol is intended to enable secure end-to-end communication over insecure networks, including the Internet. Unfortunately, this goal has been thwarted a number of times throughout the protocol's tumultuous lifetime, resulting in the need for a new version of the protocol, namely TLS 1.3. Over the past three years, in an unprecedented joint design effort with the academic community, the TLS Working Group has been working tirelessly to enhance the security of TLS.

Force (IETF) in the mid-nineties, the protocol has been incrementally modified and extended. In the case of TLS 1.2 and below, these modifications have taken place in a largely retroactive fashion; following the announcement of an attack [6, 7, 18, 20, 32, 43, 49], the TLS Working Group (WG) would either respond by releasing a protocol extension (A Request for Comments (RFC) intended to provide increased functionality and/or security enhancements) or by applying the appropriate "patch" to the next version of the protocol. For a more detailed analysis of the development and standardisation of TLS see [45]

A Formal Analysis of 5G Authentication

David Basin
Department of Computer Science
ETH Zurich
Switzerland
basin@inf.ethz.ch

Jannik Dreier
Universite de Lorraine
CNRS, Inria, LORIA
Nancy, France
jannik.dreier@loria.fr

Lucca Hirschi
Department of Computer Science
ETH Zurich
Switzerland
lucca.hirschi@inf.ethz.ch

Saša Radomirović
School of Science and Engineering
University of Dundee
UK
s.radomirovic@dundee.ac.uk

Ralf Sasse
Department of Computer Science
ETH Zurich
Switzerland
ralf.sasse@inf.ethz.ch

Vincent Stettler
Department of Computer Science
ETH Zurich
Switzerland
svincent@student.ethz.ch

ABSTRACT

Mobile communication networks connect much of the world's population. The security of users' calls, SMSs, and mobile data depends on the guarantees provided by the Authenticated Key Exchange protocols used. For the next-generation network (5G), the 3GPP group has standardized the 5G AKA protocol for this purpose.

We provide the first comprehensive formal model of a protocol from the AKA family: 5G AKA. We also extract precise requirements from the 3GPP standards defining 5G and we identify missing security goals. Using the security protocol verification tool Tamarin, we conduct a full, systematic, security evaluation of the model with respect to the 5G security goals. Our automated analysis identifies the minimal security assumptions required for each security goal and we find that some critical security goals are not met, except under additional assumptions missing from the standard. Finally, we make explicit recommendations with provably secure fixes for the attacks and weaknesses we found.

enable the subscribers and the *HNs* to mutually authenticate each other and to let the subscribers and the *SNs* establish a session key.

Next-Generation (5G). Since 2016, the 3GPP group has been standardizing the next generation of mobile communication (5G) with the aim of increasing network throughput and offering an ambitious infrastructure encompassing new use cases. The 5G standard will be deployed in two phases. The first phase (Release 15, June 2018) addresses the most critical requirements needed for commercial deployment and forms the basis for the first deployment. The second phase (Release 16, to be completed by the end of 2019) will address all remaining requirements.

In June 2018, the 3GPP published the *final* version `v15.1.0` of Release 15 of the Technical Specification (TS) defining the 5G security architecture and procedures [4]. The authentication in 5G Release 15 is based on new versions of the AKA protocols, notably the new *5G AKA* protocol, which enhances the AKA protocol currently used in 4G (EPS AKA) and which supposedly provides improved security

Getting Chip Card Payments Right*

vid Basin(✉)[1][0000−0003−2952−939X], Xenia Hofmeier[1][0009−0002−6909−80
Ralf Sasse[1][0000−0002−5632−6099], and Jorge Toro-Pozo[2]

[1] Department of Computer Science, ETH Zurich, Switzerland
{basin,xenia.hofmeier,ralf.sasse}@inf.ethz.ch
[2] SIX Digital Exchange, Switzerland
jorge.toro@sdx.com

Abstract. EMV is the international protocol standard for smart card payments and is used in billions of payment cards worldwide. Despite the standard's advertised security, various issues have been previously uncovered, deriving from logical flaws that are hard to spot in EMV's lengthy and complex specification. We have formalized various models of EMV in Tamarin, a symbolic model checker for cryptographic protocols. Tamarin was extremely effective in finding critical flaws, both known and new, and in many cases exploitable on actual cards. We report on these past problems as well as followup work where we verified the latest, improved version of the protocol, the EMV kernel C8. This work puts C8's correctness on a firm, formal basis, and clarifies which guarantees hold for C8 and under which assumptions. Overall our work supports the thesis that cryptographic protocol model checkers like Tamarin have

Use in Industry

EMV Standard

EMV is the global standard for smartcard payments: 13+ billion cards in use!

Founded by Europay, Mastercard, and Visa. Others have joined too

The standard claims to offer the highest security

EMV: Security and Convenience

Low-value purchases do not need a PIN

High-value purchases should be protected by a PIN

But they are not!

Hand image from https://pngtree.com/so/extend-a-finger
Devil image from https://pngtree.com/so/emoji-icons

Take Home Messages

1. **Developed first comprehensive model of EMV**
   Paper specification runs over 2,000 pages
   ⇝ directly formalized in Tamarin

2. **Found both known and new security issues**
   The PINs for your credit cards are useless!

3. **We proposed and machine-checked fixes (disclosed to relevant vendors)**
   Multiple iterations of (2) and (3)!

4. **Verified new C8 kernel with EMV partner**

5. **Experience supports general hypothesis:**
   Don't trust, verify!

Details described on the web at emvrace.github.io and FM 2024 paper.
Attack/disclosure timeline and other papers listed at end

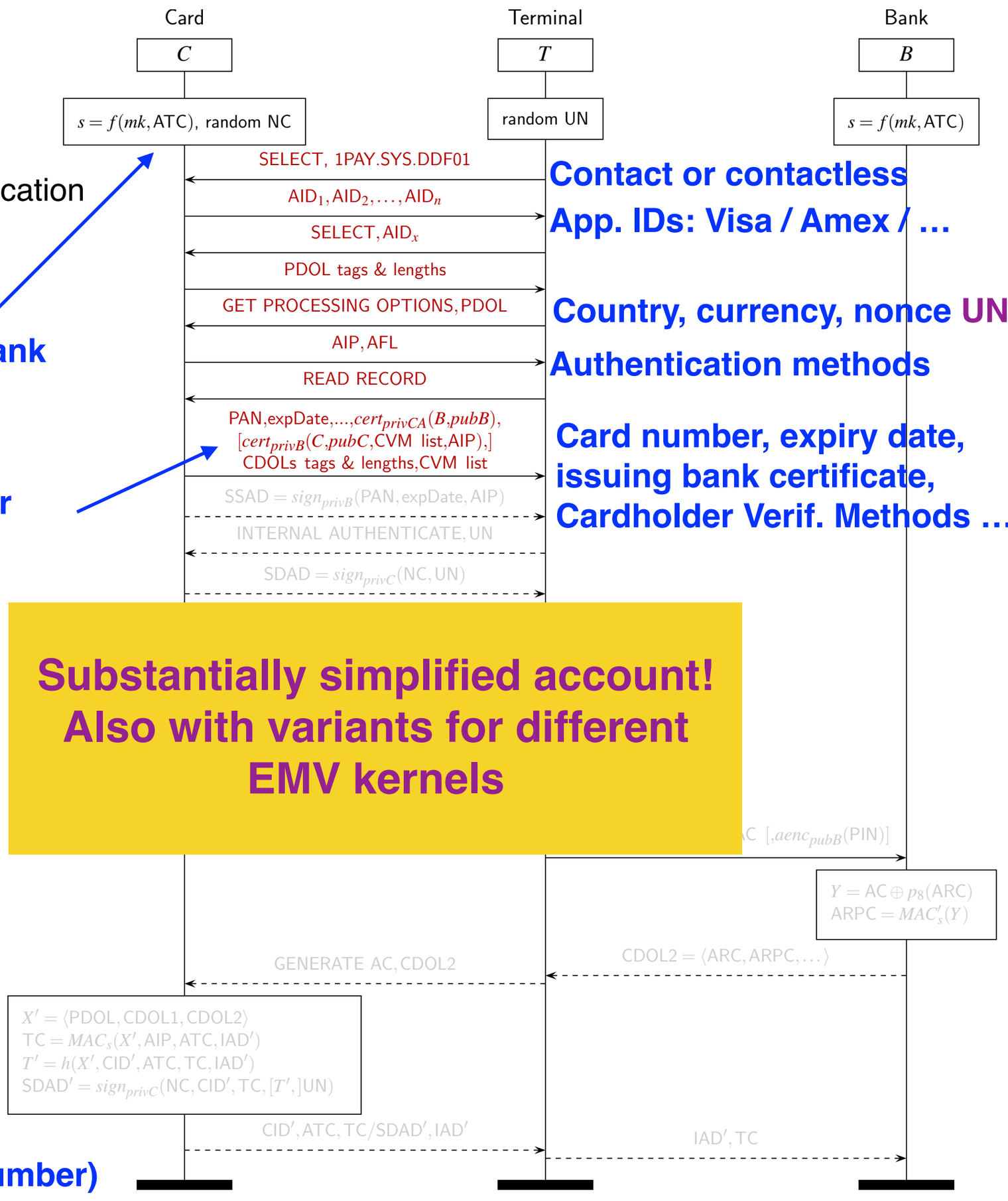
EMV Protocol

1. **Initialization:** card & terminal agree on application used for transaction & exchange static data.

