ARXtools: A toolkit for ARX analysis

Gaëtan Leurent
University of Luxembourg

Presented by Pierre-Alain Fouque
ENS

Third NIST SHA-3 conference
Motivation

- Most of the cryptanalysis of ARX designs is **bit-twiddling**
  - As opposed to SBox based designs

- Building/Verifying differential path for ARX designs is **hard**
  - Many paths built by hand
  - Problems with MD5 and SHA-1 attacks
  - Problems reported with boomerang attacks (incompatible paths):
    - HAVAL
    - SHA-256

- Some tools are described in literature, but most are not available
Our tools

1. Tool for S-systems
   - Similar to [Mouha & al., SAC 2010]
   - Completely automated

2. Representation of differential paths as sets of constraints, and analysis with S-systems
   - Similar to [De Cannière & Rechberger, Asiacrypt 2006]
   - New set of constraints
   - Propagation of necessary constraints

3. Graphical tool for bit-twiddling with differential paths
Outline

Introduction

S-system Analysis

Differential characteristics

Application

# S-Systems

## Definition

**T-function**  \( \forall t, \)  \( t \) bits of the output can be computed from \( t \) bits of the input.

**S-function**  *There exist a set of states \( S \) so that:*
\[ \forall t, \]  bit \( t \) of the output and state \( S[t] \in S \) can be computed from bit \( t \) of the input, and state \( S[t - 1] \).

**S-system**  \( f(P, x) = 0 \)
\[ f \] is an S-function, \( P \) is a parameter, \( x \) is an unknown

- Operations mod \( 2^n \), Boolean functions are T-functions
- Addition, Xor, and Boolean operations are S-functions
Solving S-Systems

Important Example

\[ x \oplus \Delta = x \oplus \delta \]

- On average one solution
- Easy to solve because it’s a T-function.
  - Guess LSB, check, and move to next bit

- How easy exactly?
- Backtracking is exponential in the worst case:
  \[ x \oplus 0x80000000 = x \]

- For random \( \delta, \Delta \), most of the time the system is inconsistent
Solving S-Systems

**Important Example**

\[ x \oplus \Delta = x \oplus \delta \]

- On average one solution
- **Easy** to solve because it’s a T-function.
  - Guess LSB, check, and move to next bit

- How easy exactly?
  - Backtracking is **exponential** in the worst case:
    \[ x \oplus 0x80000000 = x \]
  - For random \( \delta, \Delta \), most of the time the system is **inconsistent**
Solving S-Systems

Important Example

\[ x \oplus \Delta = x \oplus \delta \]

- On average one solution
- **Easy** to solve because it’s a T-function.
  - Guess LSB, check, and move to next bit

- How easy exactly?
- Backtracking is **exponential** in the worst case:
  \[ x \oplus 0x80000000 = x \]

- For random \( \delta, \Delta \), most of the time the system is **inconsistent**
Solving S-Systems

Important Example

\[ x \oplus \Delta = x \oplus \delta \]

- On average one solution
- **Easy** to solve because it’s a T-function.
  - Guess LSB, check, and move to next bit

- How easy exactly?
- Backtracking is **exponential** in the worst case:
  \[ x \oplus 0x80000000 = x \]

- For random \( \delta, \Delta \), most of the time the system is **inconsistent**
Transition Automata

**Carry transitions for** $x \oplus \Delta = x \oplus \delta$.

<table>
<thead>
<tr>
<th>$c$</th>
<th>$\Delta$</th>
<th>$\delta$</th>
<th>$x$</th>
<th>$c'$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$c$</th>
<th>$\Delta$</th>
<th>$\delta$</th>
<th>$x$</th>
<th>$c'$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>-</td>
</tr>
</tbody>
</table>

We use automata to study $S$-systems: [Mouha & al., SAC 2010]

- States represent the carries
- Transitions are labeled with the variables
- Automaton accepts solutions to the system.
- Can count the number of solutions.
Transition Automata

Carry transitions for $x \oplus \Delta = x \oplus \delta$. The edges are indexed by $\Delta, \delta, x$

- 0,0,0
- 0,0,1
- 1,1,0
- 1,0,0
- 0,1,0
- 0,1,1

We use automata to study S-systems: [Mouha & al., SAC 2010]

- States represent the carries
- Transitions are labeled with the variables
- Automaton accepts solutions to the system.
- Can count the number of solutions.
Transition Automata

Carry transitions for $x \oplus \Delta = x \oplus \delta$. The edges are indexed by $\Delta, \delta, x$.

