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

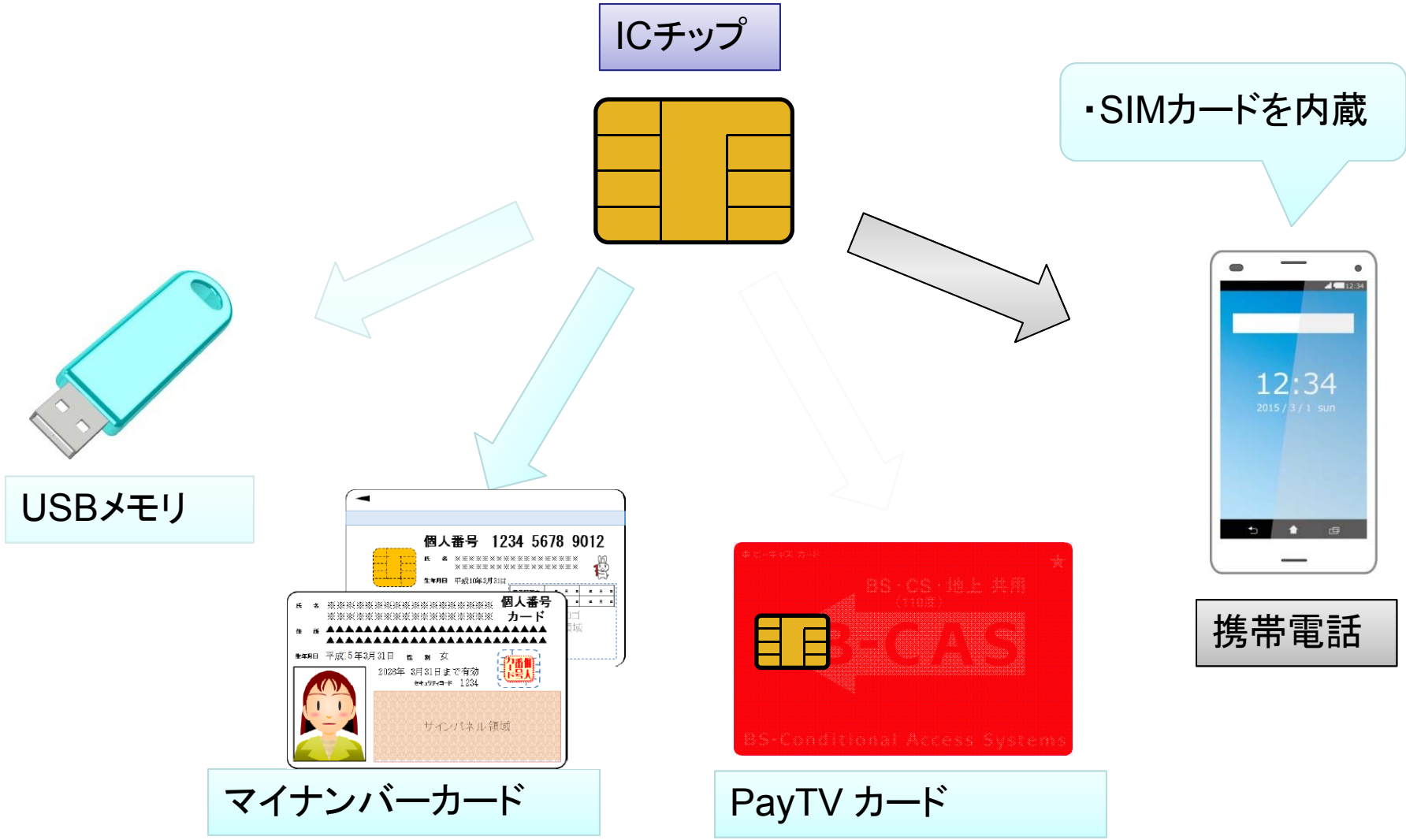
― CHES/FDTC参加報告 ―

2015年12月3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

ハードウェアセキュリティトピックの紹介

様々な情報機器に用いられる セキュリティICチップ



利用環境によって異なる脅威


サーバ


デスクトップPC


複合機

オフィス等物理アクセスが制限された環境に設置される。
部外者が直接アクセスする可能性は低い。（適切に管理されていれば）

ICカード


USBメモリ

携帯電話

どこにでも自由に持ち運べる。
様々な環境で利用されるデバイス。（紛失・盗難等のリスク、通信データの盗聴等のリスクは固定環境より増大）

想定される利用環境における脅威・攻撃の分析が重要。
脅威・攻撃に対抗できるセキュリティ対策が必要となる。

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
  - 消費電力 → 電力解析 (Power Analysis)
  - 電磁場 → 電磁解析 (Electromagnetic Analysis)
  - 処理時間 → タイミングアタック
  - その他
    - キャッシュヒット/ミス
    - 分岐予測

