Cryptanalysis of short variants of SHA-256-XOR

Krystian Matusiewicz

supervised by: Josef Pieprzyk, Huaxiong Wang

kmatus@ics.mq.edu.au

Centre For Advanced Computing, Algorithms and Cryptography,

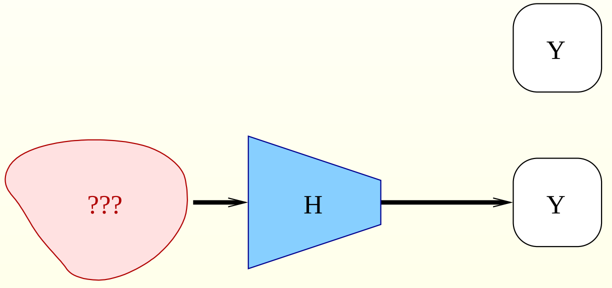
Department of Computing,

Macquarie University

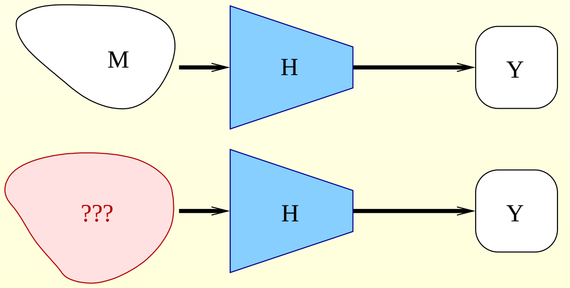
Properties of cryptographic hash functions

hash function : $H : \{0,1\}^* \rightarrow \{0,1\}^n$

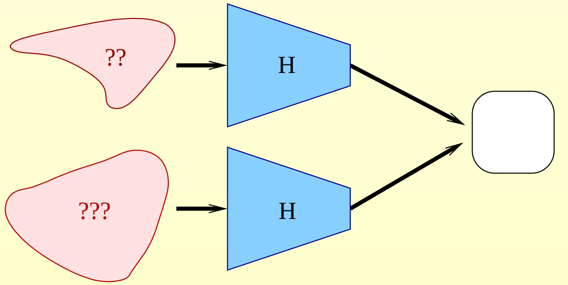
Preimage resistant : Given an output Y of the hash function it is difficult to find any *preimage* - an input X such that $h(X) = Y$.



Second preimage resistant : Given a fixed input X to the hash function and the corresponding output $h(X)$ it is difficult to find a *second preimage* - another input X', $X' \neq X$ such that $h(X) = h(X')$.



Collision resistant : It is hard to find any pair of distinct messages (X, X'), $X \neq X'$ such that $h(X) = h(X')$.



Applications

- digital signatures

```
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA1

This is a sample message to be signed. PG Miniconference is great!!!
-----BEGIN PGP SIGNATURE-----
Version: GnuPG v1.4.3 (GNU/Linux)

iD8DBQFEjjXm32L5tWc+oHoRAomJAKCpgQnBrMw/sTLqZ99Qt00eejkm6gCgrxve
mXkmY//9J7Nspav5nB+r2wU=
=7Z0s
-----END PGP SIGNATURE-----
```

- password-based user identification

```
gdm:!!:13245:0:99999:7:::
kmatus:$1$5IwJJ/.O$9Em5P./CiGE48TVO2QWbz/:13245:0:99999:7:::
bogo:$1$MPW3Z.Au$SJlzrNZUA1qBa8nkUF6Ki.:13245:0:99999:7:::
```

- data integrity

The CD ISO image names are listed below. After downloading, you can verify the file against the
file is not corrupted. A full installation will require all of the discs for the desired architecture.

- For x86-compatible (32-bit):