2,000+ pages

mk: symmetric master key shared between card and bank
ATC: transaction counter
result's used for MACs

Uses PKI with certificates for CAs, Banks & Cards (but not Terminals)



Contact or contactless
App. IDs: Visa / Amex / …

Country, currency, nonce UN

Authentication methods

Card number, expiry date, issuing bank certificate, Cardholder Verif. Methods …

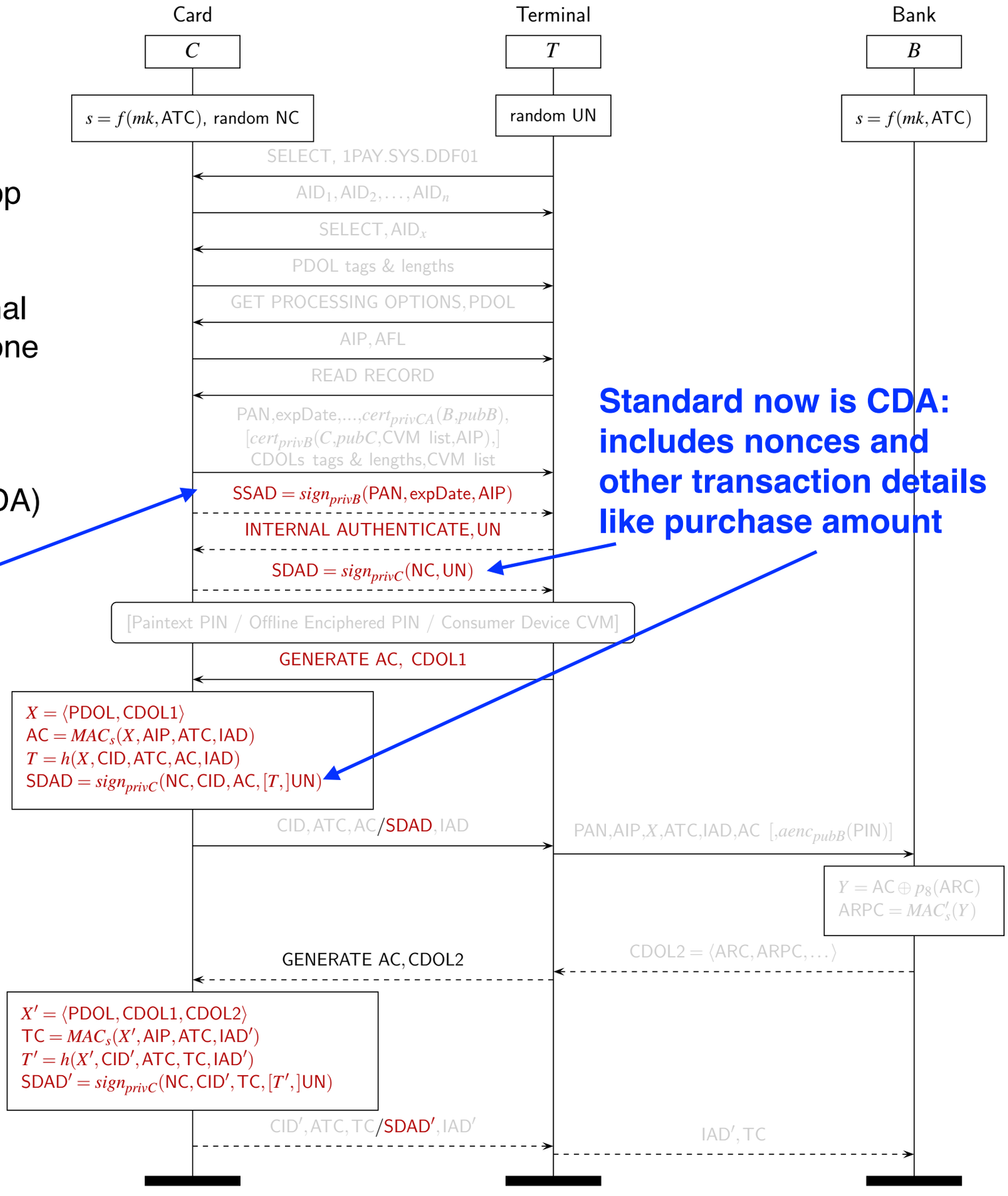
Acronym Zoo:
PDOL/CDOL: Data Object Lists
AID: Application Identifiers
PAN: Primary Account Number (Card number)
CVM: Cardholder Verification Methods
…

EMV Protocol

1. **Initialization:** card and terminal agree on app used for transaction & exchange static data.

2. **Offline Data Authentication (ODA):** terminal performs PKI-based **card validation** using one of three methods:
   - Static Data Authentication (SDA)
   - Dynamic Data Authentication (DDA)
   - Combined Dynamic Data Authentication (CDA)

Static data like card number and exp. date signed earlier by bank and stored on card. Legacy status.



Standard now is CDA: includes nonces and other transaction details like purchase amount

Acronym Zoo:

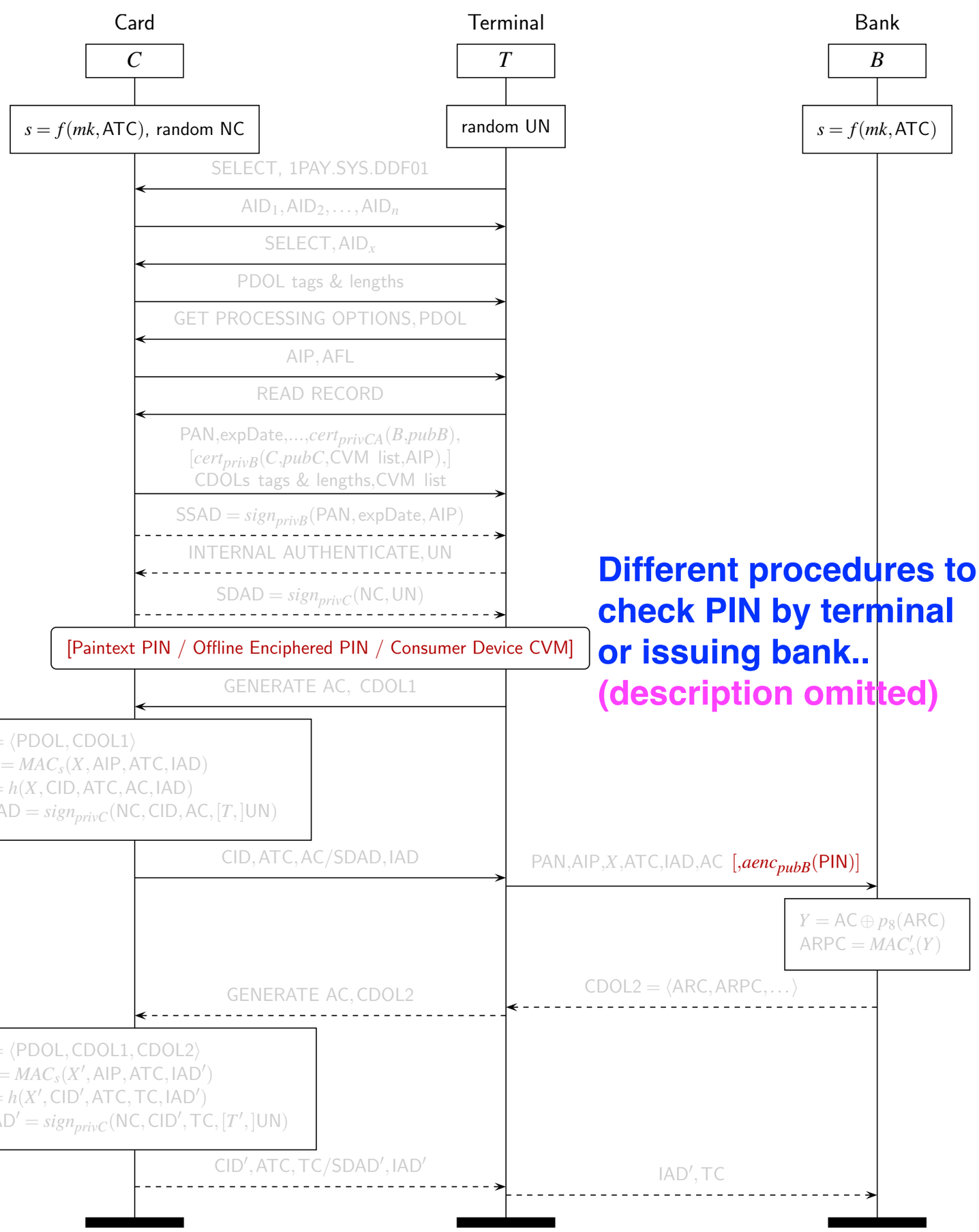
SSAD: Signed Static Authentication Data
SDAD: Signed Dynamic Authentication Data
AC: Authentication Cryptogram

EMV Protocol

1. **Initialization:** card and terminal agree on app used for transaction & exchange static data.

2. **Offline Data Authentication(ODA):** terminal performs PKI-based card validation using one of three methods:
   - Static Data Authentication (SDA)
   - Dynamic Data Authentication (DDA)
   - Combined Dynamic Data Authentication (CDA)

3. **Cardholder Verification:** terminal determines if person presenting card is legitimate cardholder using a Cardholder Verification Methods (CVM):
   - Signature / No PIN / No CVM
   - Plaintext PIN (terminal sends PIN to card)
   - Offline Enciphered PIN (terminal encrypts PIN and sends to card)
   - Online PIN (PIN sent encrypted to issuing bank)
   - Customer Device CVM (mobile phone auth.)



Different procedures to check PIN by terminal or issuing bank.. (description omitted)

EMV Protocol

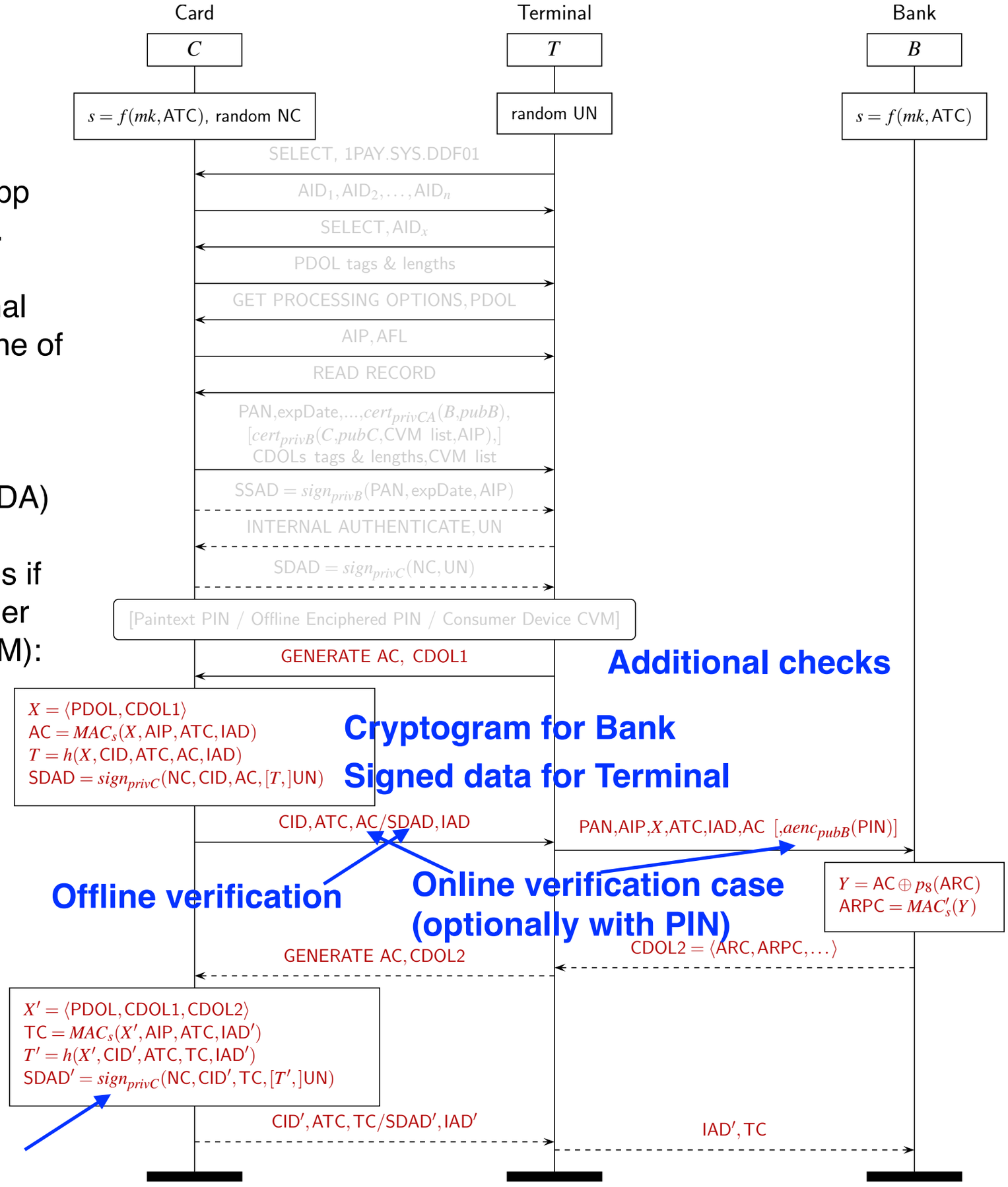
1. **Initialization:** card and terminal agree on app used for transaction & exchange static data.

