Cryptanalysis of a Hash Function Based on Quasi-Cyclic Codes

Pierre-Alain Fouque, Gaëtan Leurent

École Normale Supérieure
Paris, France

RSA CONFERENCE 2008

Hash Functions

$$F : \{0,1\}^* \mapsto \{0,1\}^n$$

Should behave "like a random oracle"...

Collision attack

Given F, find $M_1 \neq M_2$ s.t. $F(M_1) = F(M_2)$.
Ideal security: $2^{n/2}$.

Second-preimage attack

Given F and M_1, find $M_2 \neq M_1$ s.t. $F(M_1) = F(M_2)$.
Ideal security: 2^n.

Preimage attack

Given F and $\overline{H}$, find M s.t. $F(M) = \overline{H}$.
Ideal security: 2^n.

Hash Functions

$$F : \{0,1\}^* \mapsto \{0,1\}^n$$

Should behave "like a random oracle"...

Collision attack

Given F, find $M_1 \neq M_2$ s.t. $F(M_1) = F(M_2)$.
Ideal security: $2^{n/2}$.

Second-preimage attack

Given F and M_1, find $M_2 \neq M_1$ s.t. $F(M_1) = F(M_2)$.
Ideal security: 2^n.

Preimage attack

Given F and $\overline{H}$, find M s.t. $F(M) = \overline{H}$.
Ideal security: 2^n.

Hash Function Design

Most hash functions are dedicated designs based on *symmetric crypto* concepts:
e.g. MDx, SHA-x, Whirlpool, RadioGatún, Grindahl, ...

Some designs are based on a provable security approach: the security relies on a given hard problem (like *public key crypto*):
e.g. FSB, LASH, SWIFFT, SQUASH, ...

Proof of security should be taken with caution

- Many of them are asymptotic proofs but the concrete function has a fixed size.
- Reduction to an NP-complete problem means that *some* instance are hard, but the fixed instance could be easy.
- Sometimes the attack model is too weak.

Hash Function Design

Most hash functions are dedicated designs based on *symmetric crypto* concepts:
e.g. MDx, SHA-x, Whirlpool, RadioGatún, Grindahl, ...

Some designs are based on a provable security approach: the security relies on a given hard problem (like *public key crypto*):
e.g. FSB, LASH, SWIFFT, SQUASH, ...

Proof of security should be taken with caution

- Many of them are asymptotic proofs but the concrete function has a fixed size.
- Reduction to an NP-complete problem means that *some* instance are hard, but the fixed instance could be easy.
- Sometimes the attack model is too weak.

Hash Function Design

Most hash functions are dedicated designs based on *symmetric crypto* concepts:
e.g. MDx, SHA-x, Whirlpool, RadioGatún, Grindahl, ...

Some designs are based on a provable security approach:
the security relies on a given hard problem
(like *public key crypto*):
e.g. FSB, LASH, SWIFFT, SQUASH, ...

Proof of security should be taken with caution

- Many of them are asymptotic proofs but the concrete function has a fixed size.
- Reduction to an NP-complete problem means that *some* instance are hard, but the fixed instance could be easy.
- Sometimes the attack model is too weak.

Hash Function Cryptanalysis

Many hash functions in use today are broken:

1990 MD4 design (Rivest)
1992 MD5 design (Rivest)
1995 SHA-1 design (NIST)
1996 MD4 collisions (Dobbertin)
2001 SHA-2 family design (NIST)
2004 MD5 collisions (Wang *et al.*)
2005 SHA-1 collision attack (Wang *et al.*)

Best collision attacks

MD4 Complexity 2^1 (Wang *et al.* – Sasaki *et al.*)
MD5 Complexity 2^{22} (Wang *et al.* – Klima)
SHA-1 Complexity $2^{60.x}$ (Wang *et al.* – Rechberger *et al.*)

Real impact is unclear, but new designs are welcome (cf. SHA-3).

Hash Function Cryptanalysis

Many hash functions in use today are broken:

1990 MD4 design (Rivest)
1992 MD5 design (Rivest)
1995 SHA-1 design (NIST)
1996 MD4 collisions (Dobbertin)
2001 SHA-2 family design (NIST)
2004 MD5 collisions (Wang *et al.*)
2005 SHA-1 collision attack (Wang *et al.*)

Best collision attacks

MD4 Complexity 2^1 (Wang *et al.* – Sasaki *et al.*)
MD5 Complexity 2^{22} (Wang *et al.* – Klima)
SHA-1 Complexity $2^{60.x}$ (Wang *et al.* – Rechberger *et al.*)

Real impact is unclear, but new designs are welcome (cf. SHA-3).

Hash Function Cryptanalysis

Many hash functions in use today are broken:

1990	MD4 design (Rivest)
1992	MD5 design (Rivest)
1995	SHA-1 design (NIST)
1996	MD4 collisions (Dobbertin)
2001	SHA-2 family design (NIST)
2004	MD5 collisions (Wang *et al.*)
2005	SHA-1 collision attack (Wang *et al.*)

Best collision attacks

MD4	Complexity 2^1 (Wang *et al.* – Sasaki *et al.*)
MD5	Complexity 2^{22} (Wang *et al.* – Klima)
SHA-1	Complexity $2^{60.x}$ (Wang *et al.* – Rechberger *et al.*)

Real impact is unclear, but new designs are welcome (cf. SHA-3).

