Practical Near-Collisions on the Compression Function of BMW

Gaëtan Leurent and Søren S. Thomsen

University of Luxembourg
Technical University of Denmark

FSE 2011

The SHA-3 competition

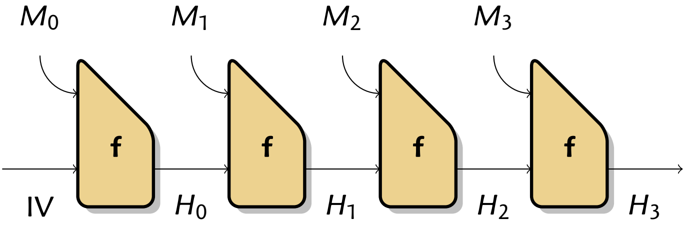
The SHA-3 competition

- 51 valid submissions
- 14 in the second round (July 2009)
- 5 finalists in December 2010
- Winner in 2012?

- BMW was the fastest second-round candidate in software
- Not selected for the third round

Hash Function Design

- Build a small compression function, and iterate.
  - Cut the message in chunks $M_0, \ldots M_k$
  - $H_i = f(M_i, H_{i-1})$
  - $F(M) = \Omega(H_k)$



Compression Function Attacks

Fist results usually target the compression function

- Because it's easier: more degrees of freedom
- Because good compression imply good hash function

MD5 cryptanalysis

- 1993: Free-start collisions [den Boer and Bosselaers]
- 1996: Semi-free-start collisions [Dobbertin]
- 2005: Collisions [Wang *et. al*]
- 2009: Rogue certificate [Stevens *et. al*]

Wang's and Stevens's attacks are based on the dBB path

Compression Function Attacks

Fist results usually target the compression function

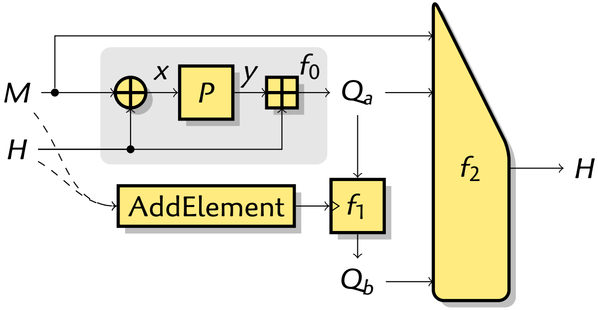
- Because it's easier: more degrees of freedom
- Because good compression imply good hash function

MD5 cryptanalysis

1993: Free-start collisions	[den Boer and Bosselaers]
1996: Semi-free-start collisions	[Dobbertin]
2005: Collisions	[Wang *et. al*]
2009: Rogue certificate	[Stevens *et. al*]

Wang's and Stevens's attacks are based on the dBB path

Blue Midnight Wish



- Wide pipe: each line is 16 words (32 or 64 bits)
- Most of the diffusion happens in f_1
- ARX: Addition, Rotations, Xors

▸ see details

Solving AX 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

Solving AX 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

Solving AX 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:
  $x \oplus \texttt{0x80000000} = x$

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

Solving AX 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:
  $x \oplus \texttt{0x80000000} = x$

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

Transition Automata

We use automata to study AX systems: [Mouha *et. al*]

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

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

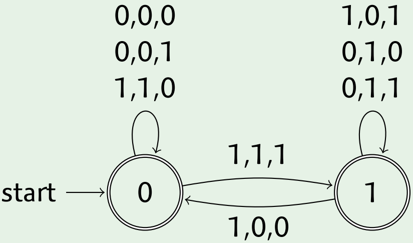
c	Δ	δ	x	c′
1	0	0	0	-
1	0	0	1	-
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	-
1	1	1	1	-

Transition Automata

We use automata to study AX systems: [Mouha *et. al*]

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

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

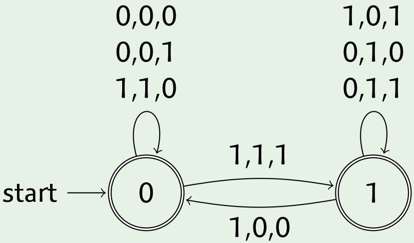


▸ see example

Decision Automata

- Remove x from the transitions
- Convert the non-deterministic automata to deterministic.

