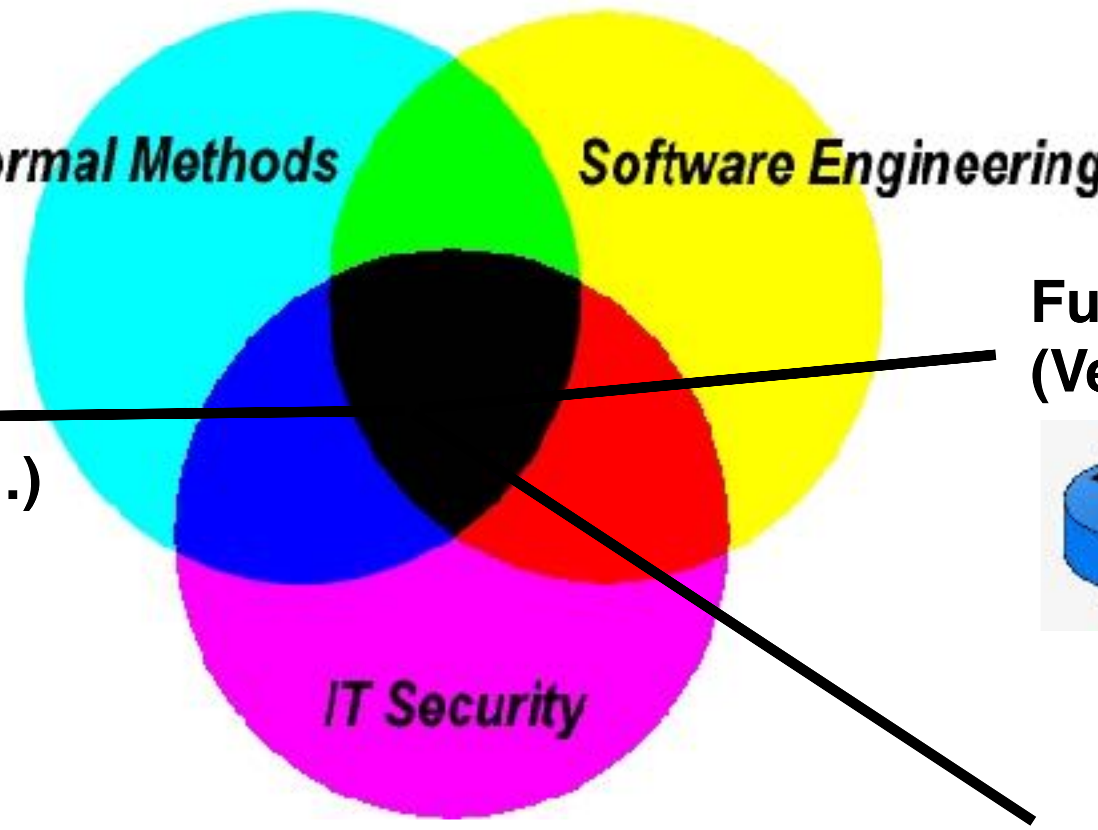
Analyzing Payment Protocols with Tamarin

David Basin
ETH Zurich

June 2022

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

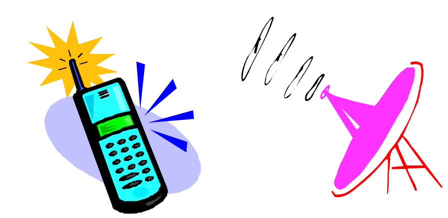
Research Areas
(Interested? Come talk to me!)



Runtime Monitoring Tools
(Critical systems, data protection, DLT, …)

Future Internet
(Verified SCION)

Security Protocol
Verification w/ Tamarin

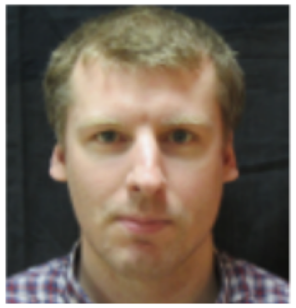


Foundations, Methods, and Tools for Analyzing and Building Security-Critical Systems

Research on Tamarin & EMV — Collaborators

Tamarin Team


Simon Meier


Benedikt Schmidt


Cas Cremers


Ralf Sasse


Jannik Dreier

...

EMV


Ralf Sasse


Jorge Toro Pozo

A Typical Protocol

IKE, Phase 1, Main Mode, Digital Signatures, Simplified

$$
\begin{array}{lll}
(1) & I \rightarrow R: & C_I,\ ISA_I \\
(2) & R \rightarrow I: & C_I,\ C_R,\ ISA_R \\
(3) & I \rightarrow R: & C_I,\ C_R,\ g^x,\ N_I \\
(4) & R \rightarrow I: & C_I,\ C_R,\ g^y,\ N_R \\
(5) & I \rightarrow R: & C_I,\ C_R,\ \{ID_I,\ SIG_I\}_{SKEYID_e} \\
(6) & R \rightarrow I: & C_I,\ C_R,\ \{ID_R,\ SIG_R\}_{SKEYID_e}
\end{array}
$$

Properties?

$$
\begin{array}{lll}
SKEYID & = & h(\{N_I, N_R\}, g^{xy}) \\
SKEYID_d & = & h(SKEYID, \{g^{xy}, C_I, C_R, 0\}) \\
SKEYID_a & = & h(SKEYID, \{SKEYID_d, g^{xy}, C_I, C_R, 1\}) \\
SKEYID_e & = & h(SKEYID, \{SKEYID_a, g^{xy}, C_I, C_R, 2\}) \\
HASH_I & = & h(SKEYID_a, \{g^x, g^y, C_I, C_R, ISA_I, ID_I\}) \\
HASH_R & = & h(SKEYID_a, \{g^y, g^x, C_R, C_I, ISA_R, ID_R\}) \\
SIG_I & = & \{HASH_I\}_{K_I^{-1}} \\
SIG_R & = & \{HASH_R\}_{K_R^{-1}}
\end{array}
$$

Does argument order matter?

Why all the nested keyed hashes?

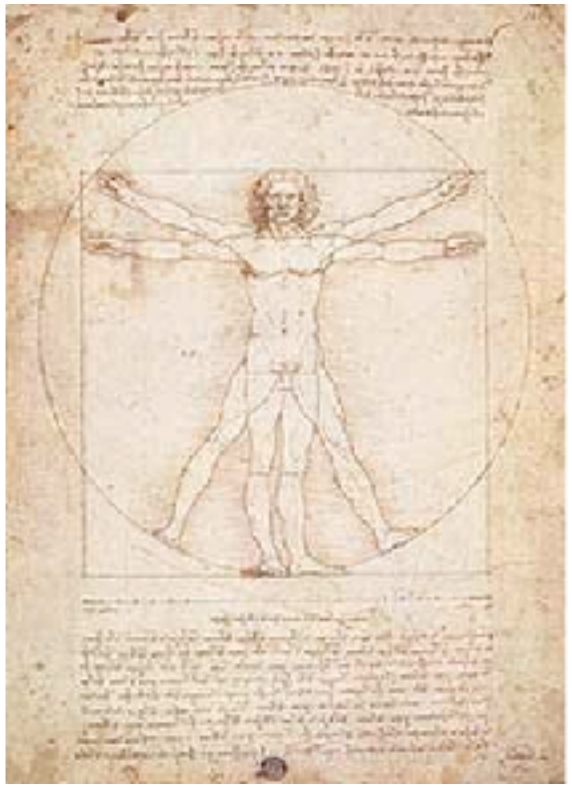
Protocol Design as an Art

Best practices, design by committee, reuse of previous protocols, ...

Whenever I made a roast, I always started off by cutting off the ends, just like my grandmother did. Someone once asked me why I did it, and I realized I had no idea. It had never occurred to me to wonder. It was just the way it was done. Eventually I asked my grandmother. “Why do you always cut off the ends of a roast?” She answered “Because my pan is small and otherwise the roasts would not fit.”