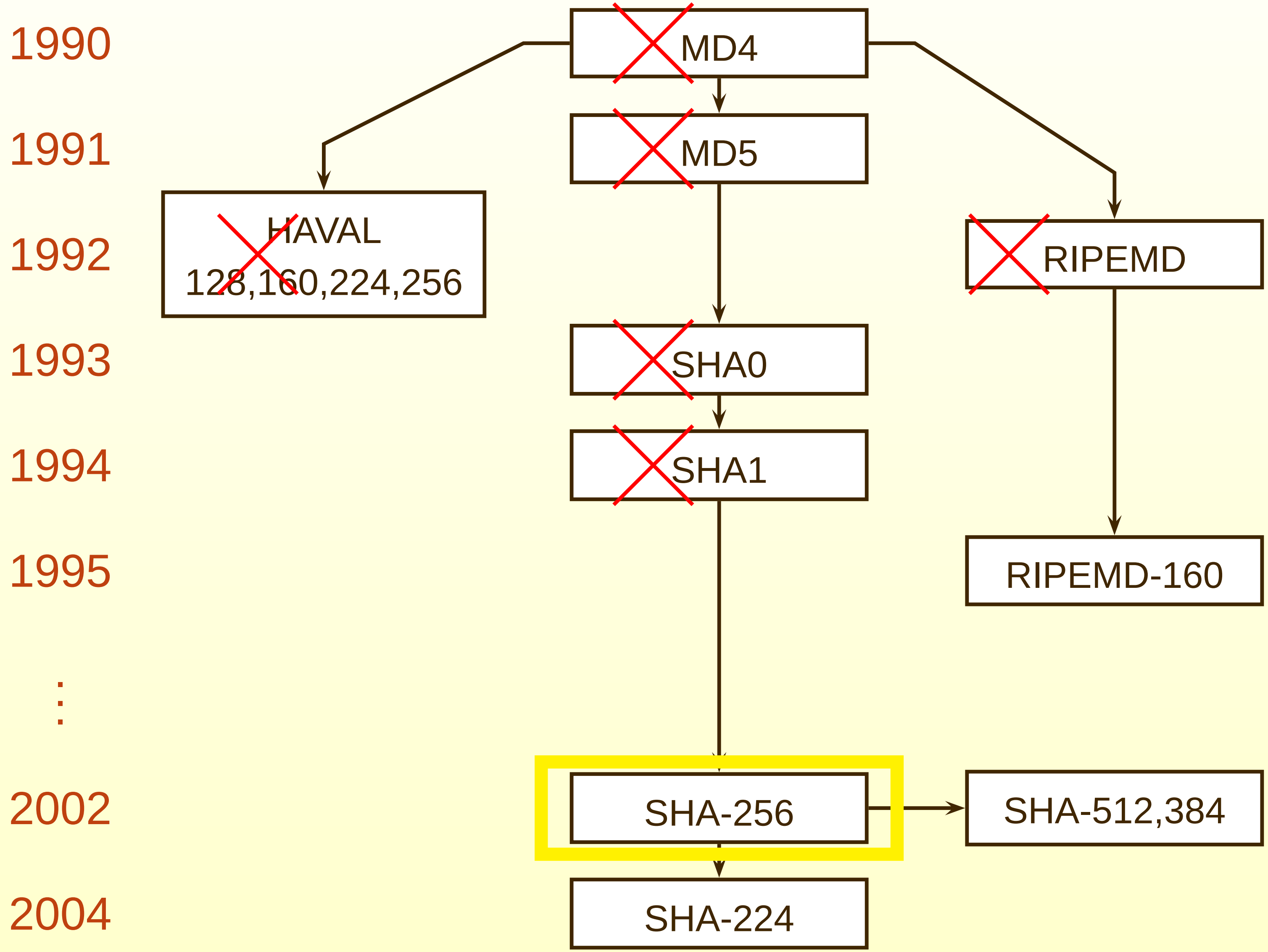
  FC5-i386-disc1.iso (sha1sum: 43546c0e0d1fc64b6b80fe1fa99fb6509af5c0a0)

  FC5-i386-disc2.iso (sha1sum: a85ed1ca5b63e2803f29a33ea6a6bc8eb7f63122)

