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 [Manuel, DCC 2011]
  - Problems reported with boomerang attacks (incompatible paths):
    - HAVAL [Sasaki, SAC 2011]
    - SHA-256 [BLMN, Asiacrypt 2011]
- 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

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 $\mathcal{S}$ so that:*
$\forall t$, bit t of the output and state $S[t] \in \mathcal{S}$ 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

$$x \oplus \Delta = x \boxplus \delta$$

- 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 δ, Δ, most of the time the system is inconsistent

Solving S-Systems

Important Example

$$x \oplus \Delta = x \boxplus \delta$$

- 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 δ, Δ, most of the time the system is inconsistent

Solving S-Systems

Important Example

$$x \oplus \Delta = x \boxplus \delta$$

- 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 δ, Δ, most of the time the system is inconsistent

Solving S-Systems

Important Example

$$x \oplus \Delta = x \boxplus \delta$$

- 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:
  $x \oplus \text{0x80000000} = x$

- For random δ, Δ, most of the time the system is inconsistent

Transition Automata

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

c	Δ	δ	x	c′
0	0	0	0	0
0	0	0	1	0
0	0	1	0	-
0	0	1	1	-
0	1	0	0	-
0	1	0	1	-
0	1	1	0	0
0	1	1	1	1

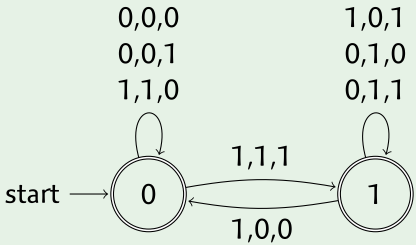
c	Δ	δ	x	c′
1	0	0	0	-
1	0	0	1	-
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	-
1	1	1	1	-

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 $x \oplus \Delta = x \boxplus \delta$. *The edges are indexed by Δ, δ, x*

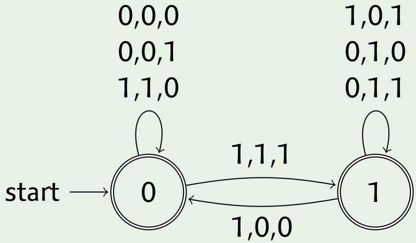


We use automata to study S-systems: [Mouha & *al.*, SAC 2010]

- 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 $x \oplus \Delta = x \boxplus \delta$. *The edges are indexed by Δ, δ, x*



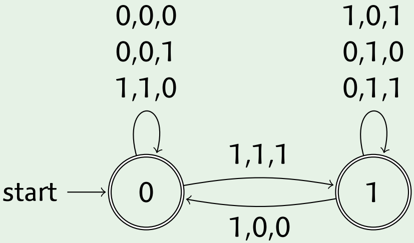
We use automata to study S-systems: [Mouha & *al.*, SAC 2010]

- States represent the carries
- Transitions are labeled with the variables

- Automaton accepts solutions to the system.
- Can count the number of solutions.

Decision Automata

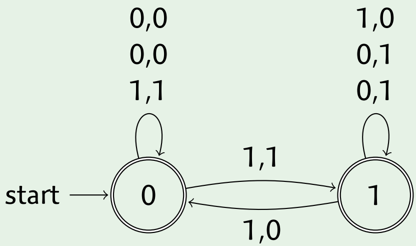
Carry transitions for $x \oplus \Delta = x \boxplus \delta$. *The edges are indexed by Δ, δ, x*



- Remove x from the transitions
- Can decide whether a given Δ, δ is compatible.
- Convert the non-deterministic automata to deterministic (optional).

Decision Automata

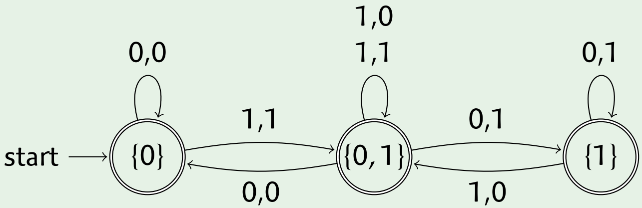
Decision automaton for $x \oplus \Delta = x \boxplus \delta$. *The edges are indexed by Δ, δ*



- Remove x from the transitions
- Can decide whether a given Δ, δ 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 Δ, δ*

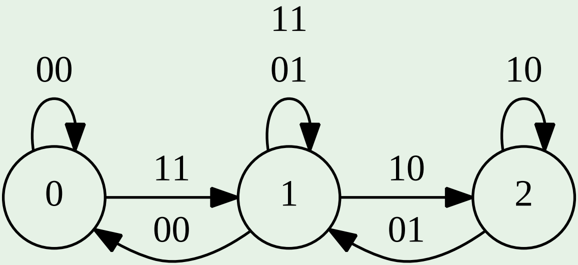


- Remove x from the transitions
- Can decide whether a given Δ, δ is compatible.
- Convert the non-deterministic automata to deterministic (optional).