<table>
<thead>
<tr>
<th>State</th>
<th>0,0,0</th>
<th>0,0,1</th>
<th>1,1,0</th>
<th>1,0,1</th>
<th>0,1,0</th>
<th>0,1,1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transitions</td>
<td>1,1,1</td>
<td>1,0,0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

We use automata to study S-systems:
- States represent the carries
- Transitions are labeled with the variables
- Automaton accepts solutions to the system.
- Can count the number of solutions.

[Mouha & al., SAC 2010]
**Decision Automata**

Carry transitions for $x \oplus \Delta = x \oplus \delta$.

The edges are indexed by $\Delta, \delta, x$.

- Remove $x$ from the transitions
- Can decide whether a given $\Delta, \delta$ is compatible.
- Convert the non-deterministic automata to deterministic (optional).
Decision Automata

Decision automaton for $x \oplus \Delta = x \boxplus \delta$.

The edges are indexed by $\Delta, \delta$.

- Remove $x$ from the transitions
- Can decide whether a given $\Delta, \delta$ is compatible.
- Convert the non-deterministic automata to deterministic (optional).
**Decision Automata**

**Decision automaton for** $x \oplus \Delta = x \oplus \delta$.

The edges are indexed by $\Delta, \delta$.

- Remove $x$ from the transitions.
- Can decide whether a given $\Delta, \delta$ is compatible.
- Convert the non-deterministic automata to deterministic (optional).
Our Tool

1. Automatic construction of the automaton from a natural expression
   Useful to study properties of the system

   build_fsm -e "V0+P0 == V0^P1" -d -g | dot -Teps

2. C functions to test compatibility, count solutions, or solve systems
Outline

Introduction

S-system Analysis

Differential characteristics

Application
Differential Characteristic

\[ \delta a = \ldots - x \quad \delta b = -x - x \]

\[ \delta c = xx -- \quad \delta d = x -- \]

\[ \delta u = -x -- \]

\[ \delta v = \ldots - x \]

- Choose a difference operation: \( \oplus \)
- A differential only specifies the input and output difference
- A difference characteristic specifies the difference of each internal variable
  - Compute probability for each operation

\[ c = a + b \]
\[ u = c + d \]
\[ v = u \ll 2 \]
Differential Characteristic

\[ \delta a = -x-x \]
\[ \delta b = -x-x \]
\[ \delta c = x-x-x \]
\[ \delta d = x-x-x \]
\[ \delta u = -x-x \]
\[ \delta v = -x-x \]

▶ Choose a difference operation: \( \oplus \)

▶ A differential only specifies the input and output difference

▶ A difference characteristic specifies the difference of each internal variable
  ▶ Compute probability for each operation

\[
\begin{align*}
    c &= a + b \\
    u &= c + d \\
    v &= u \ll 2
\end{align*}
\]
Signed difference

- A path defines a set of good pairs:
  - $x[i] \oplus x'[i] = 1 \iff (x[i], x'[i]) \in \{(0, 1), (1, 0)\}$

- Wang introduced a signed difference:
  - $\delta(x[i], x'[i]) = +1 \iff (x[i], x'[i]) \in \{(0, 1)\}$
  - $\delta(x[i], x'[i]) = -1 \iff (x[i], x'[i]) \in \{(1, 0)\}$
  - Captures both xor difference and modular difference

- Generalized constraints
  - [De Cannière & Rechberger 06]

