“Investigations of Power Analysis Attacks on Smartcards”*

Thomas S. Messerges
Motorola Labs
Motorola
tomas@ccrl.mot.com

Ezzy A. Dabbish
Motorola Labs
Motorola
dabbish@ccrl.mot.com

Robert H. Sloan[1]
Dept. of EE and Computer Science
University of Illinois at Chicago
sloan@eecs.uic.edu

* Proceedings of USENIX Workshop on Smartcard Technology, May 1999, to appear.

---

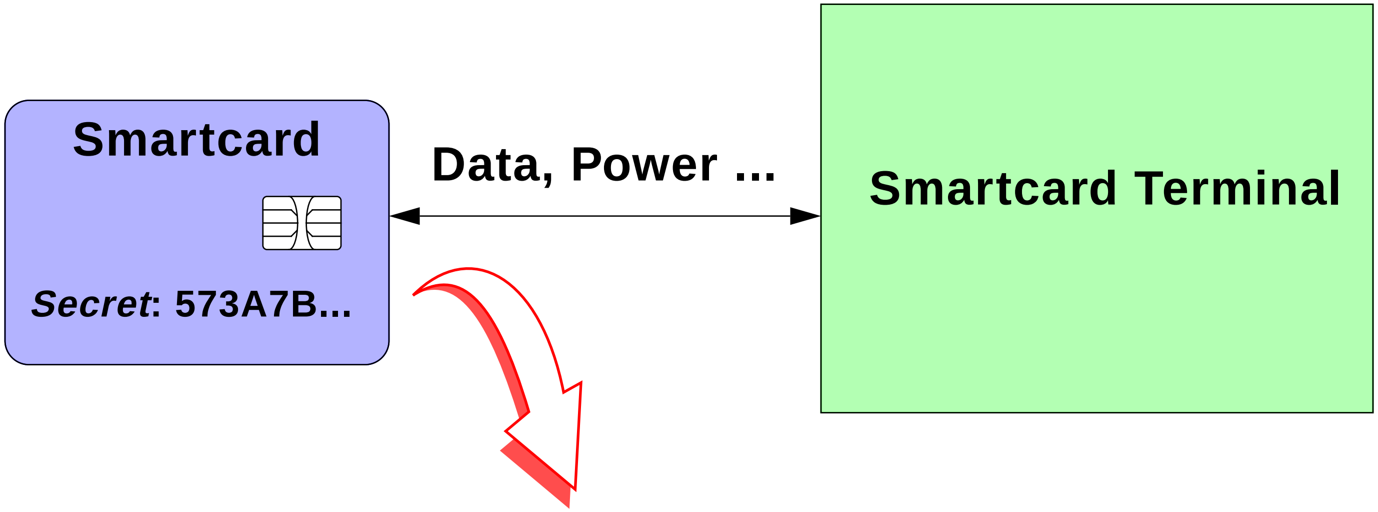
[1] Partially supported by NSF Grant CCR-9800070.

Summary of Presentation

- Motivation for this research – review underlying issues
- Review power analysis attacks:
  - Simple Power Analysis (SPA)
  - Differential Power Analysis (DPA)
  - Show results
- Noise analysis results – Statistical model
- Introduce multiple-bit DPA and results
- Discuss design goals for countermeasures
- Future work and concluding remarks
- Presentation slides available at:

http://www.eecs.uic.edu/~tmesserg/papers.html

Problem Description



Smartcard

***Secret*: 573A7B...**

Data, Power ...

Smartcard Terminal

<u>Power Dissipation</u>: Can leak information about the *Secret* !

Attackers That Learn A Smartcard's *Secret* Key

- Clone cards
- Make fraudulent payments
- Impersonate others
- Access private information (i.e. medical records)

Related Attacks

- Timing Measurements
- Fault Insertion
- EM Emissions
- Other “side-channel” attacks (Kelsey, et. al. ESORICS ‘98)

Motivations for Our Research

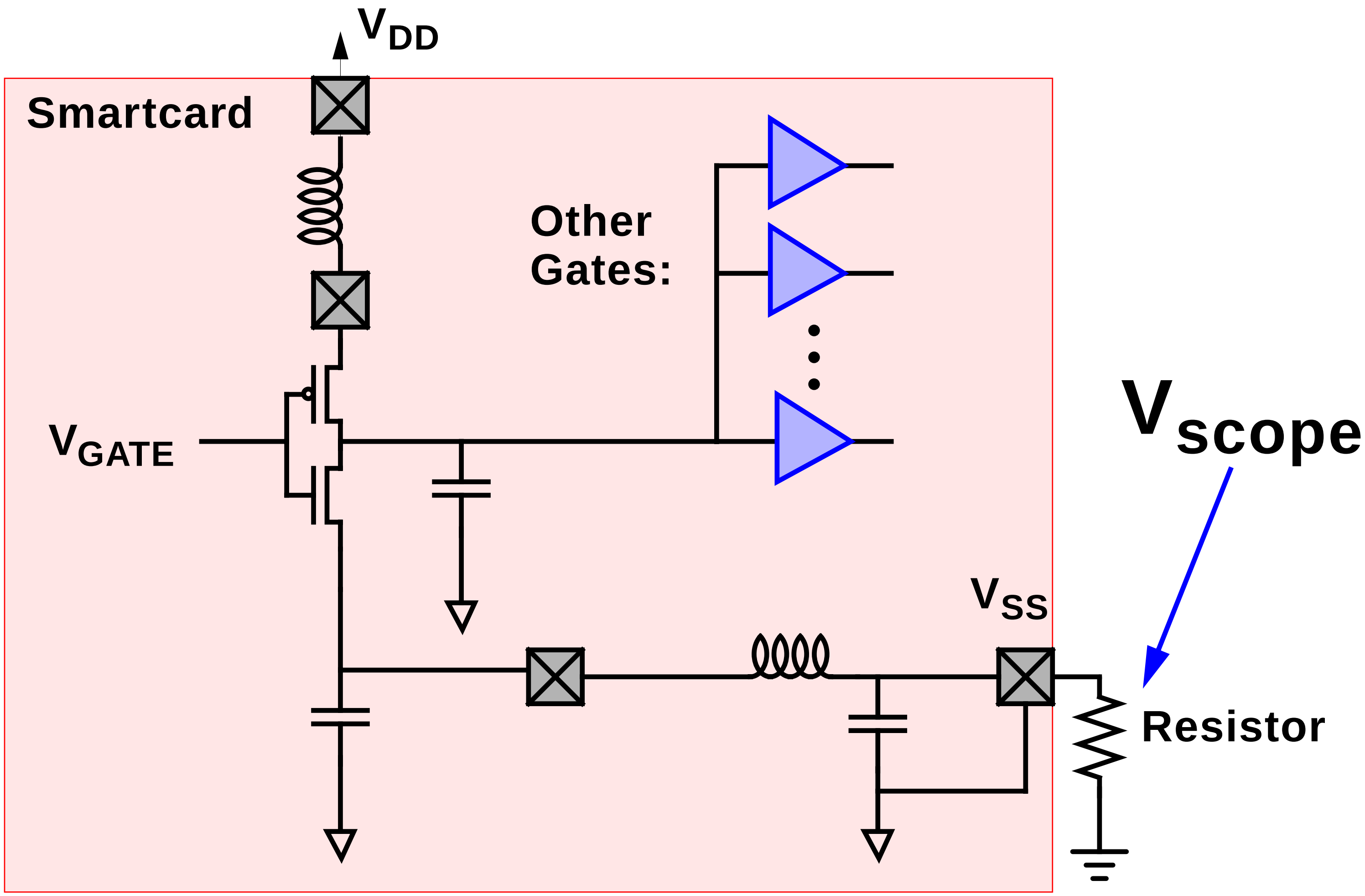
- Understand principles of how power analysis works
- Evaluate existing power analysis attacks
- Examine effectiveness of new, more powerful attacks
- Develop a statistical model to describe power analysis attacks
- Quantify the extent of a threat that actual power analysis attacks may pose
- Evaluate countermeasures to attacks
- **Develop more secure smartcards**

Previous Power Analysis Work

P. Kocher, J. Jaffe, and B. Jun:
"Introduction to Differential Power Analysis and Related Attacks," http://www.cryptography.com/dpa/technical, 1998.