Our Tool

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

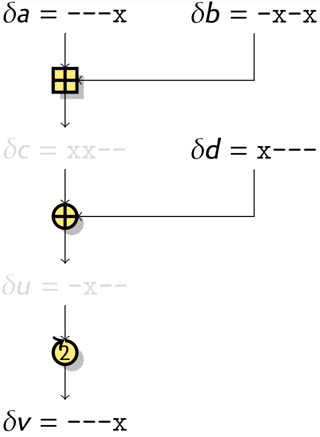
```
build_fsm -e "V0+P0 == V0^P1" -d -g | dot -Teps
```



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

Outline

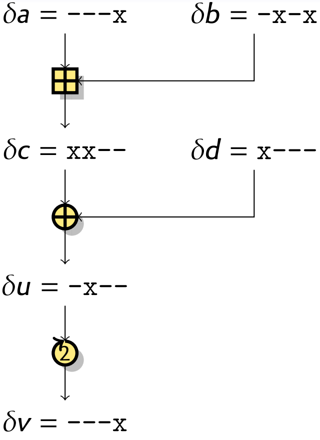
Differential Characteristic



$$\boxed{\begin{aligned} c &= a + b \\ u &= c + d \\ v &= u \lll 2 \end{aligned}}$$

- 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

Differential Characteristic



$$c = a + b$$
$$u = c + d$$
$$v = u \lll 2$$

- 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

Signed difference

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

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

- Generalized constraints [De Cannière & Rechberger 06]

- Problem: how to compute probabilities?

Generalized constraints [De Cannière & Rechberger 06]

	(x,x'):	$(0,0)$	$(0,1)$	$(1,0)$	$(1,1)$
?	*anything*	✓	✓	✓	✓
-	$x = x'$	✓	-	-	✓
x	$x \neq x'$	-	✓	✓	-
0	$x = x' = 0$	✓	-	-	-
u	$(x,x') = (0,1)$	-	✓	-	-
n	$(x,x') = (1,0)$	-	-	✓	-
1	$x = x' = 0$	-	-	-	✓
#	*incompatible*	-	-	-	-
3	$x = 0$	✓	✓	-	-
5	$x' = 0$	✓	-	✓	-
7		✓	✓	✓	-
A	$x' = 1$	-	✓	-	✓
B		✓	✓	-	✓
C	$x = 1$	-	-	✓	✓
D		✓	-	✓	✓
E		-	✓	✓	✓

Signed difference

- A path defines a set of good pairs:
  - $x^{[i]} \oplus x'^{[i]} = 1 \qquad \Leftrightarrow \qquad (x^{[i]}, x'^{[i]}) \in \{(0,1),(1,0)\}$
- Wang introduced a signed difference:
  - $\delta(x^{[i]}, x'^{[i]}) = +1 \qquad \Leftrightarrow \qquad (x^{[i]}, x'^{[i]}) \in \{(0,1)\}$
  - $\delta(x^{[i]}, x'^{[i]}) = -1 \qquad \Leftrightarrow \qquad (x^{[i]}, x'^{[i]}) \in \{(1,0)\}$
  - 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:

$$P_0 = 0 \Rightarrow (x,x') \neq (0,0) \qquad P_1 = 0 \Rightarrow (x,x') \neq (0,1)$$
$$P_2 = 0 \Rightarrow (x,x') \neq (1,0) \qquad P_3 = 0 \Rightarrow (x,x') \neq (1,1)$$

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

	(x,x'):	$(0,0)$	$(0,1)$	$(1,0)$	$(1,1)$	P_0	P_1	P_2	P_3
?	*anything*	✓	✓	✓	✓	1	1	1	1
-	$x = x'$	✓	-	-	✓	1	0	0	1
x	$x \neq x'$	-	✓	✓	-	0	1	1	0
0	$x = x' = 0$	✓	-	-	-	1	0	0	0
u	$(x,x') = (0,1)$	-	✓	-	-	0	1	0	0
n	$(x,x') = (1,0)$	-	-	✓	-	0	0	1	0
1	$x = x' = 0$	-	-	-	✓	0	0	0	1
#	*incompatible*	-	-	-	-	0	0	0	0

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 $\left\{(x^{[i]}, x'^{[i]}, x^{[i-1]})\right\}$ instead of $\left\{(x^{[i]}, x'^{[i]})\right\}$
- This can still be written as an S-system
  with Boolean filtering on $x, x', x \boxplus x$.

