UNIVERSITY OF
CAMBRIDGE
Computer Laboratory

Security APIs
- Digital Battlefields

Mike Bond

Computer Security Group

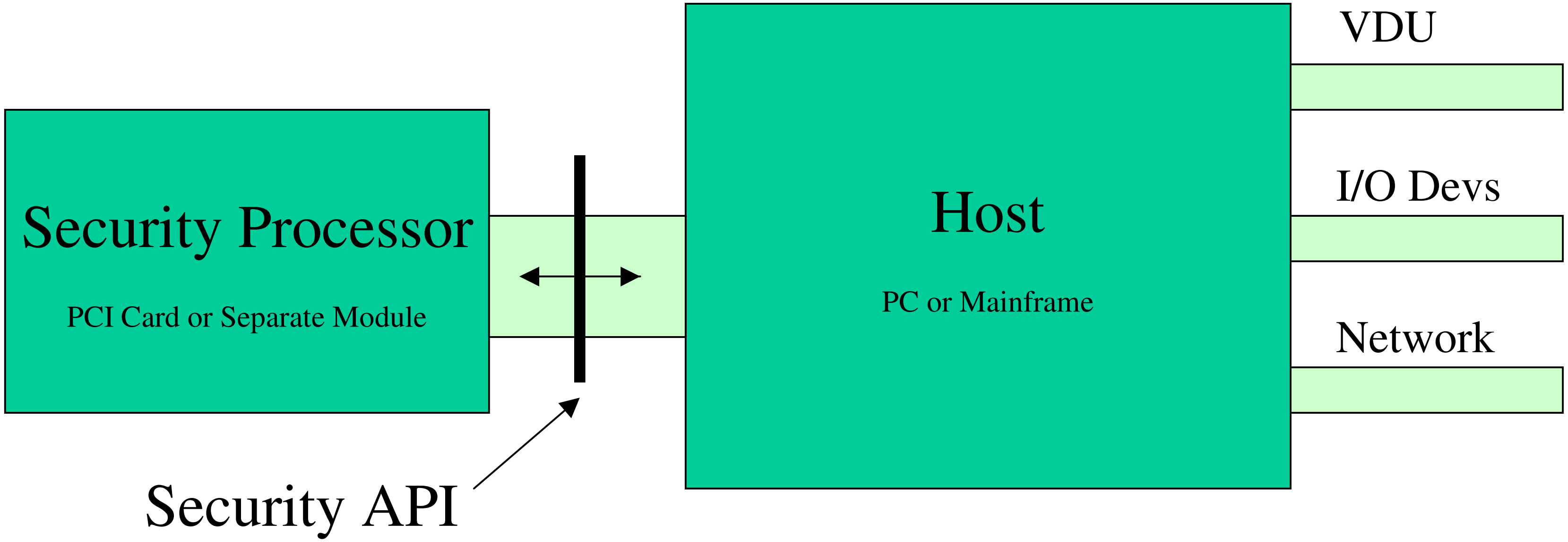
University of Bristol – Information Security Group **4th Nov ‘03**

Summary

- What is a Security API?
- Origins of Security APIs : the Military
- The “killer-app” : Banking Security
  - Introduction to banking security
  - Classic banking security failures
  - New banking security attacks
  - Lessons learned
- The “Digital Battlefield”
- Conclusions

What is a Security API ?

- A command set that uses cryptography to control processing of and access to sensitive data, according to a certain policy



Example Security API Commands

U->C : $\{ A \}_{KM}$, $\{ B \}_{KM}$

C->U : $\{ A+B \}_{KM}$

U->C : GUESS , $\{ ANS \}_{KM}$

C->U : YES (if GUESS=ANS else NO)

U->C : $\{ X \}_{K1}$, $\{ K1 \}_{KM}$, $\{ K2 \}_{KM}$

C->U : $\{ X \}_{K2}$

Research into API Attacks

- Some work in early 90’s using prolog style search to find attacks, but few documented atttacks
- Work started in 2000 at University of Cambridge with analysis of hardware security modules used in banks to protect PINs for ATMs
- New work found many more attacks, and produced first significant catalogue of API failures
- Scope has been broadened to include security modules used by certification authorities and also general purpose crypto libraries (eg MSCAPI, PKCS#11)
- Latest work revisiting financial APIs examining PIN generation and verification procedures

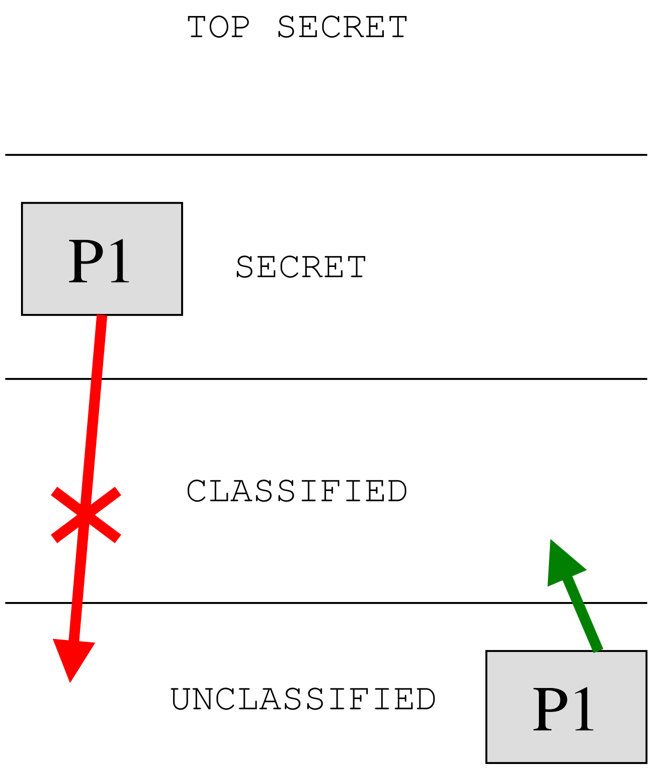
Origins of Security APIs: Military Security

Two threads…

- Tamper-resistant Control Devices
  - gives us notion of a “Hardware Security Module”
  - Provides a well defined boundary at which the API is presented
  - Provides concepts of authorisation and dual control
- Multi-Level Secure Operating Systems
  - provided sophisticated information flow policy
  - provided large multi-purpose API
  - used cryptography to maintain confidentiality of classified data

Multi-Level Security

- Information flow security, as formalised by Bell-LaPadula
  - Golden rules: No read up, No write down
- In practice, the OS system calls can be viewed as a security API enforcing this policy
  - API commands to create processes, change security tags, declassify etc.

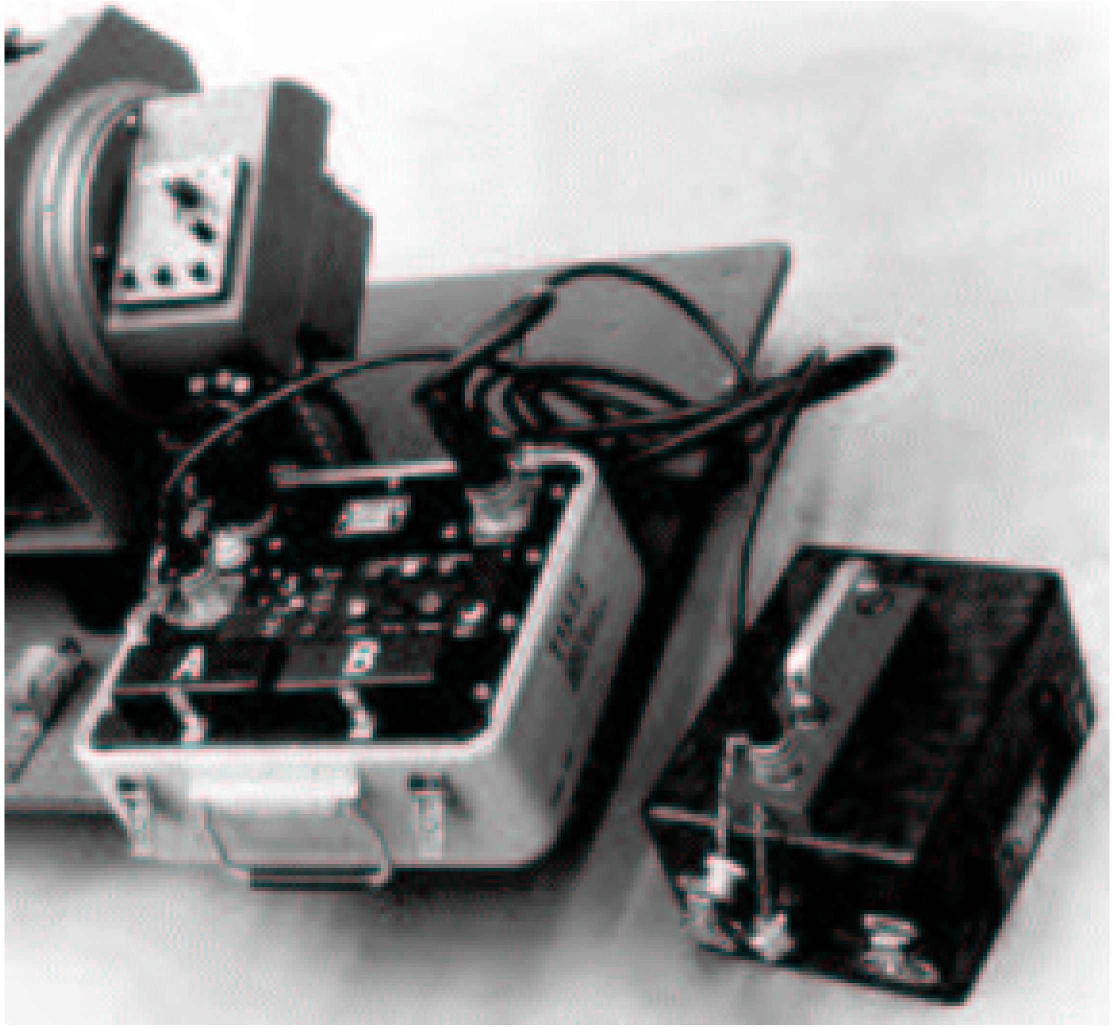


- Getting the OS bug-free and avoiding covert channels turned out to be the biggest problems. Were there any weaknesses in the APIs?

Nuclear Command and Control

- After Cuban missile crisis, all US nuclear ordinance had to be got under “positive control”
- ‘PAL’s – Permissive Action Links
- ‘PACS’ – Permissive Action Control System
- Very simple API: control systems would only arm the weapon upon presentation of a code
- Dual control / “split knowledge” policies used at command nodes
- Main worry became bypass of authorisation system – solution: tamper detecting membranes would trigger (non-nuclear) explosive destruction of warhead, or chemical reactions rendering the plutonium non-fissile.

An Early PAL (c. 1960)

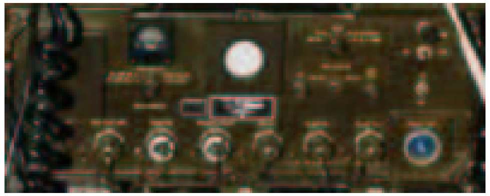


Disassembled Warhead

Today’s Digital Battlefield

- Access control first used for nukes extended
  - Artillery
  - Communications Equipment
  - Nowadays: tactical control systems, tanks, radars, mobile SAM sites
  - Anything which may be captured on battlefield

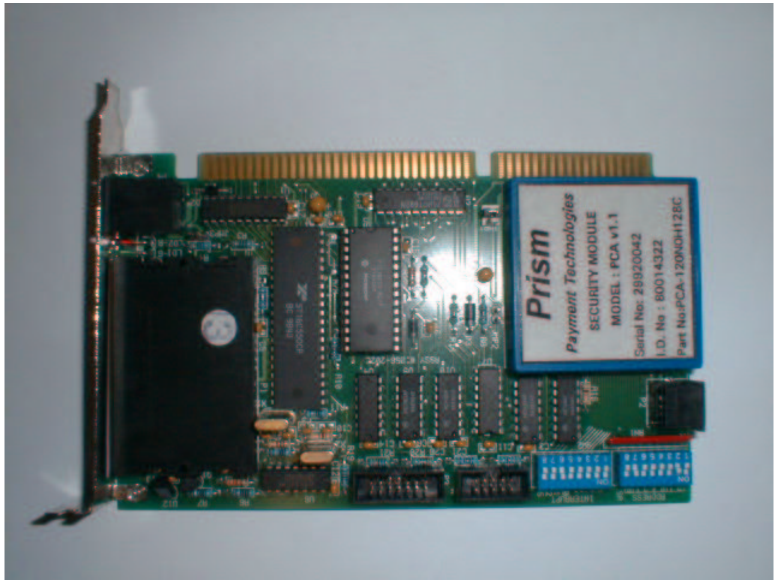
- Other uses of crypto on the battlefield
  - IFF radar systems, Covert radio



Commercial Hardware Security Modules

- Government defence contractors begin to offer similar technology to secure business communications and transactions
- Commercial HSMs drew together the sophisticated API of a secure OS, coupled with tamper-resistance as developed to protect military hardware

Hardware Security Modules



Who Needs Security Modules ?

- Those who need to enforce access policies to sensitive information

  Examples: Granting signing permission at a Certification Authority
  Enforcing split control policies on nuclear weapons & arming codes

- Those who need to protect mission critical sensitive data

  Example: Protecting PIN generation keys at banks

- Those who need to protect data in hostile environments

  Examples: Protecting Token Vending Machines (Electricity, Lottery etc…)
  Protecting communications keys in battlefield radios

- Those with high crypto throughput requirements

  Example: SSL acceleration for webservers

Studying APIs : Financial Security

- Concrete and simple security policy for APIs

  "Only the customer should know her PIN."

