Experience Using a Low-Cost FGPA Design to Crack DES Keys

Michael Bond &

Richard Clayton

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

CHES Workshop

15th August 2002

Contents

- Attacks on the IBM 4758 CCA
- Attack optimisation
- History of H/W crackers
- The low-cost hardware “DES cracker”
- DES cracker design issues
- Results and responses


CAUTION
VARTA
Lithium 3V

Last Year at CHES …

- Attacks on IBM 4758 Common Cryptographic Architecture, used in retail banking to protect customer PINs and ATM infrastucture
- Flaws identified
  - Poor control of integrity on key entry
  - Meet-in-the-middle attack on DES
  - 3DES Key binding problem

Key Entry under CCA

- Used for transferring top-level encryption keys between banks
- Each key is split into several parts, transferred by separate couriers
- Security Officers at destination receive one part each, and enter them into the 4758
- The parts are recombined using XOR
- Problem : any of the security officers can modify the key value (though all must collude to discover it)

Example Roles

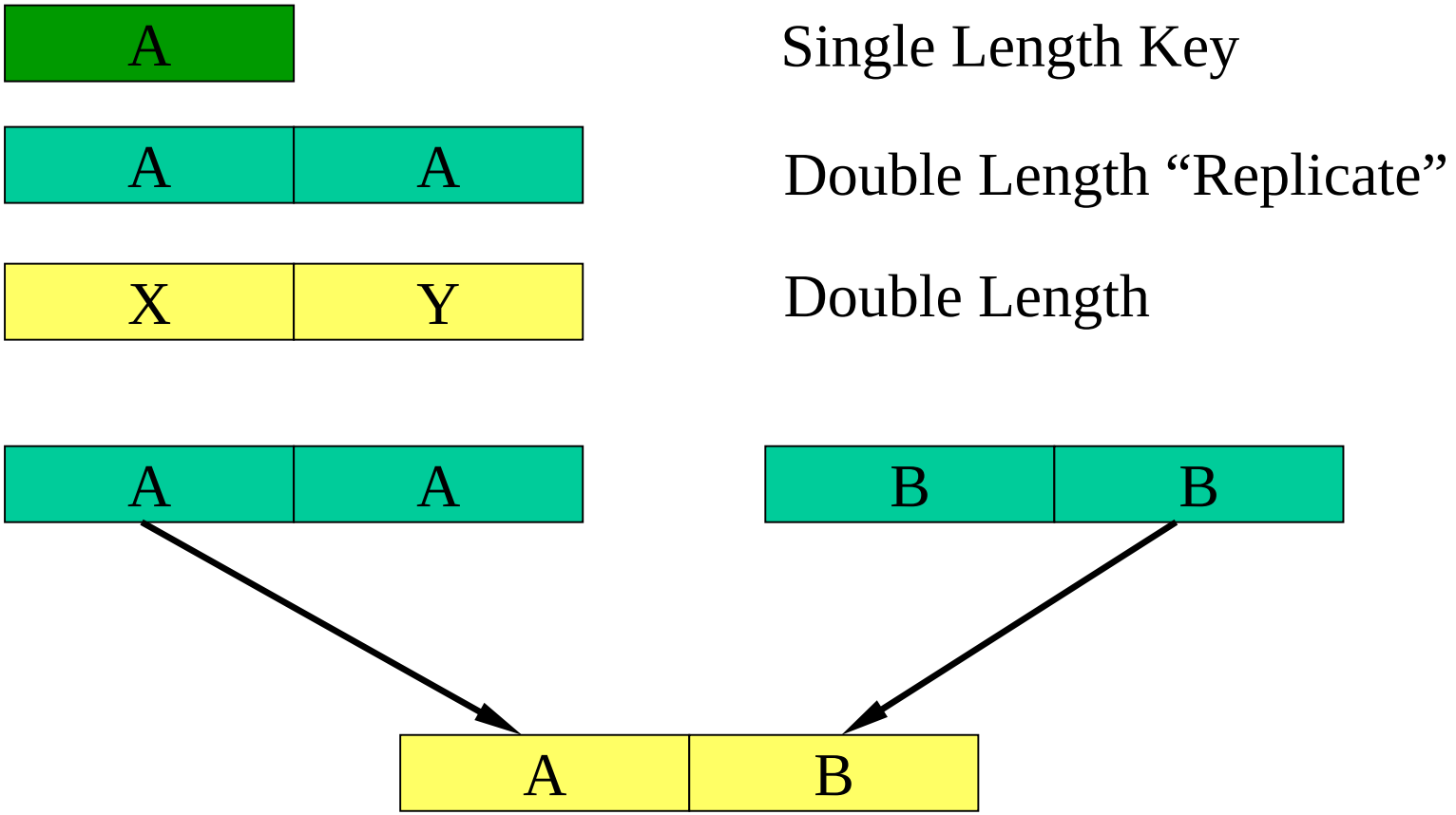
Role	**CCA Permissions**	**Abilities**
Security Officer 1	Load_First_Key_Part	Can modify key Can start key load **Cannot** discover key value
Security Officer 2	Combine_Key_Parts	Can modify key Can complete key load **Cannot** discover key value
Security Officer 3	Combine_Key_Parts Key_Test	Can modify key Can complete key load Can check key integrity **Cannot** discover key value