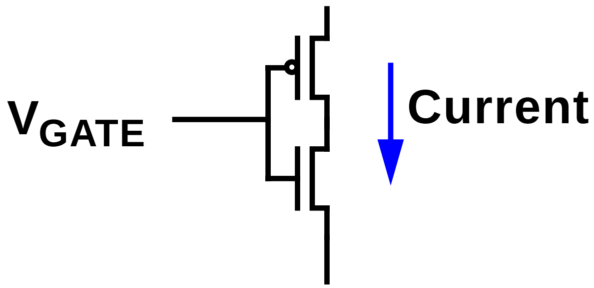
J. Kelsey, B. Schneier, D. Wagner, and C. Hall:
"Side Channel Cryptanalysis of Product Ciphers," in Proceedings of *ESORICS '98*, Springer-Verlag, September 1998, pp. 97-110.

Measuring Power Consumption

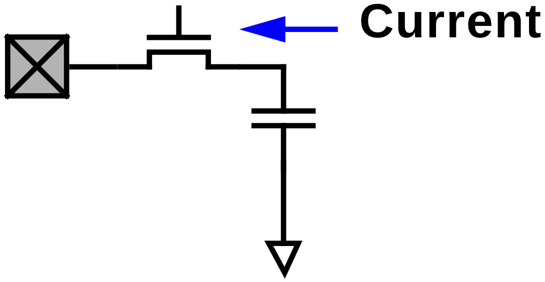


Information from Power Consumption

Dynamic Current:



Discharge Current:

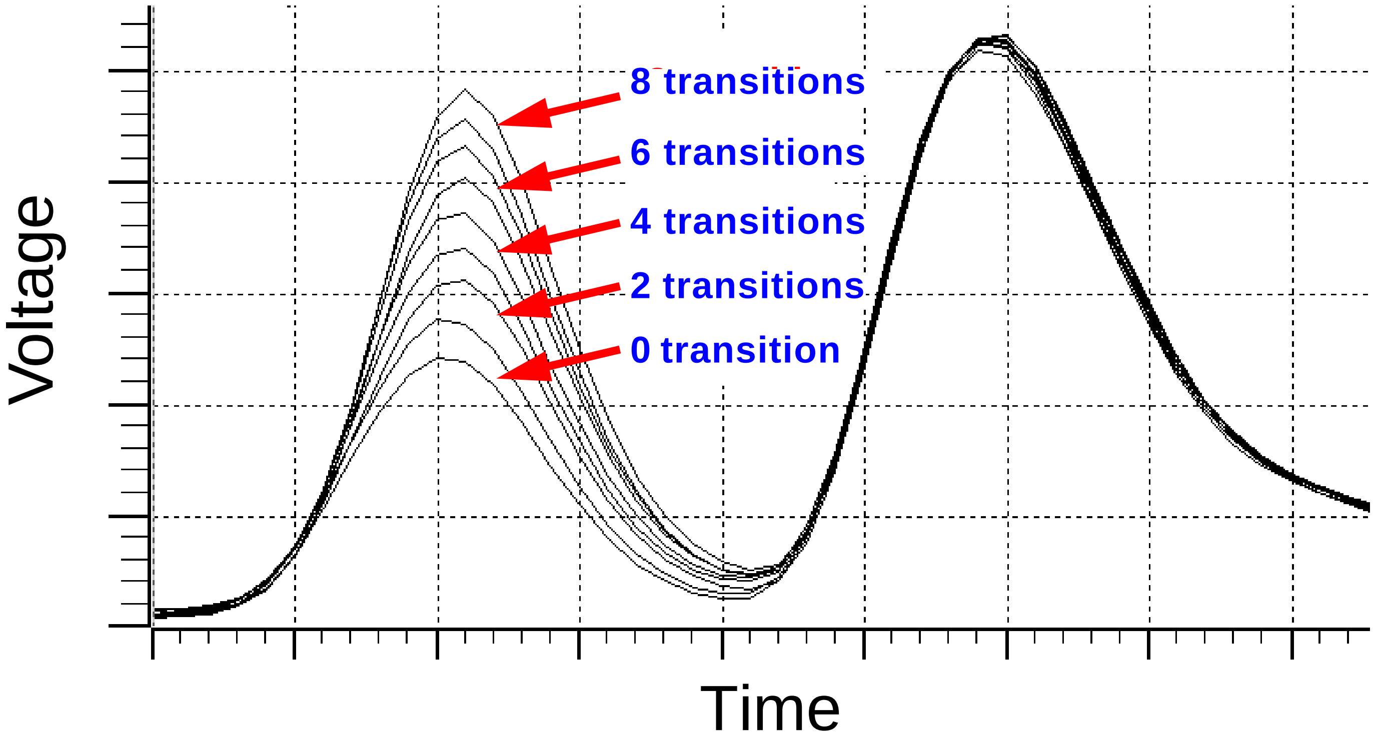


Information Leaked

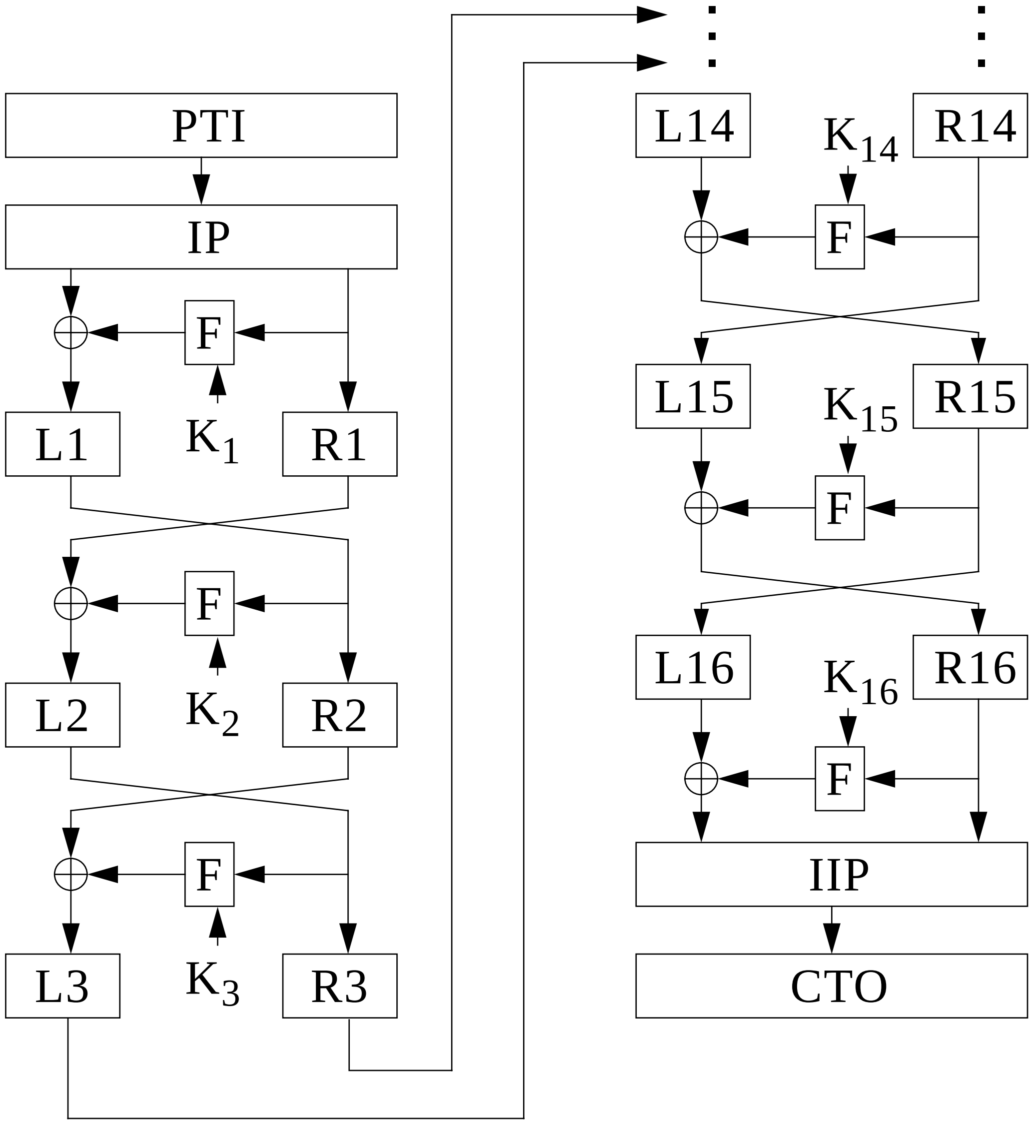
Transition Count

Hamming Weight

Example of Power Consumption Information Leakage

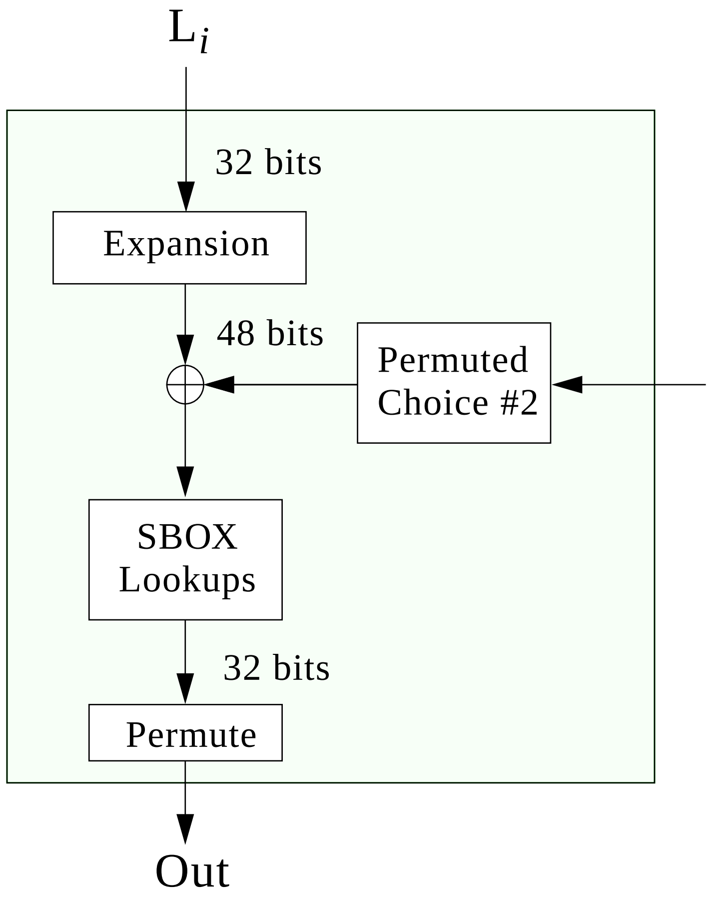


Review of DES

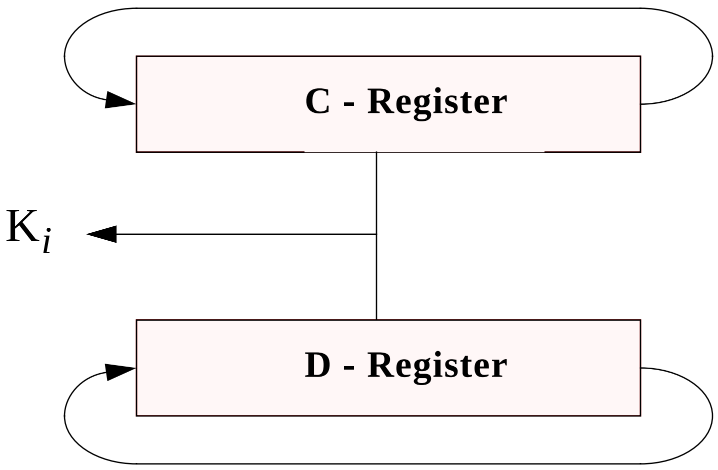


Review of DES (continued)