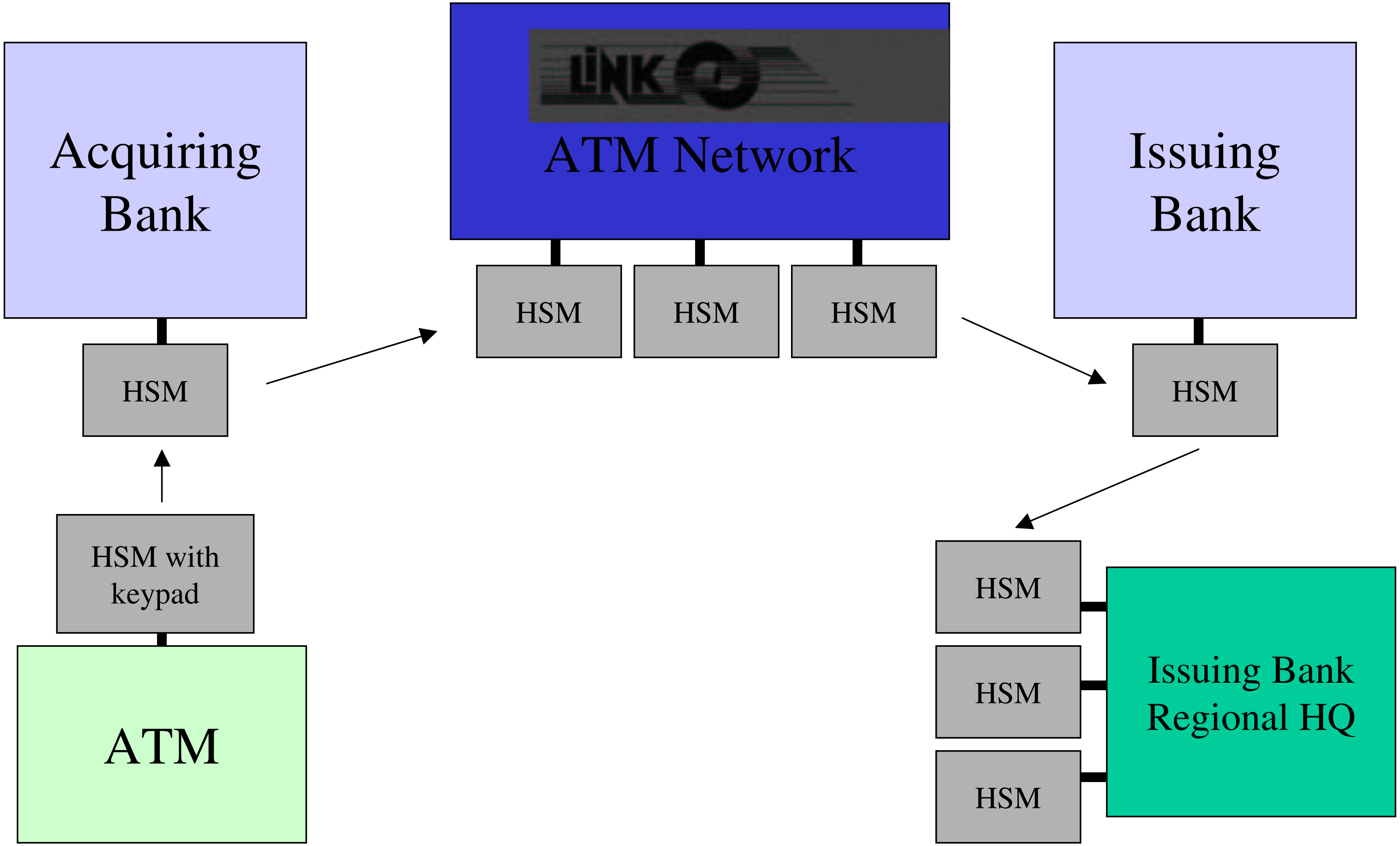
  "Keys protecting PINs may only be manipulated when authorised by two different employees."

- API manuals are often publicly available
  - IBM put 4758 CCA manual on its website
  - Diversity: many manufacturers have APIs performing same broad functionality – good for comparison
- ATM security was the "killer-app" that brought cryptography into the commercial mainstream – so long history of financial API development

Introduction to ATM Security

- The crucial secret is the customer PIN. The customer should be the only person that knows the value of this PIN
- PINs need to be protected from malicious insiders and outsiders
- PINs must be protected when generated, in storage, when issued to customers, when travelling via the international ATM network, and when being verified
- To this end, banks use Hardware Security Modules (HSMs) to perform cryptography and implement a policy which prevents both insiders and outsiders from gaining unauthorised access to PINS.

Security Modules in Banks



How are PINs Generated ?

Start with your bank account number (PAN)

```
5641 8203 3428 2218
```

Encrypt with **PIN Derivation Key**
(aka **PMK** – Pin Master Key)

```
22BD 4677 F1FF 34AC
```

decimalise

Chop off the
End

```
2213                 (B->1)
                     (D->3)
```

What's a Decimalisation Table ?

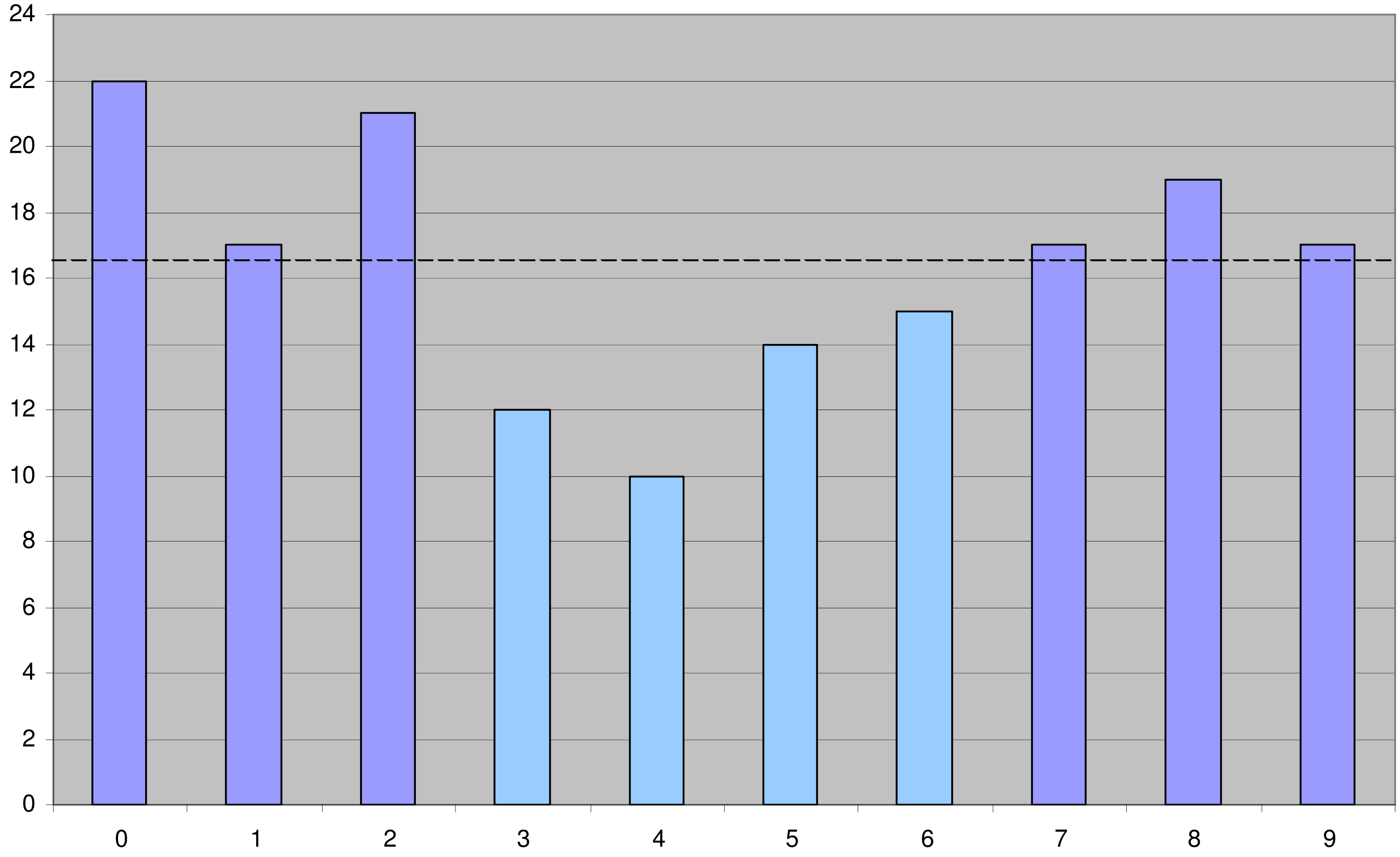
- Remember encrypted result was in hexadecimal?
- Encryption produces output that looks uniformly distributed, so `0-F` are all equally likely
- Decimalisation Table used to map `0-F` back to `0-9`

digit in	0123456789ABCDEF
digit out	0123456789012345

```
e.g. 22BD -> 2213
```

- Because some numbers have several hexadecimal digits mapped to them, they are more likely to occur in issued PINs than others

Example Distribution : HSBC



(Sample size: 45 people)

XOR to Null Key Attack

- Top-level crypto keys exchanged between banks in several parts carried by separate couriers, which are recombined using the exclusive-OR function
- A single operator could feed in the same part twice, which cancels out to produce an ‘all zeroes’ test key. PINs could be extracted in the clear using this key

U->C : $\{KP1\}_{KM}$, $\{KP2\}_{KM}$

C->U : $\{KP1 \text{ xor } KP2\}_{KM}$

U->C : $\{KP1\}_{KM}$, $\{KP1\}_{KM}$

C->U : $\{KP1 \text{ xor } KP1\}_{KM}$ (= $\{0\}_{KM}$)

(Anderson 2000)

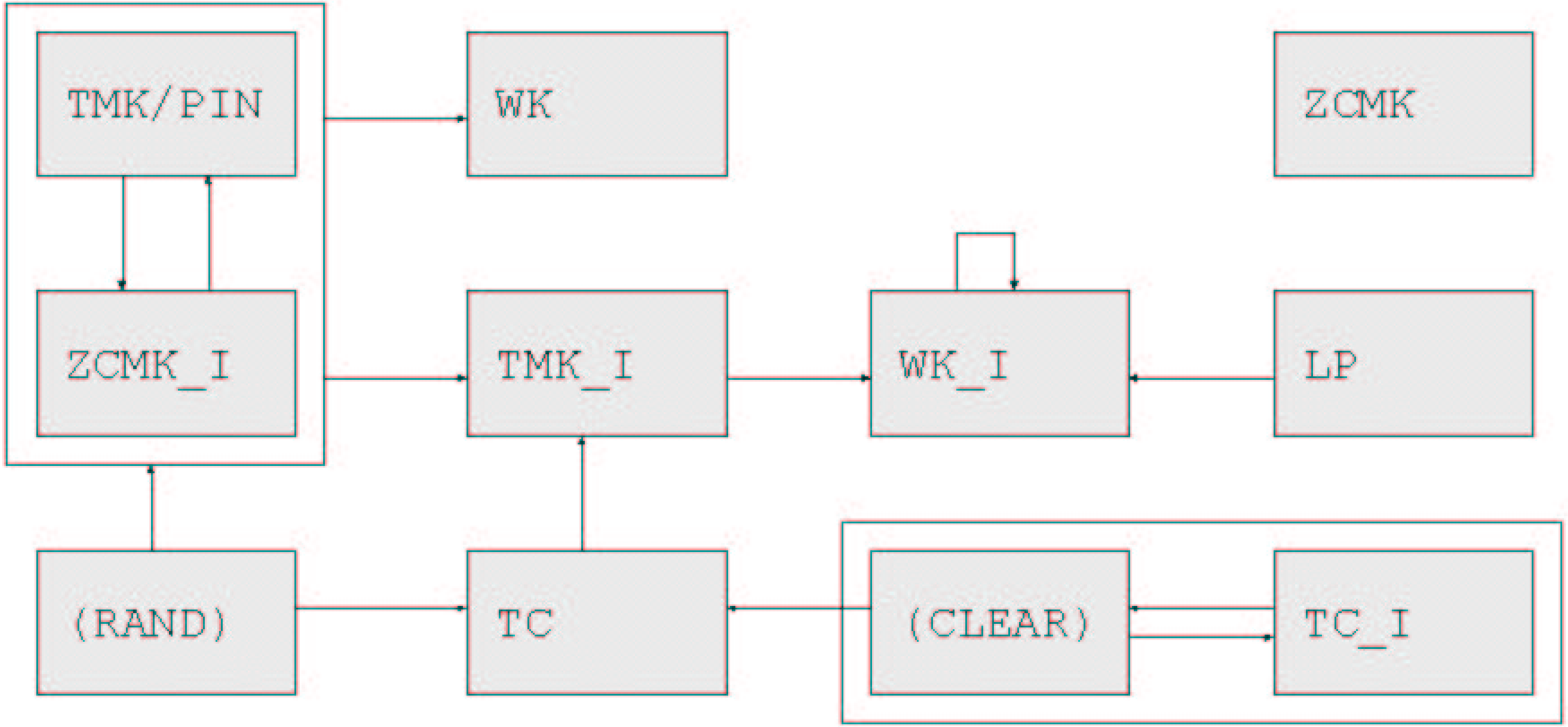
VSM Type System Attack

- Encrypting communication keys for transfer to an ATM used exactly the same process as calculating a customer PIN
- Customer PINs could be generated by re-labelling an account number as a communications key, and using the same encryption process

