

Generalizing the Herding
Attacks to Concatenated
Hashing Schemes

Orr Dunkelman & Bart Preneel

Dept. of Electrical Engineering

ESAT/SCD-COSIC

Katholieke Universiteit Leuven

Belgium

Outline

1. Joux's Multicollision Attacks
2. Joux's Attacks on Concatenated Hashing Schemes
3. The Herding Attack
4. Herding Attack on Concatenated Hashing Schemes

Multicollisions in Iterated Hashing

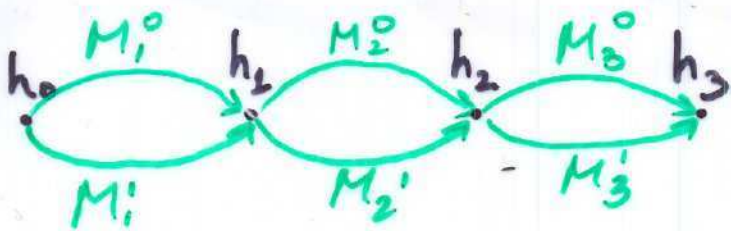
$$\begin{aligned} h_0 &= IV & M &= M_1 \dots M_t \\ h_{i+1} &= C(h_i, M_{i+1}) \\ h(M) &= h_t \end{aligned}$$

iterated hashing
(Merkle-Damgård)

collision - $O(2^{n/2})$ $n = |h_t|$

k-collision - $O(2^{n \cdot k / (k+1)})$

[J04]:



$$h(M_1^{e_1} M_2^{e_2} M_3^{e_3}) = h_3 \text{ independent of } e_i\text{'s}$$

⇓

k-collision - $O(\lceil \log_2 k \rceil \cdot 2^{n/2})$
(in iterated hashing)

Attacks on Concatenated Hashing

$$h(M) = h_1(M) || h_2(M) || \dots$$

should be more secure than $h_i(\cdot)$.

[J04]: Yes.

security gain exponential

$$\text{collision: } O(2^{(n_1+n_2)/2})$$

$$\text{pre-image: } O(2^{n_1+n_2})$$

[J04]: No when $h_i(\cdot)$ iterated.

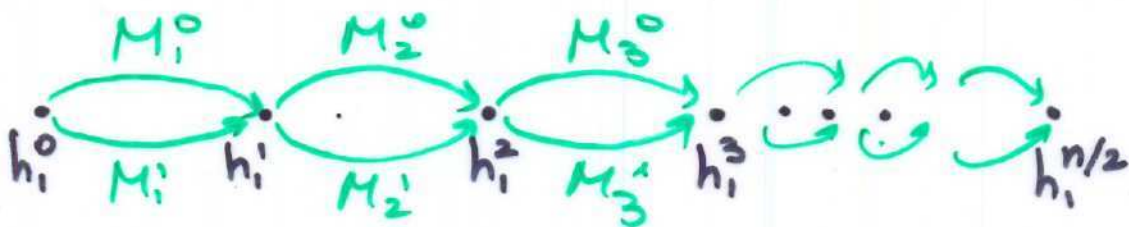
$$\text{collisions: } O(n_2 \cdot (2^{n_1/2} + 2^{n_2/2}))$$

$$\text{pre-image: } O(2^{n_1} + n_2 \cdot 2^{n_2})$$

Collision for $h(x) = h_1(x) || h_2(x)$ [J04]

Assume $|h_1(x)| = |h_2(x)| = n$

Let $h_1^0 = IV_1$. Find $2^{n/2}$ multi collision in $h_1(\cdot)$



$$h_1(M_1^{e_1} M_2^{e_2} \dots M_{n/2}^{e_{n/2}}) = h_1^{n/2}$$

for $2^{n/2}$ choices of e_i 's

By birthday paradox:

$$\exists \{e_i\}, \{e_j\} \quad h_2(M^{e_i}) = h_2(M^{e_j})$$

$$\Downarrow$$
$$h(M^{e_i}) = h_1(M^{e_i}) || h_2(M^{e_i})$$

$$h_1(M^{e_j}) || h_2(M^{e_j}) = h(M^{e_j})$$

Preimage for $h(x) = h_1(x) || h_2(x)$

Let target = (h^1, h^2)