— *Anonymous*

Protocol Design as a Science



Science in the root sense

The discovery and knowledge of something that can be demonstrated and verified within a community

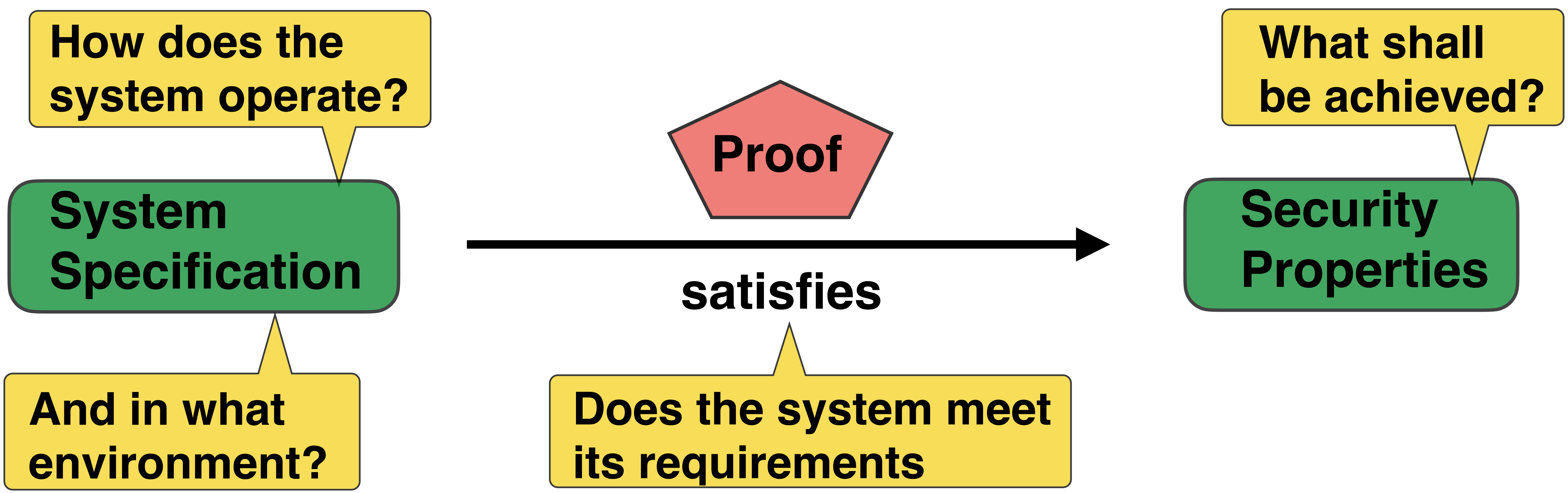
Formal methods as a way to better protocols

- Precise specification of system, environment, properties
- Tool support to debug, verify, and explore alternatives

Progress is being made applying tools to protocols that matter

- 5G, TLS 1.3, EMV, …
- Companies are (slowly) becoming tool users

Where is the Difficulty?



- Design documents are incomplete and imprecise
- Unclear adversary model

- Undecidability
- Even restricted cases intractable

- Properties implicit or imprecise. E.g. “authenticate”

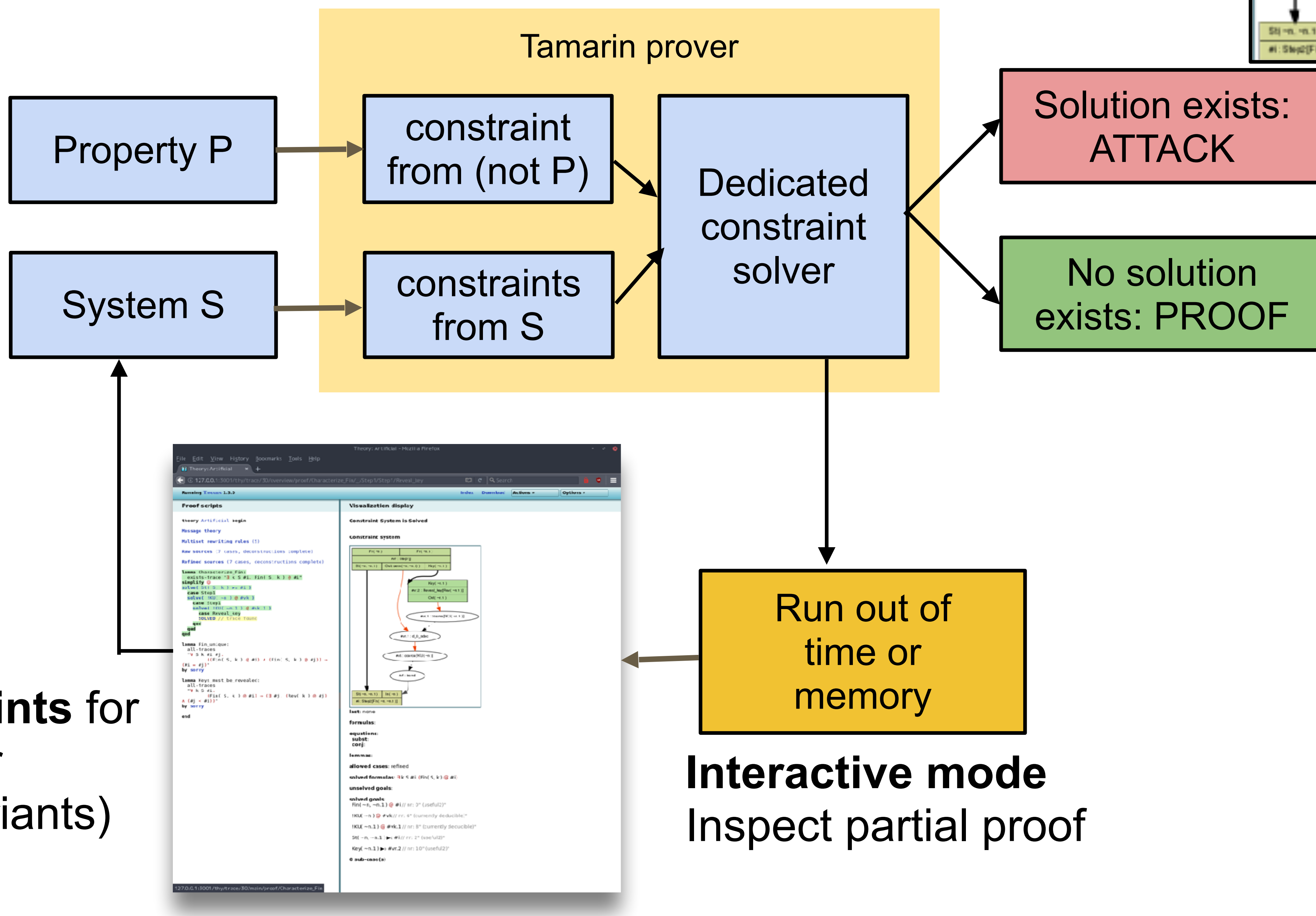
What is Tamarin?

Theorem Prover

Constraint solver

Tamarin prover

Tamarin Prover



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

Specifying Protocols with Multiset Rewrite Rules

LHS --[actions]-> RHS

```
[ In( K ),
  State( ThreadID, `step1’ ) ]
```

premises (LHS)

```
  --[ Accepted( ThreadID, K) ]->
```

actions

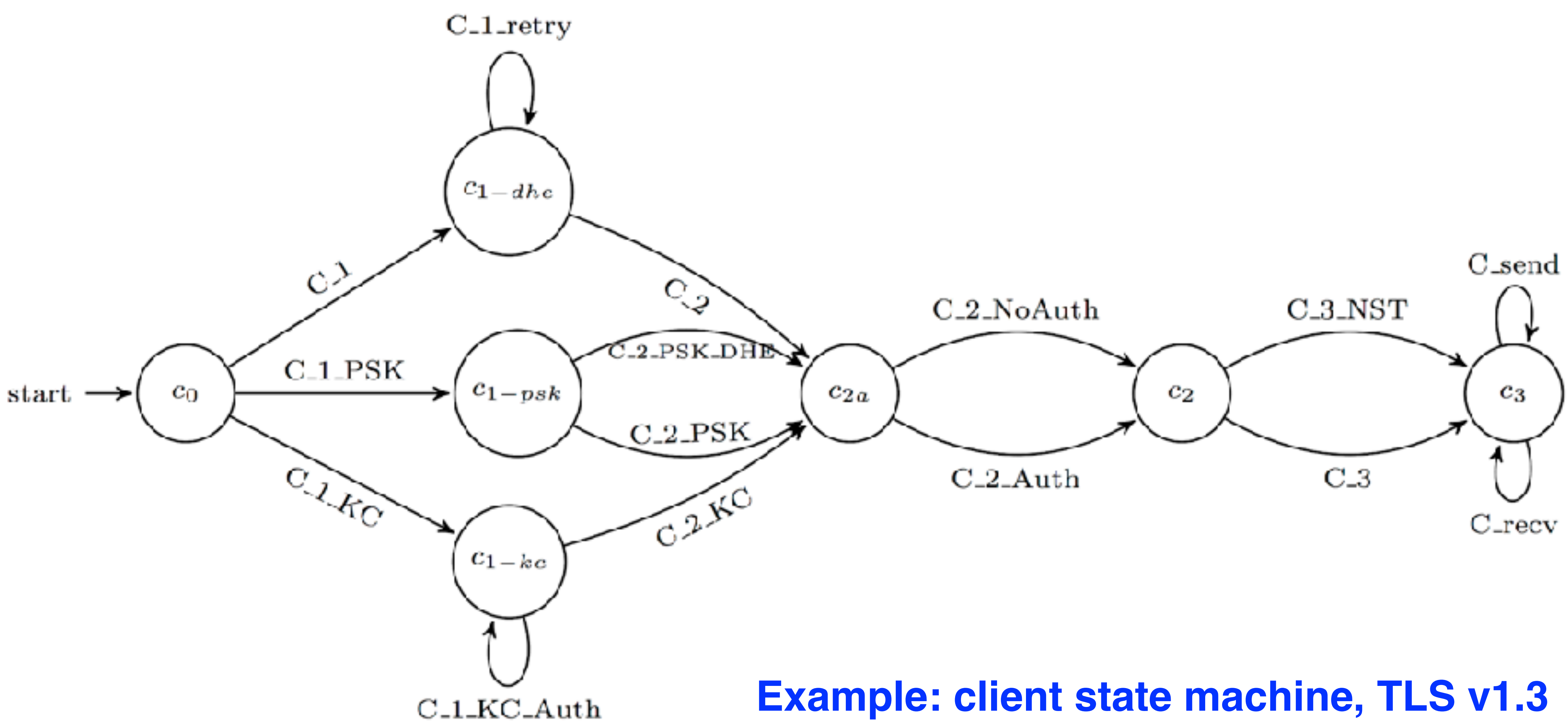
```
[ Out( `ack` ),
  State( ThreadID, `step2’, K ) ]
```

