How (not) to efficiently dither blockcipher-based hash functions?

Jean-Philippe Aumasson, Raphael C.-W. Phan

FHNW, Switzerland
Loughborough University, UK

CONTENT OF THE TALK

Dithered hashing

Blockcipher-based hashing

Case study: $E_H(M) \oplus M$

DITHERED HASHING

HASH FUNCTIONS

Contract an object of arbitrary size to an object of fixed, small size

- inversion impossible
- collisions exist
- loss of information

$h : \{0,1\}^\star \mapsto \{0,1\}^n$

Collision: pair $x \neq y$ such that $h(x) = h(y)$

Designed so that it is effectively impossible to

- find a **collision**
- find a **preimage** of a random image

and that the output is “random-looking”

ITERATED HASH

From a finite-domain **compression** function f

To hash $M_1 \| M_2 \| M_3$, compute

$$\begin{aligned} H_1 &\leftarrow f(IV, M_1) \\ H_2 &\leftarrow f(H_1, M_2) \\ H_3 &\leftarrow f(H_2, M_3) \end{aligned}$$

and return the **hash value** $H_3 = h(M_1 \| M_2 \| M_3)$

Call H_1, H_2 **chain values**, IV the **initial value**

FIXED-POINTS

Pair (H, M) such that $f(H, M) = H$

Length extension attack, if $IV = H$:

$$h(M) = h(M\|M\|M) = h(M\|\ldots\|M) = H$$

$$H \xrightarrow{M} H \xrightarrow{M} \cdots \xrightarrow{M} H$$

Message length **padding** avoids these collisions

FIXED-POINTS

"not very dangerous" [Preneel et al.]

"not really worth worrying about" [Schneier]

"of concern if it can be arranged that the chaining variable has a value for which a fixed point is known" [HAC]

but fixed-points exploited for shortcut second-preimage attacks
[Dean, Kelsey/Schneier]

AGAINST FIXED-POINT ATTACKS: "DITHERING"

Redefine the compression function to

$$f(H_{i-1}, M_i, D_i) = H_i$$

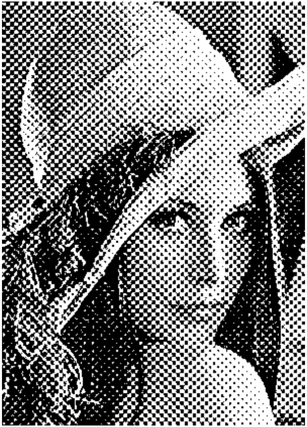
Call D_i the **dither value** (fixed, public)

Example of dither values: **counter**

$$H \xrightarrow{D_1=1} H_1 \xrightarrow{D_2=2} \dots \xrightarrow{D_n=n} H_n$$

Goal: simulate different round functions

“DITHERING” IN IMAGE PROCESSING



ETHYMOLOGY

326 PROVINCIALISMS.

DIKE; a ditch; alſo a puddle, or ſmall pool of water.

To DILL; to ſoothe, blunt or ſilence pain or ſound.

To DITHER (the *i ſhort*, as in wither); to tremble or ſhiver with cold.

To DOCK; to trim the buttocks, &c. of ſheep.

DOCKEN; *rumex*; dock.

DOGFINKIL; *anthemis cotula*; maithe-weed.

D I

or in one or ſeveral Sciences, explaining the Signification of them.

Dictitate, to Speak or Tell often.

Didactick, Inſtructive.

Didder, to Shiver or Shake with Cold.

Diennial, of, or belonging to two Years.

Dieſis, a Term in Muſick.

BLOCKCIPHER-BASED HASHING

BLOCKCIPHER-BASED HASHING

Blockcipher = permutation family

$$E_k : \{0,1\}^n \mapsto \{0,1\}^n$$

Every key selects a permutation

Motivations

- Trust
- Compact implementation

Obstacles

- Slow (key schedule)
- Structural problems (short blocks)

BLOCKCIPHER-BASED HASHING

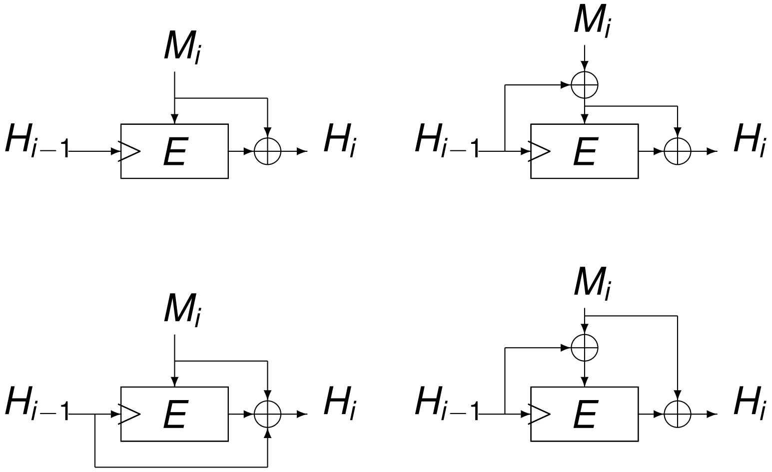
Many proposals in the 80's

12 schemes conjectured secure in 1993
[Preneel et al.] (the **PGV** schemes)...