AESアルゴリズム

IPA

- 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

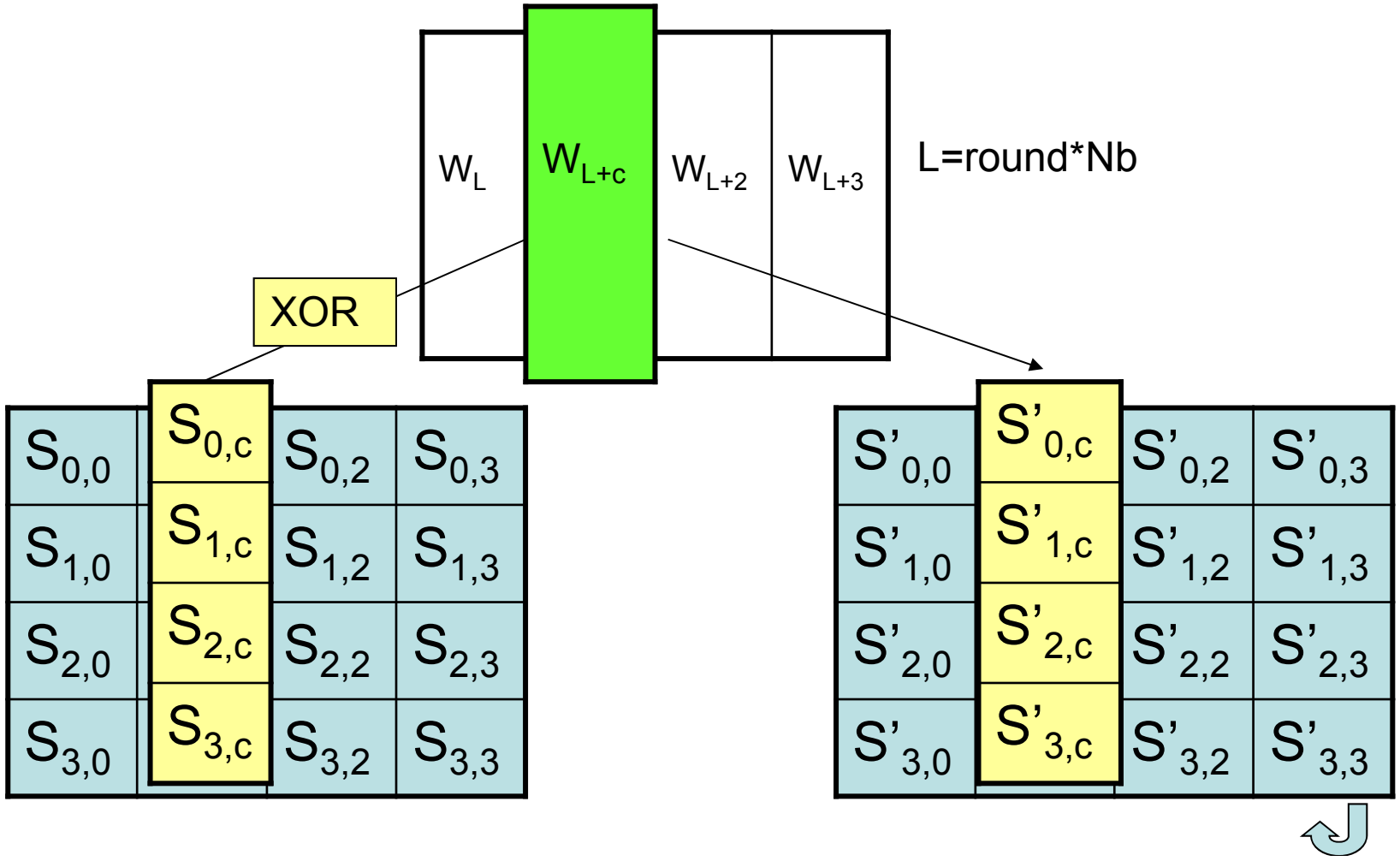
ShiftRows:
行単位の左シフト処理

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

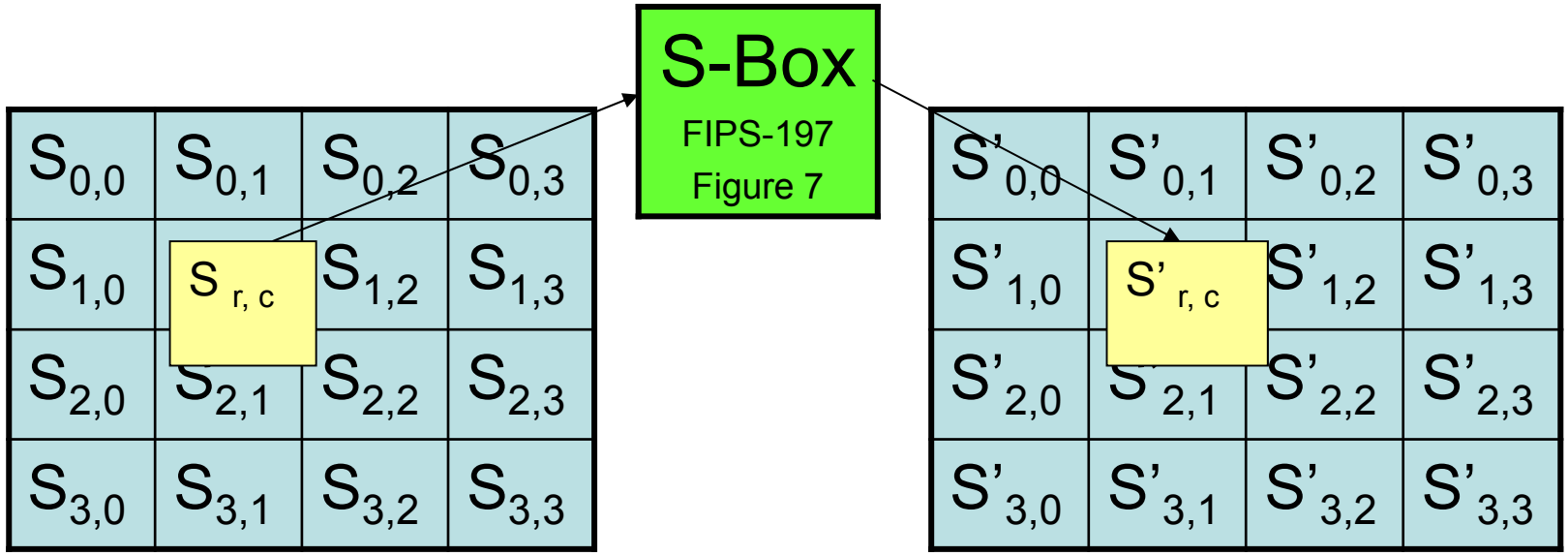
- 4ブロックを1列として，列ごとに拡大鍵とXOR処理。



AES：SubBytesの処理

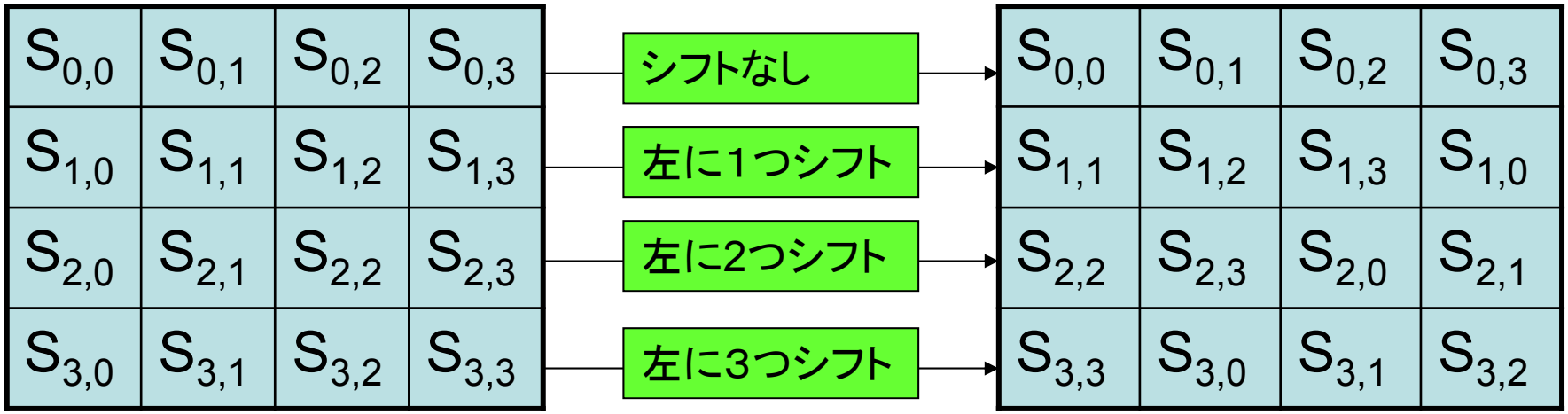
IPA

- 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。

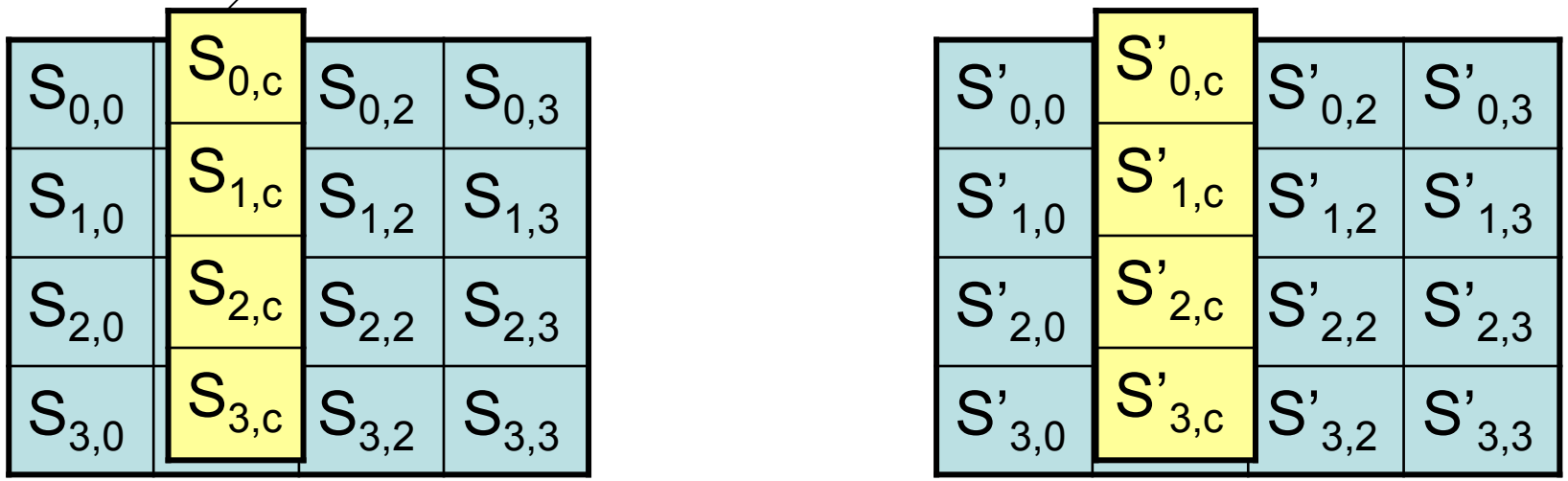


AES：MixColumsの処理

IPA

•4ブロックを1列として，列ごとに列ベクトルの変換。

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



AES: S-Boxの定義の詳細

- 1バイト(8ビット)の値aに対し、
  - 逆元: $c = a^{-1}, a$の$GF(2^8)$における乗法の逆元
    (ただし、$a = 0$のときは$c = 0$)
  - アフィン変換: 出力$s = Mc \oplus b$:

$$\begin{pmatrix} s_7 \\ s_6 \\ s_5 \\ s_4 \\ s_3 \\ s_2 \\ s_1 \\ s_0 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} c_7 \\ c_6 \\ c_5 \\ c_4 \\ c_3 \\ c_2 \\ c_1 \\ c_0 \end{pmatrix} \oplus \begin{pmatrix} 0 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{pmatrix}$$

AES: S-Boxの実装

- テーブル参照
  - ソフトウェア向き

x \ y	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

FIPS197より

- 数学的定義にしたがって逆元計算やアフィン変換を実装
  - GF(2^8)をGF((2^4)^2)やGF(((2^2)^2)^2)と見て逆元演算回路を構成
  - ハードウェア向き

AES: S-Boxの実装

- BitSlice実装の概要
  - ソフトウェア実装
  - アルゴリズムをブール演算の列として記述
  - 非線形であるS-Boxの実装が一番難しい
  - ハードウェア実装で使用される逆元演算の方法をソフトウェアに適用
  - 複数バイトに対する並列動作が可能
  - SIMD命令のあるCPUでは有利
  - テーブル参照がない
    - → キャッシュミス/ヒットへのサイドチャネル攻撃を受けない
  - 以下の条件を満たせば高速実装となる可能性
    - アルゴリズムのビットレベルの複雑さが小さい
    - 実装するCPUのレジスタ数が多い
    - 実装するCPUのレジスタ長が長い
      - x86プロセッサには不向き。x64プロセッサには適正あり。

公開鍵暗号の例: RSA暗号の原理

- *p, q:* 巨大な素数
- *n=pq:* 巨大な素数の積
- *e*: 公開鍵, *d*: 秘密鍵
- 暗号化: $c \equiv m^e \pmod{n}$
- 復号: $m \equiv c^d \pmod{n}$
- 署名: $s \equiv m^d \pmod{n}$
- 署名検証: $m \equiv s^e \pmod{n}$

巨大な数のべき乗剰余演算が必要

べき乗剰余演算の実装

- バイナリ法(Square-and-Multiply)アルゴリズム

入力: M, d
$d = d_1 d_2 \cdots d_n$: dの2進数表現 $(d_i = 0\ or\ 1)$

$S \leftarrow M$
for i from 1 to n-1 do
  $S \leftarrow S * S \bmod N$
  if $d_i = 1$ then
    $S \leftarrow S * M \bmod N$
  end
end
return S

乗算剰余演算の実装

- モンゴメリ乗算
  - *a*b (mod N)* を計算したい
  - 剰余演算は非常に時間がかかる演算である
  - モンゴメリ乗算を使うと、乗算剰余演算の繰り返し時に、時間がかかる剰余演算を最初と最後に1回ずつ行うだけでよくなり、高速化できる。

入力: N, a, b
N: kビット整数
$R = 2^k$
$A = a * R \pmod R, B = b * R \pmod R$

$S \leftarrow A * B$
$S \leftarrow (S + (S * N^{-1} \bmod R) * N)/R$
if $S > N$ then
　　$S \leftarrow S - N$
end
return S

タイミングアタック

- 暗号演算の実行時間の差をサイドチャネル情報として利用

タイミングアタックに使える実行時間の差の発生例

入力: N, a, b
N: kビット整数
$R = 2^k$
$A = a * R \pmod R, B = b * R \pmod R$

$S \leftarrow A * B$
$S \leftarrow (S + (S * N^{-1} \bmod R) * N)/R$
if S > N then
　　$S \leftarrow S - N$
end
return S

← Extra reduction (条件分岐がある)

処理時間の差として現れる

バイナリ法とモンゴメリ乗算を使用したべき乗剰余演算に対するタイミングアタック例

- $M^d(d = d_1 d_2 \cdots d_n : d$の2進数表現 $(d_i = 0\ or\ 1))$のべき乗剰余演算において、$d_1, d_2, \cdots, d_k$ までが既知とする
- d_{k+1}の桁での二乗演算におけるモンゴメリ乗算でExtra Reductionが起こるかどうかは、M とd_{k+1}の値で決定する
- Mをランダムに変化させ、d_{k+1}=0の場合と1の場合とで、Extra Reductionが起こるかどうかでMの集合を2つに分類する
- d_{k+1}の推測が正しければ、Mの集合の分類ごとに、処理時間に差が現れる
- d_{k+1}の推測が誤っていれば、Mの集合の分類はランダムになり、処理時間の差がない
- d_{k+1}が推測できれば、この結果を使って同様の方法でd_{k+2}を推測する

ワークショップの内容紹介

ワークショップ情報

- CHES (Cryptographic Hardware and Embedded Devices)
  - Saint Malo, France
  - 2015/9/13-9/16
  - 参加者は435名
  - 34本の論文が採用
  - 11個のセッション
    - Processing Techniques in Side-Channel Analysis
    - Cryptographic Hardware Implementations
    - Homomorphic Encryption in Hardware
    - **Side-Channel Attacks on Public-Key Cryptography**
    - Cipher Design and Cryptanalysis
    - TRNGs and Entropy Estimations
    - **Side-Channel Analysis and Fault Injection Attacks**
    - Higher-Order Side-Channel Attacks
    - Physically Unclonable Functions and Hardware Trojans
    - **Side-Channel Attacks in Practice**
    - Lattice-Based Implementations
  - 1件の招待講演

ワークショップ情報

- FDTC (Fault Diagnosis and Tolerance in Cryptography)
  - Saint Malo, France
  - 2015/9/13
  - 参加者は114名
  - 10本の論文が採用
  - 4個のセッション
    - **Fault Injection: Models and Techniques**
    - DFA: Models and Techniques
    - Fault Injection Attacks to Cipher Families
    - Fault Attacks to Cryptographic Devices
  - 2件の招待講演

