Side-channel analysis of six SHA-3 candidates in HMAC scheme

Olivier Benoît and Thomas Peyrin

CHES 2010 Workshop

Santa Barbara - August 18, 2010

ingenico beyond payment

Outline

Outline

Introduction

- NIST launched the SHA−3 competition in order to replace the collision-broken SHA−1 function
- 14 candidates are still in the race, the winner will be determined in 2012
- it makes sense to consider side-channel attack on these SHA−3 candidates in the HMAC scheme
- Retrieving the key would lead to the ability to forge correct MAC
- We will therefore analyse a panel of six candidates deemed representative

ECHO Grøstl SHAvite-3 HAMSI BLAKE CubeHash

Prior works

- DPA on n-bit sized boolean and arithmetic operations and its application to IDEA, RC6, and HMAC construction
(CHES 2005), Lemke *et al.*

- Side channel attacks against HMAC based on block-cipher based hash functions
(ACISP 2006), Okeya *et al.*

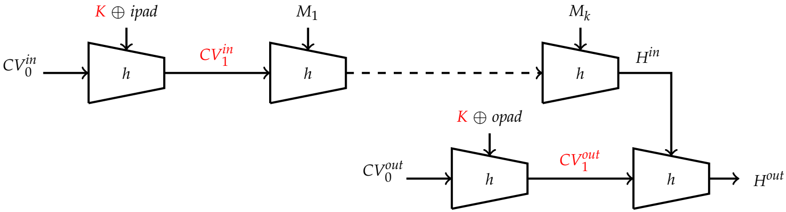
- DPA of HMAC based on SHA-2, and countermeasures
(WISA2007), McEvoy *et al.*

- An update on the side channel cyrptanalysis of MAC based on crytopgaphic hash functions
(INDOCRYPT 2007), Gauravaram *et al.*

- Practical Electromagnetic Template Attack on HMAC
(CHES 2009), Fouque *et al.*

HMAC

$$HMAC(K,M) = H((K \oplus opad)||H((K \oplus ipad)||M))$$



- The possible targets of a side-channel analysis attack are:

K, CV_1^{in} and CV_1^{out}

Outline

Correlation

- A selection function is defined as $w = f(cv, m)$
- The theoretical correlation between a data set x_i for a key guess j and the data set y_i for an arbitrary real key r is:

$$c(j,r) = \frac{\sum(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum(x_i - \overline{x})^2}.\sqrt{\sum(y_i - \overline{y})^2}}$$

- Assumming a leakage in the Hamming Weight model:

$$x_i = HW(f(j, m_i)) \text{ and } y_i = HW(f(r, m_i))$$

- Given a selection function, it is possible to compute $c(j,r)$ for all key guess and look a the correlation contrast between the real key and the wrong keys

SHA−3 Selection functions

The typical selection functions that will be found in SHA−3 candidates are:

- AES sbox ($256 \rightarrow 256$ substitution):

$$w = SBOX_{AES}(cv \oplus m)$$

- Modular addition:

$$w = (cv \boxplus m) mod 256$$

- Exclusive OR logic operation:

$$w = cv \oplus m$$

- HAMSI sbox ($16 \rightarrow 16$ substitution):

$$w = SBOX_{HAMSI}(cv_{i+1} || m_{i+1} || cv_i || m_i)$$

SHA−3 Selection functions

The typical selection functions that will be found in SHA−3 candidates are:

- AES sbox (256 → 256 substitution):

$$w = SBOX_{AES}(cv \oplus m)$$

- Modular addition:

$$w = (cv \boxplus m) mod 256$$

- Exclusive OR logic operation:

$$w = cv \oplus m$$

- HAMSI sbox (16 → 16 substitution):

$$w = SBOX_{HAMSI}(cv_{i+1}||m_{i+1}||cv_i||m_i)$$

SHA−3 Selection functions

The typical selection functions that will be found in SHA−3 candidates are:

- AES sbox (256 → 256 substitution):

$$w = SBOX_{AES}(cv \oplus m)$$

- Modular addition:

$$w = (cv \boxplus m) mod 256$$

- Exclusive OR logic operation:

$$w = cv \oplus m$$

- HAMSI sbox (16 → 16 substitution):

$$w = SBOX_{HAMSI}(cv_{i+1}||m_{i+1}||cv_i||m_i)$$

SHA−3 Selection functions

The typical selection functions that will be found in SHA−3 candidates are:

- AES sbox (256 → 256 substitution):

$$w = SBOX_{AES}(cv \oplus m)$$

- Modular addition:

$$w = (cv \boxplus m) mod 256$$

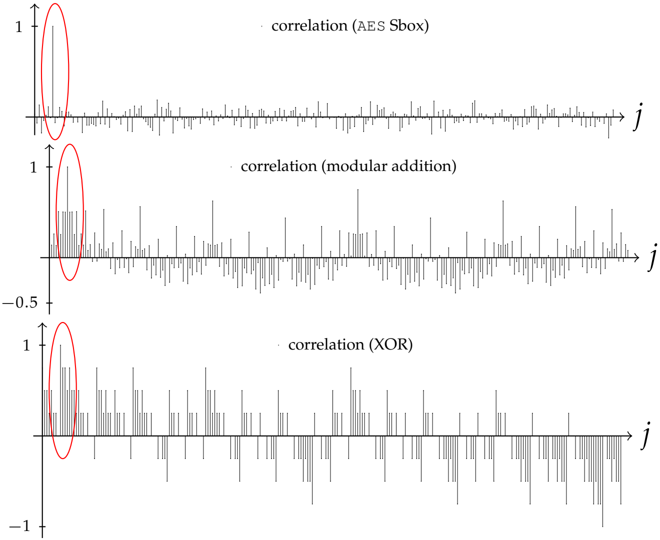
- Exclusive OR logic operation:

$$w = cv \oplus m$$

- HAMSI sbox (16 → 16 substitution):

$$w = SBOX_{HAMSI}(cv_{i+1}||m_{i+1}||cv_i||m_i)$$

Selection function efficiency, $r = 8$



Selection function efficiency

- Results for the HAMSI sbox selection function:

real and guess key	$j = 0$	$j = 1$	$j = 2$	$j = 3$
$r = 0$	$+1.00$	-0.17	-0.56	-0.87
$r = 1$	-0.17	$+1.00$	$+0.87$	-0.09
$r = 2$	-0.56	$+0.87$	$+1.00$	$+0.17$
$r = 3$	-0.87	-0.09	$+0.17$	$+1.00$

Correlation Contrast

- The correlation contrast is computed from the highest correlation for a wrong guess (c_w)

selection function	AES Sbox	*modular* addition	HAMSI Sbox	XOR
c_w	0.23	0.75	0.87	-1
c_c	3.34	0.33	0.15	0

$$c_c = \frac{1-|c_w|}{|c_w|}$$

- The selection function efficiency E is linked to the correlation contrast

$$E(\text{AES Sbox}) > E(\text{modular addition}) > E(\text{HAMSI Sbox}) > E(\text{XOR})$$

Correlation Contrast

- The correlation contrast is computed from the highest correlation for a wrong guess (c_w)

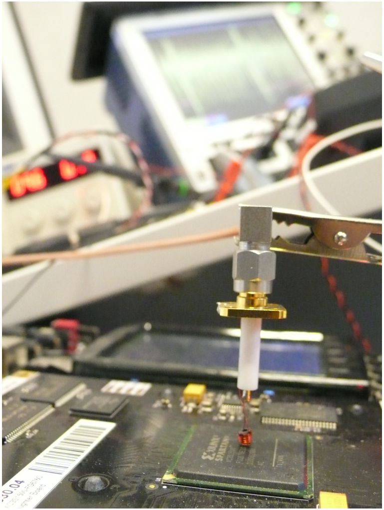
selection function	AES Sbox	*modular* addition	HAMSI Sbox	XOR
c_w	0.23	0.75	0.87	-1
c_c	3.34	0.33	0.15	0

$$c_c = \frac{1-|c_w|}{|c_w|}$$

- The selection function efficiency E is linked to the correlation contrast

$$E(\texttt{AES}\text{ Sbox}) > E(\text{modular addition}) > E(\texttt{HAMSI}\text{ Sbox}) > E(\text{XOR})$$