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

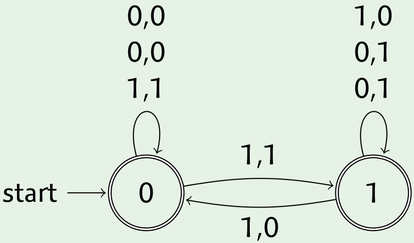


- Can decide whether a given Δ, δ is compatible.

Decision Automata

- Remove x from the transitions
- Convert the non-deterministic automata to deterministic.

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

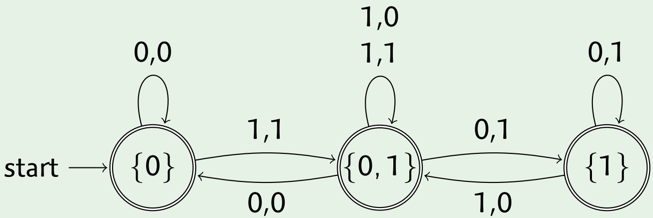


- Can decide whether a given Δ, δ is compatible.

Decision Automata

- Remove x from the transitions
- Convert the non-deterministic automata to deterministic.

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



- Can decide whether a given Δ, δ is compatible.

Solving AX systems

Take an AX system with variables and parameters.
e.g. $x \oplus \Delta = x \boxplus \delta$

1. Compute carry transitions
2. Build transition automaton
3. Remove variables and compute equivalent deterministic automaton

- For each values of the parameters:
  - Test if system is coherent in linear time
  - Find a solution in linear time

Can also study properties of the systems.

Some Properties

Important Example

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

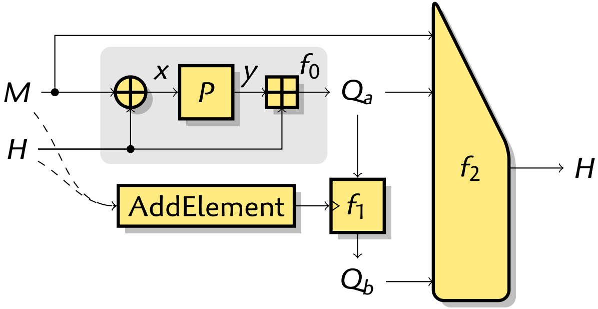
- For this particular system, we can build very efficient test:
- Consistent iff $\begin{cases} \Delta_0 = \delta_0 \\ \forall i : \Delta_i = 1 \quad \text{or} \quad \delta_i \oplus \Delta_{i+1} \oplus \delta_{i+1} = 0 \end{cases}$

```
!((D^d)&1) && !(((((D^d)>>1)^d) & (~D)) << 1)
```

  - Probability $2^{-13.9}$ for random δ, Δ
  - Probability 2^{-1} for random δ and $\Delta = -1$
- Solutions:

```
(D^d)>>1 ^ (r&(~D|0x8000000))
```

Application to BMW

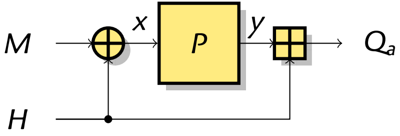


- If we have
  - a (near) collision in Q_a
  - a (near) collision in M
  - a (near) collision in the the first rounds of f_1

this can be seen in the output:

$$HH_0 = (XH^{\gg 5} \oplus Q_{16}^{\gg 5} \oplus M_0) \boxplus (XL \oplus Q_{24} \oplus Q_0)$$

Inside f_0

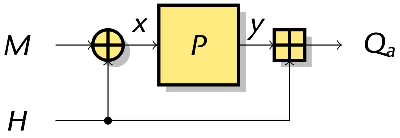


- We want no difference in Q_a, no difference in M
- Pick a random pair x/x', compute y/y' through P
- Solve the AX system:

$$\begin{aligned} M \oplus H &= x \\ y \boxplus H &= Q_a \end{aligned} \qquad \begin{aligned} M \oplus H' &= x' \\ y' \boxplus H' &= Q_a \end{aligned}$$

where H, H', M, Q_a are unknown, x, x', y, y' are given parameters

Inside f_0



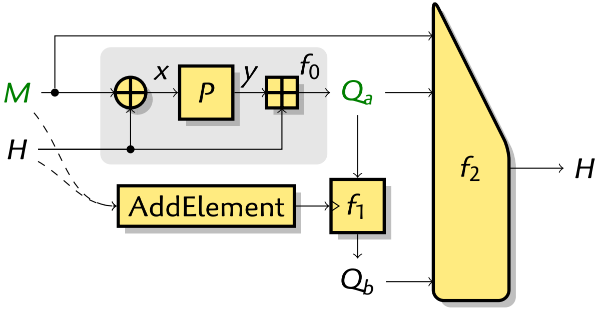
- We want no difference in Q_a, no difference in M
- Pick a random pair x/x', compute y/y' through P
- Solve the AX system:

$$H \oplus \Delta = H \boxplus \delta$$

$$\Delta = (x \oplus x') \quad \delta = (y \boxminus y')$$

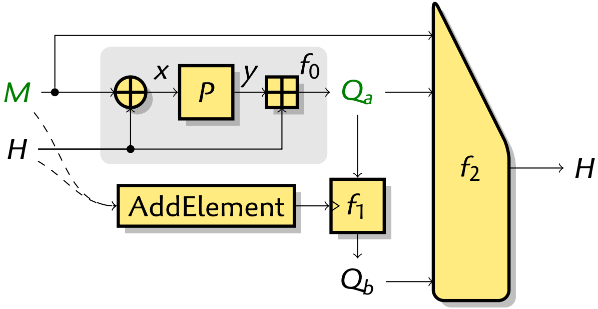
where H, H', M, Q_a are unknown, x, x', y, y' are given parameters

Basic BMW Attack



1. Chose a random x, x' so that $x' \oplus x$ has a high weight
2. Compute y, y'
3. Solve $H \oplus \Delta = H \boxplus \delta$.

Basic BMW Attack

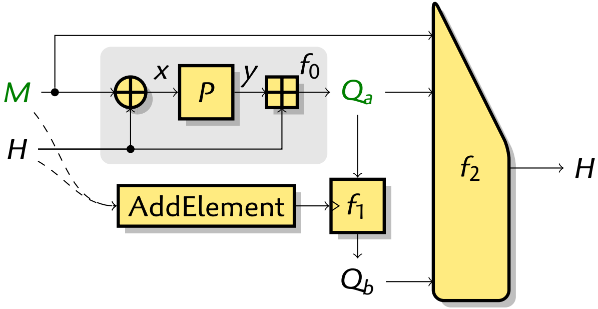


- The analysis of the f_0 function is the core of the attack
- We use degrees of freedom in x, x' to improve the attack

- First improvement: make some words of H inactive
  - f_1 is a FSR
  - $\mathrm{AddElement}(16) = (M_0^{\lll 1} \boxplus M_3^{\lll 4} \boxminus M_{10}^{\lll 11} \boxplus K_{16}) \oplus H_7$

▸ see details

Basic BMW Attack

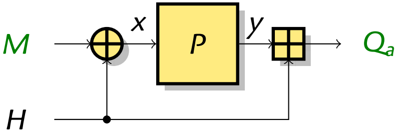


- The analysis of the f_0 function is the core of the attack
- We use degrees of freedom in x, x' to improve the attack

- First improvement: make some words of H inactive
  - f_1 is a FSR
  - $\text{AddElement}(16) = (M_0^{\lll 1} \boxplus M_3^{\lll 4} \boxminus M_{10}^{\lll 11} \boxplus K_{16}) \oplus H_7$

▸ see details

Inside f_0



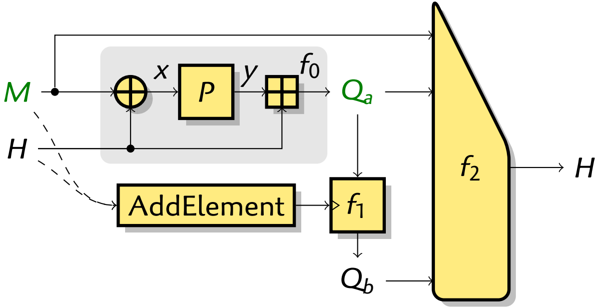
The P permutation

- $\boxplus$-Linear layer
  - $z = M.x$
- Word-wise operations
  - $y_i = f_i(z_i)$

▸ see details

- H_i is inactive iff x_i, y_i and z_i are inactive
  - Linear constraints
- We can have $H_7, H_8, \ldots H_{13}$ inactive
  - This gives $Q_{16}, Q_{17}, \ldots Q_{22}$ inactive
- Reduce the x, x' space by fixing $x' \boxminus x$
  - When $x' \boxminus x \neq 0$, x, x' constrained by high Hamming distance
  - When $x' \boxminus x = 0$, x is free, and H is free

Using Collisions in AddElement



- Second improvement: allow differences in M, cancel M differences and H differences in AddElement
  - Can use degrees of freedom in the inactive H