Outline

The IFSB Hash Function

- Description
- Previous cryptanalysis
  - Wagner's Generalized Birthday
  - Linearization Attack

The Cyclic attack

- Using Periodic Messages
- Description of the attack
- Solving the cyclic equations
- Scope of the attack

Description of FSB

$x = \quad 1 \quad 1 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1 \quad 1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 1 \quad 1$

$$F(x) = \mathcal{H} \times \varphi(x)$$

$\mathcal{H}$: random $r \times n$ matrix

φ : encodes s bits to n bits with weight w

Description of FSB

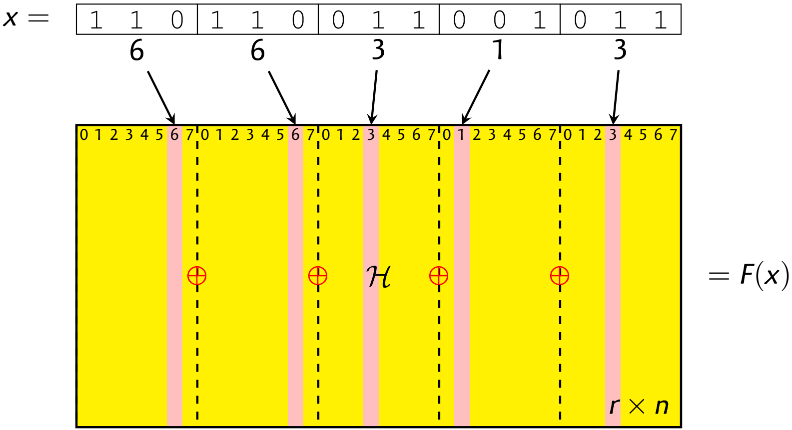
$$x = \begin{array}{|ccc|ccc|ccc|ccc|ccc|} \hline 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ \hline \end{array}$$

$$\begin{array}{ccccc} 6 & 6 & 3 & 1 & 3 \end{array}$$

$$\boxed{F(x) = \mathcal{H} \times \varphi(x)}$$

$\mathcal{H}$: random $r \times n$ matrix
φ: encodes s bits to n bits with weight w

Description of FSB

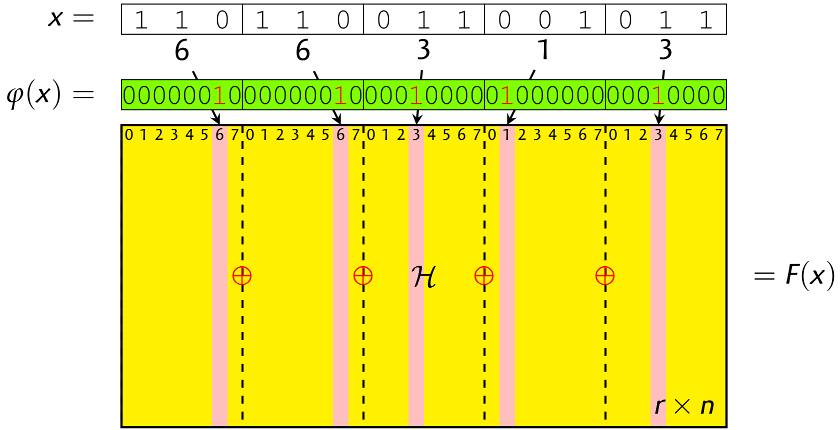


$$F(x) = \mathcal{H} \times \varphi(x)$$

$\mathcal{H}$: random $r \times n$ matrix

φ : encodes s bits to n bits with weight w

Description of FSB



$$\boxed{F(x) = \mathcal{H} \times \varphi(x)}$$

$\mathcal{H}$: random $r \times n$ matrix

φ : encodes s bits to n bits with weight w

Rationale of FSB

$$\boxed{F(x) = \mathcal{H} \times \varphi(x)}$$

$\mathcal{H}$: random $r \times n$ matrix
φ : encodes s bits to n bits with weight w

- $\mathcal{H}$ is the parity matrix of a linear code.
- $\varphi(x)$ is an error pattern.
- $\mathcal{H} \times \varphi(x)$ is a syndrome.
- Inversion and collision are related to coding theory problems (syndrome decoding) on $\mathcal{H}$.

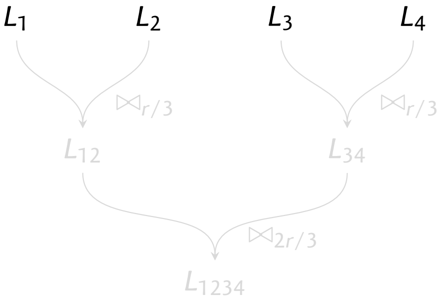
From FSB to IFSB

- Main problem: the matrix $\mathcal{H}$ is huge...
- Use a quasi-cyclic matrix:

$$\mathcal{H} = \left[\begin{array}{ccccc|ccccc|c}
\alpha_0 & \alpha_1 & \dots & \alpha_{r-2} & \alpha_{r-1} & \beta_0 & \beta_1 & \dots & \beta_{r-2} & \beta_{r-1} & \\
\alpha_{r-1} & \alpha_0 & \alpha_1 & & \alpha_{r-2} & \beta_{r-1} & \beta_0 & \beta_1 & & \beta_{r-2} & \\
\vdots & \alpha_{r-1} & \alpha_0 & \ddots & \vdots & \vdots & \beta_{r-1} & \beta_0 & \ddots & \vdots & \dots \\
\alpha_2 & & \ddots & \ddots & \alpha_1 & \beta_2 & & \ddots & \ddots & \beta_1 & \\
\alpha_1 & \alpha_2 & \dots & \alpha_{r-1} & \alpha_0 & \beta_1 & \beta_2 & \dots & \beta_{r-1} & \beta_0 &
\end{array}\right]$$

