Lowering the cost of Bank Robbery

(or, “Why I was wearing a tie on the telly”)

Richard Clayton

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

Presented at: BA Festival of Science, 12th September 2002

Summary

- Keys and Ciphers
- The IBM 4758 cryptoprocessor
- How PIN values work
- Mike Bond’s “API attacks”
- The low-cost hardware “DES cracker”
- How to extract 3DES keys from a IBM 4758
- Some thoughts on “full disclosure”

Cambridge University Computer Laboratory Security Group

Keys and Ciphers

- Kerckhoff’s doctrine (1883)
  - the security of a system should depend upon its key and not upon its design remaining obscure
- If there is no shortcut then the security of a system depends upon its key length
  - trying all possibilities @ 33 million keys/sec
    - 2^{40} = 9.25 hours
    - 2^{56} = 69.2 years
    - 2^{80} = 1.2 billion years

A History of Tamper Resistance

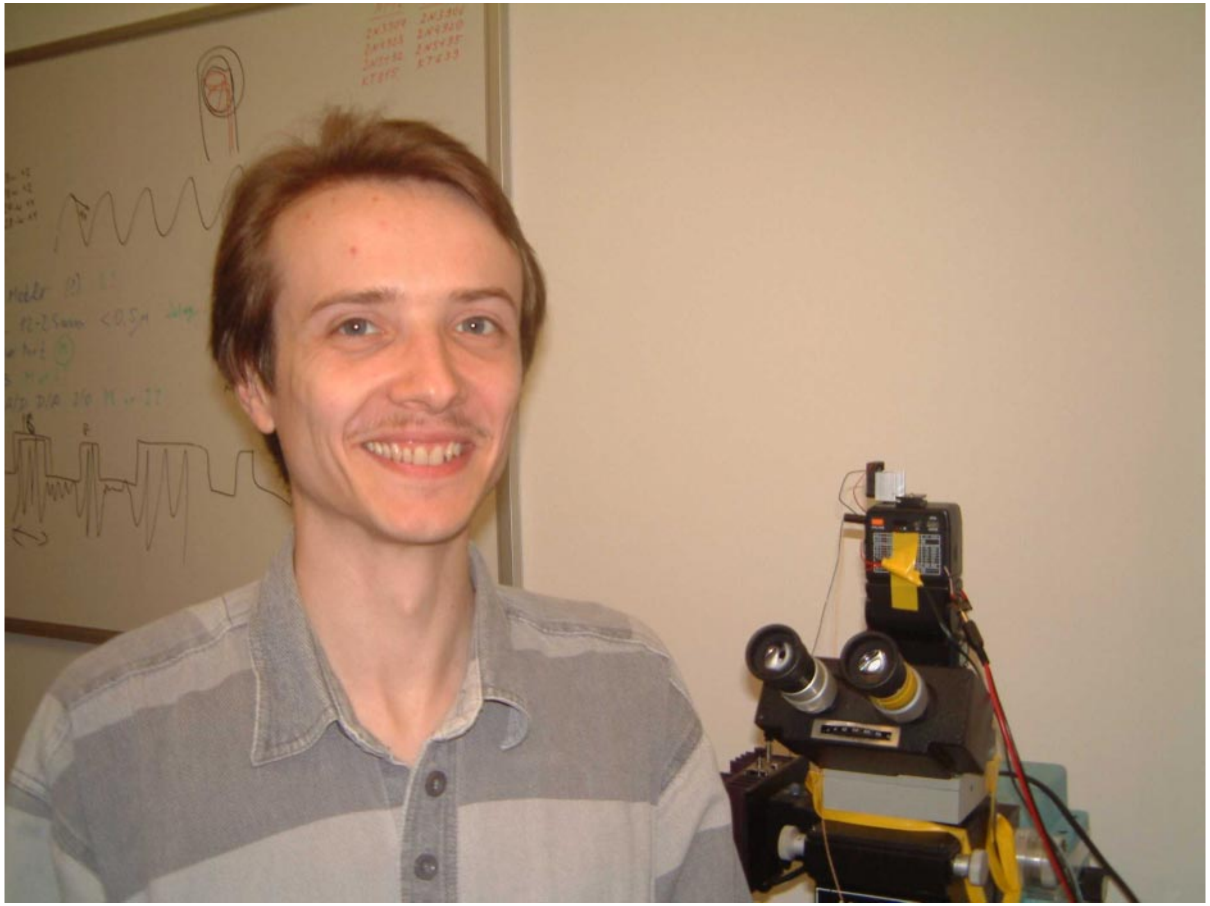
Problem: another program on the same machine can access your sensitive data

- Put keys into separate microprocessor
- Put microprocessor into a tin box
- Lid opening switches and photocells
- Epoxy “potting”
- Tamper detecting barriers

Smartcards: **NOT** Well Protected

- Simple attacks on Vpp, slow clocks &c
- Damage the processor to access all RAM
- Probing
- Focused Ion Beam (FIB) workstations
- Power analysis
- Attacks with flashguns!