New Constraints Table

$(x \oplus x', x \oplus 2x, x)$:		$(0,0,0)$	$(0,0,1)$	$(0,1,0)$	$(0,1,1)$	$(1,0,0)$	$(1,0,1)$	$(1,1,0)$	$(1,1,1)$
?	*anything*	✓	✓	✓	✓	✓	✓	✓	✓
-	$x = x'$	✓	✓	✓	✓	-	-	-	-
x	$x \neq x'$	-	-	-	-	✓	✓	✓	✓
0	$x = x' = 0$	✓	-	✓	-	-	-	-	-
u	$(x, x') = (0, 1)$	-	-	-	-	✓	-	✓	-
n	$(x, x') = (1, 0)$	-	-	-	-	-	✓	-	✓
1	$x = x' = 0$	-	✓	-	✓	-	-	-	-
#	*incompatible*	-	-	-	-	-	-	-	-
3	$x = 0$	✓	-	✓	-	✓	-	✓	-
C	$x = 1$	-	✓	-	✓	-	✓	-	✓
5	$x' = 0$	✓	-	✓	-	-	✓	-	✓
A	$x' = 1$	-	✓	-	✓	✓	-	✓	-
=	$x = x' = 2x$	✓	✓	-	-	-	-	-	-
!	$x = x' \neq 2x$	-	-	✓	✓	-	-	-	-
>	$x \neq x' = 2x$	-	-	-	-	✓	✓	-	-
<	$x \neq x' \neq 2x$	-	-	-	-	-	-	✓	✓

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.

$\texttt{?} \longrightarrow \texttt{-/x}, \texttt{3/C}, \texttt{5/A}$	$\texttt{-} \longrightarrow \texttt{0/1}, \texttt{=/!}$	$\texttt{x} \longrightarrow \texttt{u/n}, \texttt{>/<}$	
$\texttt{3} \longrightarrow \texttt{0/u}$	$\texttt{C} \longrightarrow \texttt{1/n}$	$\texttt{5} \longrightarrow \texttt{0/n}$	$\texttt{A} \longrightarrow \texttt{1/u}$
$\texttt{=} \longrightarrow \texttt{0/1}$	$\texttt{!} \longrightarrow \texttt{0/1}$	$\texttt{>} \longrightarrow \texttt{u/n}$	$\texttt{<} \longrightarrow \texttt{u/n}$

Outline

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

$\delta a = \texttt{-x-}$ $\quad \delta b = \texttt{---}$ $\quad$ *Top path:* $(a^{(0)}, b^{(0)}; a^{(2)}, b^{(2)})$ $(a^{(1)}, b^{(1)}; a^{(3)}, b^{(3)})$

$\delta a = \texttt{-x-}$ $\quad \delta b = \texttt{-x-}$ $\quad$ *Bottom path:* $(a^{(0)}, b^{(0)}; a^{(1)}, b^{(1)})$ $(a^{(2)}, b^{(2)}; a^{(3)}, b^{(3)})$

$\delta u = \texttt{---}$

$$\boxed{u = a + b}$$

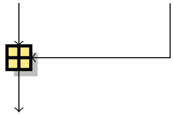
	$x^{(0)}$	$x^{(1)}$	$x^{(2)}$	$x^{(3)}$
a	0	1	1	0
b	1	0	0	1

- 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

$\delta a = \text{-x-}$ $\quad$ $\delta b = \text{---}$ $\quad$ *Top path:* $(a^{(0)}, b^{(0)}; a^{(2)}, b^{(2)})$ $(a^{(1)}, b^{(1)}; a^{(3)}, b^{(3)})$

$\delta a = \text{-x-}$ $\quad$ $\delta b = \text{-x-}$ $\quad$ *Bottom path:* $(a^{(0)}, b^{(0)}; a^{(1)}, b^{(1)})$ $(a^{(2)}, b^{(2)}; a^{(3)}, b^{(3)})$



$\delta u = \text{---}$

$\boxed{u = a + b}$

	$x^{(0)}$	$x^{(1)}$	$x^{(2)}$	$x^{(3)}$
a	0	1	1	0
b	1	0	0	1

- 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!

