IPA Better Life with IT

ハードウェア脆弱性評価の最新技術動向に関するセミナー

― CHES/FDTC参加報告 ―

2014年12月8日

独立行政法人　情報処理推進機構
技術本部 セキュリティセンター

目次

- ハードウェアセキュリティトピックの紹介
- CHES2014の発表の概要紹介
- FDTC2014の発表の概要紹介
- IPAの取り組み

ハードウェアセキュリティトピックの紹介

攻撃方法の分類

- Non-Invasive Attack
  - チップ内部への物理的侵入を伴わない攻撃
  - 例: サイドチャネル解析
- Invasive Attack
  - チップ内部への物理的侵入を伴う攻撃
  - 例: プロービング、回路改変
- Semi-Invasive Attack
  - パッケージの開封(穴開け)程度は行うが、パシベーション層までは破壊しない
  - 例: レーザー攻撃

サイドチャネル攻撃 (Side Channel Analysis)

- 暗号機能を実装したハードウェア(スマートカード等)の動作中に、そのハードウェアの状態を観測することで得られる情報を利用して、暗号鍵といった秘密情報の復元を試みる
  - 電力解析(Power Analysis)
    - ハードウェアの消費電力を測定し、その情報から解析する。
      - SPA (Simple Power Analysis): 1つの電力波形を直接調べる。IC内の処理のパターンを見る。
      - DPA (Differential Power Analysis): 多数の電力波形を統計処理して解析する。消費電力のデータ依存部分を抽出することができ、また、ノイズを軽減することができる。
    - CMOS半導体の特性上、トランジスタのスイッチング(0→1, 1→0)が起こる時に消費電力が大きくなることを利用。
  - 電磁波解析(Electromagnetic Analysis)
    - 動作中のハードウェアのからの漏洩電磁波から解析する。電力解析同様、1つの波形から解析するSEMA、多数の波形から解析するDEMAがある。
    - 局所的なアクティビティを検知することが可能。

AESアルゴリズム

♦ 暗号化処理の流れ

```
Cipher(byte in[4*Nb], byte out[4*Nb], word w[Nb*(Nr+1)])
begin   平文              暗号文              拡大鍵
     byte state[4,Nb]                  //内部変数(4行, Nb列の行列)

     state = in

     AddRoundKey(state, w[0, Nb-1]) //

     for round = 1 step 1 to Nr-1
          SubBytes(state) //
          ShiftRows(state) //
          MixColumns(state) //
          AddRoundKey(state, w[round*Nb, (round+1)*Nb-1])
     end for

     SubBytes(state)
     ShiftRows(state)
     AddRoundKey(state, w[Nr*Nb, (Nr+1)*Nb-1])

     out = state
end
```

Nb:4,
Nr:10,12,14
　for 128, 192, 256-bit key,
w:拡大鍵, 要素数Nb *(Nr+1)

