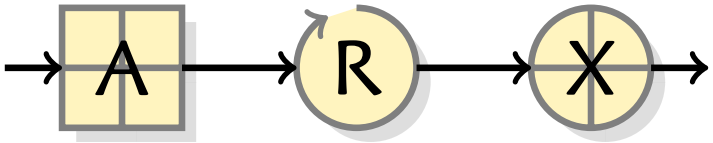
Construction of Differential Characteristics in ARX Designs

Application to Skein

Gaëtan Leurent

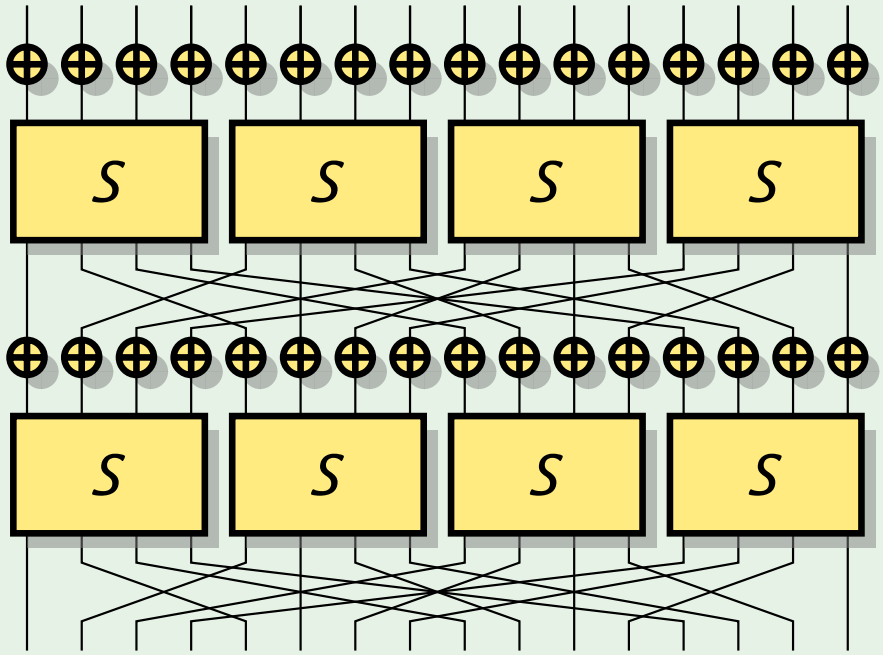
UCL Crypto Group

Crypto 2013

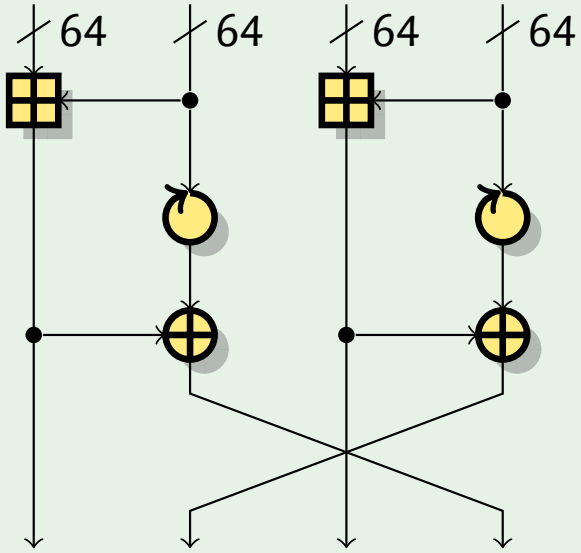


Symmetric key designs: two main categories

SmallPresent



Threefish



Symmetric key designs: two main categories

SBox designs

- S-Boxes and Linear Layers
- Important example: AES
- Few heavy rounds
- S S-Boxes
- Wire-crossing
- M MDS matrices

ARX designs

- Additions, Rotations, Xors (32/64-bit words)
- Inspired by MD/SHA
- Lots of light rounds
- ⊞ Addition
- ↻ Rotation
- ⊕ Xor

