CASA
CYBER SECURITY IN THE AGE OF LARGE-SCALE ADVERSARIES

Simon's Algorithm and Symmetric Crypto: Generalizations and Automatized Applications

CRYPTO 2022 – Santa Barbara, August 17, 2022

Federico Canale, Gregor Leander, Lukas Stennes
Ruhr-Universität Bochum

RUHR UNIVERSITÄT BOCHUM RUB

Gefördert durch DFG Deutsche Forschungsgemeinschaft

HGI HORST GÖRTZ INSTITUT

Simon's Problem and Algorithm

CASA
Cyber Security in the Age of Large-Scale Adversaries

- Given a function $f : \mathbb{F}_2^n \to \mathbb{F}_2^n$
- Simon's promise: $f(x) = f(x \oplus s)$
- Goal: find period s
- Classical: $\Theta(2^{n/2})$ queries to f (birthday problem)
- Quantum: $O(n)$ quantum queries and basic linear algebra give s

Simon's Problem and Algorithm

CASA
CYBER SECURITY IN THE AGE OF LARGE-SCALE ADVERSARIES

- Given a function $f : \mathbb{F}_2^n \to \mathbb{F}_2^n$
- Simon's promise: $f(x) = f(x \oplus s)$
- Goal: find period s
- Classical: $\Theta(2^{n/2})$ queries to f (birthday problem)
- Quantum: $O(n)$ quantum queries and basic linear algebra give s

Simon's Problem and Algorithm

CASA
CYBER SECURITY IN THE AGE OF LARGE-SCALE ADVERSARIES

- Given a function $f : \mathbb{F}_2^n \to \mathbb{F}_2^n$
- Simon's promise: $f(x) = f(x \oplus s)$
- Goal: find period s
- Classical: $\Theta(2^{n/2})$ queries to f (birthday problem)
- Quantum: $O(n)$ quantum queries and basic linear algebra give s

Simon's Problem and Algorithm

CASA
CYBER SECURITY IN THE AGE OF LARGE-SCALE ADVERSARIES

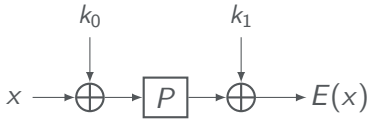
- Given a function $f : \mathbb{F}_2^n \to \mathbb{F}_2^n$
- Simon's promise: $f(x) = f(x \oplus s)$
- Goal: find period s
- Classical: $\Theta(2^{n/2})$ queries to f (birthday problem)
- Quantum: $O(n)$ quantum queries and basic linear algebra give s

Simon's Problem and Algorithm

CASA
CYBER SECURITY IN THE AGE OF LARGE-SCALE ADVERSARIES

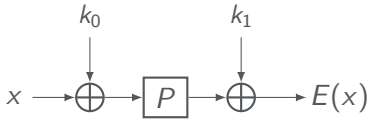
- Given a function $f : \mathbb{F}_2^n \to \mathbb{F}_2^n$
- Simon's promise: $f(x) = f(x \oplus s)$
- Goal: find period s
- Classical: $\Theta(2^{n/2})$ queries to f (birthday problem)
- Quantum: $O(n)$ quantum queries and basic linear algebra give s

Attack on Even-Mansour [Kuwakado and Morii 2012]



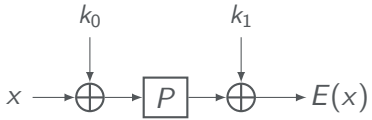
- $E(x) = P(x \oplus k_0) \oplus k_1$
- $f(x) = E(x) \oplus P(x) = P(x \oplus k_0) \oplus P(x) \oplus k_1$
- $f(x) = f(x \oplus k_0) \Rightarrow$ Simon's algorithm finds k_0
- Finding k_1 is easy: $k_1 = E(x) \oplus P(x \oplus k_0)$

Attack on Even-Mansour [Kuwakado and Morii 2012]



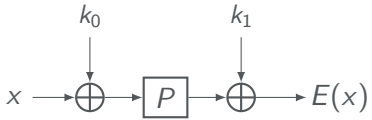
- $E(x) = P(x \oplus k_0) \oplus k_1$
- $f(x) = E(x) \oplus P(x) = P(x \oplus k_0) \oplus P(x) \oplus k_1$
- $f(x) = f(x \oplus k_0) \Rightarrow$ Simon's algorithm finds k_0
- Finding k_1 is easy: $k_1 = E(x) \oplus P(x \oplus k_0)$

Attack on Even-Mansour [Kuwakado and Morii 2012]



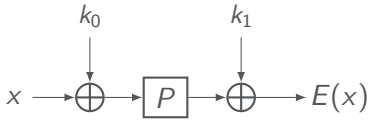
- $E(x) = P(x \oplus k_0) \oplus k_1$
- $f(x) = E(x) \oplus P(x) = P(x \oplus k_0) \oplus P(x) \oplus k_1$
- $f(x) = f(x \oplus k_0) \Rightarrow$ Simon's algorithm finds k_0
- Finding k_1 is easy: $k_1 = E(x) \oplus P(x \oplus k_0)$

Attack on Even-Mansour [Kuwakado and Morii 2012]



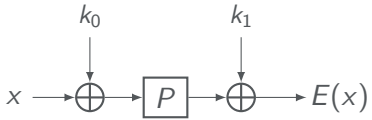
- $E(x) = P(x \oplus k_0) \oplus k_1$
- $f(x) = E(x) \oplus P(x) = P(x \oplus k_0) \oplus P(x) \oplus k_1$
- $f(x) = f(x \oplus k_0) \Rightarrow$ Simon's algorithm finds k_0
- Finding k_1 is easy: $k_1 = E(x) \oplus P(x \oplus k_0)$

Attack on Even-Mansour [Kuwakado and Morii 2012]



- $E(x) = P(x \oplus k_0) \oplus k_1$
- $f(x) = E(x) \oplus P(x) = P(x \oplus k_0) \oplus P(x) \oplus k_1$
- $f(x) = f(x \oplus k_0) \Rightarrow$ Simon's algorithm finds k_0
- Finding k_1 is easy: $k_1 = E(x) \oplus P(x \oplus k_0)$

Attack on Even-Mansour [Kuwakado and Morii 2012]



- $E(x) = P(x \oplus k_0) \oplus k_1$
- $f(x) = E(x) \oplus P(x) = P(x \oplus k_0) \oplus P(x) \oplus k_1$
- $f(x) = f(x \oplus k_0) \Rightarrow$ Simon's algorithm finds k_0
- Finding k_1 is easy: $k_1 = E(x) \oplus P(x \oplus k_0)$