Row Hammer

Yoongu Kim[1], Ross Daly[*], Jeremie Kim[1], Chris Fallin[*], Ji Hye Lee[1], Donghyuk Lee[1], Chris Wilkerson[2], Konrad Lai, and Onur Mutlu[1]

[1]Carnegie Mellon University
[2]Intel Labs

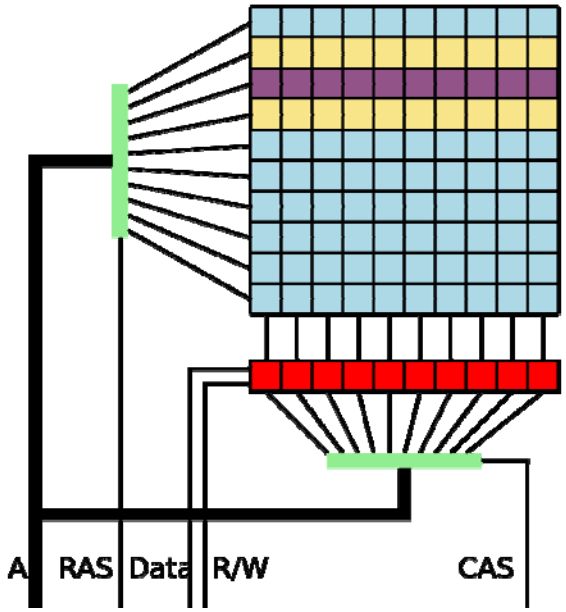
*Work done while at Carnegie Mellon University

http://users.ece.cmu.edu/~yoonguk/papers/kim-isca14.pdf

Row Hammer

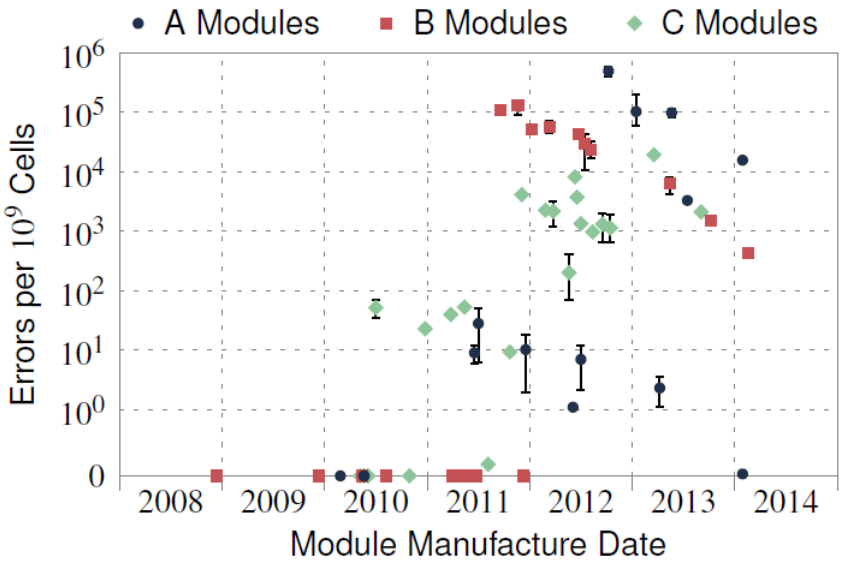
- DRAMの微細化により、メモリセルの同じ行(row)に対する連続したアクセスが、隣接したrowに影響を与えてデータが化ける現象が起こりうるようになっている

- 悪用できるか?



Row Hammer

- 3社のベンダの、129のDRAMモジュールをテスト
- 110のモジュールで、Row Hammer問題の脆弱性が確認された



(Graph from Kim et al)

- 2010年頃からこの問題が現れ始めている
- Row Hammerの悪用によって、Linux kernelの権限昇格を起こしたことがGoogleの研究者によって発表された
  - 仮想メモリと物理メモリのマッピング機構において、誤った物理アドレスをポイントさせることでアクセス可能範囲外の物理メモリへのアクセスを可能にする
- Javascriptを仕込んでWebブラウザに実行させることにより、リモートからクライアントを攻撃する手段ともなる

Who Watches the Watchman?: Utilizing Performance Monitors for Compromising Keys of RSA on Intel Platforms

Sarani Bhattacharya[1], and Debdeep Mukhopadhyay[1]

[1]Department of Computer Science and Enginerring, Indian Institude of Technology Kharagpur, India

Who Watches the Watchman?: Utilizing Performance Monitors for Compromising Keys of RSA on Intel Platforms

IPA

- サイドチャネル情報
  - キャッシュヒット/ミス
  - 分岐予測
- Montgomery MultiplicationのExtra reductionの条件分岐に注目

IPA

- RSAのような公開鍵暗号のアルゴリズムのべき乗剰余演算
  - Binary version of Square & Multiply Exponentiation
- 個々のSquare及びMultiplyでの乗算剰余演算を高速に行うため
  - Montgomery Multiplication
    - Extra reductionをするかどうかの判定に条件分岐がある

Who Watches the Watchman?: Utilizing Performance Monitors for Compromising Keys of RSA on Intel Platforms

IPA

- 分岐予測をミスすると、ペナルティが発生して実行時間が延びる → タイミング攻撃が可能
- タイミングは、分岐予測ミス以外にも影響される (ノイズ)
- Intelプロセッサのハードウェアパフォーマンスカウンタ(HPC)には、分岐予測ミス回数の情報もある
  - Linuxのperfコマンドはユーザ権限で動作し、HPCの情報を取得できる
  - 分岐予測ミスの回数をサイドチャネル情報として利用する

Exclusive Exponent Blinding May Not Suffice to Prevent Timing Attacks on RSA

Werner Schindler

Bundesamt für Sicherheit in der Informationstechnik (BSI), Germany

https://eprint.iacr.org/2014/869.pdf

Exclusive Exponent Blinding May Not Suffice to Prevent Timing Attacks on RSA

- RSA-CRTの実装
  - $N = p_1 p_2$
  - d: private key, $d_1 = d \mod p_1 - 1$, $d_2 = d \mod p_2 - 1$
  - $s_1 = m^{d_1} \bmod p_1$
  - $s_2 = m^{d_2} \bmod p_2$
- サイドチャネル攻撃対策: Exponent Blinding
  - $d_{1,b} = d_1 + r_1(p_1 - 1)$, $d_{2,b} = d_1 + r_2(p_2 - 1)$
  - $s_1 = m^{d_{1,b}} \bmod p_1$
  - $s_2 = m^{d_{2,b}} \bmod p_2$

Exclusive Exponent Blinding May Not Suffice to Prevent Timing Attacks on RSA

- Exponent Blindingは、タイミングアタックを完全に防ぐと考えられていた
- しかし、それは一般に正しくないことを示す
- 以下の対策が考えられる
  - モンゴメリ乗算において、$R>4p_1, 4p_2$とすることでExtra Reductionを完全になくす
  - base blindingも併用する

IPA

Stealing Keys from PCs Using a Radio: Cheap Electromagnetic Attacks on Windowed Exponentiation

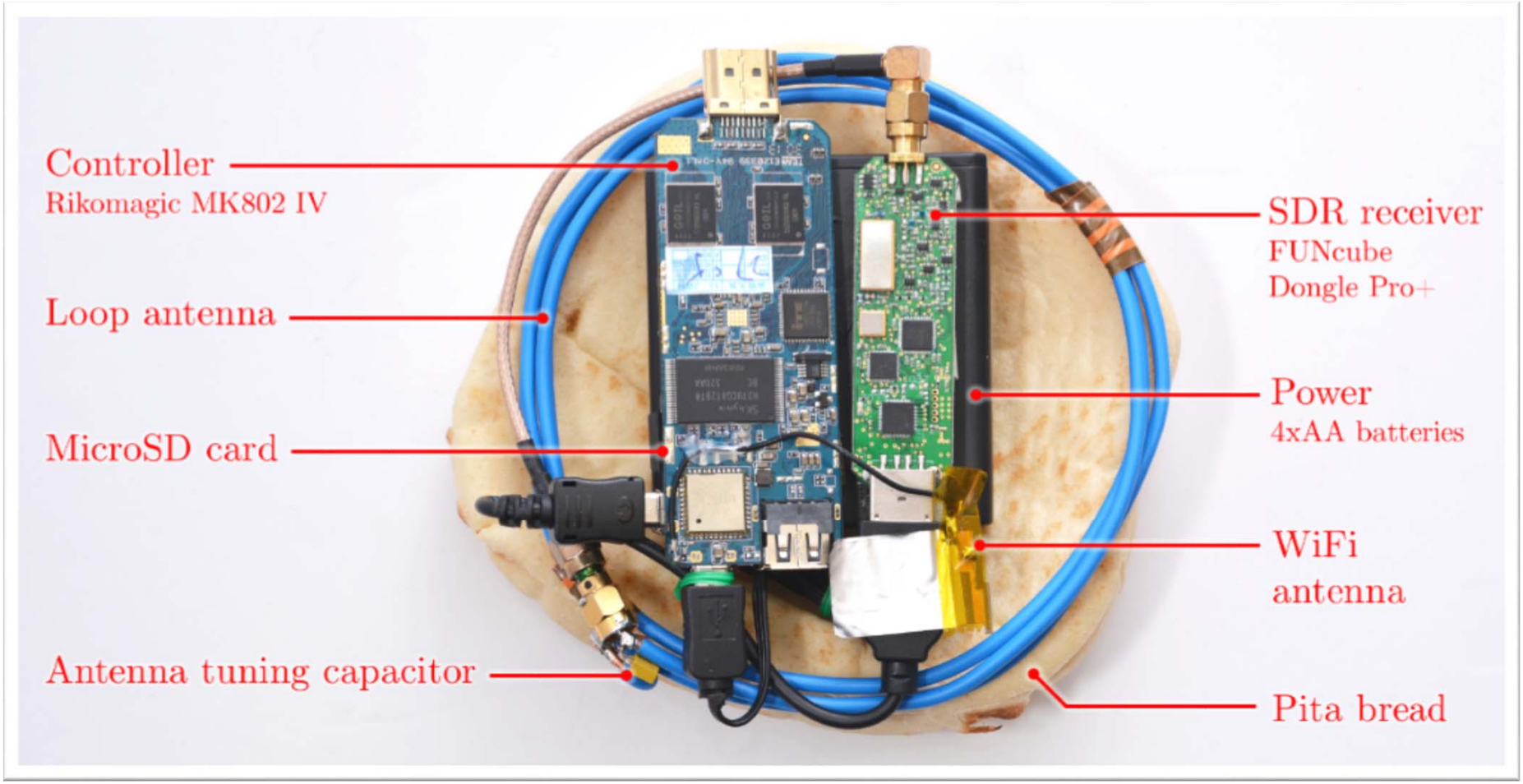
Daniel Genkin, Lev Pachmanov, Itamar Pipman, and Eran Tromer

Department of Computer Science and Engg. IIT Kharagpur, India

- ノートパソコンに安価なSDR (Software Defined Radio)を近づけて電磁解析攻撃を行って鍵を暴露する
- 一般用ラジオでも攻撃可能
- GnuPGのRSA暗号及びElGamal暗号
  - 乗算にはWindow Methodを使用している
- 電磁解析の波形取得は、低い帯域で十分