2. **Offline Data Authentication(ODA):** terminal performs PKI-based card validation using one of three methods:
   - Static Data Authentication (SDA)
   - Dynamic Data Authentication (DDA)
   - Combined Dynamic Data Authentication (CDA)

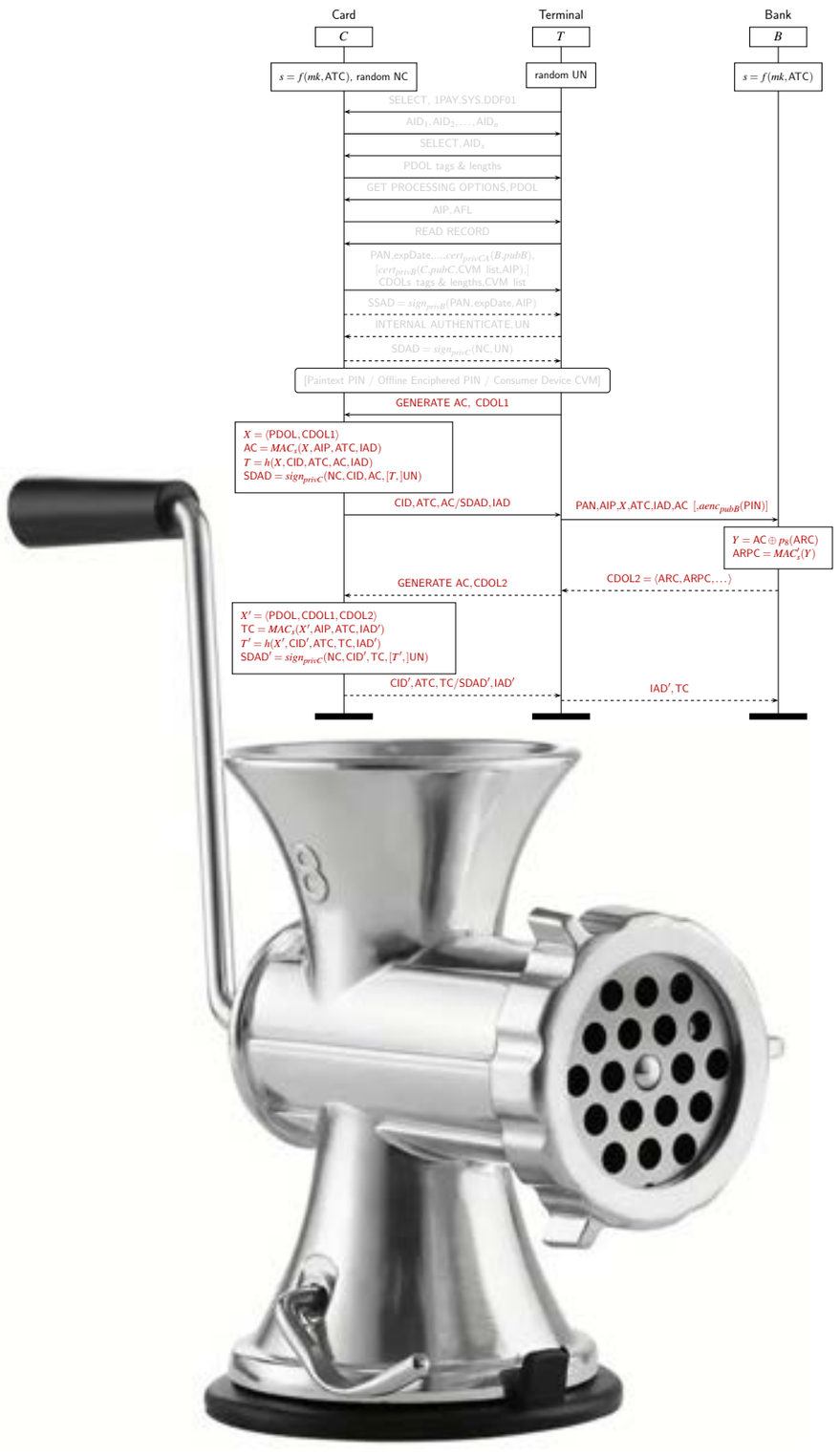
3. **Cardholder Verification:** terminal determines if person presenting card is legitimate cardholder using a Cardholder Verification Methods (CVM):
   - Signature / No PIN / No CVM
   - Plaintext PIN
   - Offline Enciphered PIN
   - Online PIN
   - Customer Device CVM

4. Transaction Authorization (TA): result is:
   - Declined offline
   - Accepted offline (typically low value)
   - Authorized online by issuer bank

This 2nd phase is for contact, where card authenticates bank and updates its state



From Protocols to Models




```
//============== Read Records ====================//
rule Terminal_Sends_ReadRecord:
    [ Terminal_Sent_GPO($Terminal, PDOL),
      In(<AIP, 'AFL'>) ]
  -->
    [ Out(<'READ_RECORD', 'AFL'>),
      Terminal_Sent_ReadRecord($Terminal, PDOL, AIP) ]

rule Card_Responds_To_ReadRecord_NotDDA:
    [ Card_Responded_To_GPO(~PAN, PDOL, ATC),
      !AIP(~PAN, AIP),
      !Records(~PAN, records),
      In(<'READ_RECORD', 'AFL'>) ]
  --[ NEq(fst(AIP), 'DDA') ]->
    [ Out(records),
      Card_Ready_For_Cryptogram(~PAN, PDOL, ATC) ]

rule Card_Responds_To_ReadRecord_DDA:
    [ Card_Responded_To_GPO(~PAN, PDOL, ATC),
      !Records(~PAN, records),
      !AIP(~PAN, <'DDA', furtherData>),
      In(<'READ_RECORD', 'AFL'>) ]
  -->
    [ Out(records),
      Card_Ready_For_DDA(~PAN, PDOL, ATC) ]
```

Protocol Modeled as 60 multiset rewriting rules

Main Properties Considered

1. **The bank accepts** transactions *t* accepted by the terminal

```
lemma bank_accepts:
  "All t #i.
    TerminalAccepts(t)@i
   ==>
    not (Ex #j. BankDeclines(t)@j) |
    Ex A #k. Honest(A)@i & Compromise(A)@k"
```

In Tamarin, protocol modeled as a labelled transition system giving rise to an infinite set of traces. Following trace would violate this property

…. BankDeclines(23581) … TerminalAccepts(23581) …

TerminalAccepts(t) iff Terminal satisfied with transaction.
BankDeclines(t) iff Bank receives authorization request with wrong cryptogram

Main Properties Considered

2. Transactions are **authenticated to the terminal** by the card and the bank

```
lemma auth_to_terminal: //injective agreement, r will be 'Card' or 'Bank'
  "All T P r t #i.
    Commit(T, P, <r, 'Terminal', t>)@i
   ==>
    ((Ex #j. Running(P, T, <r, 'Terminal', t>)@j & j < i) &
      not (Ex T2 P2 #i2. Commit(T2, P2, <r, 'Terminal', t>)@i2 & not(#i2 = #i))
    ) |
    Ex A #k. Honest(A)@i & Compromise(A)@k"
```

Whenever terminal *T* *Commits* to a transaction *t* with communication parter *P*, then either *P* in the role *r* ∈ {'card', 'Bank'} was previously *Running* the protocol with *T* and they agree on *t*, or an agent presumed honest was compromised. (Also there is a *unique Commit* for each pair of accepting transaction and accepting agent, so replay attacks are prevented. **)**

3. Transactions are **authenticated to the bank** by the card and the terminal. Property same as (2), but **'Terminal'** is now **'Bank'.**

Results for EMV Contact Protocol

Target Model	executable	bank accepts	auth. to terminal	auth. to bank
Contact_SDA_PlainPIN_Online	✓	×(2)	×(1,2)	×(1)
Contact_SDA_PlainPIN_Offline	✓	×(2)	×(1,2)	×(1)
Contact_SDA_OnlinePIN_Online	✓	×(2)	×(1,2)	×(1)
Contact_SDA_OnlinePIN_Offline	–	–	–	–
Contact_SDA_NoPIN_Online	✓	×(2)	×(1,2)	×(1)
Contact_SDA_NoPIN_Offline	✓	×(2)	×(1,2)	×(1)
Contact_SDA_EncPIN_Online	–	–	–	–
Contact_SDA_EncPIN_Offline	–	–	–	–
Contact_DDA_PlainPIN_Online	✓	×(2)	×(1,2)	×(1)
Contact_DDA_PlainPIN_Offline	✓	×(2)	×(1,2)	×(1)
Contact_DDA_OnlinePIN_Online	✓	×(2)	×(2)	✓
Contact_DDA_OnlinePIN_Offline	–	–	–	–
Contact_DDA_NoPIN_Online	✓	×(2)	×(2)	✓
Contact_DDA_NoPIN_Offline	✓	×(2)	×(2)	✓
Contact_DDA_EncPIN_Online	✓	×(2)	×(1,2)	×(1)
Contact_DDA_EncPIN_Offline	✓	×(2)	×(1,2)	×(1)
Contact_CDA_PlainPIN_Online	✓	✓	×(1)	×(1)
Contact_CDA_PlainPIN_Offline	✓	✓	×(1)	×(1)
Contact_CDA_OnlinePIN_Online	✓	✓	✓	✓
Contact_CDA_OnlinePIN_Offline	–	–	–	–
Contact_CDA_NoPIN_Online	✓	✓	✓	✓
Contact_CDA_NoPIN_Offline	✓	✓	✓	✓
Contact_CDA_EncPIN_Online	✓	✓	×(1)	×(1)
Contact_CDA_EncPIN_Offline	✓	✓	×(1)	×(1)

