Combining Compression Functions and Block Cipher-Based Hash Functions

Asiacrypt 2006

Thomas Peyrin[1], Henri Gilbert[1], Frédéric Muller[2], Matt Robshaw[1]

[1] France Télécom R&D

[2] HSBC France

December 6, 2006

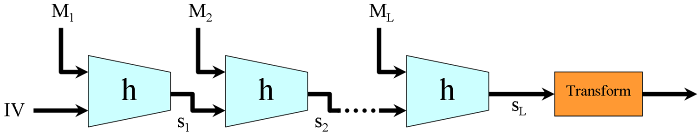
Outline

Outline

1. Introduction
2. The Framework
3. Known Generic Attacks Against Multiple Block Length Hashing
4. How to Avoid Known Generic Attacks ?
5. Conclusions

Reminder of Merkle-Damgård Construction

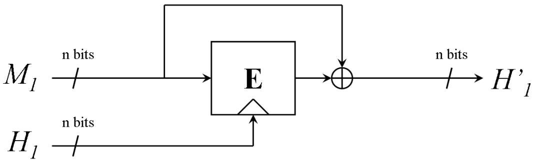
- Merkle-Damgård iteration:



- If *h* is collision resistant then *H* is collision resistant.
- But building a good and efficient compression function is hard !

Reminder of Existing Block Cipher-Based Hash Functions

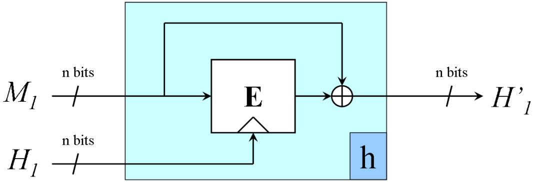
- In 1993, Preneel *et al.* studied several block cipher-based hash functions with single block length output, e.g.:



- Security proofs in the black-box model provided by Black *et al.* in 2002.
- Most hash functions are of dedicated design but recent attacks renewed interest in block cipher-based hashing.

Reminder of Existing Block Cipher-Based Hash Functions

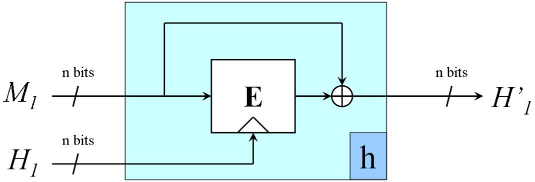
- In 1993, Preneel *et al.* studied several block cipher-based hash functions with single block length output, e.g.:



- Security proofs in the black-box model provided by Black *et al.* in 2002.
- Most hash functions are of dedicated design but recent attacks renewed interest in block cipher-based hashing.

Reminder of Existing Block Cipher-Based Hash Functions

- In 1993, Preneel *et al.* studied several block cipher-based hash functions with single block length output, e.g.:



- Security proofs in the black-box model provided by Black *et al.* in 2002.
- Most hash functions are of dedicated design but recent attacks renewed interest in block cipher-based hashing.

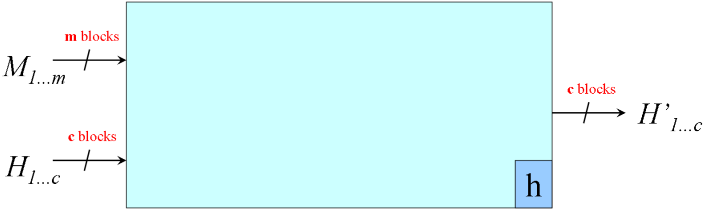
Need for Double Block Length Hash Functions

- Level of security provided by block cipher-based hash functions with single block length output is too low.
- Ideal case: with n-bit output, no attack providing a collision in less than $\Theta(2^{n/2})$ or a preimage in less than $\Theta(2^n)$ evaluations of h.
- We need double length hash functions or more generally multiple length hash functions if we want for instance AES-based hash functions.
- Previous work: [KL94], [KP96], [KP97], [KP02], [H04], [H06], [NLSL05].
- Many schemes, very few unbroken.

