Grøstl – a SHA-3 candidate

Krystian Matusiewicz

Wroclaw University of Technology

CECC 2010, June 12, 2010

Talk outline

- Cryptographic hash functions
- NIST SHA-3 Competition
- Grøstl

Cryptographic hash functions

Cryptographic hash functions: why?

- We want to have a short, fixed length "fingerprint" of any piece of data
- Different fingerprints – certainly different data
- Identical fingerprints – most likely the same data
- No one can get any information about the data from the fingerprint

Random Oracle

Construction:

- Box with memory
- On a new query: pick randomly and uniformly the answer, remember it and return the result
- On a repeating query, repeat the answer (function)

Random Oracle

Construction:

- Box with memory
- On a new query: pick randomly and uniformly the answer, remember it and return the result
- On a repeating query, repeat the answer (function)

Properties:

- No information about the data
- To find a preimage: 2^n queries for n-bit outputs
- To find collisions: $2^{n/2}$ queries
- Random behaviour

Random Oracle

Construction:

- Box with memory
- On a new query: pick randomly and uniformly the answer, remember it and return the result
- On a repeating query, repeat the answer (function)

Properties:

- No information about the data
- To find a preimage: 2^n queries for n-bit outputs
- To find collisions: $2^{n/2}$ queries
- Random behaviour

We want to construct concrete algorithms which are as close to this behaviour as possible.

NIST SHA-3 Competition

The world after Wang (after 2004)

hash	designed	Wang-inspired attacks
MD4	1990	collisions by hand calculation
MD5	1992	collisions in ms on a computer
SHA-0	1993	collisions in $\approx 1hr$
SHA-1	1995	attacked, 2^{69}
SHA-2	2002	no attack

Many other hash functions similar to MD5/SHA-1 were subsequently broken using Wang's method.

NIST SHA-3 competition

- There is only one hash function standardized by NIST which is not broken (SHA-256/SHA-512)
- But design principles are somehow similar to the broken ones, is it secure enough?
- Would be good to have a backup plan
- Let's organize a competition like the one for Advanced Encryption Standard!

First-round SHA-3 candidates

Abacus	ARIRANG	AURORA	BLAKE	Blender
Blue Midnight Wish	BOOLE	Cheetah	CHI	CRUNCH
CubeHash	DCH	Dynamic SHA	Dynamic SHA2	ECHO
ECOH	EDON-R	EnRUPT	ESSENCE	FSB
Fugue	Grøstl	Hamsi	JH	Keccak
Khichidi-1	LANE	Lesamnta	Luffa	LUX
MCSSHA-3	MD6	MeshHash	NaSHA	SANDstorm
Sarmal	Sgail	Shabal	SHAMATA	SHAvite-3
SIMD	Skein	Spectral Hash	StreamHash	SWIFFTX
Tangle	TIB3	Twister	Vortex	WaMM
Waterfall				

Second-round SHA-3 candidates

Abacus	ARIRANG	AURORA	BLAKE	Blender
Blue Midnight Wish	BOOLE	Cheetah	CHI	CRUNCH
CubeHash	DCH	Dynamic SHA	Dynamic SHA2	ECHO
ECOH	EDON-R	EnRUPT	ESSENCE	FSB
Fugue	Grøstl	Hamsi	JH	Keccak
Khichidi-1	LANE	Lesamnta	Luffa	LUX
MCSSHA-3	MD6	MeshHash	NaSHA	SANDstorm
Sarmal	Sgail	Shabal	SHAMATA	SHAvite-3
SIMD	Skein	Spectral Hash	StreamHash	SWIFFTX
Tangle	TIB3	Twister	Vortex	WaMM
Waterfall				

What happens next?

