# Exploring SAT for Cryptanalysis

## (Quantum) Collision Attacks against 6-Round SHA-3

Jian Guo[1], Guozhen Liu[1], Ling Song[2], Yi Tu[1]

[1]Nanyang Technological University, Singapore

[2]Jinan University, China

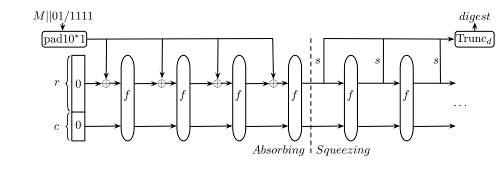
Asiacrypt 2022

# Overview

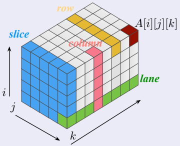
1. Summary of Collision Attacks on SHA-3
   - SHA-3 Hash Function
   - Collision Attacks - Review and Progress
2. SAT-based Automatic Search Toolkit
   - SAT Implementation
   - Truncated Differential Trail Search
   - Colliding Trail Search
   - Connecting Trail Search
3. 6-round Collision Attacks
   - Basic Attack Strategy
   - Classical Collision Attack
   - Quantum Collision Attacks
4. Conclusion

# SHA-3 Hash Family

## Sponge construction




## State




## Keccak-$f$ permutation

$\theta$: $A[i][j][k] \leftarrow A[i][j][k] \oplus \sum_{j'=0}^{4} A[i-1][j'][k] \oplus \sum_{j'=0}^{4} A[i+1][j'][k-1]$

$\rho$: $A[i][j] \leftarrow A[i][j] \lll T(i,j)$, where $T(i,j)$s are constants

$\pi$: $A[j][2i+3j] \leftarrow A[i][j]$

$\chi$: $A[i][j][k] \leftarrow A[i][j][k] \oplus (A[i+1][j][k] \oplus 1) \cdot A[i+2][j][k]$

$\iota$: $A[0][0] \leftarrow A[0][0] \oplus RC_{i_r}$, where $RC_{i_r}$ is the $i_r$-th round constant

## 6 Instances

**SHA-3**
- SHA3-224
- SHA3-256
- SHA3-384
- SHA3-512

**SHAKE**
- SHAKE128
- SHAKE256

# Collision Attacks against the SHA-3 family

## Overview of the state-of-the-art cryptanalytic results

| Target | Type | Rounds | Time Complexity | Reference |
|---|---|---|---|---|
| SHA3-224 | Classical | 5 | Practical | Guo et al. 2020[1] |
| | **Quantum** | **6** | $2^{97.75}/\sqrt{S}$ | Sect.4.4 |
| SHA3-256 | Classical | 5 | Practical | Guo et al. 2020[1] |
| | **Quantum** | **6** | $2^{104.25}/\sqrt{S}$ | Sect.4.3 |
| SHA3-384 | Classical | 4 | $2^{59.64}$ | Huang et al. 2022[2] |
| SHA3-512 | Classical | 3 | Practical | Dinur et al. 2013[3] |
| SHAKE128 | Classical | 5 | Practical | Guo et al. 2020[1] |
| | **Classical** | **6** | $2^{123.5}$ | Sect.4.2 |
| | **Quantum** | **6** | $2^{67.25}/\sqrt{S}$ | |
| SHAKE256 | - | - | - | - |

[1]Guo et al, ***JoC2020***, Practical collision attacks against round-reduced SHA-3

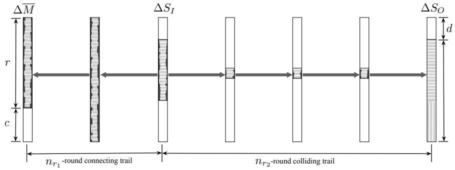
[2]Huang et al, ***ToSC2022***, Finding Collisions against 4-round SHA3-384 in Practical Time

[3]Dinur et al, ***FSE2013***, Collision attacks on up to 5 rounds of sha-3 using generalized internal differentials

# Collision Attacks - Revisit Previous Works

## Basic attack framework[1,2,3,4,5]

1. $n_{r_2}$-round colliding trail
2. $n_{r_1}$-round connector
3. exhaustive collision search




## Limitations & Obstacles

- *Colliding trail search*: highly dependent on truncated differential trail search
- *Connector construction*:
  - difficult to generate connecting trails
  - quick consumption of the Degree of Freedom

Lack of efficient trail search strategy.