Collisions in AddElement

Our path

- differences in M_{13}, M_{14}, M_{15};
- differences in $H_1 \ldots H_6, H_{10}, H_{11}$ and H_{12}.

$$\begin{array}{ll}
\textit{AddElement(16)} & (M_0^{\lll 1} \boxplus M_3^{\lll 4} \boxminus M_{10}^{\lll 11} \boxplus K_{16}) \oplus H_7 \\
\textit{AddElement(17)} & (M_1^{\lll 2} \boxplus M_4^{\lll 5} \boxminus M_{11}^{\lll 12} \boxplus K_{17}) \oplus H_8 \\
\textit{AddElement(18)} & (M_2^{\lll 3} \boxplus M_5^{\lll 6} \boxminus M_{12}^{\lll 13} \boxplus K_{18}) \oplus H_9 \\
\textit{AddElement(19)} & (M_3^{\lll 4} \boxplus M_6^{\lll 7} \boxminus \underline{M_{13}^{\lll 14}} \boxplus K_{19}) \oplus H_{10} \\
\textit{AddElement(20)} & (M_4^{\lll 5} \boxplus \underline{M_7^{\lll 8}} \boxminus \underline{M_{14}^{\lll 15}} \boxplus K_{20}) \oplus H_{11}
\end{array}$$

...

Just another AX system

- Use the degrees of freedom form the inactive x_i's

Summary of the attack

1. Select a difference $x' \boxminus x$ such that selected words of x and y are inactive
2. Select a value for x so that $x' \oplus x$ has a high weight
   - By extending the carries
   - Increases the probability that the system is consistent.
3. Solve $H \oplus \Delta = H \boxplus \delta$. If inconsistent, goto 1.
4. Use degrees of freedom in H to make AddElement (near) collide. If impossible, goto 1.
5. Randomize with remaining degrees of freedom until XH collides.

Output function

$$
\begin{array}{llllllllllllll}
HH_0 & = & & & (XH^{\gg 5} & \oplus & Q_{16}^{\gg 5} & \oplus & M_0) & \boxplus & (XL & \oplus & Q_{24} & \oplus\ Q_0) \\
HH_1 & = & & & (XH^{\ll 7} & \oplus & Q_{17}^{\ll 8} & \oplus & M_1) & \boxplus & (XL & \oplus & Q_{25} & \oplus\ Q_1) \\
HH_2 & = & & & (XH^{\gg 5} & \oplus & Q_{18}^{\ll 5} & \oplus & M_2) & \boxplus & (XL & \oplus & Q_{26} & \oplus\ Q_2) \\
HH_3 & = & & & (XH^{\gg 1} & \oplus & Q_{19}^{\ll 5} & \oplus & M_3) & \boxplus & (XL & \oplus & Q_{27} & \oplus\ Q_3) \\
HH_4 & = & & & (XH^{\gg 3} & \oplus & Q_{20} & \oplus & M_4) & \boxplus & (XL & \oplus & Q_{28} & \oplus\ Q_4) \\
HH_5 & = & & & (XH^{\ll 6} & \oplus & Q_{21}^{\gg 6} & \oplus & M_5) & \boxplus & (XL & \oplus & Q_{29} & \oplus\ Q_5) \\
HH_6 & = & & & (XH^{\gg 4} & \oplus & Q_{22}^{\ll 6} & \oplus & M_6) & \boxplus & (XL & \oplus & Q_{30} & \oplus\ Q_6) \\
HH_7 & = & & & (XH^{\gg 11} & \oplus & Q_{23}^{\ll 2} & \oplus & M_7) & \boxplus & (XL & \oplus & Q_{31} & \oplus\ Q_7) \\
HH_8 & = & HH_4^{\ll 9} & \boxplus & (XH & \oplus & Q_{24} & \oplus & M_8) & \boxplus & (XL^{\ll 8} & \oplus & Q_{23} & \oplus\ Q_8) \\
HH_9 & = & HH_5^{\ll 10} & \boxplus & (XH & \oplus & Q_{25} & \oplus & M_9) & \boxplus & (XL^{\gg 6} & \oplus & Q_{16} & \oplus\ Q_9) \\
HH_{10} & = & HH_6^{\ll 11} & \boxplus & (XH & \oplus & Q_{26} & \oplus & M_{10}) & \boxplus & (XL^{\ll 6} & \oplus & Q_{17} & \oplus\ Q_{10}) \\
HH_{11} & = & HH_7^{\ll 12} & \boxplus & (XH & \oplus & Q_{27} & \oplus & M_{11}) & \boxplus & (XL^{\ll 4} & \oplus & Q_{18} & \oplus\ Q_{11}) \\
HH_{12} & = & HH_0^{\ll 13} & \boxplus & (XH & \oplus & Q_{28} & \oplus & M_{12}) & \boxplus & (XL^{\gg 3} & \oplus & Q_{19} & \oplus\ Q_{12}) \\
HH_{13} & = & HH_1^{\ll 14} & \boxplus & (XH & \oplus & Q_{29} & \oplus & M_{13}) & \boxplus & (XL^{\gg 4} & \oplus & Q_{20} & \oplus\ Q_{13}) \\
HH_{14} & = & HH_2^{\ll 15} & \boxplus & (XH & \oplus & Q_{30} & \oplus & M_{14}) & \boxplus & (XL^{\gg 7} & \oplus & Q_{21} & \oplus\ Q_{14}) \\
HH_{15} & = & HH_3^{\ll 16} & \boxplus & (XH & \oplus & Q_{31} & \oplus & M_{15}) & \boxplus & (XL^{\gg 2} & \oplus & Q_{22} & \oplus\ Q_{15})
\end{array}
$$