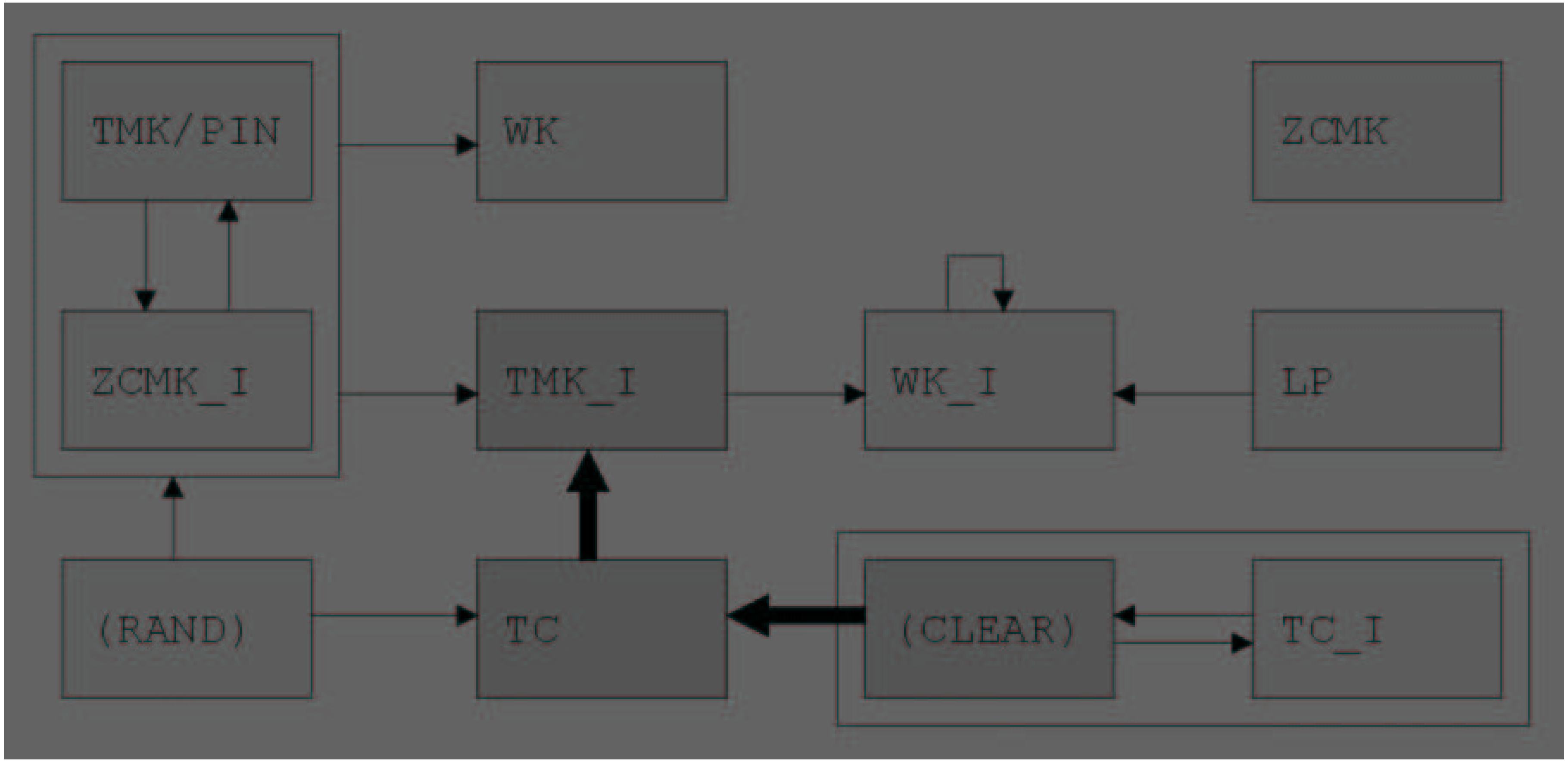
(Bond 2000)

The Visa Security Module

VSM Type Diagram



VSM Type System Attack



Type System Attack (Protocol Notation)

U->C : 5641 8203 3428 2218

C->U : $\{5641\ 8203\ 3428\ 2218\}_{\mathbf{TC}}$

U->C : $\{5641\ 8203\ 3428\ 2218\}_{\mathbf{TC}}$, $\{\ \mathbf{PMK}\ \}_{\mathbf{TMK}}$

C->U : $\{5641\ 8203\ 3428\ 2218\}_{\mathbf{PMK}}$

$\{5641\ 8203\ 3428\ 2218\}_{\mathbf{PMK}}$ **=** 22BD 4677 F1FF 34AC

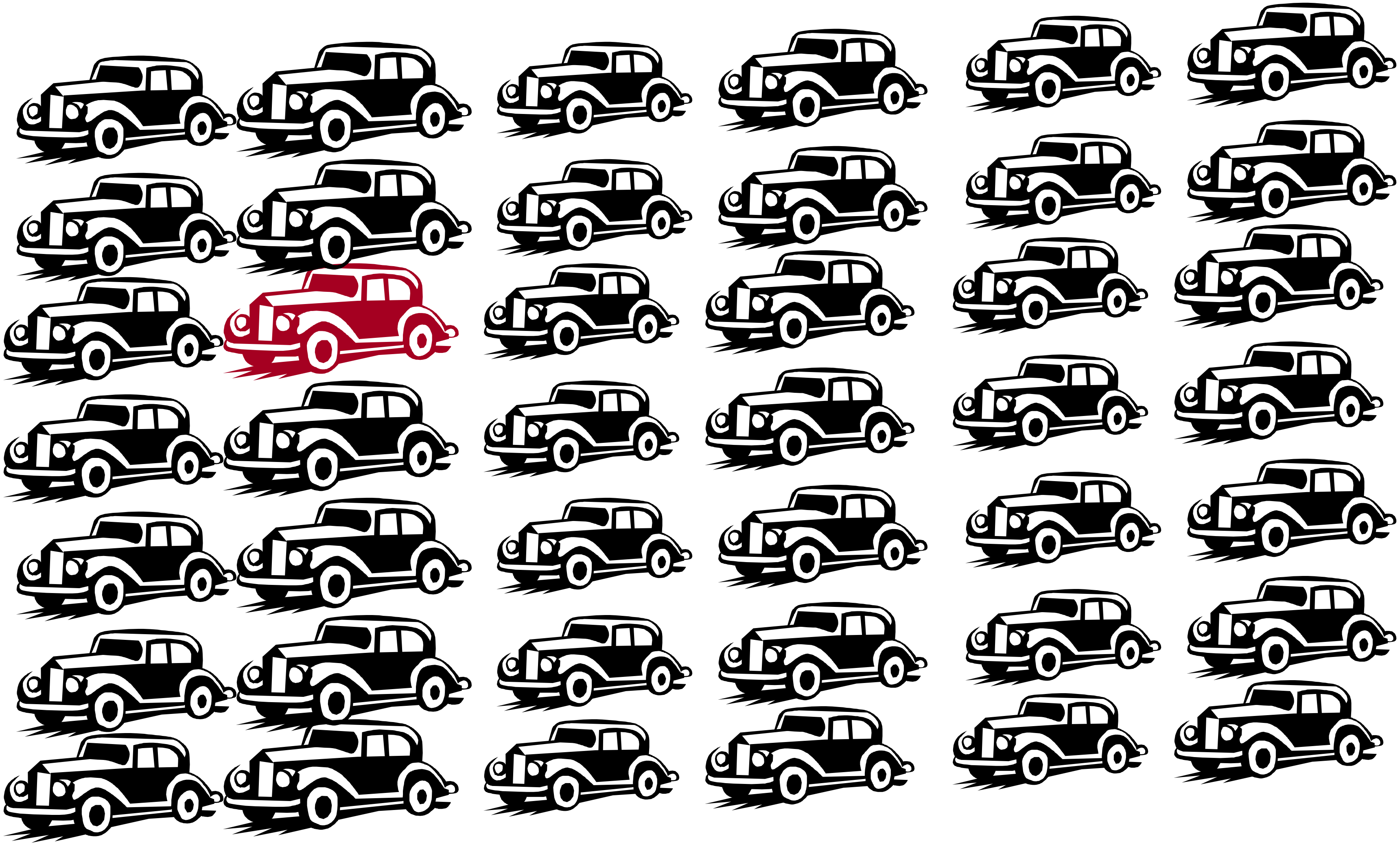
So customer PIN is 22BD i.e. 2213

Car Park Analogy

- A thief walks into a car park and tries to steal a car...

- How many keys must he try?

Car Park Analogy 1900



Car Park Analogy 2000

The Meet in the Middle Attack

- Common sense statistics
- Attack multiple keys in parallel
- Need the same plaintext under each key
- Encrypt this plaintext to get a ‘test vector’
- Typical case: A 2^{56} search for one key becomes a 2^{40} search for 2^{16} keys
- Poor implementations of 3DES key storage allow 3DES key halves to be attacked individually