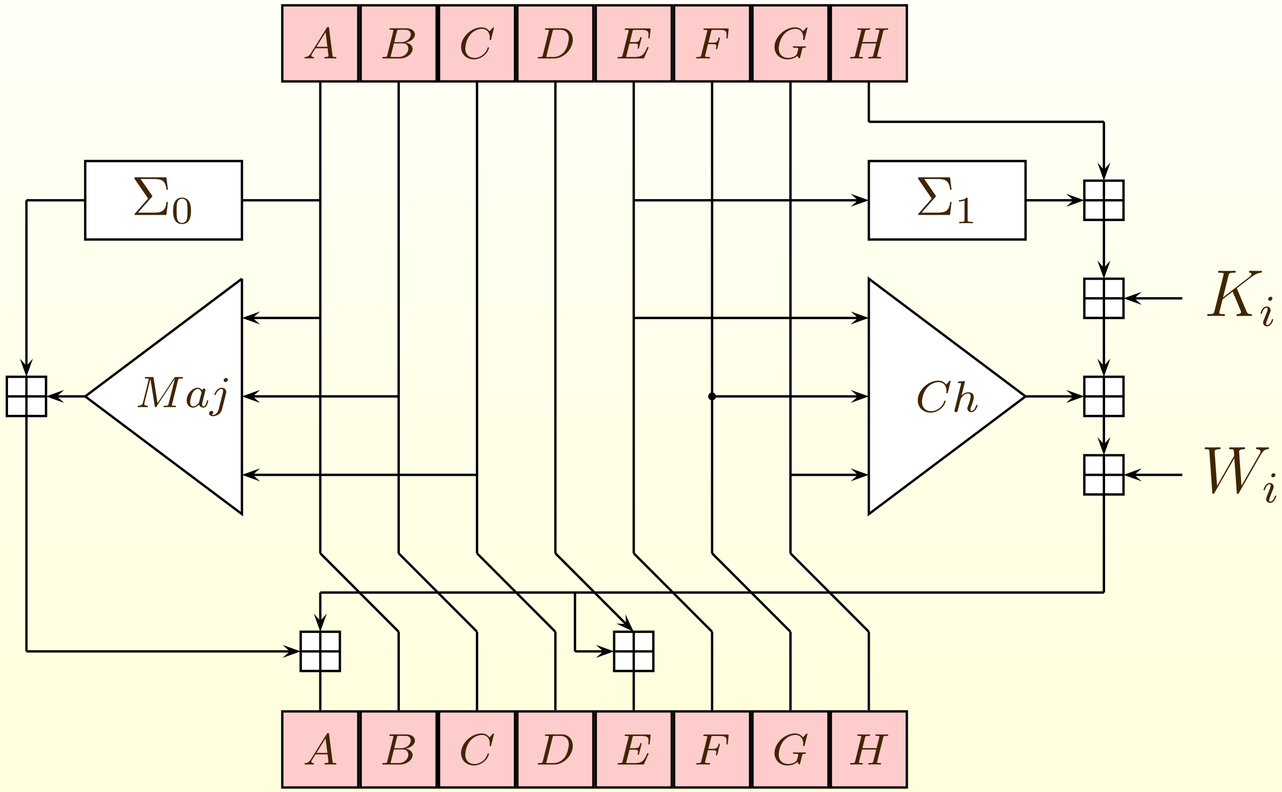
- many others...

