Crypto protocols

ACS R209: Computer Security – Principles and Foundations

Ross Anderson

Security Protocols

- Security protocols are the intellectual core of security engineering
- They are where cryptography and system mechanisms meet
- They allow trust to be taken from where it exists to where it's needed
- But they are much older then computers…

Real-world protocol

- Ordering wine in a restaurant
  - Sommelier presents wine list to host
  - Host chooses wine; sommelier fetches it
  - Host samples wine; then it's served to guests
- Security properties?

Real-world protocol

- Ordering wine in a restaurant
  - Sommelier presents wine list to host
  - Host chooses wine; sommelier fetches it
  - Host samples wine; then it's served to guests
- Security properties
  - Confidentiality – of price from guests
  - Integrity – can't substitute a cheaper wine
  - Non-repudiation – host can't falsely complain

Car unlocking protocols

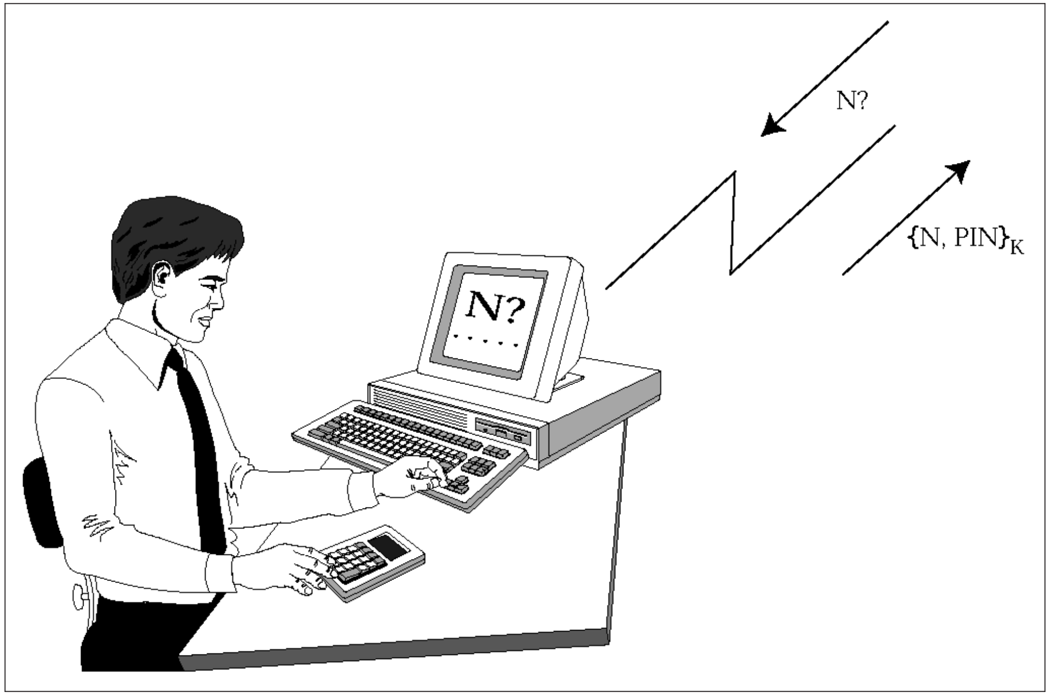
- Principals are the engine controller E and the car key transponder T
- Static ($T \rightarrow E$: KT)
- Non-interactive

  $T \rightarrow E: T, \{T,N\}_{KT}$
- Interactive

  $E \rightarrow T: N$

  $T \rightarrow E: \{T,N\}_{KT}$
- N is a ‘nonce’ for ‘number used once’. It can be a sequence number, a random number or a timestamp

Two-factor authentication



S → U: N

U → P: N, PIN

P → U: $\{N, PIN\}_{KP}$

Key management protocols

- Suppose Alice and Bob each share a key with Sam, and want to communicate?
  – Alice calls Sam and asks for a key for Bob
  – Sam sends Alice a key encrypted in a blob only she can read, and the same key also encrypted in another blob only Bob can read
  – Alice calls Bob and sends him the second blob
- How can they check the protocol's fresh?