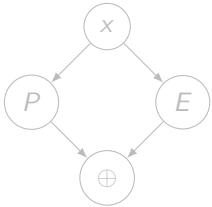
1. Motivation
2. Automatized Applications
3. Generalization
4. Discussion

Automatized Applications

Automatized Applications

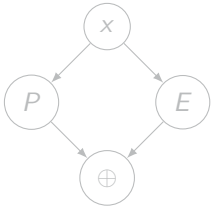
CASA
Cyber Security in the Age of Large-Scale Adversaries

- Periodic function $\overset{Simon}{\Rightarrow}$ (fast) quantum attack
- Searching periodic functions is cumbersome
- How can we automatize it?
  - Enumerate all *sensible* functions
  - Instantiate with small block size to check for periodicity
- For Even-Mansour: $f(x) = E(x) \oplus P(x)$



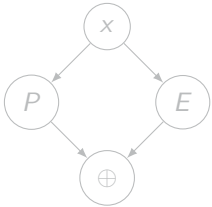
Automatized Applications

- Periodic function $\overset{Simon}{\Rightarrow}$ (fast) quantum attack
- Searching periodic functions is cumbersome
- How can we automatize it?
  - Enumerate all *sensible* functions
  - Instantiate with small block size to check for periodicity
- For Even-Mansour: $f(x) = E(x) \oplus P(x)$



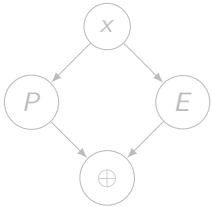
Automatized Applications

- Periodic function $\overset{Simon}{\Rightarrow}$ (fast) quantum attack
- Searching periodic functions is cumbersome
- How can we automatize it?
  - Enumerate all *sensible* functions
  - Instantiate with small block size to check for periodicity
- For Even-Mansour: $f(x) = E(x) \oplus P(x)$



Automatized Applications

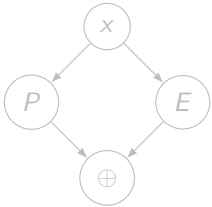
- Periodic function $\overset{Simon}{\Rightarrow}$ (fast) quantum attack
- Searching periodic functions is cumbersome
- How can we automatize it?
  - Enumerate all *sensible* functions
  - Instantiate with small block size to check for periodicity
- For Even-Mansour: $f(x) = E(x) \oplus P(x)$



Automatized Applications

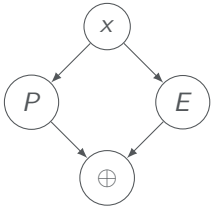
CASA
Cyber Security in the Age of Large-Scale Adversaries

- Periodic function $\overset{Simon}{\Rightarrow}$ (fast) quantum attack
- Searching periodic functions is cumbersome
- How can we automatize it?
  - Enumerate all *sensible* functions
  - Instantiate with small block size to check for periodicity
- For Even-Mansour: $f(x) = E(x) \oplus P(x)$



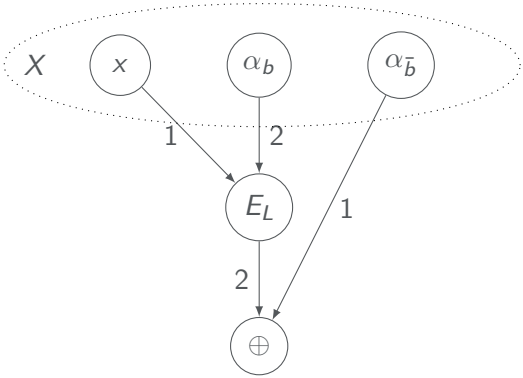
Automatized Applications

- Periodic function $\overset{Simon}{\Rightarrow}$ (fast) quantum attack
- Searching periodic functions is cumbersome
- How can we automatize it?
  - Enumerate all *sensible* functions
  - Instantiate with small block size to check for periodicity
- For Even-Mansour: $f(x) = E(x) \oplus P(x)$



Circuits

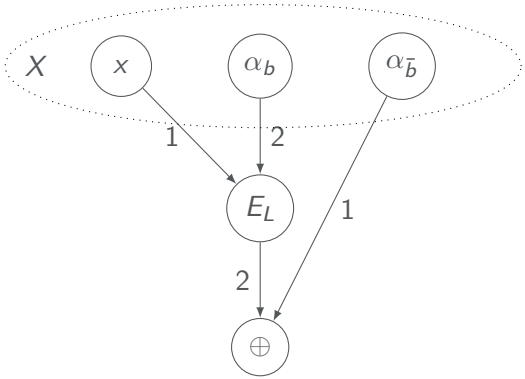
- A circuit C is defined by
  - A DAG $D = (V, E)$
  - A set of input nodes $X \subset V$
  - A set of gate functions $\mathscr{G}$ where $G_i : \mathbb{F}_2^n \times \mathbb{F}_2^n \to \mathbb{F}_2^n$
  - An output node $v_{out} \in V$



Attack on 3-round Feistel [KM 2010]

Circuits

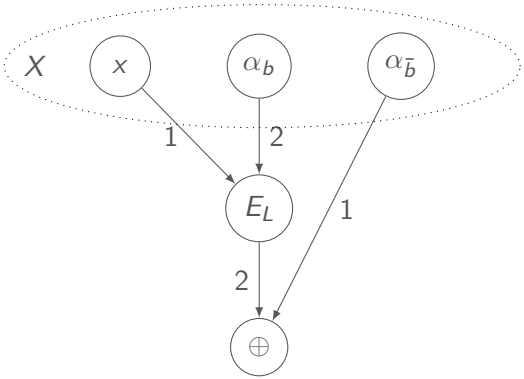
- A circuit C is defined by
  - A DAG $D = (V, E)$
  - A set of input nodes $X \subset V$
  - A set of gate functions $\mathcal{G}$ where $G_i : \mathbb{F}_2^n \times \mathbb{F}_2^n \to \mathbb{F}_2^n$
  - An output node $v_{out} \in V$



Attack on 3-round Feistel [KM 2010]

Circuits

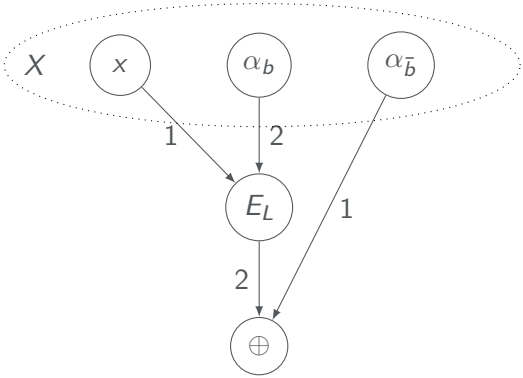
- A circuit C is defined by
  - A DAG $D = (V, E)$
  - A set of input nodes $X \subset V$
  - A set of gate functions $\mathscr{G}$ where $G_i : \mathbb{F}_2^n \times \mathbb{F}_2^n \to \mathbb{F}_2^n$
  - An output node $v_{out} \in V$