conclusions (RHS)

Gives rise to a transition system with a trace semantics

{In(key), State(tid3,`step1’), …}** —**Accepted(tid3,key)**→ **{Out(`ack’), State(tid3,`step2’,key), …} —**…**→ **…**

Specifying Protocols



Example: client state machine, TLS v1.3

Rules correspond to edges

Specifying Adversary Capabilities

Example of "Session Reveal"

```
[ State( ThreadID, … , Key ) ]
    --[ SessionKeyReveal( ThreadID, Key ) ]->
 [ Out( Key ) ]
```

Similar to oracles in computational model

Specifying Properties

Guarded fragment of first order logic with timepoints

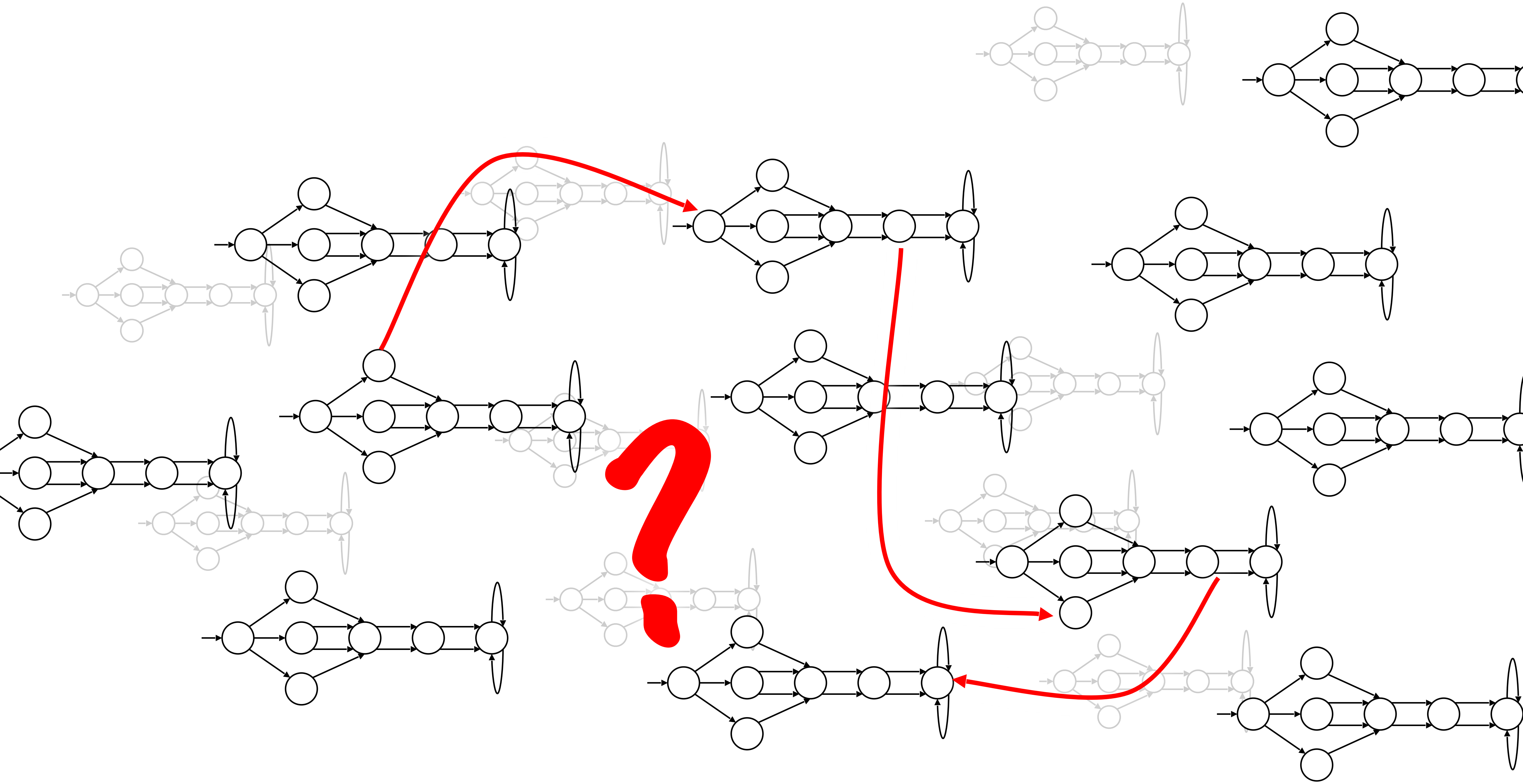
```
lemma my_secret_key:
   "Forall tid key #i.
      Accepted( tid, key )@i => ( not Ex #j. K(key)@j ) "
```

Interpreted over traces

{In(key),
State(tid3,`step1'),
...} —Accepted(tid3,key)→ {Out(`ack'),
State(tid3,`step2',key),
...} → ...

Does Protocol Satisfy Property?

Or can the adversary attack it?



See references at end of talk

EMV Standard

EMV is the global standard for smartcard payments: 9+ billion cards used!

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

The standard claims to offer the highest security

PAYMENTS ARE SAFE AND SECURE

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!

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!

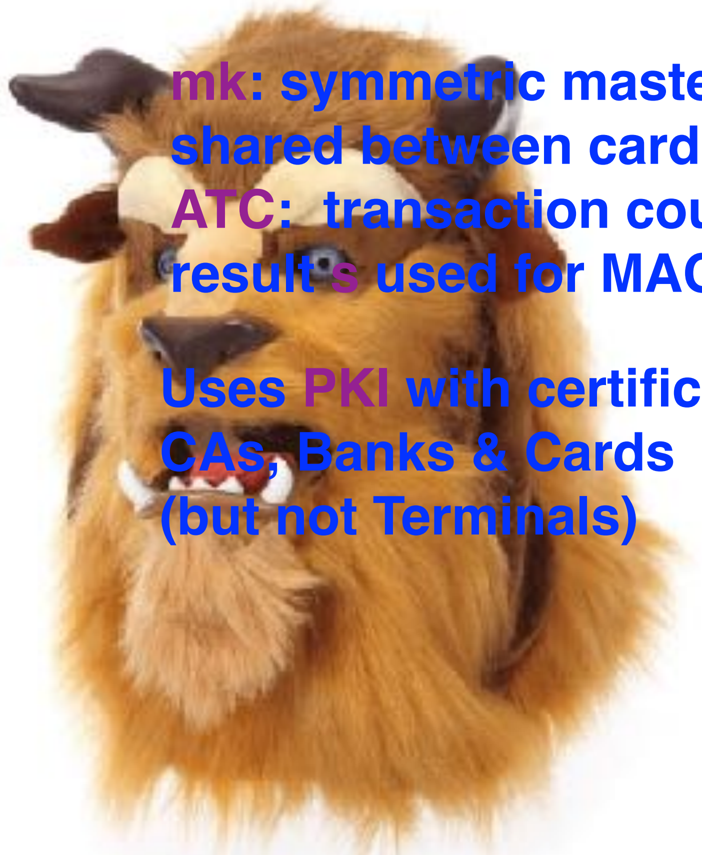
3. **We proposed and machine-checked fixes (disclosed to relevant vendors)**
   Fixes do not affect cards in circulation