SubBytes:行列要素の置換

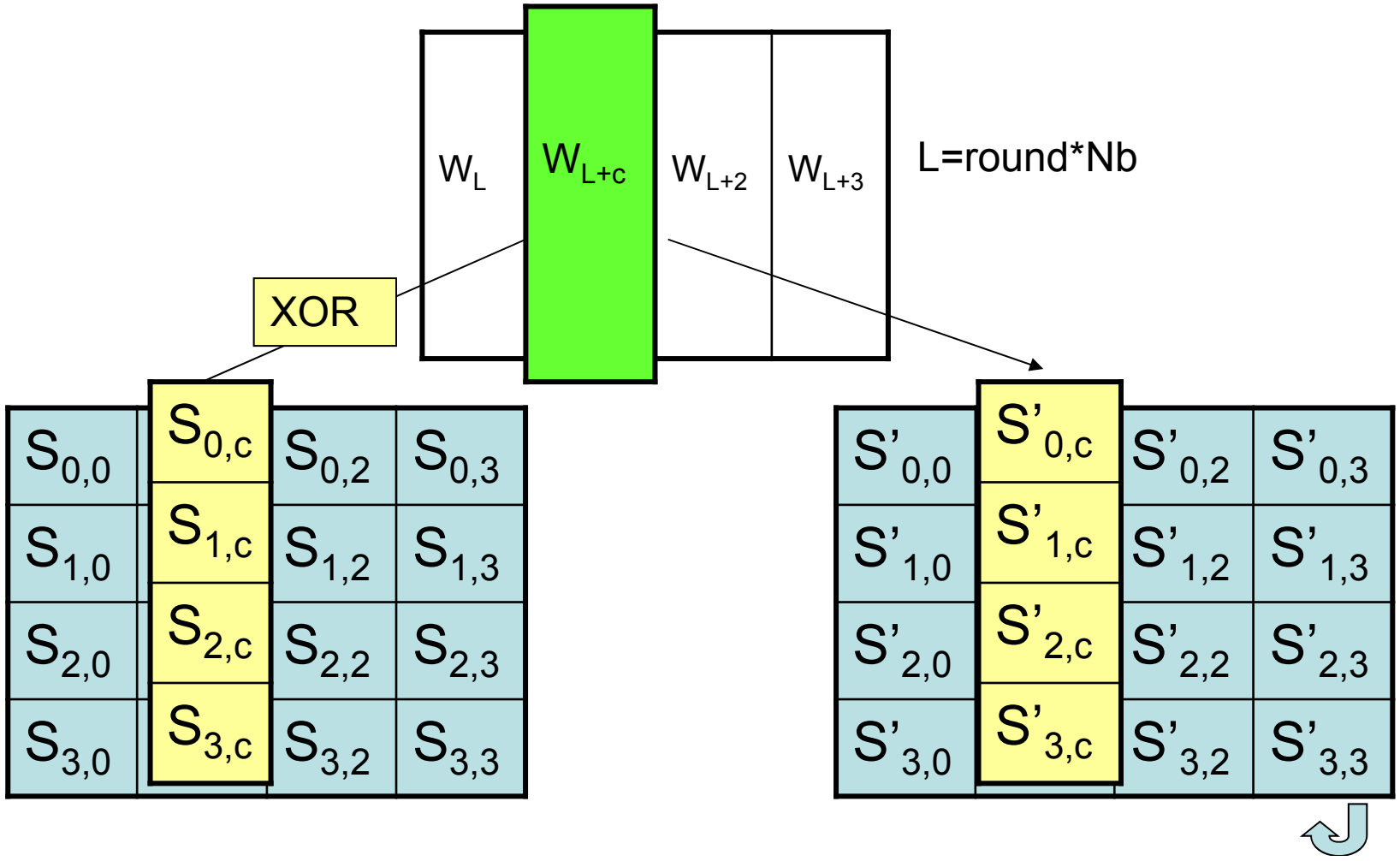
ShiftRows:
行単位の左シフト処理

MixColumns:
列ベクトル単位のデータの変換

AddRoundKey:
列ベクトルと拡大鍵wとのXOR演算

AES：AddRoundKeyの処理

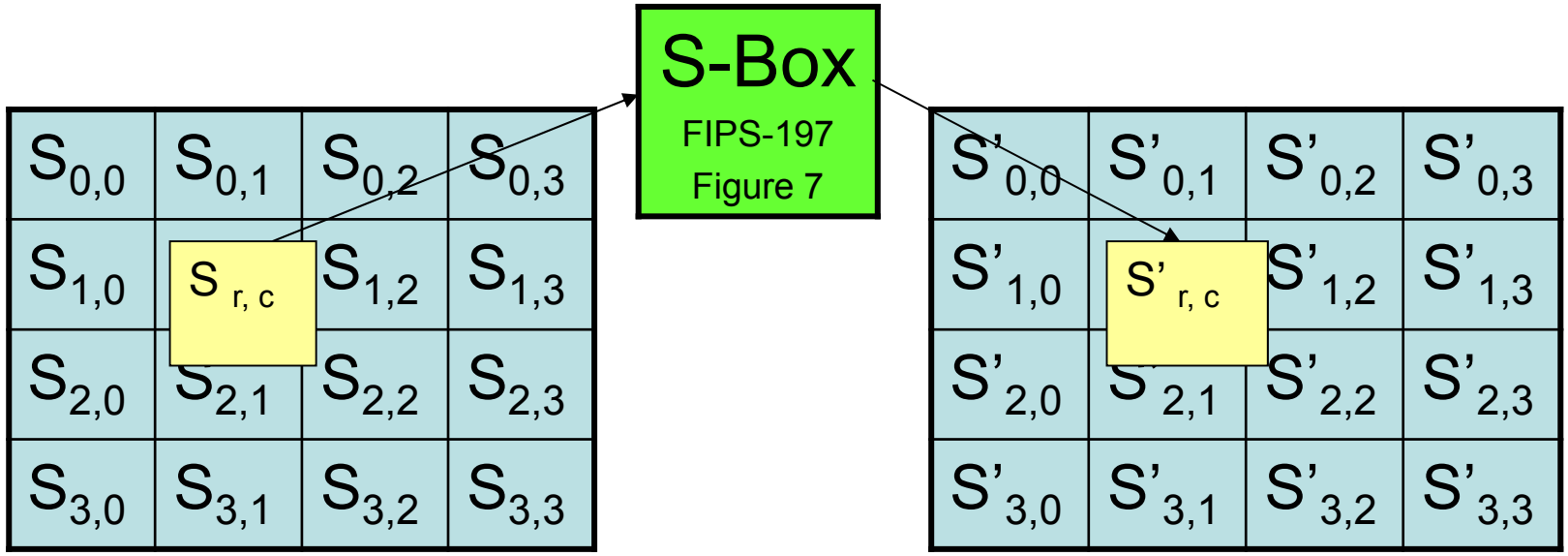
- 4ブロックを1列として，列ごとに拡大鍵とXOR処理。



AES：SubBytesの処理

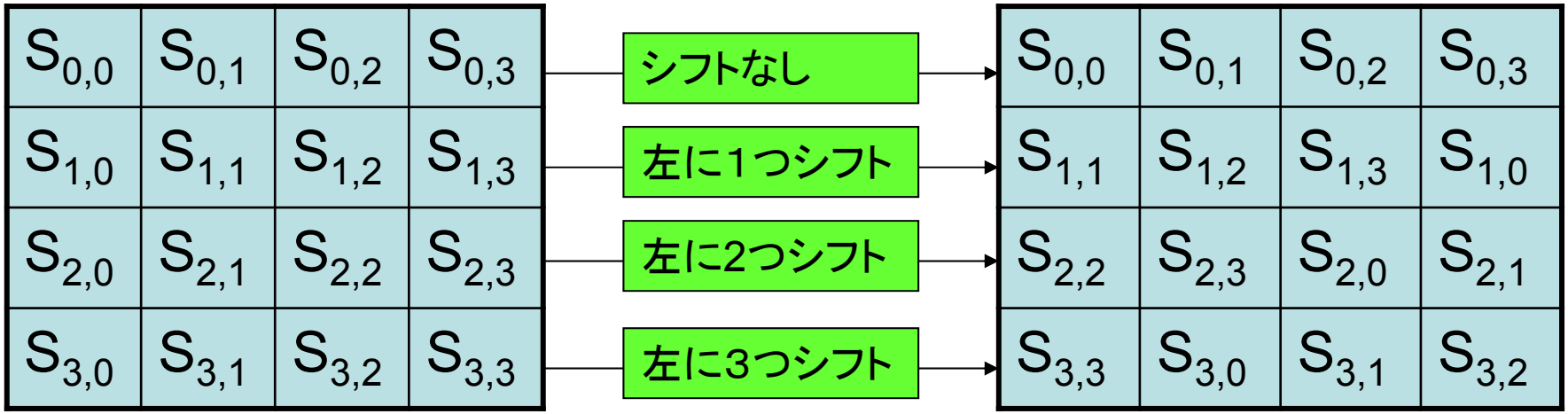
IPA

- 128ビットのデータを1バイト（8ビット）ごとに16のサブブロックに分割。
- 各ブロックでは1バイトの入力データを1バイトの出力データへ置換。



AES：ShiftRowsの処理

- 4ブロックを1行として，行ごとに左シフト処理。

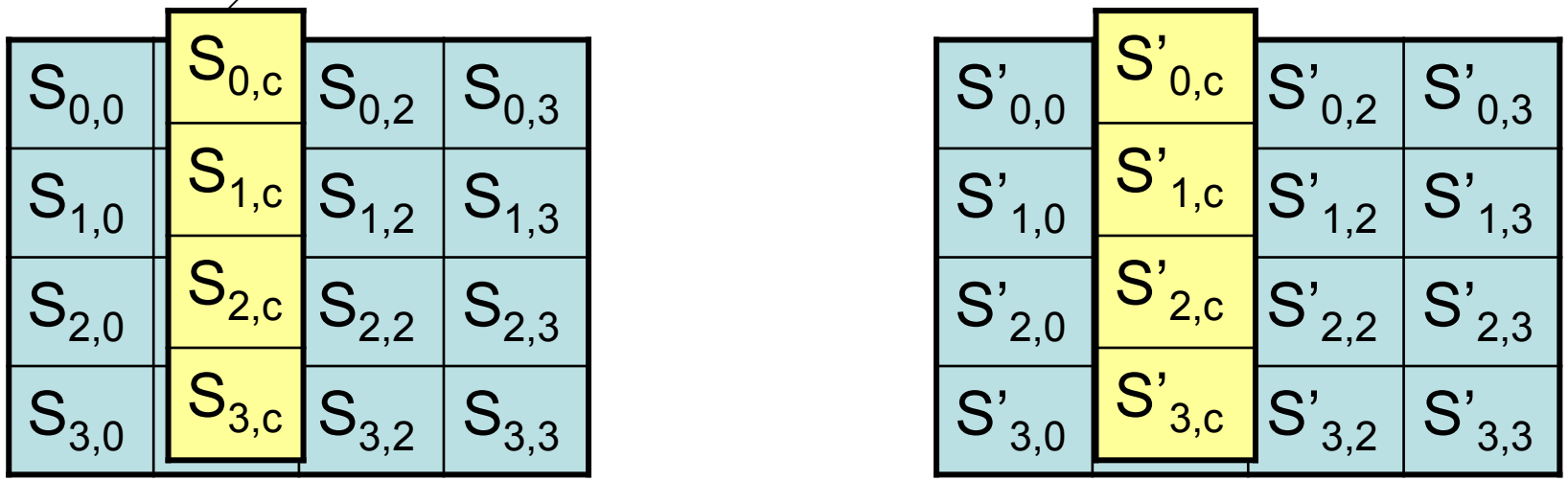


AES：MixColumsの処理

IPA

•4ブロックを1列として，列ごとに列ベクトルの変換。

$$\begin{pmatrix} S'_{0,c} \\ S'_{1,c} \\ S'_{2,c} \\ S'_{3,c} \end{pmatrix} = \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \begin{pmatrix} S_{0,c} \\ S_{1,c} \\ S_{2,c} \\ S_{3,c} \end{pmatrix}$$



DPAの例1

- 初期のDPA: DoM (Difference of Mean)

鍵の先頭1バイト=00と仮定

平文(の先頭1バイト)	第1ラウンドのsboxの出力の先頭1バイト	MSB
7d	21	0
7c	10	0
6a	02	0
da	57	0
17	f0	1
…	…	…

鍵の先頭1バイト=01と仮定

平文(の先頭1バイト)	第1ラウンドのsboxの出力の先頭1バイト	MSB
7d	10	0
7c	21	0
6a	7f	0
da	b9	1
17	47	0
…	…	…

1. 平文をランダムに変化させて暗号化を行い、消費電力を測定する
2. 中間値のあるビットに注目し、それが0か1かにとって電力波形を分類する。それを各鍵仮説(AES鍵の先頭1バイトの場合、256通り)に対して行う。
3. 2群に分けた電力波形について、それぞれ平均を取る
4. 誤った鍵仮説に対しては、平均値の差がゼロに近い値になるが、正しい鍵仮説に対しては、平均値の差が大きい値になると考えられるので、これによって正しい鍵の値が判明する

DPAの例2

- CPA (Correlation Power Analysis: 相関係数を使用する)

鍵の先頭1バイト=00と仮定

平文(の先頭1バイト)	第1ラウンドのsboxの出力の先頭1バイト	HW
7d	21	2
7c	10	1
6a	02	1
da	57	5
17	f0	4
…	…	…

鍵の先頭1バイト=01と仮定

平文(の先頭1バイト)	第1ラウンドのsboxの出力の先頭1バイト	HW
7d	10	1
7c	21	2
6a	7f	7
da	b9	6
17	47	4
…	…	…

1. 平文をランダムに変化させて暗号化を行い、消費電力を測定する
2. 中間値のhamming weightを計算する。それを各鍵仮説(AES鍵の先頭1バイトの場合、256通り)に対して行う。
3. 消費電力と、hamming weightとの間の相関係数を計算する
4. 誤った鍵仮説に対しては、相関係数の値がゼロに近い値になるが、正しい鍵仮説に対しては、相関係数の値が大きい値になると考えられるので、これによって正しい鍵の値が判明する

DPAの例3

- MIA (Mutual Information Analysis)
  - ある中間値をターゲットにする (例: $W=S(P \oplus k)$)
  - O: 消費電力を測定し、その分布を求める。
  - L_k: leakageに現れるpower model。ただし、power modelが不明なら、L_k=Id (恒等関数) としてもよい。
  - $H=L_k(W)$ を計算し、その分布を求める。
  - 各鍵仮説に対して、Mutual Information I(O;H)を計算する。
  - 最も高いMutual Informationを与える鍵仮説を鍵と推定する。
  - 理論的には、正しい鍵仮説に対しては正のMutual Informationが得られ、誤った鍵仮説に対しては、OとHは独立となり、I(O;H)=0となるはずである。