Outline

The Problem

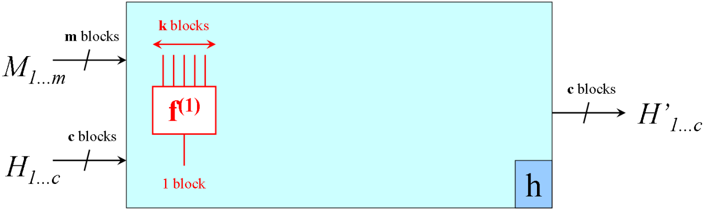
- We consider modes of operation of compression functions.
- How to build an ideal multiple length compression function h from t ideal single length with ideal and "independent" compression functions $f^{(i)}$ with one block output.



- We restrict ourselves to "parallel" constructions.

The Problem

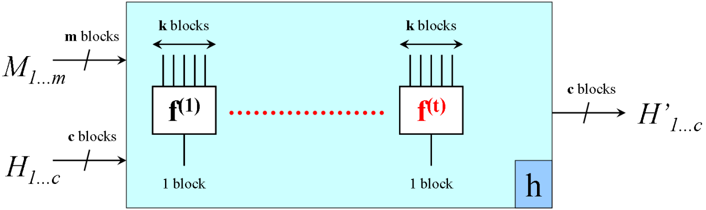
- We consider modes of operation of compression functions.
- How to build an ideal multiple length compression function h from t ideal single length with ideal and "independent" compression functions $f^{(i)}$ with one block output.



- We restrict ourselves to "parallel" constructions.

The Problem

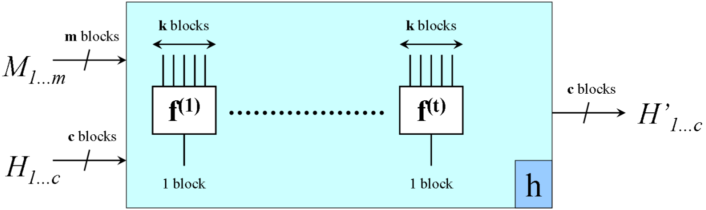
- We consider modes of operation of compression functions.
- How to build an ideal multiple length compression function h from t ideal single length with ideal and "independent" compression functions $f^{(i)}$ with one block output.



- We restrict ourselves to "parallel" constructions.

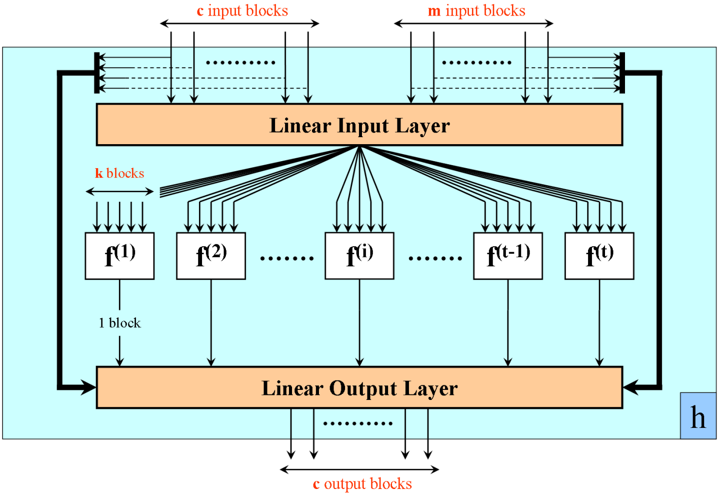
The Problem

- We consider modes of operation of compression functions.
- How to build an ideal multiple length compression function h from t ideal single length with ideal and "independent" compression functions $f^{(i)}$ with one block output.



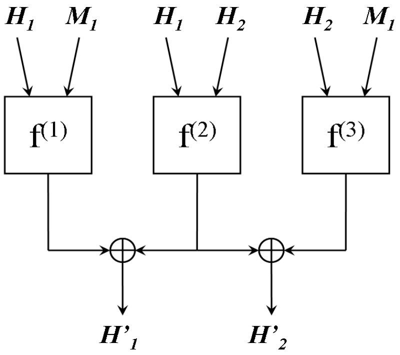
- We restrict ourselves to "parallel" constructions.