- August 2010 – second NIST workshop
- Around 5 finalists will be announced before the end of the year
- The winner selected in 2012

The SHA-3 Zoo / Hash function Zoo

Everything you ever wanted to know about SHA-3 candidates

- `ehash.iaik.tugraz.at/wiki/The_SHA-3_Zoo`
- `ehash.iaik.tugraz.at/wiki/The_Hash_Function_Zoo`

Grøstl

The Grøstl team


Søren S. Thomsen


Martin Schläffer


Christian Rechberger

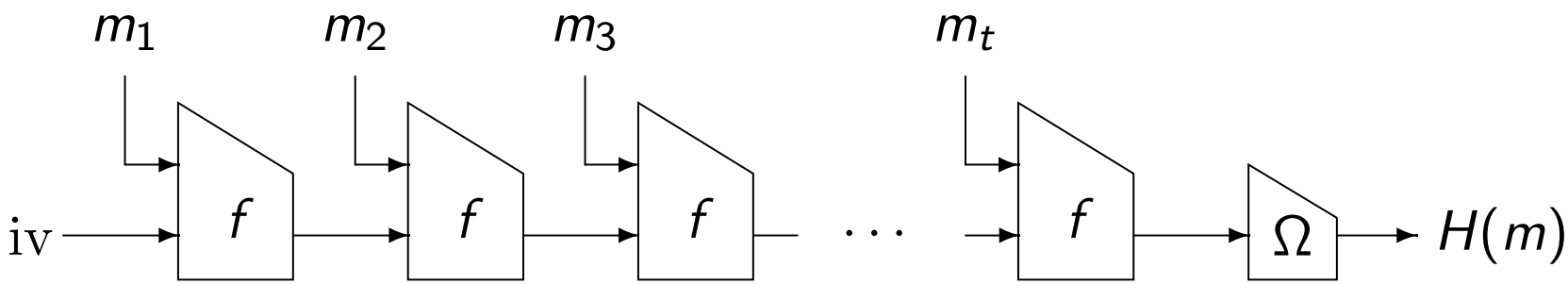

Florian Mendel


Lars R. Knudsen


Praveen Gauravaram

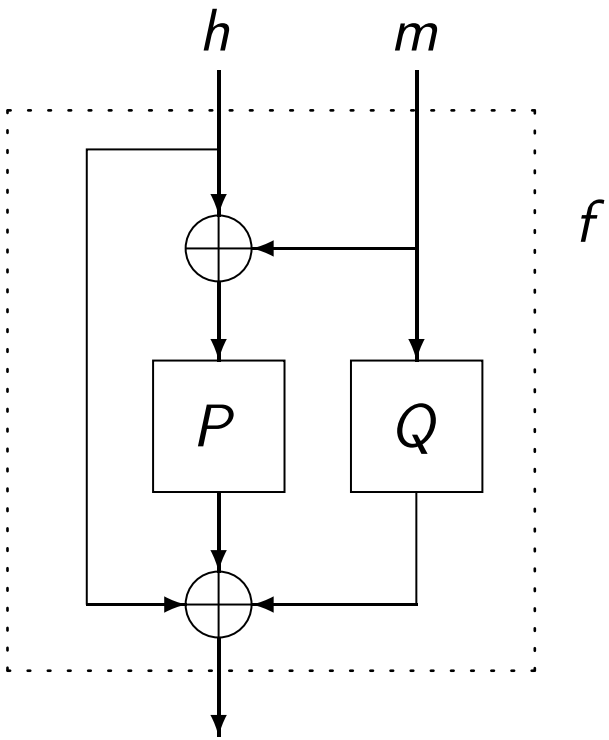
& K.M.

Grøstl hash function



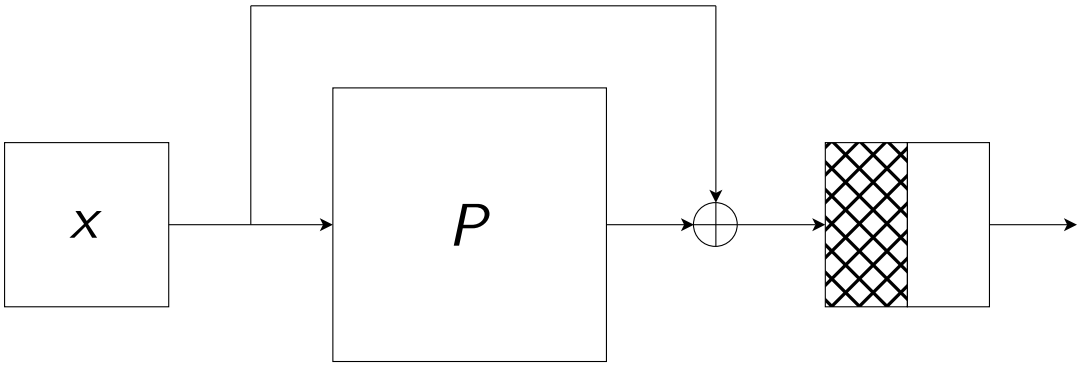
- Iterated wide-pipe
- Output transformation