サイドチャネル攻撃対策 — Hiding —

- サイドチャネル攻撃は、暗号演算過程の中間値の情報が漏れることを利用する
  - → 消費電力と中間値との相関をなくそうとすることで対策する
- 例
  - Random Delay: 演算の間にランダムに遅延を挿入する
    - 電力波形の位置合わせを困難にする
  - ノイズ付加: 消費電力にノイズを付加して、データに依存する消費電力波形を見づらくする
  - Dual Rail Pre-Charge Logic
    - 普通のICでは1ビットを1本の信号線で表現する
    - Dual Rail では、1ビットを2本の信号線で表現する
      - 論理的なビットの値が0でも1でも消費電力が(理論的には)変わらない。

値	内部表現
0	01
1	10

サイドチャネル攻撃対策 — Masking —

- 中間値に、ランダムな値を「マスク」して、生の中間値の情報が漏れることを防ぐ。
- blinding とも言う。
- 秘密情報を2個(以上)の値に分散して持たせているとの観点から、Secret Sharingと言うこともある。

サイドチャネル攻撃対策 — Masking —

- マスキングの種類
  - Boolean Masking
    - 論理演算 (排他的論理和) によるmasking
  - Arithmetic Masking
    - 算術演算 (加法や乗法) によるmasking

サイドチャネル攻撃対策 — Masking —

IPA

- Boolean Maskingの例

AESの第1ラウンド

$a \leftarrow p_i \oplus k_i$　(p_i: 平文の第iバイト, k_i : 鍵の第iバイト)
$b \leftarrow Sbox(a)$
↑
この中間値が攻撃される

マスキング

m_1, m_2: ランダムなマスク
$a' \leftarrow (p_i \oplus m_1) \oplus k_i$
$b' \leftarrow Sbox'(a)$　(*Sbox'*: マスクを計算に入れたSbox (=$Sbox(x \oplus m_1) \oplus m_2$))
↑
この値は、鍵の値に依存しない (ランダムなマスクのため) ので、
この値に対して攻撃されても鍵は復元できない

最終ラウンド終了後にマスクを外して暗号文を出力する

サイドチャネル攻撃対策 — Masking —

- Arithmetic Maskingの例

RSA暗号での復号

$c^d \bmod n$ (*c*: 暗号文, *d*: 秘密鍵, *n*: 法)

↑

指数*d*が攻撃される可能性

exponent blinding

r: 乱数
$d' = d + r\varphi(n)$ (φ: Eulerのトーシェント関数)
$c^{d'} \bmod n$ を計算する

↑

生の指数*d*を使用しないので、*d*そのものが攻撃対象になることはない

サイドチャネル攻撃
— Higher-Order Attack —

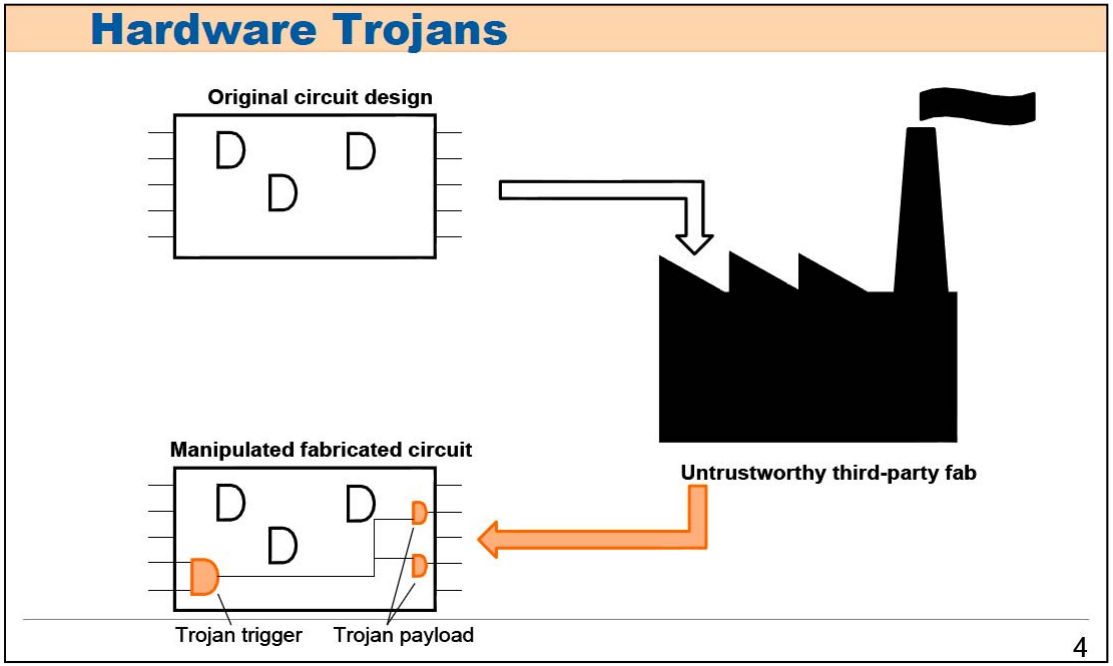
- Higher-Order Attack
  - Maskingを使った実装に対する攻撃
  - 2nd-order attackは、2個の中間値からのjoint leakageを悪用する攻撃
  - 例えば、maskingを使用したAES実装において、1個のマスク値を使い続けた場合、2nd-order attackが有効

故障利用攻撃 (Fault Injection Attack)

- 暗号機能を実装したハードウェアの動作中に故意に故障(fault)を起こし、計算誤りを利用して解析を行う
  - クロックグリッチ
  - 電源グリッチ
  - レーザー照射
  - 電磁場印加
- 故障利用攻撃の例: RSA-CRTに対するBellCoRe Attack
  - $s=m^d \bmod n$ とする(正しい署名)
  - $s'_p=m^{dp} \bmod p$ (s_pの計算にfaultを入れる)
  - $s_q=m^{dq} \bmod q$
  - $s'=s_q+q(i_q(s'_p-s_q) \bmod p)$
  - このとき、$\gcd(s-s',n)=q$。また、$\gcd(m-s'^e,n)=q$ → 秘密の素因数が判明

Hardware Trojan

- ICチップに対する、悪意ある回路の改竄
  - チップの製造をアウトソース → 工場が悪意を持ってチップの回路を改竄するかも知れない
  - セキュリティ機能の無効化等を目的とする



出典: Raghavan Kumar et al., Parametric Trojans for Fault-Injection Attacks on Cryptographic Hardware FDTC2011

CHES/FDTC

CHES	9/23-26	Workshop on Cryptographic Hardware and Embedded Systems	暗号等のハードウェア実装に関するセキュリティ
FDTC	9/23	Fault Diagnosis and Tolerance in Cryptography	故障利用攻撃

CHES2014の発表内容の概要紹介

EM Attack Is Non-Invasive? – Design Methodology and Validity of EM Attack Sensor

Naofumi Homma[1], Yu-ichi Hayashi[1], Noriyuki Miura[2], Daisuke Fujimoto[2], Daichi Tanaka[2], Makoto Nagata[2], and Takafumi Aoki[1]

[1]Graduate School of Information Sciences, Tohoku University, Japan
[2]Graduate School of System Informatics, Kobe University, Japan

https://eprint.iacr.org/2014/541

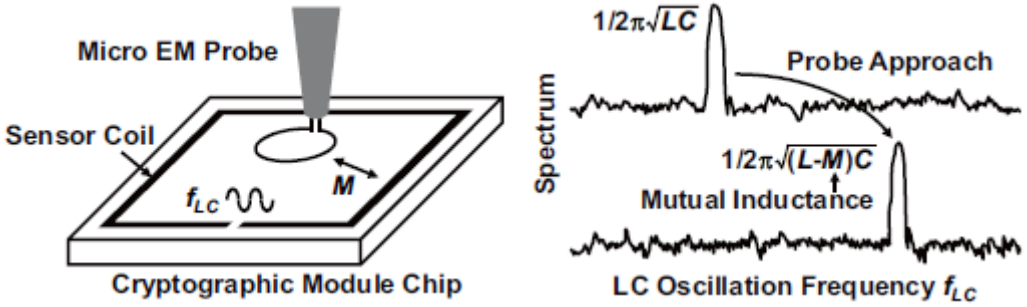
BEST PAPER AWARD

EM Attack Is Non-Invasive? – Design Methodology and Validity of EM Attack Sensor

- EM Attack (電磁放射解析)は、ICチップに接触することなく、サイドチャネル情報を取得することができる、強力な攻撃である。
- 電力解析と異なり、局所的なリークを検出できるので、対策が厄介である。
- チップへの接触はないので、電磁プローブを近づけられてもチップの状態は全く変化せず、検知できない？