Our Framework



Example

- Nandi *et al.* scheme N_1:



$c = 2$

$m = 1$

$k = 2$

$t = 3$

Motivation of the Framework

- Very natural framework in which every known parallel double block length scheme fits in.

Name	c	t	k	m
MDC-2	2	2	2	1
PBGV	2	2	2	2
ABREAST-DM	2	2	3	1
PARALLEL-DM	2	2	2	2
Hirose family	2	2	3	1
Nandi *et al.* N_1	2	3	2	1
Nandi *et al.* N_2	2	3	3	2

- Less restrictive than previous frameworks.
- Allows to easily study all the known generic attacks, and even to find criteria to avoid them.
- Aim: derive necessary conditions on the parameters of ideal constructions.

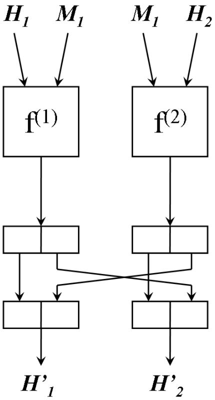
Outline

The "DF" Attack

- The "DF" attack (Degrees of Freedom):
  - possible when one can compute directly a collision or a preimage on some output blocks while keeping some degrees of freedom.
  - works for MDC-2, PGBV and Parallel-DM schemes.

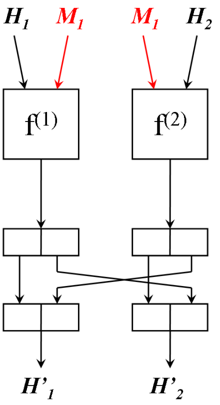
- Some output blocks can then be attacked independently !

Example of the "DF" Attack



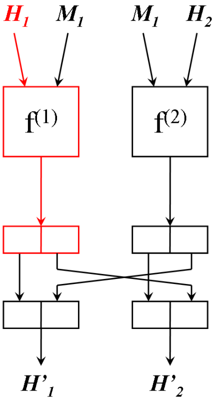
- Choose a random M_1.
- Find a collision/preimage on the left side using H_1.
- Find a collision/preimage on the right side using H_2.
- We obtain a collision/preimage with $\Theta(2^{n/2})$ and $\Theta(2^n)$ function evaluations.

Example of the "DF" Attack



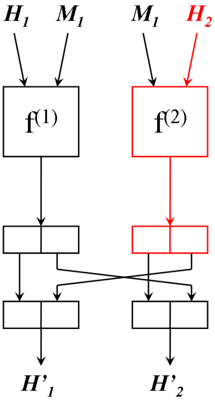
- Choose a random M_1.
- Find a collision/preimage on the left side using H_1.
- Find a collision/preimage on the right side using H_2.
- We obtain a collision/preimage with $\Theta(2^{n/2})$ and $\Theta(2^n)$ function evaluations.

Example of the "DF" Attack



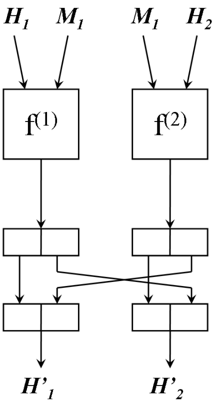
- Choose a random M_1.
- Find a collision/preimage on the left side using H_1.
- Find a collision/preimage on the right side using H_2.
- We obtain a collision/preimage with $\Theta(2^{n/2})$ and $\Theta(2^n)$ function evaluations.

Example of the "DF" Attack



- Choose a random M_1.
- Find a collision/preimage on the left side using H_1.
- Find a collision/preimage on the right side using H_2.
- We obtain a collision/preimage with $\Theta(2^{n/2})$ and $\Theta(2^n)$ function evaluations.