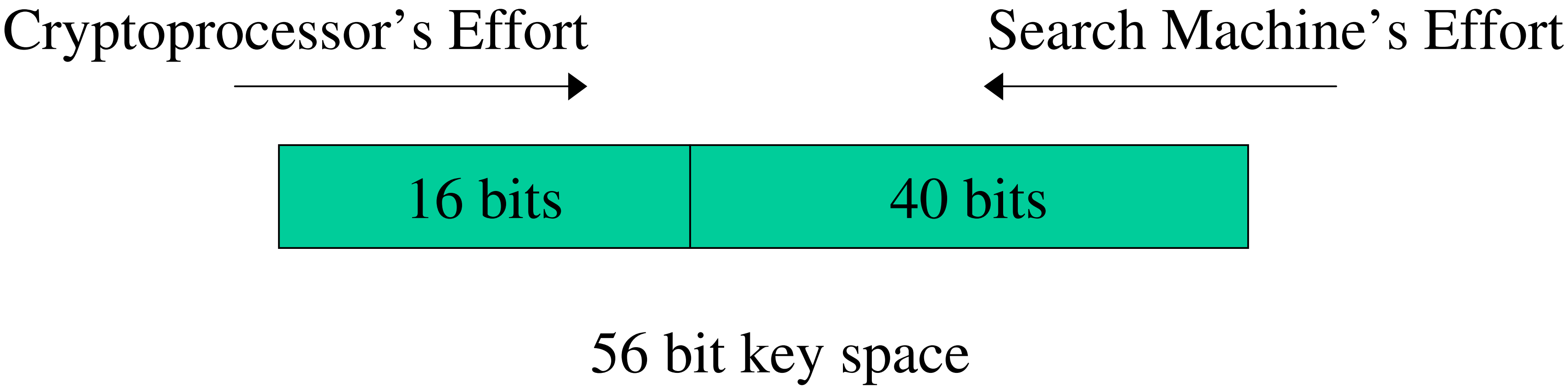
MIM Attack on DES Security Modules

- Generate 2^{16} keys
- Encrypt test vectors

$U\text{->}C : \{ \mathtt{KEY1} \}_{KM}$

$C\text{->}U : \{ \mathtt{0000000000000000} \}_{KEY1}$

- Do 2^{40} search



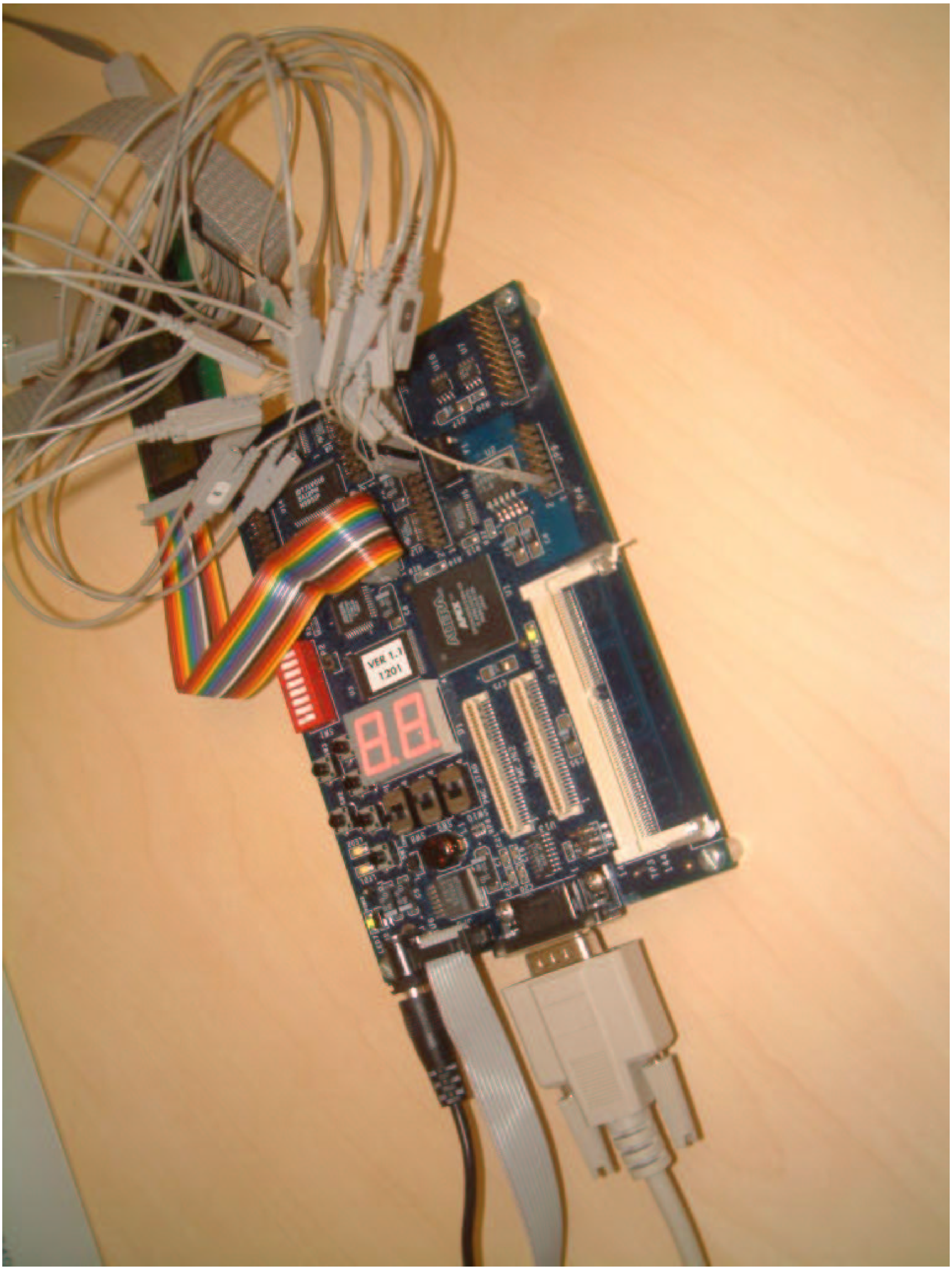
Cryptoprocessor's Effort

Search Machine's Effort

16 bits

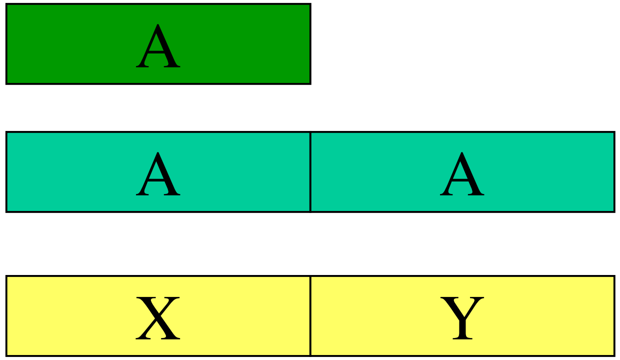
40 bits

56 bit key space


VER 1.1
1201

MIM Attack on Triple-DES HSMs

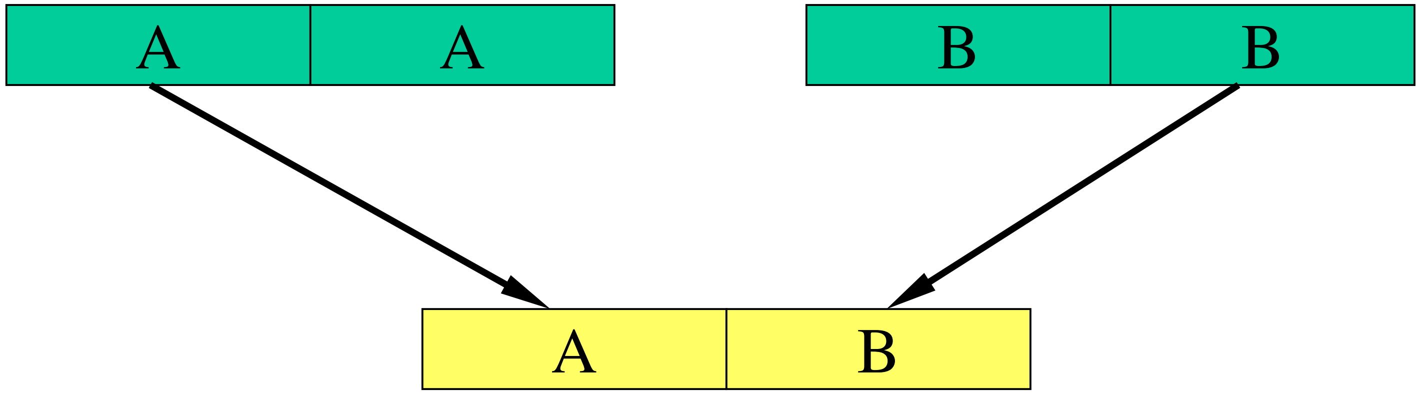
$$E_K(D_K(E_K(\text{ KEY }) = E_K(\text{KEY})$$



Single Length Key

Double Length “Replicate”

Double Length



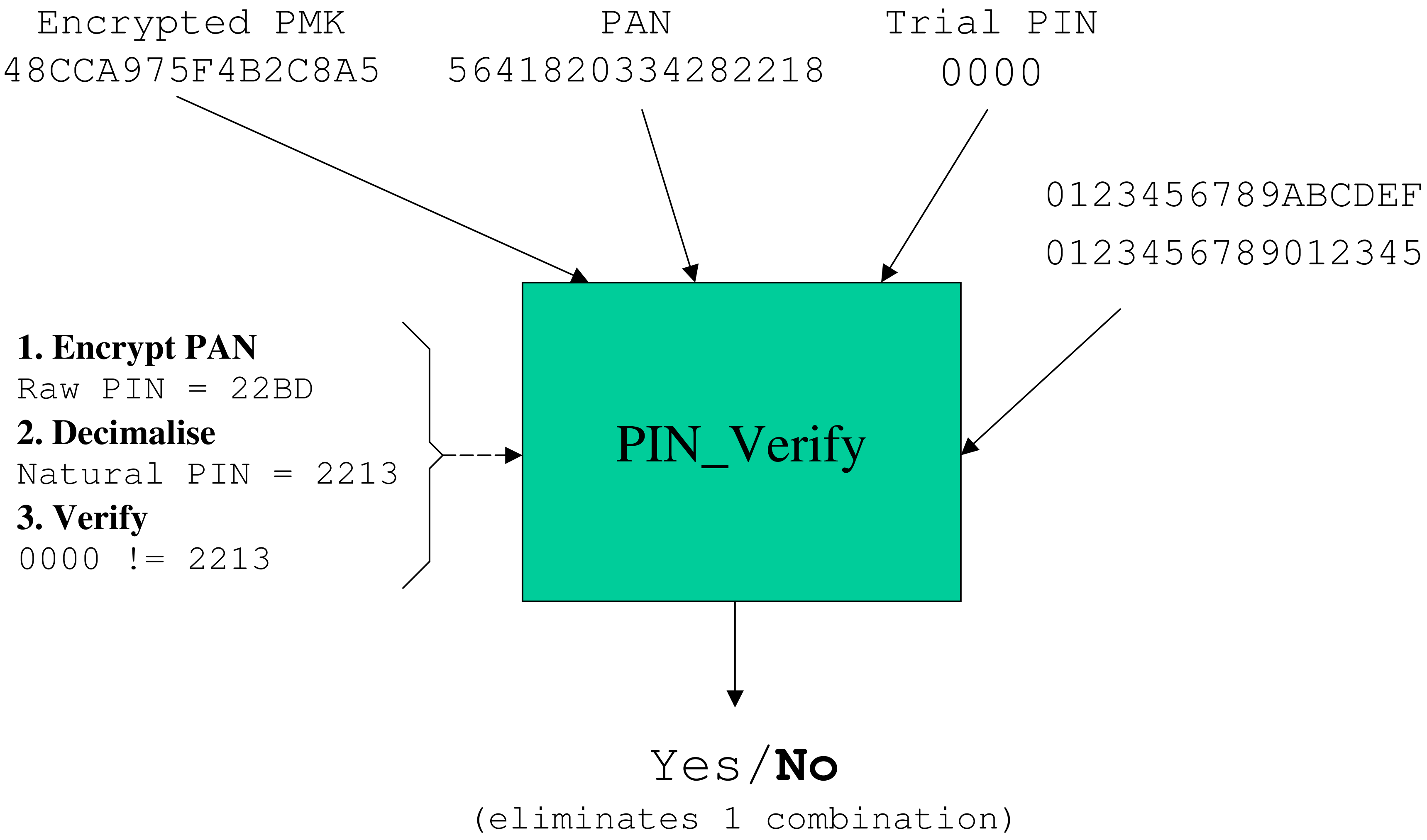
Decimalisation Table Attack

- Remember PINs derived from account numbers
- Hexadecimal raw PIN is converted to decimal using decimalisation table
- Most APIs allow the decimalisation table to be specified with each PIN verification command
- A normal verification command eliminates one of 10,000 combinations of PIN for the attacker.
- If the table is altered, whether or not the alteration affects correct verification leaks much more information about the PIN

examples…

(Bond/Clulow 2002)

Decimalisation Table Attack (1)



Encrypted PMK
48CCA975F4B2C8A5

PAN
5641820334282218

Trial PIN
0000

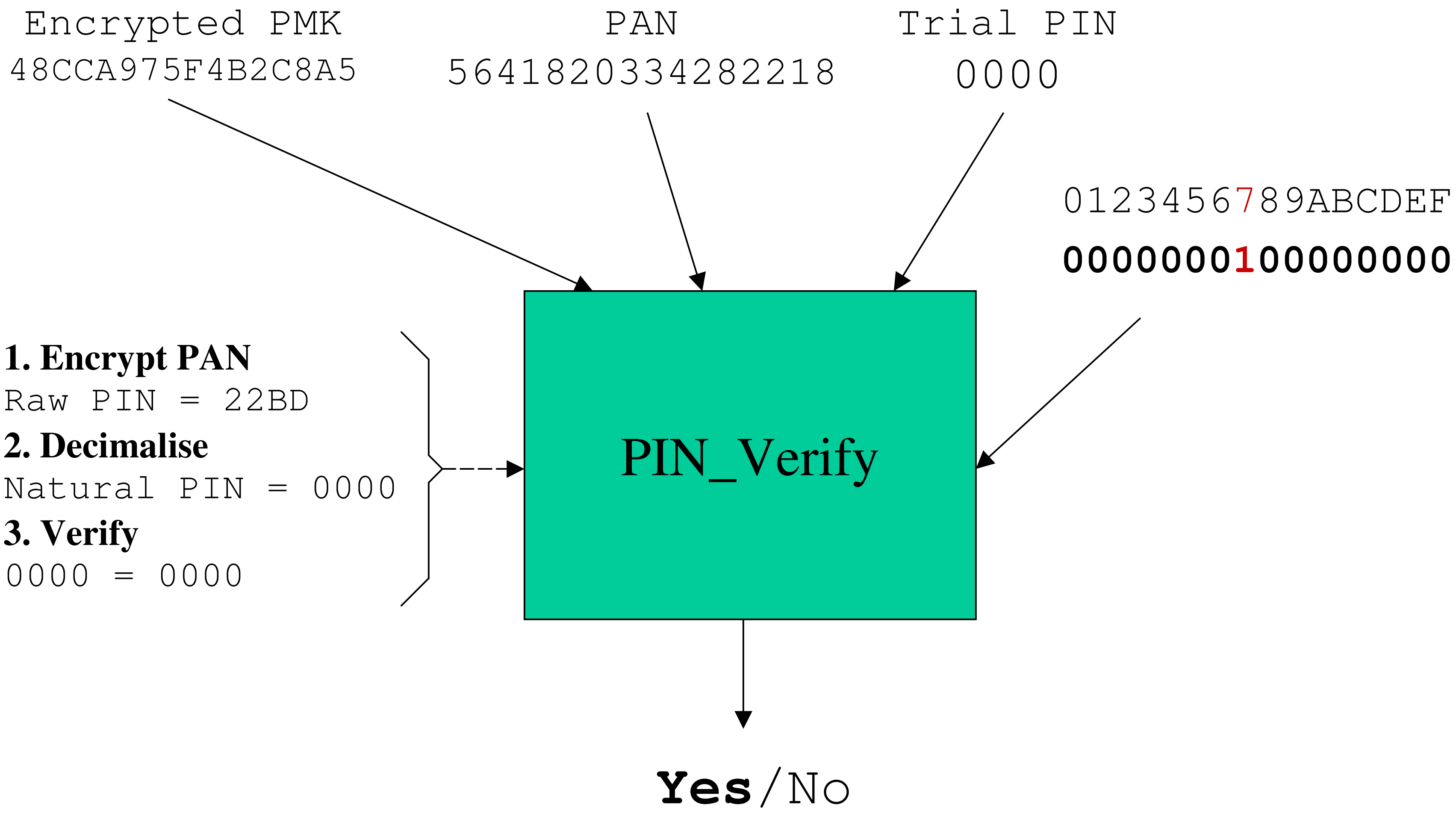
0123456789ABCDEF
0123456789012345

1. Encrypt PAN
Raw PIN = 22BD
2. Decimalise
Natural PIN = 2213
3. Verify
0000 != 2213

PIN_Verify

Yes/**No**
(eliminates 1 combination)

Decimalisation Table Attack (2)