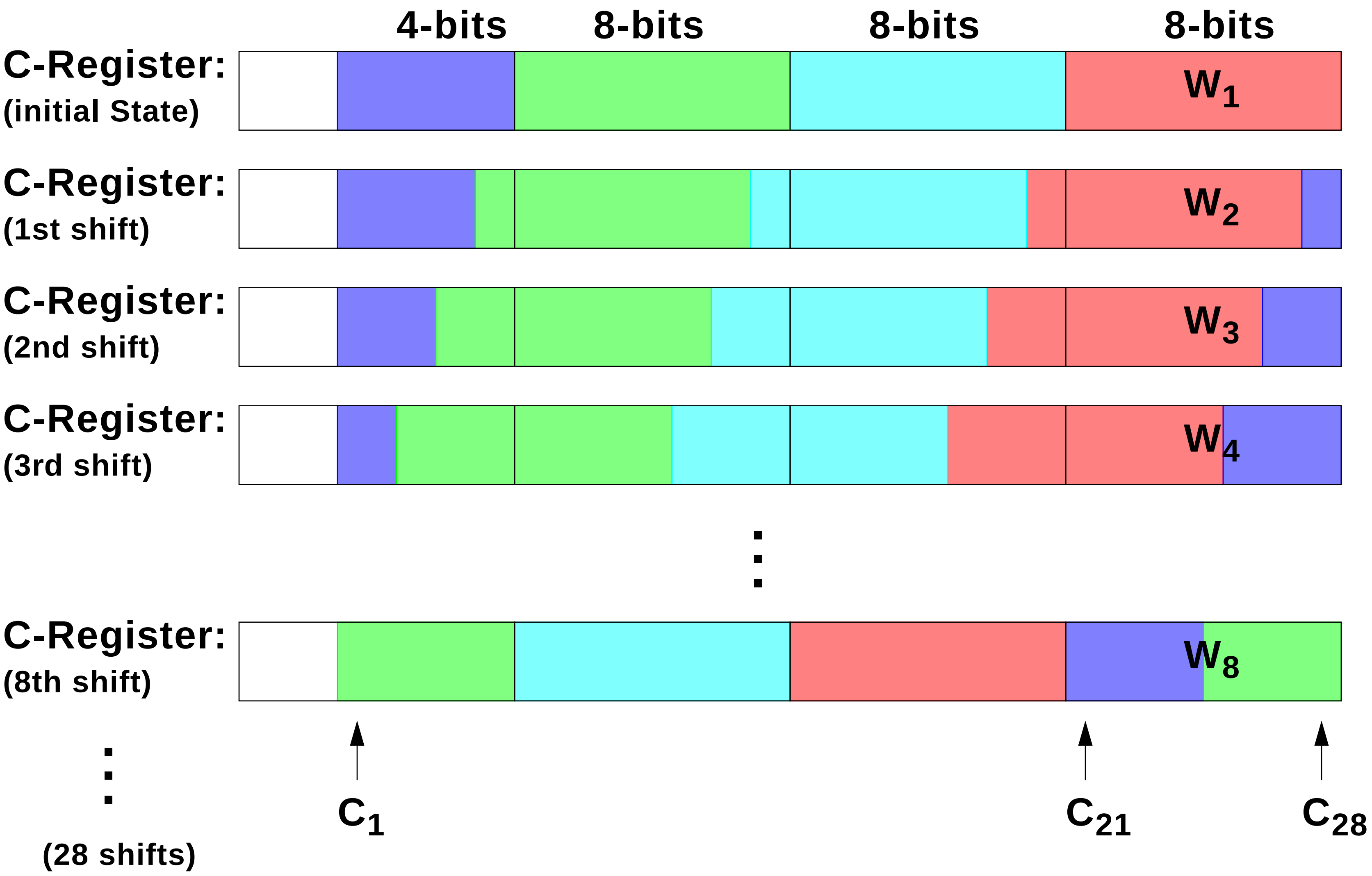
The *F* Function



Subkey Generation



Using Hamming Weight Data to Break DES



C-Register:
(initial State)

C-Register:
(1st shift)

C-Register:
(2nd shift)

C-Register:
(3rd shift)

C-Register:
(8th shift)

(28 shifts)

Using Hamming Weight Data to Break DES

$$A\vec{k} = \vec{w}$$

$$\begin{array}{cc} & \xleftrightarrow{\quad 28 \quad} \\ 28 \updownarrow & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ & & & & \vdots & & & & & & & & & & \\ 0 & 0 & 0 & 0 & 0 & \cdots & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix} \end{array} \begin{bmatrix} C_1 \\ C_2 \\ C_3 \\ C_4 \\ C_5 \\ C_6 \\ \vdots \\ C_{28} \end{bmatrix} = \begin{bmatrix} W_1 \\ W_2 \\ W_3 \\ W_4 \\ W_5 \\ W_6 \\ \vdots \\ W_{28} \end{bmatrix}$$

28 Equations, 28 Unknowns – Solve for C-register Bits

Simple Power Analysis – Summary

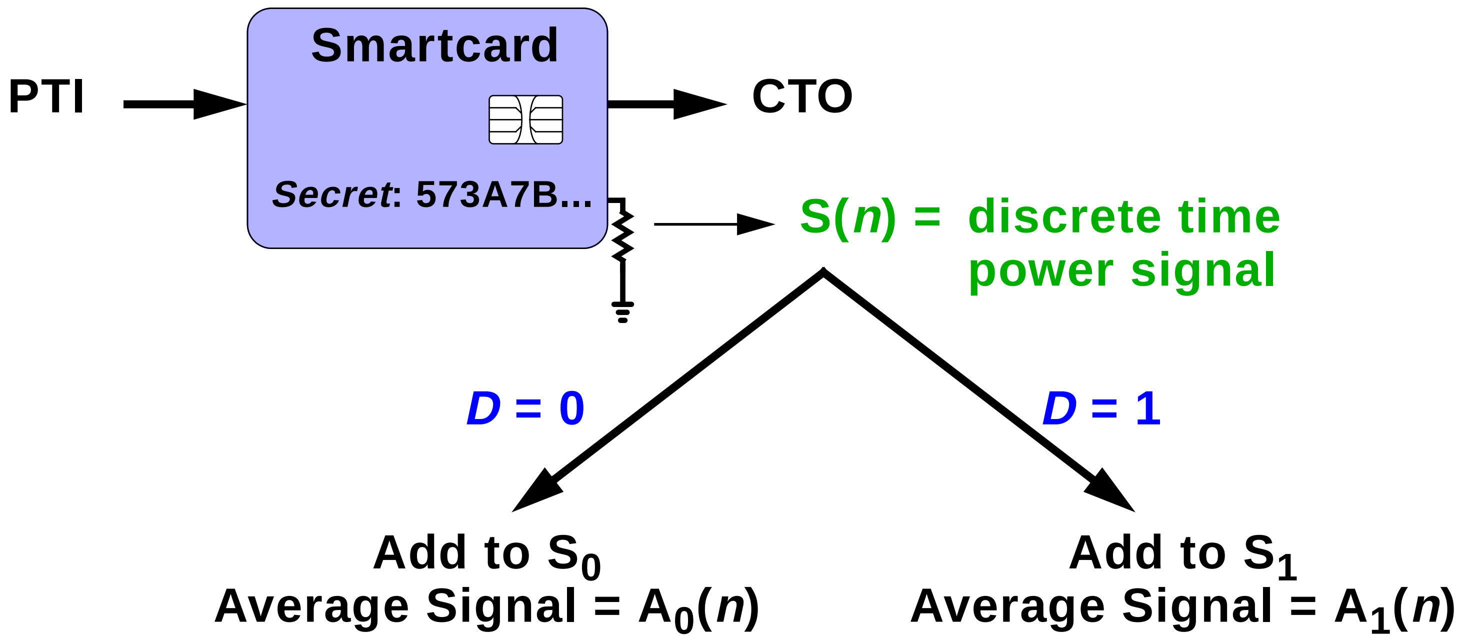
1. Run a single encryption
2. Acquire power consumption data
3. Convert power data to Hamming weight data
4. Solve for the C and D bits (i.e., the key bits)

Conclusions

- Adversary needs knowledge of the implementation to mount the attack
- Easy to protect against – (reduce power emissions, prevent attacker from learning implementation ...)

Differential Power Analysis (DPA) (Kocher, et. al.)