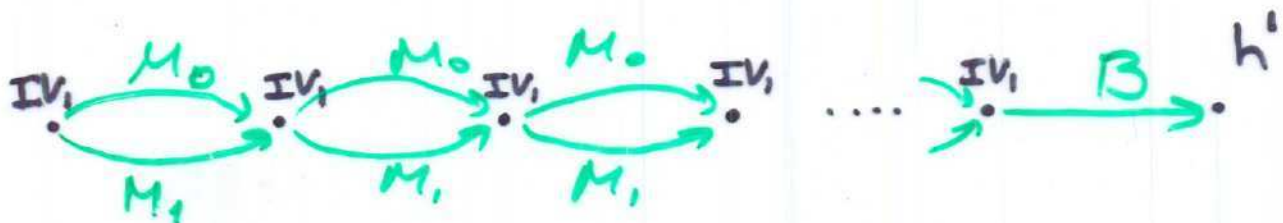
Find M_0, M_1 s.t.

$$C_1(IV_2, M_0) = IV_2$$

$$C_1(IV_1, M_1) = IV_2$$

Find B s.t.

$$C_1(IV_1, B) = h^1$$



Can generate 2^n messages $\{M_{e_i} || B\}$

s.t. $h_1(\{M_{e_i} || B\}) = h^1$.

if $h_2(\cdot)$ is "random enough",

$\exists \{e_i\}$ s.t. $h_2(\{M_{e_i}\} || B) = h^2$

The Herding Attack [kk05]

Commit to a digest value h_t .

Given a prefix p find suffix s s.t.

$$h(p||s) = h_t$$

Time complexity expected to be 2^n

time-memory trade off for iterated

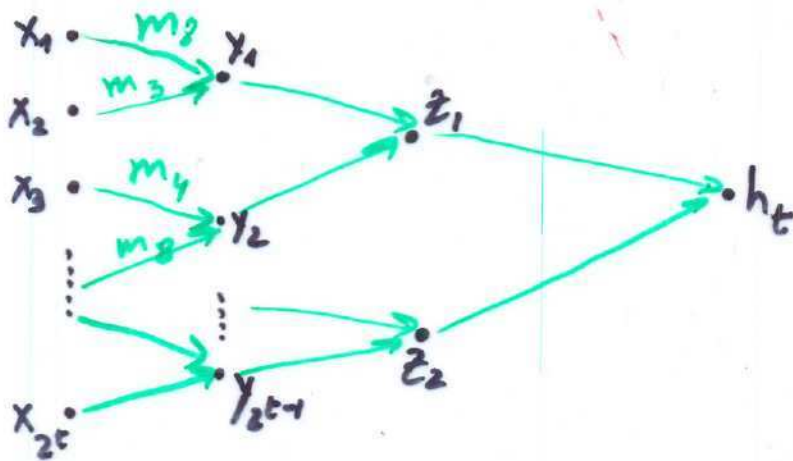
hashing:

$x_1 \circ$
 $x_2 \circ$
 $x_3 \circ$
⋮
 $x_{2^t} \circ$

1. pick 2^t values.
2. pick $2^{(n-t)/2}$ messages m_i
3. compute all $C(x_i, m_i)$

4. get collisions!

The Herding Attack (cont.)



5. Repeat

6. Publish h_t
Store the structure
(diamond str.)

Online phase:

given p , try m 's until

$$h(p||m) = X_i$$

for some i .

set $s = m||$

the message blocks
that lead to h_t

Time complexity: Pre-processing - $O(2^{(n+t)/2})$

On-line - $O(2^{n-t})$

Herding Attacks on Concatenated Hash.

1st attempt - standard herding:

2^t starting points out of $2^{n_1+n_2}$
Off line $2^{(n_1+n_2+t)/2}$
On line $2^{(n_1+n_2)-t}$