- Problem: how to compute probabilities?
### Generalized constraints

<table>
<thead>
<tr>
<th>(x, x'):</th>
<th>(0, 0)</th>
<th>(0, 1)</th>
<th>(1, 0)</th>
<th>(1, 1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>?</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>anything</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>–</td>
<td>✓</td>
<td>-</td>
<td>-</td>
<td>✓</td>
</tr>
<tr>
<td>x</td>
<td>-</td>
<td>✓</td>
<td>✓</td>
<td>-</td>
</tr>
<tr>
<td>x ≠ x'</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>✓</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>x = x' = 0</td>
<td>-</td>
<td>✓</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>u</td>
<td>(x, x') = (0, 1)</td>
<td>-</td>
<td>✓</td>
<td>-</td>
</tr>
<tr>
<td>n</td>
<td>(x, x') = (1, 0)</td>
<td>-</td>
<td>-</td>
<td>✓</td>
</tr>
<tr>
<td>1</td>
<td>x = x' = 0</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>#</td>
<td>incompatible</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>3</td>
<td>x = 0</td>
<td>✓</td>
<td>✓</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>x' = 0</td>
<td>✓</td>
<td>-</td>
<td>✓</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>A</td>
<td>x' = 1</td>
<td>-</td>
<td>✓</td>
<td>-</td>
</tr>
<tr>
<td>B</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>-</td>
</tr>
<tr>
<td>C</td>
<td>x = 1</td>
<td>-</td>
<td>-</td>
<td>✓</td>
</tr>
<tr>
<td>D</td>
<td></td>
<td>✓</td>
<td>-</td>
<td>✓</td>
</tr>
<tr>
<td>E</td>
<td></td>
<td>-</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
Signed difference

▷ A path defines a set of good pairs:
  ▷ $x[i] \oplus x'[i] = 1 \iff (x[i], x'[i]) \in \{(0, 1), (1, 0)\}$

▷ Wang introduced a signed difference:
  ▷ $\delta(x[i], x'[i]) = +1 \iff (x[i], x'[i]) \in \{(0, 1)\}$
  ▷ $\delta(x[i], x'[i]) = -1 \iff (x[i], x'[i]) \in \{(1, 0)\}$

▷ Captures both xor difference and modular difference

▷ Generalized constraints [De Cannière & Rechberger 06]

▷ Problem: how to compute probabilities?
## Generalized Characteristics

- We can write generalized constraints as an S-system:

\[
\begin{align*}
P_0 &= 0 \Rightarrow (x, x') \neq (0, 0) & P_1 &= 0 \Rightarrow (x, x') \neq (0, 1) \\
P_2 &= 0 \Rightarrow (x, x') \neq (1, 0) & P_3 &= 0 \Rightarrow (x, x') \neq (1, 1)
\end{align*}
\]

- We can now **compute the probability** of a generalized characteristic.
  - Addition, Xor, Boolean functions are S-functions
  - Rotations just rotate the constraints

<table>
<thead>
<tr>
<th>$(x, x')$:</th>
<th>$(0, 0)$</th>
<th>$(0, 1)$</th>
<th>$(1, 0)$</th>
<th>$(1, 1)$</th>
<th>$P_0$</th>
<th>$P_1$</th>
<th>$P_2$</th>
<th>$P_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>?</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>anything</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$-$ $x = x'$</td>
<td>✓</td>
<td>-</td>
<td>-</td>
<td>✓</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>$x$ $x \neq x'$</td>
<td>-</td>
<td>✓</td>
<td>✓</td>
<td>-</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0 $x = x' = 0$</td>
<td>✓</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$u$ $(x, x') = (0, 1)$</td>
<td>-</td>
<td>✓</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$n$ $(x, x') = (1, 0)$</td>
<td>-</td>
<td>-</td>
<td>✓</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1 $x = x' = 0$</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>✓</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td># <code>incompatible</code></td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
New Constraints

- Carry propagation leads to constraints of the form $x[i] = x[i-1]$

- We use new constraints to capture this information

- We consider subsets of $\{(x[i], x'[i], x[i-1])\}$ instead of $\{(x[i], x'[i])\}$

- This can still be written as an S-system with Boolean filtering on $x, x', x \boxplus x$. 
## New Constraints Table