Wagner's Generalized Birthday

- Solves the k-sum problem:
  find $l_1 \in L_1, .., l_k \in L_k$ s.t. $\bigoplus_{i=1}^{k} l_k = 0$.
- $L \bowtie_j L' = \{(l, l') \in L \times L' \mid (l \oplus l')^{[0..j-1]} = 0^j\}$.
- For r bits, start with 2^a lists of $2^{r/(a+1)}$ elements.



4 lists of $2^{r/3}$ elements

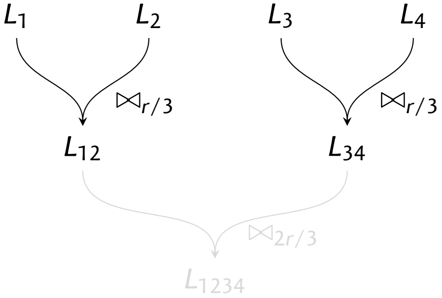
2 lists of $2^{r/3}$ elements with $2^{r/3}$ zeros

1 list of $2^{r/3}$ elements with $2^{2r/3}$ zeros

- One element in L_{1234} is zero.
- Complexity: 7 sorts.

Wagner's Generalized Birthday

- Solves the k-sum problem:
  find $l_1 \in L_1, .., l_k \in L_k$ s.t. $\bigoplus_{i=1}^{k} l_k = 0$.
- $L \bowtie_j L' = \{(l, l') \in L \times L' \mid (l \oplus l')^{[0..j-1]} = 0^j\}$.
- For r bits, start with 2^a lists of $2^{r/(a+1)}$ elements.



4 lists of $2^{r/3}$ elements

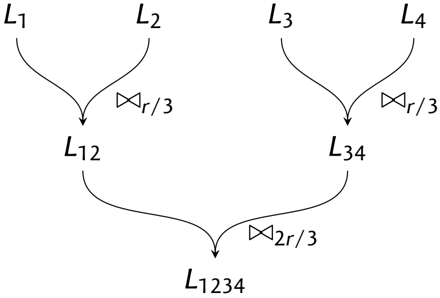
2 lists of $2^{r/3}$ elements with $2^{r/3}$ zeros

1 list of $2^{r/3}$ elements with $2^{2r/3}$ zeros

- One element in L_{1234} is zero.
- Complexity: 7 sorts.

Wagner's Generalized Birthday

- Solves the k-sum problem:
  find $l_1 \in L_1, .., l_k \in L_k$ s.t. $\bigoplus_{i=1}^{k} l_k = 0$.
- $L \bowtie_j L' = \{(l, l') \in L \times L' \mid (l \oplus l')^{[0..j-1]} = 0^j\}$.
- For r bits, start with 2^a lists of $2^{r/(a+1)}$ elements.



4 lists of $2^{r/3}$ elements

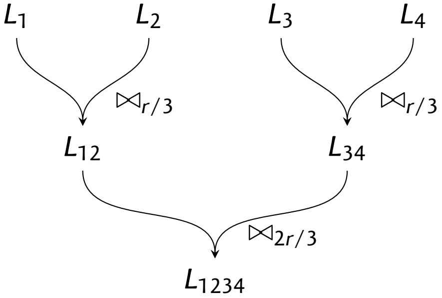
2 lists of $2^{r/3}$ elements with $2^{r/3}$ zeros

1 list of $2^{r/3}$ elements with $2^{2r/3}$ zeros

- One element in L_{1234} is zero.
- Complexity: 7 sorts.

Wagner's Generalized Birthday

- Solves the k-sum problem:
  find $l_1 \in L_1, .., l_k \in L_k$ s.t. $\bigoplus_{i=1}^{k} l_k = 0$.
- $L \bowtie_j L' = \{(l, l') \in L \times L' \mid (l \oplus l')^{[0..j-1]} = 0^j\}$.
- For r bits, start with 2^a lists of $2^{r/(a+1)}$ elements.



4 lists of $2^{r/3}$ elements

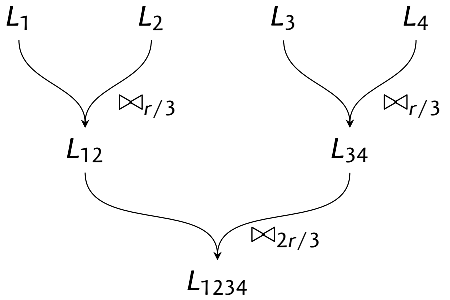
2 lists of $2^{r/3}$ elements with $2^{r/3}$ zeros

1 list of $2^{r/3}$ elements with $2^{2r/3}$ zeros

- One element in L_{1234} is zero.
- Complexity: 7 sorts.