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

Details described on the web at emvrace.github.io

EMV Protocol

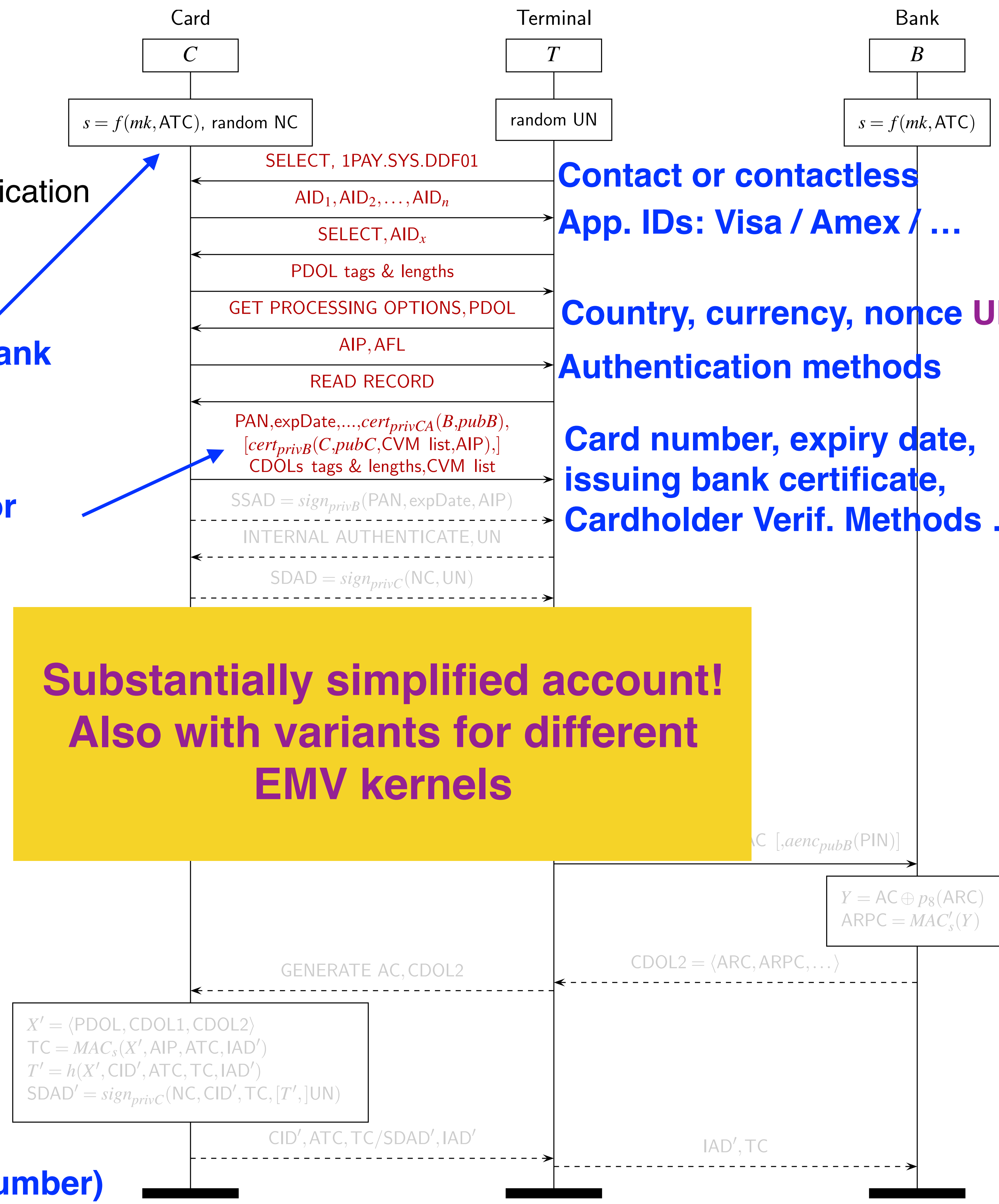
1. **Initialization:** card & terminal agree on application used for transaction & exchange static data.



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

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

2,000+ pages



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

Country, currency, nonce UN
Authentication methods

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

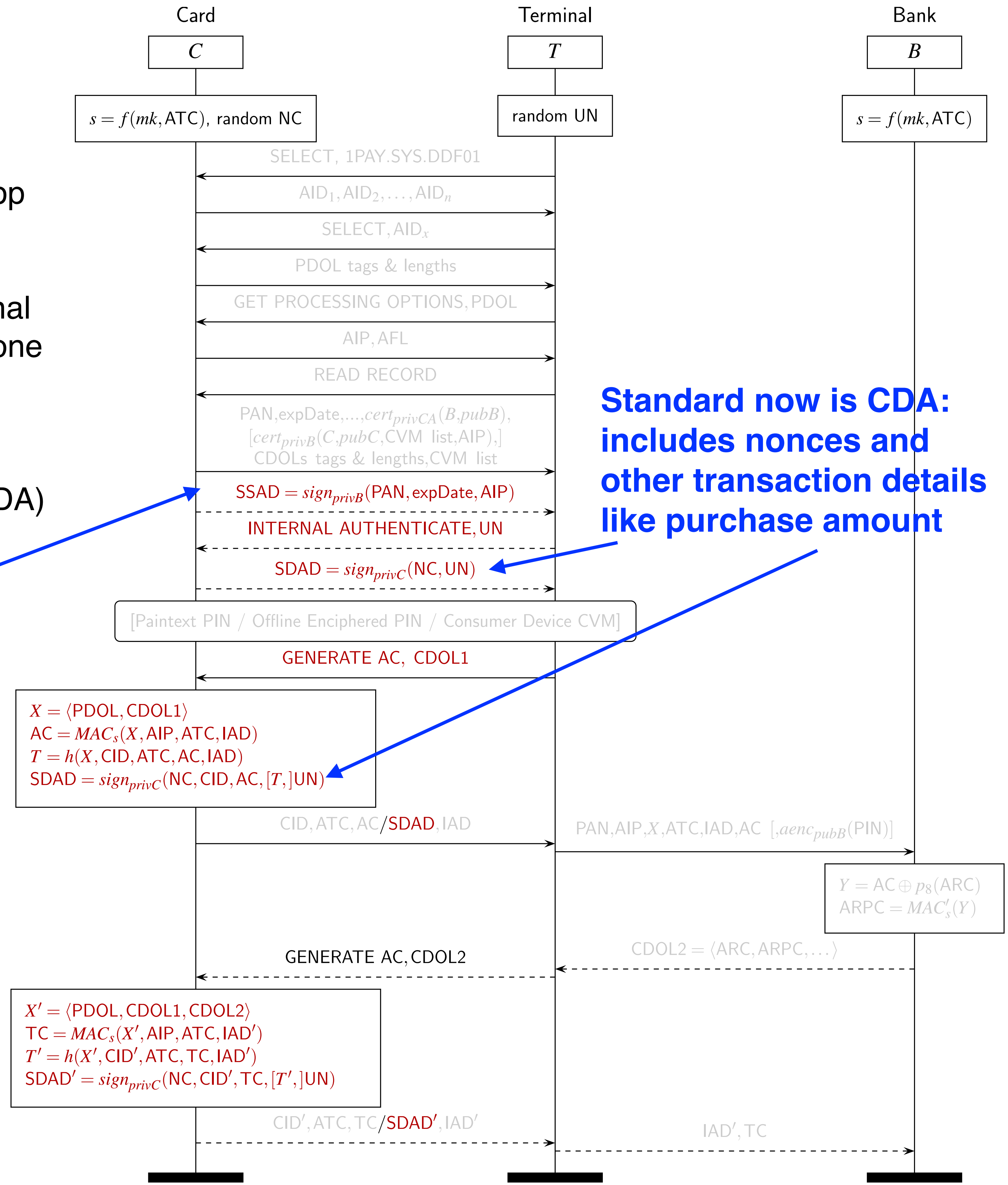
Substantially simplified account! Also with variants for different EMV kernels

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)



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

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

Acronym Zoo:

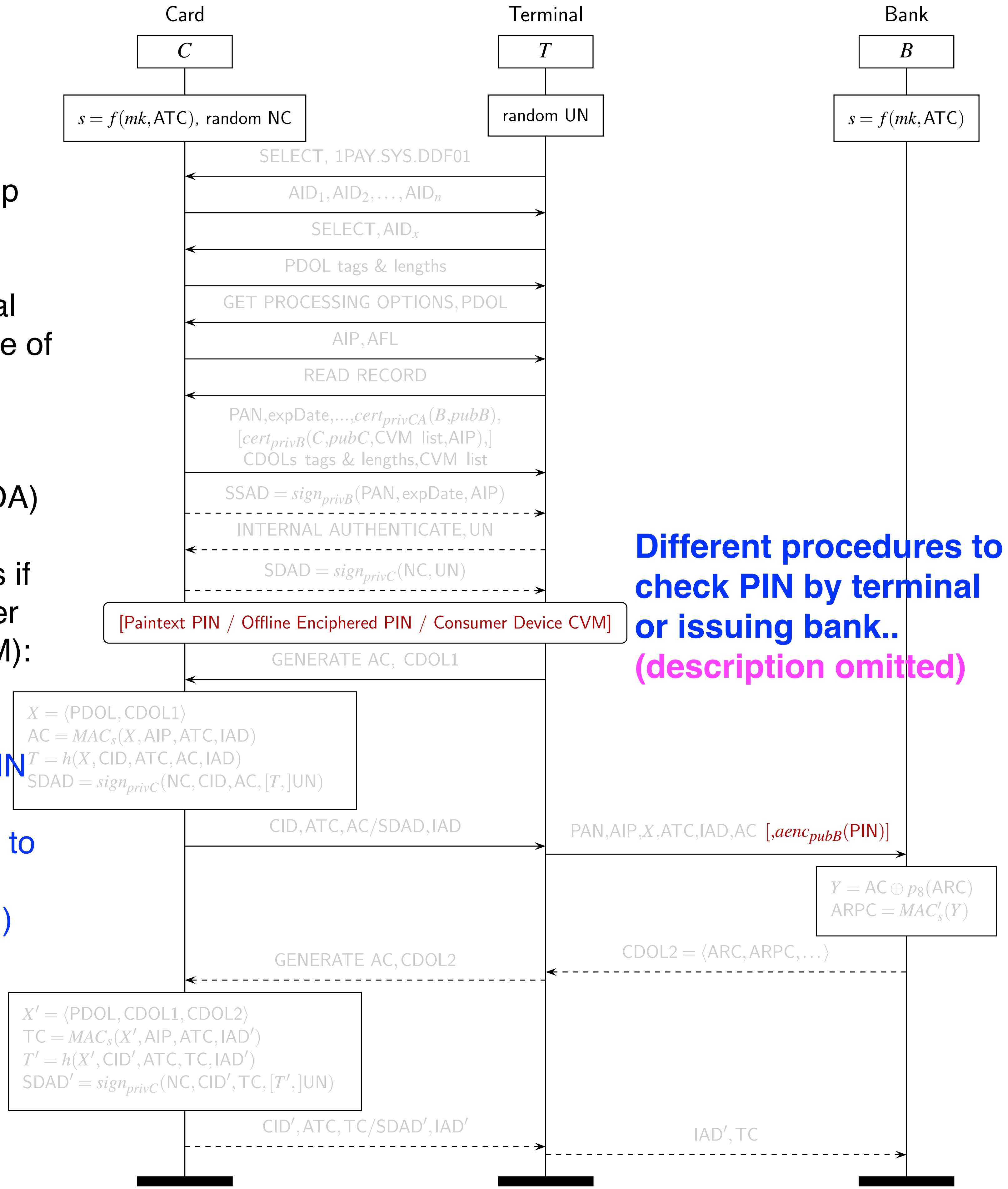
SDAD = Signed Dynamic Authentication Data

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)

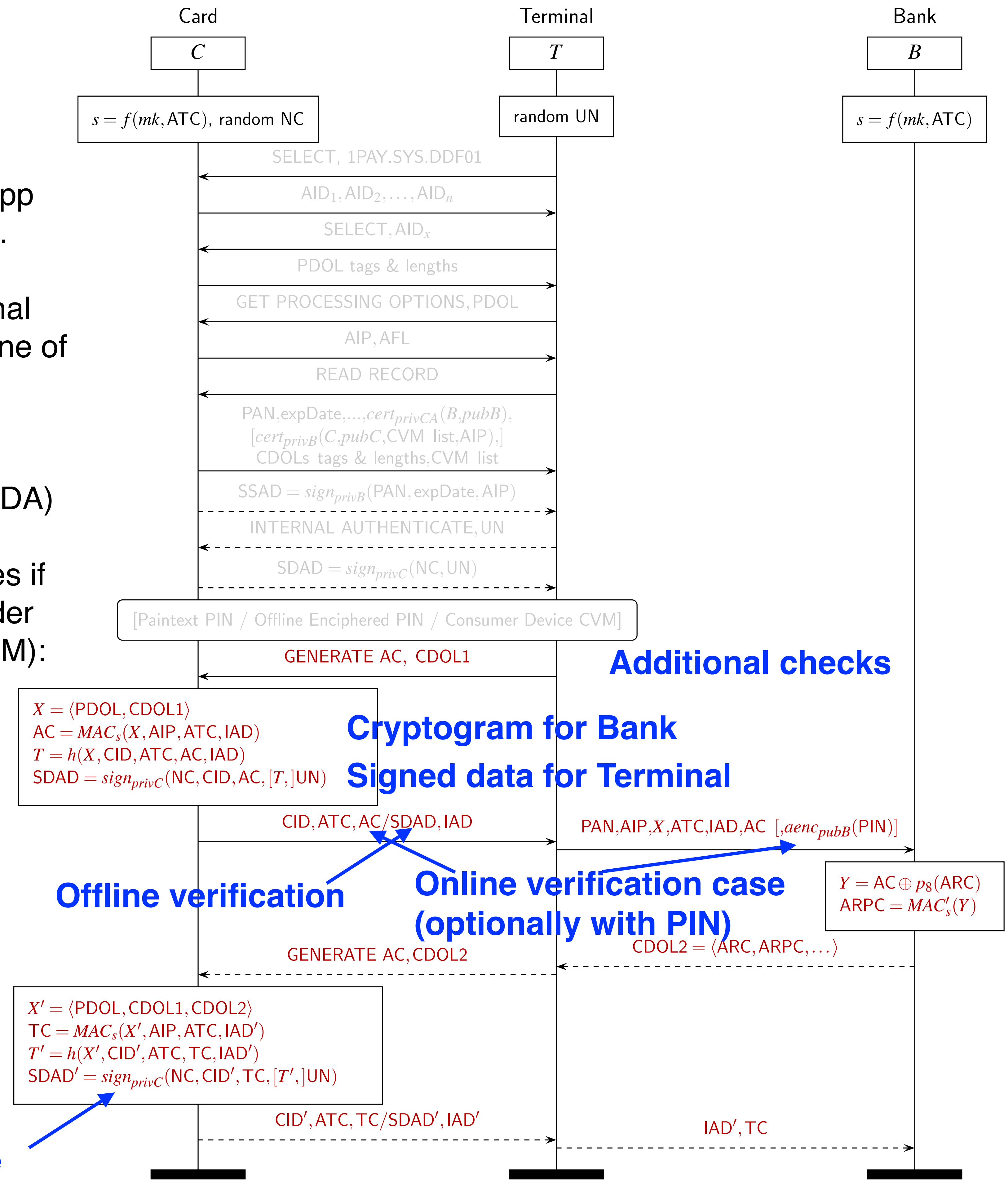
EMV Protocol

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

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 a (possibly 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

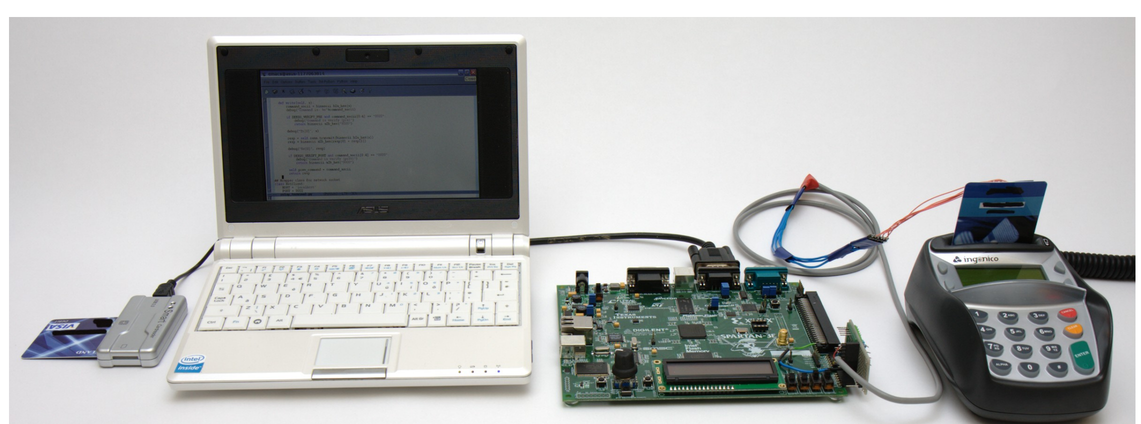
Attack: downgrade to plain PIN verification, and read PIN via MITM