IPA

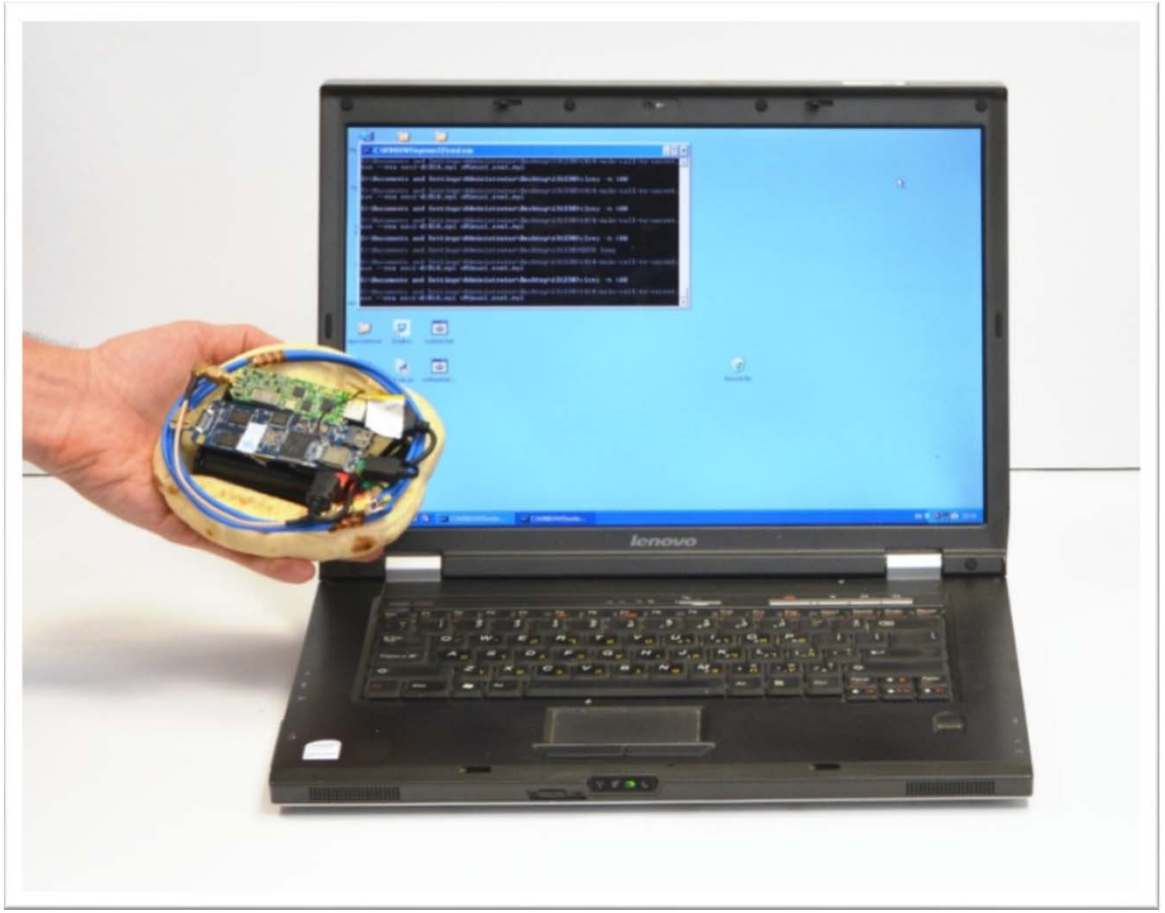
- Portable Instrument for Trace Acquisition (PITA)



https://www.cryptoexperts.com/ches2015/slides/day2/session1/talk1.pptx

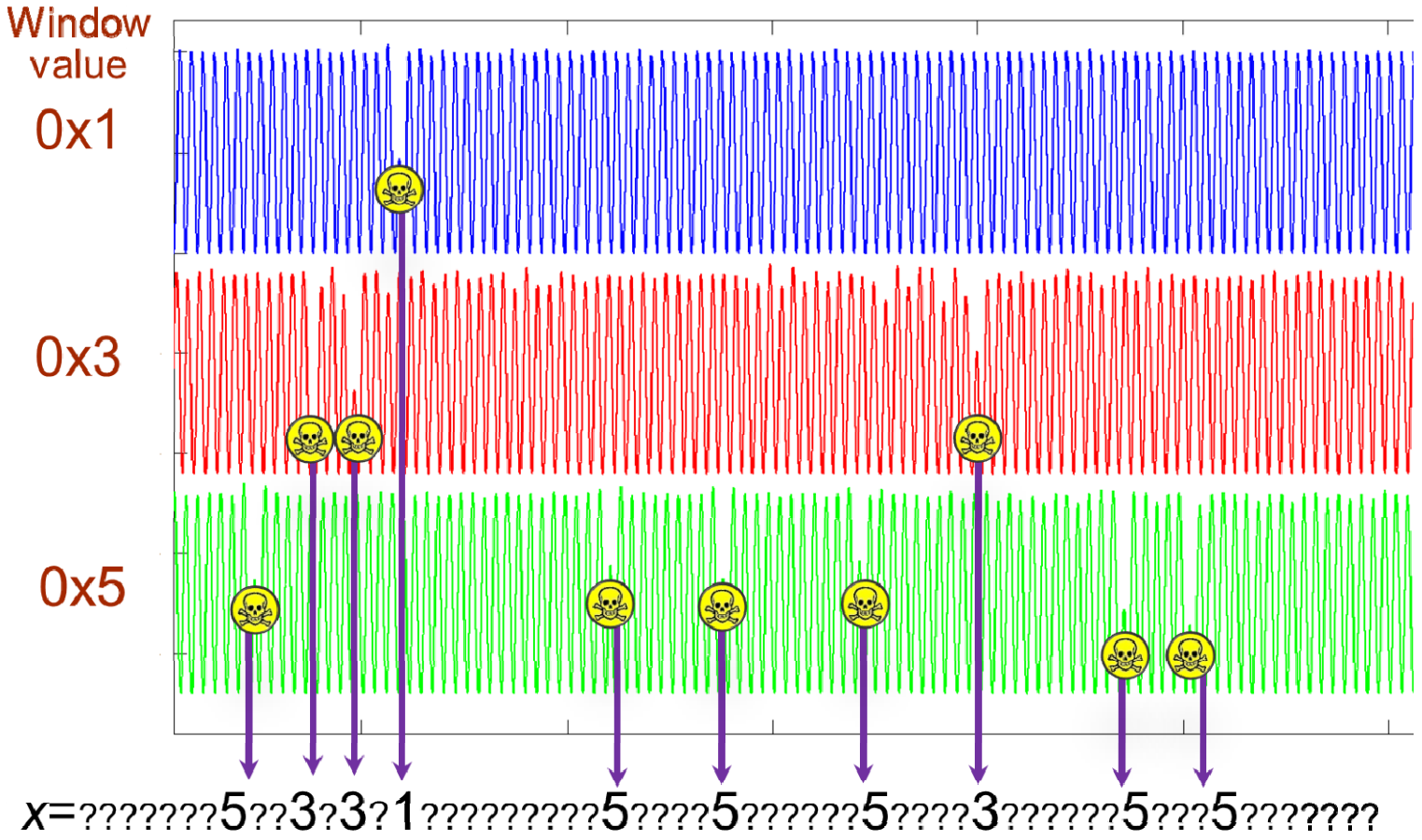
IPA

- Electromagnetic Attack



https://www.cryptoexperts.com/ches2015/slides/day2/session1/talk1.pptx

- 信号処理技術を駆使して、鍵が抽出できる
  - 参照するテーブルのインデックスが現れる



https://www.cryptoexperts.com/ches2015/slides/day2/session1/talk1.pptx

IPA

- 一般のラジオでも攻撃できる
  - ラジオで電波を受信 → 音声として記録

https://www.cryptoexperts.com/ches2015/slides/day2/session1/talk1.pptx

DPA, Bitslicing and Masking at 1 GHz

Josep Balasch, Benedikt Gierlichs, Oscar Reparaz, and Ingrid Verbauwhede

Department of Electrical Engineering-ESAT/COSIC and iMinds, KU Leuven

https://eprint.iacr.org/2015/727

DPA, Bitslicing and Masking at 1 GHz

IPA

- 暗号が、スマートカード等の小さいCPUへの実装だけでなく、メインプロセッサのソフトウェアによる実装の事例が拡大している
- サイドチャネル攻撃の研究は、近年のギガヘルツ級のクロックで動く複雑なプロセッサに対しても適用できるか?

DPA, Bitslicing and Masking at 1 GHz

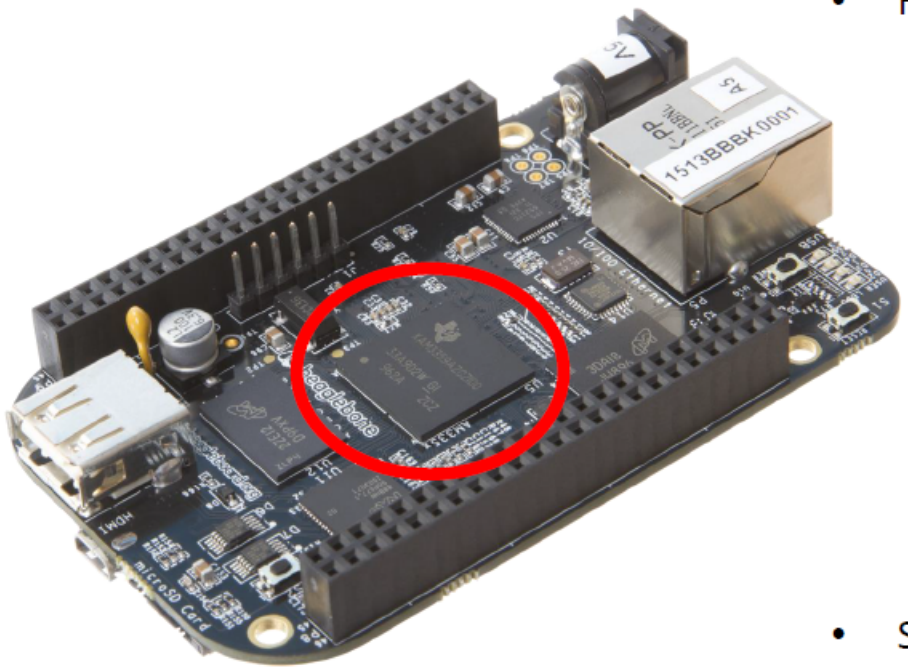
IPA

- Bitslice実装
  - 長いレジスタをうまく使用することで、複数バイトに対する並列動作が可能
  - 鍵の値に依存するテーブル参照が不要 → キャッシュタイミング攻撃に対して安全
    - AESのS-Box実装も、テーブル参照ではなく、有限体の逆元演算をXORなどの論理演算の組み合わせで実行