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

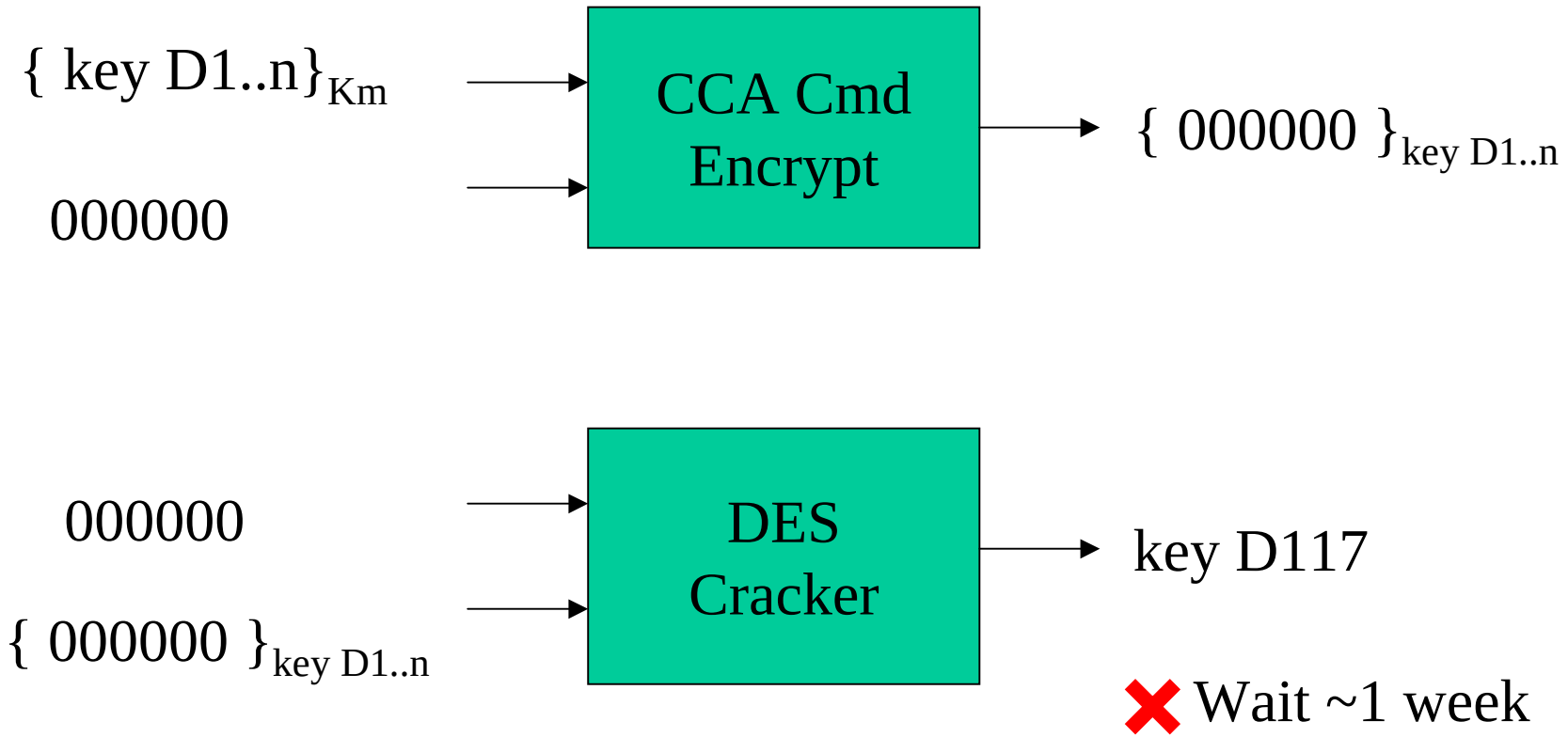
3DES Key Binding



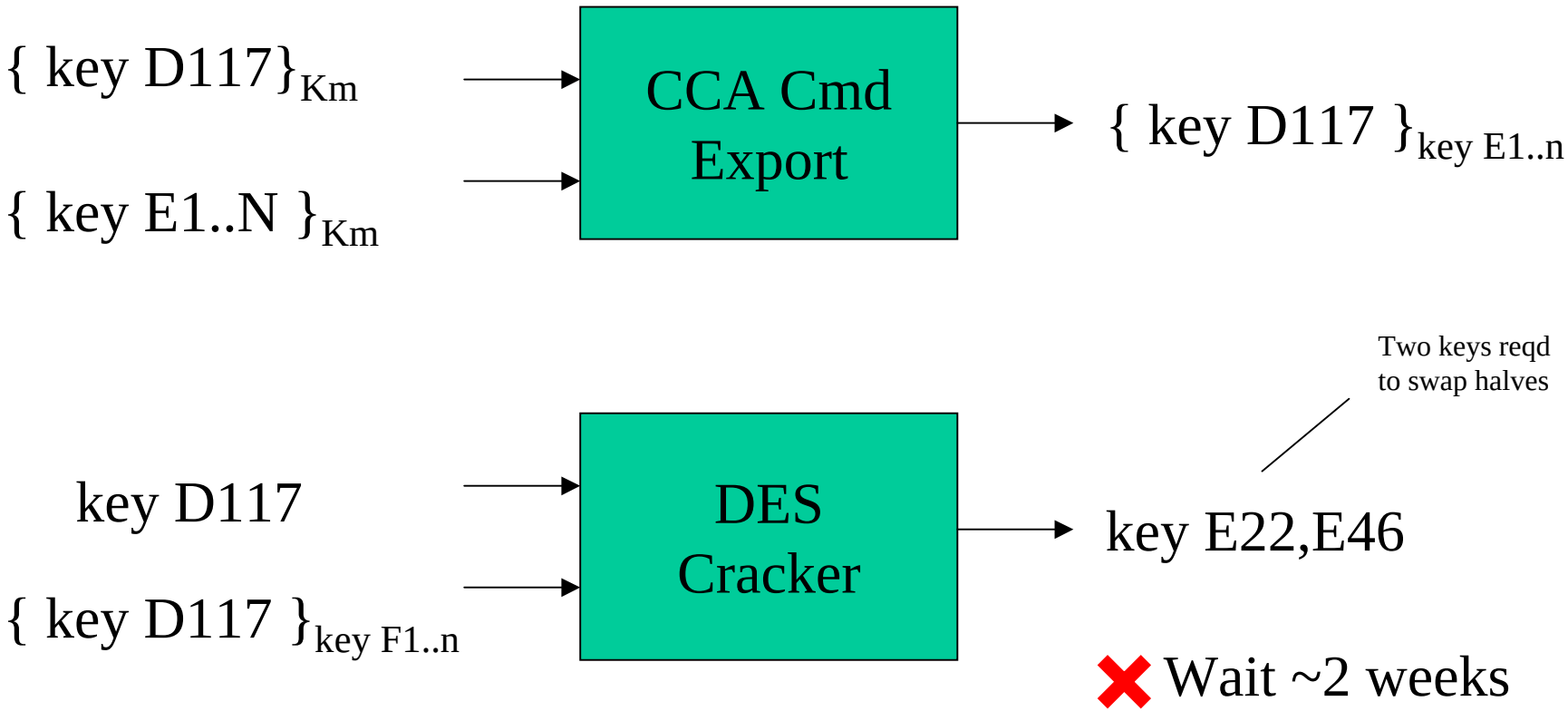
Implementing the Attack

- Naïve implementation of attack requires 3 sessions of access to 4758, and three DES cracks. Total time ~3.5 weeks
- Our aims
  - Require only a single session of access < 30 mins
  - Have the cracking complete within a long weekend
- Our solution
  - Restructure and optimise attack code
  - Use hardware to assist in the cracking