Needham-Schroder

- 1978: uses 'nonces' rather than timestamps

  $A \rightarrow S$: A, B, NA

  $S \rightarrow A$: $\{NA, B, K_{AB}, \{K_{AB}, A\}_{KBS}\}_{KAS}$

  $A \rightarrow B$: $\{K_{AB}, A\}_{KBS}$

  $B \rightarrow A$: $\{NB\}_{KAB}$

  $A \rightarrow B$: $\{NB - 1\}_{KAB}$

- The bug, and the controversy…

Identify Friend or Foe (IFF)

- Basic idea: fighter challenges bomber

  $F \rightarrow B: N$

  $B \rightarrow F: \{N\}_K$

- What can go wrong?

Identify Friend or Foe (IFF)

- Basic idea: fighter challenges bomber

  $F \rightarrow B: N$

  $B \rightarrow F: \{N\}_K$

- What if the bomber reflects the challenge back at the fighter's wingman?

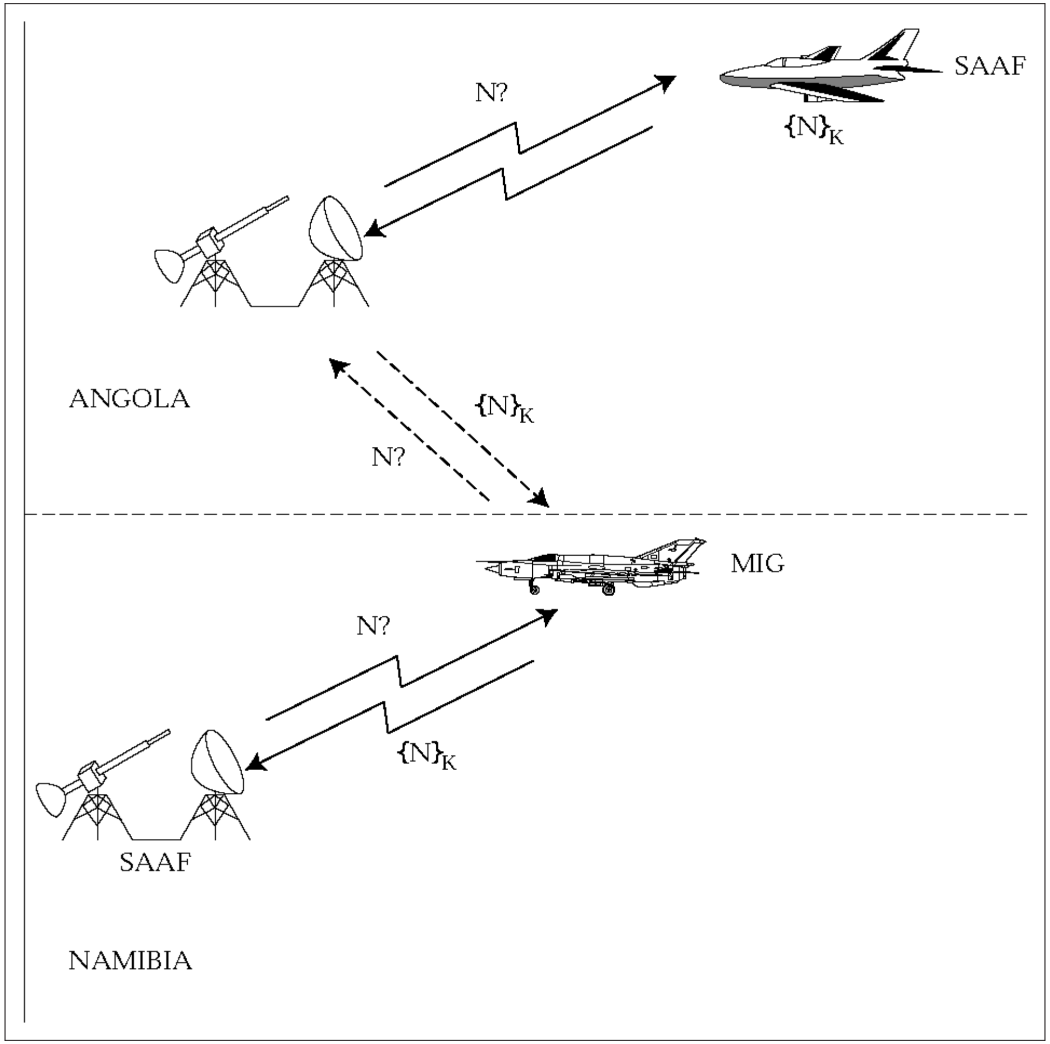
  $F \rightarrow B: N$

  $B \rightarrow \mathit{F}: N$

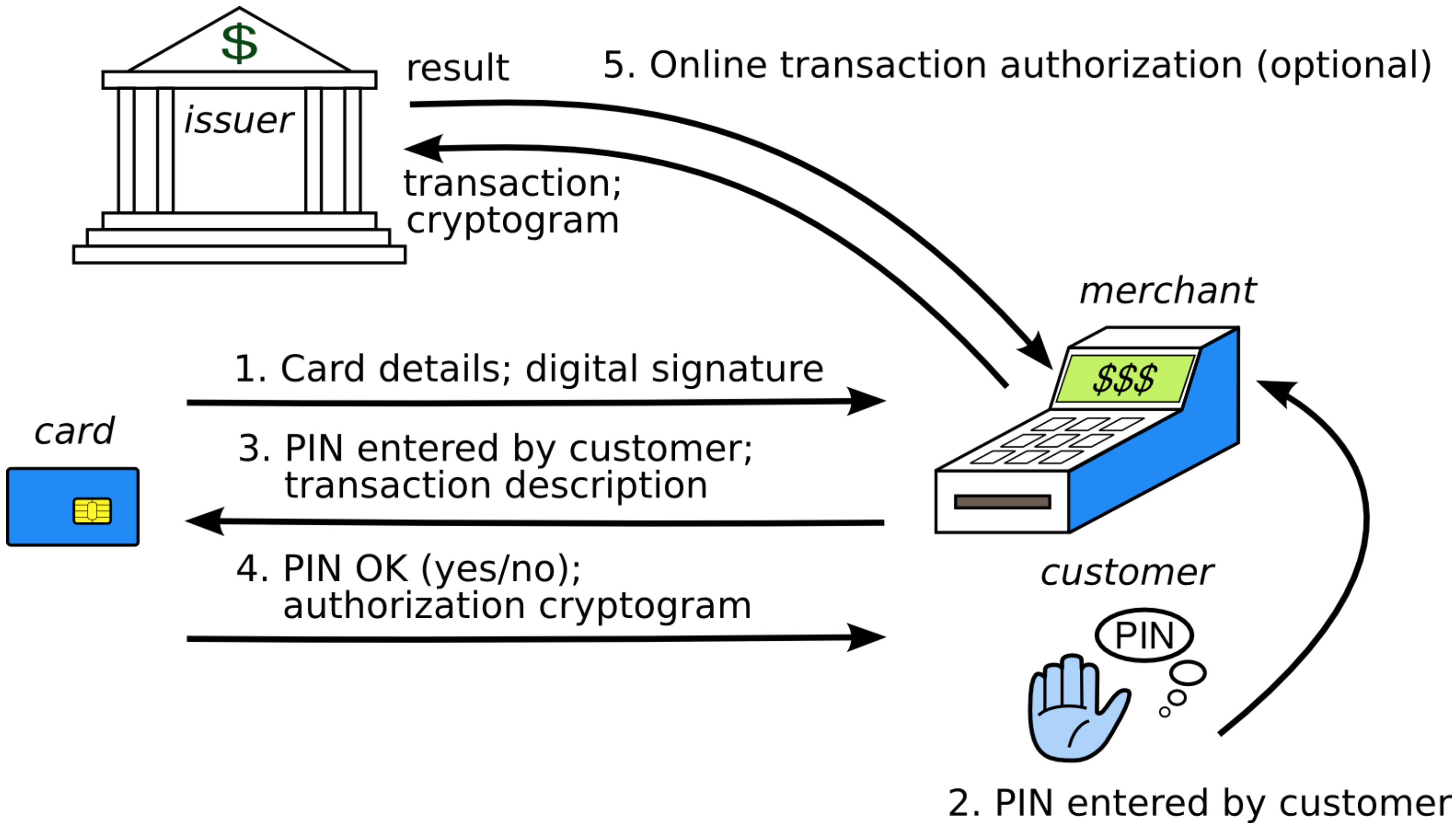
  $\mathit{F} \rightarrow B: \{N\}_K$

  $B \rightarrow F: \{N\}_K$

IFF (2)



A normal EMV transaction



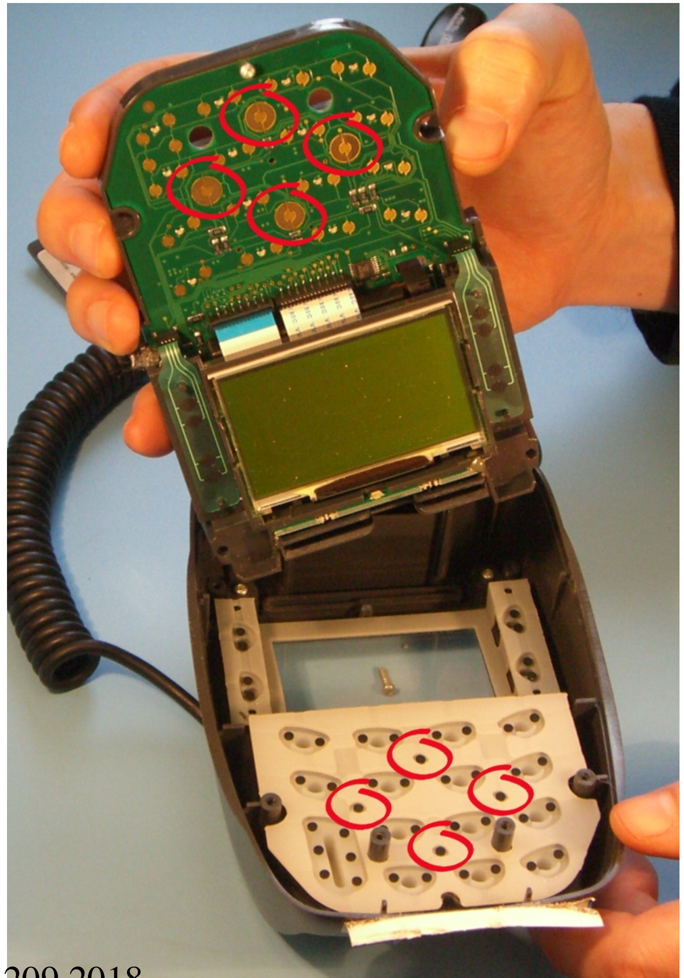
What about a false terminal?