Attack on 3-round Feistel [KM 2010]

Circuits

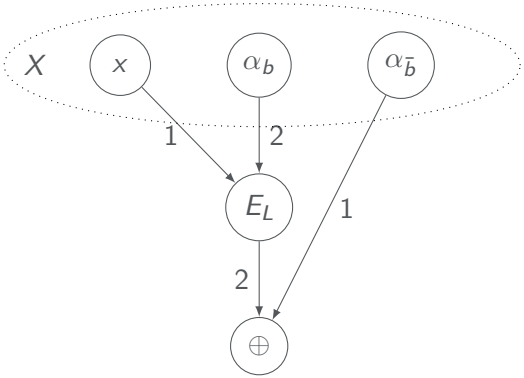
- A circuit C is defined by
  - A DAG $D = (V, E)$
  - A set of input nodes $X \subset V$
  - A set of gate functions $\mathscr{G}$ where $G_i : \mathbb{F}_2^n \times \mathbb{F}_2^n \to \mathbb{F}_2^n$
  - An output node $v_{out} \in V$



Attack on 3-round Feistel [KM 2010]

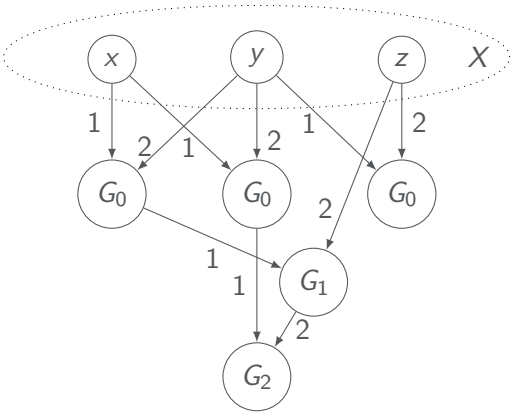
Circuits

- A circuit C is defined by
  - A DAG $D = (V, E)$
  - A set of input nodes $X \subset V$
  - A set of gate functions $\mathscr{G}$ where $G_i : \mathbb{F}_2^n \times \mathbb{F}_2^n \to \mathbb{F}_2^n$
  - An output node $v_{out} \in V$



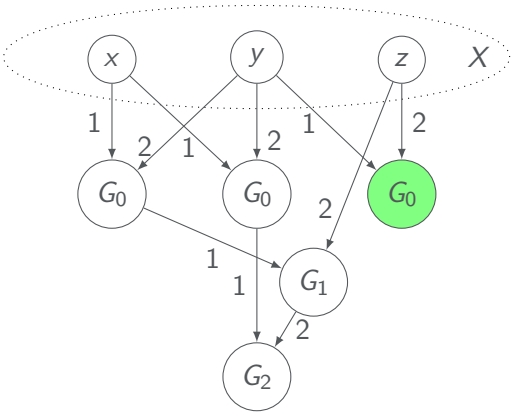
Attack on 3-round Feistel [KM 2010]

Normal Circuits



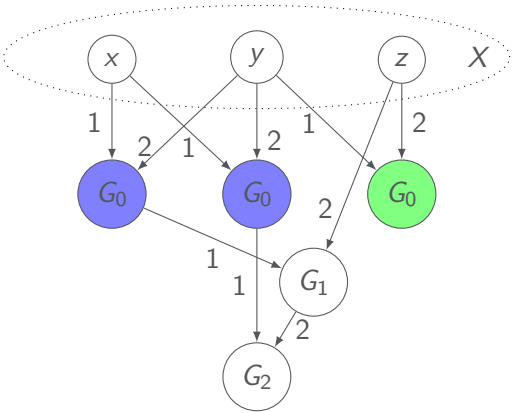
(a) Circuit in non-normal form.

Normal Circuits



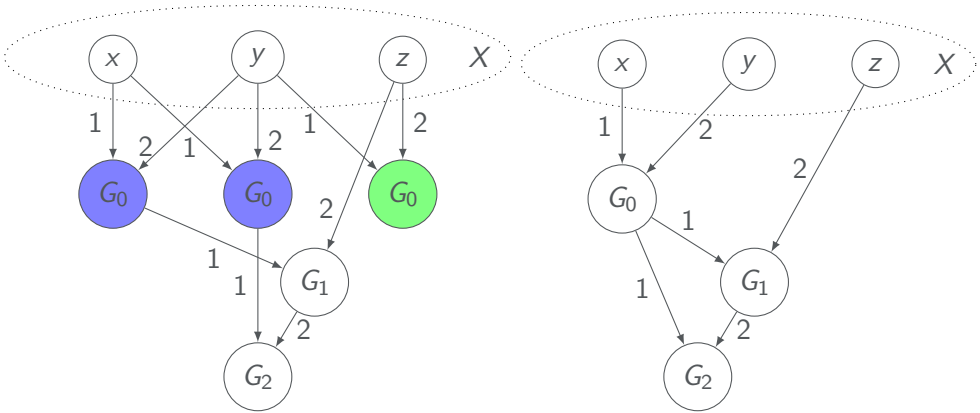
(a) Circuit in non-normal form.

Normal Circuits



(a) Circuit in non-normal form.

Normal Circuits



(a) Circuit in non-normal form.

(b) Circuit in normal form.

More Restrictions

CASA
Cyber Security in the Age of Large-Scale Adversaries

- Define rules $r\colon \mathscr{C} \to \{0,1\}$
- Examples for such rules
  - Exclude circuits that compute $a \oplus a$
  - Restrict number of oracle queries
  - Query must depend on non constant input
  - ...
- Check must be efficient
- Test also on partial circuits

More Restrictions

CASA
Cyber Security in the Age of Large-Scale Adversaries

- Define rules $r\colon \mathscr{C} \to \{0,1\}$
- Examples for such rules
  - Exclude circuits that compute $a \oplus a$
  - Restrict number of oracle queries
  - Query must depend on non constant input
  - ...
- Check must be efficient
- Test also on partial circuits

More Restrictions

CASA
Cyber Security in the Age of Large-Scale Adversaries

- Define rules $r\colon \mathscr{C} \to \{0,1\}$
- Examples for such rules
  - Exclude circuits that compute $a \oplus a$
  - Restrict number of oracle queries
  - Query must depend on non constant input
  - ...
- Check must be efficient
- Test also on partial circuits

More Restrictions

CASA
Cyber Security in the Age of Large-Scale Adversaries

- Define rules $r\colon \mathscr{C} \to \{0,1\}$
- Examples for such rules
  - Exclude circuits that compute $a \oplus a$
  - Restrict number of oracle queries
  - Query must depend on non constant input
  - ...
- Check must be efficient
- Test also on partial circuits