- Knowledge of implementation is not required
- Statistical approach “amplifies” power information



PTI → **Smartcard** → CTO

***Secret*: 573A7B...**

$S(n)$ = discrete time power signal

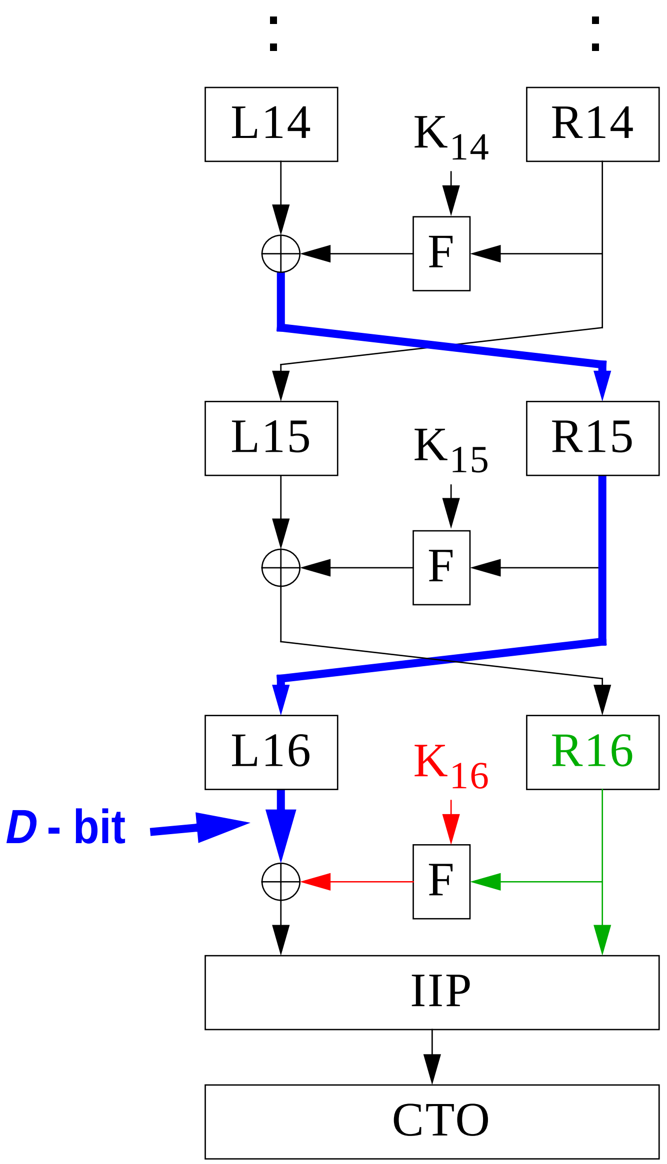
***D* = 0**

Add to S_0
Average Signal = $A_0(n)$

***D* = 1**

Add to S_1
Average Signal = $A_1(n)$

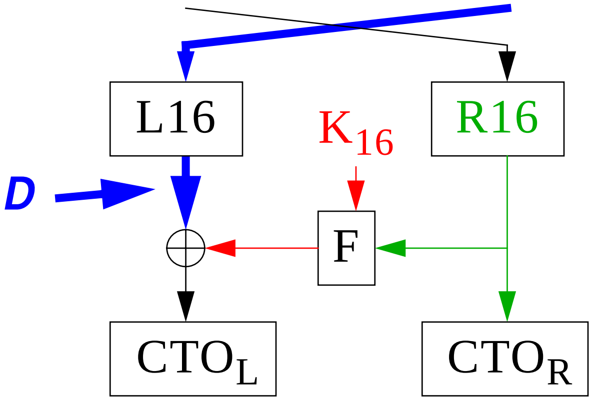
Define: DPA Bias Signal = $T(n) = A_1(n) - A_0(n)$



Review DPA Attack on DES

1. Guess 6-bits of K_{16}
2. Initialize: $A_1 = A_0 = 0$
3. Get a CTO and power trace
4. Reverse-calculate the *D* - bit
5. If ($D = 1$) then
    add power trace to A_1
   else
    add power trace to A_0
6. If not enough averages goto 3.
7. DPA Bias Signal: $T = A_1 - A_0$

Defining the *D* Function



$$CTO_L = D \oplus SBOX(K16 \oplus R16)$$

Solve for *D*

$$D = CTO_L \oplus SBOX(K16 \oplus CTO_R)$$

- Smartcard must calculate *D* at some time – say at time j^*
- The expected power consumption when $D=1$ is greater than when $D=0$:

$$E[\ S(j^*) \mid D = 1\ ] > E[\ S(j^*) \mid D = 0\ ]$$

- A_0 and A_1 are estimates of the expected power consumption:

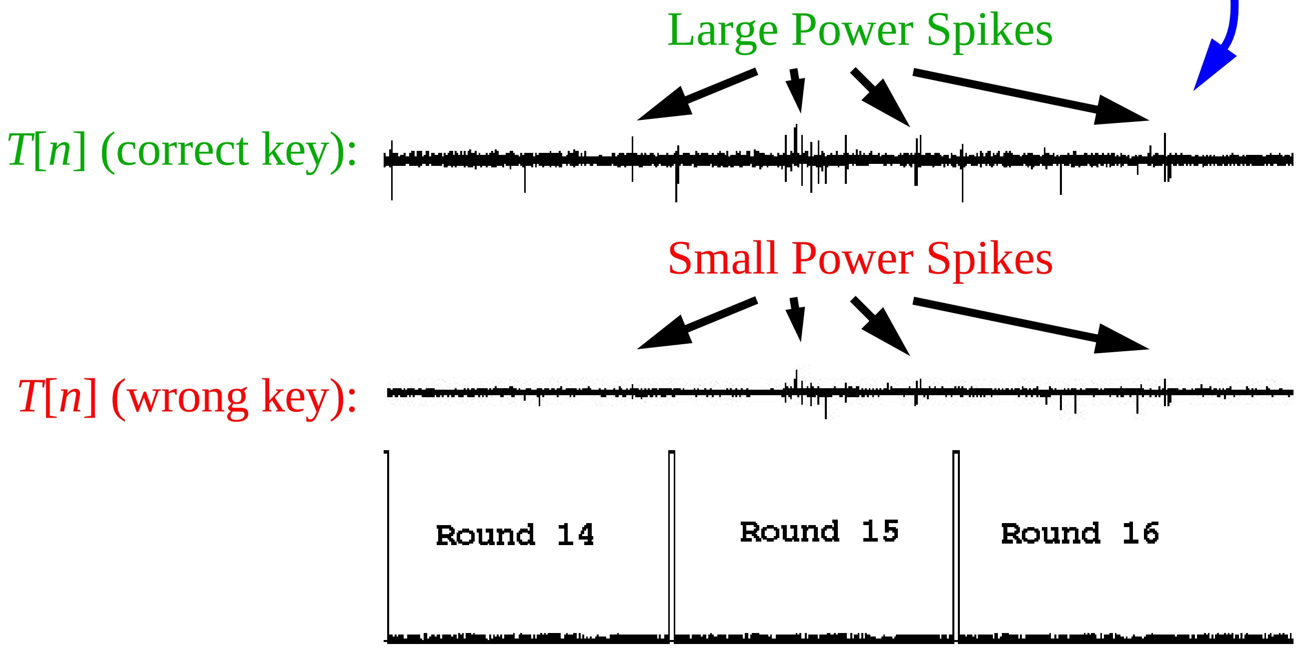
$$A_0 \approx E[\ S(n) \mid D = 0\ ] \text{ and } A_1 \approx E[\ S(n) \mid D = 1\ ]$$

The Power Bias Signal

- The power bias signal will have a spike at time *j** :

$$T(n) = A_1 - A_0 = \begin{cases} \varepsilon & n = j^* \\ 0 & n \neq j^* \end{cases}$$