...proven secure in 2002
[Black/Rogaway/Shrimpton]

Active research topic
(see ICALP, EUROCRYPT, CRYPTO 2008)

EXAMPLES OF PGV SCHEMES



PROBLEM

Build **dithered hashing from blockciphers**

Generic methods

- are unefficient (increase hashed message length)
- have no security proof

OUR RESULTS

56 dithered blockcipher-based constructions

Proofs in the **ideal cipher** model that

- 12 give secure hash functions
- 37 have easy-to-find fixed points

for any dither sequence

8 schemes can give weak hash functions
despite strong compression functions

CASE STUDY

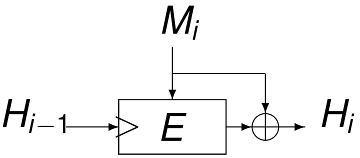
CONSTRUCTION OF DITHERED SCHEMES

Starting from a blockcipher-based scheme the dither value D can be XORed with

- the message M
- the chain value H_{i-1}
- the output H_i
- the blockcipher E's key
- the blockcipher E's input

$\Rightarrow$ 5 (nondisjoint) classes of dithered schemes

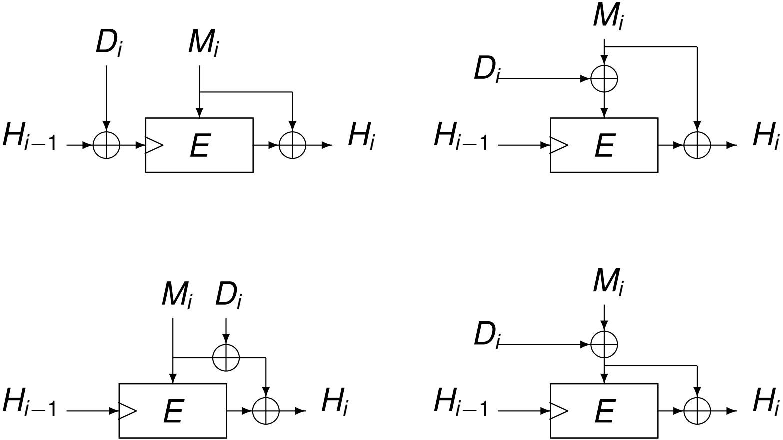
THE MMO SCHEME



Chain value used as a key:

$$H_i = E_{H_{i-1}}(M_i) \oplus M_i$$

DITHERED MMO SCHEMES



OBSERVATIONS

1. "$D_i \oplus H_{i-1}$" equivalent to "$D_i \oplus$key"

$$E_{H_{i-1} \oplus D_i}(M_i) \oplus M_i$$

2. "$D_i \oplus M_i$" and "$D_i \oplus H_i$" **equivalent** up to renaming

$$E_{H_{i-1}}(M_i \oplus D_i) \oplus M_i = E_{H_{i-1}}(M_i') \oplus (M_i' \oplus D_i)$$

Simplifies proofs for collision/preimage resistance…

PROOF IDEA: SIMPLE CASE

Reduction: construct collisions for the basic (undithered) scheme from the dithered variants

$\Rightarrow$ Tight bounds, dither-independent

Example: given a preimage for the dithered scheme

$$E_{H_{i-1} \oplus D_i}(M_i) \oplus M_i$$

one finds a preimage for the original scheme

$$E_{H'_{i-1}}(M_i) \oplus M_i$$

PROOF IDEA: NONEASY CASE

1: **Synthesis**

= reduce several schemes to one generic construction

2: **Simulation-based proof**

= bound success probability as a function of the number of queries to the blockcipher

EXAMPLE

For the schemes

$$\begin{aligned} H_i &= E_{H_{i-1}}(M_i \oplus D_i) \oplus M_i \\ H_i &= E_{H_{i-1}}(M_i) \oplus M_i \oplus D_i \end{aligned}$$

finding a **collision** (distinct salts) equivalent to finding (A, B, C, D) such that

$$A \oplus B \oplus E_C(A) \oplus E_D(B)$$

is in a specific ensemble

CONCLUSIONS

HOW TO DITHER…

$D_i \oplus H_{i-1}$

- preserves collision- and preimage-resistance
- security **independent** of the dither values

$D_i \oplus M_i$

- preserves preimage-resistance
- not collision-resistant in very special cases

Recommendation: **counter**: $D_1 = 1, \ldots, D_n = n$

NEED FOR DITHERING?

Arguments for dithering

- prevents from generic shortcut attacks
- safety net against dedicated attacks

Arguments against

- prevents from attacks far above the $2^{n/2}$ barrier
- complicates description, implementation, analysis

QUESTIONS (EXAMPLES)

"Are there concrete applications?"

Yes, e.g. MD5, SHA-1 (construction $E_M(H) \oplus H$)

"What do say your security proofs in practice?"

If something goes wrong, it comes from a flaw of the compression function, not of the construction

"Should I use dithering in my SHA-3 proposal?"

Yes if doesnt slow down hashing