Wagner's Generalized Birthday

- Solves the k-sum problem:
  find $l_1 \in L_1, .., l_k \in L_k$ s.t. $\bigoplus_{i=1}^{k} l_k = 0$.
- $L \bowtie_j L' = \{(l, l') \in L \times L' \mid (l \oplus l')^{[0..j-1]} = 0^j\}$.
- For r bits, start with 2^a lists of $2^{r/(a+1)}$ elements.



4 lists of $2^{r/3}$ elements

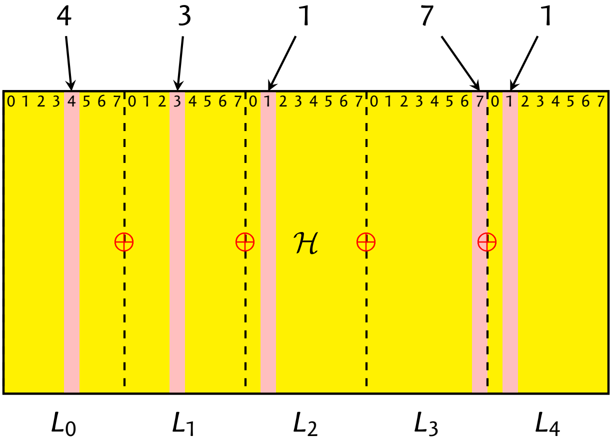
2 lists of $2^{r/3}$ elements with $2^{r/3}$ zeros

1 list of $2^{r/3}$ elements with $2^{2r/3}$ zeros

- One element in L_{1234} is zero.
- Complexity: 7 sorts.

Wagner's Generalized Birthday

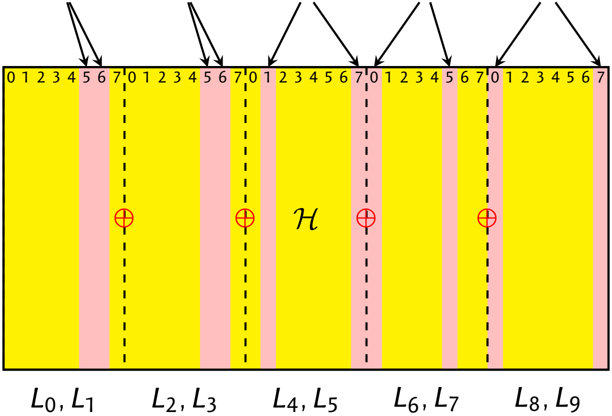
Application to FSB is straightforward: preimage and collision.



Complexity is still exponential; used as security parameter.

Wagner's Generalized Birthday

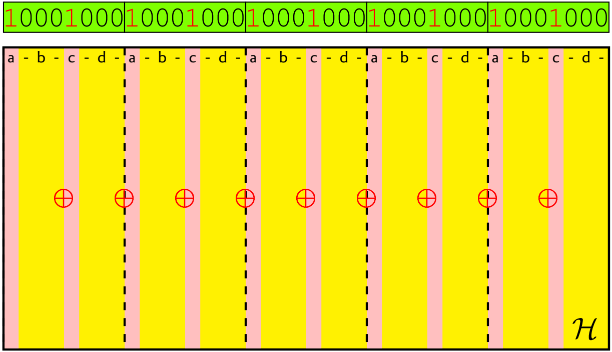
Application to FSB is straightforward: preimage and collision.



Complexity is still exponential; used as security parameter.

Linearization Attack

- Choose a, b, c, d. Start with $x = a^w$ and $x' = c^w$.

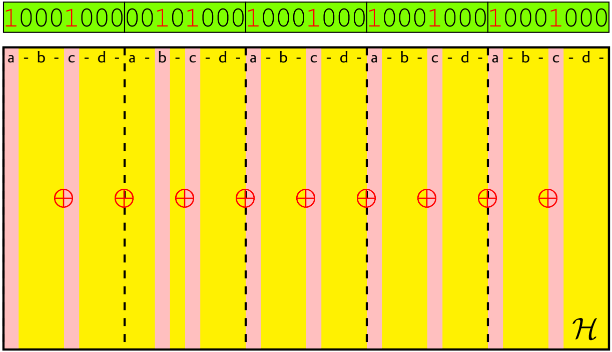


$$= F(a^w) \oplus F(b^w)$$

- Each a can be changed to b. And c to d.
- Everything stays linear; $2w$ degrees of freedom.

Linearization Attack

- Choose a, b, c, d. Start with $x = a^w$ and $x' = c^w$.

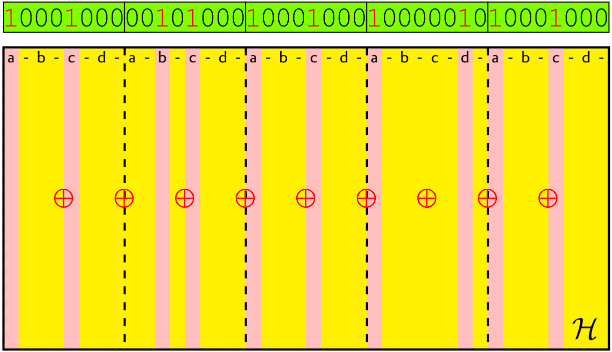


$$= F(a^w) \oplus F(b^w) \oplus K_{ab1}$$

- Each a can be changed to b. And c to d.
- Everything stays linear; $2w$ degrees of freedom.

Linearization Attack

- Choose a, b, c, d. Start with $x = a^w$ and $x' = c^w$.