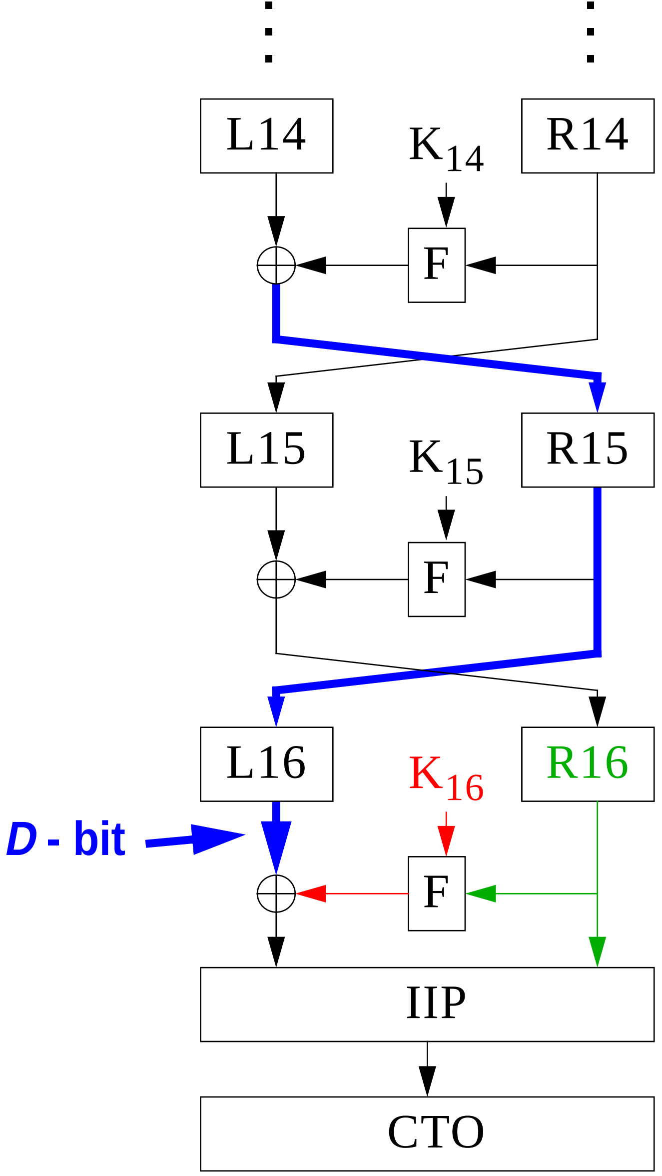
Size of Power Spikes = ε


Large Power Spikes

T[*n*] (correct key):

Small Power Spikes

T[*n*] (wrong key):

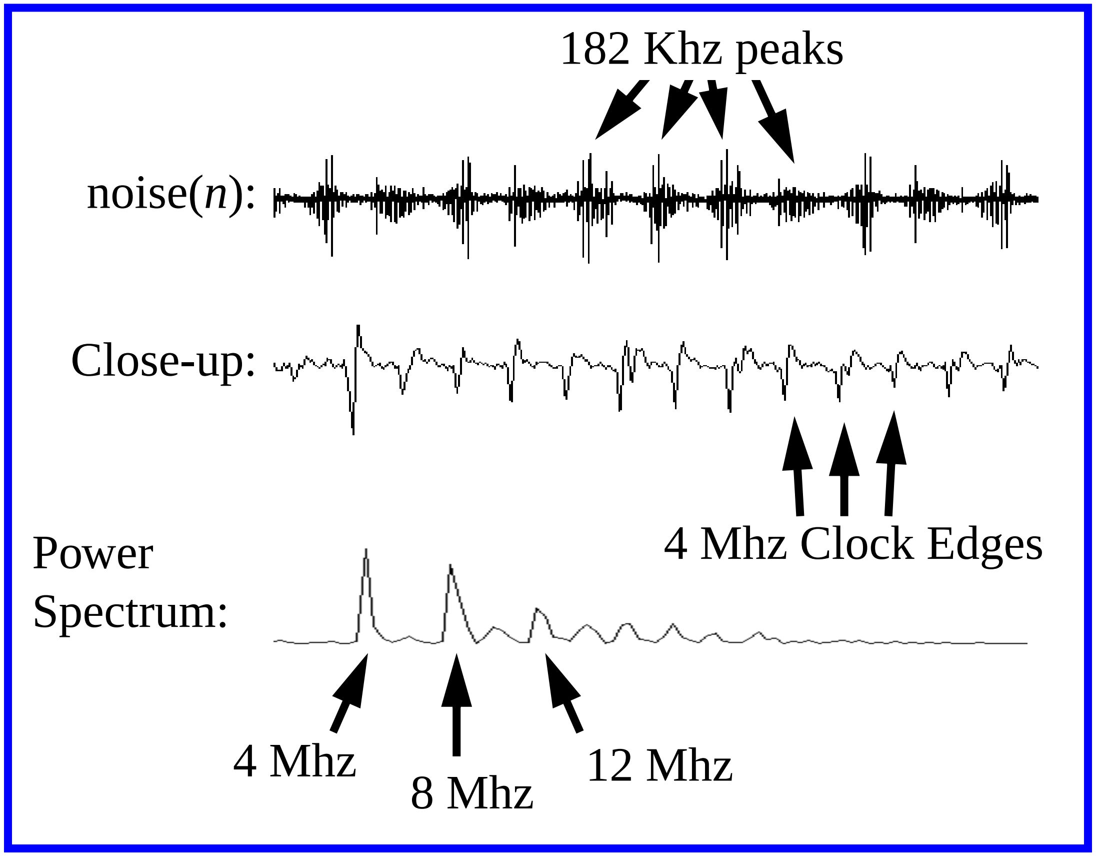


Review DPA Attack on DES

1. Guess 6-bits of K_{16}
2. Initialize: $A_1 = A_0 = 0$
3. Get a CTO and power trace
4. Reverse-calculate the *D* - bit
5. If (*D* = 1) then
   add power trace to A_1
   else
   add power trace to A_0
6. If not enough averages goto 3.
7. DPA Bias Signal: $T = A_1 - A_0$

DPA Signal Noise

$$\text{noise}(n) = E\left[S(n)\right] - S(n)$$



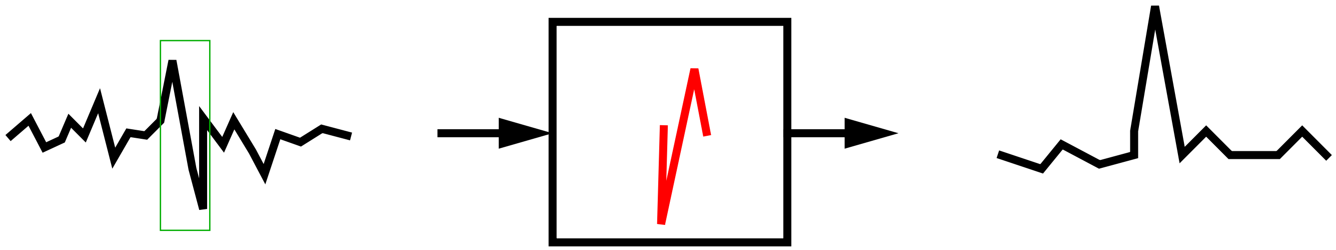
- 182 KHz “beat” frequency
- Noise at clock edges
- Quantization noise
- External noise
- Internal noise

- Algorithm noise

$$x(j) = \sum_i \boldsymbol{c}_i F(j - \boldsymbol{j}_i)$$

Filtering the Noise

- Averaging reduces the noise
- Use “Matched Filter” to reduce the noise



Original DPA Signal

(SNR = 17.3)

Matched Filter

Filtered DPA Signal

(SNR = 23.2)

- Improvement is small
- Use knowledge of noise properties to get cleaner DPA signal (i.e. noise is maximum at clock edges)