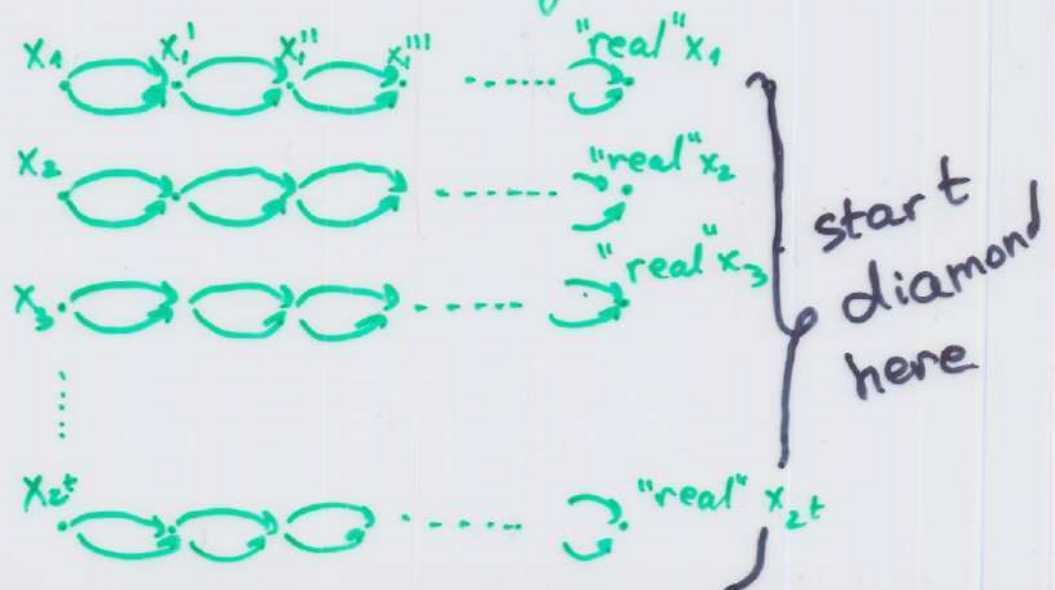
(here and after $n_1=n_2=n$)

2nd attempt - build diamond for $h_1(\cdot)$

found a path for $h_1(\cdot)$

doesn't solve prob for $h_2(\cdot)$

"solution" - have 2^n paths
for $h_1(\cdot)$ for each
starting value



2^t starting points for $h_1(\cdot)$
(out of 2^n)

Off line $\underbrace{n \cdot 2^t \cdot 2^{n/2}}_{\text{starting}} + \underbrace{2^{(n+t)/2}}_{\text{real diamond}}$

On line $\underbrace{2^{n-t}}_{\text{for } h_1(\cdot)} + \underbrace{n \cdot 2^n}_{\text{for } h_2(\cdot)}$

Better, but still requires 2^n

The New Attack

main observation:

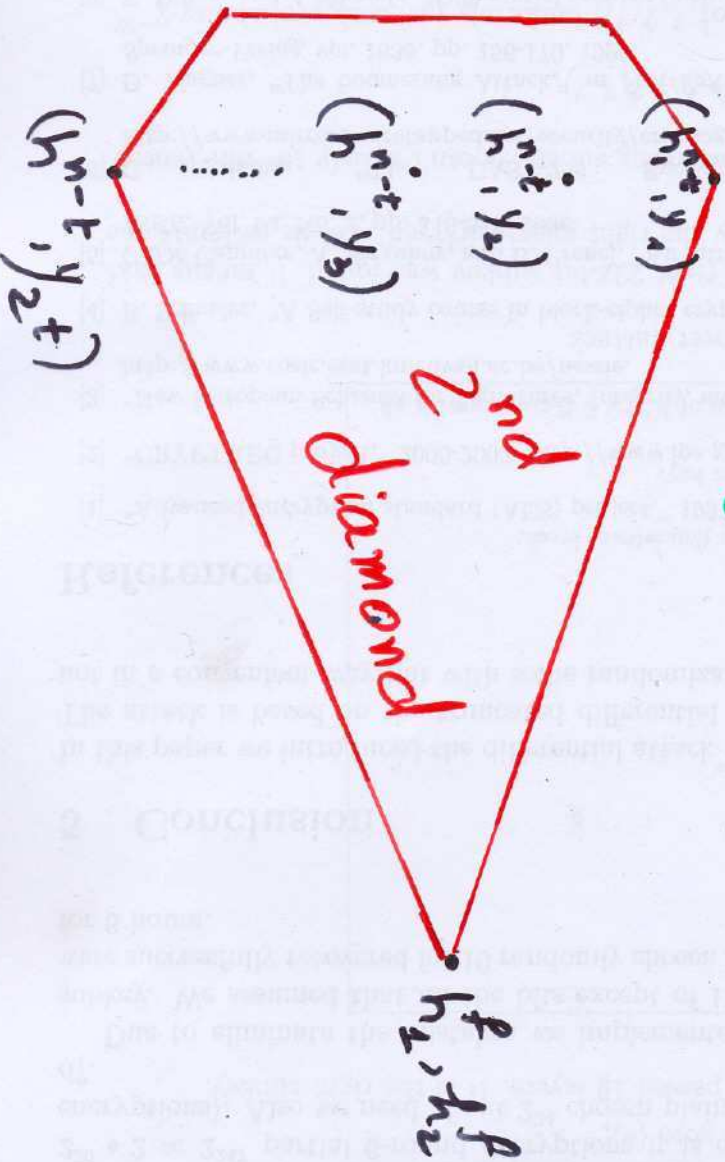
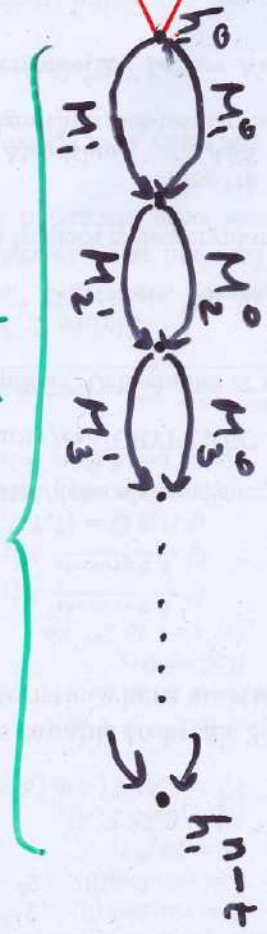
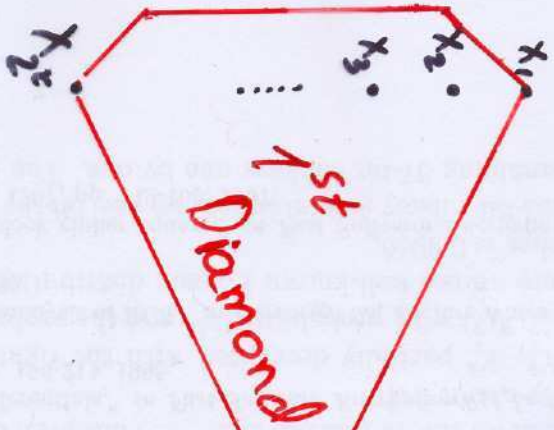
once we solved $h_1(\cdot)$, i.e., found a path that leads to $h_1(\cdot) = x$, no matter what we concatenate - we still know what's going on with $h_1(\cdot)$.

usage:

solve $h_1(\cdot)$.

then solve $h_2(\cdot)$ (without affecting $h_1(\cdot)$)

The New Attack pre computation



Building the 2nd Diamond

Consider (h_i^{n-t}, y_1) & (h_i^{n-t}, y_2)

if we take $2^{n/2}$ message blocks $\{m_i\}$

we expect m_i & m_j s.t.