Original Attack : Stage 1

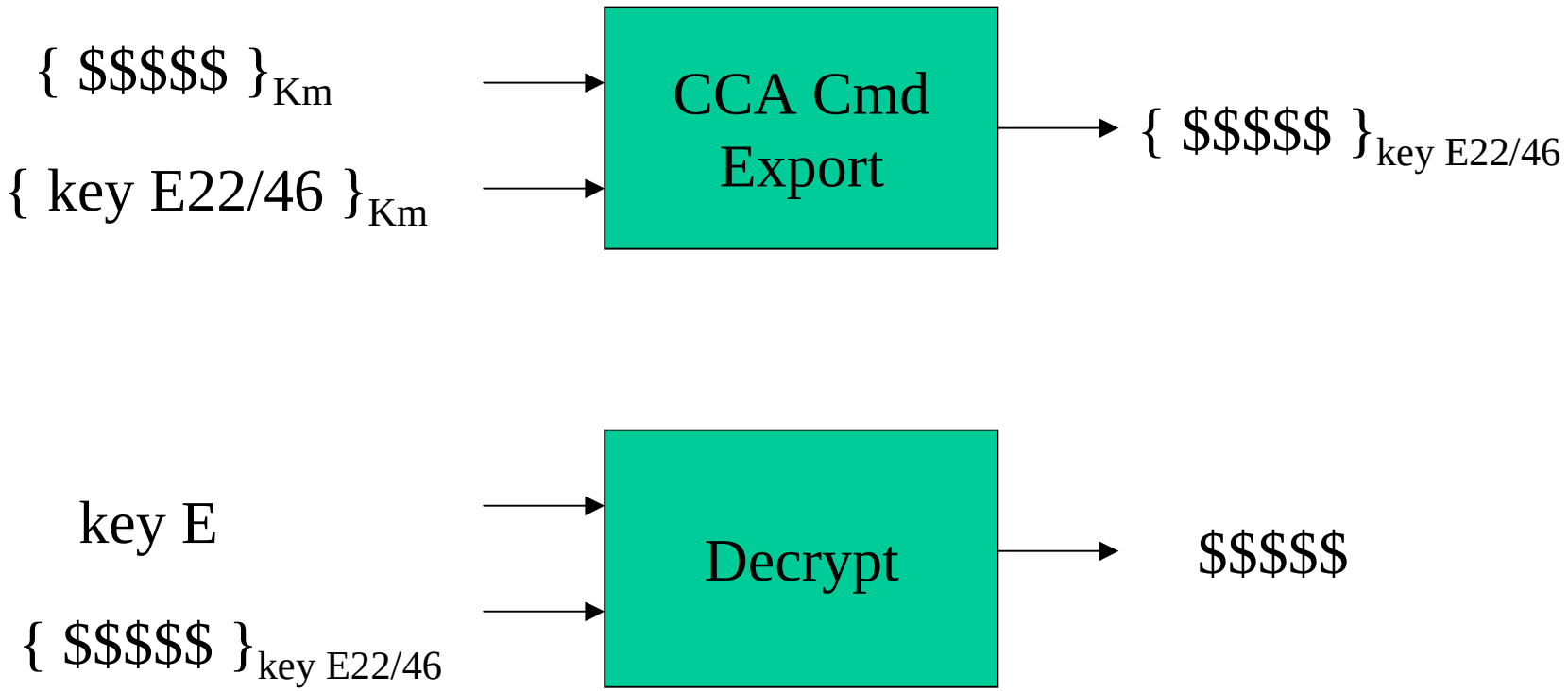


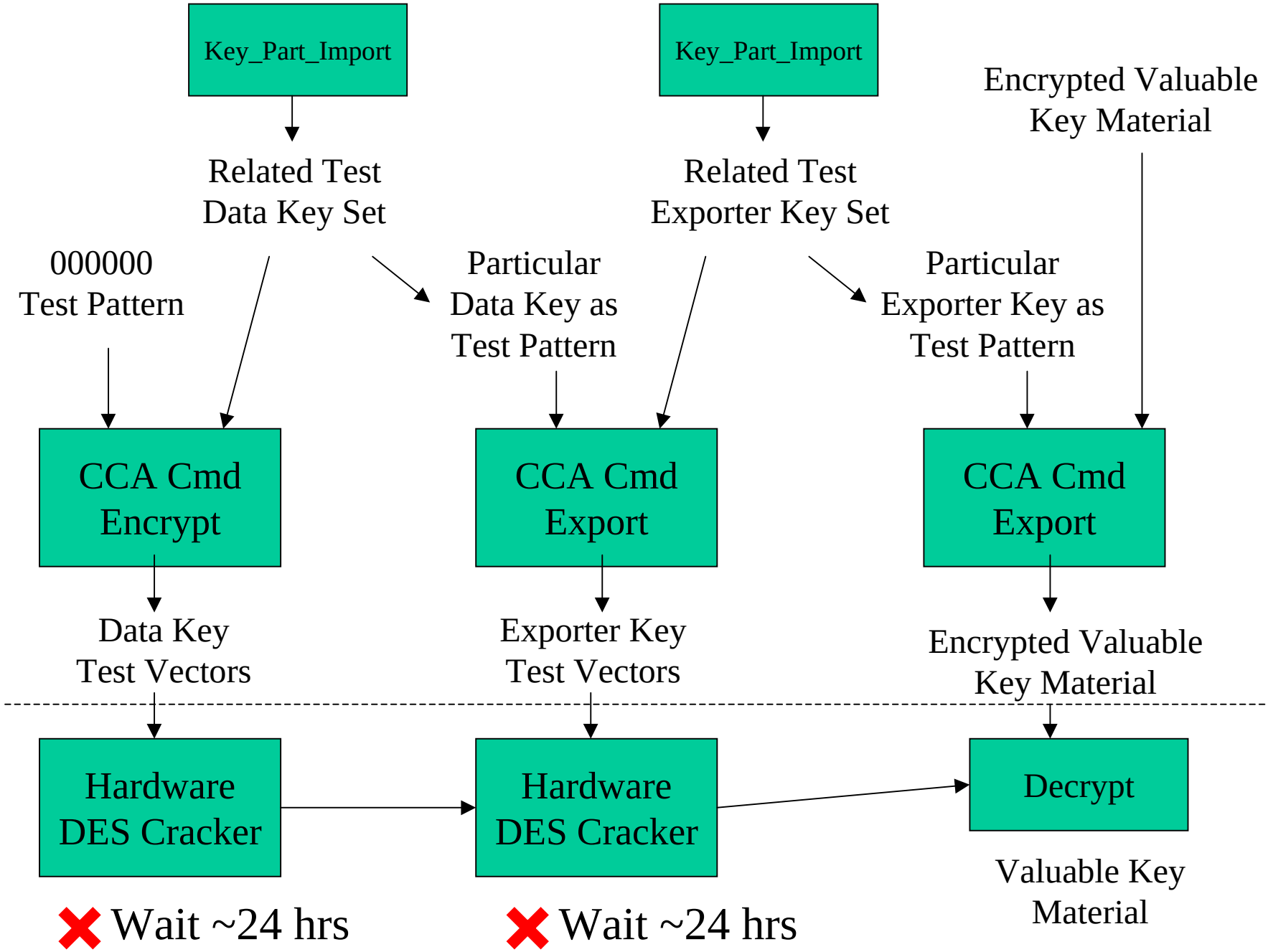
Original Attack : Stage 2

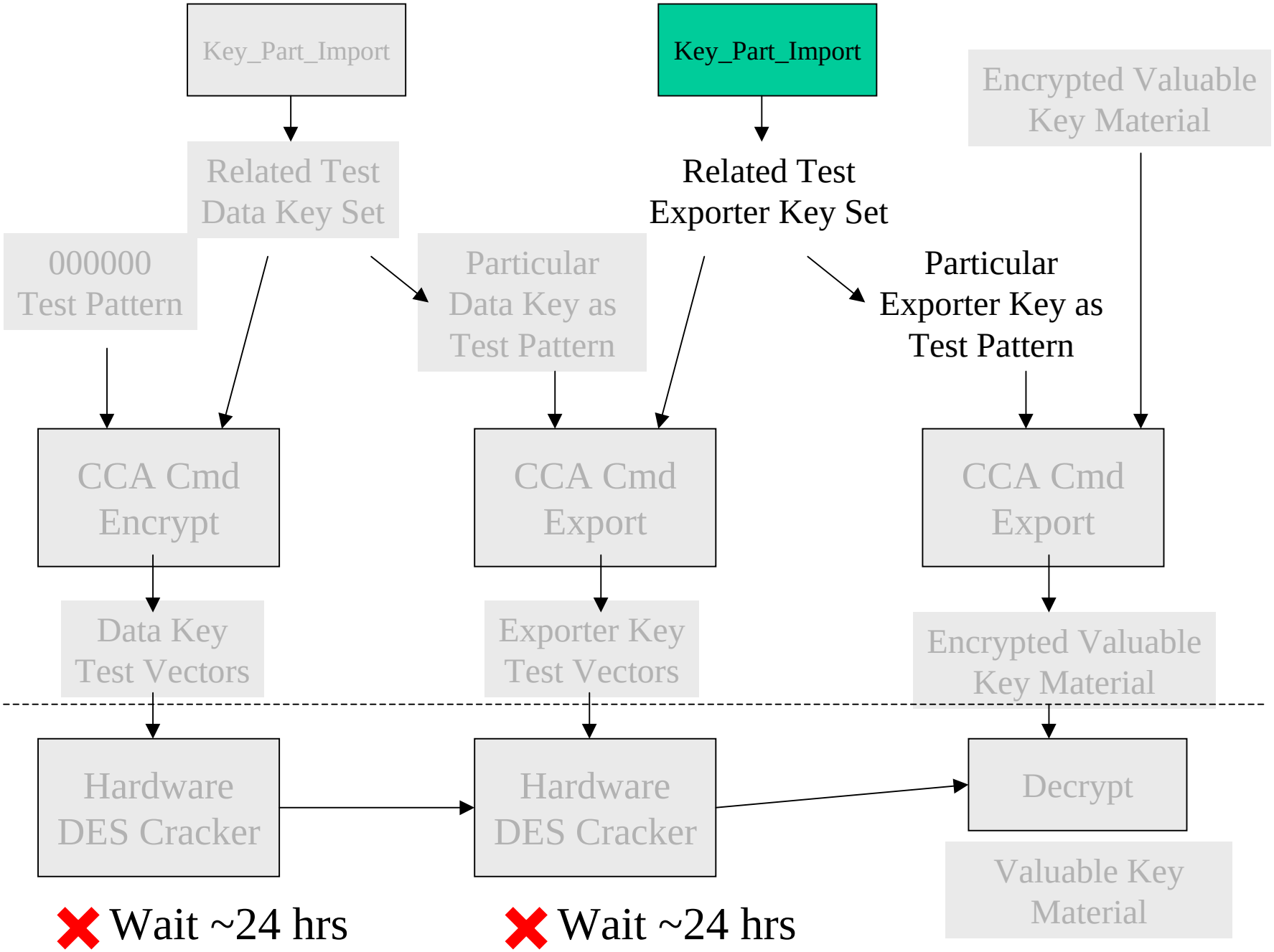


Original Attack : Stage 3

First, swap halves of keys E22 & E46




Key_Part_Import
Key_Part_Import
Encrypted Valuable Key Material
Related Test Data Key Set
Related Test Exporter Key Set
000000 Test Pattern
Particular Data Key as Test Pattern
Particular Exporter Key as Test Pattern
CCA Cmd Encrypt
CCA Cmd Export
CCA Cmd Export
Data Key Test Vectors
Exporter Key Test Vectors
Encrypted Valuable Key Material
Hardware DES Cracker
Hardware DES Cracker
Decrypt
Valuable Key Material
Wait ~24 hrs
Wait ~24 hrs


Key_Part_Import
Key_Part_Import
Encrypted Valuable Key Material
Related Test Data Key Set
Related Test Exporter Key Set
000000 Test Pattern
Particular Data Key as Test Pattern
Particular Exporter Key as Test Pattern
CCA Cmd Encrypt
CCA Cmd Export
CCA Cmd Export
Data Key Test Vectors
Exporter Key Test Vectors
Encrypted Valuable Key Material
Hardware DES Cracker
Hardware DES Cracker
Decrypt
Valuable Key Material
Wait ~24 hrs
Wait ~24 hrs

Finishing off

- Download lists of account numbers and PIN offsets
- Use magnetic stripe writer to create cards