Incompatibility with additions

Some "natural" differentials do not work with additions:

$\delta a = \texttt{-x}$ $\quad \delta b = \texttt{-x}$ $\quad \delta c = \texttt{-x}$

$\delta u = \texttt{-x}$

$$\boxed{u = a + b + c}$$

- Linearized path

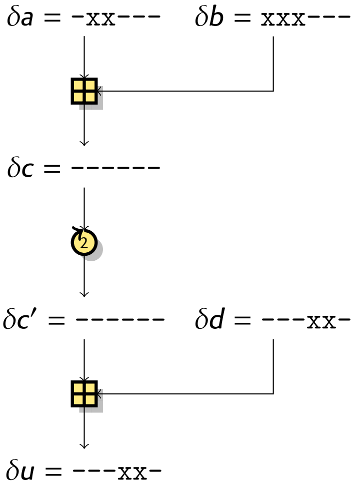
$\delta a = \texttt{--xxxxx-}$ $\quad \delta b = \texttt{---xx---}$

$\delta u = \texttt{-xxxx-x-}$

$$\boxed{u = a + b}$$

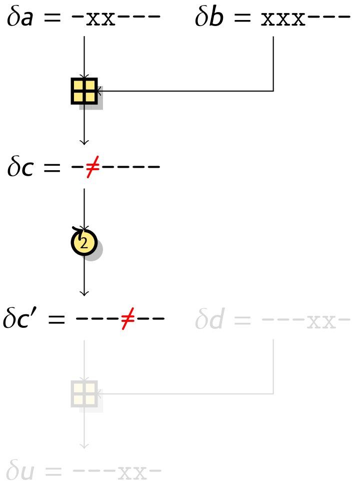
- Seems valid with signed difference
- Found in Skein near-collision [eprint 2011/148]

Carry incompatibility



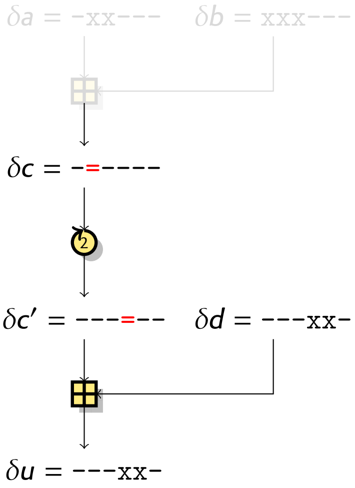
- Each operation has a non-zero probability
- Path seems valid with signed difference
- Consider the 1st addition
  - Constraint: $c^{[4]} \neq c^{[5]}$
- Consider the 2nd addition
  - Constraint: $c'^{[2]} = c'^{[3]}$
- Incompatible!
  - Detected with the new constraints

Carry incompatibility



- Each operation has a non-zero probability
- Path seems valid with signed difference
- Consider the 1st addition
  - Constraint: $c^{[4]} \neq c^{[5]}$
- Consider the 2nd addition
  - Constraint: $c'^{[2]} = c'^{[3]}$
- Incompatible!
  - Detected with the new constraints

Carry incompatibility



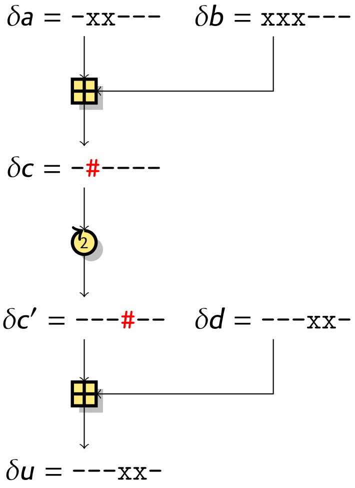
- Each operation has a non-zero probability
- Path seems valid with signed difference

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

- Consider the 2nd addition
  - Constraint: $c'^{[2]} = c'^{[3]}$

- Incompatible!
  - Detected with the new constraints

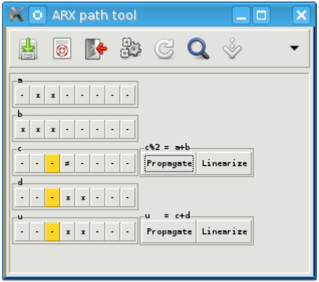
Carry incompatibility



- Each operation has a non-zero probability
- Path seems valid with signed difference
- Consider the 1st addition
  - Constraint: $c^{[4]} \neq c^{[5]}$
- Consider the 2nd 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 [Biryukov & *al.*, FSE 2011]
  - 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 [Chen & Jia, ISPEC 2010]
  - 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 [eprint 2011/148]
  - Complex rebound-like handcrafted path
  - Path is not satisfiable

Conclusion

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

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