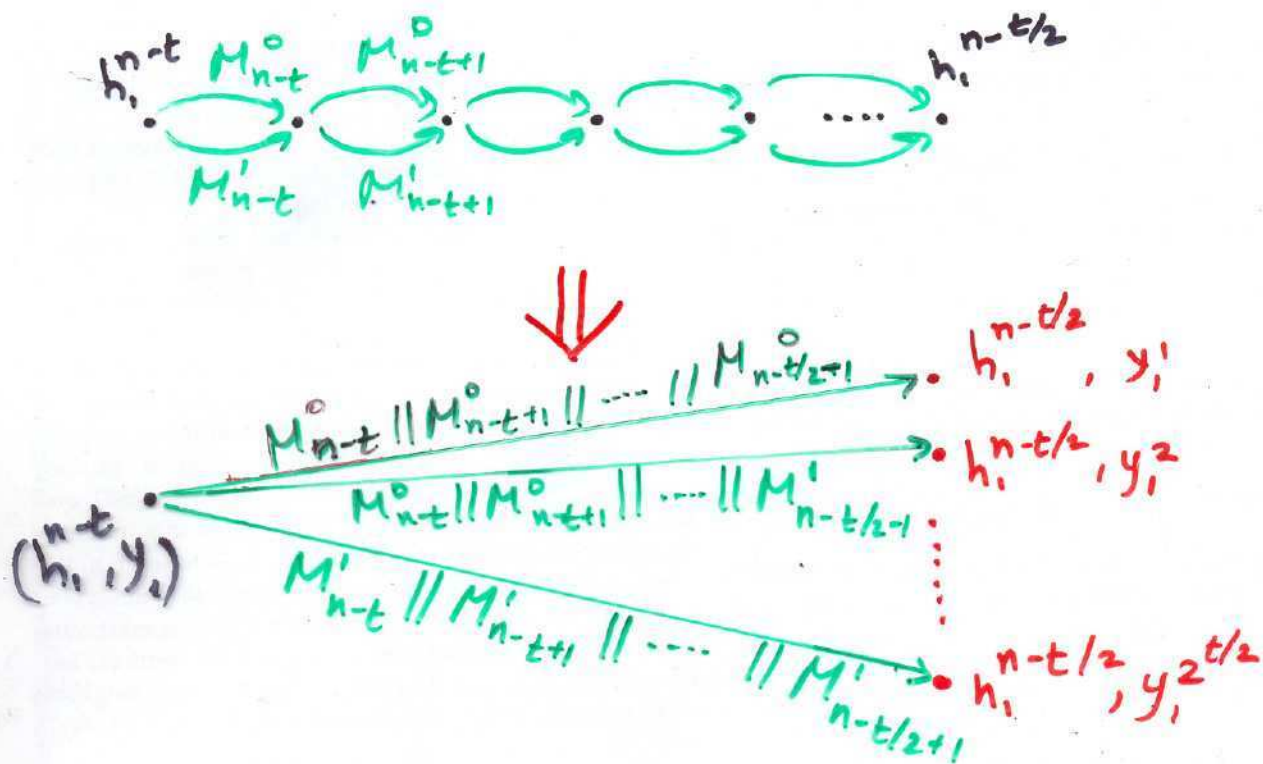
$$C_2(y_1, m_i) = C_2(y_2, m_j)$$

but it is highly unlikely that

$$C_2(h_i^{n-t}, m_i) = C_2(h_i^{n-t}, m_j)$$

as well.

The solution: Let m_i & m_j be
a multi-collision for $C_2(\cdot)$!



Several Points:

1. The same multicollisions of $C_i(\cdot)$ are used for all the values of (h_i^{n-t}, y_j)



It is possible to store the multicollision once & for each "point" store only the multicollision index.

2. The time complexity of producing the multicollision is negligible w.r.t. the overall pre-computation.
3. When there are more hash functions, e.g., $h(x) = h_1(x) || h_2(x) || h_3(x)$ the i th diamond is based on multicollisions in $h_1(x) || h_2(x) || \dots || h_{i-1}(\cdot)$

The Online Phase

1. Find a linking message to the first diamond.
2. Obtain the resulting (h_1^0, x) from the diamond.
3. Try the 2^{n-t} possible paths in the linking chain, till (h_1^{n-t}, y_i) is encountered
4. Follow the second diamond.

Online T.C. : $2^{n-t} + n \cdot 2^{n-t}$

\uparrow \uparrow
 The 1st diamond linking to the 2nd

Off line: $2^{(n+t)/2} + (n-t)2^{n/2} + \left(\frac{n-t}{2}\right) \cdot 2^{(n-t)/2}$

\uparrow \uparrow \uparrow
 generating generating generating
 the 1st diamond the linking chain the 2nd diamond

K Concatenated Hash Functions

Let $n_i = |h_i(\cdot)|$.

Off line: $2^{(n_1+t)/2} + \frac{n_2-t}{2} \cdot 2^{(n_2+t)/2} + \dots +$

T.C.

$$\left[\prod_{i=2}^{k-1} \frac{n_i}{2} \right] \cdot \frac{n_k-t}{2} \cdot 2^{(n_k+t)/2}$$

On line: $2^{n_1-t} + (n_2-t)2^{n_2-t} + n_2(n_3-t)2^{n_3-t} +$

T.C.

$$\dots + \left[\prod_{i=2}^{k-1} n_i \right] \cdot (n_k-t) \cdot 2^{n_k-t}$$

Conclusion:

The security of concatenated hashing scheme against the herding attack is linearly stronger than the strongest hash functions.