EM Attack Is Non-Invasive? – Design Methodology and Validity of EM Attack Sensor

♦ EM Attack Sensor



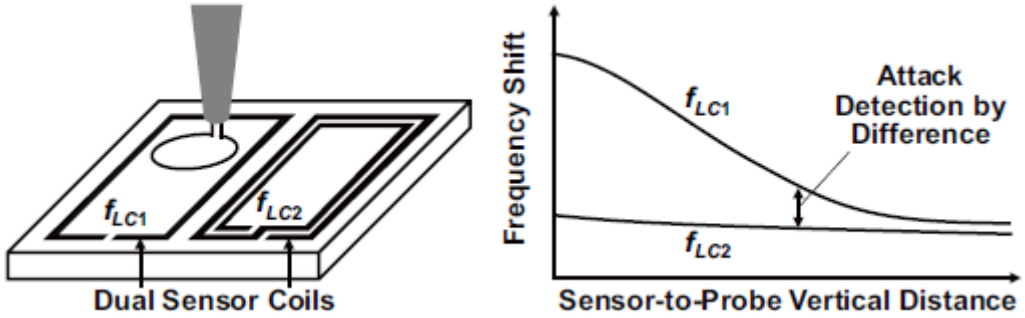
センサーコイル(LC回路)に電流が流れているときに、電磁プローブを近づけると、センサーコイルとプローブとの相互インダクタンスMのために、周波数が変化する。

$$f_{LC} \approx 1/(2\pi\sqrt{LC}) \Rightarrow f_{LC} \approx 1/(2\pi\sqrt{(L-M)C})$$

ただし、このセンサーでは、周波数リファレンスが必要であるが、外部クロックなどは攻撃者に操作されているかも知れないので、信頼できる周波数リファレンスとしては使えない。
さらに、オンチップの周波数リファレンスは、エリアや電力を浪費するアナログ回路が必要となる。

EM Attack Is Non-Invasive? – Design Methodology and Validity of EM Attack Sensor

♦ Dual-coil Sensor



形状や巻き数の異なる2個のコイルを使う。
周波数の差を測定することで、周波数リファレンスなしでプローブの接近を検知できる。

EM Attack Is Non-Invasive? – Design Methodology and Validity of EM Attack Sensor

- Demo

http://www.youtube.com/watch?v=9stWXvo-dp0

How to Estimate the Success Rate of Higher-Order Side-Channel Attacks

Victor Lomné[1], Emmanuel Prouff[1], Matthieu Rivain[2], Thomas Roche[1], and Adrian Thillard[1]

[1]ANSSI, France
[2]CryptExperts

https://eprint.iacr.org/2014/673

How to Estimate the Success Rate of Higher-Order Side-Channel Attacks

- サイドチャネル攻撃の成功率 (推測した鍵が正しい鍵である確率)を理論的に考察
  - 経験的には、実際に攻撃を何回か行い、成功した回数を記録することで成功率を見積もることができる。
  - しかし、効果的な対策が施されたチップに対しては、攻撃はコストが高過ぎ、攻撃を何度も(場合によっては1回でも)行うことは現実的には困難である。
  - このような状況は、実装がセキュアであることを意味しない。評価者の手に余っているだけである。
  - そこで、この論文で、マスキングで保護している実装を対象にしたhigher-order side-channel attackの成功率を見積もる手法を提案する。
- 1st-order attackの成功率に関する研究は存在する
- 1st-order attackの手法を拡張して、higher-order attackに適用できるようにして、higher-order attackの成功率の理論値を導出する
- 実際のプロセッサと実装に対して実験を行い、理論値と実験値がよく一致していることを確認

Good is Not Good Enough
Deriving Optimal Distinguishers from Communication Theory

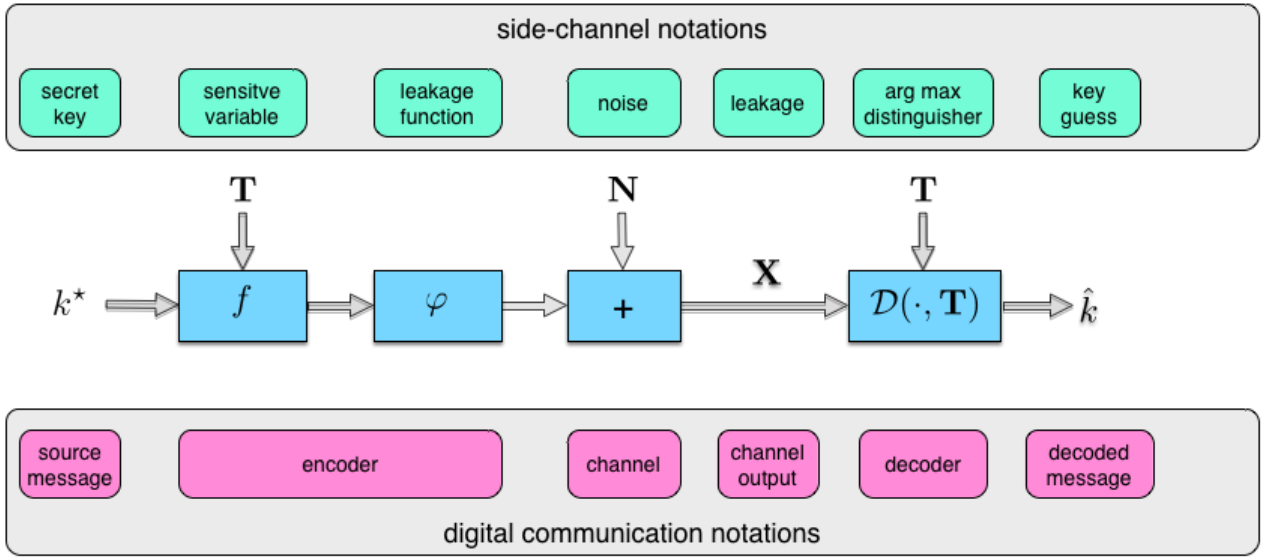
Annelie Heuser[1], Olivier Rioul[1], and Sylvain Guilley[1,2]

[1]Télécom ParisTech, Institut Mines-Télécom, CNRS LTCI
[2]Secure-IC S.A.S

https://eprint.iacr.org/2014/527

Good is Not Good Enough: Deriving Optimal Distinguishers from Communication Theory

- サイドチャネル攻撃にCommunication Theoryの考え方を応用し、最適 (成功率が最大) なdistinguisherを導く



Good is Not Good Enough: Deriving Optimal Distinguishers from Communication Theory

IPA

- Leakage Modelが既知で、ノイズが正規分布の場合の例
  - リーク

    $X = \mathrm{HW}[\mathrm{Sbox}[T \oplus k^\star]] + N$ （k^*=正しい鍵、N: ノイズ）
  - リーク予想

    $Y(k) = \mathrm{HW}[\mathrm{Sbox}[T \oplus k]]$ for all $k \in \mathcal{K}$

Theorem 4 (Optimal expression for Gaussian noise). *When the noise is zero mean Gaussian, $N \sim \mathcal{N}(0, \sigma^2)$, the optimal distinguishing rule is*

$$\mathcal{D}^{M,G}_{opt}(\mathbf{x}, \mathbf{t}) = \arg\max_k \langle \mathbf{x} | \mathbf{y}(k) \rangle - \frac{1}{2}\|\mathbf{y}(k)\|_2^2. \tag{9}$$

Theorem 5 (Correlation power analysis). *When the leakage arises from $X = aY(k^\star) + b + N$ where N is zero-mean Gaussian, $\hat{k} = \arg\min_k \min_{a,b} \|\mathbf{x} - a\mathbf{y}(k) - b\|^2$, is equivalent to maximizing the absolute value of the empirical Pearson's coefficient:*

$$\hat{k} = \arg\max_k |\hat{\rho}(k)| = |\widehat{\mathrm{Cov}}(\mathbf{x}, \mathbf{y}(k))| \Big/ \sqrt{\widehat{\mathrm{Var}}(\mathbf{x})\widehat{\mathrm{Var}}(\mathbf{y}(k))} \tag{12}$$

where the empirical (co)variances are defined by $\widehat{\mathrm{Cov}}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{m}(x_i - \bar{x})(y_i - \bar{y})$ and $\widehat{\mathrm{Var}}(\mathbf{x}) = \widehat{\mathrm{Cov}}(\mathbf{x}, \mathbf{x})$.

Good is Not Good Enough: Deriving Optimal Distinguishers from Communication Theory

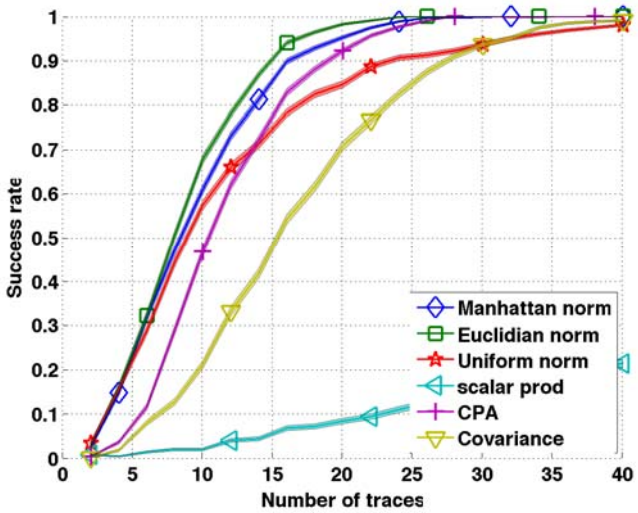
- Leakage Modelが既知で、ノイズが非正規分布の場合の例

Theorem 7 (Optimal expression for uniform and Laplacian noises). *When f and φ are known such that $Y(k) = \varphi(f(k,T))$, and the leakage arises from $X = Y(k^\star) + N$ with $N \sim \mathcal{U}(0,\sigma^2)$ or $N \sim \mathcal{L}(0,\sigma^2)$, then the optimal distinguishing rule becomes*