More Restrictions

CASA
Cyber Security in the Age of Large-Scale Adversaries

- Define rules $r\colon \mathscr{C} \to \{0,1\}$
- Examples for such rules
  - Exclude circuits that compute $a \oplus a$
  - Restrict number of oracle queries
  - Query must depend on non constant input
  - ...
- Check must be efficient
- Test also on partial circuits

More Restrictions

CASA
Cyber Security in the Age of Large-Scale Adversaries

- Define rules $r\colon \mathscr{C} \to \{0,1\}$
- Examples for such rules
  - Exclude circuits that compute $a \oplus a$
  - Restrict number of oracle queries
  - Query must depend on non constant input
  - ...
- Check must be efficient
- Test also on partial circuits

More Restrictions

CASA
Cyber Security in the Age of Large-Scale Adversaries

- Define rules $r\colon \mathscr{C} \to \{0,1\}$
- Examples for such rules
  - Exclude circuits that compute $a \oplus a$
  - Restrict number of oracle queries
  - Query must depend on non constant input
  - ...
- Check must be efficient
- Test also on partial circuits

More Restrictions

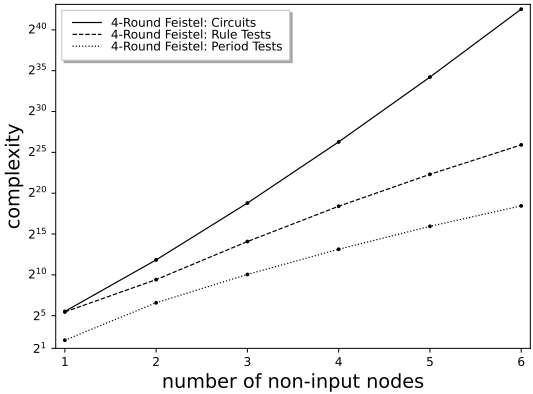
CASA
Cyber Security in the Age of Large-Scale Adversaries

- Define rules $r\colon \mathscr{C} \to \{0,1\}$
- Examples for such rules
  - Exclude circuits that compute $a \oplus a$
  - Restrict number of oracle queries
  - Query must depend on non constant input
  - ...
- Check must be efficient
- Test also on partial circuits

More Restrictions

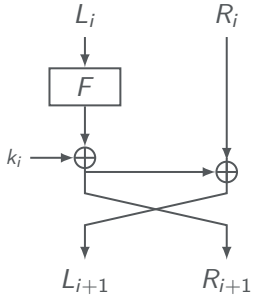
CASA
Cyber Security in the Age of Large-Scale Adversaries

- Define rules $r\colon \mathscr{C} \to \{0,1\}$
- Examples for such rules
  - Exclude circuits that compute $a \oplus a$
  - Restrict number of oracle queries
  - Query must depend on non constant input
  - ...
- Check must be efficient
- Test also on partial circuits



Example: Search for 4-Round MISTY R-FK

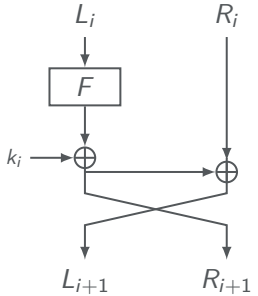
- Setup
  - Input nodes: $X = \{x\}$
  - Gates: $\mathscr{G} = \{\oplus, E_L, E_R, F, F^{-1}\}$
  - Rules
    - Normality
    - XORs must be sensible (e.g. no $a \oplus b \oplus a$, $a \oplus b$ vs $b \oplus a$)
    - No $F(F^{-1}(\cdot))$ and vice versa
    - F, F^{-1} only take single input
- Costs
  - Size of circuit tree: 2^{25}
  - Number of rule tests: 2^{18}
  - Number of periodicity tests: 2^{12}



MISTY R-FK

Example: Search for 4-Round MISTY R-FK

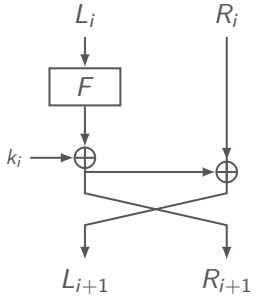
- Setup
  - Input nodes: $X = \{x\}$
  - Gates: $\mathscr{G} = \{\oplus, E_L, E_R, F, F^{-1}\}$
  - Rules
    - Normality
    - XORs must be sensible (e.g. no $a \oplus b \oplus a$, $a \oplus b$ vs $b \oplus a$)
    - No $F(F^{-1}(\cdot))$ and vice versa
    - F, F^{-1} only take single input
- Costs
  - Size of circuit tree: 2^{25}
  - Number of rule tests: 2^{18}
  - Number of periodicity tests: 2^{12}



MISTY R-FK

Example: Search for 4-Round MISTY R-FK

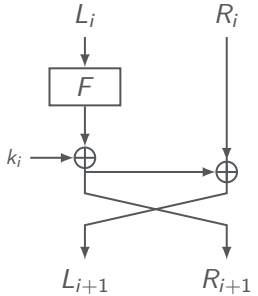
- Setup
  - Input nodes: $X = \{x\}$
  - Gates: $\mathscr{G} = \{\oplus, E_L, E_R, F, F^{-1}\}$
  - Rules
    - Normality
    - XORs must be sensible (e.g. no $a \oplus b \oplus a$, $a \oplus b$ vs $b \oplus a$)
    - No $F(F^{-1}(\cdot))$ and vice versa
    - F, F^{-1} only take single input
- Costs
  - Size of circuit tree: 2^{25}
  - Number of rule tests: 2^{18}
  - Number of periodicity tests: 2^{12}



MISTY R-FK

Example: Search for 4-Round MISTY R-FK

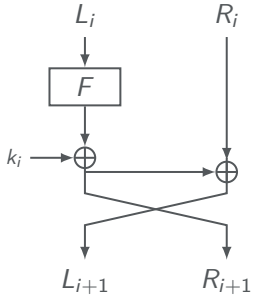
- Setup
  - Input nodes: $X = \{x\}$
  - Gates: $\mathscr{G} = \{\oplus, E_L, E_R, F, F^{-1}\}$
  - Rules
    - Normality
    - XORs must be sensible (e.g. no $a \oplus b \oplus a$, $a \oplus b$ vs $b \oplus a$)
    - No $F(F^{-1}(\cdot))$ and vice versa
    - F, F^{-1} only take single input
- Costs
  - Size of circuit tree: 2^{25}
  - Number of rule tests: 2^{18}
  - Number of periodicity tests: 2^{12}