- Use any ATM to extract money from accounts
- Go to Bermuda!

Next : design of cracker

Predicting Brute Force of DES

- Diffie/Hellman 1977 $20M for 1 key/day
- Jueneman 1980 : by 1985 $10M for 2 secs
- Hoornaert 1984 : $1M for 4 weeks
- Desmedt 1987 : $3M for 4 weeks
  (as Hoornaert but 1M keys in parallel)
- Wiener 1993 : <$1M for 1 key/3 hours

The EFF Machine (1998)

- 1 unit tests 1 key in 16 clocks @40 MHz
- 24 units/ASIC
- 64 ASICs/board
- 12 boards/chassis, 2 chassis = 1 machine
- Looking for “known plaintext”
- Full 2^{56} search takes 9 days
- $210,000 – of which $80,000 was chips

RSA Challenges

- June 97 : 96 days (25% of space)

  DESCHALL – peak day: 2^{32} keys/sec
- February 98 : 41 days (90% of space)

  Distributed Net – peak day: 2^{36} keys/sec
- July 98: 56 hours (27% of space)

  EFF “Deep Crack” – $2^{36.5}$ keys/sec
- January 99 : 22 hours (25% of space)

  Distributed Net + EFF – reached $2^{37.8}$ keys/sec

Later Machines

- Transmogrifier 2a (Univ. Toronto) 1999
  - 32 * Altera 10K100 FPGAs + glue!
  - 25MHz
  - $2^{29.6}$ keys/sec : ie 2.85 years/key
  - $30K cost (estimated – chips were free!)
  - For $210K they estimate 8X EFF speed
- Not many more actually built !