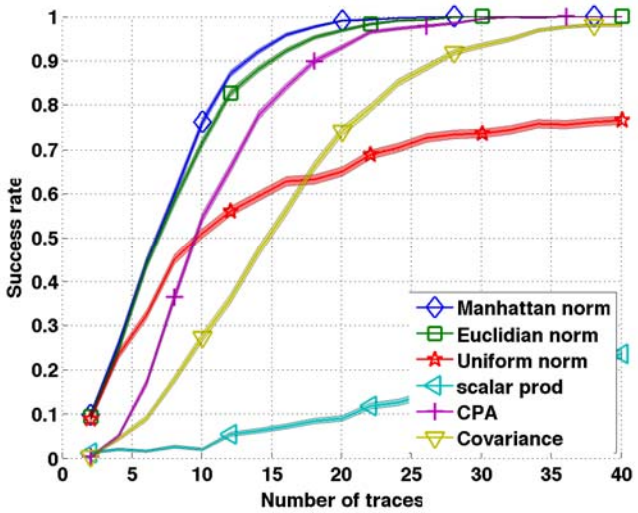
- *Uniform noise distribution:* $\mathcal{D}_{opt}^{M,U}(\mathbf{x},\mathbf{t}) = \arg\max_k \ -\|\mathbf{x} - \mathbf{y}(k)\|_\infty$,
- *Laplace noise distribution:* $\mathcal{D}_{opt}^{M,L}(\mathbf{x},\mathbf{t}) = \arg\max_k \ -\|\mathbf{x} - \mathbf{y}(k)\|_1$.

Good is Not Good Enough: Deriving Optimal Distinguishers from Communication Theory

Success rate, with a known model



(a) Gaussian Noise, σ=1



(b) Laplacian Noise, σ=1

- CPAは必ずしも最適なdistinguisherではない。最適なdistinguisherはCPAとは異なるものであるケースが多い。

図は https://eprint.iacr.org/2014/527.pdf から一部抜粋

“Ooh Aah... Just a Little Bit”: A small amount of side channel can go a long way

Naomi Benger[1], Joop van de Pol[2], Nigel P. Smart[2], and Yuval Yarom[1]

[1]School of Computer Science, The University of Adelaide, Australia
[2]Dept. Computer Science, University of Bristol, United Kingdom

https://eprint.iacr.org/2014/161

“Ooh Aah... Just a Little Bit”: A small amount of side channel can go a long way

IPA

- FLUSH+RELOAD attack
  - キャッシュのヒット/ミス時の時間の差を利用したサイドチャネル攻撃
    - 同じCPU上でスパイプロセスを走らせ、そこから攻撃する
    - あるメモリ領域をキャッシュから強制的に追い出す
    - 後でその領域を読み込む
    - 標的のプロセスがその領域にアクセスしていたら、キャッシュに読み込まれるので、読み込み時間が早い。これによって、標的のプロセスがメモリのある領域にアクセスしたかどうかを突き止められる
  - プロセッサのLast Level Cache (Intel x86の場合はL3 Cache)へのサイドチャネル攻撃が可能なので、CPUの違うコアにあるプロセスを攻撃できる

“Ooh Aah... Just a Little Bit”: A small amount of side channel can go a long way

- FLUSH+RELOAD attackを、OpenSSLのECDSAの実装に適用する
- ECDSAで使用するephemeral private keyのかなりのビットを復元することに成功

Destroying Fault Invariant with Randomization

A Countermeasure for AES against Differential Fault Attacks

Harshal Tupsamudre, Shikha Bisht, and Debdeep Mukhopadhyay

Department of Computer Science and Engg. IIT Kharagpur, India

Destroying Fault Invariant with Randomization – A Countermeasure for AES against Differential Fault Attacks

IPA

- Fault Injection対策
  - Detection Countermeasure
    - Faultの注入を検出するタイプ
    - 例えば、計算を2重化して結果を比較する実装
    - 比較演算をつぶされると対策が破られる
  - Infection Countermeasure
    - Faultの効果を拡散させ、faulty outputをDFAに使えないようにする
    - LatinCrypt 2012[1]で方式を提案
    - FDTC2013[2]で、この実装に対する攻撃が発表される
    - これらの対策と攻撃のさらなる分析と、新たな対策の提案

[1]Gierlichs et al., Infective Computation and Dummy Rounds: Fault Protection for Block Ciphers without Check-before-Output, LatinCrypt 2012
[2]Battistello et al., Fault Analysis of Infective AES Computations, FDTC2013

Destroying Fault Invariant with Randomization – A Countermeasure for AES against Differential Fault Attacks

IPA

LatinCrypt2012 Countermeasure

Algorithm 1. Infection Countermeasure [10]

Inputs : P, k^j for $j \in \{1, \ldots, n\}$, (β, k^0), $(n = 11)$ for AES128
Output : C = BlockCipher(P, K)

1. State $R_0 \leftarrow P$, Redundant state $R_1 \leftarrow P$, Dummy state $R_2 \leftarrow \beta$
2. $C_0 \leftarrow 0$, $C_1 \leftarrow 0$, $C_2 \leftarrow \beta$, $i \leftarrow 1$
3. while $i \leq 2n$ do
4. $\quad \lambda \leftarrow RandomBit()$ // $\lambda = 0$ implies a dummy round
5. $\quad \kappa \leftarrow (i \wedge \lambda) \oplus 2(\neg\lambda)$
6. $\quad \zeta \leftarrow \lambda \cdot \lceil i/2 \rceil$ // ζ is actual round counter, 0 for dummy
7. $\quad R_\kappa \leftarrow RoundFunction(R_\kappa, k^\zeta)$
8. $\quad C_\kappa \leftarrow R_\kappa \oplus C_2 \oplus \beta$ // infect C_κ to propagate a fault
9. $\quad \epsilon \leftarrow \lambda(\neg(i \wedge 1)) \cdot SNLF(C_0 \oplus C_1)$ // check if i is even
10. $\quad R_2 \leftarrow R_2 \oplus \epsilon$
11. $\quad R_0 \leftarrow R_0 \oplus \epsilon$
12. $\quad i \leftarrow i + \lambda$
13. end
14. $R_0 \leftarrow R_0 \oplus RoundFunction(R_2, k^0) \oplus \beta$
15. return(R_0)

最終ラウンドや最後から2番目のラウンドへのfault injectionに弱い
誤ったバイトと正しいバイトのマスクに同じ未知の値を使用するため

改良版Countermeasure

Algorithm 2. Improved Countermeasure

Inputs : P, k^j for $j \in \{1, \ldots, n\}$, (β, k^0), $(n = 11)$ for AES128
Output : C = BlockCipher(P, K)

1. State $R_0 \leftarrow P$, Redundant state $R_1 \leftarrow P$, Dummy state $R_2 \leftarrow \beta$
2. $i \leftarrow 1$, $q \leftarrow 1$
3. $rstr \leftarrow \{0,1\}^t$ // $\#1(rstr) = 2n, \#0(rstr) = t - 2n$
4. while $q \leq t$ do
5. $\quad \lambda \leftarrow rstr[q]$ // $\lambda = 0$ implies a dummy round
6. $\quad \kappa \leftarrow (i \wedge \lambda) \oplus 2(\neg\lambda)$
7. $\quad \zeta \leftarrow \lambda \cdot \lceil i/2 \rceil$ // ζ is actual round counter, 0 for dummy
8. $\quad R_\kappa \leftarrow RoundFunction(R_\kappa, k^\zeta)$
9. $\quad \gamma \leftarrow \lambda(\neg(i \wedge 1)) \cdot BLFN(R_0 \oplus R_1)$ // check if i is even
10. $\quad \delta \leftarrow (\neg\lambda) \cdot BLFN(R_2 \oplus \beta)$
11. $\quad R_0 \leftarrow (\neg(\gamma \vee \delta) \cdot R_0) \oplus ((\gamma \vee \delta) \cdot R_2)$
12. $\quad i \leftarrow i + \lambda$
13. $\quad q \leftarrow q + 1$
14. end
15. return(R_0)

独立したランダムな値がすべての誤ったバイトと正しいバイトに影響するため、弱点をなくしている。

Reversing Stealthy Dopant-Level Circuits

Takeshi Sugawara[1], Daisuke Suzuki[1], Ryoichi Fujii[1], Shigeaki Tawa[1]
Ryohei Hori[2], Mitsuru Shiozaki[2], and Takeshi Fujino[2]