MISTY R-FK

Example: Search for 4-Round MISTY R-FK

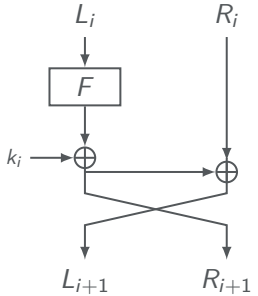
- Setup
  - Input nodes: $X = \{x\}$
  - Gates: $\mathscr{G} = \{\oplus, E_L, E_R, F, F^{-1}\}$
  - Rules
    - Normality
    - XORs must be sensible (e.g. no $a \oplus b \oplus a$, $a \oplus b$ vs $b \oplus a$)
    - No $F(F^{-1}(\cdot))$ and vice versa
    - F, F^{-1} only take single input
- Costs
  - Size of circuit tree: 2^{25}
  - Number of rule tests: 2^{18}
  - Number of periodicity tests: 2^{12}



MISTY R-FK

Example: Search for 4-Round MISTY R-FK

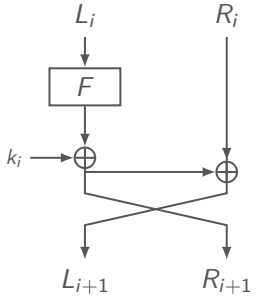
- Setup
  - Input nodes: $X = \{x\}$
  - Gates: $\mathscr{G} = \{\oplus, E_L, E_R, F, F^{-1}\}$
  - Rules
    - Normality
    - XORs must be sensible (e.g. no $a \oplus b \oplus a$, $a \oplus b$ vs $b \oplus a$)
    - No $F(F^{-1}(\cdot))$ and vice versa
    - F, F^{-1} only take single input
- Costs
  - Size of circuit tree: 2^{25}
  - Number of rule tests: 2^{18}
  - Number of periodicity tests: 2^{12}



MISTY R-FK

Example: Search for 4-Round MISTY R-FK

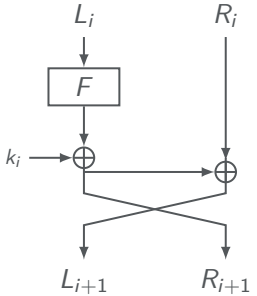
- Setup
  - Input nodes: $X = \{x\}$
  - Gates: $\mathscr{G} = \{\oplus, E_L, E_R, F, F^{-1}\}$
  - Rules
    - Normality
    - XORs must be sensible (e.g. no $a \oplus b \oplus a$, $a \oplus b$ vs $b \oplus a$)
    - No $F(F^{-1}(\cdot))$ and vice versa
    - F, F^{-1} only take single input
- Costs
  - Size of circuit tree: 2^{25}
  - Number of rule tests: 2^{18}
  - Number of periodicity tests: 2^{12}



MISTY R-FK

Example: Search for 4-Round MISTY R-FK

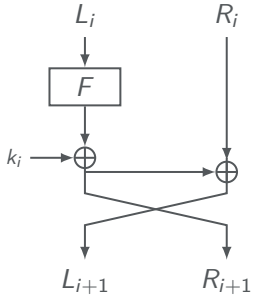
- Setup
  - Input nodes: $X = \{x\}$
  - Gates: $\mathscr{G} = \{\oplus, E_L, E_R, F, F^{-1}\}$
  - Rules
    - Normality
    - XORs must be sensible (e.g. no $a \oplus b \oplus a$, $a \oplus b$ vs $b \oplus a$)
    - No $F(F^{-1}(\cdot))$ and vice versa
    - F, F^{-1} only take single input
- Costs
  - Size of circuit tree: 2^{25}
  - Number of rule tests: 2^{18}
  - Number of periodicity tests: 2^{12}



MISTY R-FK

Example: Search for 4-Round MISTY R-FK

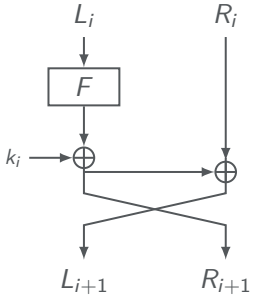
- Setup
  - Input nodes: $X = \{x\}$
  - Gates: $\mathscr{G} = \{\oplus, E_L, E_R, F, F^{-1}\}$
  - Rules
    - Normality
    - XORs must be sensible (e.g. no $a \oplus b \oplus a$, $a \oplus b$ vs $b \oplus a$)
    - No $F(F^{-1}(\cdot))$ and vice versa
    - F, F^{-1} only take single input
- Costs
  - Size of circuit tree: 2^{25}
  - Number of rule tests: 2^{18}
  - Number of periodicity tests: 2^{12}



MISTY R-FK

Example: Search for 4-Round MISTY R-FK

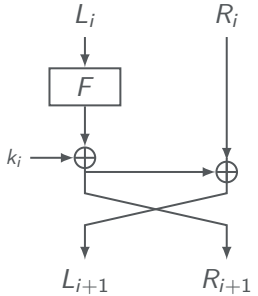
- Setup
  - Input nodes: $X = \{x\}$
  - Gates: $\mathscr{G} = \{\oplus, E_L, E_R, F, F^{-1}\}$
  - Rules
    - Normality
    - XORs must be sensible (e.g. no $a \oplus b \oplus a$, $a \oplus b$ vs $b \oplus a$)
    - No $F(F^{-1}(\cdot))$ and vice versa
    - F, F^{-1} only take single input
- Costs
  - Size of circuit tree: 2^{25}
  - Number of rule tests: 2^{18}
  - Number of periodicity tests: 2^{12}



MISTY R-FK

Example: Search for 4-Round MISTY R-FK

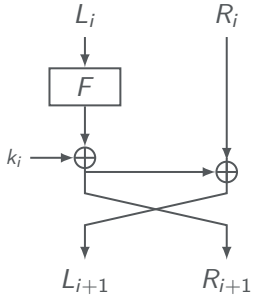
- Setup
  - Input nodes: $X = \{x\}$
  - Gates: $\mathscr{G} = \{\oplus, E_L, E_R, F, F^{-1}\}$
  - Rules
    - Normality
    - XORs must be sensible (e.g. no $a \oplus b \oplus a$, $a \oplus b$ vs $b \oplus a$)
    - No $F(F^{-1}(\cdot))$ and vice versa
    - F, F^{-1} only take single input
- Costs
  - Size of circuit tree: 2^{25}
  - Number of rule tests: 2^{18}
  - Number of periodicity tests: 2^{12}