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

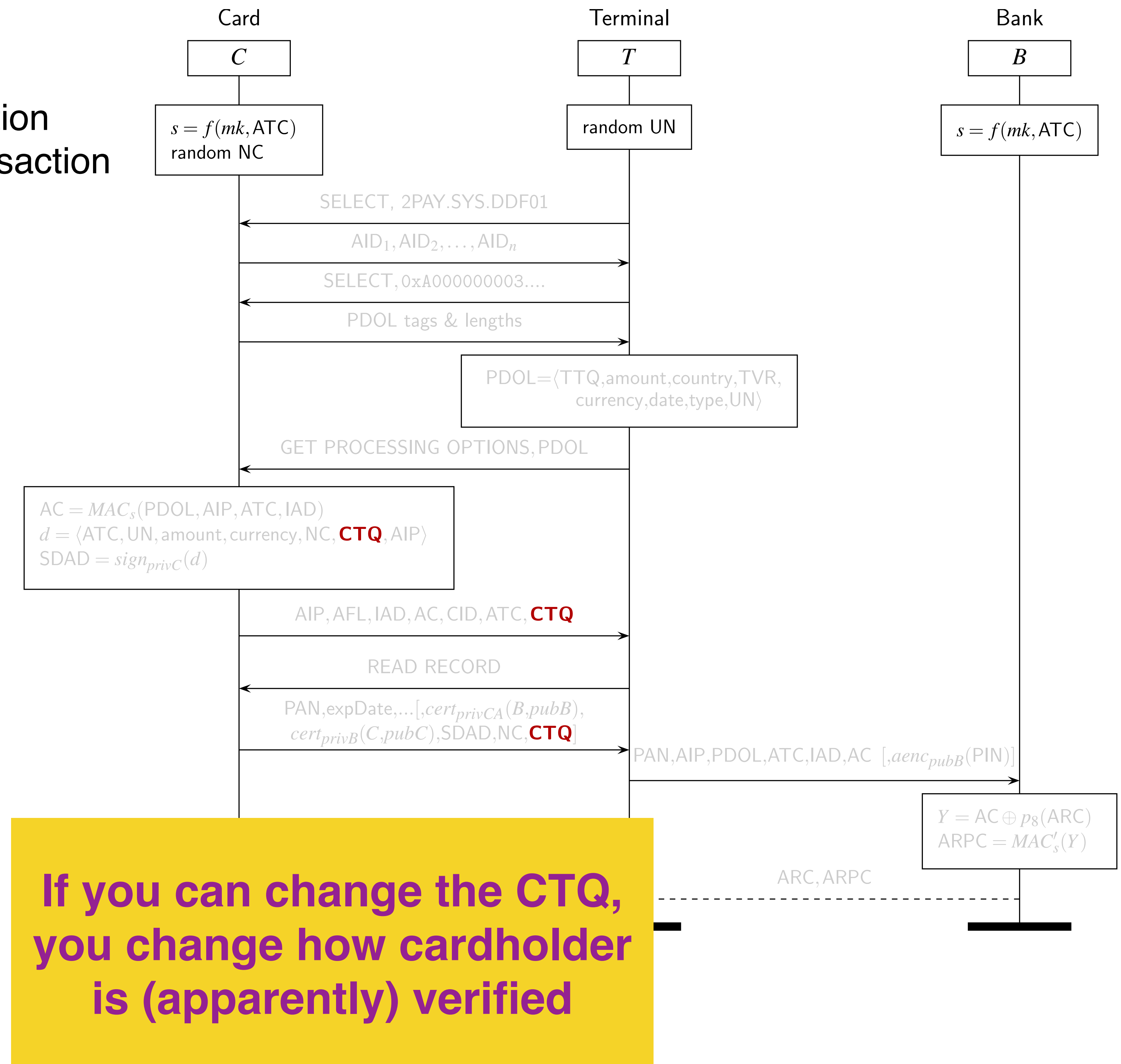
Target Model	exec.	bank accepts	auth. to terminal	auth. to bank
Visa_EMV_Low	✓	✓	× (1)	× (1)
Visa_EMV_High	✓	✓	× (1)	× (1)
Visa_DDA_Low	✓	× (2)	× (2)	✓
Visa_DDA_High	✓	✓	✓	✓
Mastercard_SDA_OnlinePIN_Low	✓	× (2)	× (2)	✓
Mastercard_SDA_OnlinePIN_High	✓	✓	✓	✓
Mastercard_SDA_NoPIN_Low	✓	× (2)	× (2)	✓
Mastercard_SDA_NoPIN_High	– (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**