[1]Mitsubishi Electric Corporation, Japan
[2]Ritsumeikan University, Japan

https://eprint.iacr.org/2014/508

Reversing Stealthy Dopant-Level Circuits

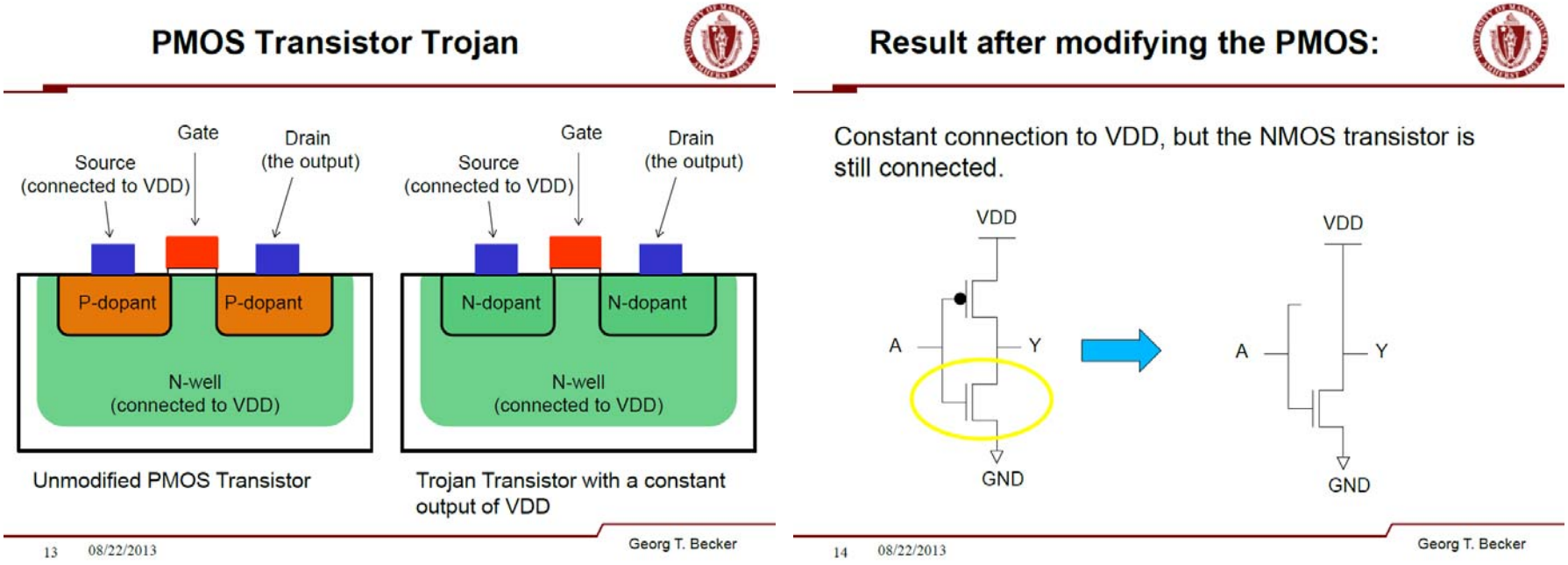
IPA

- Hardware Trojan
  - ハードウェアレベルで埋め込まれるトロイの木馬。攻撃者が不正な動作をさせるためにひそかに不正な論理ゲートをICに埋め込む
  - 埋め込み方はいろいろ考えられるが、シリコンに打ち込むドーパント(n型/p型)を変えることで論理を変更する方法がある

Reversing Stealthy Dopant-Level Circuits

- CHES2013 Stealthy Dopant-Level Hardware Trojan
  - Georg T. Becker, et. al

半導体のドーパントを改変することで、ゲートの動作を変える
ゲートそのものの追加はないので、視覚的に発見することは難しい



http://www.iacr.org/workshops/ches/ches2013/presentations/CHES2013_Session4_3.pdf

Reversing Stealthy Dopant-Level Circuits

- ドーパントだけが異なるようなICチップ内の回路を、見分けられるのか?
  - SEM (Scanning Electron Microscopy)やFIB (Focused Ion Beam)を使えば判別可能である
    - PVC (Passive Voltage Contrast) という手法
    - SEMやFIBから電子やビームを打ち込んだ時に、n型シリコンとp型シリコンでは、表面電位が異なるために二次電子の出方に差が現れ、像にコントラストの差が現れる

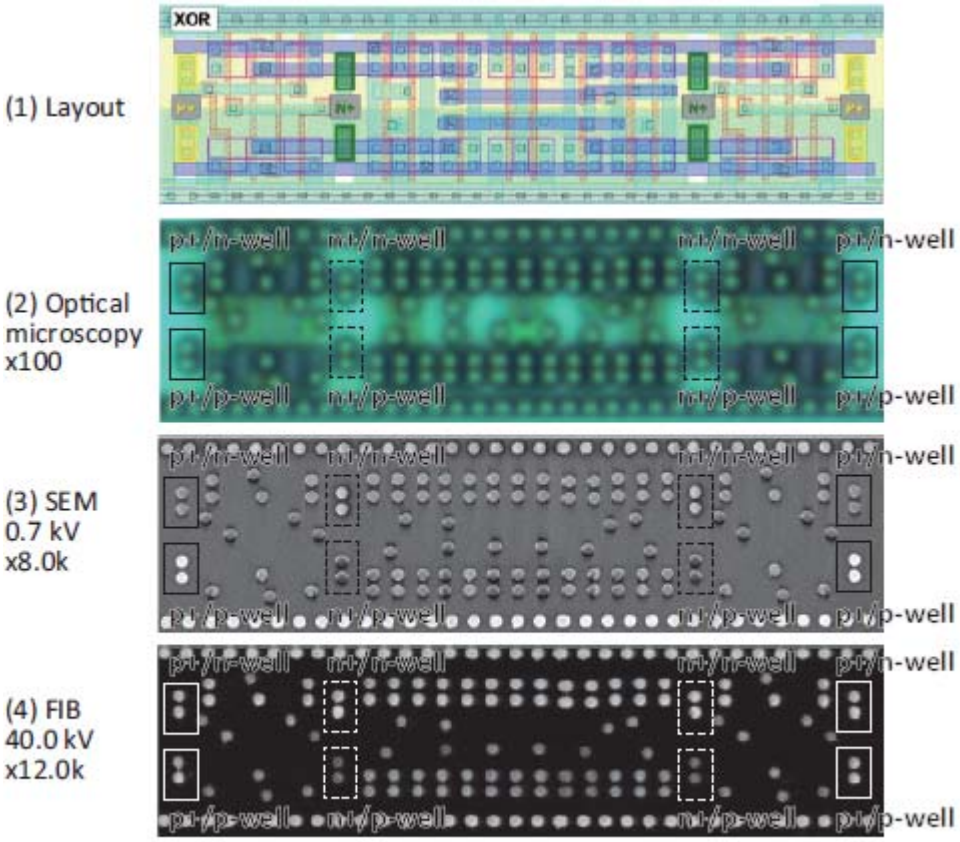


Fig. 8. Image of DPD-LE configured as XOR

出典: http://eprint.iacr.org/2014/508.pdf

IPA

Secure Conversion between Boolean and Arithmetic Masking of Any Order

Jean-Sébastien Coron, Johann Großschädl, and Praveen Kumar Vadnala

Laboratory of Algorithmics, Cryptology and Security (LACS),
University of Luxembourg, Luxembourg

Secure Conversion between Boolean and Arithmetic Masking of Any Order

- Boolean MaskingとArithmetic Maskingの変換をセキュアに行いたい
- 以前の結果: Goubin's Algorithms (Goubin, L: CHES1999)
  - Boolean MaskingとArithmetic Masking間のセキュアな変換
  - ただし、1st Order Attackに対してのみセキュア
- この論文は、任意のnに対して、n-orderの攻撃に対してセキュアな変換アルゴリズムを提案
- 実際に、この変換アルゴリズムを使用してHMAC-SHA-1を実装
  - SHA-1は、論理演算とモジュロ加算の両方の種類の演算を使用するので、Boolean MaskingとArithmetic Maskingの変換を用いる
  - パフォーマンスのペナルティはかなり大きいが、challenge-response認証のように入力データが1ブロックの場合には有用と考えられる

Making RSA–PSS Provably Secure against Non-random Faults

Gilles Barthe[1], François Dupressoir[1], Pierre-Alain Fouque[2], Benjamin Grégoire[4], Mehdi Tibouchi[3], and Jean-Christophe Zapalowicz[4]

[1]IMDEA Software Institute, Madrid, Spain,
[2]Université de Rennes 1 and Institut universitaire de France, France,
[3]NTT Secure Platform Laboratories, Japan,
[4]INRIA, France

Making RSA–PSS Provably Secure against Non-random Faults

- RSA-PSSは、random faultに対してはsecureである (Coron and Mandal, AsiaCrypt2009)
- RSA-PSSは、non-random faultに対してはsecureではない (Fonque et al., CHES2012)

- RSA-PSSの実装で、non-random faultに対してsecureになるような対策を提案
  - Infective Countermeasure
  - Secureであることは、EasyCrypt(*)を用いて形式的に証明

* http://www.easycrypt.info

Side-Channel Attack against RSA Key Generation Algorithms

Aurélie Bauer, Eliane Jaulmes, Victor Lomné, Emmanuel Prouff, and Thomas Roche

ANSSI

http://www.ssi.gouv.fr/IMG/pdf/CHES2014_Side_Channel_Attack_against_RSA_Key_Generation_Algorithms.pdf