MISTY R-FK

Example: Search for 4-Round MISTY R-FK

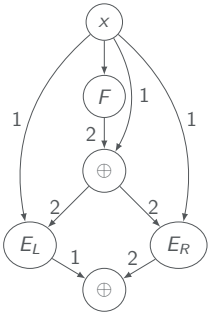
- Setup
  - Input nodes: $X = \{x\}$
  - Gates: $\mathscr{G} = \{\oplus, E_L, E_R, F, F^{-1}\}$
  - Rules
    - Normality
    - XORs must be sensible (e.g. no $a \oplus b \oplus a$, $a \oplus b$ vs $b \oplus a$)
    - No $F(F^{-1}(\cdot))$ and vice versa
    - F, F^{-1} only take single input
- Costs
  - Size of circuit tree: 2^{25}
  - Number of rule tests: 2^{18}
  - Number of periodicity tests: 2^{12}



MISTY R-FK

Example: Search for 4-Round MISTY R-FK

- Setup
  - Input nodes: $X = \{x\}$
  - Gates: $\mathscr{G} = \{\oplus, E_L, E_R, F, F^{-1}\}$
  - Rules
    - Normality
    - XORs must be sensible (e.g. no $a \oplus b \oplus a$, $a \oplus b$ vs $b \oplus a$)
    - No $F(F^{-1}(\cdot))$ and vice versa
    - F, F^{-1} only take single input
- Costs
  - Size of circuit tree: 2^{25}
  - Number of rule tests: 2^{18}
  - Number of periodicity tests: 2^{12}



- $f(x) = f(x \oplus k_0)$
- $O(n)$ quantum queries to break four rounds of MISTY R-FK

More New Polynomial Key Recoveries

CASA
Cyber Security in the Age of Large-Scale Adversaries

Construction	Prior Best Attack	Our Attack
4-round MISTY R-FK	distinguisher [1]	key recovery
5-round MISTY L-FK	distinguisher [1]	key recovery
4-round Feistel-FK	distinguisher [2]	key recovery
5-round Feistel-FK	distinguisher [2]	key recovery*

[1]: Cui, Guo, Ding: Applications of Simon's algorithm in quantum attacks on Feistel variants, Quantum Information Processing 2021
[2]: Ito, Hosoyamada, Matsumoto, Sasaki, Iwata: Quantum Chosen-Ciphertext Attacks Against Feistel Ciphers, CT-RSA 2019

*round function must be invertible

Generalization

Beyond period finding?

- Fourier transform of $f : \mathbb{F}_2^n \to \mathbb{F}_2$

$$\widehat{f}(x) = \sum_{y \in \mathbb{F}_2^n} (-1)^{x \cdot y + f(y)} = (H_0 \phi_f)_x, \qquad (\phi_f)_y = (-1)^{f(y)}$$

  where $(H_0)_{x,y} = (-1)^{x \cdot y}$ is the standard Hadamard matrix
- Bernstein-Vazirani ($\approx$ Simon) circuit yields

$$z \in \operatorname{span}\left\{ x \mid \widehat{f}(x) = (H_0 \phi_f)_x \neq 0 \right\}$$

- If f has unique period s, then $z \in \{x \mid x \cdot s = 0\}$.
- Generalized circuit yields

$$z \in \operatorname{span}\left\{ x \mid \widehat{f^H}(x) = (H \phi_f)_x \neq 0 \right\}$$

  where $(H)_{x,y} = (-1)^{g(x,y)}$ is any Hadamard matrix ($H^T H = I$)

Beyond period finding?

CASA
CYBER SECURITY IN THE AGE OF LARGE-SCALE ADVERSARIES

- Fourier transform of $f : \mathbb{F}_2^n \to \mathbb{F}_2$

$$\widehat{f}(x) = \sum_{y \in \mathbb{F}_2^n} (-1)^{x \cdot y + f(y)} = (H_0 \phi_f)_x, \qquad (\phi_f)_y = (-1)^{f(y)}$$

  where $(H_0)_{x,y} = (-1)^{x \cdot y}$ is the standard Hadamard matrix
- Bernstein-Vazirani ($\approx$ Simon) circuit yields

$$z \in \operatorname{span}\left\{ x \mid \widehat{f}(x) = (H_0 \phi_f)_x \neq 0 \right\}$$

- If f has unique period s, then $z \in \{x \mid x \cdot s = 0\}$.
- Generalized circuit yields

$$z \in \operatorname{span}\left\{ x \mid \widehat{f^H}(x) = (H \phi_f)_x \neq 0 \right\}$$

  where $(H)_{x,y} = (-1)^{g(x,y)}$ is any Hadamard matrix ($H^T H = I$)

Beyond period finding?

- Fourier transform of $f : \mathbb{F}_2^n \to \mathbb{F}_2$

$$\widehat{f}(x) = \sum_{y \in \mathbb{F}_2^n} (-1)^{x \cdot y + f(y)} = (H_0 \phi_f)_x, \qquad (\phi_f)_y = (-1)^{f(y)}$$

  where $(H_0)_{x,y} = (-1)^{x \cdot y}$ is the standard Hadamard matrix

- Bernstein-Vazirani ($\approx$ Simon) circuit yields

$$z \in \operatorname{span}\left\{ x \mid \widehat{f}(x) = (H_0 \phi_f)_x \neq 0 \right\}$$

- If f has unique period s, then $z \in \{x \mid x \cdot s = 0\}$.
- Generalized circuit yields

$$z \in \operatorname{span}\left\{ x \mid \widehat{f}^H(x) = (H \phi_f)_x \neq 0 \right\}$$

  where $(H)_{x,y} = (-1)^{g(x,y)}$ is any Hadamard matrix ($H^T H = I$)

Beyond period finding?

- Fourier transform of $f : \mathbb{F}_2^n \to \mathbb{F}_2$

$$\widehat{f}(x) = \sum_{y \in \mathbb{F}_2^n} (-1)^{x \cdot y + f(y)} = (H_0 \phi_f)_x, \qquad (\phi_f)_y = (-1)^{f(y)}$$

  where $(H_0)_{x,y} = (-1)^{x \cdot y}$ is the standard Hadamard matrix
- Bernstein-Vazirani ($\approx$ Simon) circuit yields

$$z \in \operatorname{span}\left\{ x \mid \widehat{f}(x) = (H_0 \phi_f)_x \neq 0 \right\}$$

- If f has unique period s, then $z \in \{x \mid x \cdot s = 0\}$.
- Generalized circuit yields

$$z \in \operatorname{span}\left\{ x \mid \widehat{f^H}(x) = (H \phi_f)_x \neq 0 \right\}$$

  where $(H)_{x,y} = (-1)^{g(x,y)}$ is any Hadamard matrix ($H^T H = I$)

Beyond period finding?