Sergei P. Skorobogatov



The IBM 4758

- Protective barrier with wires of chemically similar compound
- Detectors for temperature & X-Rays
- “Tempest” shielding for RF emission
- Low pass filters on power supply rails
- Multi-stage “ratchet” boot sequence

= STATE OF THE ART PROTECTION!

CAUTION
VARTA
Lithium 3V

CCA and PIN values

- Common Cryptographic Architecture
  - runs on many IBM platforms
  - available for free to run on a 4758
- A PIN value (in the CCA world) is the account number encrypted with (112 bit) 3DES key and last few bytes made decimal
- Changing a PIN => changing an offset

Key Entry under CCA

- Each key is loaded in two parts, which are then XORed together
  - XOR means that knowing one part tells you NOTHING about the final key value
- Two security officers, “trusted” not to collude, are given one part of the key each.
  - They authenticate themselves and then separately load these into the 4758.
- This makes the key entirely secure...

Mike Bond

Michael Bond’s “API attacks”

- New type of attack: use standard API in non-standard way to cause dumb things
  - Overloaded key types
  - Unauthorised type casting
  - 3DES binding attack
  - Related keys

Mike’s PhD topic targets formal methods that will detect (and avoid) these problems

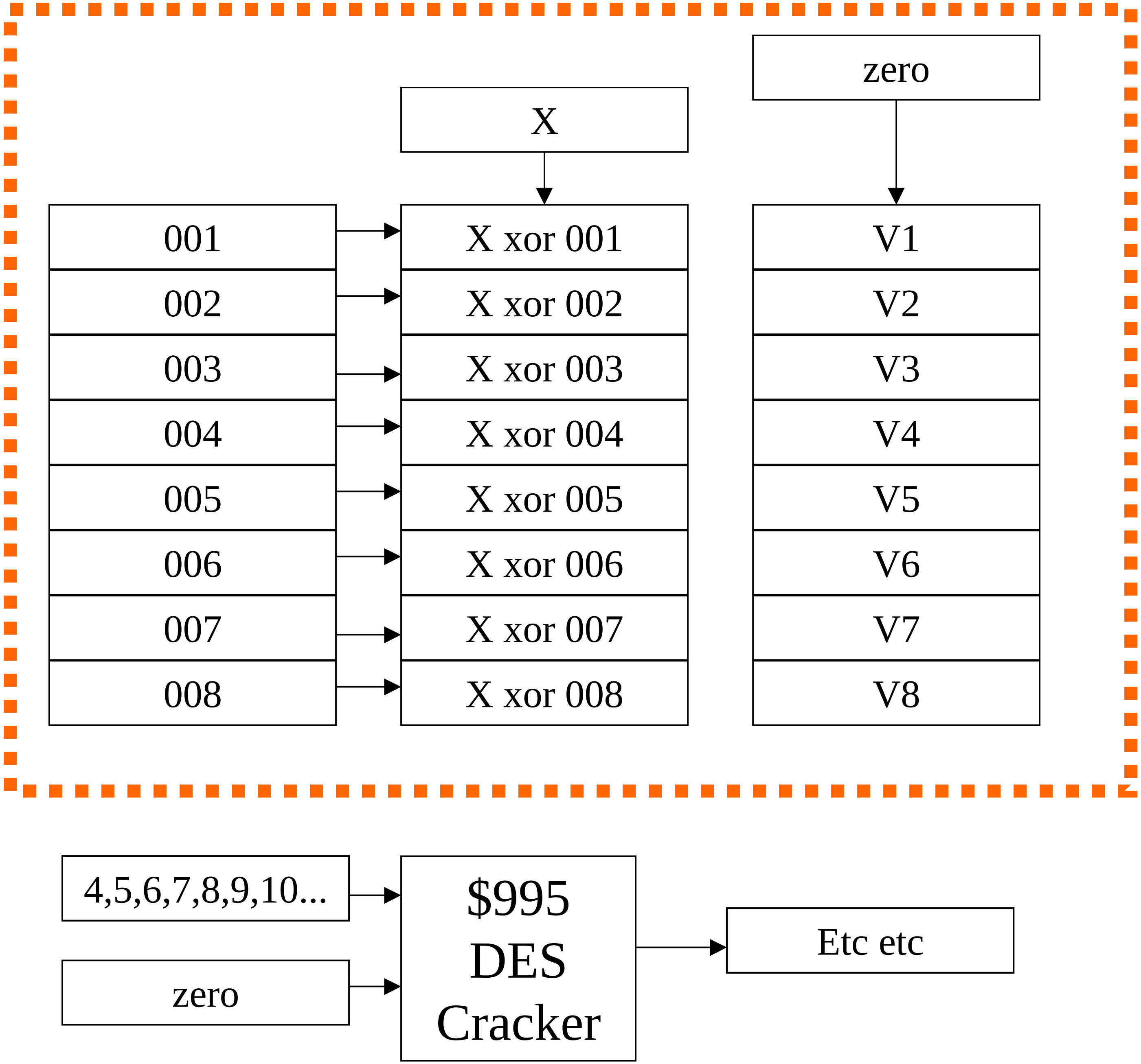
The Meet-in-the-Middle Attack

Idea: Attack multiple keys in parallel

- Encrypt the same plaintext under each of the multiple keys to get a “test vector”
- Attack by trying all keys in sequence but check for a match against any test vector value (check is faster than encrypt)
- Typical case: A 2^{56} search for one key becomes a 2^{42} search for 2^{14} keys

Attacking the CCA : Part 1

- Create unknown DES key part
- XOR in “...001”, “...002”, “...003” etc
- Encrypt zero value under each key
- Repeat to get 16384 (2^{14}) results
- Some complexity because of parity issues, but essentially simple & takes 10 minutes.
- Use “brute-force” attack to get the DES key



X

zero

001
002
003
004
005
006
007
008

X xor 001
X xor 002
X xor 003
X xor 004
X xor 005
X xor 006
X xor 007
X xor 008

V1
V2
V3
V4
V5
V6
V7
V8

4,5,6,7,8,9,10...