Attacks on dedicated cryptographic hash functions

Attack – showing how to find two colliding messages



SHA-256

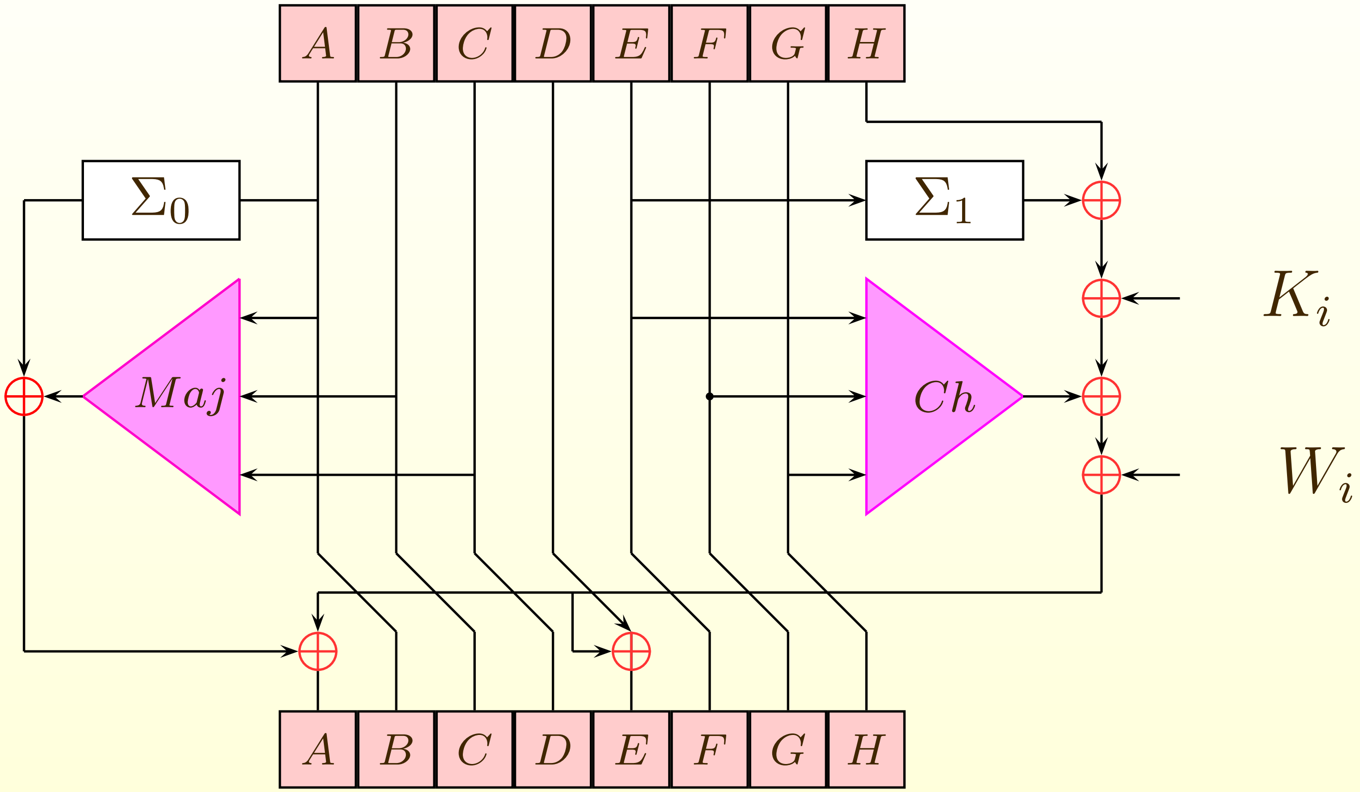


$$\Sigma_0(A) = A^{\lll 2} \oplus A^{\lll 13} \oplus A^{\lll 22} \qquad \Sigma_1(E) = E^{\lll 6} \oplus E^{\lll 11} \oplus E^{\lll 25}$$

$$MAJ(A, B, C) = (A \wedge B) \oplus (B \wedge C) \oplus (A \wedge C)$$

$$IF(E, F, G) = (E \wedge F) \oplus (\neg E \wedge G)$$

SHA-256-XOR



If the function were linear, it would be trivial to find collisions.

Linear approximations of Boolean functions

$$\Delta x = x \oplus x', \ldots, \qquad \Delta\,\mathrm{IF} = IF(x,y,z) \oplus IF(x',y',z')$$

Δx	Δy	Δz	$\Delta\,\mathrm{IF}$	$\Delta\,\mathrm{MAJ}$	$\Delta\,\mathrm{IF} = \Delta y$
0	0	0	0	0	-
0	0	1	$x \oplus 1$	$x \oplus y$	$x = 1$
0	1	0	x	$x \oplus z$	$x = 1$
0	1	1	1	$y \oplus z \oplus 1$	-
1	0	0	$y \oplus z$	$y \oplus z$	$y + z = 0$
1	0	1	$x \oplus y \oplus z$	$x \oplus z \oplus 1$	$x \oplus y \oplus z = 0$
1	1	0	$x \oplus y \oplus z \oplus 1$	$x \oplus y \oplus 1$	$x \oplus y \oplus z = 0$
1	1	1	$y \oplus z \oplus 1$	1	$y + z = 0$

The outline of the attack

- choose linear approximations of MAJ and IF and construct a $\mathbb{F}_2$–linear model of SHA-256-XOR,
- find a suitable collision-producing difference for the linearized SHA-256-XOR,
- derive a set of conditions under which the real SHA-256-XOR behaves like the linear model with respect to difference propagation
- find a message for which all the conditions (approximating equations) are satisfied.

The outline of the attack : step 1

- choose linear approximations of MAJ and IF and construct a $\mathbb{F}_2$–linear model of SHA-256-XOR,

There are better and worse approximations – which are the best ones?