Grøstl Compression Function



- P, Q – large, fixed permutations
- Security reductions: collision, preimage resistance provided that permutations behave like ideal ones

Grøstl Output Transformation

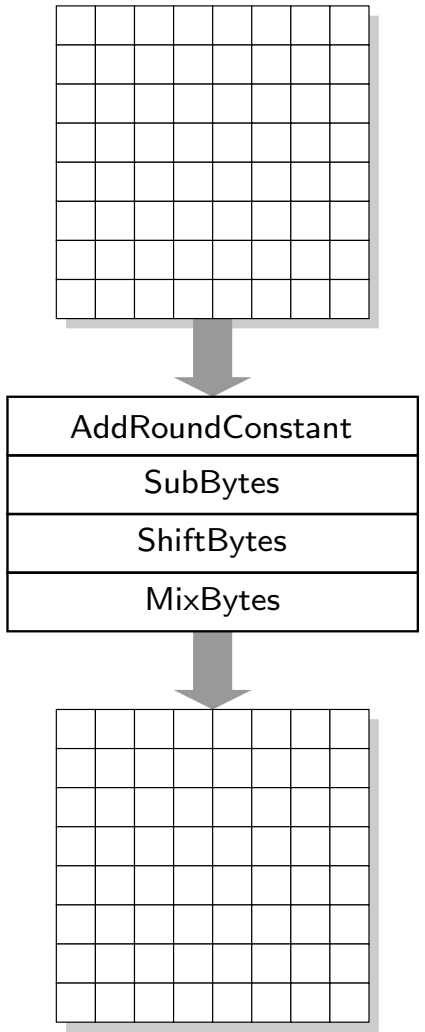


- Provides protection against some attacks
- Random behaviour

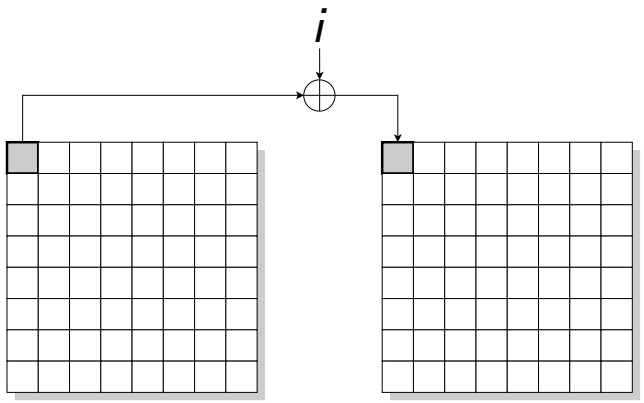
Designing an "ideal permutation"

- How to design an "ideal permutation"?
- No way of telling it apart from randomly chosen permutation
- No visible structure
- No differential trail with significant probability
- Wide-trail strategy

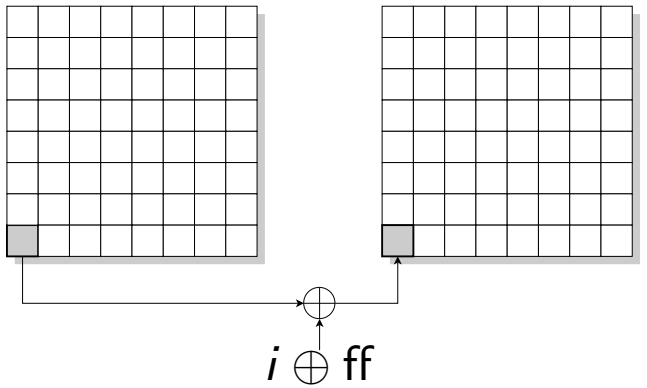
Grøstl : One round of P, Q ($\times 10$)