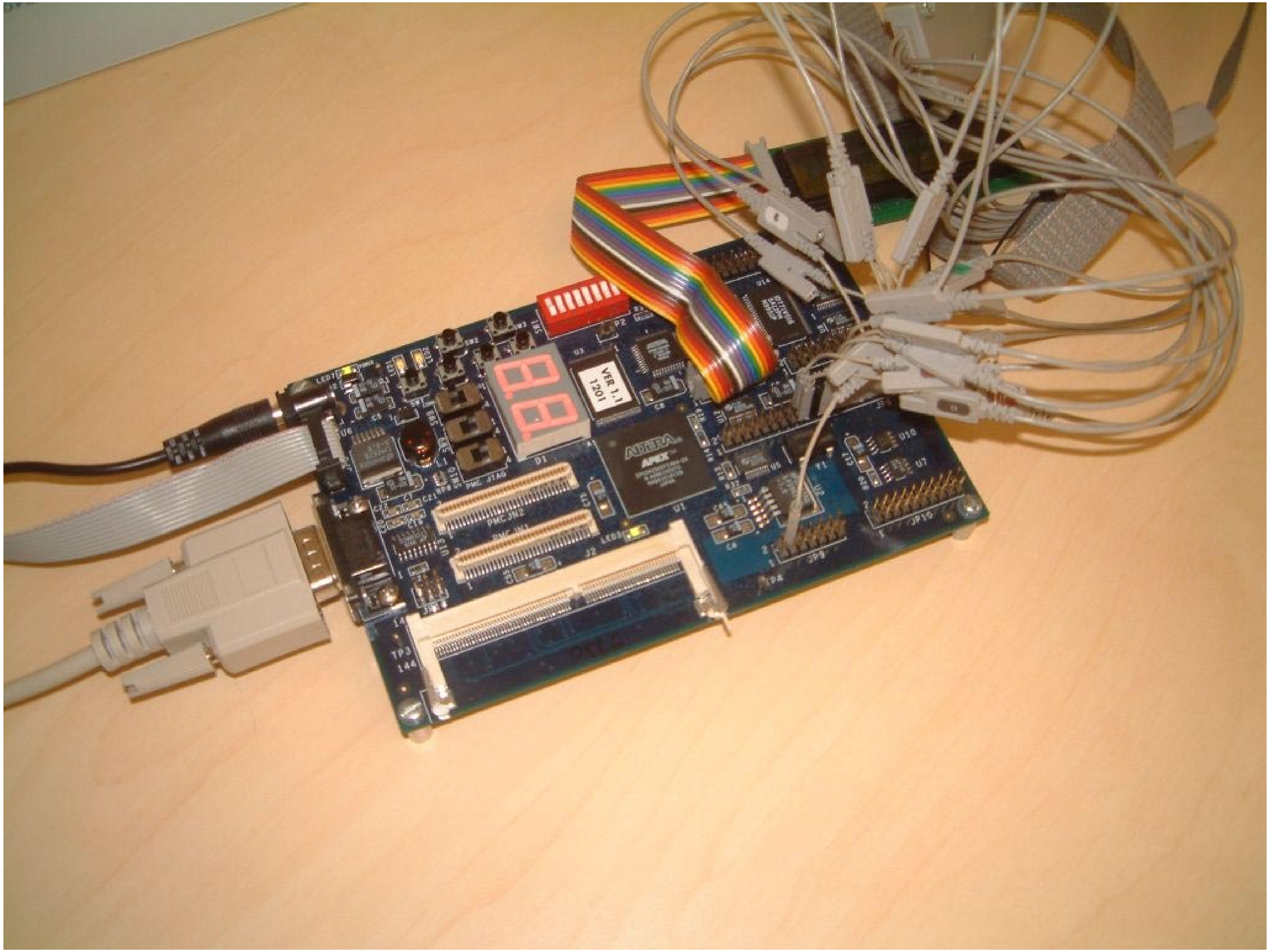
zero

$995
DES
Cracker

Etc etc

Low-cost DES Cracker

- $995 Excalibur kit (Altera 20K200 FPGA)
  - chip cost is ~$5 (in volume; $178 one-off)
- 33MHz pipeline (& 60MHz possible)
- 2^{25} keys/second
  - 56 bit DES = 68 years
- However.. it looks for 16384 keys in parallel
  - with average luck find first key in 25.4 hours


VER 1.1
1201
ALTERA
APEX
PMC JN2
PMC JN1
PMC JTAG
J2
U1
D1
LED3
JP9
JP10
U7
U10
U14
U6
TP3
144

Why Use Hardware Anyway?

Hardware DES implementation is >>25 times faster than the best software implementations.

- eg: Software [seeking any 1 of 64K keys]
  - 6 modern PCs running in parallel
  - £4500
  - 84 hours (3.5 days)
- & Hardware [seeking any 1 of 16K keys]
  - Altera evaluation board (no soldering required)
  - $995
  - 22.5 hours (for same example, NB: 1/4 parallelism)

Attacking the CCA : Part 2

- Recall we had 16K related DES keys
- We can crack one of these in ~1 day
- Now create 16K related 3DES keys with "replicate" halves and "exporter" capability
  - 3DES = EncryptA; DecryptB; EncryptA
- Export the DES key under the 3DES keys
- Since replicate can also crack in ~1 day

Attacking the CCA : Part 3

- Create non-replicate 3DES key by combining two unequal halves with the replicate halves that we’ve now determined
- Export all the CCA keys under this key
- Download list of PIN offsets
- Use magnetic stripe writer to create cards
- Use any ATM to extract money from accounts
- Go to Bermuda!

IBM’s Response

- Nov 2000 (Mike’s first results)
  - nothing (typecasting seen as legitimate)
- May 2001 (Mike’s CHES paper)
  - nothing
- Nov 2001 (Newsnight program)
  - attack “infeasible in realistic system implementations”
  - followed by advice to disable **`Combine_Key_Parts`**
- Feb 2002
  - new version of CCA available [+ bug fix]

“Full Disclosure”

- Should you tell vendor & keep quiet ?
  - vendor has limited incentive to act
- Should you publish & be damned ?
  - “black hats” may be unaware of problem
- Should exploits be published ?
  - “script kiddies” & sysadmins both need them
- Current consensus is to tell vendor and publish after pre-set delay. Recent decisions to suppress exploit info are controversial.

Make Your Own!

`http://www.cl.cam.ac.uk/~rnc1/descrack/`