Side-Channel Attack against RSA Key Generation Algorithms

- 確率的素数判定を用いた素数生成アルゴリズムの実装に対するサイドチャネル攻撃
- 素数生成アルゴリズム(の1例)
  1. 素数候補vを乱数生成器で生成
  2. 素数判定
     - 小さい素数(例えば256以下)で割り切れるかどうかを直接確かめる
     - Miller-Rabin's tests, Lucas test
  3. 素数であれば、vを出力。素数でなければ、vにτ(偶数の定数)を加え、1に戻る
- 小さい素数での剰余演算が多数回行われるので、それをDPAのように攻撃する
- 小さい素数での剰余がDPAで判明すれば、CRTで合成する

Get Your Hands Off My Laptop: Physical Side-Channel Key-Extraction Attacks on PCs

Daniel Genkin[1,2], Itamar Pipman[2], and Eran Tromer[2]

[1]Technion, Israel
[2]Tel Aviv University, Israel

https://eprint.iacr.org/2014/626

Get Your Hands Off My Laptop: Physical Side-Channel Key-Extraction Attacks on PCs

IPA

- ラップトップPCそのものに対するサイドチャネル攻撃
  - PCのシャーシの電位
  - PCに接続したイーサーネット、USB、VGA、HDMI等のケーブルの端の電位
  - ノートPCに素手で触り、身体の電位を反対の手のリストバンド等から測定する
  - 電磁放射解析 (EMA)
  - 電力解析 (PC全体の消費電力から)
- 上記の攻撃で、GnuPGのRSA4096ビット、ElGamal3072ビット鍵を復元することに成功

RSA Meets DPA: Recovering RSA Secret Keys from Noisy Analog Data

Noboru Kunihiro and Junya Honda

The University of Tokyo, Japan

https://eprint.iacr.org/2014/513

RSA meets DPA: Recovering RSA Secret Keys from Noisy Analog Data

IPA

- Cold Boot Attack[1]
  - DRAMは、電源を切った後でも直ちにはメモリ内容が消えない
  - Cold Bootによって、メモリに格納されていた鍵 (例えば、ディスク暗号化用鍵) が、いくらかのノイズ入りの状態(何%かのビットが変化している)で復元できる。
  - ノイズ入りの鍵から完全な鍵の復元を行う
    - 27%以上のビットが正しければ、鍵の復元が成功する
- Cold Boot Attackや、Side Channel Attackによって得られた、ノイズ入りのRSA秘密鍵から、完全な鍵の復元を行うアルゴリズムの提案
  - ノイズがさらに大きい場合にも適用できる
  - ノイズの分布が不明な場合、DPAに似たアルゴリズムを使用する

[1]J. Alex Halderman, Seth D. Schoen, Nadia Heninger, William Clarkson, William Paul, Joseph A. Calandrino, Ariel J. Feldman, Jacob Appelbaum, and Edward W. Felten, Lest We Reminder: Cold Boot Attacks on Encryption Keys
http://citp.princeton.edu/research/memory/

Simple Power Analysis on AES Key Expansion Revisited

Christophe Clavier, Damien Marion, and Antoine Wurcker

Université de Limoges, XLIM-CNRS

Simple Power Analysis on AES Key Expansion Revisited

- AESの各round keyのHWをサイドチャネル情報から取得することによるSPA
- 以前の研究では、対策なしの実装に関する攻撃はあった
- この発表では、いくつかの対策されたチップに対しても攻撃が成功することを示す
  - ラウンドごとのマスキング (11-byte Entropy Boolean Masking)
  - バイトごとのマスキング (16-byte Entropy Boolean Masking)
  - Column内の計算順序のランダム化
- 各round keyの各バイトのHWが全く同じになるような異なるAES鍵が存在することも示す
  - 例: K = B3 65 58 9D B4 EB 57 72 1F 51 F7 58 02 0C 00 17
  - K‘ = F2 65 19 DC B4 EB 57 33 5E 51 F7 19 02 0C 00 56

Simple Power Analysis on AES Key Expansion Revisited

- AESの各round keyのHWをサイドチャネル情報から取得する
- 以
- こ
  - ラウンドごとのマスキング (11-byte Entropy Boolean Masking)
  - バイトごとのマスキング (16-byte Entropy Boolean Masking)
- 行

K_0			
m_0	m_0	m_0	m_0
m_0	m_0	m_0	m_0
m_0	m_0	m_0	m_0
m_0	m_0	m_0	m_0

K_1			
m_1	m_1	m_1	m_1
m_1	m_1	m_1	m_1
m_1	m_1	m_1	m_1
m_1	m_1	m_1	m_1

K_2			
m_2	m_2	m_2	m_2
m_2	m_2	m_2	m_2
m_2	m_2	m_2	m_2
m_2	m_2	m_2	m_2

K_0			
m_{00}	m_{04}	m_{08}	m_{12}
m_{01}	m_{05}	m_{09}	m_{13}
m_{02}	m_{06}	m_{10}	m_{14}
m_{03}	m_{07}	m_{11}	m_{15}

K_1			
m_{00}	m_{04}	m_{08}	m_{12}
m_{01}	m_{05}	m_{09}	m_{13}
m_{02}	m_{06}	m_{10}	m_{14}
m_{03}	m_{07}	m_{11}	m_{15}

K_2			
m_{00}	m_{04}	m_{08}	m_{12}
m_{01}	m_{05}	m_{09}	m_{13}
m_{02}	m_{06}	m_{10}	m_{14}
m_{03}	m_{07}	m_{11}	m_{15}

Side-Channel Leakage through Static Power Should We Care about in Practice?

Amir Moradi

Horst Gortz Institute for IT Security, Ruhr University Bochum, Germany

https://eprint.iacr.org/2014/025

Side-Channel Leakage through Static Power — Should We Care about in Practice?

- 電力解析は、ゲートのスイッチング時に流れる”dynamic power consumption”を対象にしてきた
  - 消費電力に占める割合は、“static power consumption”は少なく、”dynamic power consumption”が支配的である
- しかし、プロセスの微細化により、static power consumptionがだんだん無視できなくなってきた
- FPGAベースの暗号デバイスのstatic power consumptionを対象にした電力解析の、初めての実際的な結果を提示

Side-Channel Leakage through Static Power — Should We Care about in Practice?

- Static Power Analysisの特性
  - FPGAでの主要な電力消費は接続(シグナルルーティング)の部分である。ASICではそうではなく、配線はリーク電流にあまり影響を与えないので、レジスタの値やゲート出力の値が主要なリーク電流源になると思われる。
  - DCシグナルを増幅し、ローパスフィルターでノイズを除去するような専用の測定環境を用意する必要がある
  - 恒温槽のような手段で温度を一定に保つべきである。
  - static powerの測定はdynamic powerの測定より時間がかかる
  - S/N比はdynamic powerを測定する場合と比較して低いので、より多数の測定が必要になる。
  - デバイスのアーキテクチャを知ることが重要である。特に、どのタイミングでI/Oをオフにすべきかを知ることは非常に重要である。
- 現状の研究では、static powerに対する電力解析は、実用的とはいえるが、dynamic powerに対する攻撃よりも効率的ではない。
- しかし、leakageは常に1変数であり、マスキング実装において、秘密情報の異なるシェアは足し合わされてstatic powerを通して観察できる。Univariate-resistant approachのような設計はstatic powerを用いる攻撃に対して脆弱であるかも知れない。

FDTC2014の発表内容の概要紹介

Parametric Trojans for Fault-Injection Attacks on Cryptographic Hardware

Raghavan Kumar[1], Philipp Jovanovic[2], Wayne Burleson[1] and Ilia Polian[2]

[1]University of Massachusetts Amherst
[2]University of Passau

https://eprint.iacr.org/2014/783

Parametric Trojans for Fault-Injection Attacks on Cryptographic Hardware

- Fault Injection Attackを手助けするHardware Trojan
- ドーパントの入れ方を改変することで、fault injection耐性を低下させる
  - ドーパント濃度の改変
  - ドーパント注入領域の改変
- この論文の例では、わずかな電圧低下(~10%)で、trojanをアクティブにできる
  - ゲートがfaultを起こしやすくなる