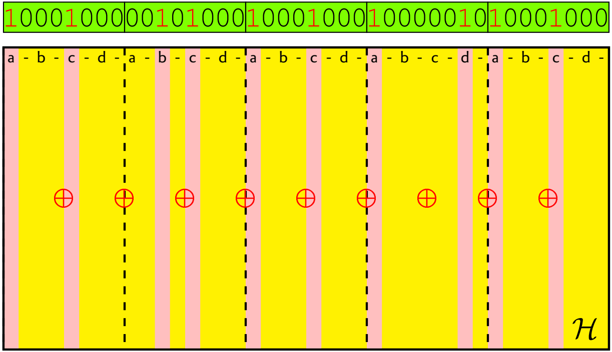


$$= F(a^w) \oplus F(b^w)$$
$$\oplus K_{ab1} \oplus K_{cd3}$$

- Each a can be changed to b. And c to d.
- Everything stays linear; $2w$ degrees of freedom.

Linearization Attack

- Choose a, b, c, d. Start with $x = a^w$ and $x' = c^w$.



$$= F(a^w) \oplus F(b^w) \oplus K_{ab1} \oplus K_{cd3}$$

- Each a can be changed to b. And c to d.
- Everything stays linear; $2w$ degrees of freedom.

Linearization attack against IFSB

- We reduce the message space: $x \in \{ab\}^*, x' \in \{cd\}^*$.
- We search a vector in the kernel of a $r \times 2w$ matrix.
- If $r \leq 2w$ we expect to find one.
- This is the case for IFSB...

Linearization attack against IFSB

- We reduce the message space: $x \in \{ab\}^*, x' \in \{cd\}^*$.
- We search a vector in the kernel of a $r \times 2w$ matrix.
- If $r \leq 2w$ we expect to find one.
- This is the case for IFSB...

State of the art

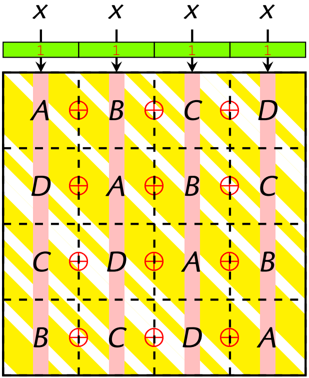
- The original FSB needs a huge matrix and is slow
- The parameters of IFSB are bad
- No structural attack against IFSB
- No attack known if r/w is big enough

Outline

Cyclic code and periodicity

Property

If $\mathcal{H}$ is cyclic and $\varphi(M)$ is periodic, then $\mathcal{H} \times \varphi(M)$ is periodic.



$$= A_x \oplus B_x \oplus C_x \oplus D_x$$

$$= A_x \oplus B_x \oplus C_x \oplus D_x$$

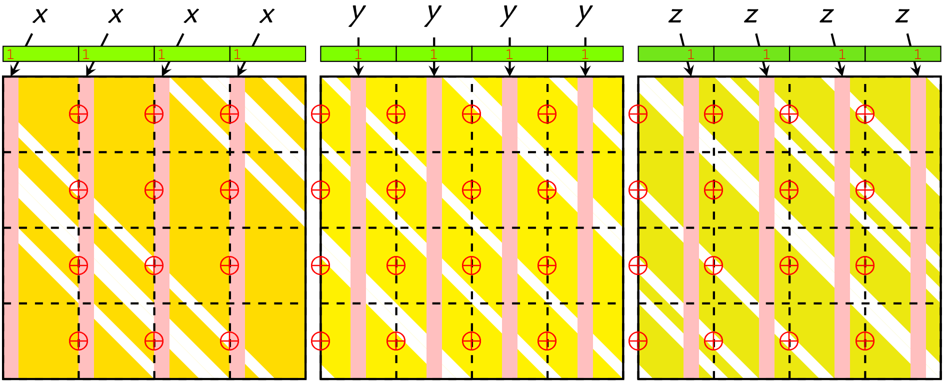
$$= A_x \oplus B_x \oplus C_x \oplus D_x$$

$$= A_x \oplus B_x \oplus C_x \oplus D_x$$

Quasi-cyclic code and periodicity

Property

If $\mathcal{H}$ is quasi-cyclic and $\varphi(M)$ is piecewise-periodic, then $\mathcal{H} \times \varphi(M)$ is periodic.



The Periodic attack

Basic idea of the attack

- Use a piecewise-periodic $\varphi(M)$
- Cancel one period of the output.

- We can use Wagner's attack to cancel one period.
- a' is a or $a-1$, but smaller matrix:

Attack	Complexity	Remarks
Wagner	$r2^{a'} \cdot 2^{r/(a+1)}$	r is typically 1024
Cyclic + Wagner	$\frac{n}{2w}2^{a'} \cdot 2^{\frac{n}{2w}/(a'+1)}$	$n/4w$ is typically 128