Measurement platform



- Xilinx Spartan FPGA
- Software selection function running on a TSK3000 RISC CPU

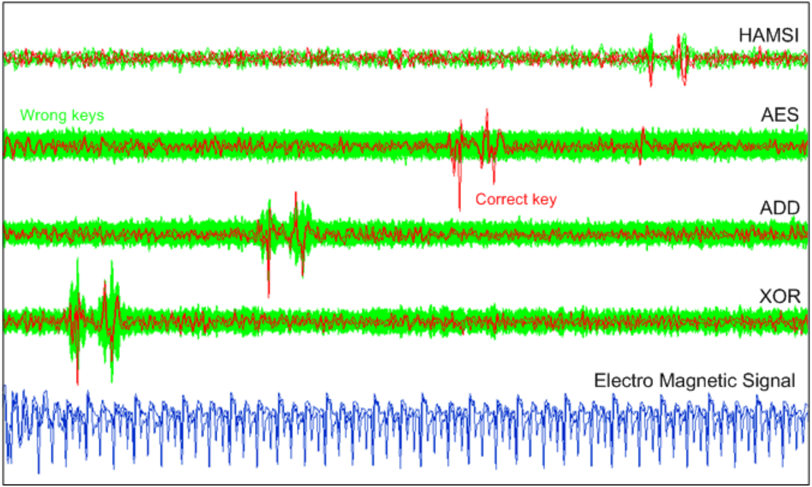
- 5 GS/s sampling frequency
- Homemade EMA sensor
- 30db Amplifier (1GHz BdW)

- 100.000 curves
- 10 curves per message

Selection functions implementation

```
// XOR sel function
for ( i=0; i<4; i++ )
{
   buffer[i] = key[i] ^ inputbuffer[i];
}
//MOD ADD sel function
for ( i=4; i<8; i++ )
{
   buffer[i] = key[i] + inputbuffer[i];
}
 // AES SBOX sel function
for ( i=8; i<12; i++ )
{
   buffer[i] = AES_SBOX[ key[i] ^ inputbuffer[i] ];
}
// HAMSI SBOX sel function
for ( i=12; i<16; i++ )
{
   temp = ((key[i] & 0x02)<<2) | ((inputbuffer[i] & 0x02)<<1) | ((key[i] & 0x01)<<1) | (inputbuffer[i] & 0x01);
   buffer[i] = HAMSI_SBOX[temp];
}
*HBIO = 0xFF;
for ( i=0; i<16; i++ )
{
   result[i] = buffer[i];
}
*HBIO = 0x00;
```

CEMA results: correlation curves for correct and wrong guess



CEMA results (5 best guess for each target byte)

```
Correlation:  XOR,   ADD,   AES    HAMSI    selection function
              0..3   4..7   8..11  12..15
Best guess selection criteria : Minimum
Previous state range: 0..0
Subkeys range:        0..255
Sample per file:      20000
Sample range:         4150..15349
Working on file ID:   0..9999
Split step: 700
Split slot size: 100
Memory requirement: 213 Mo  ,press y for cache memory y
```