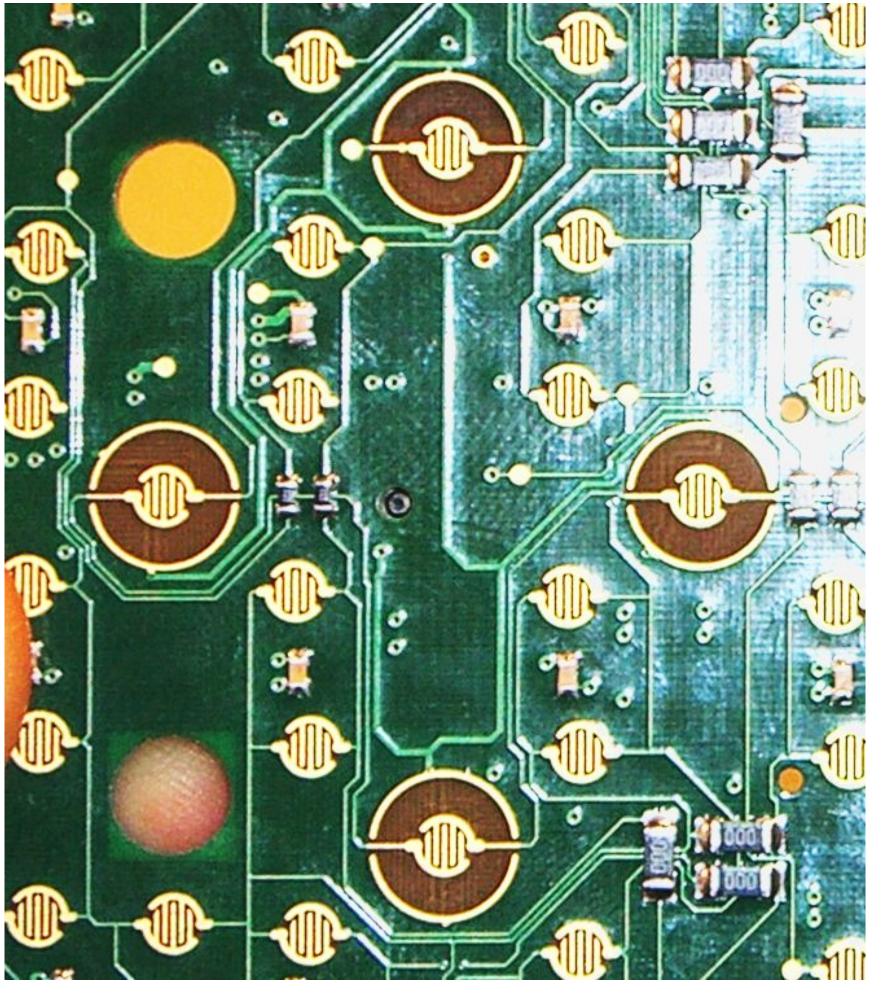
- Replace a terminal's insides with your own electronics
- Capture cards and PINs from victims
- Use them to do a man-in-the-middle attack in real time on a remote terminal in a merchant selling expensive goods

Attacks in the real world

- The relay attack is almost unstoppable, and we showed it in TV in February 2007
- But it seems never to have happened!
- So far, mag-strip fallback fraud has been easy
- PEDs tampered at Shell garages by 'service engineers' (PED supplier was blamed)
- Then 'Tamil Tigers'
- After fraud at BP Girton: we investigate

Tamper switches (Ingenico i3300)