Practical example

Chaining Value							
59dfd94b	30b036e3	44ad8a65	47461712	59dfd94b	30b036e2	bb52759b	b8b9e8ed
6f56e9b4	425e2d65	40000003	94e62f58	90a9164c	bda1d29a	bffffffc	94e62f58
12c4bf76	17b18302	4f74ffd3	3ec30f93	12c4bf76	17b18302	b08b002c	c13cf06c
8b0f9f9b	7071a4a5	28becf17	6954724f	74f06064	7071a4a5	28becf17	6954724f

Message							
bd050fb4	c6925351	991aa15f	60327d4b	bd050fb4	c6925351	991aa15f	60327d4b
0212e457	9feb065e	d6ab8dac	7b52f8ca	0212e457	9feb065e	d6ab8dac	7b52f8ca
2f8a9774	1f189302	2043dc85	7b0eac19	2f8a9774	1f189302	2043dc85	7b0eac19
08fe0408	01c2f910	19abe45b	00000000	08fe0408	01c6f910	e6541ba4	ffffffe0

Output							
70588aa3	62e38880	4b32cd23	7da56fd2	70588aa3	62e38880	4b32cd23	7da56fd1
54827a61	d78e6b5f	17cce172	0ae88e5a	54827a62	d78e6b5e	f6942bb0	35a96499
232a8830	7f31780e	f0865b01	28cb4150	232a8a30	7f31740e	2ad851f7	362f33fb
39ba3bd2	277e9d52	316a7411	c8dbc618	39ba3bd3	27829d53	d239cc6e	29aa1db7

Our result

Output difference			
00000000	00000000	00000000	00000003
00000003	00000001	e158cac2	3f41eac3
00000200	00000c00	da5e0af6	1ee472ab
00000001	00fc0001	e353b87f	e171dbaf

For a cost of 2^{32}, we have for BMW-256:

- Collision for 300 pre-specified bits
  - Generic cost: 2^{150}
- Near-collision with 122 active bits
  - Generic cost: 2^{55}

Similar results for BMW-512.

New Improvement

Can we get a small difference in Q_{30} using degrees of freedom?

Chaining Value							
59dfd94b	30b036e3	44ad8a65	47461712	59dfd94b	30b036e2	bb52759b	b8b9e8ed
6f56e9b4	425e2d65	40000003	94662f58	90a9164c	bda1d29a	bffffffc	94662f58
12848c76	24f94ccd	4f74ffd3	3ec30f93	12848c76	24f94ccd	b08b002c	c13cf06c
8b0f9f9b	7071a4a5	4552a192	b30f47f5	74f06064	7071a4a5	4552a192	b30f47f5

Message							
bd050fb4	c6925351	991aa15f	60327d4b	bd050fb4	c6925351	991aa15f	60327d4b
0212e457	9feb065e	d6ab8dac	7bd2f8ca	0212e457	9feb065e	d6ab8dac	7bd2f8ca
2fcaa474	2c505ccd	2043dc85	7b0eac19	2fcaa474	2c505ccd	2043dc85	7b0eac19
08fe0408	01c2f910	74478ade	da5b35ba	08fe0408	01c6f910	8bb87521	25a4ca5a

Output difference			
00000000	00000000	00000000	00000007
03cc0005	03c70000	43610ac2	4728125a
98000601	34000001	08de7209	81246c5b
00c40007	00c10000	7f32d109	9300111e

- Complexity $\approx 2^{32}$
- 112 active bits
- Generic near-collision: 2^{64}

New Improvement

Can we get a small difference in Q_{30} using degrees of freedom?