[1] Dinur et al, ***FSE2012***, New attacks on Keccak-224 and Keccak-256

[2] Dinur et al, ***JoC2014***, Improved practical attacks on round-reduced Keccak

[3] Qiao et al, ***EuroCrypt2017***, New collision attacks on round-reduced Keccak

[4] Song et al, ***Crypto2017***, Non-full sbox linearization: Applications to collision attacks on round-reduced Keccak

[5] Guo et al, ***JoC2020***, Practical collision attacks against round-reduced SHA-3

# Collision Attacks - Our Progress

SAT-based Collision Attacks on SHA-3

## **SAT-based** trail search toolkit

- ***colliding trail search***
  - satisfying any digest length
  - covering more rounds
  - following specific differential pattern
  - supporting exact probability constraint
- ***connecting trail search***
  - a simpler and more efficient method compared with the previous **T**arget **D**ifference **A**lgorithm (TDA)
  - providing adequate Degree of Freedom (DF)

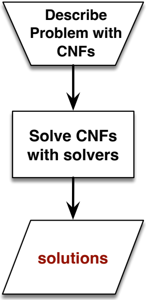
## **Improved collision attacks**

- 6-round collision attacks on SHAKE128
- 6-round quantum collision attacks on SHA3-224 and SHA3-256

# SAT implementation

## Basics

**General approach**




**The SAT-based automatic search**

### **SAT** the boolean **SAT**isfiability problems

- whether there exist valid assignments for a set of boolean formulas

### **CNF** the *conjunctive normal form*

- a literal, e.g., $x$ or $\neg x$
- a clause is a disjunction of literals
- a CNF is a conjunction of clauses or one clause

### **Solvers**

- DPLL solvers, the systematic backtracking search strategy
- CDCL solvers, the conflict-driven clause learning method
- CryptoMiniSAT, CaDiCal, MapleSAT, Lingeling, ...

# SAT implementation

Description of difference propagation over round function

## CryptoMiniSAT

- high efficiency
- supporting XOR clauses
- simple interfaces

## Propagation over 1-round

$$\alpha_r \xrightarrow{\theta} c_r \xrightarrow{\pi \circ \rho} \beta_r \xrightarrow{\chi} \alpha_{r+1}$$

- each difference bit $\alpha_r[i][j][k]$ is represented by a variable indexed with $(320 \times j + 64 \times i + k)_{\alpha_r}$.

## Describing propagation over Keccak-$f$ with CNFs

$\theta$: adding the XOR clauses to the solver

$\rho$ and $\pi$: simple index mapping

$\chi$: relation between $\beta_r$ and $\alpha_{r+1}$, for each Sbox

1. represent DDT with truth table
2. generate CNFs of truth table with ***Logic Friday***

# SAT implementation

## Description of objective function

### Cardinality encodings

- The Cardinality constraint, e.g., $\sum_{i=0}^{n-1} x_i \leq w$ or $\sum_{i=0}^{n-1} x_i \geq w$
- Translate the problem to CNFs with the *sequential encoding method*[1]
    - $(n \cdot (w+1) - w)$ auxiliary variables
    - $\mathcal{O}(n \cdot w)$ clauses

### Describing the objective function with CNFs

- constraints on *the number of active Sboxes* - Describe relation between difference and the variables that represent the activeness of an Sbox.
- constraints on *propagation weight* - Describe relation between difference and the variables that represent the propagation weight.