Index	Rank 1 [x,cor]	Rank 2 [x,cor]	Rank 3 [x,cor]	Rank 4 [x,cor]	Rank 5 [x,cor]	Contrast
S00 :	00 [04209,-0.390]	02 [04209,-0.382]	08 [04209,-0.378]	0A [04209,-0.371]	10 [04210,-0.304]	1.9%
S01 :	0B [04910,-0.430]	03 [04909,-0.393]	09 [04909,-0.370]	4B [04910,-0.341]	01 [04908,-0.339]	9.2%
S02 :	08 [05610,-0.412]	00 [05610,-0.373]	0A [05609,-0.371]	48 [05610,-0.337]	02 [05609,-0.334]	10.5%
S03 :	0B [06309,-0.417]	09 [06310,-0.406]	03 [06309,-0.384]	01 [06309,-0.380]	4B [06309,-0.333]	2.6%
S04 :	00 [07010,-0.409]	FE [07010,-0.322]	02 [07010,-0.292]	F8 [07010,-0.287]	08 [07010,-0.287]	26.9%
S05 :	01 [07709,-0.323]	FF [07710,-0.283]	03 [07710,-0.254]	FB [07710,-0.242]	F9 [07709,-0.237]	14.0%
S06 :	02 [08409,-0.361]	04 [08410,-0.312]	00 [08410,-0.311]	FA [08410,-0.283]	FC [08411,-0.281]	15.8%
S07 :	03 [09109,-0.422]	01 [09110,-0.313]	FB [09110,-0.300]	0B [09110,-0.294]	83 [09109,-0.275]	34.8%
S08 :	00 [09810,-0.399]	9C [09810,-0.098]	26 [09810,-0.094]	28 [09808,-0.089]	33 [09810,-0.086]	304.8%
S09 :	01 [10508,-0.307]	32 [10509,-0.076]	0B [10507,-0.073]	09 [10509,-0.072]	9D [10508,-0.064]	301.6%
S10 :	02 [11209,-0.362]	9E [11209,-0.088]	08 [11210,-0.086]	31 [11209,-0.080]	24 [11212,-0.078]	313.2%
S11 :	03 [11910,-0.425]	AD [11910,-0.090]	25 [11908,-0.086]	CA [11910,-0.080]	9F [11910,-0.079]	374.7%
S12 :	00 [12609,-0.183]	01 [12626,-0.063]	02 [12640,-0.059]	03 [12550,-0.027]	188.5%	
S13 :	02 [13307,-0.326]	01 [13308,-0.266]	03 [13324,-0.093]	00 [13345,-0.006]	22.3%	
S14 :	03 [14010,-0.130]	01 [14049,-0.049]	00 [14026,-0.047]	02 [14049,-0.041]	163.0%	
S15 :	00 [14726,-0.257]	03 [14709,-0.233]	01 [14650,-0.140]	02 [14650,-0.124]	10.3%	

CEMA results versus number of curves

Files	S00	S01	S02	S03	S04	S05	S06	S07	S08	S09	S10	S11	S12	S13	S14	S15	contrast
00050	5E	73	DA	11	02	30	02	72	F6	47	91	03	00	02	01	00	12.2
00100	00	01	48	03	00	DF	02	3D	00	A3	02	03	00	02	03	00	9.5
00150	00	03	08	0B	02	FB	02	FB	00	65	02	03	00	02	03	00	17.6
00200	00	03	08	0B	00	FB	00	FB	00	65	02	03	00	02	03	00	25.5
00250	00	03	08	0B	00	FF	00	FB	00	01	02	03	00	02	03	00	32.3
00300	00	03	08	0B	00	FF	00	03	00	01	02	03	00	02	03	00	53.6
00350	00	03	08	0B	00	FF	00	03	00	01	02	03	00	02	03	00	49.9
00400	00	03	08	0B	00	01	00	03	00	01	02	03	00	02	03	00	57.6
00450	00	03	08	0B	00	01	00	03	00	01	02	03	00	02	03	00	58.7
00500	00	03	08	0B	00	01	02	03	00	01	02	03	00	02	03	00	56.0
00600	02	03	08	0B	00	01	02	03	00	01	02	03	00	02	03	03	58.1
00700	00	0B	08	0B	00	01	02	03	00	01	02	03	00	02	03	00	64.4
00800	08	0B	08	0B	00	01	02	03	00	01	02	03	00	02	00	00	66.1
00900	0A	0B	08	0B	00	01	02	03	00	01	02	03	00	02	00	00	62.4
01000	08	0B	08	0B	00	01	02	03	00	01	02	03	00	02	00	00	76.0
02000	00	0B	08	0B	00	01	02	03	00	01	02	03	00	02	03	00	88.4
03000	00	0B	08	0B	00	01	02	03	00	01	02	03	00	02	03	00	114.2
04000	00	0B	08	0B	00	01	02	03	00	01	02	03	00	02	03	00	120.1
05000	08	0B	08	0B	00	01	02	03	00	01	02	03	00	02	03	00	127.6
06000	08	0B	08	0B	00	01	02	03	00	01	02	03	00	02	03	00	123.9
07000	00	0B	08	0B	00	01	02	03	00	01	02	03	00	02	03	00	129.6
08000	00	0B	08	0B	00	01	02	03	00	01	02	03	00	02	03	00	126.8
09000	00	0B	08	0B	00	01	02	03	00	01	02	03	00	02	03	00	136.4
10000	00	0B	08	0B	00	01	02	03	00	01	02	03	00	02	03	00	146.7