Grøstl : AddRoundConstant



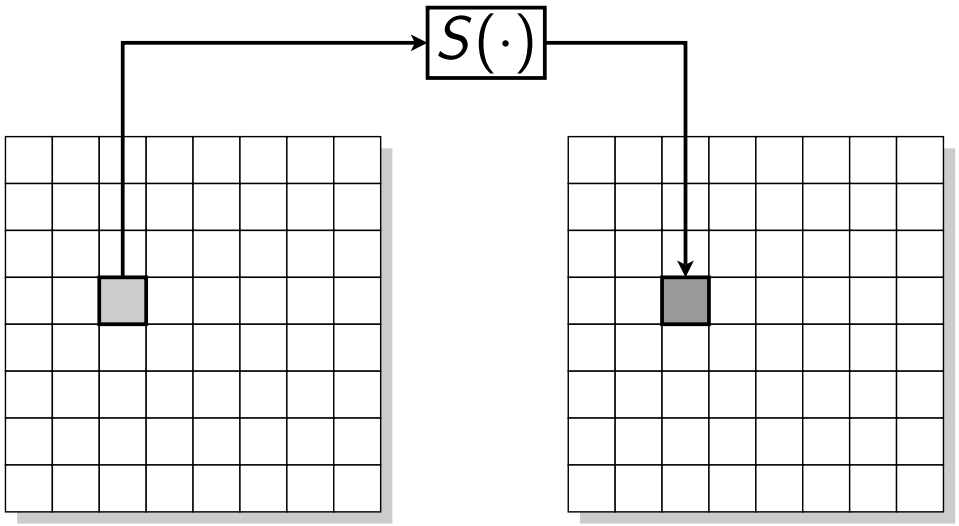
AddRoundConstant of P



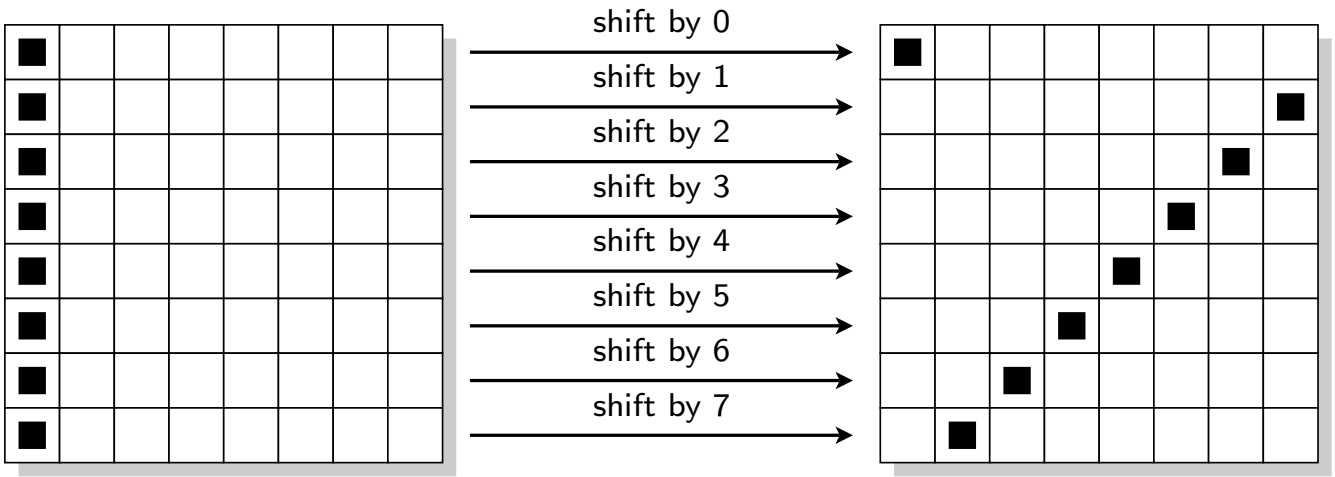
AddRoundConstant of Q

- Permutations must be different (security)
- Different constants for P and Q

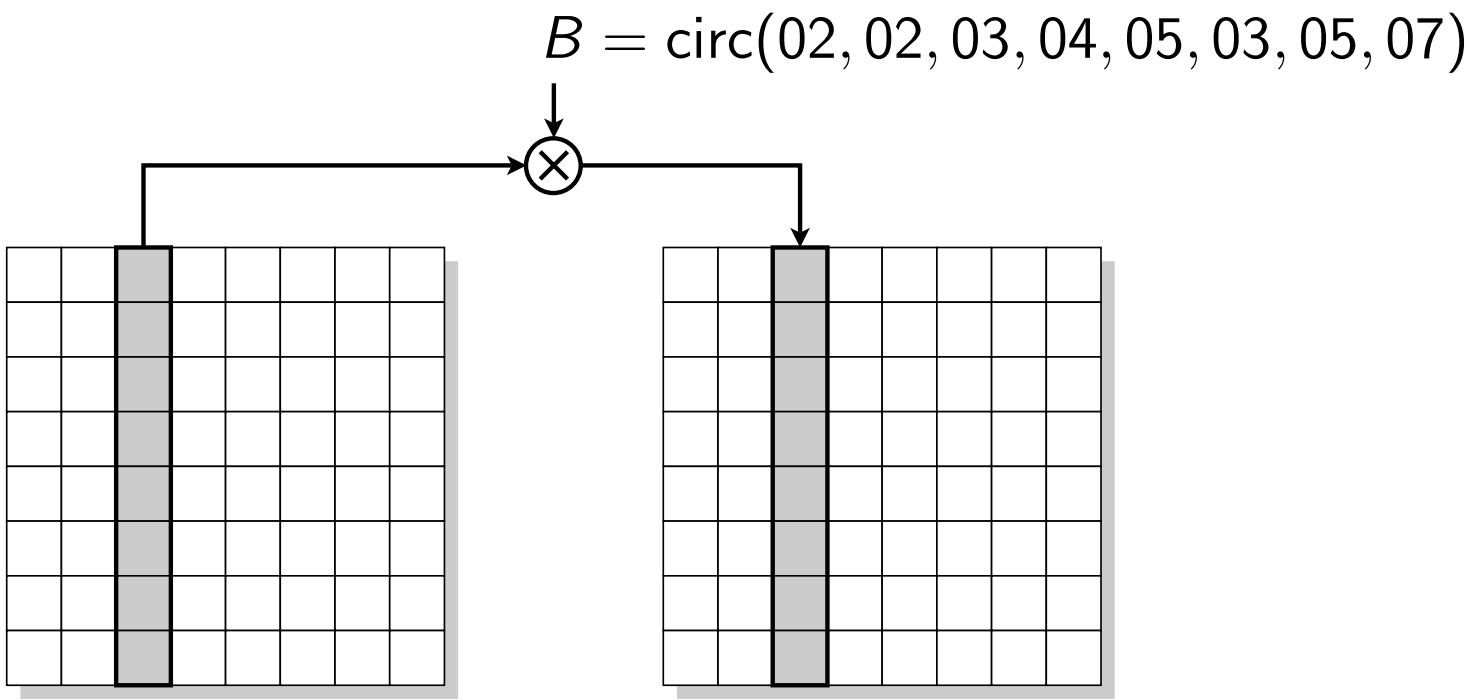
Grøstl : SubBytes



Grøstl : ShiftBytes



Grøstl : MixBytes



Cryptanalytic Results

Hash function collisions	
Grøstl-256	4/10
Grøstl-512	5/14

Compression function collisions	
Grøstl-256	7/10
Grøstl-512	7/14

Permutation non-randomness	
Grøstl-256	8/10

Grøstl : Enjoy!