Encrypted PMK
48CCA975F4B2C8A5

PAN
5641820334282218

Trial PIN
0000

0123456789ABCDEF
0000000100000000

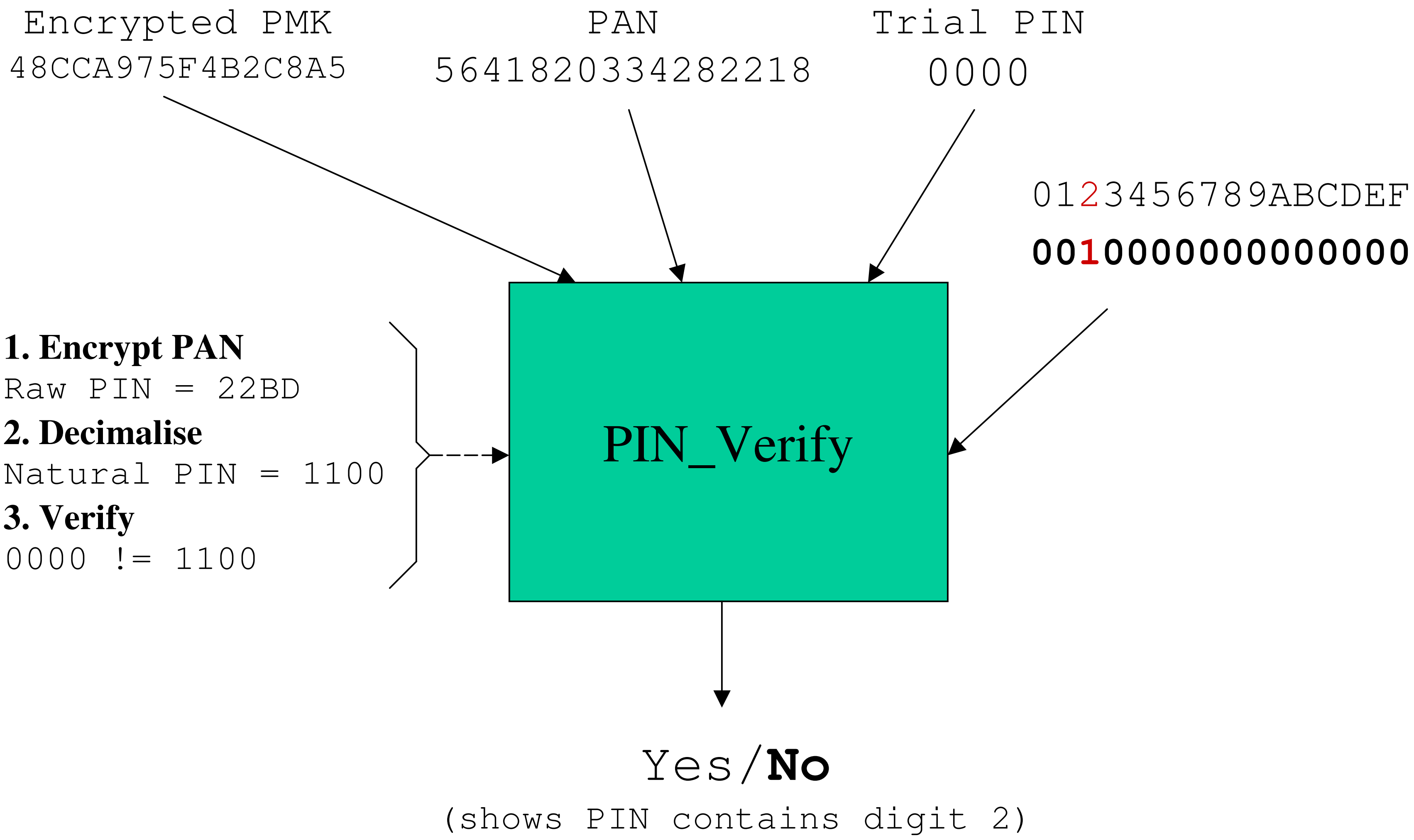
1. Encrypt PAN
Raw PIN = 22BD
2. Decimalise
Natural PIN = 0000
3. Verify
0000 = 0000

PIN_Verify

Yes/No

(eliminates all PINs containing digit 7)

Decimalisation Table Attack (3)



Encrypted PMK
48CCA975F4B2C8A5

PAN
5641820334282218

Trial PIN
0000

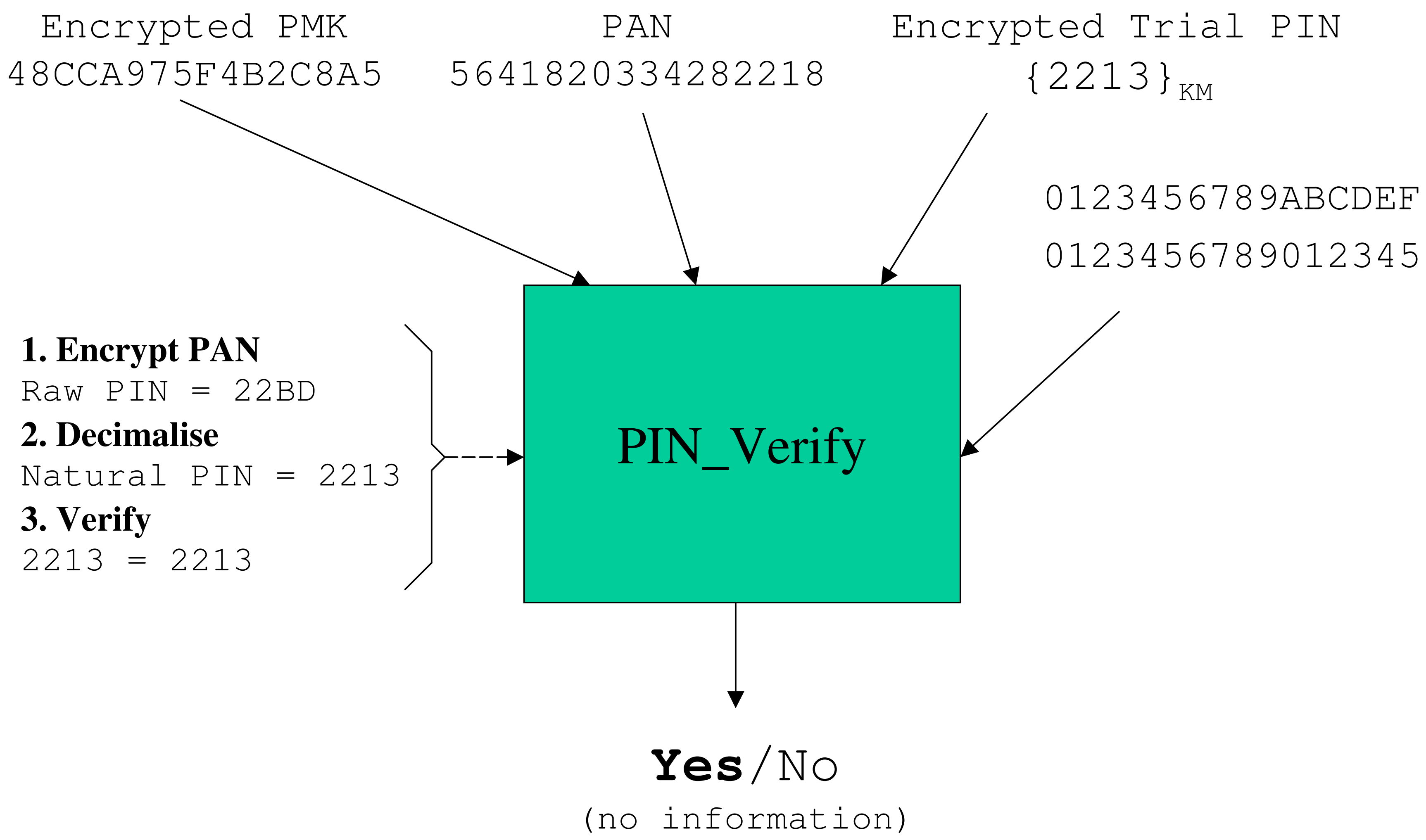
0123456789ABCDEF
0010000000000000

1. Encrypt PAN
Raw PIN = 22BD
2. Decimalise
Natural PIN = 1100
3. Verify
0000 != 1100

PIN_Verify

Yes/**No**
(shows PIN contains digit 2)

Decimalisation Table Attack (4)



Encrypted PMK
48CCA975F4B2C8A5

PAN
5641820334282218

Encrypted Trial PIN
$\{2213\}_{KM}$

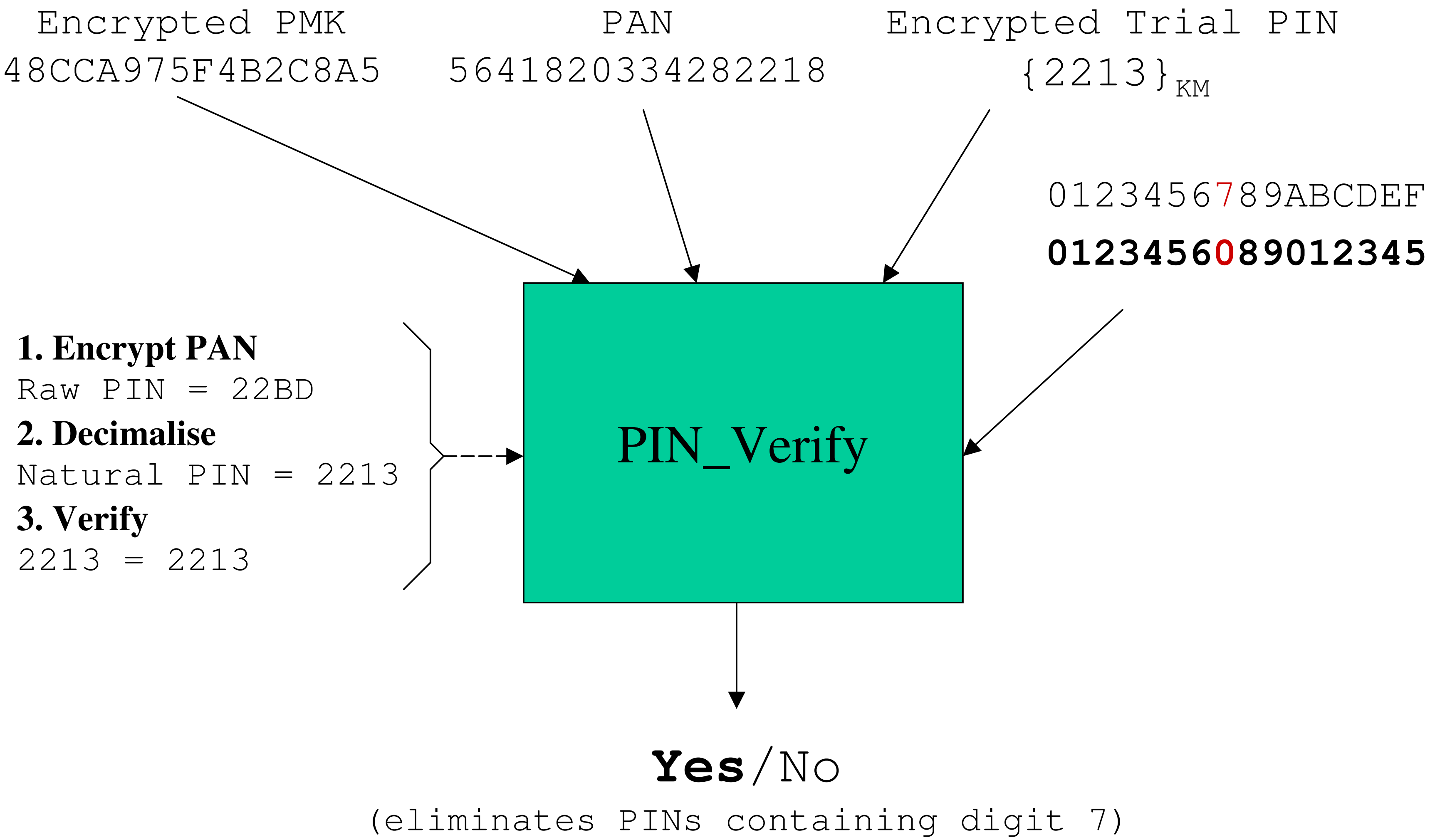
0123456789ABCDEF
0123456789012345

1. Encrypt PAN
Raw PIN = 22BD
2. Decimalise
Natural PIN = 2213
3. Verify
2213 = 2213

PIN_Verify

Yes/No
(no information)

Decimalisation Table Attack (5)



Encrypted PMK
48CCA975F4B2C8A5

PAN
5641820334282218

Encrypted Trial PIN
$\{2213\}_{KM}$

0123456789ABCDEF

0123456089012345

1. Encrypt PAN
Raw PIN = 22BD
2. Decimalise
Natural PIN = 2213
3. Verify
2213 = 2213

PIN_Verify

Yes/No
(eliminates PINs containing digit 7)