Outline

ECHO side channel analysis

- Internal state at the end of the first round:

$$w_{i_0}[b] = \alpha \cdot cv'_{i_1}[b] \oplus \beta \cdot m'_{i_2}[b] \oplus \gamma \cdot m'_{i_3}[b] \oplus \delta \cdot m'_{i_4}[b]$$

- Internal state in second round, after AES Sbox operation:

$$w'_i[b] = Sbox(w_i[b] \oplus t_i[b])$$

- 64 AES Sbox side-channel attacks to retrieve CV
- For each cv'_i, four selection functions can be exploits

Grøstl side channel analysis

- Internal state after the AES Sbox operation during first round of P_G

$$w'[b] = Sbox(m[b] \oplus CV[b])$$

- In this case, CPA is straightforward
- 64 AES Sbox side-channel attacks to retrieve CV
- It is possible to speed up the attack by a factor 64 by choosing all $m[b]$ equals

SHAvite-3 side channel analysis

- Internal state after the AES Sbox operation during first round of E^S

$$w'[b] = Sbox(CV^R[b] \oplus m_0^1[b])$$

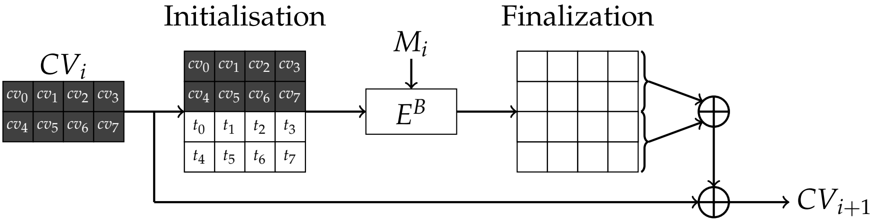
- Internal state after the AES Sbox operation during second round of E^S

$$z'[b] = Sbox(CV^L[b] \oplus w''[b] \oplus m_0^2[b])$$

- 32 AES Sbox side-channel attacks to retrieve CV
- In order to retrieve CV^L, the right part CV^R must be found without errors

BLAKE description

- Overview: $CV_{i+1} = final(E^B_{M_i}(init(CV_i)), CV_i)$
- E^B is a block cipher composed of 10 rounds, each consisting of the application of eight 128-bit sub-functions G_i



BLAKE description

- One round of E^B computes:

$$G_0(v_0, v_4, v_8, v_{12}) \quad G_1(v_1, v_5, v_9, v_{13}) \quad G_2(v_2, v_6, v_{10}, v_{14}) \quad G_3(v_3, v_7, v_{11}, v_{15})$$
$$G_4(v_0, v_5, v_{10}, v_{15}) \quad G_5(v_1, v_6, v_{11}, v_{12}) \quad G_6(v_2, v_7, v_8, v_{13}) \quad G_7(v_3, v_4, v_9, v_{14})$$

- The function $G_s(a, b, c, d)$ processes the following steps:

$$\begin{aligned}
a &\leftarrow (a \boxplus b) \boxplus (m_i \oplus k_j) \\
d &\leftarrow (d \oplus a) \ggg 16 \\
c &\leftarrow (c \boxplus d) \\
d &\leftarrow (b \oplus c) \ggg 12 \\
a &\leftarrow (a \boxplus b) \boxplus (m_j \oplus k_i) \\
d &\leftarrow (d \oplus a) \ggg 8 \\
c &\leftarrow (c \boxplus d) \\
d &\leftarrow (b \oplus c) \ggg 7
\end{aligned}$$

BLAKE side channel analysis

- the first four execution of G_s manipulates the secret chaining variable:

$$G_0(cv_0, cv_4, t_0, t_4) \qquad G_1(cv_1, cv_5, t_1, t_5)$$
$$G_2(cv_2, cv_6, t_2, t_6) \qquad G_3(cv_3, cv_7, t_3, t_7)$$