Differential Fault Intensity Analysis

Nahid Farhady Ghalaty, Bilgiday Yuce, Mostafa Taha, Patrick Schaumont

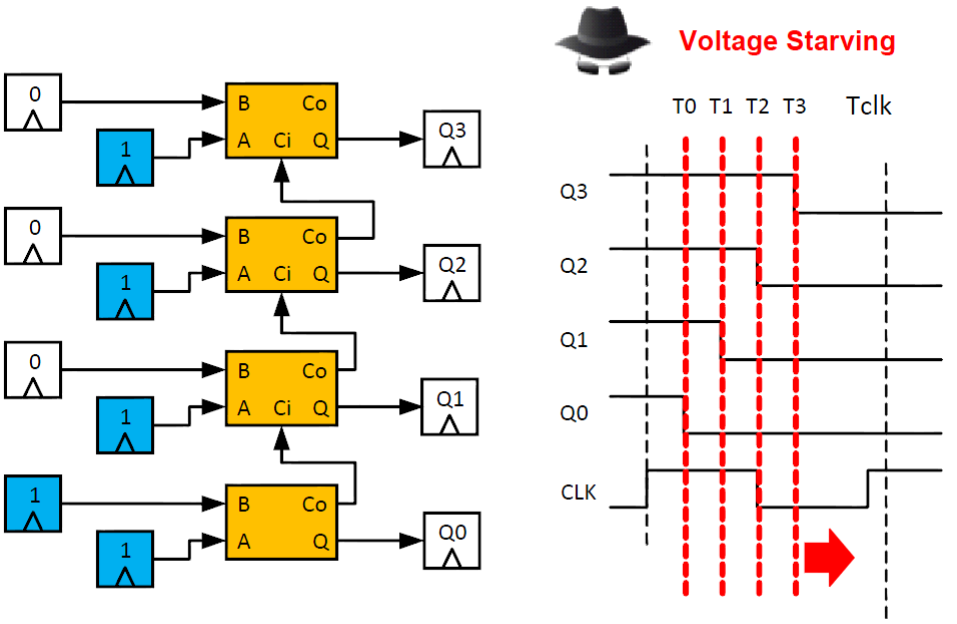
Bradley Department of Electrical and Computer Engineering
Virginia Tech

http://rijndael.ece.vt.edu/schaum/papers/2014fdtc.pdf

Differential Fault Intensity Analysis

- あるバイトの処理中にFault (ここではグリッチ)を受けたときの感受性は、ビットによって偏りがあることが考えられる。
  - 伝播時間は一様でないから、faultに対する反応も一様ではない
- そのため、バイトの中で、faultへの感受性に偏りがあるかもしれない。
- このような、faultに対する感受性の偏りを利用するのがDFIAである。
- DPAと同様に、各鍵仮説に対して、攻撃目標の中間値を計算し、fault modelに最も近いものを正しい鍵と推定する。

Where do Biased Faults come from?



$P_{fault}(Q3) > P_{fault}(Q2) > P_{fault}(Q1) > P_{fault}(Q0)$

Fault Sensitivity Analysis Meets Zero-Value Attack

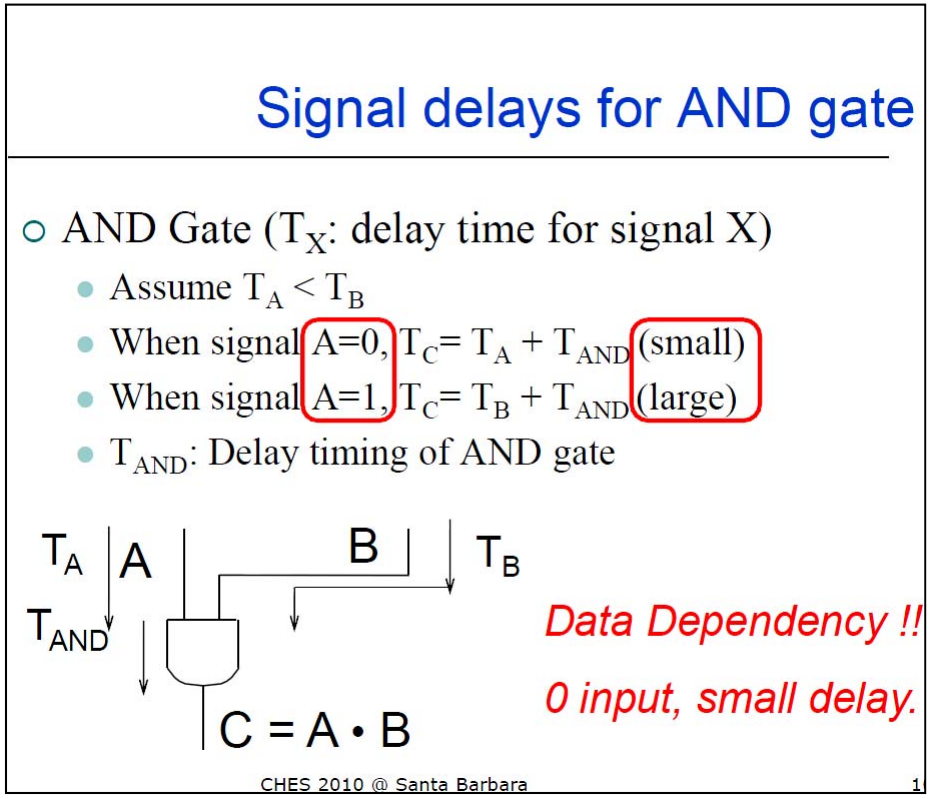
Oliver Mischke, Amir Moradi, Tim Güneysu

Horst Görtz Institute for IT-Security
Ruhr-Universität Bochum

https://www.emsec.rub.de/media/ei/veroeffentlichungen/2014/08/22/submitted_final_FDTC_before_check.pdf

Fault Sensitivity Analysis Meets Zero-Value Attack

- Fault Sensitivity Analysis (FSA) とは、faultに対する感受性がデータ依存することがあることを利用した攻撃
- Y. Li et al. がCHES2010で発表



出典: Y. Li, et al. Fault Sensitivity Analysis. In CHES 2010

Fault Sensitivity Analysis Meets Zero-Value Attack

- Zero-Value Attack は、サイドチャネル攻撃において、中間値が0になる状態を狙った攻撃
- 中間値が0かそれ以外かで電力消費に差が出る
- FSAにおいても、0という値は特徴的なのでZero-Value modelは応用可能
- 乗算は0という値を変えないので、乗算マスキングの弱点となる
- Fault Sensitivity Analysis (FSA) と、Zero-Value Attackの手法を組み合わせることで、FSAそのものが効果がないようなハードウェア実装(例: Concurrent Error Detection (CED scheme))に対しても攻撃が有効になることがある。
  - CEDとは、冗長性を持たせてエラーを検出する仕組み
- Invariance-based CED[1]に対しても、攻撃が成功することを示す

[1] X. Guo and R. Karri. Invariance-Based Concurrent Error Detection for Advanced Encryption Standard. In DAC 2012, pages 573–578. ACM, 2012.
論文は https://eprint.iacr.org/2013/603.pdf からも取得可

Countermeasures Against High-Order Fault-Injection Attacks on CRT-RSA

Pablo Rauzy and Sylvain Guilley

SEN group ; COMELEC Dept
Institut Mines-Télécom ; Télécom ParisTech ; CNRS LTCI

https://eprint.iacr.org/2014/559

Countermeasures Against High-Order Fault-Injection Attacks on CRT-RSA

- High-Order Fault Injection
  - 与えるfaultの数を攻撃のorderという
  - orderが2以上の攻撃を、high-order attackという
- CRT-RSAに対するfault injection attack対策
  - ほとんどがfirst-order attackへの対策
  - high-order attackに有効な対策はあまり研究されていない
- High-order Attackへの対策を提案
  - Fault attackを形式化して考察
  - 任意のorderの攻撃に対抗できる対策を構築できる

IPAの取り組み

IPAの取り組み

- ハードウェア脆弱性評価に関する人材育成
  - 新しい攻撃への耐性を評価する最先端のツールを整備して、日本の半導体ベンダ、ICカードベンダ、評価機関、大学などの研究機関が利用できる評価環境の整備を進めている。
    - 最先端の評価ツール及びテストビークル(評価対象のIC)を使用し、脆弱性を評価することで新しい攻撃手法を修得
    - ICカードの開発過程で利用し、対抗策を検証することで、高い攻撃耐性を持った製品開発が可能
    - 将来的な攻撃手法の研究活動に活用
    - 興味深い攻撃については、IPA所有の装置での再現実験の実施を検討
  - 技術セミナーの開催
    - 次回は2015年1月に、CARTES、CARDISの内容についてのセミナーを開催予定
    - 2015年度から、ハードウェアセキュリティに関する初級者向けの技術セミナーの開催を検討中

IPAの取り組み

- 興味深い攻撃に関する論文
  - How to Estimate the Success Rate of Higher-Order Side-Channel Attacks
  - Good is Not Good Enough: Deriving Optimal Distinguishers from Communication Theory
  - Side-Channel Attack against RSA Key Generation Algorithms
  - RSA Meets DPA: Recovering RSA Secret Keys from Noisy Analog Data
  - Simple Power Analysis on AES Key Expansion Revisited
  - Side-Channel Leakage through Static Power: Should We Care about in Practice?
  - Differential Fault Intensity Analysis
  - Fault Sensitivity Analysis Meets Zero-Value Attack

参考文献

- Cryptology ePrint Archive
  https://eprint.iacr.org
  CHESの論文の多くはePrint Archiveにも登録されている

- CHES 2014 Program
  http://www.chesworkshop.org/ches2014/program.php
  プレゼンテーションスライドがダウンロード可能

- FDTC 2014 Presentation Slides
  http://conferenze.dei.polimi.it/FDTC14/slides.html

ご清聴ありがとうございました。

当セミナーに関する質問は以下のメールアドレスまでどうぞ。

jcmvp-info@ipa.go.jp