Legend:
✓: property verified ×: property falsified –: not applicable
(1): disagrees with card on CVM (2): disagrees with card on last AC
bold: satisfies all 4 properties

- Only transactions using the **CDA** authentication method and **Online PIN** or **No PIN** as CVM are **secure**
- Transactions using **Plaintext PIN** or **Offline Enciphered PIN** as CVM admit the PIN bypass of [Murdoch et al., S&P 2010]
- Transactions using the **SDA** or **DDA** authentication methods admit an attack where the terminal accepts them but the bank declines them
- We also found other issues related to secrecy
- In general, weaponizing these issues in practice is challenging as one would need control of the contact chip channel

Decomposed analysis: contact(less), and methods for data authentication and cardholder verification

Results for EMV Contact Protocol

Target Model	executable	bank accepts	auth. to terminal	auth. to bank
Contact_SDA_PlainPIN_Online	✓	×(2)	×(1,2)	×(1)
Contact_SDA_PlainPIN_Offline	✓	×(2)	×(1,2)	×(1)
Contact_SDA_OnlinePIN_Online	✓	×(2)	×(1,2)	×(1)
Contact_SDA_OnlinePIN_Offline	–	–	–	–
Contact_SDA_NoPIN_Online	✓	×(2)	×(1,2)	×(1)
Contact_SDA_NoPIN_Offline	✓	×(2)	×(1,2)	×(1)
Contact_SDA_EncPIN_Online	–	–	–	–
Contact_SDA_EncPIN_Offline	–	–	–	–
Contact_DDA_PlainPIN_Online	✓	×(2)	×(1,2)	×(1)
Contact_DDA_PlainPIN_Offline	✓	×(2)	×(1,2)	×(1)
Contact_DDA_OnlinePIN_Online	✓	×(2)	×(2)	✓
Contact_DDA_OnlinePIN_Offline	–	–	–	–
Contact_DDA_NoPIN_Online	✓	×(2)	×(2)	✓
Contact_DDA_NoPIN_Offline	✓	×(2)	×(2)	✓
Contact_DDA_EncPIN_Online	✓	×(2)	×(1,2)	×(1)
Contact_DDA_EncPIN_Offline	✓	×(2)	×(1,2)	×(1)
Contact_CDA_PlainPIN_Online	✓	✓	×(1)	×(1)
Contact_CDA_PlainPIN_Offline	✓	✓	×(1)	×(1)
Contact_CDA_OnlinePIN_Online	✓	✓	✓	✓
Contact_CDA_OnlinePIN_Offline	–	–	–	–
Contact_CDA_NoPIN_Online	✓	✓	✓	✓
Contact_CDA_NoPIN_Offline	✓	✓	✓	✓
Contact_CDA_EncPIN_Online	✓	✓	×(1)	×(1)
Contact_CDA_EncPIN_Offline	✓	✓	×(1)	×(1)

Legend:
✓: property verified ×: property falsified –: not applicable
(1): disagrees with card on CVM (2): disagrees with card on last AC
bold: satisfies all 4 properties

- Only transactions using the **CDA** authentication method and **Online PIN** or **No PIN** as CVM are **secure**
- Transactions using **Plaintext PIN** or **Offline Enciphered PIN** as CVM admit the PIN bypass of [Murdoch et al., S&P 2010]
- Transactions using the **SDA** or **DDA** authentication methods admit an attack where the terminal accepts them but the bank declines them
- We also found other issues related to secrecy
- In general, weaponizing these issues in practice is challenging as one would need control of the contact chip channel

Results for EMV Contact Protocol

Target Model	executable	bank accepts	auth. to terminal	auth. to bank
Contact_SDA_PlainPIN_Online	✓	✗(2)	✗(1,2)	✗(1)
Contact_SDA_PlainPIN_Offline	✓	✗(2)	✗(1,2)	✗(1)
Contact_SDA_OnlinePIN_Online	✓	✗(2)	✗(1,2)	✗(1)
Contact_SDA_OnlinePIN_Offline	–	–	–	–
Contact_SDA_NoPIN_Online	✓	✗(2)	✗(1,2)	✗(1)
Contact_SDA_NoPIN_Offline	✓	✗(2)	✗(1,2)	✗(1)
Contact_SDA_EncPIN_Online	–	–	–	–
Contact_SDA_EncPIN_Offline	–	–	–	–
Contact_DDA_PlainPIN_Online	✓	✗(2)	✗(1,2)	✗(1)
Contact_DDA_PlainPIN_Offline	✓	✗(2)	✗(1,2)	✗(1)
Contact_DDA_OnlinePIN_Online	✓	✗(2)	✗(2)	✓
Contact_DDA_OnlinePIN_Offline	–	–	–	–
Contact_DDA_NoPIN_Online	✓	✗(2)	✗(2)	✓
Contact_DDA_NoPIN_Offline	✓	✗(2)	✗(2)	✓
Contact_DDA_EncPIN_Online	✓	✗(2)	✗(1,2)	✗(1)
Contact_DDA_EncPIN_Offline	✓	✗(2)	✗(1,2)	✗(1)
Contact_CDA_PlainPIN_Online	✓	✓	✗(1)	✗(1)
Contact_CDA_PlainPIN_Offline	✓	✓	✗(1)	✗(1)
Contact_CDA_OnlinePIN_Online	✓	✓	✓	✓
Contact_CDA_OnlinePIN_Offline	–	–	–	–
Contact_CDA_NoPIN_Online	✓	✓	✓	✓
Contact_CDA_NoPIN_Offline	✓	✓	✓	✓
Contact_CDA_EncPIN_Online	✓	✓	✗(1)	✗(1)
Contact_CDA_EncPIN_Offline	✓	✓	✗(1)	✗(1)

Legend:
✓: property verified ✗: property falsified –: not applicable
(1): disagrees with card on CVM (2): disagrees with card on last AC
bold: satisfies all 4 properties

- Only transactions using the **CDA** authentication method and **Online PIN** or **No PIN** as CVM are **secure**
- Transactions using **Plaintext PIN** or **Offline Enciphered PIN** as CVM admit the PIN bypass of [Murdoch et al., S&P 2010]
- Transactions using the **SDA** or **DDA** authentication methods admit an attack where the terminal accepts them but the bank declines them
- We also found other issues related to secrecy
- In general, weaponizing these issues in practice is challenging as one would need control of the contact chip channel

Attack: fake the Card's response, which is not authenticated

Results for EMV Contact Protocol

Target Model	executable	bank accepts	auth. to terminal	auth. to bank
Contact_SDA_PlainPIN_Online	✓	×(2)	×(1,2)	×(1)
Contact_SDA_PlainPIN_Offline	✓	×(2)	×(1,2)	×(1)
Contact_SDA_OnlinePIN_Online	✓	×(2)	×(1,2)	×(1)
Contact_SDA_OnlinePIN_Offline	–	–	–	–
Contact_SDA_NoPIN_Online	✓	×(2)	×(1,2)	×(1)
Contact_SDA_NoPIN_Offline	✓	×(2)	×(1,2)	×(1)
Contact_SDA_EncPIN_Online	–	–	–	–
Contact_SDA_EncPIN_Offline	–	–	–	–
Contact_DDA_PlainPIN_Online	✓	×(2)	×(1,2)	×(1)
Contact_DDA_PlainPIN_Offline	✓	×(2)	×(1,2)	×(1)
Contact_DDA_OnlinePIN_Online	✓	×(2)	×(2)	✓
Contact_DDA_OnlinePIN_Offline	–	–	–	–
Contact_DDA_NoPIN_Online	✓	×(2)	×(2)	✓
Contact_DDA_NoPIN_Offline	✓	×(2)	×(2)	✓
Contact_DDA_EncPIN_Online	✓	×(2)	×(1,2)	×(1)
Contact_DDA_EncPIN_Offline	✓	×(2)	×(1,2)	×(1)
Contact_CDA_PlainPIN_Online	✓	✓	×(1)	×(1)
Contact_CDA_PlainPIN_Offline	✓	✓	×(1)	×(1)
Contact_CDA_OnlinePIN_Online	✓	✓	✓	✓
Contact_CDA_OnlinePIN_Offline	–	–	–	–
Contact_CDA_NoPIN_Online	✓	✓	✓	✓
Contact_CDA_NoPIN_Offline	✓	✓	✓	✓
Contact_CDA_EncPIN_Online	✓	✓	×(1)	×(1)
Contact_CDA_EncPIN_Offline	✓	✓	×(1)	×(1)

Legend:
✓: property verified ×: property falsified –: not applicable
(1): disagrees with card on CVM (2): disagrees with card on last AC
bold: satisfies all 4 properties

- Only transactions using the **CDA** authentication method and **Online PIN** or **No PIN** as CVM are **secure**
- Transactions using **Plaintext PIN** or **Offline Enciphered PIN** as CVM admit the PIN bypass of [Murdoch et al., S&P 2010]
- Transactions using the **SDA** or **DDA** authentication methods admit an attack where the terminal accepts them but the bank declines them
- We also found other issues related to secrecy
- In general, weaponizing these issues in practice is challenging as one would need control of the contact chip channel

Attack: transaction cryptogram modified, which goes undetected by terminal and is only later detected by bank

Results for EMV Contact Protocol

Target Model	executable	bank accepts	auth. to terminal	auth. to bank
Contact_SDA_PlainPIN_Online	✓	×(2)	×(1,2)	×(1)
Contact_SDA_PlainPIN_Offline	✓	×(2)	×(1,2)	×(1)
Contact_SDA_OnlinePIN_Online	✓	×(2)	×(1,2)	×(1)
Contact_SDA_OnlinePIN_Offline	–	–	–	–
Contact_SDA_NoPIN_Online	✓	×(2)	×(1,2)	×(1)
Contact_SDA_NoPIN_Offline	✓	×(2)	×(1,2)	×(1)
Contact_SDA_EncPIN_Online	–	–	–	–
Contact_SDA_EncPIN_Offline	–	–	–	–
Contact_DDA_PlainPIN_Online	✓	×(2)	×(1,2)	×(1)
Contact_DDA_PlainPIN_Offline	✓	×(2)	×(1,2)	×(1)
Contact_DDA_OnlinePIN_Online	✓	×(2)	×(2)	✓
Contact_DDA_OnlinePIN_Offline	–	–	–	–
Contact_DDA_NoPIN_Online	✓	×(2)	×(2)	✓
Contact_DDA_NoPIN_Offline	✓	×(2)	×(2)	✓
Contact_DDA_EncPIN_Online	✓	×(2)	×(1,2)	×(1)
Contact_DDA_EncPIN_Offline	✓	×(2)	×(1,2)	×(1)
Contact_CDA_PlainPIN_Online	✓	✓	×(1)	×(1)
Contact_CDA_PlainPIN_Offline	✓	✓	×(1)	×(1)
Contact_CDA_OnlinePIN_Online	✓	✓	✓	✓
Contact_CDA_OnlinePIN_Offline	–	–	–	–
Contact_CDA_NoPIN_Online	✓	✓	✓	✓
Contact_CDA_NoPIN_Offline	✓	✓	✓	✓
Contact_CDA_EncPIN_Online	✓	✓	×(1)	×(1)
Contact_CDA_EncPIN_Offline	✓	✓	×(1)	×(1)