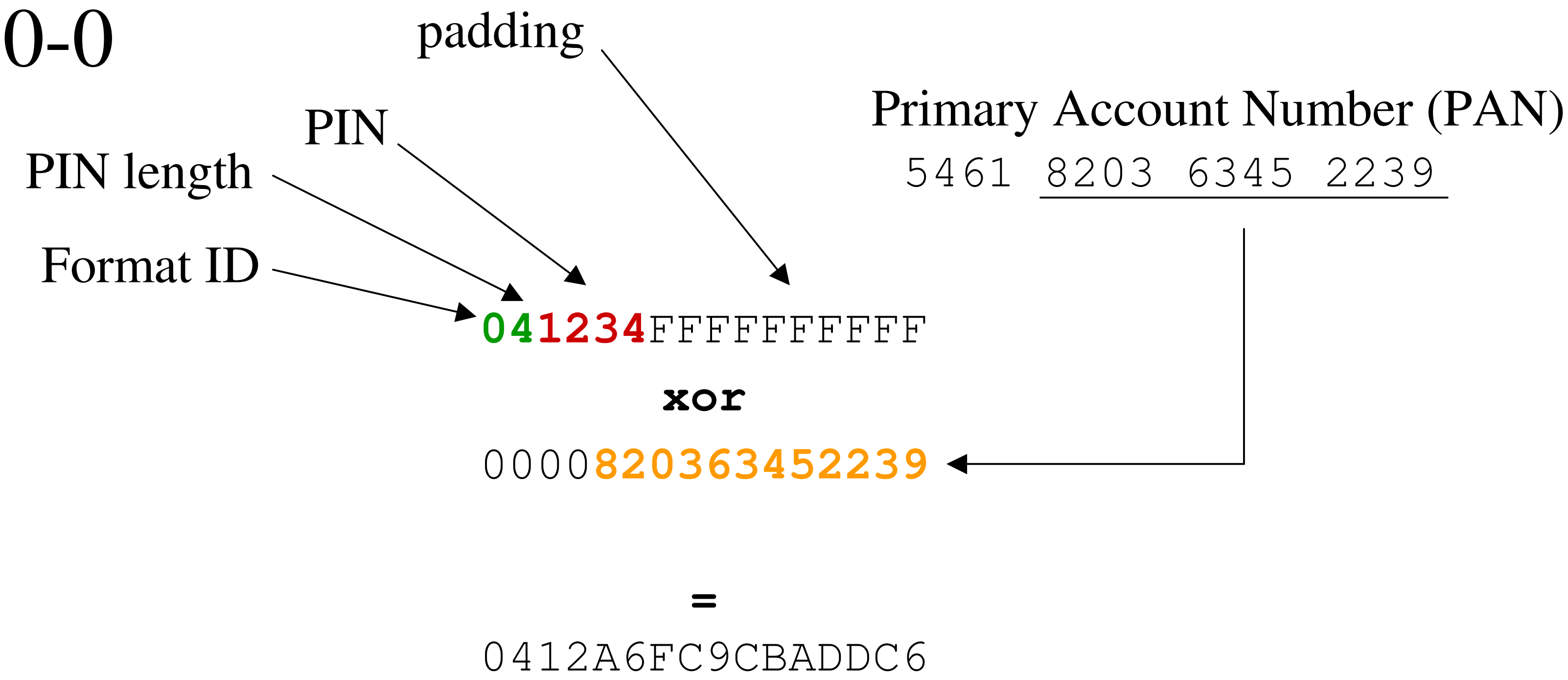
PAN Modification Attack (1)

- Encrypted PINs transferred from ATM to issuing bank via ATM network using point to point encryption
- At each node PIN block must be decrypted with incoming key, and re-encrypted with outgoing key
- Common ISO standard “binds” PIN to particular customer by exclusive-ORing PAN with PIN before encryption
- Attack: specifying incorrect PAN may make deduced PIN contain hexadecimal digit ‘A’-’F’, which causes formatting error. Conditions under which formatting error arises leaks information about PIN.

(Clulow 2002)

PIN Block Formats

IS0-0



padding

PIN

PIN length

Format ID

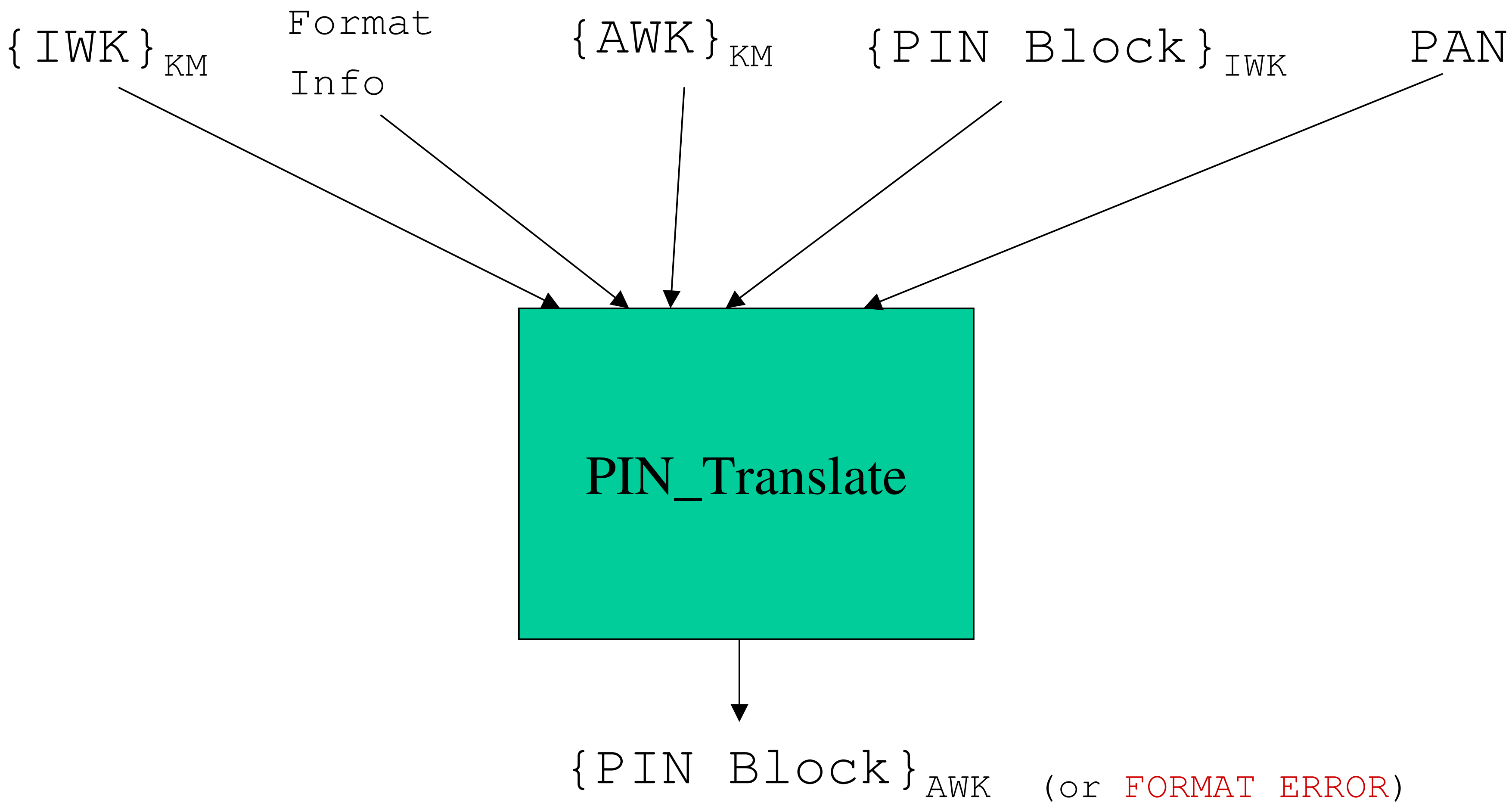
Primary Account Number (PAN)

```
5461 8203 6345 2239

041234FFFFFFFFFF
      xor
0000820363452239

       =
0412A6FC9CBADDC6
```

IS0-2

```
241234FFFFFFFFFF
```

PAN Modification Attack (2)



PAN Modification Attack (3)

041234FFFFFFFFFF

xor

0000820363452239

=

0412B6FC9CBADDC6

construction of PIN block

0412B6FC9CBADDC6

xor

0000820363452239

=

041234FFFFFFFFFF

correct PAN removed

0412B6FC9CBADDC6

xor

0000**7**20363452239

=

0412C4FFFFFFFFFF

modified PAN Removed – PIN contains ‘C’ – **error**

PIN

PAN

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
0	0	1	2	3	4	5	6	7	8	9
1	1	0	3	2	5	4	7	6	9	8
2	2	3	0	1	6	7	4	5	**A**	**B**
3	3	2	1	0	7	6	5	4	**B**	**A**
4	4	5	6	7	0	1	2	3	**C**	**D**
5	5	4	7	6	1	0	3	2	**D**	**C**
6	6	7	4	5	2	3	0	1	**E**	**F**
7	7	6	5	4	3	2	1	0	**F**	**E**
8	8	9	**A**	**B**	**C**	**D**	**E**	**F**	0	1
9	9	8	**B**	**A**	**D**	**C**	**F**	**E**	1	0
A	**A**	**B**	8	9	**E**	**F**	**C**	**D**	2	3
B	**B**	**A**	9	8	**F**	**E**	**D**	**C**	3	2
C	**C**	**D**	**E**	**F**	8	9	**A**	**B**	4	5
D	**D**	**C**	**F**	**E**	9	8	**B**	**A**	5	4
E	**E**	**F**	**C**	**D**	**A**	**B**	8	9	6	7
F	**F**	**E**	**D**	**C**	**B**	**A**	9	8	7	6

Lessons Learned from Banking APIs

- Classic protocol problems (e.g. binding) can hit security APIs hard
- Legacy system support and unnecessary flexibility can undermine security
- Sophisticated attacks are always possible

- Trading standard of the security with cost creates instability – constant attack and defence of new exploits and minimal fixes

“Digital Battlefields”

Question : What do you get if you cross…

- Legislation
  - Against piracy and copyright infringement

    but also…
  - Against anti-competitive behaviour
- New Marketing Models
  - Rental model for software and services
  - Accessory control and subsidised central units
- Trusted Computing
  - Greater control
  - DRM & IRM

Legislation : Legitimised Attack

- Ongoing Microsoft anti-trust case – how much functionality should Microsoft integrate into its dominant OS?
- Lexmark sued SCC for hacking printer cartridge authentication chips, and replicating them to make compatible cartridges. SCC won (but still have to defeat Lexmark's security to achieve compatibility)
- SONY has tried to sue Datel (unauthorised PS2 accessory manufacturers) several times but failed.
- We may see new legislation overriding DMCA protection against reverse-engineering when it is used anti-competitively.

New Marketing Models

- Ever more subsidised main devices, money recuperated from accessories, refills and software
  - accessory revenue stream must be protected
- New payment schemes
  - who has billing relationships with you? *Banks, phone companies, ISPs*
  - who has the DRM and control technology? *Platform manufacturers, OS manufacturers*
- Increased ease of manufacturer lock-in – encrypted file-formats

Accessory Control Examples

- SONY MagicGate chip – only authorised memory cartridges will work in SONY playstations, mp3 players, laptops
- Printer cartridges – only authorised catridges will work; refill impractical
- Mobile phone batteries must be authenticated, for “increased safety”
- Spare parts for cars may soon be authenticated cryptographically, to protect against “substandard manufacturing” (BMW has plans)

- As the functionality and range of services of devices authenticated increases – authentication protocols turn into full blown APIs

Trusted Computing – A double-edged sword

- IRM – Information Rights Management
  - Companies can stop leaks
  - Mafia can keep their records secret
- DRM – Digital Rights Management
- Trusted IO – Enter your ATM PIN at your PC
- Global PKI – All devices potentially indentifiable
- Trusted Anonymity Systems
- Truly Anonymous peer-to-peer systems
- High-availability systems
- Reverse-engineering resistant viruses

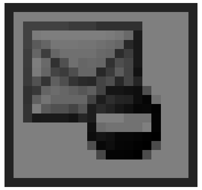
Digital Rights Management

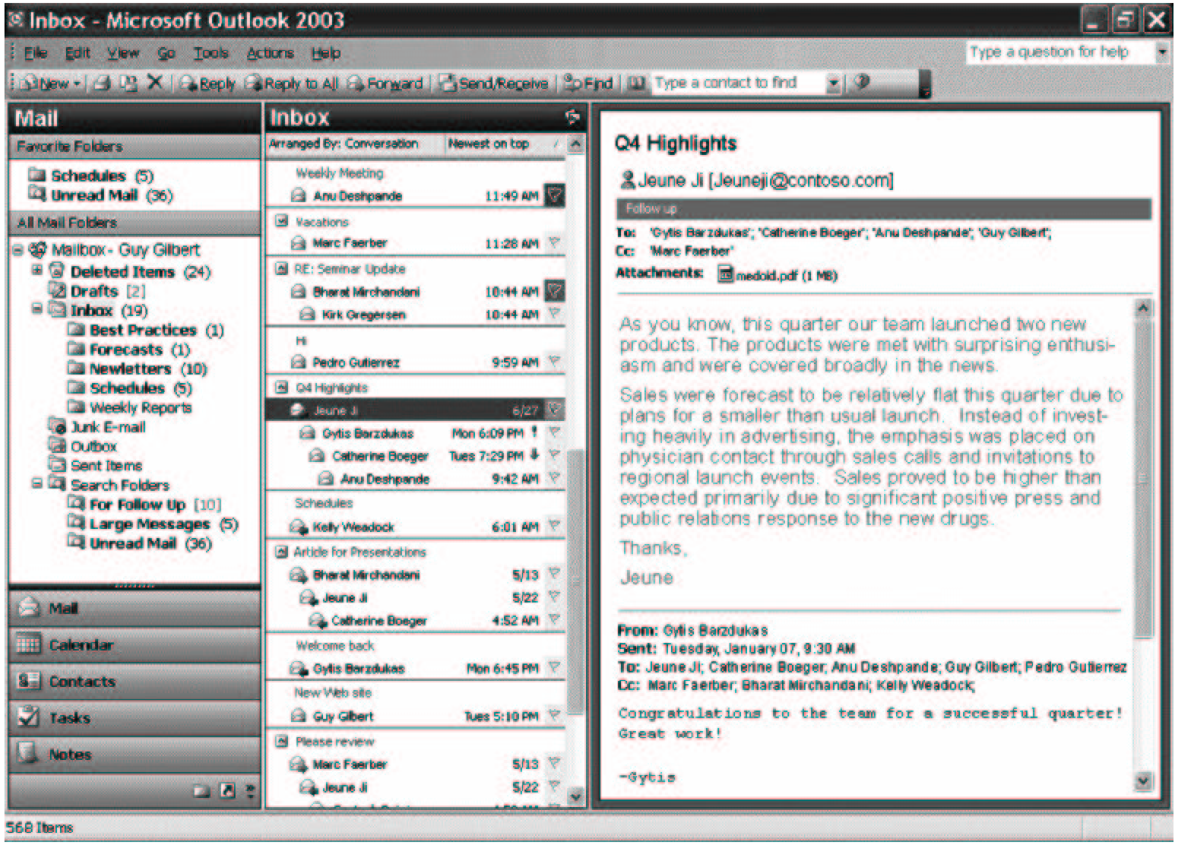
- Nowadays, DRM refers mainly to digital entertainment media
  - DVDs that can’t be ripped, better region control for market segmentation, more sophisticated rental models
  - Control the flow of legitimately downloadable music & video from the internet
  - Mobile phone ringtones
- New terminology “IRM” introduced…