Chaining Value							
59dfd94b	30b036e3	44ad8a65	47461712	59dfd94b	30b036e2	bb52759b	b8b9e8ed
6f56e9b4	425e2d65	40000003	94662f58	90a9164c	bda1d29a	bffffffc	94662f58
12848c76	24f94ccd	4f74ffd3	3ec30f93	12848c76	24f94ccd	b08b002c	c13cf06c
8b0f9f9b	7071a4a5	4552a192	b30f47f5	74f06064	7071a4a5	4552a192	b30f47f5

Message							
bd050fb4	c6925351	991aa15f	60327d4b	bd050fb4	c6925351	991aa15f	60327d4b
0212e457	9feb065e	d6ab8dac	7bd2f8ca	0212e457	9feb065e	d6ab8dac	7bd2f8ca
2fcaa474	2c505ccd	2043dc85	7b0eac19	2fcaa474	2c505ccd	2043dc85	7b0eac19
08fe0408	01c2f910	74478ade	da5b35ba	08fe0408	01c6f910	8bb87521	25a4ca5a

Output difference			
00000000	00000000	00000000	00000007
03cc0005	03c70000	43610ac2	4728125a
98000601	34000001	08de7209	81246c5b
00c40007	00c10000	7f32d109	9300111e

- Complexity $\approx 2^{32}$
- 112 active bits
- Generic near-collision: 2^{64}

Better near-collision

- For a cost of 2^{64}, we can get a collision in XH, with near-collisions in Q_{30}, Q_{31}
- This should give near-collision with about 64 active bits:
  - Small differences in HH_3 to HH_{13}
  - Random differences in HH_{14} and HH_{15}
- A generic near-collision attack with 64 active bits would cost

$$\sqrt{2^{512} \Big/ \binom{512}{64}} \approx 2^{119}$$

Conclusion

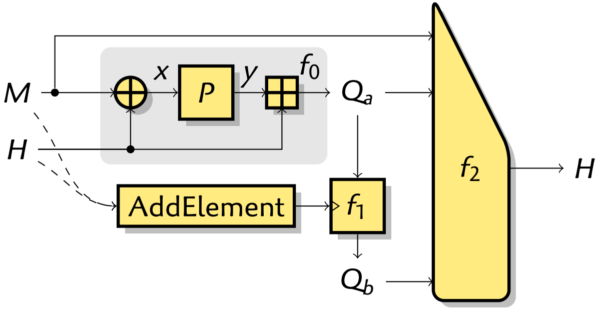
- Tools to solve AX system
- Path avoiding most of the rotations in BMW
  - Using degrees of freedom
  - Making some rotations inactive

Results (BMW-256 compression function)

- Partial-collisions
  - 300 chosen bits in 2^{32}
- Near-collisions:
  - 400 bits in 2^{32}
  - 450? bits in 2^{64}

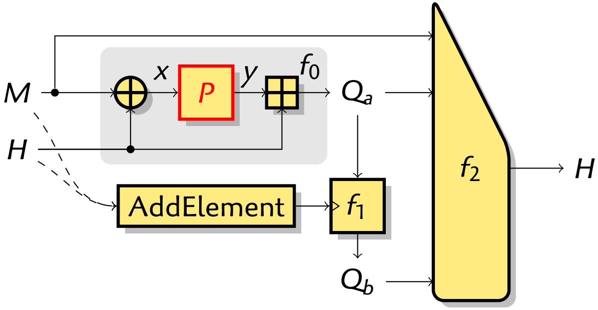
Appendix

Blue Midnight Wish



- Wide pipe: each line is 16 × 32 bits
- ARX: Addition, Rotations, Xors

Blue Midnight Wish

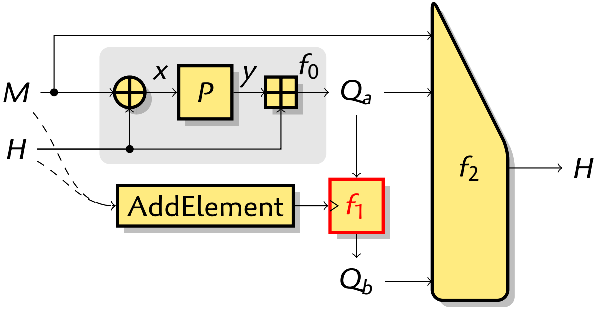


The P permutation

- $z_0 = x_5 \boxminus x_7 \boxplus x_{10} \boxplus x_{13} \boxplus x_{14}$
- $z_1 = x_6 \boxminus x_8 \boxplus x_{11} \boxplus x_{14} \boxminus x_{15}$
- …