Legend:
✓: property verified ×: property falsified –: not applicable
(1): disagrees with card on CVM (2): disagrees with card on last AC
bold: satisfies all 4 properties

- Only transactions using the **CDA** authentication method and **Online PIN** or **No PIN** as CVM are **secure**
- Transactions using **Plaintext PIN** or **Offline Enciphered PIN** as CVM admit the PIN bypass of [Murdoch et al., S&P 2010]
- Transactions using the **SDA** or **DDA** authentication methods admit an attack where the terminal accepts them but the bank declines them
- We also found other issues related to secrecy
- In general, weaponizing these issues in practice is challenging as one would need control of the contact chip channel

Attack: downgrade to plain PIN verification, and read PIN via MITM

Results for EMV Contact Protocol

Target Model	executable	bank accepts	auth. to terminal	auth. to bank
Contact_SDA_PlainPIN_Online	✓	×(2)	×(1,2)	×(1)
Contact_SDA_PlainPIN_Offline	✓	×(2)	×(1,2)	×(1)
Contact_SDA_OnlinePIN_Online	✓	×(2)	×(1,2)	×(1)
Contact_SDA_OnlinePIN_Offline	–	–	–	–
Contact_SDA_NoPIN_Online	✓	×(2)	×(1,2)	×(1)
Contact_SDA_NoPIN_Offline	✓	×(2)	×(1,2)	×(1)
Contact_SDA_EncPIN_Online	–	–	–	–
Contact_SDA_EncPIN_Offline	–	–	–	–
Contact_DDA_PlainPIN_Online	✓	×(2)	×(1,2)	×(1)
Contact_DDA_PlainPIN_Offline	✓	×(2)	×(1,2)	×(1)
Contact_DDA_OnlinePIN_Online	✓	×(2)	×(2)	✓
Contact_DDA_OnlinePIN_Offline	–	–	–	–
Contact_DDA_NoPIN_Online	✓	×(2)	×(2)	✓
Contact_DDA_NoPIN_Offline	✓	×(2)	×(2)	✓
Contact_DDA_EncPIN_Online	✓	×(2)	×(1,2)	×(1)
Contact_DDA_EncPIN_Offline	✓	×(2)	×(1,2)	×(1)
Contact_CDA_PlainPIN_Online	✓	✓	×(1)	×(1)
Contact_CDA_PlainPIN_Offline	✓	✓	×(1)	×(1)
Contact_CDA_OnlinePIN_Online	✓	✓	✓	✓
Contact_CDA_OnlinePIN_Offline	–	–	–	–
Contact_CDA_NoPIN_Online	✓	✓	✓	✓
Contact_CDA_NoPIN_Offline	✓	✓	✓	✓
Contact_CDA_EncPIN_Online	✓	✓	×(1)	×(1)
Contact_CDA_EncPIN_Offline	✓	✓	×(1)	×(1)

Legend:
✓: property verified ×: property falsified –: not applicable
(1): disagrees with card on CVM (2): disagrees with card on last AC
bold: satisfies all 4 properties

- Only transactions using the **CDA** authentication method and **Online PIN** or **No PIN** as CVM are **secure**
- Transactions using **Plaintext PIN** or **Offline Enciphered PIN** as CVM admit the PIN bypass of [Murdoch et al., S&P 2010]
- Transactions using the **SDA** or **DDA** authentication methods admit an attack where the terminal accepts them but the bank declines them
- We also found other issues related to secrecy
- In general, weaponizing these issues in practice is challenging as one would need control of the contact chip channel

Results for EMV Contactless Protocol

Target Model	exec.	bank accepts	auth. to terminal	auth. to bank
Visa_EMV_Low	✓	✓	×(1)	×(1)
Visa_EMV_High	✓	✓	×(1)	×(1)
Visa_DDA_Low	✓	×(2)	×(2)	✓
Visa_DDA_High	✓	✓	✓	✓
Mastercard_SDA_OnlinePIN_Low	✓	×(2)	×(2)	✓
Mastercard_SDA_OnlinePIN_High	✓	✓	✓	✓
Mastercard_SDA_NoPIN_Low	✓	×(2)	×(2)	✓
Mastercard_SDA_NoPIN_High	–(3)	–	–	–
Mastercard_DDA_OnlinePIN_Low	✓	×(2)	×(2)	✓
Mastercard_DDA_OnlinePIN_High	✓	✓	✓	✓
Mastercard_DDA_NoPIN_Low	✓	×(2)	×(2)	✓
Mastercard_DDA_NoPIN_High	–(3)	–	–	–
Mastercard_CDA_OnlinePIN_Low	✓	✓	✓	✓
Mastercard_CDA_OnlinePIN_High	✓	✓	✓	✓
Mastercard_CDA_NoPIN_Low	✓	✓	✓	✓
Mastercard_CDA_NoPIN_High	–(3)	–	–	–

Legend:
✓: property verified ×: property falsified –: not applicable
(1): disagrees with card on CVM (2): disagrees with card on AC
(3): high-value transactions without CVM are not completed contactless
bold: satisfies all 4 properties

- Most common Mastercard transactions are **secure**
- Most common Visa transactions are **not secure**

Results for EMV Contactless Protocol

Target Model	exec.	bank accepts	auth. to terminal	auth. to bank
Visa_EMV_Low	✓	✓	×(1)	×(1)
Visa_EMV_High	✓	✓	×(1)	×(1)
Visa_DDA_Low	✓	×(2)	×(2)	✓
Visa_DDA_High	✓	✓	✓	✓
Mastercard_SDA_OnlinePIN_Low	✓	×(2)	×(2)	✓
Mastercard_SDA_OnlinePIN_High	✓	✓	✓	✓
Mastercard_SDA_NoPIN_Low	✓	×(2)	×(2)	✓
Mastercard_SDA_NoPIN_High	–(3)	–	–	–
Mastercard_DDA_OnlinePIN_Low	✓	×(2)	×(2)	✓
Mastercard_DDA_OnlinePIN_High	✓	✓	✓	✓
Mastercard_DDA_NoPIN_Low	✓	×(2)	×(2)	✓
Mastercard_DDA_NoPIN_High	–(3)	–	–	–
Mastercard_CDA_OnlinePIN_Low	✓	✓	✓	✓
Mastercard_CDA_OnlinePIN_High	✓	✓	✓	✓
Mastercard_CDA_NoPIN_Low	✓	✓	✓	✓
Mastercard_CDA_NoPIN_High	–(3)	–	–	–

Legend:
✓: property verified ×: property falsified –: not applicable
(1): disagrees with card on CVM (2): disagrees with card on AC
(3): high-value transactions without CVM are not completed contactless
bold: satisfies all 4 properties

- Most common Mastercard transactions are **secure**
- Most common Visa transactions are **not secure**

Recall: CDA is what is commonly used in practice
(We return to this result for Mastercard later!)

Results for EMV Contactless Protocol

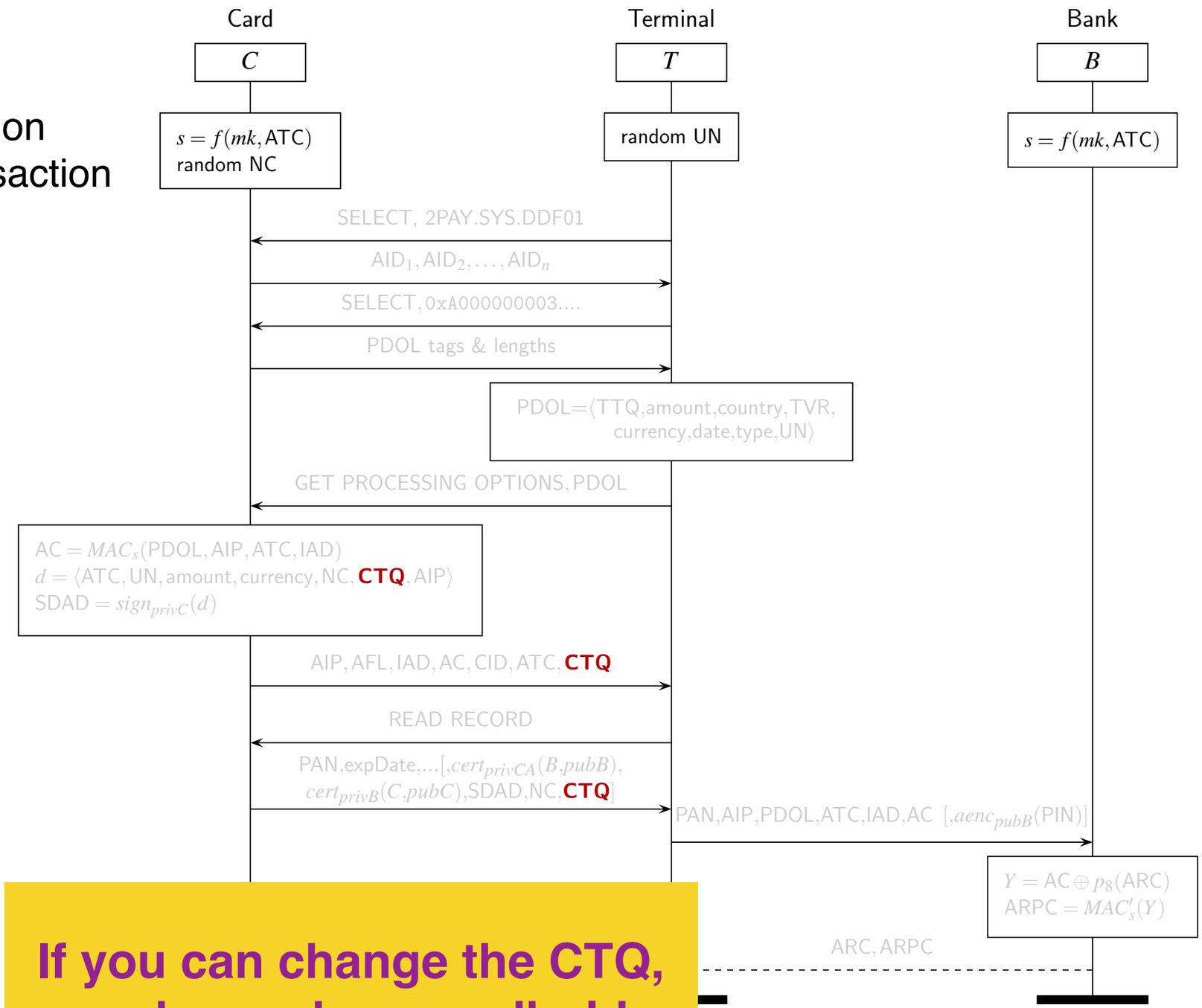
Target Model	exec.	bank accepts	auth. to terminal	auth. to bank
Visa_EMV_Low	✓	✓	×(1)	×(1)
Visa_EMV_High	✓	✓	×(1)	×(1)
Visa_DDA_Low	✓	×(2)	×(2)	✓
Visa_DDA_High	✓	✓	✓	✓
Mastercard_SDA_OnlinePIN_Low	✓	×(2)	×(2)	✓
Mastercard_SDA_OnlinePIN_High	✓	✓	✓	✓
Mastercard_SDA_NoPIN_Low	✓	×(2)	×(2)	✓
Mastercard_SDA_NoPIN_High	–(3)	–	–	–
Mastercard_DDA_OnlinePIN_Low	✓	×(2)	×(2)	✓
Mastercard_DDA_OnlinePIN_High	✓	✓	✓	✓
Mastercard_DDA_NoPIN_Low	✓	×(2)	×(2)	✓
Mastercard_DDA_NoPIN_High	–(3)	–	–	–
Mastercard_CDA_OnlinePIN_Low	✓	✓	✓	✓
Mastercard_CDA_OnlinePIN_High	✓	✓	✓	✓
Mastercard_CDA_NoPIN_Low	✓	✓	✓	✓
Mastercard_CDA_NoPIN_High	–(3)	–	–	–