Averaging is the Best Way to Reduce Noise

Noise Signal:

$$E\left[T[j] \mid (j \neq j^*)\right] = 0$$

$$\operatorname{var}\left[T[j] \mid (j \neq j^*)\right] = \frac{4\sigma^2 + \alpha m \varepsilon^2}{N}$$

DPA Signal:

$$E\left[T[j^*]\right] = \varepsilon$$

$$\operatorname{var}\left[T[j^*]\right] = \frac{4\sigma^2 + (m-1)\varepsilon^2}{N}$$

Theoretical Voltage SNR $= \dfrac{\sqrt{N}\varepsilon}{\sqrt{8\sigma^2 + \varepsilon^2(\alpha m + m - 1)}}$

Noise Model vs. Experimental Results

$$\text{Theoretical Voltage SNR} = \frac{\sqrt{N}\,\varepsilon}{\sqrt{8\sigma^2 + \varepsilon^2(\alpha m + m - 1)}}$$

$\sigma = 7.5\ \mathrm{mV}$ $\varepsilon = 6.5\ \mathrm{mV}$ $m = 8$

$N = 1000$ $\alpha = 0$

Theoretical Voltage SNR = 7.5

⇩

Experimental Voltage SNR = 7 to 10

How Many Samples Are Needed?

1. Solve for *N*:

$$N = \frac{8\sigma^2 + \varepsilon^2(\alpha m + m - 1)}{\varepsilon^2 \cdot SNR^2}$$

2. Determine parameters for a specific smartcard:

$\sigma = 7.5\text{ mV}$ $\varepsilon = 6.5\text{ mV}$ $m = 8$ $\alpha = 0$

3. Assign *SNR*:

$SNR = 0.67$ ← **Median for Gaussian Distributed Noise**

4. Calculate *N*:

Theoretical Minimal Number of Samples: *N* = 40

Maximizing the DPA Signal

$$S_0 = \left\{ S_{ij} \middle| D(.,.,.) = 0^d \right\}$$

$$S_1 = \left\{ S_{ij} \middle| D(.,.,.) = 1^d \right\}$$

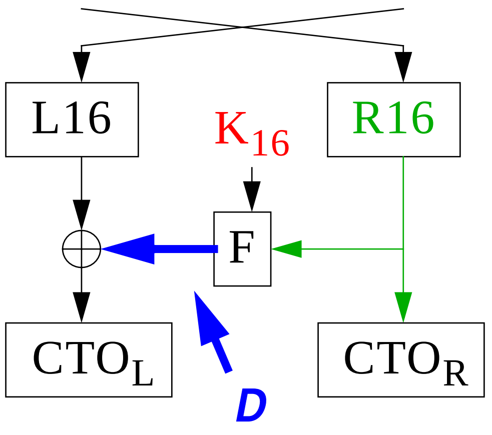
$$S_2 = \left\{ S_{ij} \middle| S_{ij} \notin S_0, S_1 \right\}$$

Multiple-Bit *D* Function

Force S_0 and S_1 to exhibit greater power differences, thus, increasing the SNR

Toss out signals that do not give a maximal power difference

4-Bit DPA Description



D = SBOX(K16 ⊕ R16)

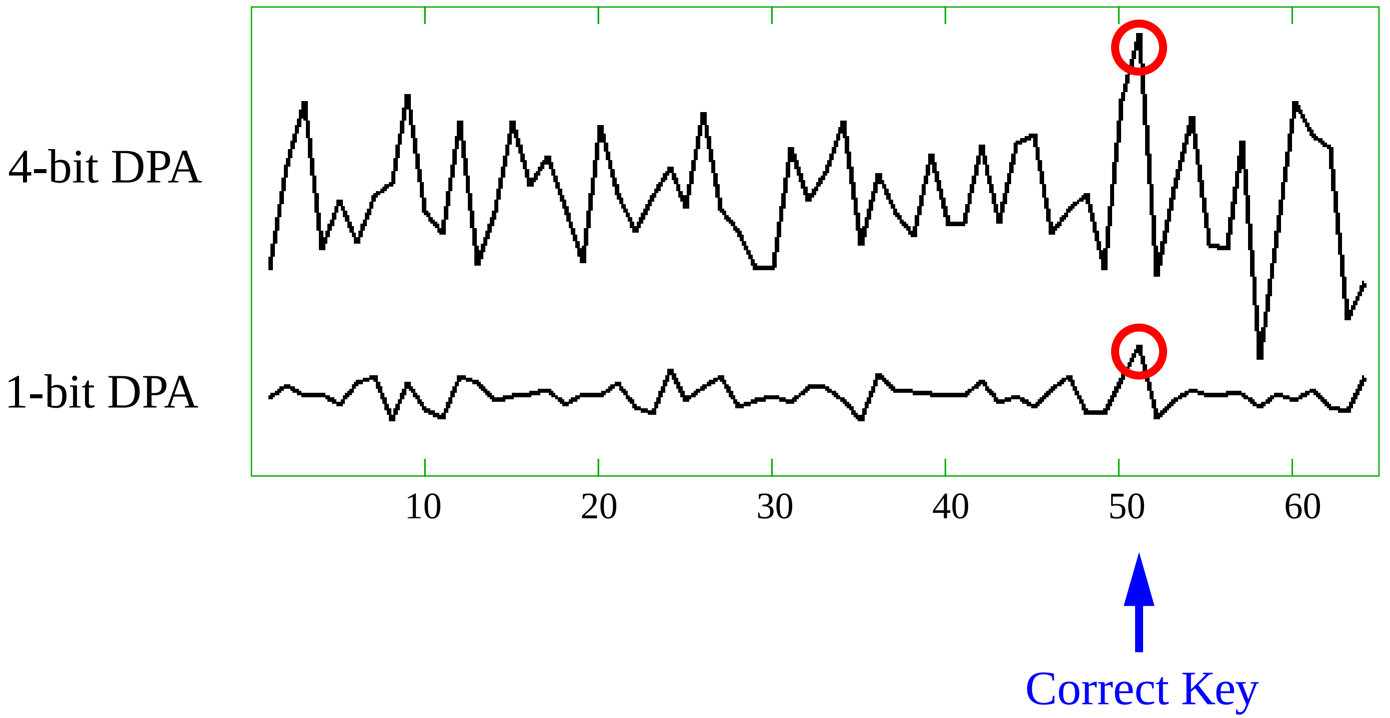
⇩ R16 is part of CTO_R

D = SBOX(K16 ⊕ CTO_R)

4-Bit DPA Attack

1. Guess 6-bits of K_{16}
2. Initialize: $A_1 = A_0 = 0$
3. Get a CTO and power trace
4. Reverse-calculate the D
5. If (D = 1111) then
    add power trace to A_1
   else if (D = 0000) then
    add power trace to A_0
   else
    do nothing
6. If not enough averages goto 3.
7. DPA Bias Signal: $T = A_1 - A_0$

DPA Bias Signal Level for Different Key Guesses



Other Strong Types of DPA Attacks

Compressed SBOX Table:

S1_S2[0] = 0xEF

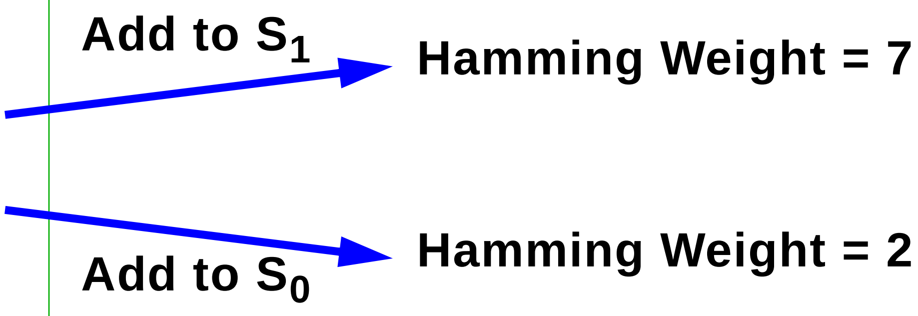
S1_S2[1] = 0x03

S1_S2[2] = 0x41

⋮

S7_S8[62] = 0x0F

S7_S8[63] = 0xE3



Add to S_1 → Hamming Weight = 7

Add to S_0 → Hamming Weight = 2

Multiple-Bit DPA can result in even larger power biases if the SBOX data is stored in a compressed table.