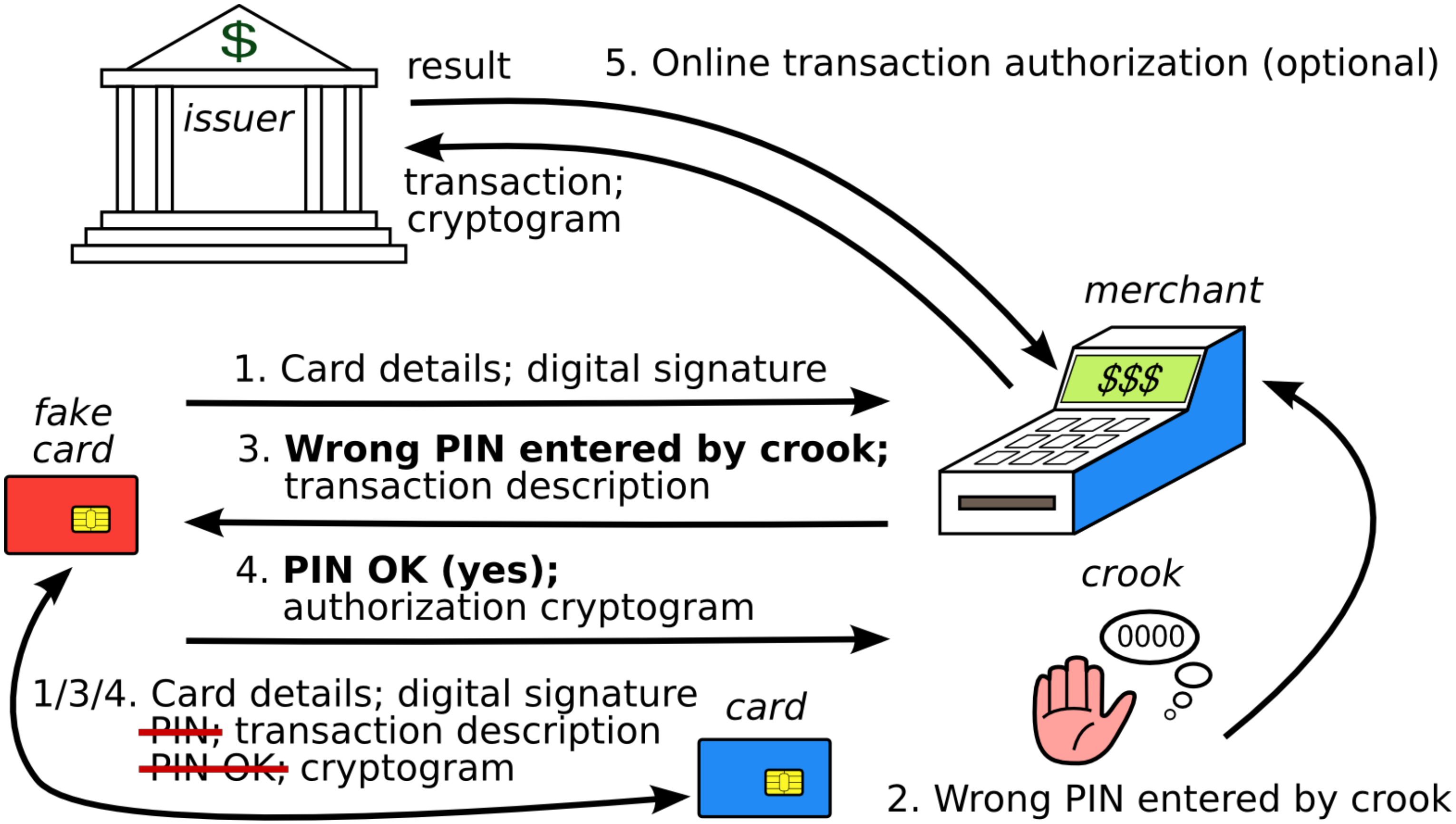




TV demo: Feb 26 2008

- PEDs 'evaluated under the Common Criteria' were trivial to tap
- Acquirers, issuers have different incentives
- GCHQ wouldn't defend the CC brand
- APACS said (Feb 08) it wasn't a problem…
- Khan case (July 2008)

The ‘No-PIN’ attack (2010)



Fixing the ‘No PIN’ attack

- In theory: might block at terminal, acquirer, issuer
- In practice: may have to be the issuer (as with terminal tampering, acquirer incentives are poor)
- Barclays introduced a fix July 2010; removed Dec 2010 (too many false positives?); banks asked for student thesis to be taken down from web instead
- Real problem: EMV spec now far too complex
- With 100+ vendors, 20,000 banks, millions of merchants … everyone passes the buck (or tries to sell ECC…)