Legend:
✓: property verified ×: property falsified –: not applicable
(1): disagrees with card on CVM (2): disagrees with card on AC
(3): high-value transactions without CVM are not completed contactless
bold: satisfies all 4 properties

- Most common Mastercard transactions are **secure**
- Most common Visa transactions are **not secure**

Problem with Visa Contactless

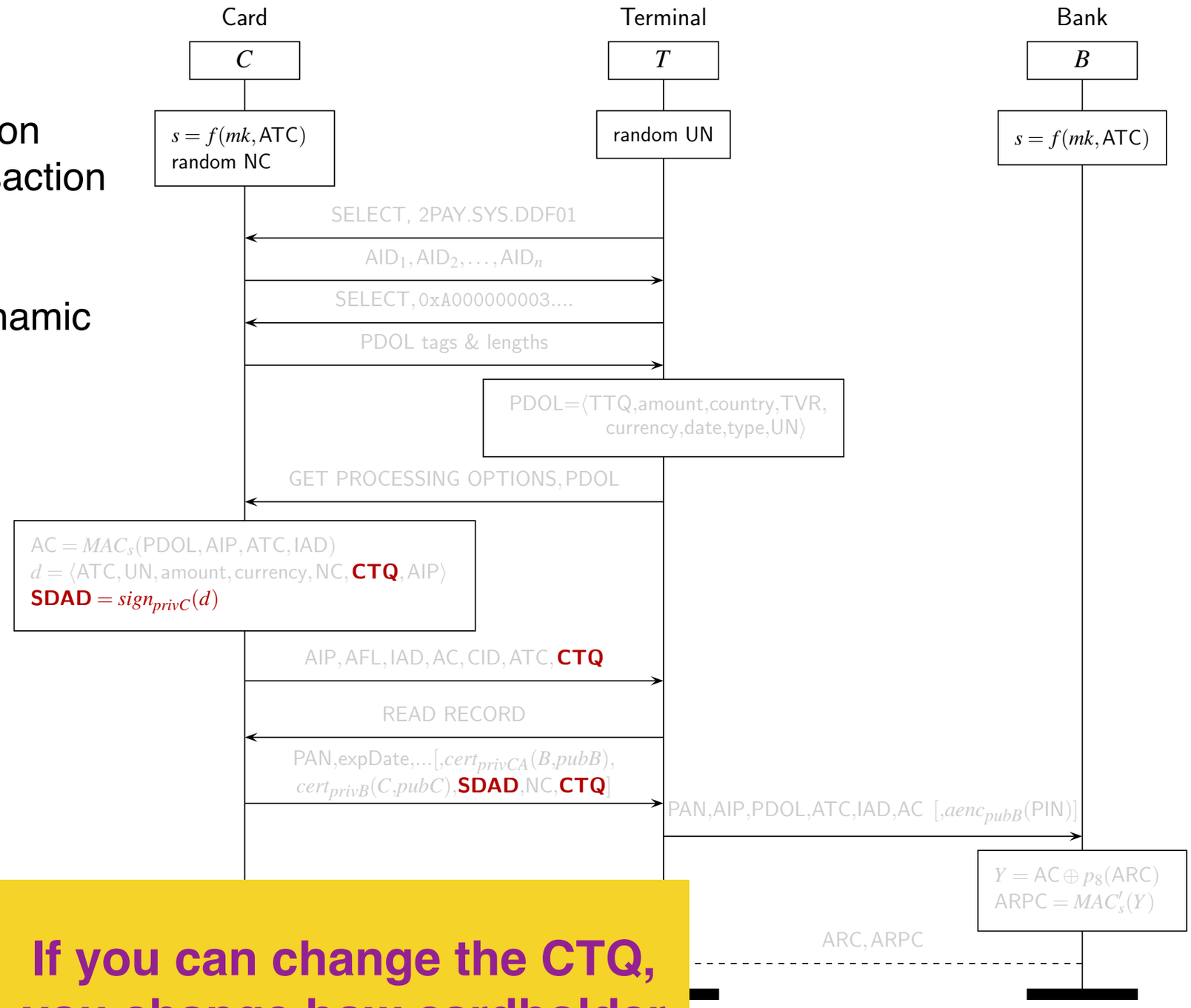
- Card's choice for Cardholder Verification Method (CVM) encoded in Card Transaction Qualifiers (CTQ)



If you can change the CTQ, you change how cardholder is (apparently) verified

Problem with Visa Contactless

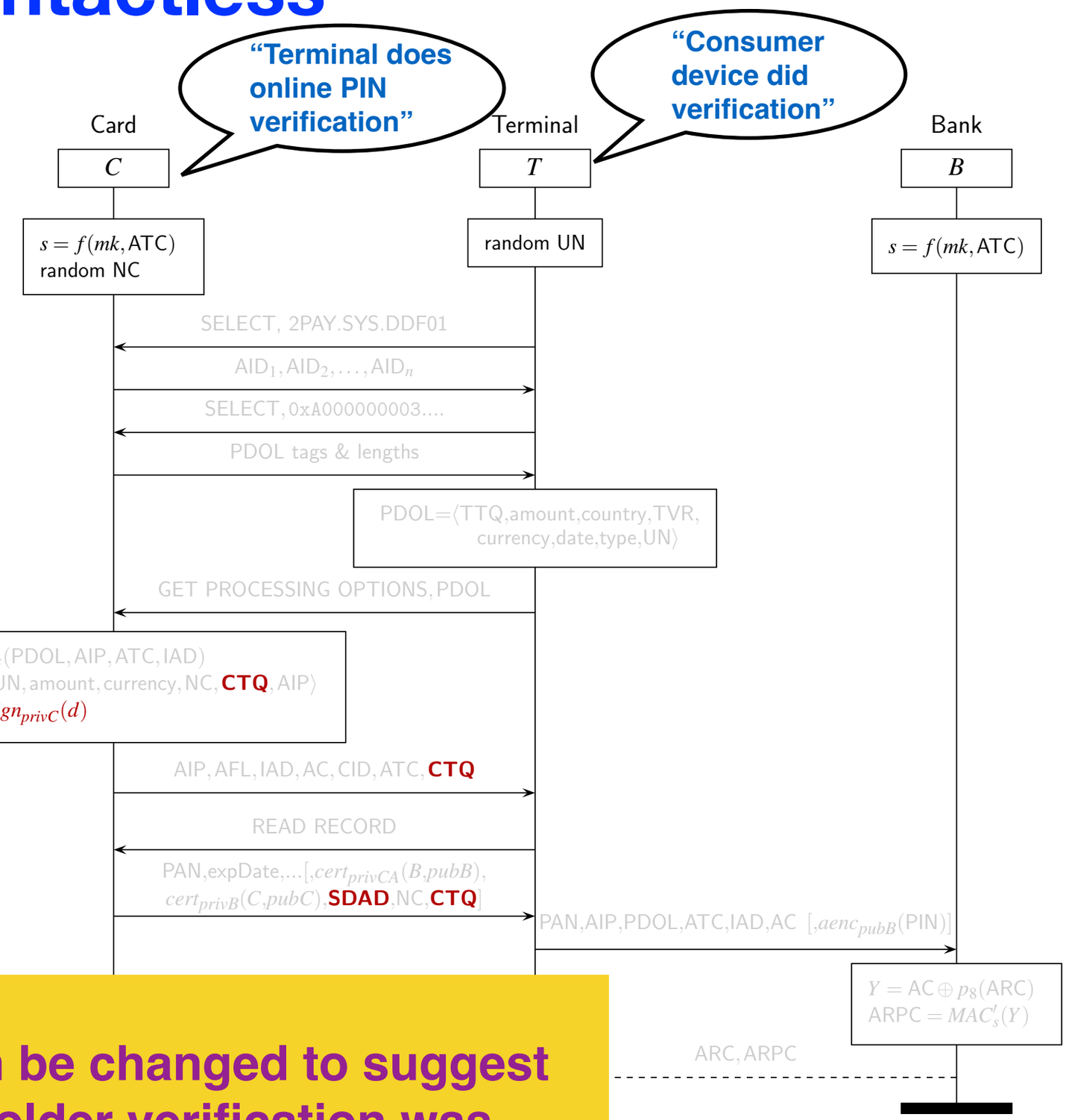
- Card's choice for Cardholder Verification Method (CVM) encoded in Card Transaction Qualifiers (CTQ)
- CTQ authenticated via the Signed Dynamic Authentication Data (SDAD)