<table>
<thead>
<tr>
<th>$(x ⊕ x', x ⊕ 2x, x)$:</th>
<th>$(0, 0, 0)$</th>
<th>$(0, 0, 1)$</th>
<th>$(0, 1, 0)$</th>
<th>$(0, 1, 1)$</th>
<th>$(1, 0, 0)$</th>
<th>$(1, 0, 1)$</th>
<th>$(1, 1, 0)$</th>
<th>$(1, 1, 1)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>? anything</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>$x = x'$</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$x ≠ x'$</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>$x = x' = 0$</td>
<td>✓</td>
<td>-</td>
<td>✓</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$(x, x') = (0, 1)$</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>✓</td>
<td>-</td>
<td>✓</td>
<td>-</td>
</tr>
<tr>
<td>$(x, x') = (1, 0)$</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>✓</td>
<td>-</td>
<td>✓</td>
</tr>
<tr>
<td>$x = x' = 0$</td>
<td>-</td>
<td>✓</td>
<td>-</td>
<td>✓</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td># incompatible</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$x = 0$</td>
<td>✓</td>
<td>-</td>
<td>✓</td>
<td>-</td>
<td>✓</td>
<td>-</td>
<td>✓</td>
<td>-</td>
</tr>
<tr>
<td>$x = 1$</td>
<td>-</td>
<td>✓</td>
<td>-</td>
<td>✓</td>
<td>-</td>
<td>✓</td>
<td>-</td>
<td>✓</td>
</tr>
<tr>
<td>$x' = 0$</td>
<td>✓</td>
<td>-</td>
<td>✓</td>
<td>-</td>
<td>-</td>
<td>✓</td>
<td>-</td>
<td>✓</td>
</tr>
<tr>
<td>$x' = 1$</td>
<td>-</td>
<td>✓</td>
<td>-</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>-</td>
<td>✓</td>
</tr>
<tr>
<td>$x = x' = 2x$</td>
<td>✓</td>
<td>✓</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$x = x' ≠ 2x$</td>
<td>-</td>
<td>-</td>
<td>✓</td>
<td>✓</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$x ≠ x' = 2x$</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>✓</td>
<td>✓</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$x ≠ x' ≠ 2x$</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
Propagation of constraints

We use S-systems to **propagate** constraints:

1. Split subsets in two smaller subsets

2. If one subset gives zero solutions, the characteristic can be restricted to the other subset.

\[
\begin{align*}
? & \rightarrow -/x, 3/C, 5/A & - & \rightarrow 0/1, =/! & x & \rightarrow u/n, >/< \\
3 & \rightarrow 0/u & C & \rightarrow 1/n & 5 & \rightarrow 0/n & A & \rightarrow 1/u \\
= & \rightarrow 0/1 & ! & \rightarrow 0/1 & > & \rightarrow u/n & < & \rightarrow u/n
\end{align*}
\]
Outline

Introduction

S-system Analysis

Differential characteristics

Application
Verifying paths

**Problem**

Most analysis assume that operations are independent and multiply the probabilities. *But sometimes, operations are not independent...* Known problem in Boomerang attacks. [Murphy, TIT 2011]

- We compute necessary conditions.
- This allows to detect cases of incompatibility
  - We have detected problems in several published works
    - Incompatible paths seem to appear quite naturally
Boomerang incompatibility

\[
\begin{align*}
\delta a &= -x^- \\
\delta b &= --- \\
\delta u &= --- \\
\end{align*}
\]

Top path: \((a^{(0)}, b^{(0)}; a^{(2)}, b^{(2)}) (a^{(1)}, b^{(1)}; a^{(3)}, b^{(3)})\)

Bottom path: \((a^{(0)}, b^{(0)}; a^{(1)}, b^{(1)}) (a^{(2)}, b^{(2)}; a^{(3)}, b^{(3)})\)

\[
\begin{array}{cccc}
 \chi^{(0)} & \chi^{(1)} & \chi^{(2)} & \chi^{(3)} \\
 a & 0 & 1 & 1 & 0 \\
 b & 1 & 0 & 0 & 1 \\
\end{array}
\]

- Appears easily with linearized paths, e.g. Blake
  [Biryukov & al., FSE 2011]

- Wlog, assume \(a^{(0)} = 0\)
- Compute \(a^{(i)}\), deduce sign of \(b\)
- Contradiction for \(b\)!
**Boomerang incompatibility**

\[ \delta a = -x - \quad \delta b = \ldots \]

- **Top path:** \((a^{(0)}, b^{(0)}; a^{(2)}, b^{(2)}) \ (a^{(1)}, b^{(1)}; a^{(3)}, b^{(3)})\)

\[ \delta a = -x - \quad \delta b = -x - \]

- **Bottom path:** \((a^{(0)}, b^{(0)}; a^{(1)}, b^{(1)}) \ (a^{(2)}, b^{(2)}; a^{(3)}, b^{(3)})\)

\[ \delta u = \ldots \]

\[ u = a + b \]

- Appears easily with linearized paths, e.g. Blake [Biryukov & al., FSE 2011]
- Wlog, assume \(a^{(0)} = 0\)
- Compute \(a^{(i)}\), deduce sign of \(b\)
- Contradiction for \(b\)!