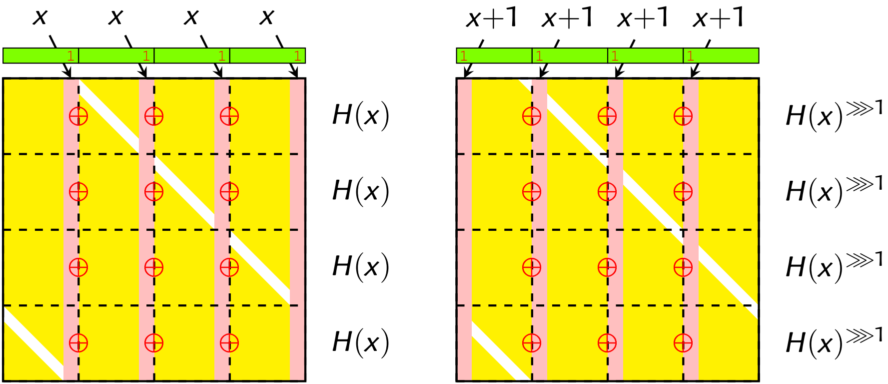
The Cyclic attack

- Is there some more structure that we can use?
- Yes: for each cyclic block, the outputs are still related:

- Collision iff $\bigoplus_{i=0}^{p-1} H_i \lll \mu_i = 0$

The Cyclic attack

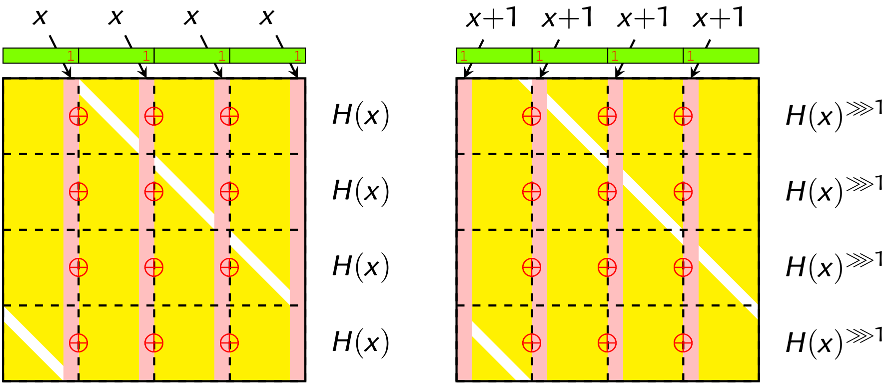
- Is there some more structure that we can use?
- Yes: for each cyclic block, the outputs are still related:



- Collision iff $\bigoplus_{i=0}^{p-1} H_i \lll \mu_i = 0$

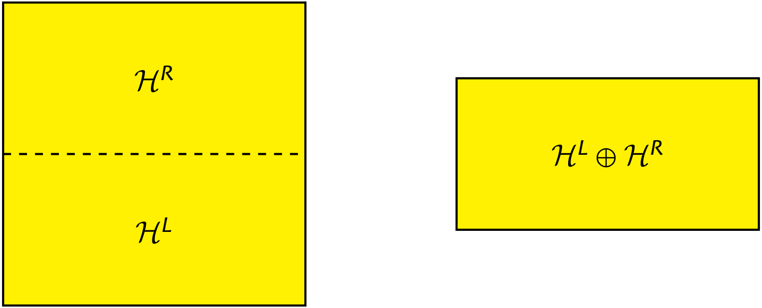
The Cyclic attack

- Is there some more structure that we can use?
- Yes: for each cyclic block, the outputs are still related:



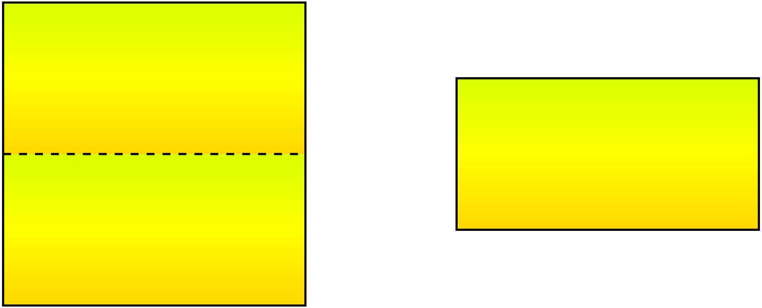
- Collision iff $\bigoplus_{i=0}^{p-1} H_i \lll \mu_i = 0$

Solve $\bigoplus_{i=0}^{p-1} H_i \lll \mu_i = 0$



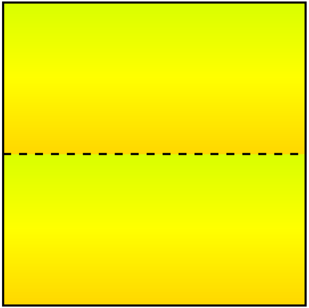
- $\mathcal{H} = \mathcal{H}^L || \mathcal{H}^R$
- Solve for $\mathcal{H}^L \oplus \mathcal{H}^R$
- Apply the rotation to $\mathcal{H}$
- The MSB of μ_i exchanges H_i^L and H_i^R
- Linear system for $\bigoplus H_i^L \lll \mu_i = 0$
- Then $\bigoplus H_i^R \lll \mu_i = 0$

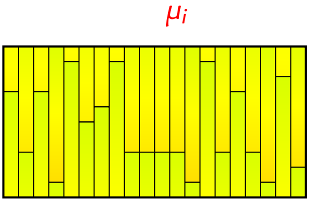
Solve $\bigoplus_{i=0}^{p-1} H_i \lll \mu_i = 0$