The addresses of the SBOX data (rather than the actual data) may also be used for an attack!

Our DPA Attack Results

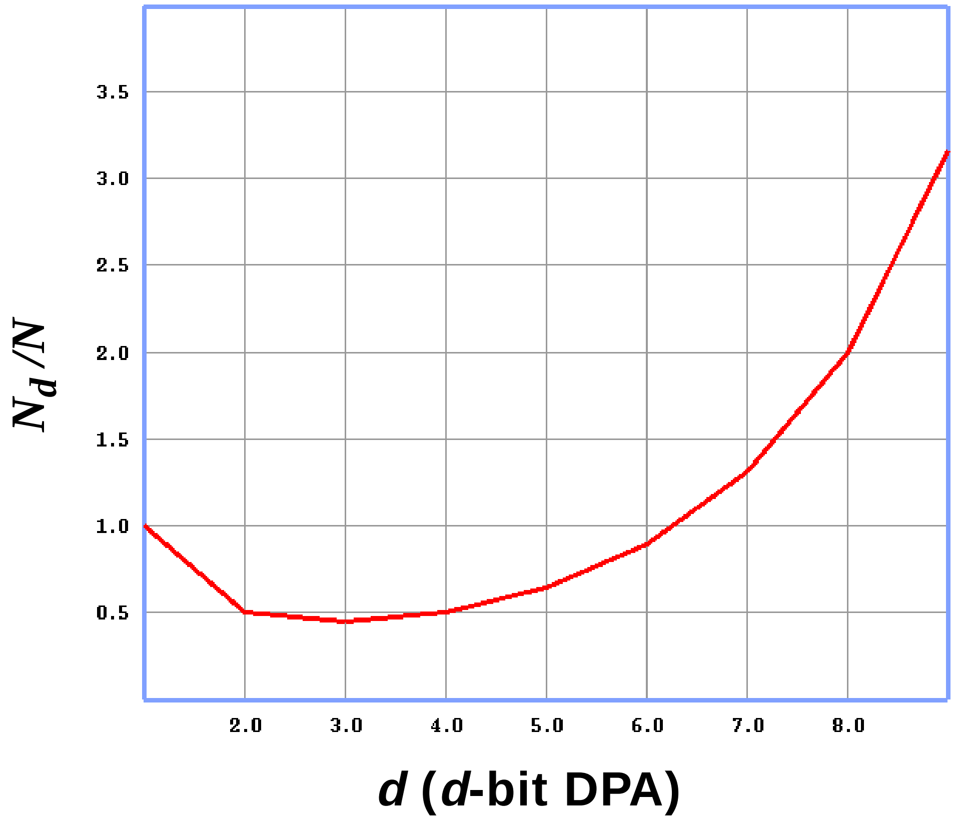
Attack Type:	1-bit DPA	4-bit DPA	8-bit DPA	Address DPA
Signal Level:	9.3 mV	38.5 mV	79.5 mV	74.4 mV

- Voltage SNR is 8 times larger
- Attacker needs fewer power signals to break the system

Diminishing Returns for Multiple-Bit DPA

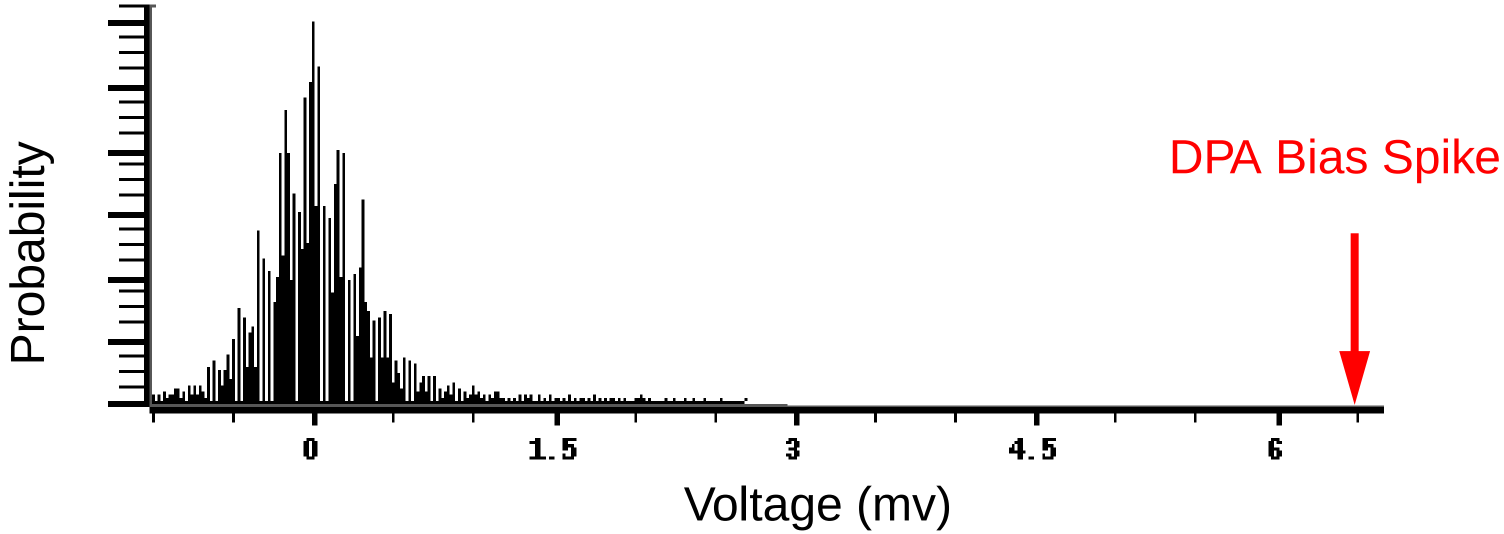
Attacker needs more power signals:

$$N_d = 2^{d-1}N/d^2$$



Design Goal for Hiding the DPA Power Spike

$N = 1300 \qquad \varepsilon = 6.5\text{ mV} \qquad m = 8$



Need to expand noise distribution

and/or

Reduce DPA bias spike

Future Work

- Examine other symmetric key algorithms
- Examine public-key algorithms
- Design modified algorithms
- Develop more advanced modeling methods
- Design countermeasures

Summary of Results

- Source of power biases is examined
- Demonstrated successful power analysis attacks
- Proved multiple-bit DPA leads to a new and more powerful attack
- Modeled the noise characteristics

Designers need to consider the power analysis attacks outlined in this paper when designing secure smartcard systems