\[
\begin{array}{cccc}
\hline
x^{(0)} & x^{(1)} & x^{(2)} & x^{(3)} \\
\hline
a & 0 & 1 & 1 & 0 \\
b & 1 & 0 & 0 & 1 \\
\hline
\end{array}
\]
Incompatibility with additions

Some “natural” differentials do not work with additions:

\[ \delta a = -x \quad \delta b = -x \quad \delta c = -x \]

\[ \delta u = -x \]

\[ u = a + b + c \]

- Linearized path

\[ \delta a = \underline{-xxxxx} \quad \delta b = \underline{---xx} \]

\[ \delta u = \underline{-xxxx-x} \]

\[ u = a + b \]

- Seems valid with signed difference

- Found in Skein near-collision [eprint 2011/148]
Carry incompatibility

- Each operation has a non-zero probability
- Path seems valid with signed difference

- Consider the 1\textsuperscript{st} addition
  - Constraint: \(c_4 \neq c_5\)
- Consider the 2\textsuperscript{nd} addition
  - Constraint: \(c'_2 = c'_3\)

- Incompatible!
  - Detected with the new constraints
Carry incompatibility

\( \delta a = -xx-- \quad \delta b = xxx-- \)

\( \delta c = -\neq---- \)

\( \delta c' = ---\neq-- \quad \delta d = ---xx- \)

\( \delta u = ---xx- \)

- Each operation has a non-zero probability
- Path seems valid with signed difference

- Consider the 1\textsuperscript{st} addition
  - Constraint: \( c^{[4]} \neq c^{[5]} \)

- Consider the 2\textsuperscript{nd} addition
  - Constraint: \( c^{[2]} = c^{[3]} \)

- Incompatible!
  - Detected with the new constraints
**Introduction**

**S-system Analysis**

**Differential characteristics**

**Application**

---

**Carry incompatibility**

\[
\begin{align*}
\delta a &= -xx--- \\
\delta b &= xxx--- \\
\delta c &= -====-
\end{align*}
\]

\[
\begin{align*}
\delta c' &= -====- \\
\delta d &= ---xx-
\end{align*}
\]

\[
\delta u = ---xx-
\]

- Each operation has a non-zero probability
- Path seems valid with signed difference
- Consider the 1st addition
- Consider the 2nd addition
  - Constraint: \( c'[2] = c'[3] \)
- Incompatible!
  - Detected with the new constraints

---

G. Leurent (pres: P.-A. Fouque)  
ARXtools: A toolkit for ARX analysis  
Third NIST SHA-3 conference
## Carry incompatibility

\[ \delta a = \text{-xx---} \quad \delta b = \text{xxx---} \]

\[ \delta c = \text{-#-----} \]

\[ \delta c' = \text{---#--} \quad \delta d = \text{---xx-} \]

\[ \delta u = \text{---xx-} \]

- Each operation has a non-zero probability
- Path seems valid with signed difference

- Consider the 1\textsuperscript{st} addition
  - Constraint: \( c^{[4]} \neq c^{[5]} \)

- Consider the 2\textsuperscript{nd} addition
  - Constraint: \( c'^{[2]} = c'^{[3]} \)

- Incompatible!
  - Detected with the new constraints
Graphical tool

- To study more complex cases, we have a graphical tool
- We can manually constrain some bits and propagate.
- Problems found in the Boomerang paths for Skein-512
  [Chen & Jia, ISPEC 2010]
Main result

Many published attacks are invalid.

- Boomerang attacks on Blake  
  - Basic linearized paths, with MSB difference
  - Proposed attack on 7/8 round for KP and 6/6.5 for CF do not work
  - 7-round KP attack can be made with the 6-round path
  - 8-round KP attack and 6/6.5-round CF attack can be fixed using another active bit (non-MSB)

- Boomerang attacks on Skein-512  
  - Basic linearized paths, with MSB difference
  - Proposed attacks do not work on Skein-512
  - Similar paths work on Skein-256 [Leurent & Roy, CT-RSA 2012]
  - Can be fixed using another active bit?

- Near-collision attack on Skein  
  - Complex rebound-like handcrafted path
  - Path is not satisfiable
Conclusion

We hope these tools will be useful to cryptanalysts...

Code and documentation available at:
http://www.di.ens.fr/~leurent/arxtools.html