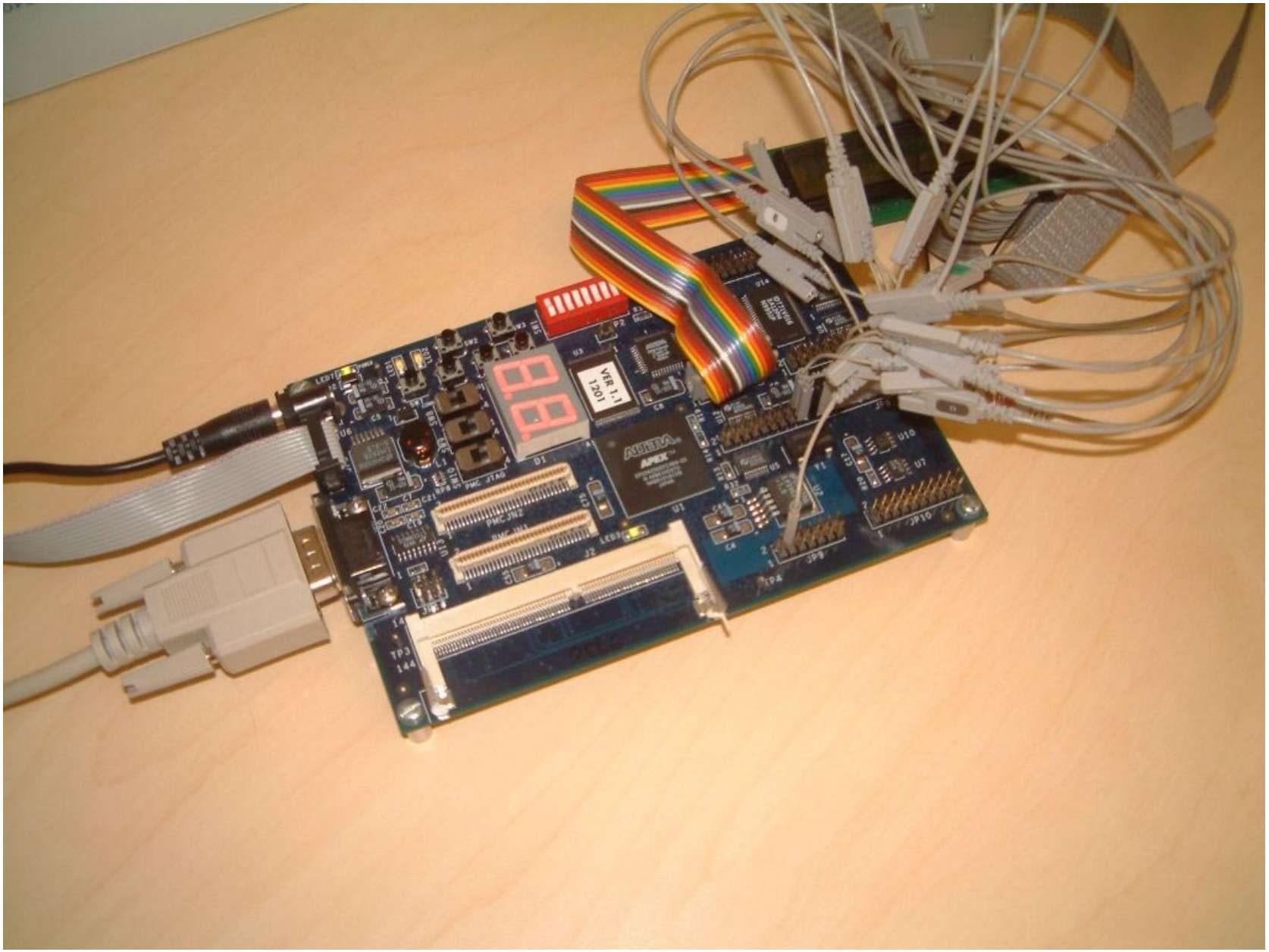
Recent Estimates

Blaze, Diffie, Rivest, Schneier, Shimomura, Thompson & Wiener (Jan 1996)

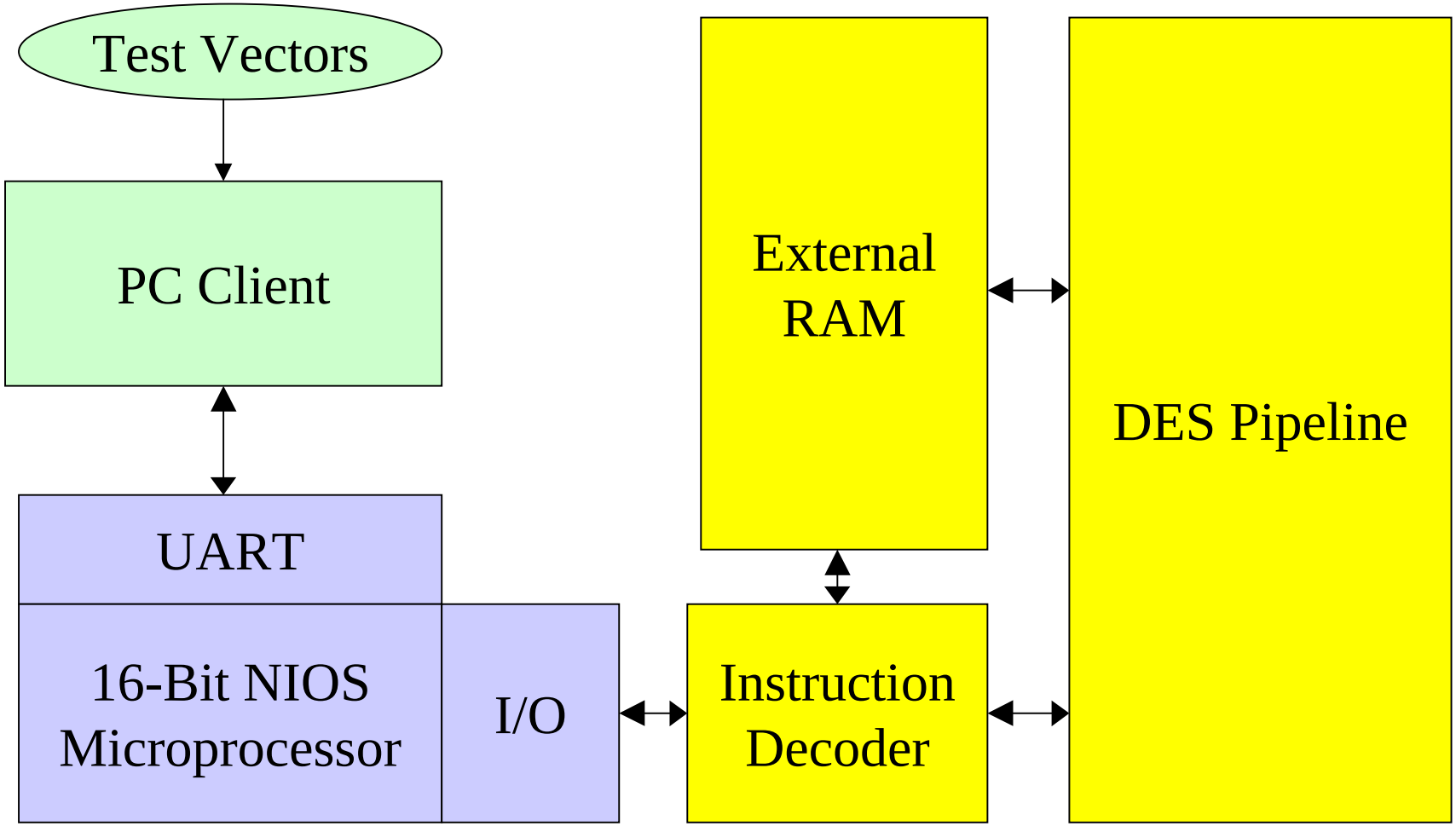
- Surveyed software & FPGA solutions
  - 40 bit keys – one week in software
  - $400 FPGA – 5 hours / 40 bit key = $0.08/key
  - Assumed 60MHz pipeline in the FPGA
- Recommended 90 bits as safe for 20 years even when targeted by major governments