- Fourier transform of $f : \mathbb{F}_2^n \to \mathbb{F}_2$

$$\widehat{f}(x) = \sum_{y \in \mathbb{F}_2^n} (-1)^{x \cdot y + f(y)} = (H_0 \phi_f)_x, \qquad (\phi_f)_y = (-1)^{f(y)}$$

  where $(H_0)_{x,y} = (-1)^{x \cdot y}$ is the standard Hadamard matrix
- Bernstein-Vazirani ($\approx$ Simon) circuit yields

$$z \in \text{span}\left\{ x \mid \widehat{f}(x) = (H_0 \phi_f)_x \neq 0 \right\}$$

- If f has unique period s, then $z \in \{x \mid x \cdot s = 0\}$.
- Generalized circuit yields

$$z \in \text{span}\left\{ x \mid \widehat{f}^H(x) = (H \phi_f)_x \neq 0 \right\}$$

  where $(H)_{x,y} = (-1)^{g(x,y)}$ is any Hadamard matrix ($H^T H = I$)

Beyond period finding?

CASA
Cyber Security in the Age of Large-Scale Adversaries

- What meaningful properties of f for *non*-standard H?
- Linear invariant properties: linearity, differential uniformity, algebraic degree...
- A property $\mathcal{P}$ is linear invariant if for any $A \in \mathrm{GL}(n, \mathbb{F}_2)$

$$f \text{ satisfies } \mathcal{P} \text{ if and only if } f \circ A \text{ does}$$

Beyond period finding?

CASA
Cyber Security in the Age of Large-Scale Adversaries

- What meaningful properties of f for *non*-standard H?
- Linear invariant properties: linearity, differential uniformity, algebraic degree...
- A property $\mathcal{P}$ is linear invariant if for any $A \in \mathrm{GL}(n, \mathbb{F}_2)$

$$f \text{ satisfies } \mathcal{P} \text{ if and only if } f \circ A \text{ does}$$

Beyond period finding?

CASA
CYBER SECURITY IN THE AGE OF LARGE-SCALE ADVERSARIES

- What meaningful properties of f for *non*-standard H?
- Linear invariant properties: linearity, differential uniformity, algebraic degree...
- A property $\mathcal{P}$ is linear invariant if for any $A \in \mathrm{GL}(n, \mathbb{F}_2)$

$$f \text{ satisfies } \mathcal{P} \text{ if and only if } f \circ A \text{ does}$$

Theorem

If for any $A \in \mathrm{GL}(n, \mathbb{F}_2)$ there exists $B \in \mathrm{GL}(n, \mathbb{F}_2)$ such that for all f we have

$$\widehat{f \circ A}^H = \widehat{f}^H \circ B$$

then H is equivalent to H_0.

Discussion

Conclusion

CASA
Cyber Security in the Age of Large-Scale Adversaries

- Automatized Applications
  - Algorithm to automatically search for periodic functions
  - New attacks against MISTY and Feistel constructions
- Generalization
  - Generalizations of Simon's algorithm and Fourier transform
  - Standard Hadamard matrix seems to be the best choice
- Full version: `https://eprint.iacr.org/2022/782`
- Code: `https://github.com/rub-hgi/period-search`
- Contact: {federico.canale, gregor.leander, lukas.stennes }@rub.de

Conclusion

CASA
Cyber Security in the Age of Large-Scale Adversaries

- Automatized Applications
  - Algorithm to automatically search for periodic functions
  - New attacks against MISTY and Feistel constructions
- Generalization
  - Generalizations of Simon's algorithm and Fourier transform
  - Standard Hadamard matrix seems to be the best choice
- Full version: `https://eprint.iacr.org/2022/782`
- Code: `https://github.com/rub-hgi/period-search`
- Contact: {federico.canale, gregor.leander, lukas.stennes }@rub.de

Conclusion

CASA
Cyber Security in the Age of Large-Scale Adversaries

- Automatized Applications
  - Algorithm to automatically search for periodic functions
  - New attacks against MISTY and Feistel constructions
- Generalization
  - Generalizations of Simon's algorithm and Fourier transform
  - Standard Hadamard matrix seems to be the best choice
- Full version: `https://eprint.iacr.org/2022/782`
- Code: `https://github.com/rub-hgi/period-search`
- Contact: {federico.canale, gregor.leander, lukas.stennes }@rub.de

Conclusion

CASA
Cyber Security in the Age of Large-Scale Adversaries

- Automatized Applications
  - Algorithm to automatically search for periodic functions
  - New attacks against MISTY and Feistel constructions
- Generalization
  - Generalizations of Simon's algorithm and Fourier transform
  - Standard Hadamard matrix seems to be the best choice
- Full version: `https://eprint.iacr.org/2022/782`
- Code: `https://github.com/rub-hgi/period-search`
- Contact: {federico.canale, gregor.leander, lukas.stennes }@rub.de

Conclusion

CASA
Cyber Security in the Age of Large-Scale Adversaries

- Automatized Applications
  - Algorithm to automatically search for periodic functions
  - New attacks against MISTY and Feistel constructions
- Generalization
  - Generalizations of Simon's algorithm and Fourier transform
  - Standard Hadamard matrix seems to be the best choice
- Full version: `https://eprint.iacr.org/2022/782`
- Code: `https://github.com/rub-hgi/period-search`
- Contact: {federico.canale, gregor.leander, lukas.stennes }@rub.de

Conclusion

CASA
Cyber Security in the Age of Large-Scale Adversaries

- Automatized Applications
  - Algorithm to automatically search for periodic functions
  - New attacks against MISTY and Feistel constructions
- Generalization
  - Generalizations of Simon's algorithm and Fourier transform
  - Standard Hadamard matrix seems to be the best choice
- Full version: `https://eprint.iacr.org/2022/782`
- Code: `https://github.com/rub-hgi/period-search`
- Contact: {federico.canale, gregor.leander, lukas.stennes }@rub.de

Conclusion

CASA
Cyber Security in the Age of Large-Scale Adversaries

- Automatized Applications
  - Algorithm to automatically search for periodic functions
  - New attacks against MISTY and Feistel constructions
- Generalization
  - Generalizations of Simon's algorithm and Fourier transform
  - Standard Hadamard matrix seems to be the best choice
- Full version: `https://eprint.iacr.org/2022/782`
- Code: `https://github.com/rub-hgi/period-search`
- Contact: {federico.canale, gregor.leander, lukas.stennes }@rub.de

Conclusion

CASA
Cyber Security in the Age of Large-Scale Adversaries

- Automatized Applications
  - Algorithm to automatically search for periodic functions
  - New attacks against MISTY and Feistel constructions
- Generalization
  - Generalizations of Simon's algorithm and Fourier transform
  - Standard Hadamard matrix seems to be the best choice
- Full version: `https://eprint.iacr.org/2022/782`
- Code: `https://github.com/rub-hgi/period-search`
- Contact: {federico.canale, gregor.leander, lukas.stennes }@rub.de