Example of the "DF" Attack



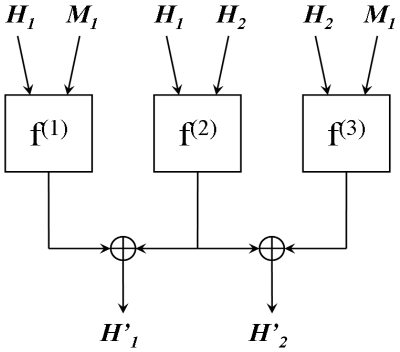
- Choose a random M_1.
- Find a collision/preimage on the left side using H_1.
- Find a collision/preimage on the right side using H_2.
- We obtain a collision/preimage with $\Theta(2^{n/2})$ and $\Theta(2^n)$ function evaluations.

The "MUL" Attack

- The "MUL" attack (Multicollisions or Multipreimages):
  - possible when one can compute multicollisions or multipreimages on some output block in less then expected for an ideal compression function.
  - works for Nandi *et al.* schemes N1 and N2.

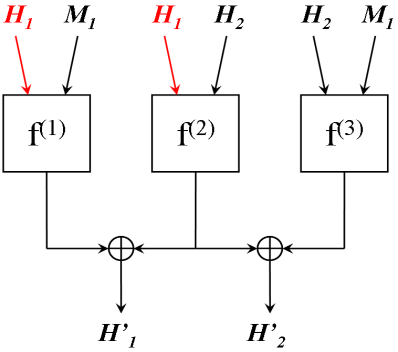
- Some output blocks can then be attacked independently !

Example of the "MUL" Attack



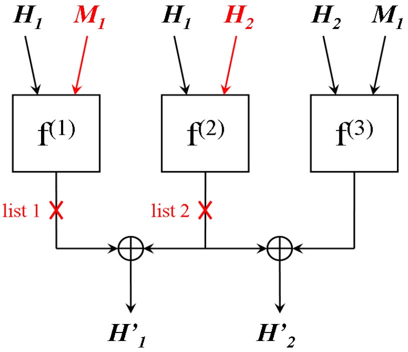
- Choose a random H_1.
- Build 2 lists of $f^{(1)}$ and $f^{(2)}$ outputs, with M_1 and H_2.
- Wagner's technique: find multicollisions/multipreimages for the left output with low cost.
- Find a collision/preimage on the right side among the previously computed multicollisions/multipreimages.
- We obtain a collision/preimage with $\Theta(2^{2n/3})$ and $\Theta(2^n)$ function evaluations.

Example of the "MUL" Attack



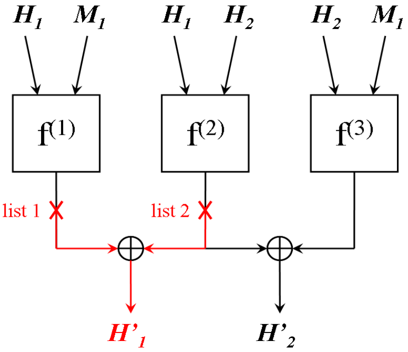
- Choose a random H_1.
- Build 2 lists of $f^{(1)}$ and $f^{(2)}$ outputs, with M_1 and H_2.
- Wagner's technique: find multicollisions/multipreimages for the left output with low cost.
- Find a collision/preimage on the right side among the previously computed multicollisions/multipreimages.
- We obtain a collision/preimage with $\Theta(2^{2n/3})$ and $\Theta(2^n)$ function evaluations.

Example of the "MUL" Attack



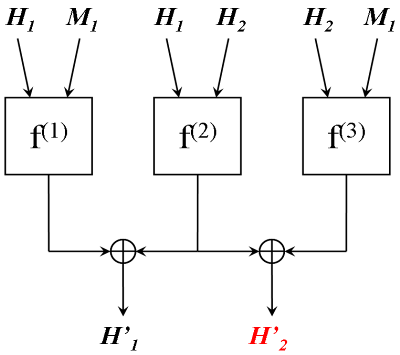
- Choose a random H_1.
- Build 2 lists of $f^{(1)}$ and $f^{(2)}$ outputs, with M_1 and H_2.
- Wagner's technique: find multicollisions/multipreimages for the left output with low cost.
- Find a collision/preimage on the right side among the previously computed multicollisions/multipreimages.
- We obtain a collision/preimage with $\Theta(2^{2n/3})$ and $\Theta(2^n)$ function evaluations.