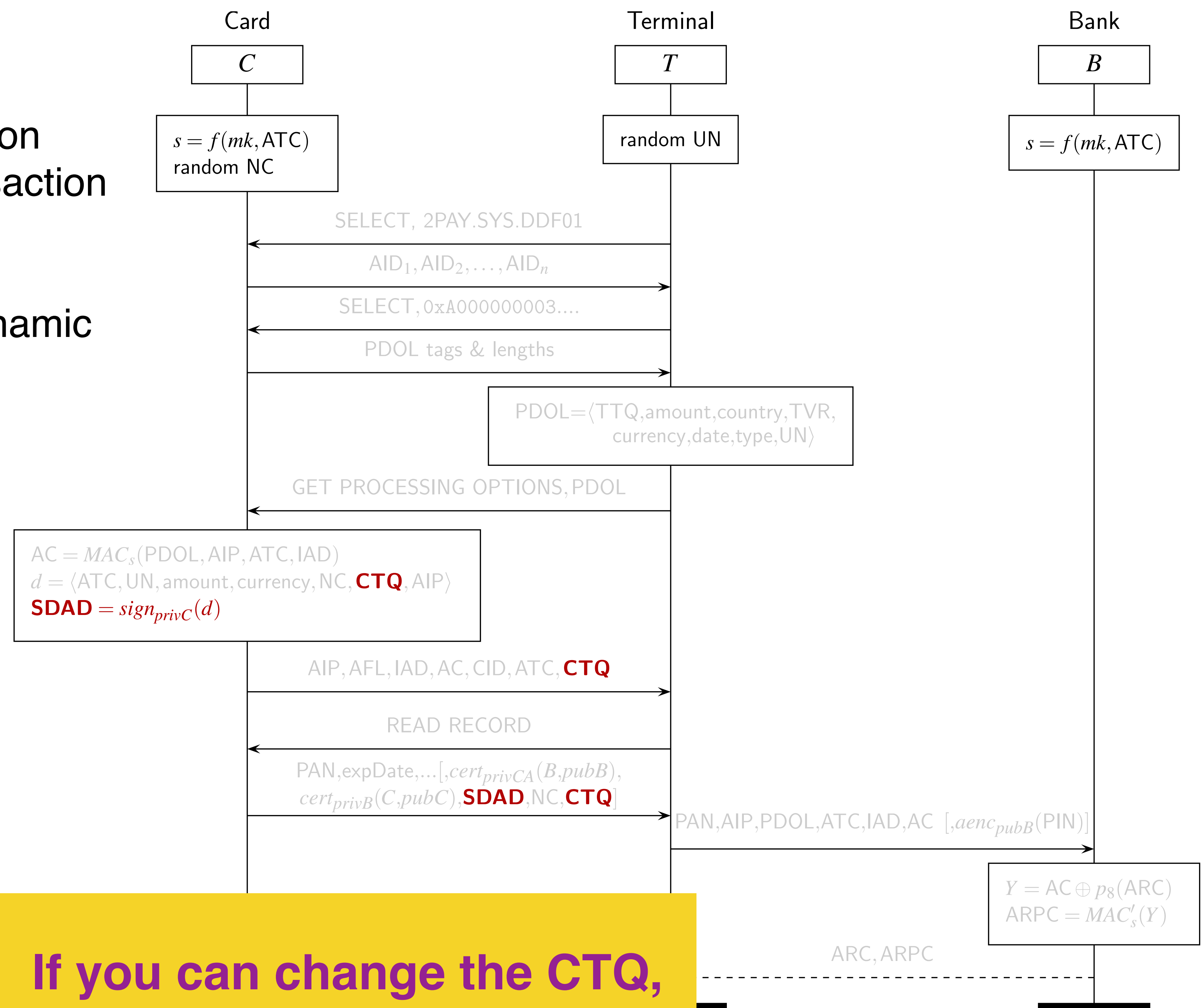
Problem with Visa Contactless

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



Problem with Visa Contactless

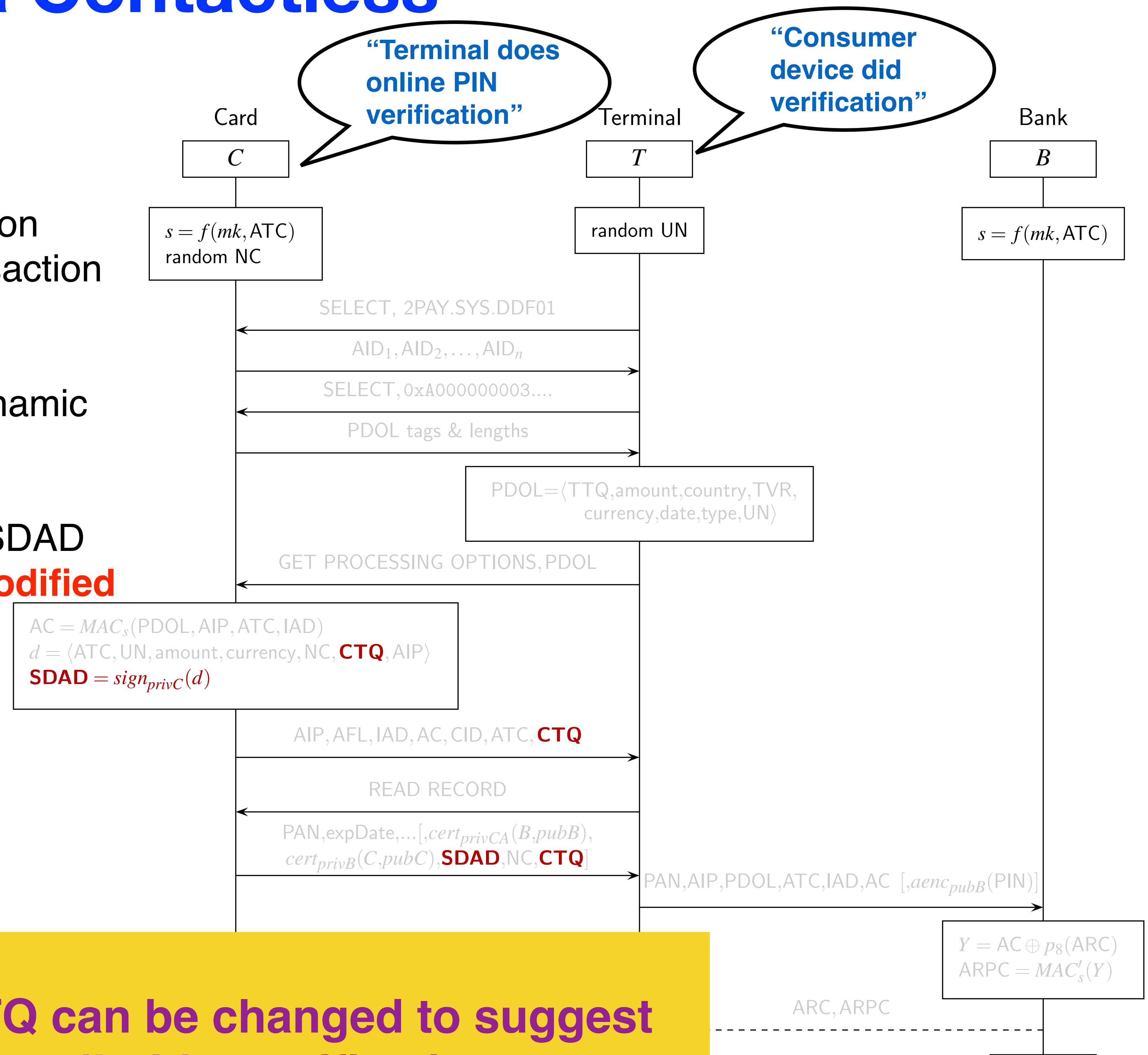
- Card's choice for Cardholder Verification Method (CVM) encoded in Card Transaction Qualifiers (CTQ)
- CTQ authenticated via the Signed Dynamic Authentication Data (SDAD)



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

Problem with Visa Contactless

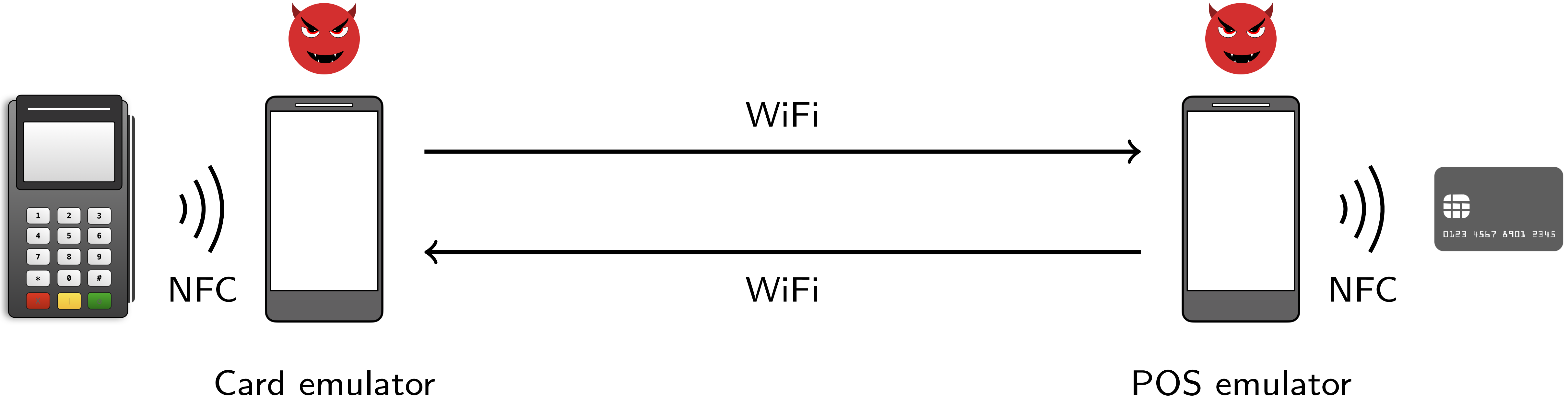
- Card's choice for Cardholder Verification Method (CVM) encoded in Card Transaction Qualifiers (CTQ)
- CTQ authenticated via the Signed Dynamic Authentication Data (SDAD)
- Most Visa transactions don't use the SDAD ⇒ CTQ and therefore **CVM can be modified**



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



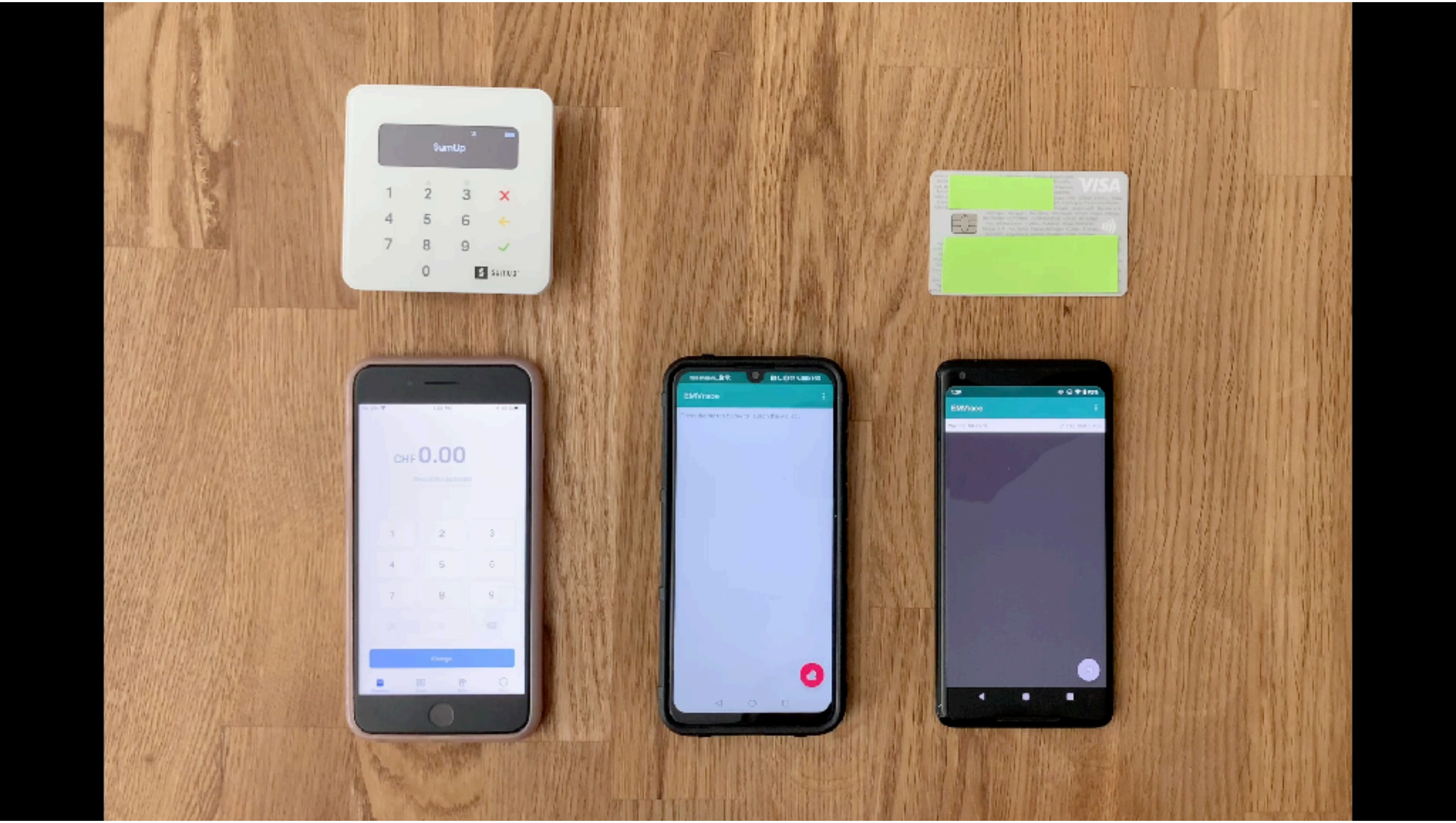
Weaponizing PIN bypass Attack

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


SumUp
CHF 0.00
EMVrace
EMVrace
VISA

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

News › Schweiz ›

ETH-Forscher warnen

Sicherheitslücke bei Visa-Kreditkarten entdeckt

Dienstag, 01.09.2020, 11:49 Uhr

Dieser Artikel wurde 8-mal geteilt.