The choice influences

- the density of differentials,
- the probability that the system of approximating conditions is consistent

The outline of the attack : step 2

- find a suitable collision-producing difference for the linearized SHA-256-XOR,

Once the hash function is linearised it can be seen as a function

$$SXL : \mathbb{F}_2^{256} \times \mathbb{F}_2^{512} \longrightarrow \mathbb{F}_2^{256}$$

linear over $\mathbb{F}_2$. Now, every bit string $(\Delta_{IV}, \Delta_M) \in \mathbb{F}_2^{256} \times \mathbb{F}_2^{512}$ such that

$$SXL(\Delta_{IV}, \Delta_M) = 0$$

is a pseudo-collision-producing difference for the linearised version.

The outline of the attack : step 2

- find a suitable collision-producing difference for the linearized SHA-256-XOR,

Out of all possible differentials $(\Delta_{IV}, \Delta_M) \in \mathrm{Ker}(SXL)$ we want to choose those that generate the smallest amount of differences in registers.

- Each register is a linear function of (Δ_{IV}, Δ_M)
- The state of all registers can be represented as a matrix with rows

$$[A_0|E_0|A_1|E_1|\ldots|A_n|E_n]$$

- look for combinations of rows with small weights - finding low weight codewords in linear codes

The outline of the attack : step 3

- derive a set of conditions under which the real SHA-256-XOR behaves like the linear model with respect to difference propagation

For each non-zero input difference to a Boolean function we have one equation on the values of inputs to that function.

Collect them all and reduce. Hopefully, the system is consistent.

$$
\begin{aligned}
A_{3,2} &= A_{2,2} \\
A_{3,2} &= A_{2,2} \\
E_{3,2} &= E_{2,3} + E_{1,2} \\
E_{3,20} &= 1 \\
A_{3,20} &= A_{2,20}
\end{aligned}
\qquad
\begin{aligned}
E_{3,21} &= 1 \\
A_{3,22} &= A_{2,22} \\
E_{3,22} &= E_{2,23} + E_{1,22} \\
A_{3,23} &= A_{1,23} + 1 \\
E_{3,23} &= E_{2,23}
\end{aligned}
$$

The outline of the attack : step 4

- find a message for which all the conditions (approximating equations) are satisfied.

Adjust the values of registers A and E by flipping some bits of the message.

$$\Delta A_{i,b} = \Delta T1_{i,b} \oplus \Delta T2_{i,b}$$

$$\Delta E_{i,b} = \Delta A_{i-4,b} \oplus \Delta T2_{i,b}$$

$$\begin{aligned} T1_{i,b} =& \mathcal{L}_{MAJ}(\Delta A_{i-1,b}, \Delta A_{i-2,b}, \Delta A_{i-3,b}) \oplus \\ & \Delta A_{i-1,(b+2) \bmod 32} \oplus \Delta A_{i-1,(b+13) \bmod 32} \oplus \Delta A_{i-1,(b+22) \bmod 32} \\ T2_{i,b} =& \mathcal{L}_{IF}(\Delta E_{i-1,b}, \Delta E_{i-2,b}, \Delta E_{i-3,b}) \oplus \\ & \Delta E_{i-1,(b+6) \bmod 32} \oplus \Delta E_{i-1,(b+11) \bmod 32} \oplus \Delta E_{i-1,(b+25) \bmod 32} \oplus \\ & \Delta E_{i-4,b} \oplus \Delta W_{i-1,b} \end{aligned}$$

Example

step	A	E
0	00000000	00000000
1	25008048	25008048
2	00813d2a	098115a8
3	08008400	4084e709
4	00000000	20915c0e
5	08000000	25000448
6	00000000	02817d0a
7	00000000	00008400
8	00000000	00000000
9	00000000	08000000
10	00000000	00000000
11	00000000	00000000
12	00000000	00000000
13	00000000	00000000
14	00000000	00000000
15	00000000	00000000
16	00000000	00000000
17	00000000	00000000

$N = 18$ steps

$\mathrm{IF} \approx z$, $\mathrm{MAJ} \approx y$.

total weight of the differential: 312

number of equations: 172

Discussion

Strengths:

- the idea works for all similarly designed hash functions
- nice mathematical model

Problems:

- Finding good differentials is hard
- for SHA-256-XOR we can attack variants with with 20-22 steps
- Any better algoritms for finding messages satisfying conditions?

Related work:

F.Mendel, N.Pramstaller, C.Rechberger, and V.Rijmen, **Analysis of Step-Reduced SHA-256**, Proc. FSE'2006, Gratz, Austria