Example of the "MUL" Attack



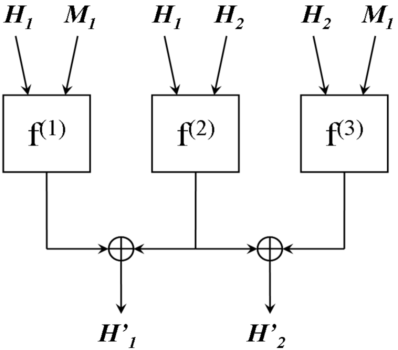
- Choose a random H_1.
- Build 2 lists of $f^{(1)}$ and $f^{(2)}$ outputs, with M_1 and H_2.
- Wagner's technique: find multicollisions/multipreimages for the left output with low cost.
- Find a collision/preimage on the right side among the previously computed multicollisions/multipreimages.
- We obtain a collision/preimage with $\Theta(2^{2n/3})$ and $\Theta(2^n)$ function evaluations.

Example of the "MUL" Attack



- Choose a random H_1.
- Build 2 lists of $f^{(1)}$ and $f^{(2)}$ outputs, with M_1 and H_2.
- Wagner's technique: find multicollisions/multipreimages for the left output with low cost.
- Find a collision/preimage on the right side among the previously computed multicollisions/multipreimages.
- We obtain a collision/preimage with $\Theta(2^{2n/3})$ and $\Theta(2^n)$ function evaluations.

Example of the "MUL" Attack



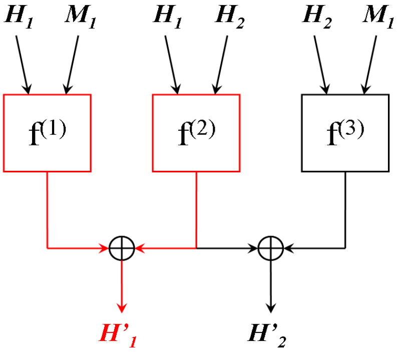
- Choose a random H_1.
- Build 2 lists of $f^{(1)}$ and $f^{(2)}$ outputs, with M_1 and H_2.
- Wagner's technique: find multicollisions/multipreimages for the left output with low cost.
- Find a collision/preimage on the right side among the previously computed multicollisions/multipreimages.
- We obtain a collision/preimage with $\Theta(2^{2n/3})$ and $\Theta(2^n)$ function evaluations.

Outline

Active Functions of an Output Block

Let d be the minimum number of active functions $f^{(i)}$ involved in the expression of a combination of the output blocks.

- $f^{(1)}$ and $f^{(2)}$ are active for the output block H'_1
- We have $d = 2$



Obtaining Security Criteria from Generic Attacks

- For the DF attack: every input block (message or chaining variable) must influence every output block.
- For the MUL attack: every possible pair of input blocks (message or chaining variable) must appear in at least one of the "active" functions $f^{(i)}$ of every output block.

"...applying any simple (in both directions) invertible transformation to the input and to the output of the hash round function yields a new hash round function with the same security as the original one. "

(Meier and Staffelbach - Eurocrypt'89)

Obtaining Security Criteria from Generic Attacks

- For the DF attack: every input block (message or chaining variable) must influence every output block.
- For the MUL attack: every possible pair of input blocks (message or chaining variable) must appear in at least one of the "active" functions $f^{(i)}$ of every output block.

The two criteria must be true for any invertible transformation of the input blocks or/and the output blocks.

Using the Security Criteria (1)

- The DF attack:
  - General bound $d \geq \lceil \frac{m+c}{k} \rceil$ for any set of parameters.
- The MUL attack:
  - General analysis is much more complicated, but case by case reasoning is possible.
  - We get better bounds on d: $d \geq 3$ for $m + c \geq 3$ and $k = 2$.
  - Generic analysis that can be reused for different parameter sets.