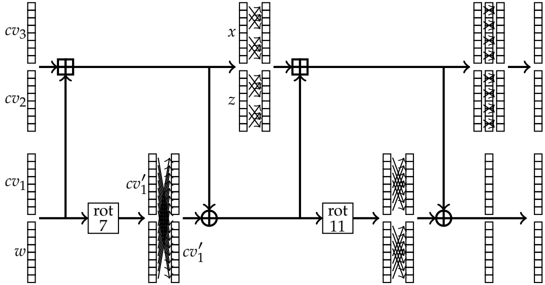
- The function $G_s(a, b, c, d)$ processes the following steps:

$$\begin{aligned} a_1 &= (a_0 \boxplus b_0) \boxplus m_k \\ d_1 &= (d_0 \oplus a_1) \ggg 16 \\ c_1 &= c_0 \boxplus d_1 \\ b_1 &= (b_0 \oplus c_1) \ggg 12 \\ a_2 &= a_1 \boxplus b_1 \boxplus m_l \end{aligned}$$

- The two selection functions are based on the Modular Addition operation

`CubeHash` side channel analysis

- Overview: $CV_{i+1} = P_C(CV_i \oplus (M_i||\{0\}^{768}))$



- Two selection functions based on the XOR operation
- Two selection functions based on the Modular Addition operation

HAMSI side channel analysis

- Generic selection function:

$$w = Sbox(m'_i[b] \mid\mid cv'_{i+2}[b] \mid\mid m'_{i+4}[b] \mid\mid cv'_{i+6}[b])$$

or

$$w = Sbox(cv'_i[b] \mid\mid m'_{i+2}[b] \mid\mid cv'_{i+4}[b] \mid\mid m'_{i+6}[b])$$

- Two bits of *CV* recovered at a time with a total of 128 HAMSI Sbox side-channel attacks (4 guess each)
- Could be enhanced by selecting multiple sbox at the same time, but must be coherant with implementation

Outline

Results summary

Candidates	Selection function	Correlation analysis
ECHO	$SBOX_{AES}$	64 analysis at byte level (x4 possibilities)
Grøstl	$SBOX_{AES}$	64 analysis at byte level
SHAvite-3	$SBOX_{AES}$	16 + 16 analysis at byte level
BLAKE	Modular addition	32 analysis at byte level
CubeHash	Modular addition and XOR	64 ADD + 64 XOR analysis at byte level
HAMSI	$SBOX_{HAMSI}$	128 analysis at 2-bit level

Conclusion

- AES-based candidates (ECHO SHAvite-3 and Grøstl)
  - Provide the same vulnerability to SCA as the AES block cipher
  - Can take advantage of protection inherited from hardware AES
- ARX candidates (BLAKE and CubeHash)
  - SCA will be less efficient (especially for CubeHash and its XOR selection function)
  - Less efficient to protect: require to constantly switch from arithmetic to boolean masking
- HAMSI candidate is quite exotic, a deeper study will be required if this candidate is choosen at the end of the SHA-3 contest

beyond payment

Conclusion

- AES-based candidates (ECHO SHAvite-3 and Grøstl)
  - Provide the same vulnerability to SCA as the AES block cipher
  - Can take advantage of protection inherited from hardware AES
- ARX candidates (BLAKE and CubeHash)
  - SCA will be less efficient (especially for CubeHash and its XOR selection function)
  - Less efficient to protect: require to constantly switch from arithmetic to boolean masking
- HAMSI candidate is quite exotic, a deeper study will be required if this candidate is choosen at the end of the SHA-3 contest

ingenico beyond payment

Conclusion

- AES-based candidates (ECHO SHAvite-3 and Grøstl)
  - Provide the same vulnerability to SCA as the AES block cipher
  - Can take advantage of protection inherited from hardware AES
- ARX candidates (BLAKE and CubeHash)
  - SCA will be less efficient (especially for CubeHash and its XOR selection function)
  - Less efficient to protect: require to constantly switch from arithmetic to boolean masking
- HAMSI candidate is quite exotic, a deeper study will be required if this candidate is choosen at the end of the SHA-3 contest

ingenico
beyond payment

Thank you for your attention

Any questions?

olivier.benoit@ingenico.com
thomas.peyrin@ingenico.com

ingenico beyond payment