- 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.

Lesezeit: 2 Min. · speichern · 360

(Bild: ETH, Basin et al.)

16:54 Uhr | Security

Von Jürgen Schmidt

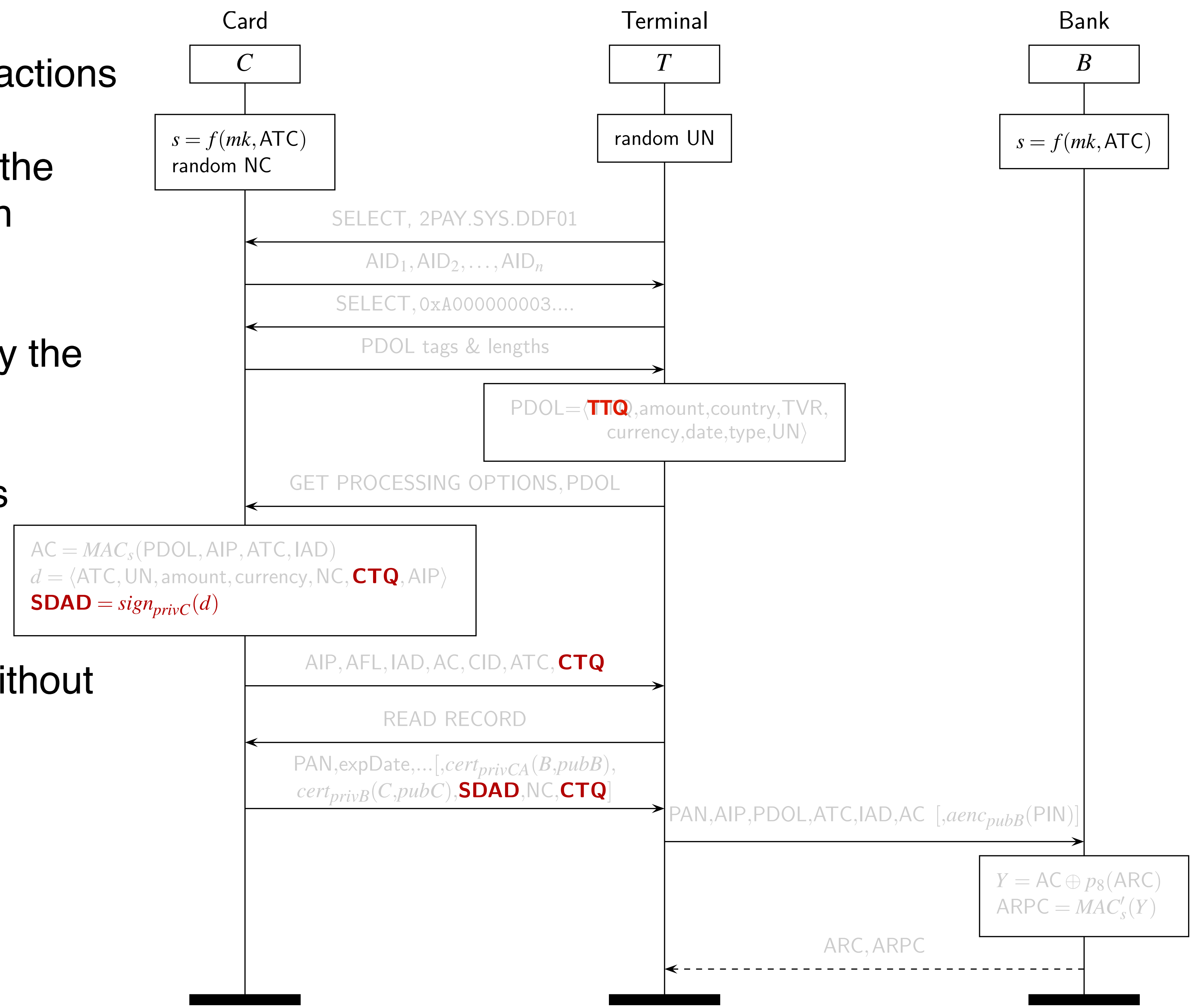
Countermeasure to PIN Bypass

- **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!



Other Issues found

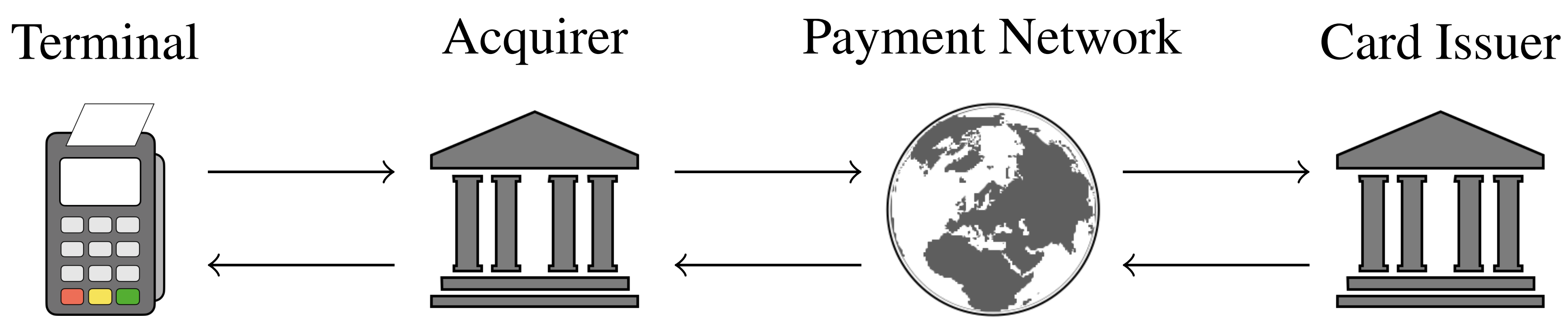
Target Model	exec.	bank accepts	auth. to terminal	auth. to bank
Visa_EMV_Low	✓	✓	×(1)	×(1)
Visa_EMV_High	✓	✓	×(1)	×(1)
Visa_DDA_Low	✓	×(2)	×(2)	✓
Visa_DDA_High	✓	✓	✓	✓
Mastercard_SDA_OnlinePIN_Low	✓	×(2)	×(2)	✓
Mastercard_SDA_OnlinePIN_High	✓	✓	✓	✓
Mastercard_SDA_NoPIN_Low	✓	×(2)	×(2)	✓
Mastercard_SDA_NoPIN_High	–(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

- Low-value **offline** transactions with Visa or old Mastercard are **not secure**
- **Weaponize:** MITM fools terminal into accepting a transaction where bank declines, only after attacker is long gone
- Didn't test in the wild for ethical reasons
- **Fix**: Change the SDAD input to authenticate additional data, e.g., the AC (cryptogram) and its input. So changes detected by terminals.
- Requires 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!**

Current work: verification project with an EMV partner to analyze upcoming changes to standard.

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

References (including some background)

- D.B., Ralf Sasse, Jorge Toro Pozo, *The EMV Standard: Break, Fix, Verify,* Oakland Security & Privacy, 2021. (Best practical paper award)
- 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.
- Simon Meier, D.B., Cas Cremers. *Efficient Construction of Machine-Checked Symbolic Protocol Security Proofs*, Journal of Computer Security 2013.
- D.B., Cas Cremers, Cathy Meadows, *Model Checking Security Protocols*, Handbook of Model Checking, 2018.
- Benedikt Schmidt, Simon Meier, Cas Cremers, D.B., *Automated Analysis of Diffie-Hellman Protocols and Advanced Security Properties*, CSF 2012.