DPA, Bitslicing and Masking at 1 GHz

IPA

- 評価対象
  - BeagleBone Black single board computer

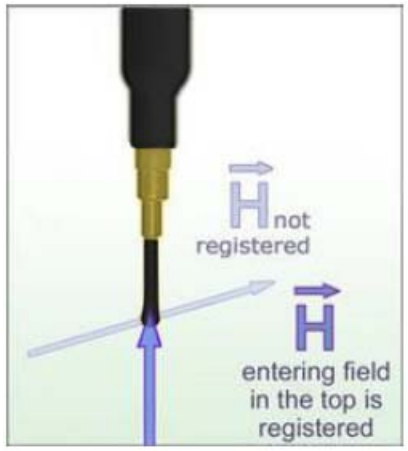


- Hardware: Texas Instruments Sitara SOC
  - DDR3 memory controller, 3D graphics, HDMI, …
  - USB, Ethernet
  - ARM Cortex-A8 processor
    - Apple Iphone4, Samsung Galaxy S, …
    - 32-bit processor
    - 13 stage pipeline
    - Dynamic branch prediction
    - L1 and L2 cache
    - Up to 1 GHz clock frequency
- Software: Complete Linux distribution
  - OS image on embedded MMC
  - 102 processes incl. X, SSH, Apache2, etc.

https://www.cryptoexperts.com/ches2015/slides/day3/session3/talk1.pdf

DPA, Bitslicing and Masking at 1 GHz

- ARMコアのサイドチャネル情報の測定
  - 非接触電力測定 (デカップリングコンデンサからの電磁放射)
  - プローブのタイプ、位置、向きは重要



Magnetic near field probe
(30 MHz to 3 GHz)

```
…
while (1) {
   SLEEP
   DO_SOMETHING
   SLEEP
}
…
```

https://www.cryptoexperts.com/ches2015/slides/day3/session3/talk1.pdf

DPA, Bitslicing and Masking at 1 GHz

IPA

- 攻撃結果
  - 対策なしの実装に対しては、10000波形からの普通のCPAで鍵の暴露に成功
    - Bitslice実装に合わせ、32ビット中2ビットのハミングウェイトから攻撃
    - 比較的容易に攻撃が成功した
  - 対策あり(マスキング)の実装に対しては、
    - 1st order attack → 120万波形ほど必要
    - 2nd order attack → 解析する時刻が分かっていれば40万波形程度で攻撃成功するが、すべてのタイムサンプルのペアを試すのは計算量的に非常に高価である

DPA, Bitslicing and Masking at 1 GHz

IPA

- 結論
  - ギガヘルツ級のクロックで動作する複雑で高性能なプロセッサ上で複雑なOSが動いているターゲットへのサイドチャネル攻撃を実行
  - 対策なしの実装に対しては、攻撃は比較的容易に成功した(10000波形で成功)
  - ただし、トリガーとアラインメントは難しい
  - ゲートレベルのマスキングは対策として有効

SoC It to EM: ElectroMagnetic Side-Channel Attacks on a Complex System-on-Chip

J. Longo[1], E. De Mulder[2], D.Page[1], and M. Tunstall[2]

[1]Department of Computer Science, University of Bristol
[2]Rambus Cryptography Research Division

SoC It to EM: ElectroMagnetic Side-Channel Attacks on a Complex System-on-Chip

- SoC (System-on-Chip) に対するサイドチャネル攻撃の可能性

- サイドチャネル攻撃に関しては以下のことが信じられている
  - 高速なクロックサイクル → 高いサンプリングレートの装置が必要
  - 複雑な組み込みシステム → 攻撃が困難
  - 高度な並列処理 → 低いSN比 ～ 高いサイドチャネル耐性

SoC It to EM: ElectroMagnetic Side-Channel Attacks on a Complex System-on-Chip

IPA

- 以下のプラットフォームを攻撃対象に

BeagleBone Black	Attack Environment
	Hardware: ▸ ARM Cortex-A8 1 GHz CPU (High clock rate) ▸ ARM NEON SIMD (High degree of parallelism) ▸ TI proprietary cryptographic hardware (RNG, SHA-1, AES) Software: ▸ Debian Wheezy (3.15) (Full unmodified Linux distribution) ▸ OpenSSL 1.0.1*j* (Bulk encryption)

https://www.cryptoexperts.com/ches2015/slides/day3/session3/talk2.pdf

SoC It to EM: ElectroMagnetic Side-Channel Attacks on a Complex System-on-Chip

IPA

- 攻撃結果

Summary of Attack Results

Implementation	Hardware	Trigger	Acquisitions	Data
T-tables	ARM core	GPIO-based	3000	46 kB
T-tables	ARM core	Network-based	100	400 kB
Hardware	Co-processor	DMA-based	500 000	7GB
Bit-sliced	NEON core	GPIO-based	5000	625 kB

- 結論
  - 高いクロックサイクルは、必ずしも高いサンプリングレートの必要性を意味しない。
  - 攻撃対象の複雑さは、必ずしも複雑な攻撃の必要性を意味しない
  - 高度なハードウェア機能(NEON SIMDなど)は実装の高速化に寄与できるが、新たなサイドチャネルリーケージも引き起こしうる。
  - セキュアな組み込みシステムのためのOSレベルソフトウェアは、サイドチャネル攻撃についても考慮すべきである。

https://www.cryptoexperts.com/ches2015/slides/day3/session3/talk2.pdf

IPA

Improved Side-Channel Analysis of Finite-Field Multiplication

Sonia Belaïd[1], Ordas, Jean-Sébastien Coron[2], Pierre-Alain Fouque[3], Benoît Gérard[4], Jean-Gabriel Kammerer[5], and Emmanuel Prouff[6]

[1]École Normale Supérieure and Thales Communications and Security
[2]University of Luxembourg
[3]Université de Rennes 1 and IRISA
[4]DGA/MI and IRISA
[5]DGA/MI and IRMAR
[6]ANSSI

Improved Side-Channel Analysis of Finite-Field Multiplication

- 有限体(例: GF(2^{128}))上の乗算に対するサイドチャネル攻撃
- Galois Counter Mode (GCM)などに適用できる

Improved Side-Channel Analysis of Finite-Field Multiplication

- 普通のAESへのサイドチャネル攻撃
  - 鍵(128ビット)を、1バイト(8ビット)ごとに攻撃する
  - 8ビットへの攻撃を繰り返して、全ビットへの攻撃を完成させる
    → Divide and Conquer Approach
- 有限体の乗算へのサイドチャネル攻撃
  - 演算の性質上、8ビットごとに分割して攻撃することは不可能
    → Divide and Conquer Approach は使えない

Improved Side-Channel Analysis of Finite-Field Multiplication

- Hidden Multiplier Problem
  - $v = M \otimes H$ ($\otimes$: GF(2^{128})での乗算, *M*: 既知, *H*: 秘密の値)の演算時に、*H*の値を暴露したい
  - vのHamming Weight (HW(v))が漏れるとする

Improved Side-Channel Analysis of Finite-Field Multiplication

- 既知の攻撃方法([BFG14])
  - HW(M⊗H)のLSBを観測
  - LSBは、MとHの各ビットの linear functionであることを利用
  - 線形連立方程式を構成
  - 解を求める (ただし、エラーがあることを考慮)
- 問題点
  - HWの下位ビットをとるため、観測のノイズに大きく影響される

LSB of the first multiplication output's Hamming weight:

$$b_0 \stackrel{\text{def}}{=} \mathrm{lsb}_0\left(\mathrm{HW}(M \otimes_P H)\right) = \bigoplus_{0 \leqslant i \leqslant 127} (M \otimes_P H)_i$$

$$= \bigoplus_{0 \leqslant j \leqslant 127} \left(\bigoplus_{0 \leqslant i \leqslant 127} (M_P)_{i,j} \right) h_j$$

Linear system to solve:

$$\mathcal{S} = \begin{cases} \bigoplus_{0 \leqslant j \leqslant 127} \left(\bigoplus_{0 \leqslant i \leqslant 127} (M_P^{(0)})_{i,j} \right) h_j = {b_0}^{(0)} \\ \bigoplus_{0 \leqslant j \leqslant 127} \left(\bigoplus_{0 \leqslant i \leqslant 127} (M_P^{(1)})_{i,j} \right) h_j = {b_0}^{(1)} \\ \cdots \\ \bigoplus_{0 \leqslant j \leqslant 127} \left(\bigoplus_{0 \leqslant i \leqslant 127} (M_P^{(t-1)})_{i,j} \right) h_j = {b_0}^{(t-1)} \end{cases}$$

http://des.cse.nsysu.edu.tw/asiacrypt2014/doc/19-1_Side-Channel%20Analysis%20of%20Multiplications%20in%20$GF(2%5E%7B128%7D)$%20Application%20to%20AES-GCM.pdf

Improved Side-Channel Analysis of Finite-Field Multiplication

- 攻撃の改良
  - HWが高いものと低いものを選別して攻撃することで、より低いSNRに対して攻撃可能となる

Usual cases:

$\mathcal{L}(v)$ low ➜ $v \approx 0$ $\qquad$ $\mathcal{L}(v)$ high ➜ $v \approx 2^n - 1$

$$\begin{cases} v_0 = \bigoplus_{0 \leqslant j < n} \left(\bigoplus_{i \in I^{(0,j)}} m_i \right) k_j = 0 \\ v_1 = \bigoplus_{0 \leqslant j < n} \left(\bigoplus_{i \in I^{(1,j)}} m_i \right) k_j = 0 \\ \vdots \quad \vdots \quad \vdots \\ v_{n-1} = \bigoplus_{0 \leqslant j < n} \left(\bigoplus_{i \in I^{(n-1,j)}} m_i \right) k_j = 0 \end{cases} \qquad \begin{cases} v_0 = \bigoplus_{0 \leqslant j < n} \left(\bigoplus_{i \in I^{(0,j)}} m_i \right) k_j = 1 \\ v_1 = \bigoplus_{0 \leqslant j < n} \left(\bigoplus_{i \in I^{(1,j)}} m_i \right) k_j = 1 \\ \vdots \quad \vdots \quad \vdots \\ v_{n-1} = \bigoplus_{0 \leqslant j < n} \left(\bigoplus_{i \in I^{(n-1,j)}} m_i \right) k_j = 1 \end{cases}$$

with an error probability p

- 既知の攻撃では、SNR=128程度が必要だったが、この攻撃ではSNR=8程度で攻撃が成功する

Finding the AES Bits in the Haystack: Reverse Engineering and SCA Using Voltage Contrast

Christian Kison[1,2], Jürgen Frinken[2], Christof Paar[1]

[1]Horst Görtz Institute for IT Security, Ruhr University Bochum, Bochum, Germany
[2]Bundeskriminalamt, Kriminaltechnisches Institut, Wiesbaden, Germany

Finding the AES Bits in the Haystack: Reverse Engineering and SCA Using Voltage Contrast

- ハードウェアのリバースエンジニアリングには、Region of Interest (ROI) を見つけなければならない
  - 1層ごと剥ぐ
  - 電磁放射解析
  - Photon Emission
  - 熱放出解析