Our Low-cost DES Cracker (2001)

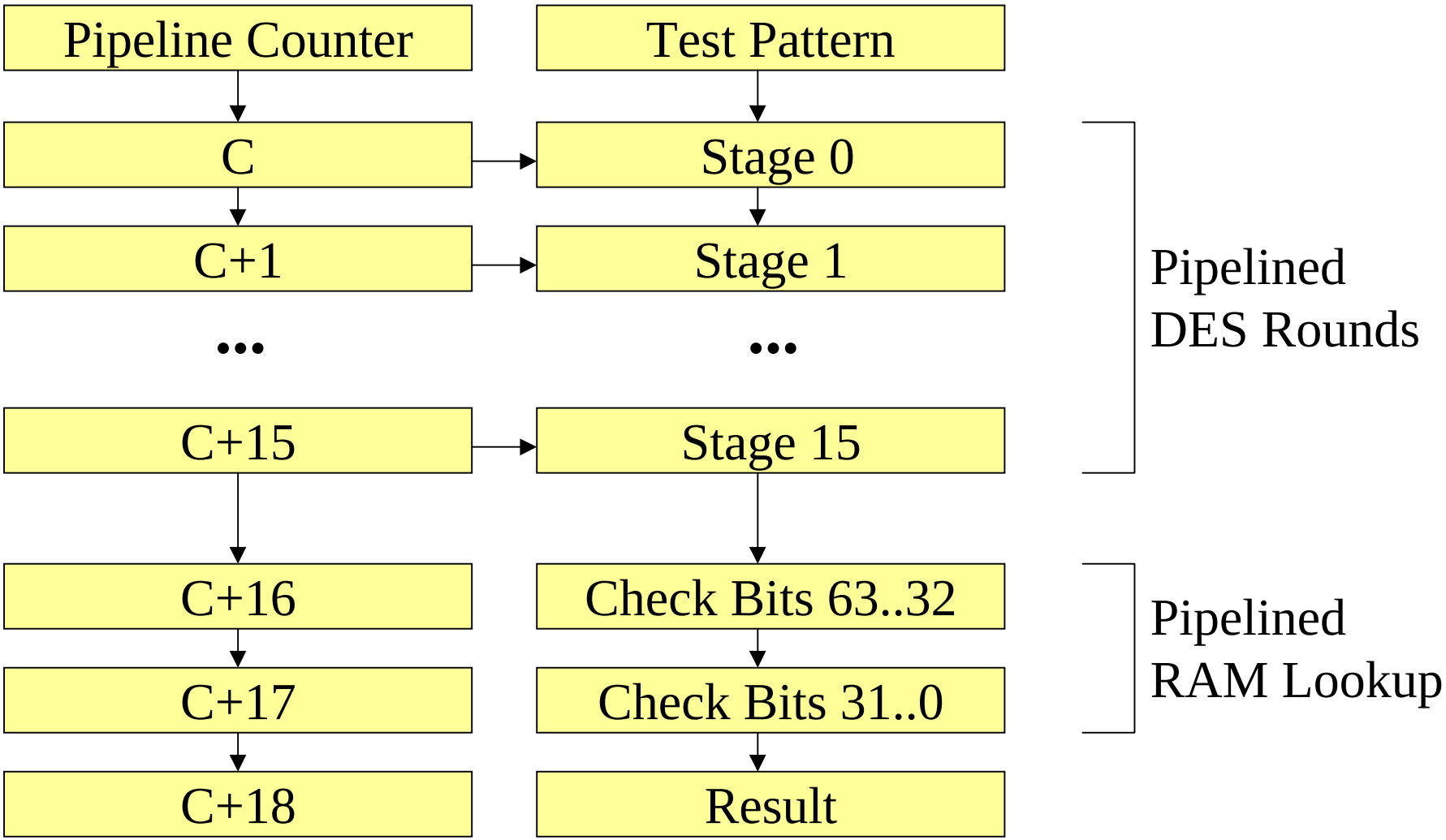
- $995 Excalibur kit (Altera 20K200 FPGA)
  - chip cost is ~$5 (in volume; $178 one-off)
- 33MHz pipeline (& 60MHz possible)
- 2^{25} keys/second
  - 56 bit DES = 68 years
- However.. it looks for 16K keys in parallel
  - with average luck find first key in 25.4 hours