If you can change the CTQ, you change how cardholder is (apparently) verified

Problem with Visa Contactless

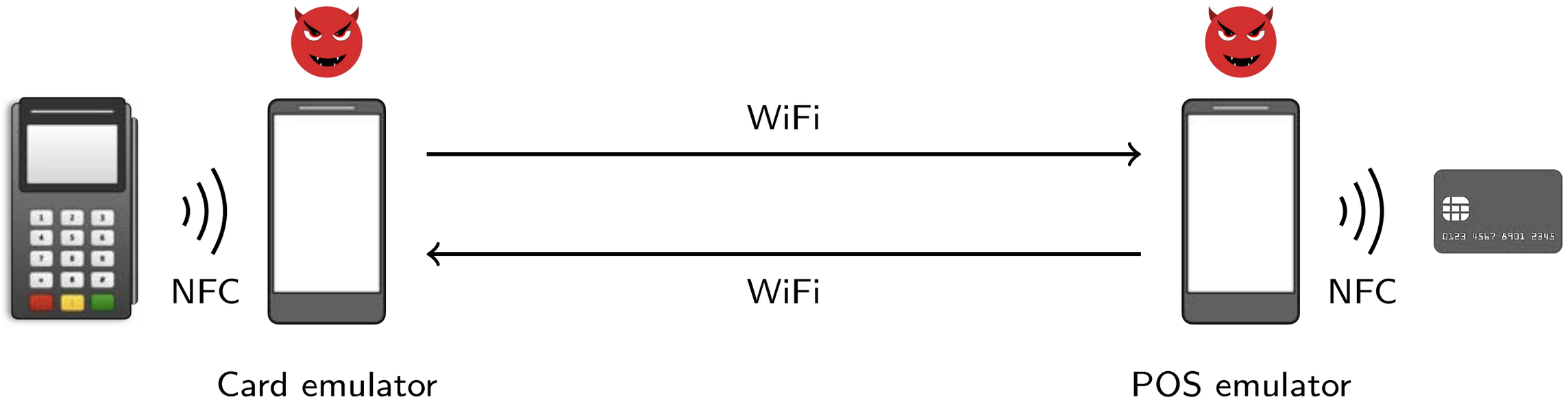
- Card's choice for Cardholder Verification Method (CVM) encoded in Card Transaction Qualifiers (CTQ)
- CTQ authenticated via the Signed Dynamic Authentication Data (SDAD)
- Most Visa transactions don't use the SDAD
  ⇒ CTQ and therefore **CVM can be modified**



CTQ can be changed to suggest cardholder verification was performed on the Consumer Device

Weaponizing PIN bypass Attack

Man-in-the-middle attack on top of a relay attack architecture



Weaponizing PIN Bypass Attack

Man-in-the-middle attack on top of a relay attack architecture

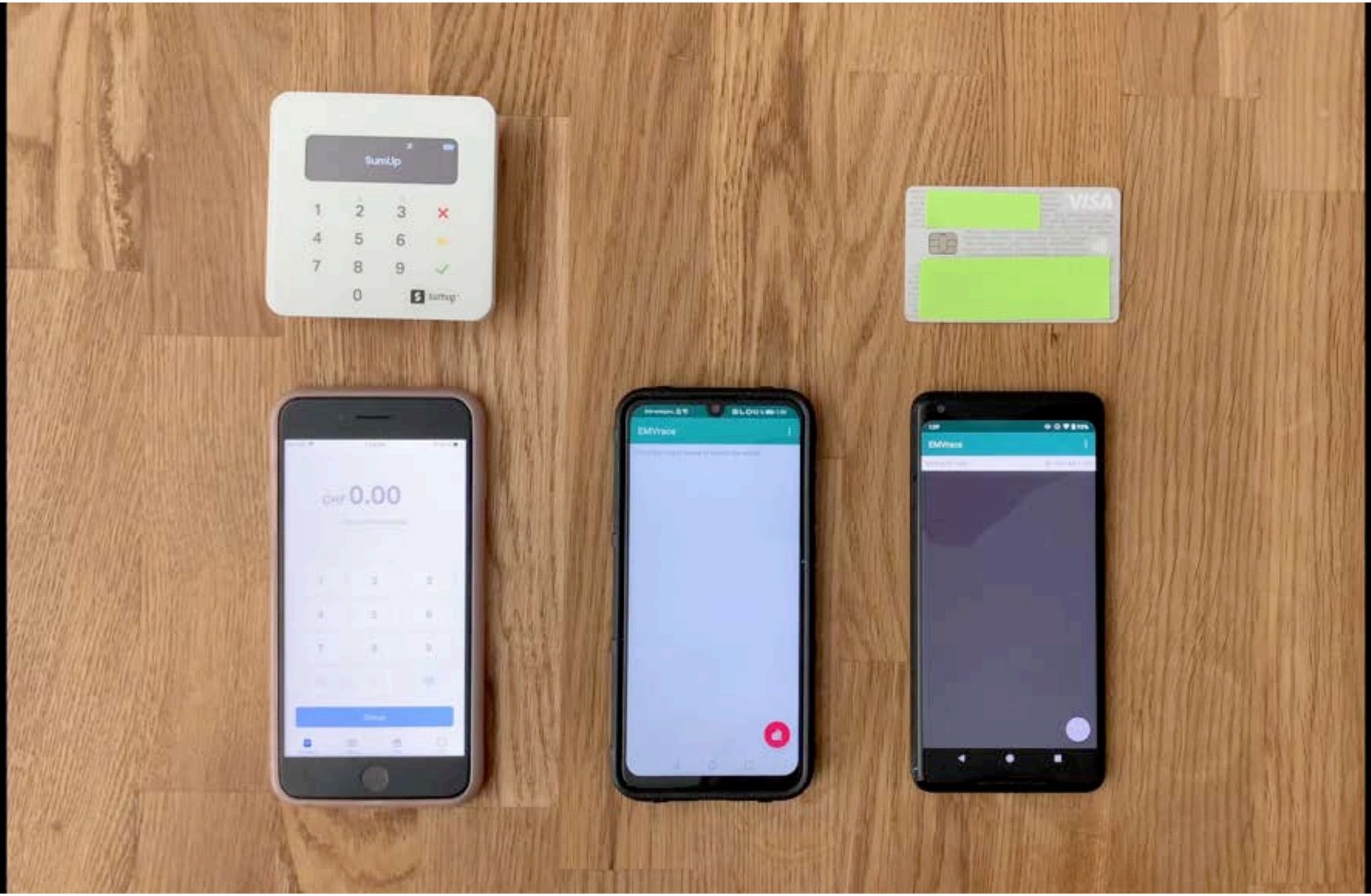
(a) Terminal sends command indicating *Cardholder Verification* required

(b) Card sends response indicating *Online PIN required*

(c) Attacker changes Card Transaction Qualifier (CTQ) to 0x028 indicating that Online PIN not required and Consumer Device CVM was performed

Card emulator

POS emulator


sumUp
1 2 3
4 5 6
7 8 9
0
VISA
CHF 0.00
EMVtest
EMVtest

Media Coverage

The Hacker News

Home | Newsletter | Offers

New PIN Verification Bypass Flaw Affects Visa Contactless Payments

September 07, 2020 Ravie Lakshmanan

ZDNet

Academics bypass PINs for Visa contactless payments

Researchers: "In other words, the PIN is useless in Visa contactless transactions."

By Catalin Cimpanu for Zero Day | August 28, 2020 -- 03:20 GMT (04:20 BST) | Topic: Security

SRF

ETH-Forscher warnen

Sicherheitslücke bei Visa-Kreditkarten entdeckt

- Forschende der ETH Zürich haben eine Sicherheitslücke bei Visa-Kreditkarten entdeckt.
- Damit könnten Betrügerinnen und Betrüger Beträge von Karten abbuchen, die eigentlich mit einem Pin-Code bestätigt werden müssten.
- Andere Unternehmen wie Mastercard oder American Express sind laut ETH nicht betroffen.

Cash Matters®

Why Cash Matters | About Us | News & Articles | Key Facts

Security alert! Visa PIN easily compromised, Swiss study finds

Sept. 3, 2020 Share

heise online

Zahlen ohne PIN – Forscher knacken Visas NFC-Bezahlfunktion

Kontaktlos und ohne PIN bezahlten Forscher mit einer Visa-Karte quasi beliebig teure Produkte.

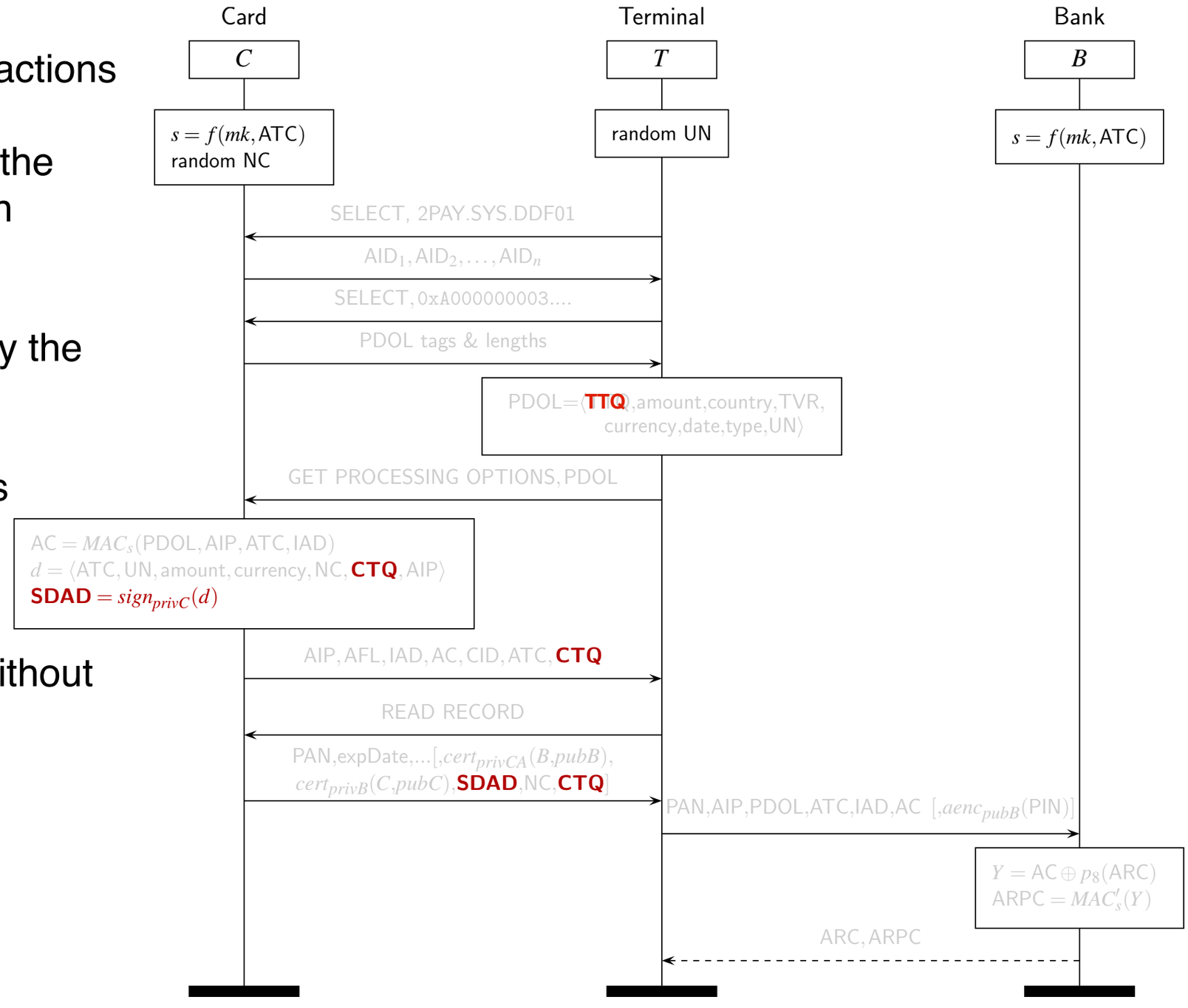
Countermeasure to PIN Bypass

- **Recall the problem**: Most VISA transactions do not use the Signed Dynamic Authentication Data (SDAD), which is the only protection to the Card Transaction Qualifiers (CTQ)