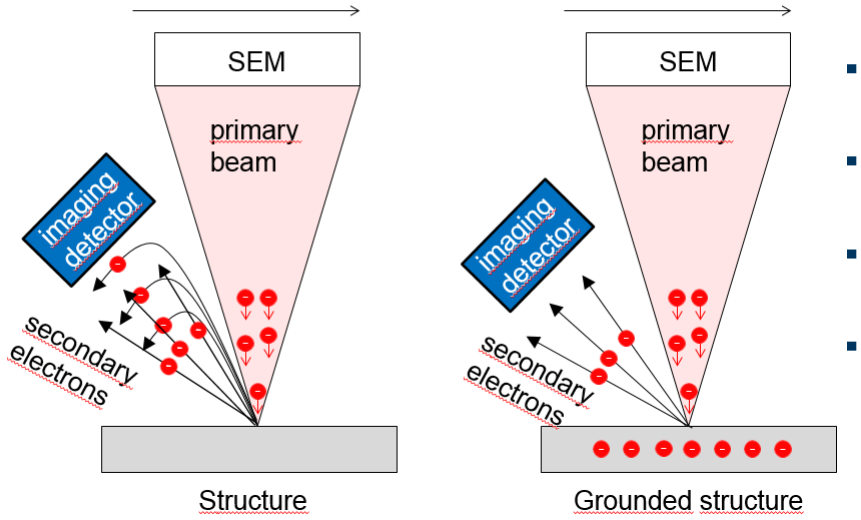
  - どの方法も、課題がある

Finding the AES Bits in the Haystack: Reverse Engineering and SCA Using Voltage Contrast

- Voltage Contrast
  - 走査型電子顕微鏡(SEM)の像
- Static Voltage Contrast
- Dynamic Voltage Contrast
  - Capacitive Coupled Voltage Contrast (CCVC)

Voltage Contrast SCA

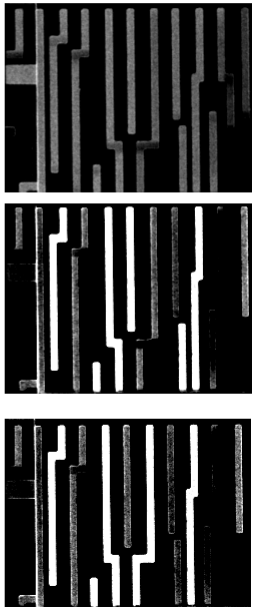
Passive/Active Voltage Contrast



Voltage Contrast SCA

VCA in a SEM

- Neutral (not connected)
- Working (cycle x)
- Working (cycle x+1)
- Slow SEM
  - Some kHz
  - **External clock control!**



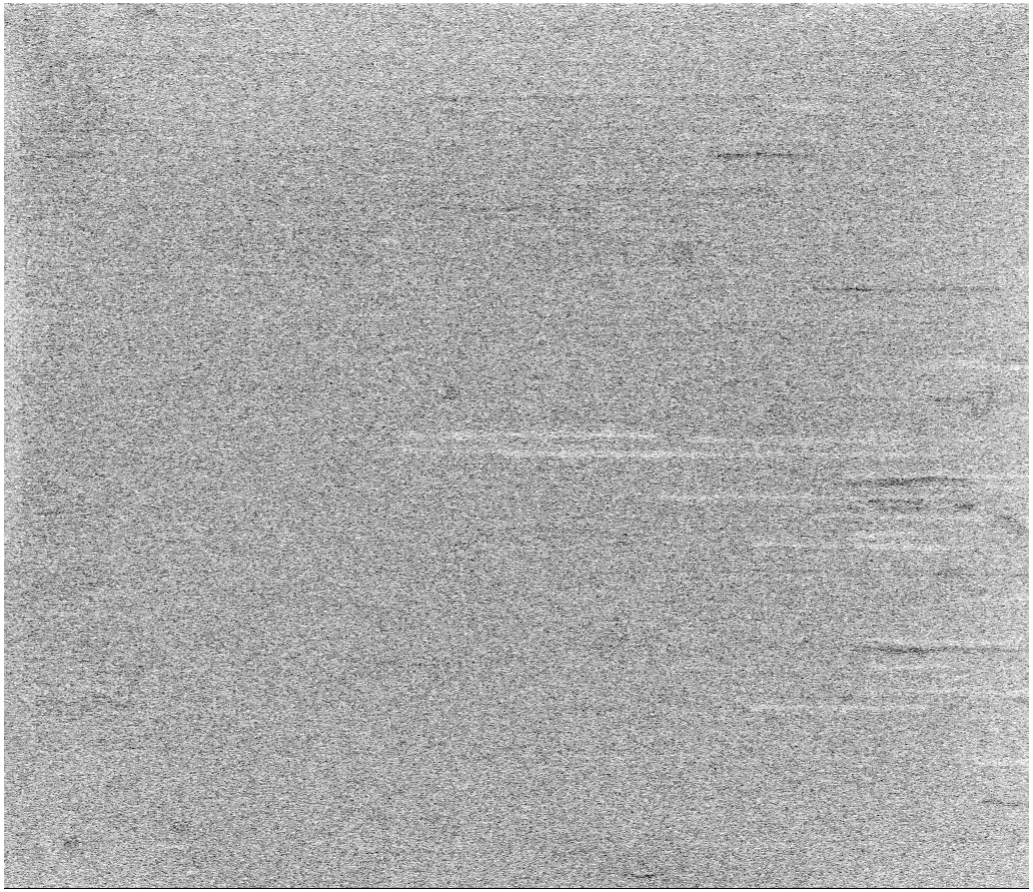
Finding the AES Bits in the Haystack: Reverse Engineering and SCA Using Voltage Contrast

- Voltage Contrast Analysis
  - AES回路の場所を特定
- Voltage Contrast Side Channel Analysis
  - Voltage Contrast Traceを取得 – 1クロックごとに1コマの動画を撮る
  - AESのビットの配線をVCSCAで特定
    - 全ピクセルと、エミュレートしたAESのビット値との相関係数をとる
- Template Attack with VCSCA
- Simple VCSCA
  - Addroundkeyの場所とタイミングが分かっていれば、平文とroundkeyのビットが分かるので、xorすることでroundkeyのビットが分かる

Finding the AES Bits in the Haystack: Reverse Engineering and SCA Using Voltage Contrast

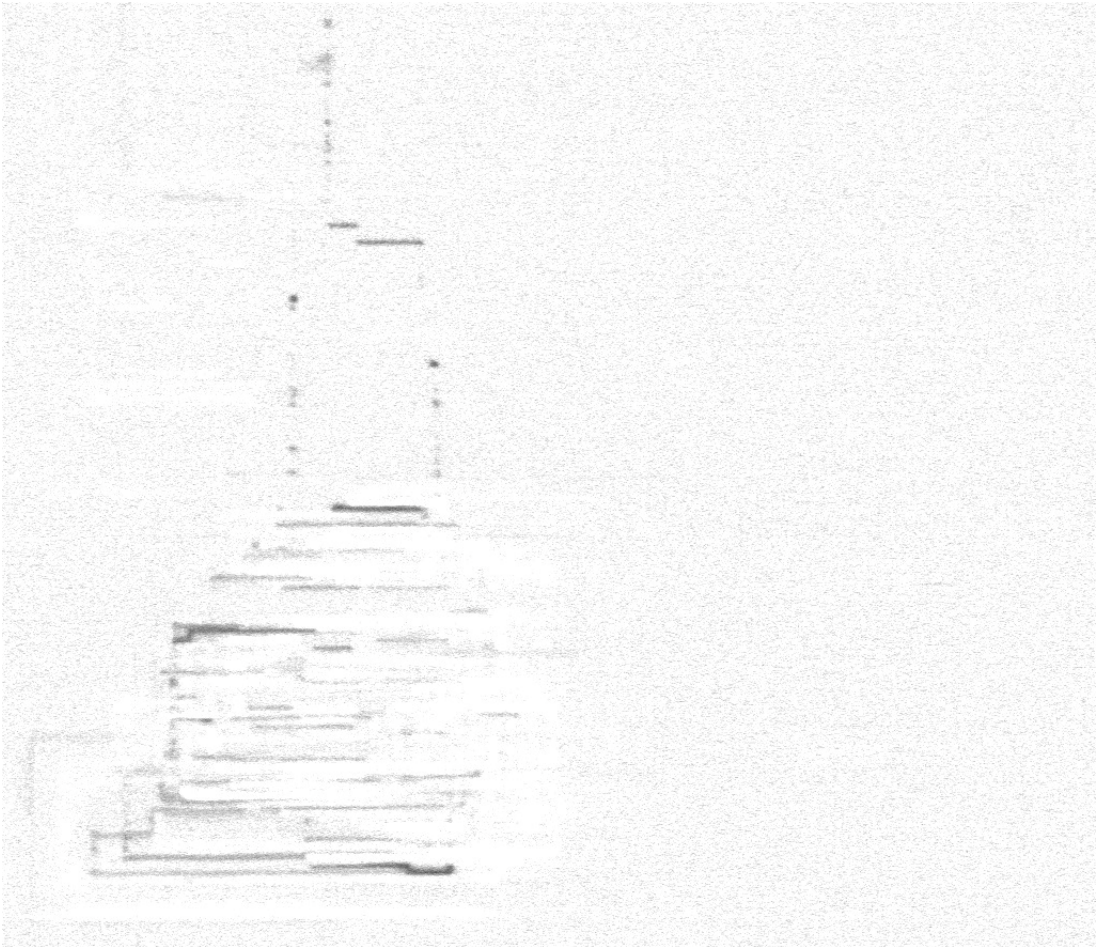
IPA

- Trace Video – 5 Frames / 1 Clock Cycle



Finding the AES Bits in the Haystack: Reverse Engineering and SCA Using Voltage Contrast

- 仮定
  - 鍵は不明
  - 平文は既知
- 100トレースを取得
- AES128ビットの完全な鍵の取得に成功



EM Injection: fault model and locality

S. Ordas, Eliane Jaulmes[1], L. Guillaume-Sage[1], P. Maurine[1,2]

LIRMM, Univ. of Montpellier, France
CEA-TECH/ LIRMM, Univ. of Montpellier France

EM Injection: Fault Model and Locality

- ICのEM fault injectionへの感受性を分析
- EM fault injectionで、ICに現実に起こっているfault modelは?
  - Timing Fault Model
    - EM injectionが、十分な電圧降下を引き起こしてデータの伝播遅延を増大させることでタイミング違反を引き起こせる
  - Sampling Fault Model
    - EM injectionが、(例えば)D Flip Flopの入力(D, CK, Set Reset)のamplitudeを変更できる

EM Injection: Fault Model and Locality

- FPGAでテスト用チップを作成して、実験
- 結果を考察すると、実際に起こっているのは Sampling Faultと思われる

On the Complexity Reduction of Laser Fault Injection Campaigns using OBIC Measurements

Falk Schellenberg*, Markus Finkeldey†, Bastian Richter*, Maximilian Schäpers†, Nils Gerhardt†, Martin Hofmann† and Christof Paar*

* Horst Görtz Institute for IT-Security
† Photonics and Terahertz-Technology
Ruhr University Bochum, Bochum, Germany

On the Complexity Reduction of Laser Fault Injection Campaigns using OBIC Measurements