EMV and Random Numbers

- In EMV, the terminal sends a random number N to the card along with the date d and the amount X
- The card computes an authentication request cryptogram (ARQC) on N, d, X
- What happens if I can predict N for d?
- Answer: if I have access to your card I can precompute an ARQC for amount X, date d

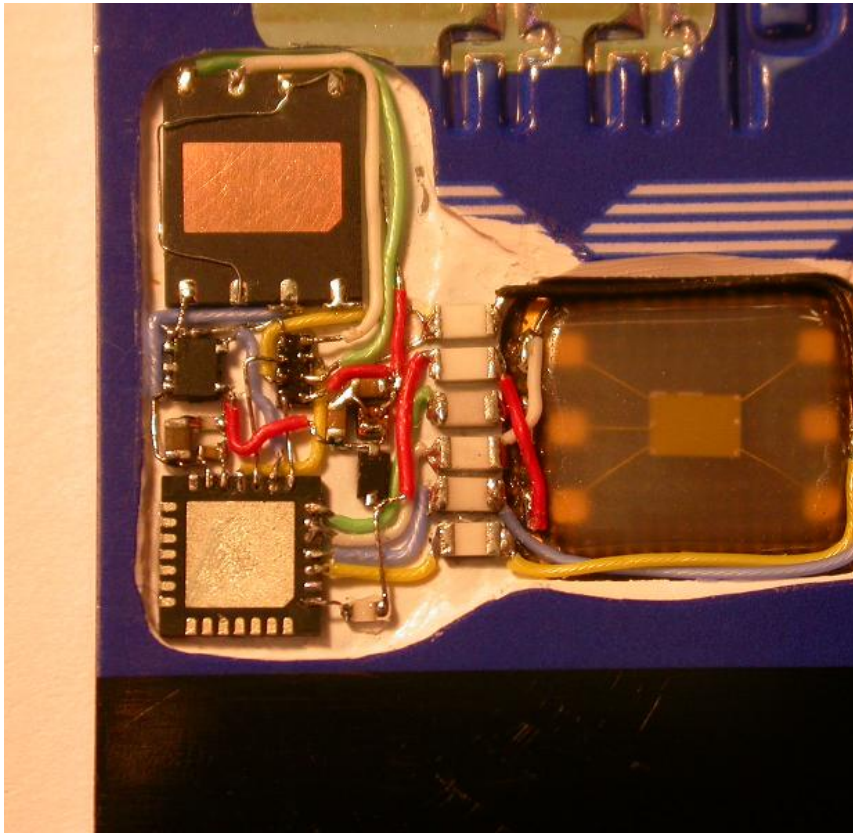
ATMs and Random Numbers (2)

- Log of disputed transactions at Majorca:

2011-06-28	10:37:24	F1246E04
2011-06-28	10:37:59	F1241354
2011-06-28	10:38:34	F1244328
2011-06-28	10:39:08	F1247348

- N is a 17 bit constant followed by a 15 bit counter cycling every 3 minutes
- We test, & find half of ATMs use counters!

ATMs and Random Numbers (3)



The preplay attack

- Collect ARQCs from a target card
- Use them in a wicked terminal at a collusive merchant, which fixes up nonces to match
- Since then, we won a live case…
- Sailor spent €33 on a drink in a Spanish bar. He got hit with ten transactions for €3300, an hour apart, from one terminal, through three different acquirers, with ATC collisions

Public Key Crypto Revision

- Digital signatures: computed using a private signing key on hashed data
- Can be verified with corresponding public verification key
- Can't work out signing key from verification key
- Typical algorithms: DSA, elliptic curve DSA
- We'll write $sig_A\{X\}$ for the hashed data X signed using A's private signing key

Public Key Crypto Revision (2)

- Public key encryption lets you encrypt data using a user's public encryption key
- She can decrypt it using her private decryption key
- Typical algorithms Diffie-Hellman, RSA
- We'll write $\{X\}_A$
- Big problem: knowing whose key it is!