- $\mathcal{H} = \mathcal{H}^L || \mathcal{H}^R$
- Solve for $\mathcal{H}^L \oplus \mathcal{H}^R$
- Apply the rotation to $\mathcal{H}$
- The MSB of μ_i exchanges H_i^L and H_i^R
- Linear system for $\bigoplus H_i^L \lll \mu_i = 0$
- Then $\bigoplus H_i^R \lll \mu_i = 0$

Solve $\bigoplus_{i=0}^{p-1} H_i \lll \mu_i = 0$



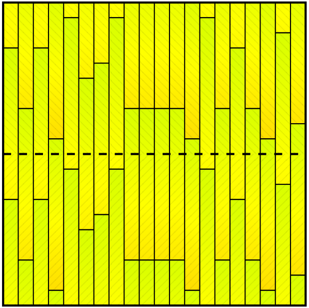


$= 0$

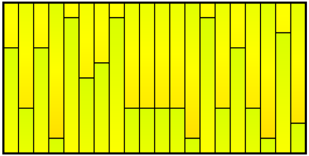
- $\mathcal{H} = \mathcal{H}^L || \mathcal{H}^R$
- Solve for $\mathcal{H}^L \oplus \mathcal{H}^R$
- Apply the rotation to $\mathcal{H}$
- The MSB of μ_i exchanges H_i^L and H_i^R
- Linear system for $\bigoplus H_i^L \lll \mu_i = 0$
- Then $\bigoplus H_i^R \lll \mu_i = 0$

Solve $\bigoplus_{i=0}^{p-1} H_i \lll \mu_i = 0$

μ_i



μ_i

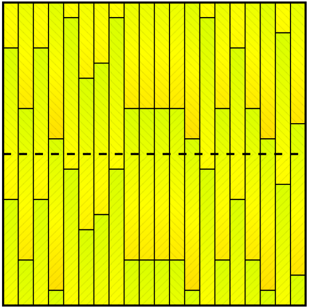


$= 0$

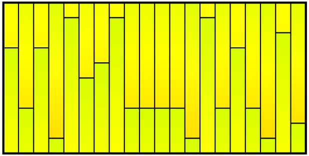
- $\mathcal{H} = \mathcal{H}^L || \mathcal{H}^R$
- Solve for $\mathcal{H}^L \oplus \mathcal{H}^R$
- Apply the rotation to $\mathcal{H}$
- The MSB of μ_i exchanges H_i^L and H_i^R
- Linear system for $\bigoplus H_i^L \lll \mu_i = 0$
- Then $\bigoplus H_i^R \lll \mu_i = 0$

Solve $\bigoplus_{i=0}^{p-1} H_i \lll \mu_i = 0$

μ_i



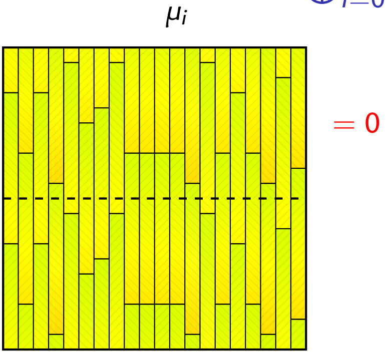
μ_i

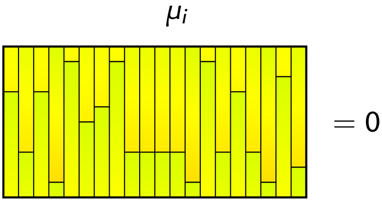


$= 0$

- $\mathcal{H} = \mathcal{H}^L || \mathcal{H}^R$
- Solve for $\mathcal{H}^L \oplus \mathcal{H}^R$
- Apply the rotation to $\mathcal{H}$
- The MSB of μ_i exchanges H_i^L and H_i^R
- Linear system for $\bigoplus H_i^L \lll \mu_i = 0$
- Then $\bigoplus H_i^R \lll \mu_i = 0$

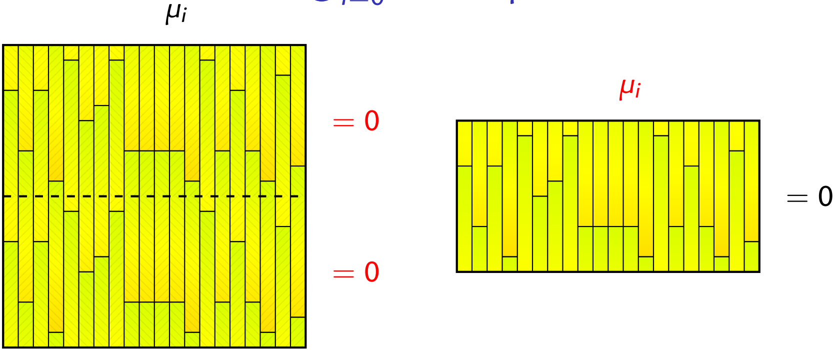
Solve $\bigoplus_{i=0}^{p-1} H_i \lll \mu_i = 0$





- $\mathcal{H} = \mathcal{H}^L || \mathcal{H}^R$
- Solve for $\mathcal{H}^L \oplus \mathcal{H}^R$
- Apply the rotation to $\mathcal{H}$
- The MSB of μ_i exchanges H_i^L and H_i^R
- Linear system for $\bigoplus H_i^L \lll \mu_i = 0$
- Then $\bigoplus H_i^R \lll \mu_i = 0$

Solve $\bigoplus_{i=0}^{p-1} H_i \lll \mu_i = 0$



- $\mathcal{H} = \mathcal{H}^L || \mathcal{H}^R$
- Solve for $\mathcal{H}^L \oplus \mathcal{H}^R$
- Apply the rotation to $\mathcal{H}$
- The MSB of μ_i exchanges H_i^L and H_i^R
- Linear system for $\bigoplus H_i^L \lll \mu_i = 0$
- Then $\bigoplus H_i^R \lll \mu_i = 0$

Overview

The full attack looks like periodic + linearization.