IPA

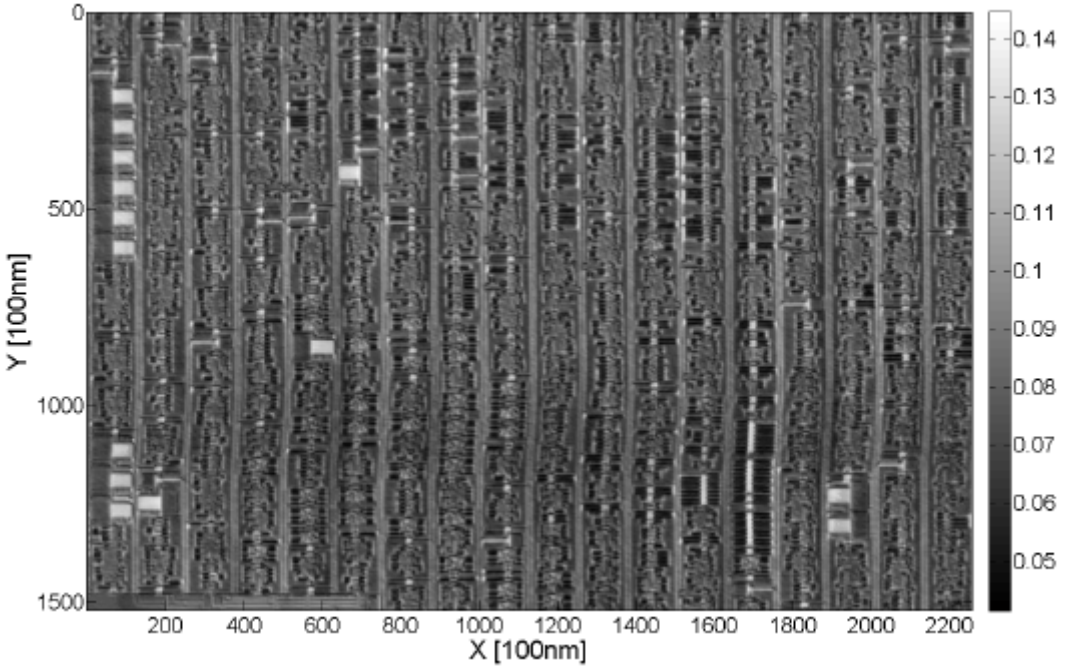
- 故障利用攻撃は、パラメタが非常に多く、攻撃を成功させるための正しいパラメタを見つけることは容易ではない
  - タイミング、強度、持続時間、…
- 特に、レーザー照射の場合は、照射位置もパラメタである
  - X座標、Y座標
- すべてのパラメタの組み合わせを探索すると、探索空間が巨大になりすぎることがある
- 場所の探索空間を減らしたい

On the Complexity Reduction of Laser Fault Injection Campaigns using OBIC Measurements

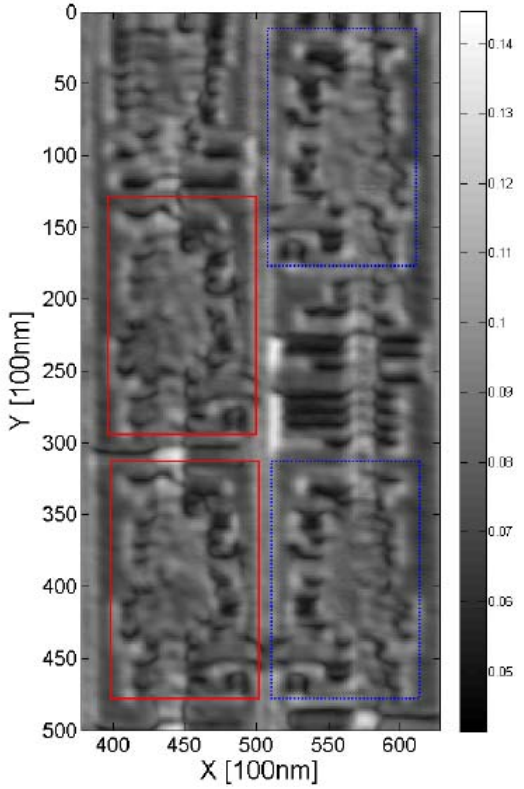
IPA

- OBIC (Optical Beam Induced Current): 光を当てることによって誘導される電流
- PN接合部に光が当たると電流が流れるので、その電流を測定することでPN接合の位置が分かる
- 相関係数を用いたパターン認識でフリップフロップの位置を見つける → レーザーのターゲット候補

On the Complexity Reduction of Laser Fault Injection Campaigns using OBIC Measurements

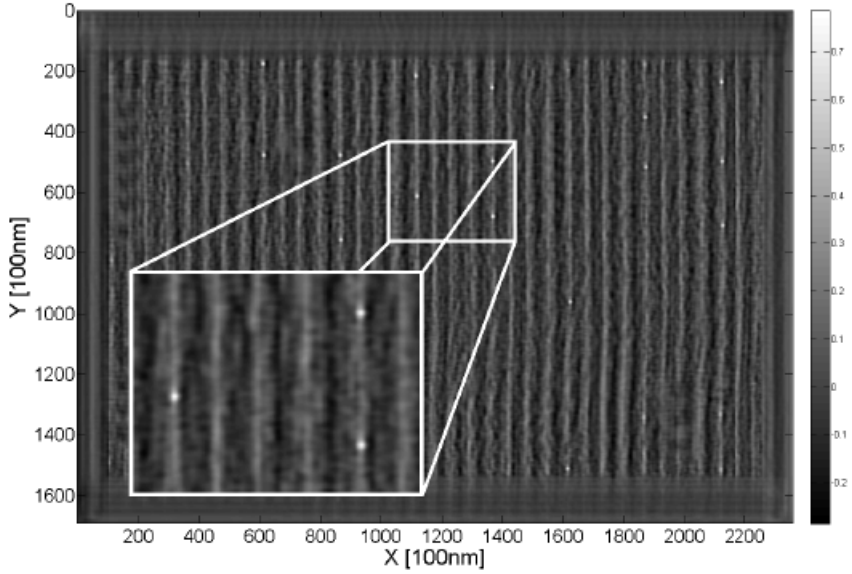


OBICの像。グレースケールがOBICの大きさを表す

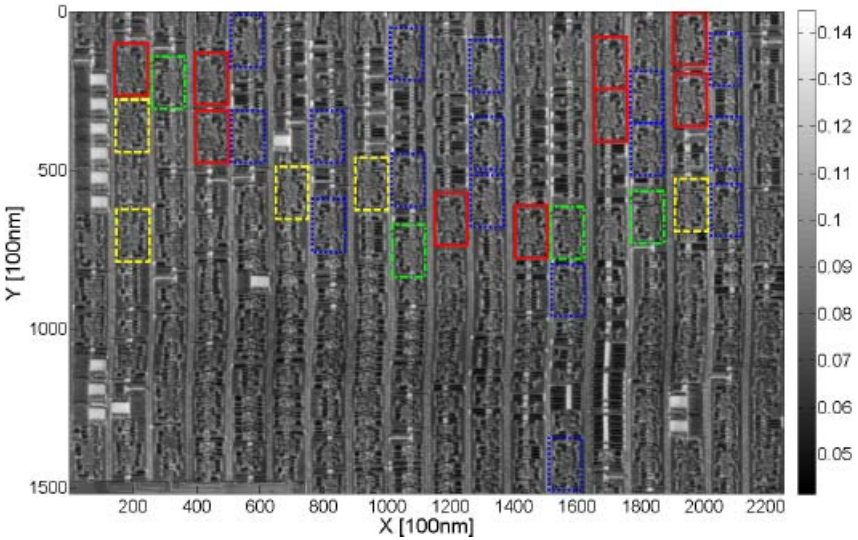


フリップフロップと思われるパターン。色は鏡像反転を表す

On the Complexity Reduction of Laser Fault Injection Campaigns using OBIC Measurements

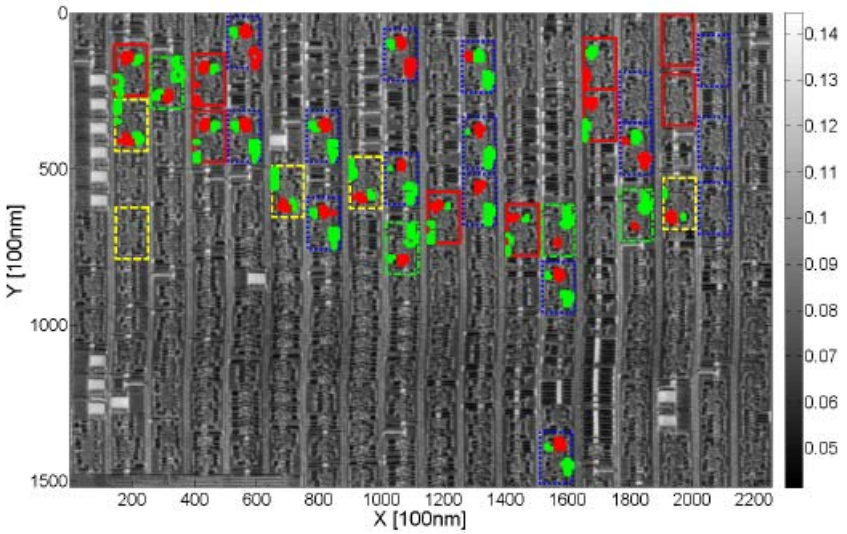


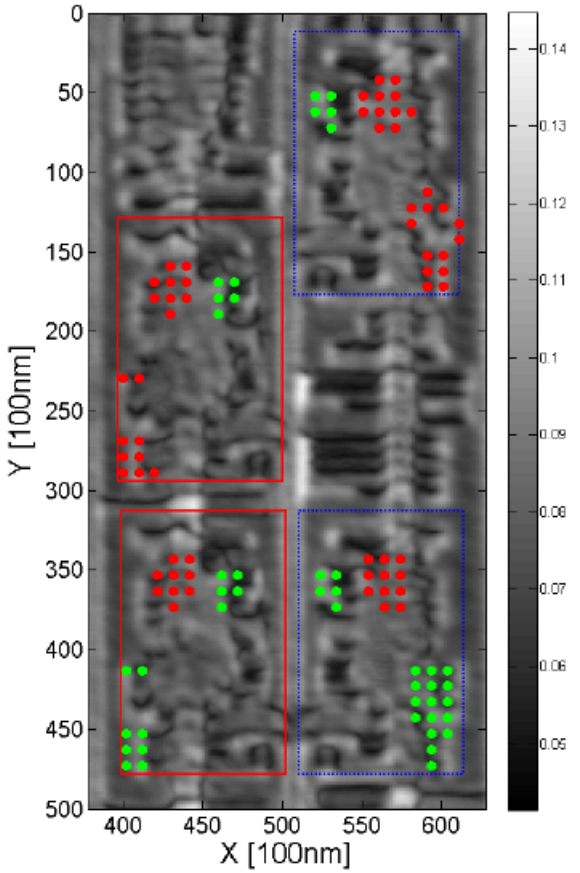
相関係数によるパターン認識でフリップフロップの場所を探索



見つかったフリップフロップをマーク。

On the Complexity Reduction of Laser Fault Injection Campaigns using OBIC Measurements