- $y_0 = z_0^{\gg 1} \oplus z_0^{\ll 3} \oplus z_0^{\lll 4} \oplus z_0^{\lll 19}$
- $y_1 = z_1^{\gg 1} \oplus z_1^{\ll 2} \oplus z_1^{\lll 8} \oplus z_1^{\lll 23}$
- …

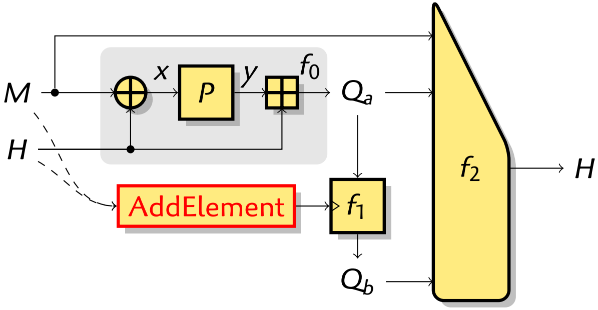
Blue Midnight Wish



The f_1 function: FSR

- $Q_{16} = s_1(Q_0) \boxplus s_2(Q_1) \boxplus \ldots \boxplus s_0(Q_{15}) \boxplus \mathrm{AddElement}(16)$
- $Q_{17} = s_1(Q_1) \boxplus s_2(Q_2) \boxplus \ldots \boxplus s_0(Q_{16}) \boxplus \mathrm{AddElement}(17)$
- $\ldots$

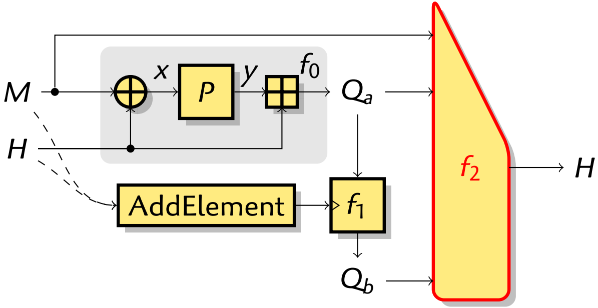
Blue Midnight Wish



The AddElement function

- $\text{AddElement}(16) = (M_0^{\lll 1} \boxplus M_3^{\lll 4} \boxminus M_{10}^{\lll 11} \boxplus K_{16}) \oplus H_7$
- $\text{AddElement}(17) = (M_1^{\lll 2} \boxplus M_4^{\lll 5} \boxminus M_{11}^{\lll 12} \boxplus K_{17}) \oplus H_8$
- ...

Blue Midnight Wish



The f_2 function: $XL = \bigoplus_{i=16}^{23} Q_i, \quad XH = \bigoplus_{i=16}^{31} Q_i$

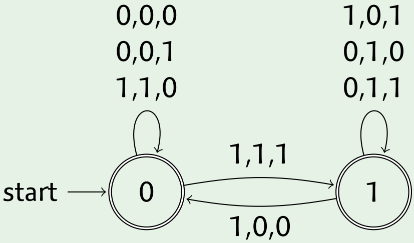
- $HH_0 = (XH^{\gg 5} \oplus Q_{16}^{\gg 5} \oplus M_0) \boxplus (XL \oplus Q_{24} \oplus Q_0)$
- ...
- $HH_8 = HH_4^{\lll 9} \boxplus (XH \oplus Q_{24} \oplus M_8) \boxplus (XL^{\ll 8} \oplus Q_{23} \oplus Q_8)$

Transition Automata

We use automata to study AX systems: [Mouha *et. al*]

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

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



$\Delta = 1110$

$\delta = 1010$

$x = 0111$

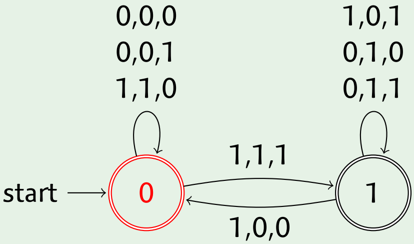
Fails

Transition Automata

We use automata to study AX systems: [Mouha *et. al*]