Conclusion

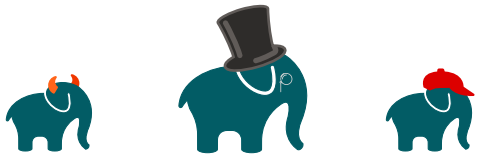
CASA
Cyber Security in the Age of Large-Scale Adversaries

- Automatized Applications
  - Algorithm to automatically search for periodic functions
  - New attacks against MISTY and Feistel constructions
- Generalization
  - Generalizations of Simon's algorithm and Fourier transform
  - Standard Hadamard matrix seems to be the best choice
- Full version: `https://eprint.iacr.org/2022/782`
- Code: `https://github.com/rub-hgi/period-search`
- Contact: {federico.canale, gregor.leander, lukas.stennes }@rub.de

Conclusion

CASA
Cyber Security in the Age of Large-Scale Adversaries

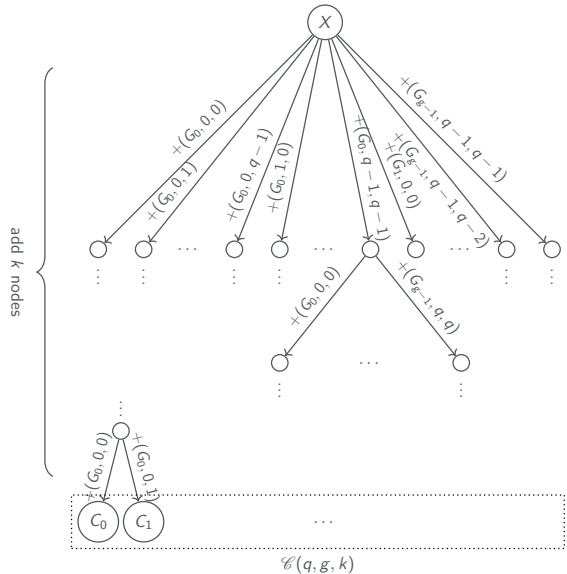
- Automatized Applications
  - Algorithm to automatically search for periodic functions
  - New attacks against MISTY and Feistel constructions
- Generalization
  - Generalizations of Simon's algorithm and Fourier transform
  - Standard Hadamard matrix seems to be the best choice
- Full version: `https://eprint.iacr.org/2022/782`
- Code: `https://github.com/rub-hgi/period-search`
- Contact: {federico.canale, gregor.leander, lukas.stennes }@rub.de



Thank You!

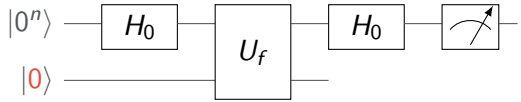
Backup Slides

Enumerating Circuits



Generalizing Simon's circuit

- Simon circuit for $f : \mathbb{F}_2^n \to \mathbb{F}_2$ *satisfying Simon's promise*

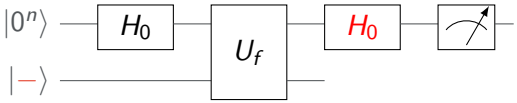


measures the state

$$\sum_{x:x\cdot s=0} 2^{-n+1} |x\rangle$$

Generalizing Simon's circuit

- Berstein-Vazirani circuit for $f : \mathbb{F}_2^n \to \mathbb{F}_2$

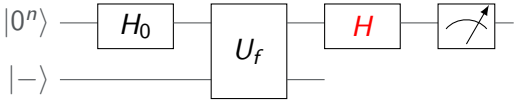


measures the state

$$\sum_{x \in \mathbb{F}_2^n} \Big(2^{-n} \underbrace{\sum_{y \in \mathbb{F}_2^n} (-1)^{x \cdot y + f(y)}}_{\widehat{f}^{H_0}(x)}\Big) |x\rangle\}$$

Generalizing Simon's circuit

- Generalized circuit for $f : \mathbb{F}_2^n \to \mathbb{F}_2$



measures the state

$$\sum_{x \in \mathbb{F}_2^n} 2^{-n} \Big(\underbrace{\sum_{y \in \mathbb{F}_2^n} (-1)^{g_H(x,y)+f(y)}}_{\widehat{f}^H(x)} \Big) |x\rangle \}$$

Proof ideas

Observe that $\widehat{f}^{H_g}$ satisfies the Theorem if and only if

$$g(x, Ay) = g(B^{-1}x, y).$$

Then consider $F_g : \mathrm{GL}(n, \mathbb{F}_2) \to \mathrm{GL}(n, \mathbb{F}_2)$ such that

$$F_g(A) \quad = \quad B.$$

This is actually an automorphism over $\mathrm{GL}(n, \mathbb{F}_2)$.

Lemma 1

For every automorphism F of $\mathrm{GL}(n, \mathbb{F}_2)$, there exists $G \in \mathrm{GL}(n, \mathbb{F}_2)$ such that either

$$F(A) = G^{-1}AG \quad \text{or} \quad F(A) = G^{-1}(A^T)^{-1}G \tag{1}$$

for all $A \in \mathrm{GL}(n, \mathbb{F}_2)$.

Proof ideas

Observe that $\widehat{f}^{H_g}$ satisfies the Theorem if and only if

$$g(x, Ay) = g(B^{-1}x, y).$$

Then consider $F_g : \mathrm{GL}(n, \mathbb{F}_2) \to \mathrm{GL}(n, \mathbb{F}_2)$ such that

$$F_g(A) \quad = \quad B.$$

This is actually an automorphism over $\mathrm{GL}(n, \mathbb{F}_2)$.

Lemma 1

For every automorphism F of $\mathrm{GL}(n, \mathbb{F}_2)$, there exists $G \in \mathrm{GL}(n, \mathbb{F}_2)$ such that either

$$F(A) = G^{-1}AG \quad \text{or} \quad F(A) = G^{-1}(A^T)^{-1}G \tag{1}$$

for all $A \in \mathrm{GL}(n, \mathbb{F}_2)$.

Proof ideas

Observe that $\widehat{f}^{H_g}$ satisfies the Theorem if and only if

$$g(x, Ay) = g(B^{-1}x, y).$$

Then consider $F_g : \mathrm{GL}(n, \mathbb{F}_2) \to \mathrm{GL}(n, \mathbb{F}_2)$ such that

$$F_g(A) \quad = \quad B.$$

This is actually an automorphism over $\mathrm{GL}(n, \mathbb{F}_2)$.

Lemma 1

For every automorphism F of $\mathrm{GL}(n, \mathbb{F}_2)$, there exists $G \in \mathrm{GL}(n, \mathbb{F}_2)$ such that either

$$F(A) = G^{-1}AG \quad \text{or} \quad F(A) = G^{-1}(A^T)^{-1}G \tag{1}$$

for all $A \in \mathrm{GL}(n, \mathbb{F}_2)$.