VER 1.1
1201
PMC JTAG
PMC JN2
PMC JN1
ALTERA
APEX

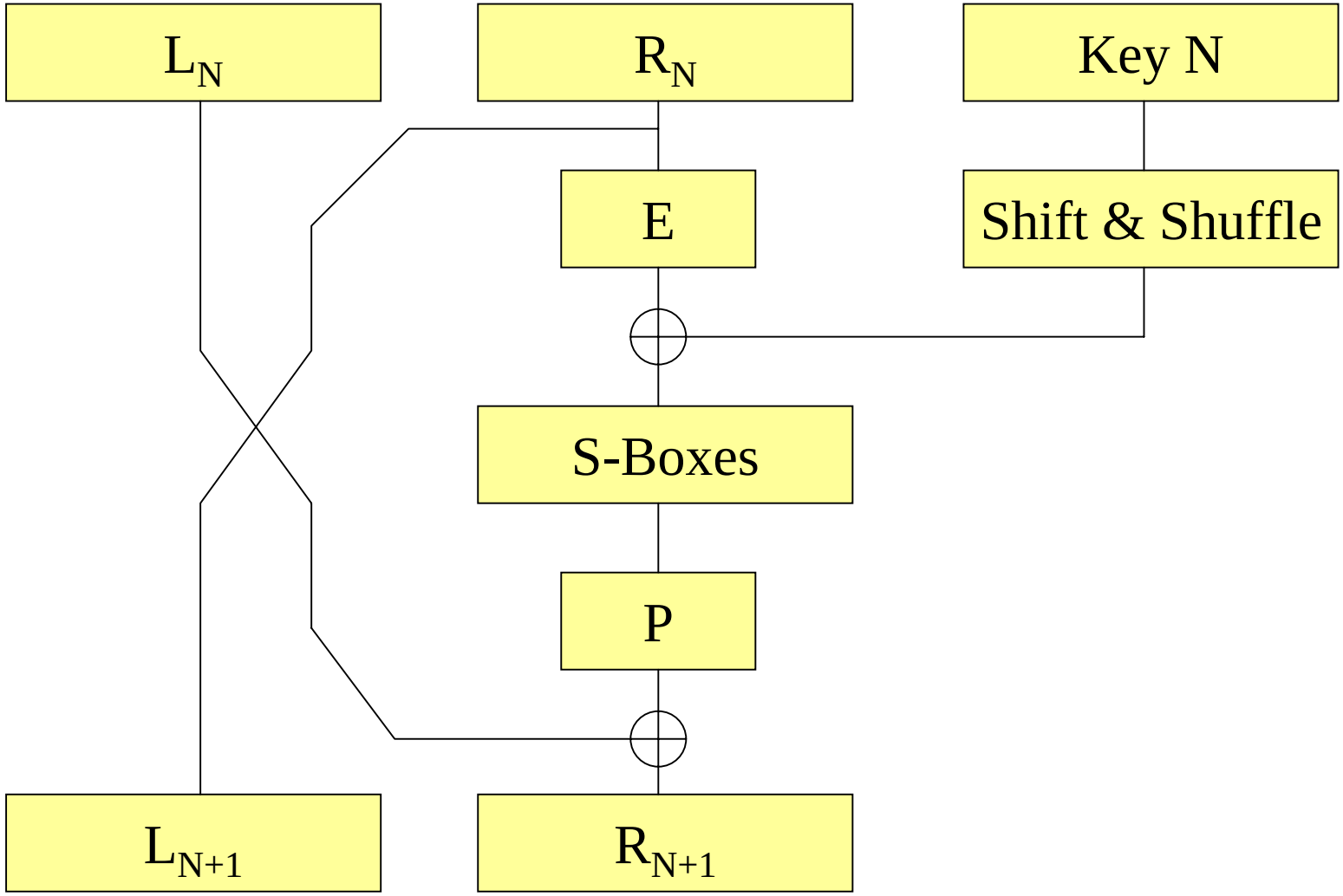
Design Overview



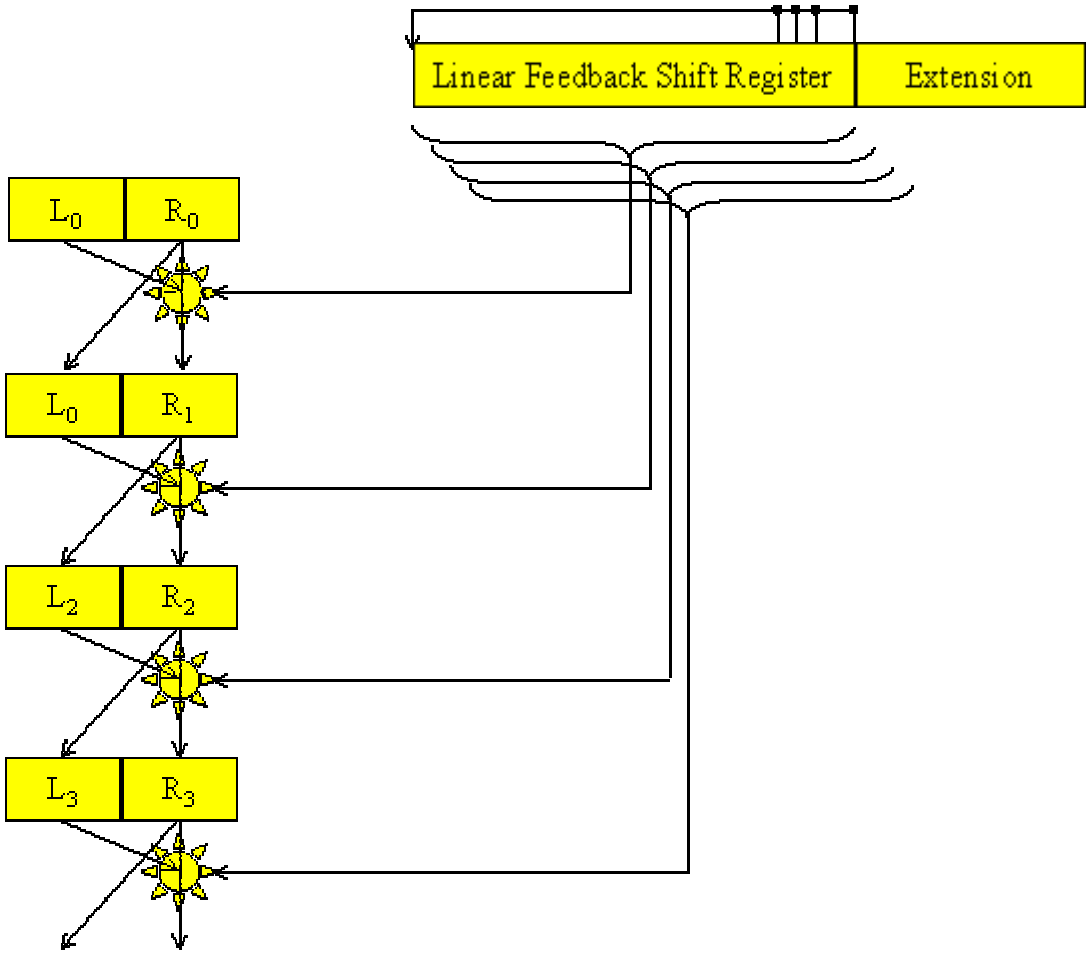
The DES Engine



A DES Pipeline Stage



Feeding in Subkeys

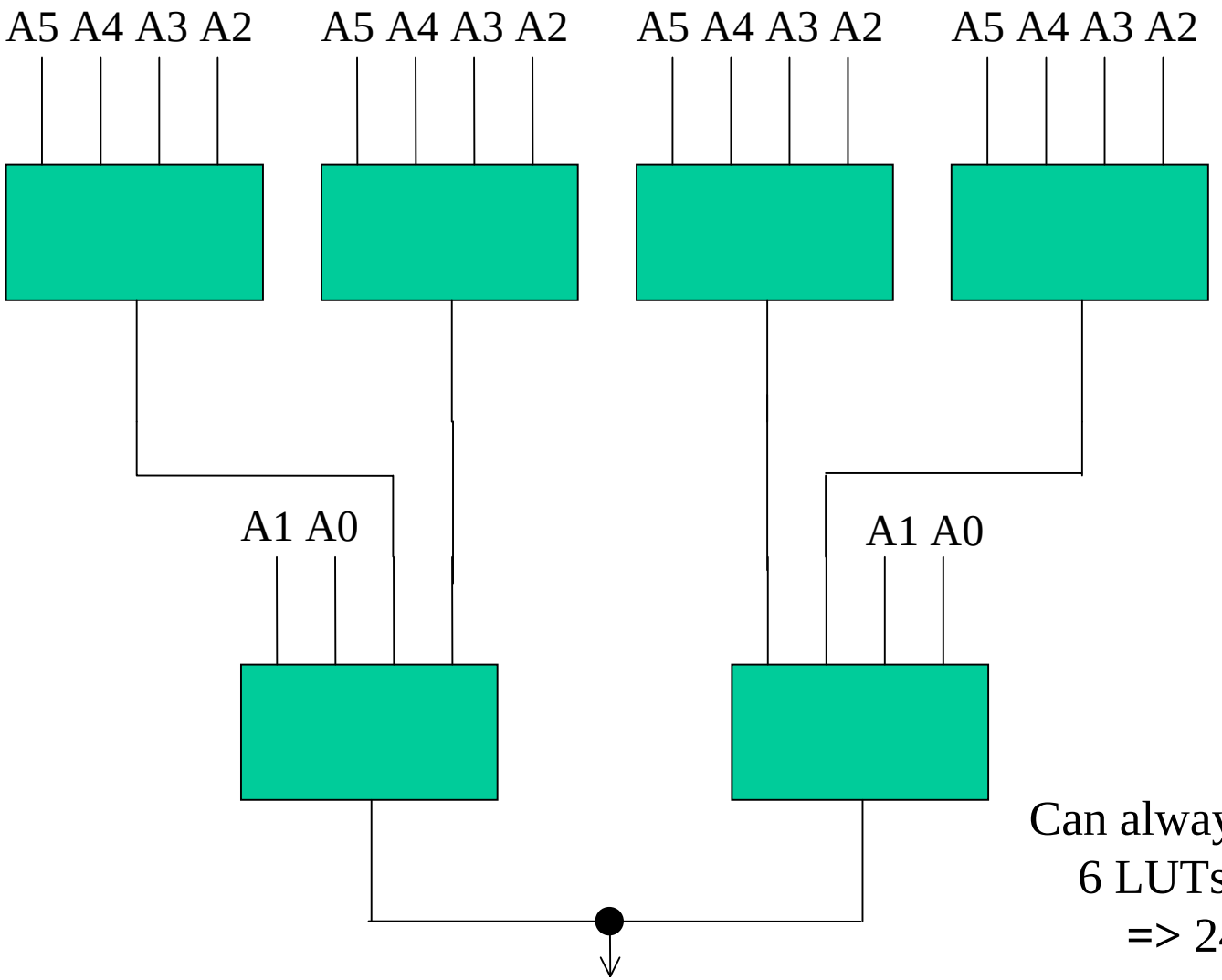


Fitting the Design Onto the Chip

Max of 8320 LUTs … and using all except 17

- LFSR saves pipelining key values
- Careful attention to instruction decoder
- Minimal settings for NIOS processor
- Redesigned S-Boxes

Straightforward S-Box Design



Can always achieve:
6 LUTs / bit
=> 24 LUTs/S-Box

Some S-Boxes Have Structure

- SBOX4 : address : 543210 : 4 bit result =

 7, 13, 14, 3, 0, 6, 9, 10, 1, 2, 8, 5, 11, 12, 4, 15,
13, 8, 11, 5, 6, 15, 0, 3, 4, 7, 2, 12, 1, 10, 14, 9,
10, 6, 9, 0, 12, 11, 7, 13, 15, 1, 3, 14, 5, 2, 8, 4,
 3, 15, 0, 6, 10, 1, 13, 8, 9, 4, 5, 11, 12, 7, 2, 14.

- Rearrange addressing in order 532104

 7, 14, 0, 9, 1, 8, 11, 4, 10, 9, 12, 7, 15, 3, 5, 8,
13, 11, **6**, 0, 4, 2, 1, 14, 3, **0**, 10, **13**, 9, 5, 12, 2,
13, 3, **6**, 10, 2, 5, 12, 15, 6, **0**, 11, **13**, 1, 14, 2, 4,
 8, 5, 15, 3, 7, 12, 10, 9, 15, 6, 1, 8, 4, 11, 7, 14.

and then feed it into the logic minimiser...

S-Box Savings

- S-Box 4 uses just 16 LUTs, not 24
- We tried all possible addressing permutations and got savings also on:

S-Box 2	23
S-Box 3	23
S-Box 7	23
S-Box 8	22

- total 13 LUTs saved per stage
  * 16 stages = 208

Pipelining vs. Looping

- We considered a looped architecture as well as the pipelined version.
  - Pipelined 8303
  - Looped 11162 ~30% larger
- Looping pros
  - easier to get right (EFF machine)
  - easy to cash in speed for space
- Pipelining pros
  - space efficiency

Results

- Implementation complete October 2001
- 4 full attack runs
  - 5hrs, 12hrs, 19hrs, 22hrs to find first key
- Informed IBM both when theory discovered, and when implementation complete – little response
- Media publicity in November 2001
- Initially – denial
- One week later – warning about Key_Part_Import
- February 2002, new CCA version 2.41 with fixes

Conclusions

- There is value in implementing attacks “for real”
  - Problem with Key_Part_Import would never have been spotted
  - IBM might never have fixed the flaws
  - A lot learned about FPGAs as attack tools in general

Make Your Own!

http://buy.altera.com/ecommerce/dkc.html

Publicity website, including source files…

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

Academic Papers

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