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

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



$\Delta = 1110$

$\delta = 1010$

$x = 0111$

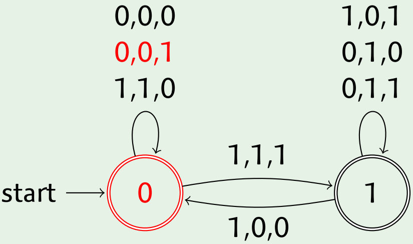
Fails

Transition Automata

We use automata to study AX systems: [Mouha *et. al*]

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

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



$\Delta = 1110$

$\delta = 1010$

$x = 0111$

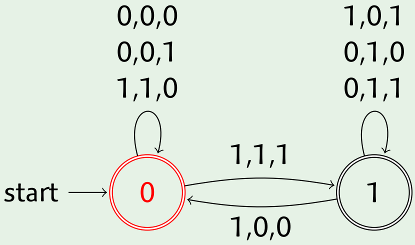
Fails

Transition Automata

We use automata to study AX systems: [Mouha *et. al*]

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

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



$\Delta = 1110$

$\delta = 1010$

$x = 0111$

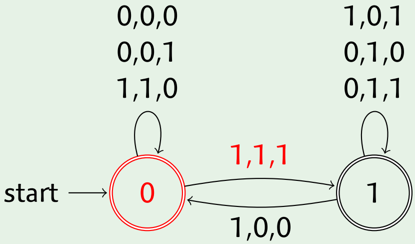
Fails

Transition Automata

We use automata to study AX systems: [Mouha *et. al*]

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

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



$\Delta = 1110$

$\delta = 1010$

$x = 0111$

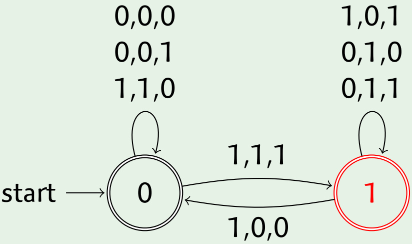
Fails

Transition Automata

We use automata to study AX systems: [Mouha *et. al*]

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

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



$\Delta = 1110$

$\delta = 1010$

$x = 0111$

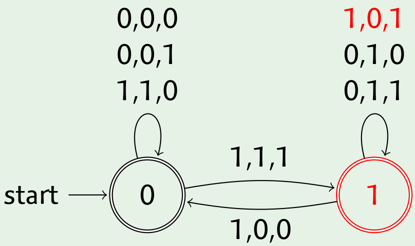
Fails

Transition Automata

We use automata to study AX systems: [Mouha *et. al*]

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

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



$\Delta = 1110$

$\delta = 1010$

$x = 0111$

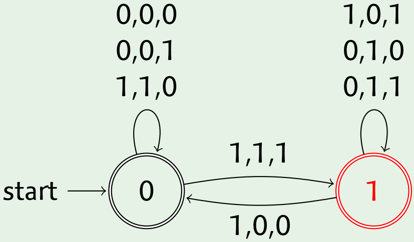
Fails

Transition Automata

We use automata to study AX systems: [Mouha *et. al*]

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

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



$\Delta = 1110$

$\delta = 1010$

$x = 0111$

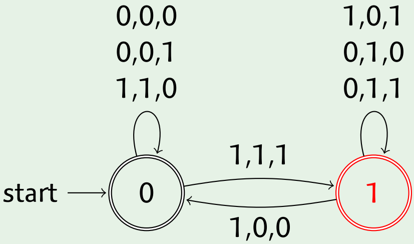
Fails

Transition Automata

We use automata to study AX systems: [Mouha *et. al*]

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

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



$\Delta = 1110$

$\delta = 1010$

$x = 0111$

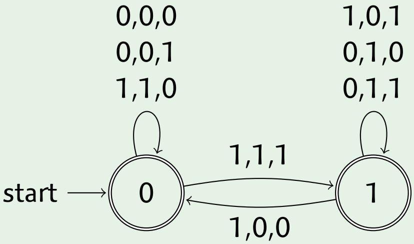
Fails

Transition Automata

We use automata to study AX systems: [Mouha *et. al*]

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

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



$\Delta = 1110$

$\delta = 1010$

$x = 0111$

Fails