- **Easy Fix:** always have the card supply the SDAD and the terminal verify it

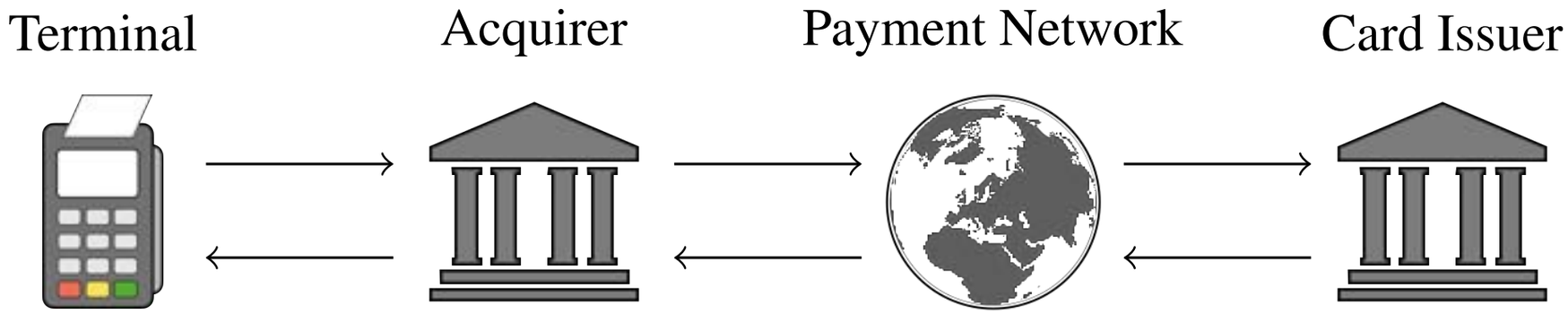
- Having the card supply it is as easy as setting bit 1 of byte 1 of the Terminal Transaction Qualifiers (TTQ)

- Fixes can be deployed on terminals without reissuing cards!



Mastercard Can Be Attacked Too!

After previous work, we enriched our model to account for the fact that there are different payment networks.



Attack idea: replace card's Application Identifiers (AIDs) with the Visa AID `A0000000031010` to deceive the terminal into activating the Visa kernel.

- **Simultaneously perform a Visa transaction with the terminal and a Mastercard transaction with the card.**
- **For Visa transaction, apply previously described attack on Visa!**

Can Also Exploit Failure Modes

Recall Offline Data Authentication (ODA) uses banking PKI.

- **Certificate lookup initiated by Card who provides a CA P*ublic Key Index***

EMV documentation has curious pseudo-code fragment (p. 255):

IF [CA Public Key Index not present in CA Public Key Database]
THEN SET `CDA failed' in Terminal Verification Results (TVR)

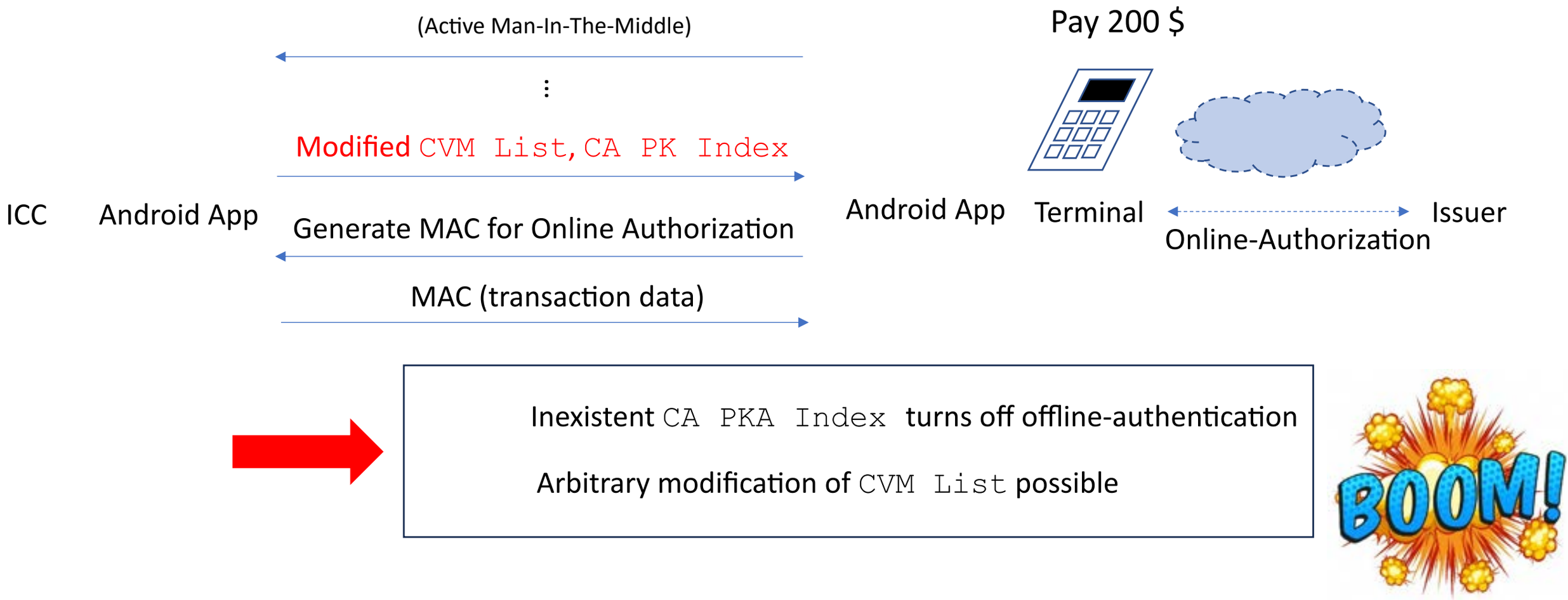
Documentation also states (p. 435):

IF `CDA failed' in *TVR* ∧ … THEN Do *not* request CDA

So, if one can induce a failure of the public key lookup, then Combined Dynamic Data Authentication is not used.

- **One can modify all data not included in the MAC used for online authorization.**
- **Supported CVMs sent from card to terminal are no longer integrity protected.**

Exploiting Errors to Bypass Cardholder Verification



Attacker uses same MITM setup to modify CA data, causing failure.
List of Card Verification Methods can then be modified, without detection.

C8 Redesign

Objective: eliminate problems with past kernels

- Not clean slate, but substantial redesign

Features

- New methods to authenticate transactions
- Modern cryptographic algorithms
- Privacy based on blinded Diffie-Hellmann
- Relay resistance protection
- Simplifications, e.g., magstripe mode compatibility eliminated

Results: substantially enhanced security

- Verification of (almost) all configurations.
  Problematic configurations should not be enabled in practice.
- Privacy (only) against passive adversaries. Validate adversary model in practice!
- Verification of relay resistance, but only in some “safe” configurations.
- **Analysis highlights constraints for secure implementation.**

Cards with C8 to be released soon

Getting Chip Card Payments Right*

vid Basin(✉)[1][0000−0003−2952−939X], Xenia Hofmeier[1][0009−0002−6909−8
Ralf Sasse[1][0000−0002−5632−6099], and Jorge Toro-Pozo[2]

[1] Department of Computer Science, ETH Zurich, Switzerland
{basin,xenia.hofmeier,ralf.sasse}@inf.ethz.ch
[2] SIX Digital Exchange, Switzerland
jorge.toro@sdx.com

Abstract. EMV is the international protocol standard for smart card payments and is used in billions of payment cards worldwide. Despite the standard’s advertised security, various issues have been previously uncovered, deriving from logical flaws that are hard to spot in EMV’s lengthy and complex specification. We have formalized various models of EMV in Tamarin, a symbolic model checker for cryptographic protocols. Tamarin was extremely effective in finding critical flaws, both known and new, and in many cases exploitable on actual cards. We report on these past problems as well as followup work where we verified the latest, improved version of the protocol, the EMV kernel C8. This work puts C8’s correctness on a firm, formal basis, and clarifies which guarantees hold for C8 and under which assumptions. Overall our work supports the thesis that cryptographic protocol model checkers like Tamarin have

Conclusions

Formal Methods matter!

- You can rob the bank with a theorem prover.

Tools sufficiently advanced that they can and should be used

- Good hygiene: be explicit about protocol, adversary, and properties
- Find errors or produce proofs
- Follow standardization efforts: check modifications for upcoming releases
  EMV not a standard but Tamarin is being used now as part of its development

Research challenges

- **COMPLEXITY, Complexity, complexity**
- Improving scope and accuracy
- Education: getting the message out and training engineers

Bibliography (see https://emvrace.github.io/)

- **Pin Bypass on VISA Cards discovered with Tamarin**
  D.B., Ralf Sasse, Jorge Toro Pozo, *The EMV Standard: Break, Fix, Verify,* Oakland Security & Privacy, **2021.** (Best practical paper award)

- **Attacks on Mastercard found via payment network "confusion"**
  D.B., Ralf Sasse, Jorge Toro Pozo, *Card Brand Mixup Attack: Bypassing the PIN in non-VISA Cards by Using Them for Visa Transactions,* Usenix Security, **2021.**

- **Attacks on Different EMV cards found by inducing PKI errors+downgrading**
  D.B., Patrick Schaller, Jorge Torzo Pozo, *Inducing Authentication Failures to Bypass Credit Card PINs,* Usenix Security, **2023.**

- **Fixes proposed using UWB replay protection (PURE)**
  Daniele Coppola, Giovanni Camurati, Claudio Anliker, Xenia Hofmeier, Patrick Schaller, D.B., Srdjan Capkun. PURE: Payments with UWB Relay-protection, Usenix Security **2024.**

- **Collaboration with Mastercard to verify redesign of EMV (C-8 Kernel),**
  D.B., Xenia Hofmeier, Ralf Sasse, *Getting Electronic Payments Right,* FM **2024.**