[1]Carsten Sinz, 2005, Towards an Optimal CNF Encoding of Boolean Cardinality Constraints

# Truncated Differential Trail Search

## SAT based Automatic Search Toolkit

### Truncated differential trail and ***trail core***

$$\alpha_i \xrightarrow{\lambda} \beta_i \xrightarrow{\chi} \alpha_{i+1} \xrightarrow{\lambda} \beta_{i+1} \xrightarrow{\chi} \alpha_{i+2} \xrightarrow{\lambda} \beta_{i+2} \xrightarrow{\chi} \alpha_{i+3} \xrightarrow{\lambda} \beta_{i+3} \xrightarrow{\chi} \alpha_{i+4}$$

$$w^{rev}(\alpha_{i+1}) \quad w(\beta_{i+1}) \quad w(\beta_{i+2}) \quad w(\beta_{i+3})$$

4-round trail: $(\alpha_i, \alpha_{i+1}, \alpha_{i+2}, \alpha_{i+3})$ or $(\beta_i, \beta_{i+1}, \beta_{i+2}, \beta_{i+3})$

4-round trail core: $(\alpha_{i+1}, \alpha_{i+2}, \alpha_{i+3})$ or $(\beta_{i+1}, \beta_{i+2}, \beta_{i+3})$

### SAT-based truncated trail search

1. Translate the trail core $(\alpha_{i+1}, \beta_{i+1}, \alpha_{i+2}, \beta_{i+2}, \alpha_{i+3}, \beta_{i+3})$ to CNFs.
2. Add constraints on propagation weight,
   $w^{rev}(\alpha_{i+1}) + w(\beta_{i+1}) + w(\beta_{i+2}) + w(\beta_{i+3}) \leq W$.

- Exhaustive 3-round trail search with $W$=52.
- 2 best 4-round truncated trail with propagation weight 133.

# Colliding Trail Search

$$\alpha_2 \xrightarrow{\lambda} \beta_2 \xrightarrow[w^{rev}=89]{\chi_2} \alpha_3 \xrightarrow{\lambda} \beta_3 \xrightarrow[w=24]{\chi_3} \alpha_4 \xrightarrow{\lambda} \beta_4 \xrightarrow[w=20]{\chi_4} \alpha_5 \xrightarrow{\lambda} \beta_5 \xrightarrow[w^d=8]{\chi_5} \alpha_6^d$$

- Translate the *digest collision* to CNFs.

$\alpha_6^d$, $d$=256

$\delta_{out}= *0000$

$\delta_{in} \in$ {00000, 00001, 00101, 10101, 00011, 01011, 00111, 10111, 01111, 11111}

Colliding trail: $(\alpha_3, \beta_3, \alpha_4, \beta_4, \alpha_5, \beta_5^d)$

- Constraints on *propagation weight* or *the number of active Sbox*

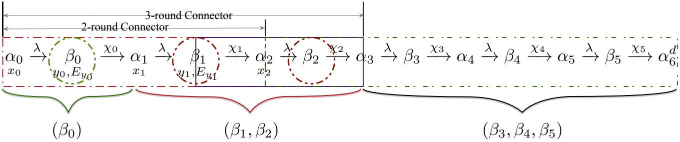
$w^{rev}(\alpha_3) + w(\beta_3) + w(\beta_4) + w(\beta_5^d) \leq W$

$AS(\alpha_3) + AS(\alpha_4) + AS(\beta_4) + AS(\beta_5^d) \leq N$

- Efficiency of colliding trail search

| Rounds | Weight | Time | Reference |
|---|---|---|---|
| 3 | 43 | Several weeks | Guo et al. |
| 3 | 32 | 2s | Our work |
| **4** | 141 | 5mins | Our work |

# Connecting Trail Search




**Phase 1.**

**Phase 2.**

## Generating $(\beta_1, \beta_2)$ trail

- describing trail $(\beta_1, \alpha_2, \beta_2, \alpha_3)$ with CNFs
- *weight constraints*
  - $w(\beta_1) + w(\beta_2) \leq W$
  - $w(\beta_1) \leq w_1$ and $w(\beta_2) \leq w_2$

## Generating $(\beta_0)$

- adding $\alpha_0$ and $\alpha_1$ to the solver
- ensuring $\beta_0$ is a valid connector by introducing $(x_0^1, x_0^2)$ variables
- *weight constraint* : the degree of freedom will be maximally produced for connectors