Information Rights Management

- Microsoft Office 2003 with Microsoft Rights Management Server

The “restrict” button





Weekly Meeting
Anu Deshpande 11:49 AM
Vacations
Marc Faerber 11:28 AM
RE: Seminar Update
Bharat Mirchandani 10:44 AM
Kirk Gregersen 10:44 AM
Hi
Pedro Gutierrez 9:59 AM
Q4 Highlights
Jeune Ji 6/27
Gytis Barzdukas Mon 6:09 PM
Catherine Boeger Tues 7:29 PM
Anu Deshpande 9:42 AM
Schedules
Kelly Weadock 6:01 AM
Article for Presentations
Bharat Mirchandani 5/13
Jeune Ji 5/22
Catherine Boeger 4:52 AM
Welcome back
Gytis Barzdukas Mon 6:45 PM
New Web site
Guy Gilbert Tues 5:10 PM
Please review
Marc Faerber 5/13
Jeune Ji 5/22

Q4 Highlights

Jeune Ji [Jeuneji@contoso.com]

Follow up

To: 'Gytis Barzdukas'; 'Catherine Boeger'; 'Anu Deshpande'; 'Guy Gilbert';
Cc: 'Marc Faerber'
Attachments: medoid.pdf (1 MB)

As you know, this quarter our team launched two new products. The products were met with surprising enthusiasm and were covered broadly in the news.

Sales were forecast to be relatively flat this quarter due to plans for a smaller than usual launch. Instead of investing heavily in advertising, the emphasis was placed on physician contact through sales calls and invitations to regional launch events. Sales proved to be higher than expected primarily due to significant positive press and public relations response to the new drugs.

Thanks,

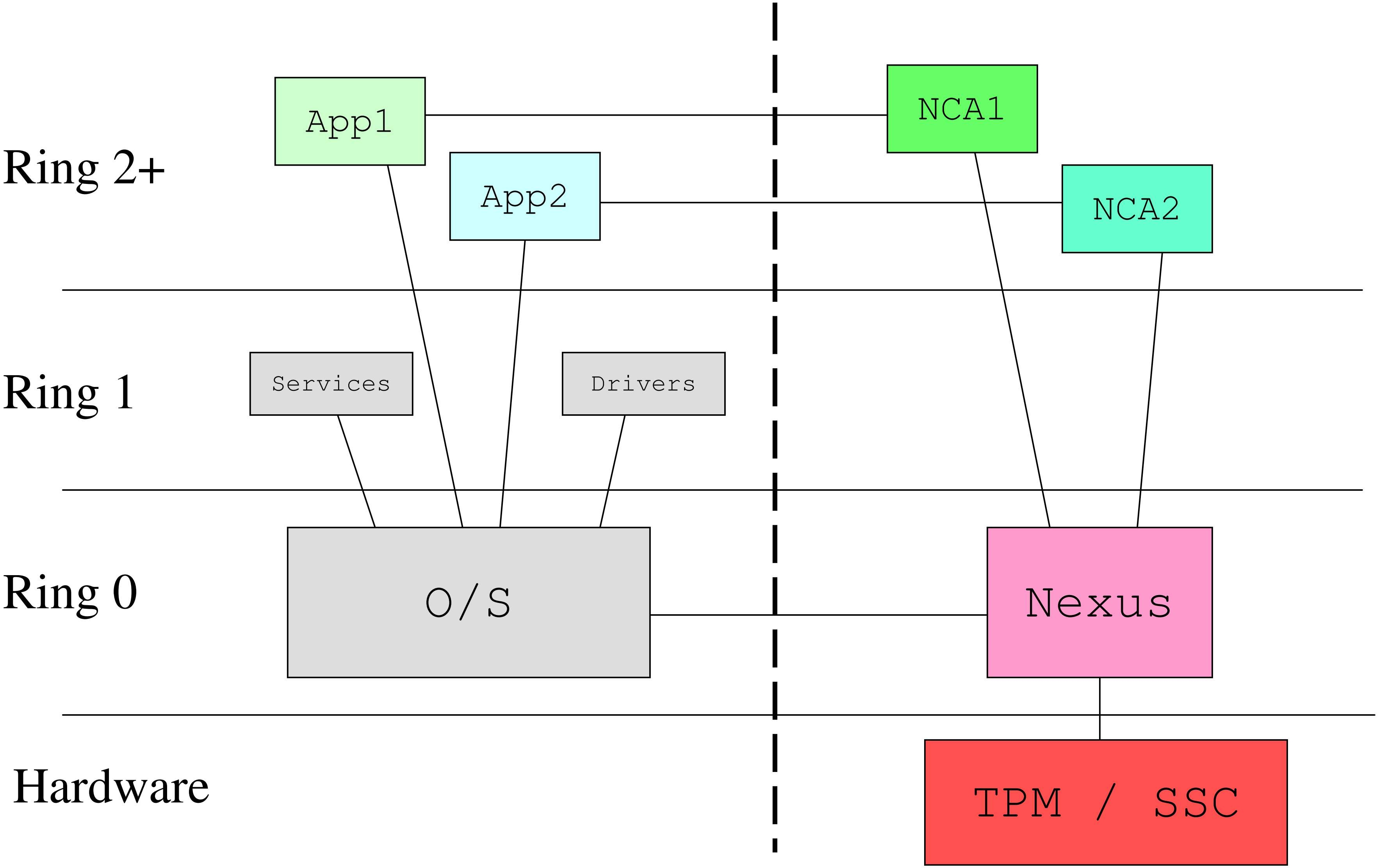
Jeune

From: Gytis Barzdukas
Sent: Tuesday, January 07, 9:30 AM
To: Jeune Ji; Catherine Boeger; Anu Deshpande; Guy Gilbert; Pedro Gutierrez
Cc: Marc Faerber; Bharat Mirchandani; Kelly Weadock;

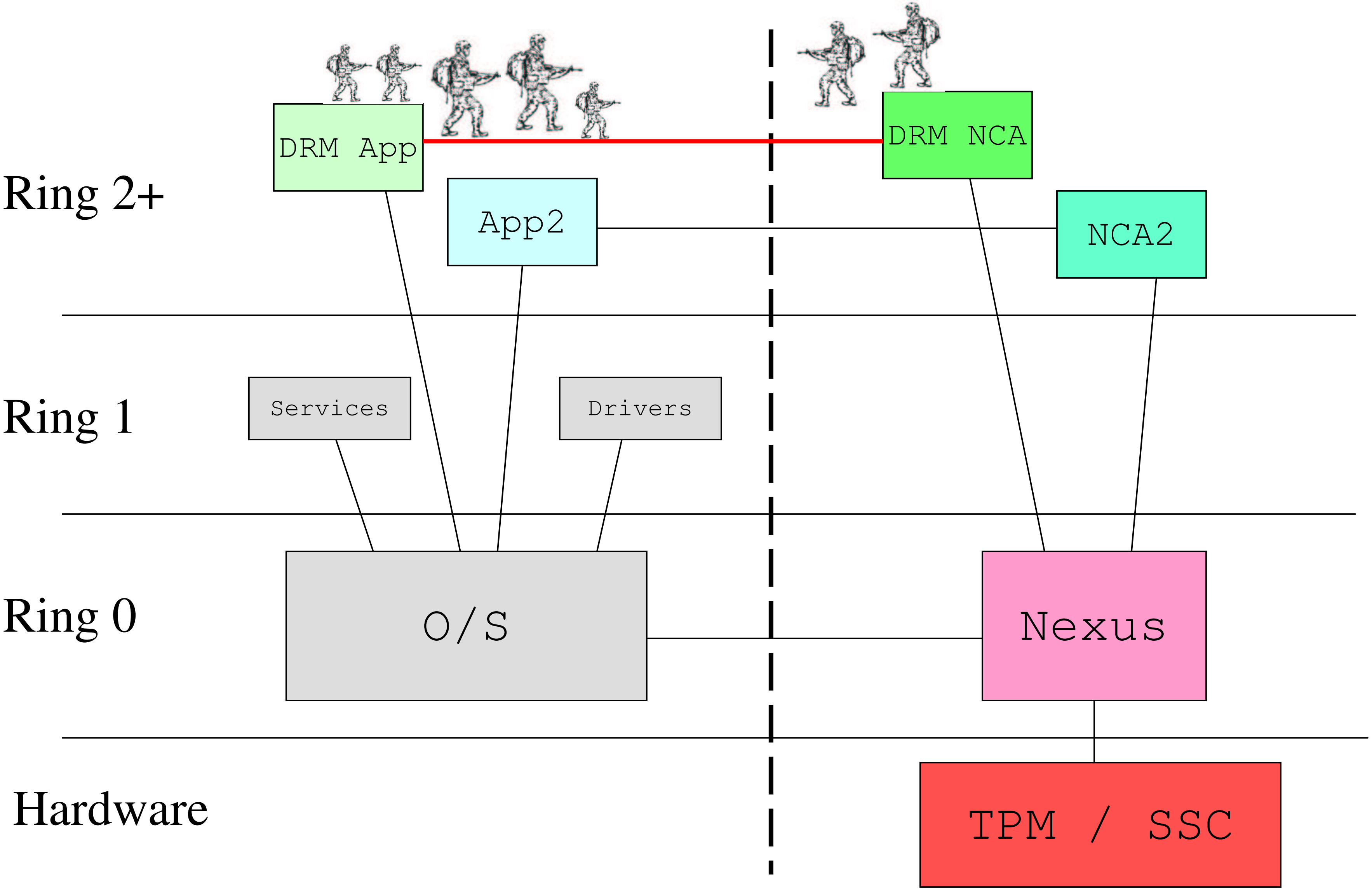
Congratulations to the team for a successful quarter!
Great work!