発見したLFI (Laser Fault Injection)に感受性のある場所。緑の点はビットセット、赤の点はビットリセットを起こす

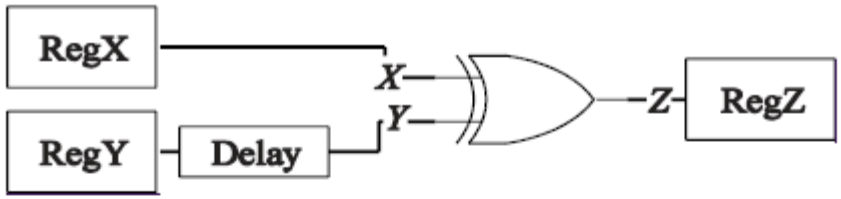
Transient-Steady Effect Attack on Block Ciphers

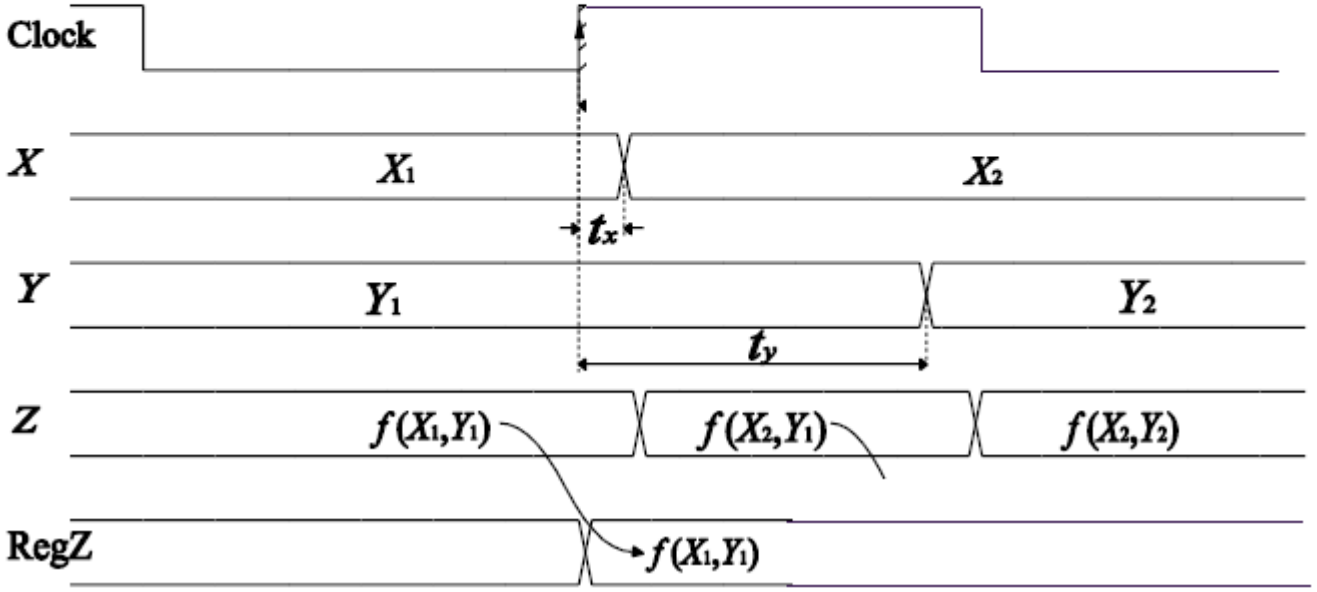
Yanting Ren[1,2], An Wang[1,2], and Liji Wu[1,2]

[1]Tsinghua National Laboratory for Information Science and Technology (TNList), Beijing, China
[2]Institute of Microelectronics, Tsinghua University, Beijing, China

Transient-Steady Effect Attack on Block Ciphers

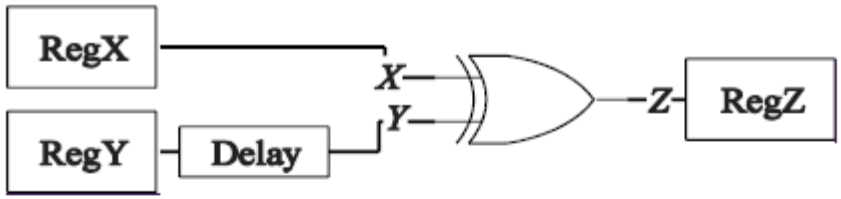
- 基本的なアイディア

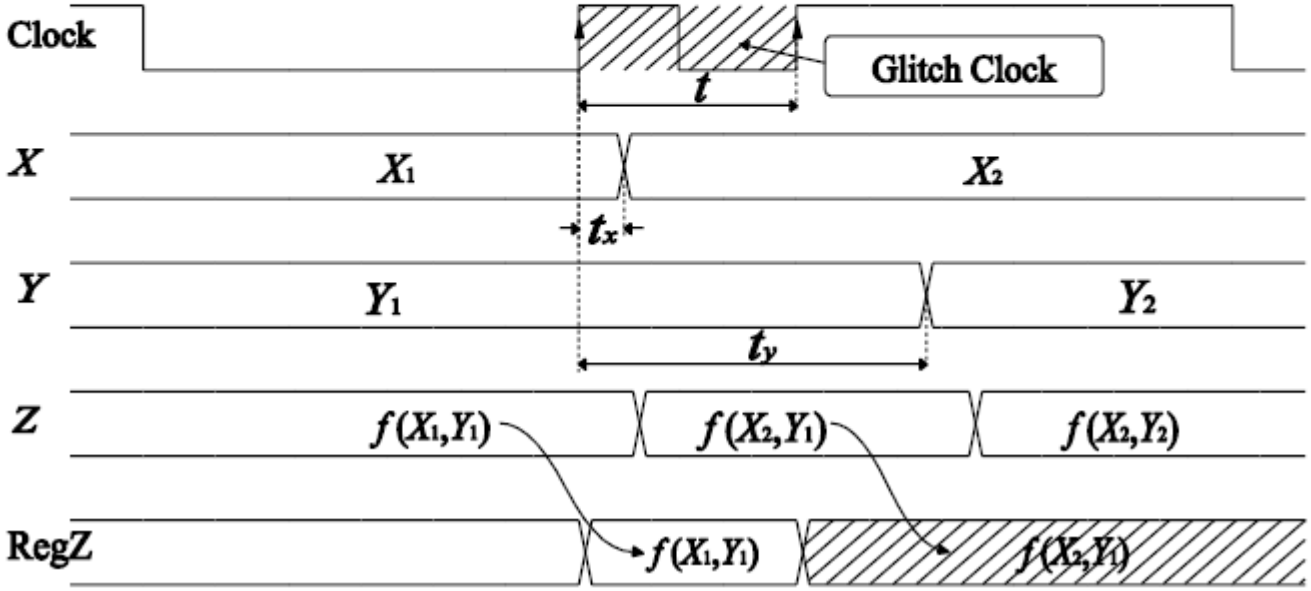




Transient-Steady Effect Attack on Block Ciphers

- 基本的なアイディア





Transient-Steady Effect Attack on Block Ciphers

- 論理ゲートに到着する2個の信号の到達時間の差を利用する
- 一方の信号が到着し、もう一方の信号が到着する前のタイミングでクロックグリッチを印加すると、一方の値の変化だけが反映された値がレジスタに格納される
- 例えば、P1⊕K1 → P2⊕K2と変化するはずが、P1⊕K1 → P1⊕K2となれば、K1とK2の差分(K1⊕K2)が分かる
- 同様に、K2⊕K3, K3⊕K4, …, K15⊕K16が判明すると、K1の値からすべての鍵の値が確定することになる
- グリッチ印加のタイミングでのはシビアである

まとめ

まとめ

- Row Hammer: DRAMの高集積化に伴う新たな脆弱性
  - リモートからのfault injection攻撃の可能性
- OSが動いているような、複雑なハードウェア機器に対する攻撃
  - 攻撃対象は限られたリソースのスマートカードだけではない
- 古くて新しいタイミングアタック
  - タイミングアタックもまだ研究されている
- HPCのような新しいサイドチャネル
  - 新たな利用可能なサイドチャネルが考え出されている
- 故障利用攻撃の実践的工夫
  - パラメタの探索空間を減らす方法が考えられている
- 攻撃の新しいアイディア
  - Transient-Steady Effect Attackのような新しいアイディア。今後の研究の進展に注目。

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

IPAの取り組み

- ハードウェアセキュリティに関する技術セミナーの開催

2015年6月8日 2015年6月30日	ハードウェアセキュリティセミナー (導入コース) 終了
2015年8月27日 2015年9月7日	ハードウェアセキュリティセミナー (技術コース・入門編) 終了
2015年12月16日 2016年1月13日	ハードウェアセキュリティセミナー (技術コース・実践編) 満員御礼。第3回の開催検討中。

- 2016年度も開催予定

- 最新技術動向に関するセミナーも随時開催
  - 次回は2016年1月を予定
  - CARDIS, CARTES, BlackHat Europeの内容を紹介する予定

セミナー情報: https://www.ipa.go.jp/security/jcmvp/seminar/

参考文献

- [FIPS197] National Institute of Standards and Technology (NIST): Advanced Encryption Standard (AES) FIPS Publication 197. http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf , Nov 2001
  AESの定義
- [DKLMQ] Dhem, J.-F., Koeune, F., Leroux, P.-A., Mestré, P.-A., Quisquater, J.-J., Willems, J.-L.: A practical implementation of the timing attack. In: Quisquater, J.-J., Schneier, B. (eds.) Smart Card - Research and Applications, LNCS, pp. 175–191. Springer, Berlin (2000)
  タイミングアタックについて
- [SD15] M. Seaborn, T. Dullien, Exploiting the DRAM rowhammer bug to gain kernel privileges, googleprojectzero.blogspot.com, http://googleprojectzero.blogspot.com/2015/03/exploiting-dram-rowhammer-bug-to-gain.html
  Row Hammerを悪用してLinuxの権限昇格を起こす方法について
- [AKS06] Acıiçmez, O., Koç, Ç.K., Seifert, J.-P.: Predicting secret keys via branch prediction. In: Abe, M. (ed.) CT-RSA 2007. LNCS, vol. 4377, pp. 225–242. Springer, Heidelberg (2006), https://eprint.iacr/org/2006/288.pdf
  分岐予測を利用したサイドチャネル攻撃について
- [BFG14] Belaïd, S., Fouque, P.-A., Gérard, B.: Side-Channel analysis of multiplications in GF(2^{128}). In: Sarkar, P., Iwata, T. (eds.) ASIACRYPT 2014, Part II. LNCS, vol. 8874, pp. 306–325. Springer, Heidelberg (2014)
  GF(2^{128})の乗算に対するサイドチャネル攻撃について
- [MM06] Matsui, M.: How far can we go on the x64 processors? In: Robshaw, M. (ed.) FSE 2006. LNCS, vol. 4047, pp. 341–358. Springer, Heidelberg (2006)
  AESのBitslice実装について

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 2015 Program
  http://www.cryptoexperts.com/ches2015/program.html
  プレゼンテーションスライドがダウンロード可能
- FDTC 2015 Presentation Slides
  http://conferenze.dei.polimi.it/FDTC15/slides.html
  FDTCのプレゼンテーションスライドがダウンロード可能になる予定

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