# Basic Attack Strategy - Trail Search

generate $n_{r_2}$-round colliding trail → generate $n_{r_1}$-round connecting trail → 2/3-round connector construction

## Generating 4-round colliding trail core

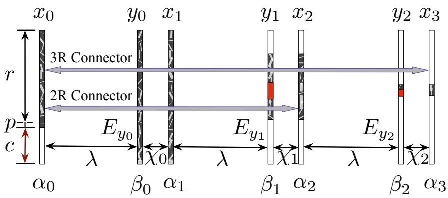
With the SAT-based toolkit, propagation weight $w \geq$141.

$$\alpha_2 \xrightarrow{\lambda} \beta_2 \xrightarrow[w^{rev}=89]{\chi_2} \alpha_3 \xrightarrow{\lambda} \beta_3 \xrightarrow[w=24]{\chi_3} \alpha_4 \xrightarrow{\lambda} \beta_4 \xrightarrow[w=20]{\chi_4} \alpha_5 \xrightarrow{\lambda} \beta_5 \xrightarrow[w^d=8]{\chi_5} \alpha_6^d$$

## Generating 2.5-round connecting trail

- With the SAT-based toolkit, (1) determine $(\beta_1, \beta_2)$, (2) determine $(\beta_0)$.
- **Advantage**: significant DF gain, e.g., increase from 124 to $330 \sim 430$.

# Basic Attack Strategy - Connector Construction




## 2-round connector

List a system of linear equations on $y_1$ satisfying

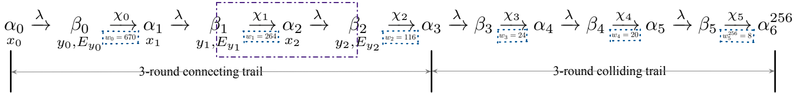
- $c+p$ condition
- $\chi_0$ propagation
- partial of $\chi_1$ propagation

Generate message pairs that partially follow $\alpha_2$.

## Key techniques

- Fully linearize $\chi_0$ to bypass the first round.
- Partially linearize $\chi_1$. Due to significant DF consumption, only part of $\beta_1$ bits are linearized. A greedy algorithm is used to determine which bits should be linearized.

# Collision Attacks on 6-round SHAKE128




## Differential trail

- *3-round colliding trail* $2^{-141} \Rightarrow 2^{-52}$
- *3-round connecting trail*

## Solution space of $E_{y_1}$

- message pairs follow partial of $\alpha_3$
- DF $= 27$

## Connector construction

List system of linear equations on $y_1$, $E_{y_1}$

- $E_{y_0}$, (1) $c+p$ (2) $\chi_0$ propagation
- $E_{y_1}$, (1) $\chi_1$ propagation
  - (2) fully linearize $\chi_0$, and transfer $E_{y_0}$ to $E_{y_1}$
- Transfer $E_{y_2}$ to $E_{y_1}$
  - select $\beta_2$ bit with greedy algorithm
  - partially linearize $\chi_2$, 36 out of 116
  - list $E_{y_2}$ and transfer to $E_{y_1}$

# Collision Attacks on 6-round SHAKE128

## Complexity analysis

### *Exhaustive search*

$2^{123.2}$ 6-round SHAKE128 computations

- $2^{132}$ SHAKE128 computations
  - The unsolved conditions of $\chi_2$, i.e., $2^{-80}=2^{-(116-36)}$
  - The colliding trail of probability $2^{-52}$
- *early-abort* technique, $2^{-9.8}$ gain for one bit condition
  1. 1st bit condition, $1/2$ pairs left
  2. 2nd bit condition, $1/4$ pairs of the remaining $1/2$ pairs left
  3. …

### *Connector construction*

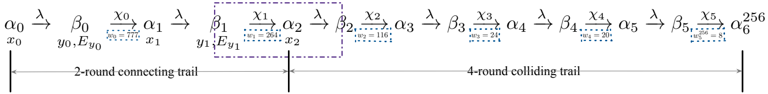
$2^{121.2}$ 6-round SHAKE128 computations

- $2^{105}$ ($= 2^{132}/2^{27}$) connectors
- the *equivalent conversion*
  - 56064 bitwise operations for 6-round SHAKE128 computation
  - $\mathcal{O}(m^2 n)$ bitwise operations for solving equation system, i.e., $\leq 1600^3 = 4.096 \times 10^9$ bitwise operations

### Total complexity

$2^{123.5}$ 6-round SHAKE128 computations

# 6-round **Quantum Collision** Attack on SHA3-256




## Basics

- The *time-space tradeoff margin* $2^{n/2}/S$
  - $n$, the digest length
  - $S$, the maximum size of quantum and classical computers
- Assume quantum circuits exist already and concentrate on complexity evaluation.