PKC Revision – Diffie-Hellman

- Diffie-Hellman: underlying metaphor is that Anthony sends a box with a message to Brutus
- But the messenger's loyal to Caesar, so Anthony puts a padlock on it
- Brutus adds his own padlock and sends it back to Anthony
- Anthony removes his padlock and sends it to Brutus who can now unlock it
- Is this secure?

PKC Revision – Diffie-Hellman (2)

- Electronic implementation:

  $A \rightarrow B$: M^{rA}

  $B \rightarrow A$: M^{rArB}

  $A \rightarrow B$: M^{rB}

- But encoding messages as group elements can be tiresome so instead Diffie-Hellman goes:

  $A \rightarrow B$: g^{rA}

  $B \rightarrow A$: g^{rB}

  $A \rightarrow B$: $\{M\}g^{rArB}$

Public-key Needham-Schroeder

- Proposed in 1978:

  $A \rightarrow B: \{NA, A\}_{KB}$

  $B \rightarrow A: \{NA, NB\}_{KA}$

  $A \rightarrow B: \{NB\}_{KB}$

- The idea is that they then use $NA \oplus NB$ as a shared key
- Is this OK?

Public-key Needham-Schroeder (2)

- Attack found eighteen years later, in 1996:

  $A \rightarrow C$: $\{NA, A\}_{KC}$

  $C \rightarrow B$: $\{NA, A\}_{KB}$

  $B \rightarrow C$: $\{NA, NB\}_{KA}$

  $C \rightarrow A$: $\{NA, NB\}_{KA}$

  $A \rightarrow C$: $\{NB\}_{KC}$

  $C \rightarrow B$: $\{NB\}_{KB}$

- Fix: explicitness. Put all names in all messages

Public Key Certification

- One way of linking public keys to principals is for the sysadmin to physically install them on machines (common with SSH, IPSEC)
- Another is to set up keys, then exchange a short string out of band to check you're speaking to the right principal (STU-II, Bluetooth simple pairing)
- Another is certificates. Sam signs Alice's public key (and/or signature verification key)

  $CA = sig_S\{T_S, L, A, K_A, V_A\}$
- But this is still far from idiot-proof…

The Denning-Sacco Protocol

- In 1982, Denning and Sacco pointed out the revocation problem with Needham-Schroder and argued that public key crypto should be used instead

  $A \rightarrow S$: A, B

  $S \rightarrow A$: CA, CB

  $A \rightarrow B$: CA, CB, $\{sig_A\{T_A, K_{AB}\}\}_{KB}$

- What's wrong?

The Denning-Sacco Protocol (2)

- Twelve years later, Abadi and Needham noticed that Bob can now masquerade as Alice to anyone in the world!

$A \rightarrow S$: A, B

$S \rightarrow A$: CA, CB

$A \rightarrow B$: CA, CB, $\{sig_A\{T_A, K_{AB}\}\}_{KB}$

$B \rightarrow S$: B, C

$S \rightarrow B$: CB, CC

$B \rightarrow C$: CA, CC, $\{sig_A\{T_A, K_{AB}\}\}_{KC}$

Public Key Protocol Problems

- It's very easy to set up keys with the wrong people – man-in-the-middle attacks get more pervasive. Assumptions are slippery to pin down
- Technical stuff too – if the math is exposed, an attacker may use it against you!
- So data being encrypted (or signed) must be suitably packaged
- Many other traps, some extremely obscure but some rather general…

Chosen protocol attack

- Suppose that we had a protocol for users to sign hashes of payment messages (such a protocol was proposed in 1990s):

  $C \rightarrow M$: order

  $M \rightarrow C$: X   [= hash(order, amount, date, …)]

  $C \rightarrow M$: $\text{sig}_K\{X\}$

- How might this be attacked?

Chosen protocol attack (2)

The Mafia demands you sign a random challenge to prove your age for porn sites!