Linearization attack

Conditions	Complexity	Remarks
$r \leq 2w$	r^3	r is typically 1024
if r is bigger	$(4/3)^{r-2w} \cdot r^3$	$\log_2(4/3) \approx 0.415$

Cyclic attack

Conditions	Complexity	Remarks
$r \leq 4w$	$(n/4w)^3$	$n/4w$ is typically 64
if r is bigger	$2^{\frac{n(r-4w)}{4wr}} \cdot (n/4w)^3$	$n/4wr$ is typically $1/16$

Scope of the Attack

In this talk, we assume that all the parameters are powers of 2.

- We need a small divisor d of r for the periodic attack.
- d must be a power of 2 for the cyclic attack.

Actually one of the parameter set of IFSB uses a prime r...

This is due to a result about quasi-cyclic codes:

Theorem

If r is a prime such that 2 is primitive modulo n,
Then the matrix generated by a word of odd weight is invertible,
and the code has the same kind of properties than a random code.

This does not prove the security of IFSB with a good r...

Our attack complements this result:
if r has a small divisor, it is easy to invert periodic syndromes.

Scope of the Attack

In this talk, we assume that all the parameters are powers of 2.

- We need a small divisor d of r for the periodic attack.
- d must be a power of 2 for the cyclic attack.

Actually one of the parameter set of IFSB uses a prime r...

This is due to a result about quasi-cyclic codes:

Theorem

If r is a prime such that 2 is primitive modulo n,
Then the matrix generated by a word of odd weight is invertible,
and the code has the same kind of properties than a random code.

This does not prove the security of IFSB with a good r...

Our attack complements this result:
if r has a small divisor, it is easy to invert periodic syndromes.

Scope of the Attack

In this talk, we assume that all the parameters are powers of 2.

- We need a small divisor d of r for the periodic attack.
- d must be a power of 2 for the cyclic attack.

Actually one of the parameter set of IFSB uses a prime r...

This is due to a result about quasi-cyclic codes:

Theorem

If r is a prime such that 2 is primitive modulo n,
Then the matrix generated by a word of odd weight is invertible,
and the code has the same kind of properties than a random code.

This does not prove the security of IFSB with a good r...

Our attack complements this result:
if r has a small divisor, it is easy to invert periodic syndromes.

Scope of the Attack

In this talk, we assume that all the parameters are powers of 2.

- We need a small divisor d of r for the periodic attack.
- d must be a power of 2 for the cyclic attack.

Actually one of the parameter set of IFSB uses a prime r...

This is due to a result about quasi-cyclic codes:

Theorem

If r is a prime such that 2 is primitive modulo n,
Then the matrix generated by a word of odd weight is invertible,
and the code has the same kind of properties than a random code.

This does not prove the security of IFSB with a good r...

Our attack complements this result:
if r has a small divisor, it is easy to invert periodic syndromes.

On Provable Security

Regular Syndrome Decoding is NP-hard but...

- There is an efficient algorithm for small matrix: *Wagner attack*.
- It is easy when $r \leq 2w$: *linearization attack*.

Quasi-Cyclic Regular Syndrome Decoding is hard but...

- For some parameters, it is easy to decode a periodic syndrome: *cyclic attack*.

On Provable Security

Regular Syndrome Decoding is NP-hard but...

- There is an efficient algorithm for small matrix: *Wagner attack*.
- It is easy when $r \leq 2w$: *linearization attack*.

Quasi-Cyclic Regular Syndrome Decoding is hard but...

- For some parameters, it is easy to decode a periodic syndrome: *cyclic attack*.

On Provable Security

Regular Syndrome Decoding is NP-hard but...

- There is an efficient algorithm for small matrix: *Wagner attack*.
- It is easy when $r \leq 2w$: *linearization attack*.

Quasi-Cyclic Regular Syndrome Decoding is hard but...

- For some parameters, it is easy to decode a periodic syndrome: *cyclic attack*.

On Provable Security

Regular Syndrome Decoding is NP-hard but...

- There is an efficient algorithm for small matrix: *Wagner attack*.
- It is easy when $r \leq 2w$: *linearization attack*.

Quasi-Cyclic Regular Syndrome Decoding is hard but...

- For some parameters, it is easy to decode a periodic syndrome: *cyclic attack*.

On Provable Security

Regular Syndrome Decoding is NP-hard but...

- There is an efficient algorithm for small matrix: *Wagner attack*.
- It is easy when $r \leq 2w$: *linearization attack*.

Quasi-Cyclic Regular Syndrome Decoding is hard but...

- For some parameters, it is easy to decode a periodic syndrome: *cyclic attack*.

IFSB Status

- The original FSB needs a huge matrix and is slow
- The parameters of IFSB are really bad
- Structural attack against IFSB with a bad r
- No attack known if r is carefully chosen and r/w is big enough

Any Questions?

Thank you for your attention.