-Gytis

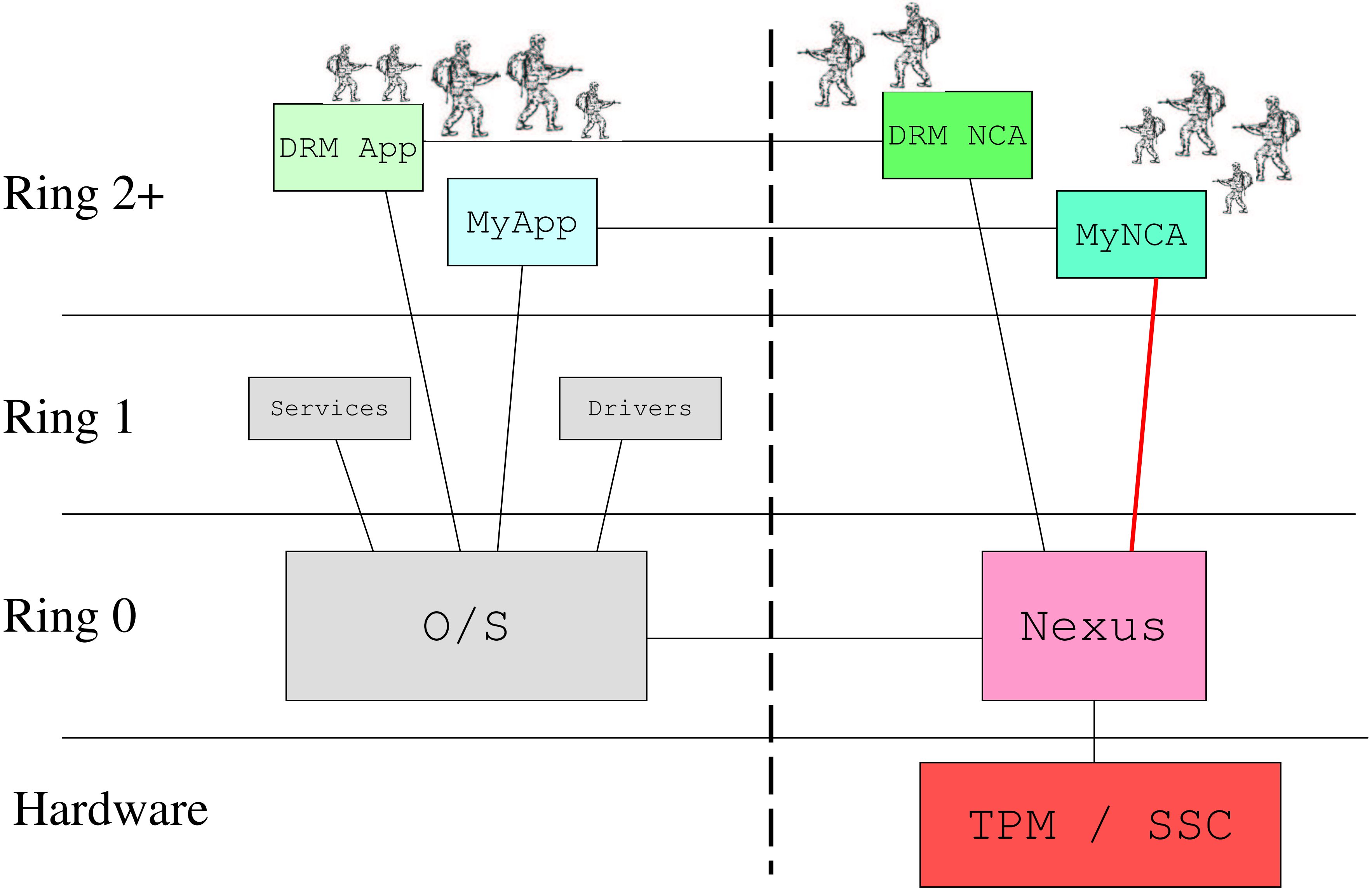
Palladium (NGSCB) Architecture



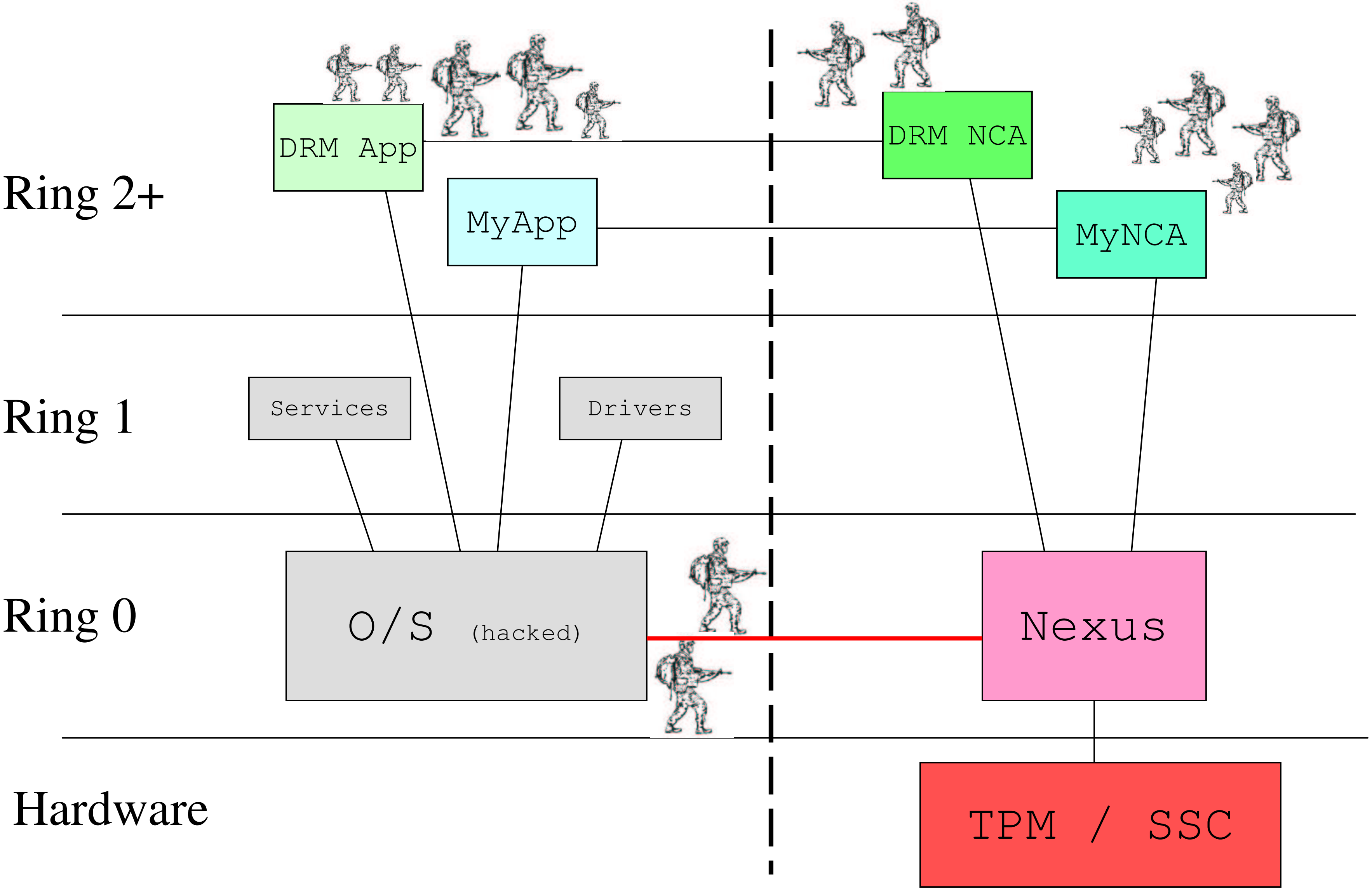
Palladium (NGSCB) Architecture



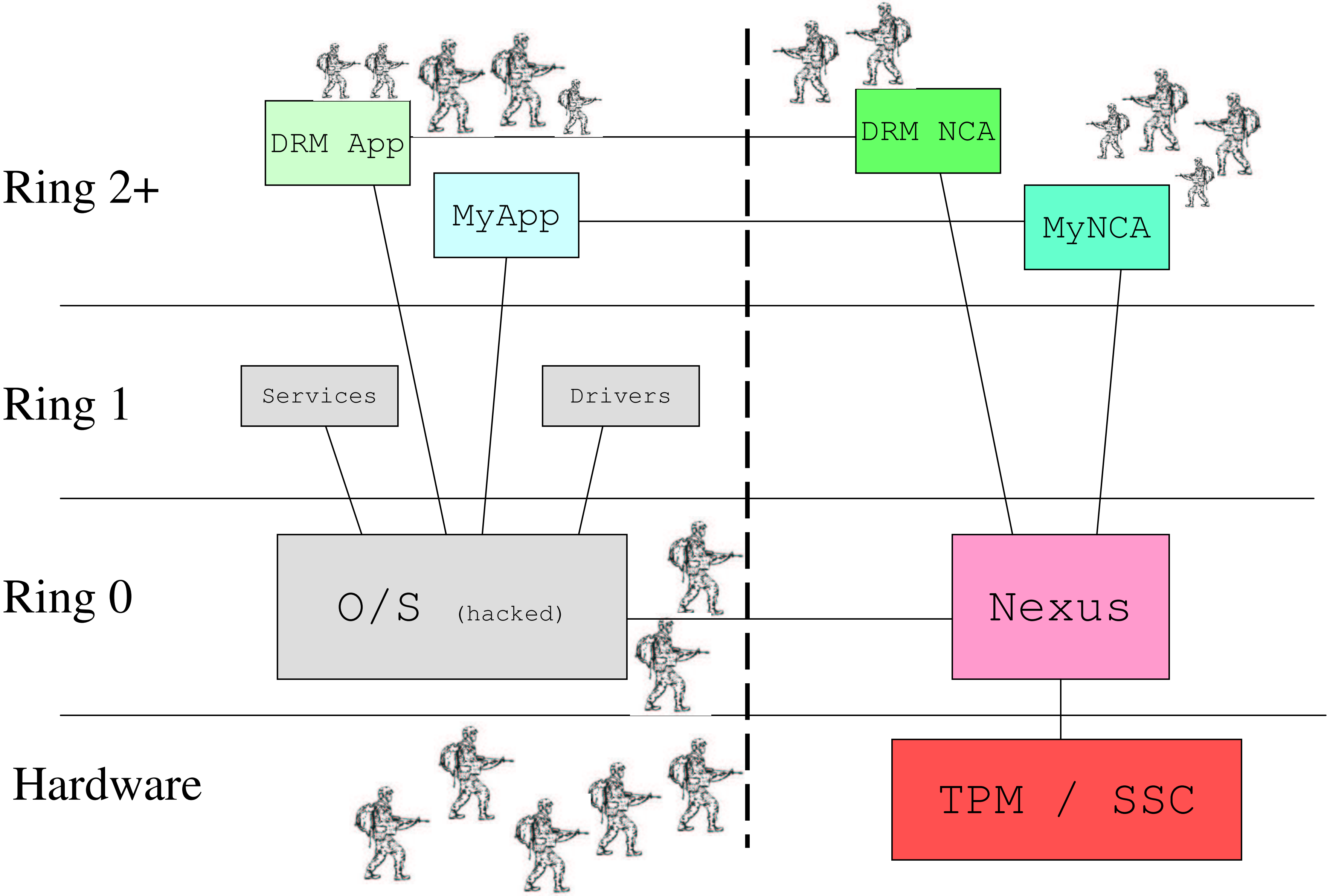
Palladium (NGSCB) Architecture



Palladium (NGSCB) Architecture



Palladium (NGSCB) Architecture

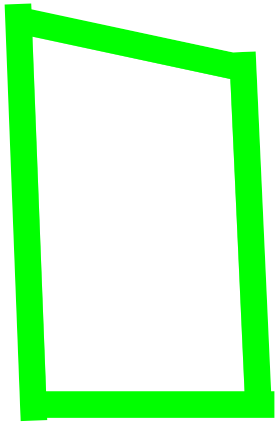


The Battlefield Expanded

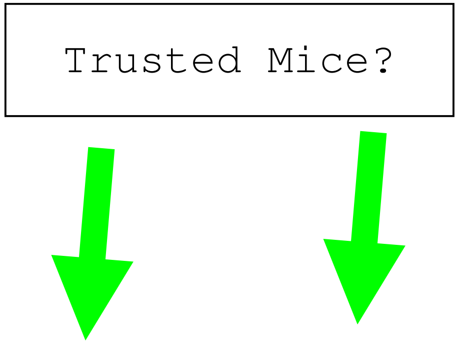
The image on this page of my office desk has been removed from the online version because it made the file much too big (applies to subsequent 6 pages too)

It wasn’t very interesting anyway…

The Battlefield Expanded

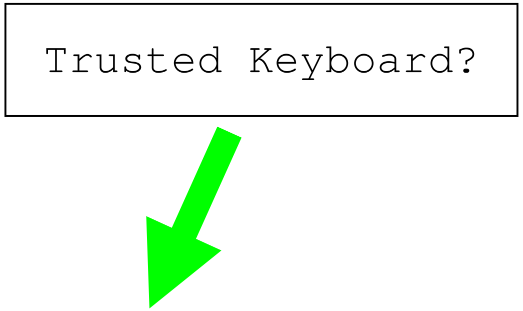


The Battlefield Expanded



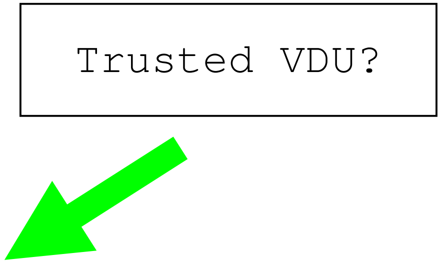
Trusted Mice?

The Battlefield Expanded



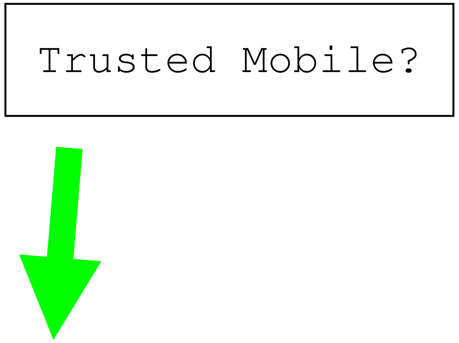
Trusted Keyboard?

The Battlefield Expanded



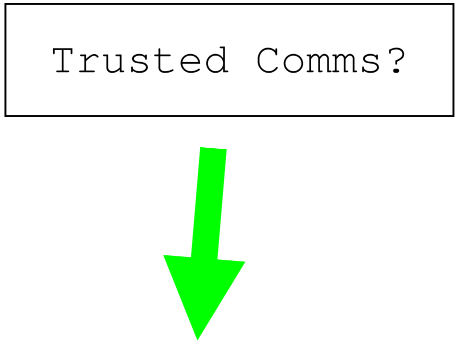
Trusted VDU?

The Battlefield Expanded



Trusted Mobile?

The Battlefield Expanded



Trusted Comms?

Conclusions

Question : What do you get if you cross new legislation, new marketing models, and trusted computing?

Answer: WAR

- Security and cryptography will be used more and more for corporations to hold onto their customer bases, protect their revenue streams, segment their markets, and generally beat back the competition
- Security APIs, simple or complex may soon be governing the interaction between devices, from PCs to Price Tags
- The corporations are already at war; devices on our PCs and on our desks could become the footsoldiers.
- Devices that should be co-operating with each other to make our lives simpler will soon be at war!
- From our previous experience of commercial security API design, getting things right is hard. If legislators allow it, these wars may rage long and hard.

More Info

- Academic Papers

“Decimalisation Table Attacks for PIN Cracking”
Bond, Zielinski, Mar 2003

“API-Level Attacks on Embedded Systems”
Bond, Anderson, Oct 2001

“The Design and Analysis of Cryptographic APIs for Security Devices”
Clulow, Jan 2003

- My Webpage

http://www.cl.cam.ac.uk/~mkb23/research.html