Using the Security Criteria (2)

- From the previous bounds on d, we can obtain bounds on t thanks to coding theory.
- Problem of finding a binary code of length t with minimal distance d and dimension c.
- Singleton bound: $c \leq t - d + 1$ and so $t \geq c + d - 1$.
- The Hamming bound is more involved but gives tighter results.
- We obtain a lower bound t_{min} on the number of internal functions to use, given the parameters m, c and k.

Results

Parameters			*Bounds*	
c	k	m	$d \geq$	$t_{\min}$
2	2	1	3	5
2	2	2	3	5
2	3	1	-	-
2	3	2	3	5
3	2	1	3	6
3	2	2	4	7
3	3	1	3	6
3	3	2	3	6
4	2	1	4	8
4	2	2	4	8
4	3	1	3	7
4	3	2	3	7

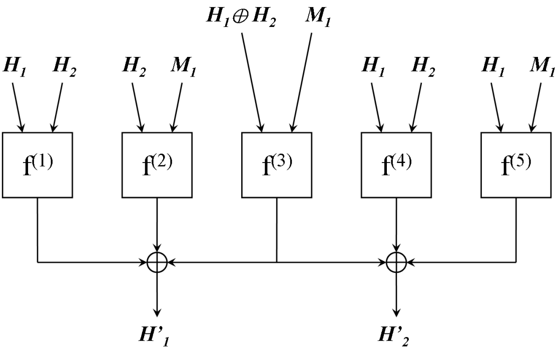
Results

Parameters			*Bounds*	
c	k	m	$d \geq$	$t_{\min}$
2	2	1	3	5
2	2	2	3	5
2	3	1	-	-
2	3	2	3	5
3	2	1	3	6
3	2	2	4	7
3	3	1	3	6
3	3	2	3	6
4	2	1	4	8
4	2	2	4	8
4	3	1	3	7
4	3	2	3	7

Results

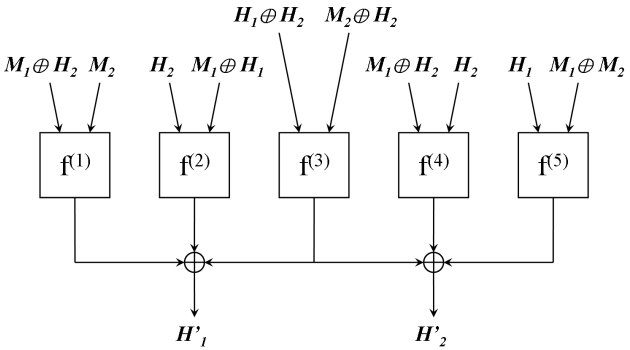
Parameters			*Bounds*	
c	k	m	$d \geq$	$t_{\min}$
2	2	1	3	5
2	2	2	3	5
2	3	1	-	-
2	3	2	3	5
3	2	1	3	6
3	2	2	4	7
3	3	1	3	6
3	3	2	3	6
4	2	1	4	8
4	2	2	4	8
4	3	1	3	7
4	3	2	3	7

Candidate Double Length Scheme



- Immune to DF and MUL attacks.
- No known attack, but no security proof.

Candidate Double Length Scheme



- Immune to DF and MUL attacks.
- No known attack, but no security proof.

Outline

Conclusions

- We introduced a new framework to build multiple block length hash functions.
- We analysed existing generic attacks and their implications on parameters of ideal constructions.
- We identified schemes which are immune to DF and MUL attacks.

- Study the serial case ==> more general and more difficult to analyse but may lead to more efficient schemes.
- Specify an efficient, generic and secure way to instantiate "independent" compression functions.
- Find other efficient schemes for interesting sets of parameters.
- Proofs of security: we get rigorous bounds in terms of number of queries to the internal compression functions.
- Open question: for the new candidate schemes, is it possible to find an attack matching the security bound or to improve the security bound in terms of number of operations.