## Quantum collision attack

- *Brute-force phase*: $2^{206}$ 6-round SHA3-256
  - colliding trail $2^{168}$
  - unsolved condition $2^{38}$
- *Solution space*: DF $= 5$, $2^{201}$ connectors.
- Suppose there exists a quantum circuit $\mathcal{C}_1$ (resp. $\mathcal{C}_2$) for connector (resp. SHA3).
  1. Prepare $(M, M')$ with $\mathcal{C}_1$.
  2. For $(M, M')$, check digests with $\mathcal{C}_2$.
  3. Repeat until collision found.

# 6-round Quantum Collision Attack on SHA3-256

## Complexity analysis

Suppose $\mathcal{C}_1$ (resp. $\mathcal{C}_2$) of depth $T_c$ (resp. $T_s$) and width $S_c$ (resp. $S_s$).

Time complexity of parallelized Grover search

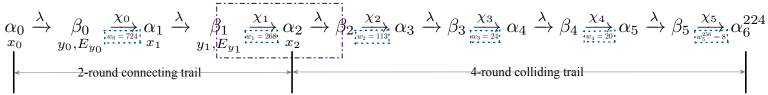
$$T_A \cdot (\pi/4) \cdot \sqrt{S_A/(p \cdot S)}$$

- Defined $T_s = 1$, $S_s = 1$ and at least 3456 qubits are required in $\mathcal{C}_2$.
- *Depth* ($T_A$). As $T_c$ is negligible, $T_A = T_s = 1$.
  - Compared to $T_s$ of nonlinear SHA3, $T_c$ of $\mathcal{C}_1$ that only contains linear operations (i.e., listing and solving equations) is negligible.
- *Width* ($S_A$). In $\mathcal{C}_1$, the quantum states include
  - $m$ qubits that mark whether to treat a condition or not
  - $k \times 1601$ qubits that store the $k$ boolean equations

  The overall $S_A = S_c + S_s = (m + k \times 1601 + 3456)/3456 \leq= 742$.
- The total **time complexity** of the quantum collision attack is

$$1 \cdot (\pi/4) \cdot \sqrt{(742 \times 2^{206})/S} = 2^{104.25}/\sqrt{S} < 2^{128}/S$$

# 6-round Quantum Collision Attack on SHA3-224




## Collision attack and complexity

- *Brute-force phase*: $2^{193}$ 6-round SHA3-224,
  - colliding trail $2^{165}$
  - unsolved condition $2^{28}$
- *Solution space of 2-round connector*: DF=22
- Complexity

$$1 \cdot (\pi/4) \cdot \sqrt{(((268 + 1600 \times 1601 + 3424)/3424) \times 2^{193})/S} = 2^{97.75}/\sqrt{S} < 2^{112}/S$$

# Conclusion

## SAT-based automatic toolkit

- *colliding trail search* - covering one more round
- *connecting trail search* - providing sufficient DF for connector construction
- *truncated differential trail search*

## Collision attacks on 6-round SHA-3 instances

| Target | Type | Connector Time | DF of Connector | Complexity |
|---|---|---|---|---|
| SHAKE128 | Classical<br>Quantum | 0.8s | 27 | $2^{123.5}$<br>$2^{67.25}/\sqrt{S}$ |
| SHA3-256 | Quantum | 3s | 5 | $2^{104.25}/\sqrt{S}$ |
| SHA3-224 | Quantum | 3s | 22 | $2^{97.75}/\sqrt{S}$ |

# Thank you for listening!

Questions? Comments?