Addition, Rotation, Xor

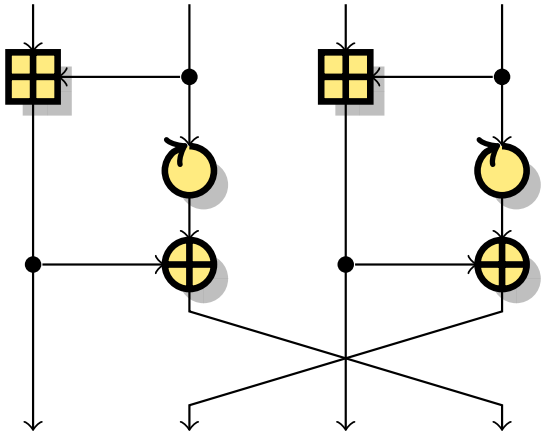
- Interaction between incompatible structures:
  - $\mathbb{Z}_{2^n}$-linear: Addition $\boxplus$
  - $\mathbb{F}_2$-linear: Rotation $\circlearrowleft$, Xor $\oplus$
- Very efficient designs: Salsa20/12, BLAKE2, Simon/Speck

ARX designs

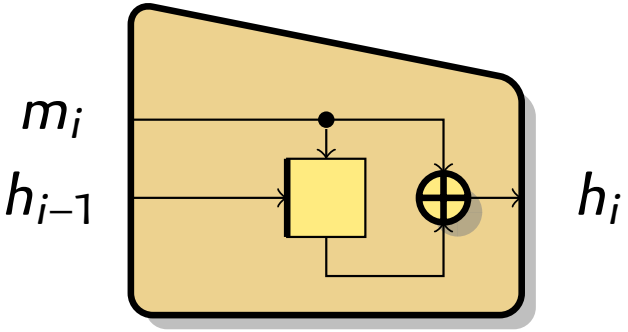
Hash functions	Skein, BLAKE (2 of the 5 SHA-3 finalists)
Stream ciphers	Slasa20, ChaCha
Block ciphers	TEA, XTEA, HIGHT, Speck
PRF	SipHash

- ARX with bitwise Boolean function: MD/SHA, Simon

Skein



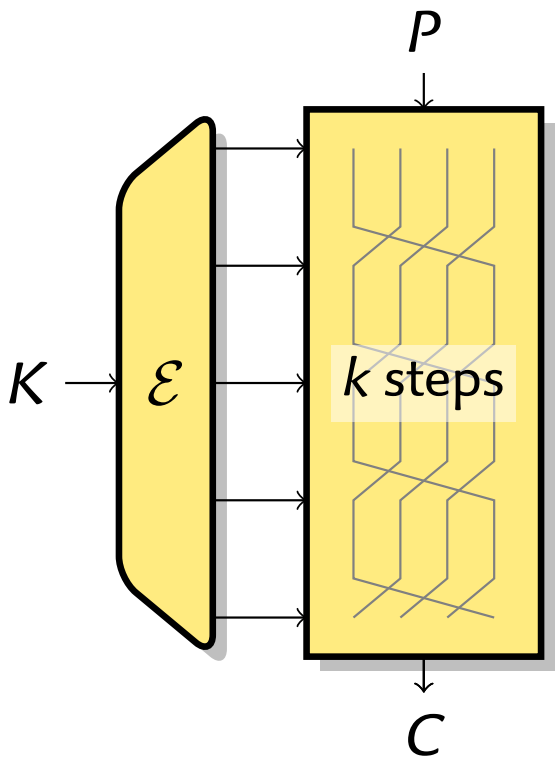
Threefish-256 round



MMO mode

- ARX design
  - 64-bit words
  - $\mathtt{MIX}_r(a,b) := ((a \boxplus b), (b \lll r) \oplus c)$
  - Word permutations
  - Key addition every four rounds
- Threefish-256:
  - 256-bit key: K_0, K_1, K_2, K_3
  - 128-bit tweak: T_0, T_1
  - 256-bit text
- MMO mode
  - Chaining value is the key

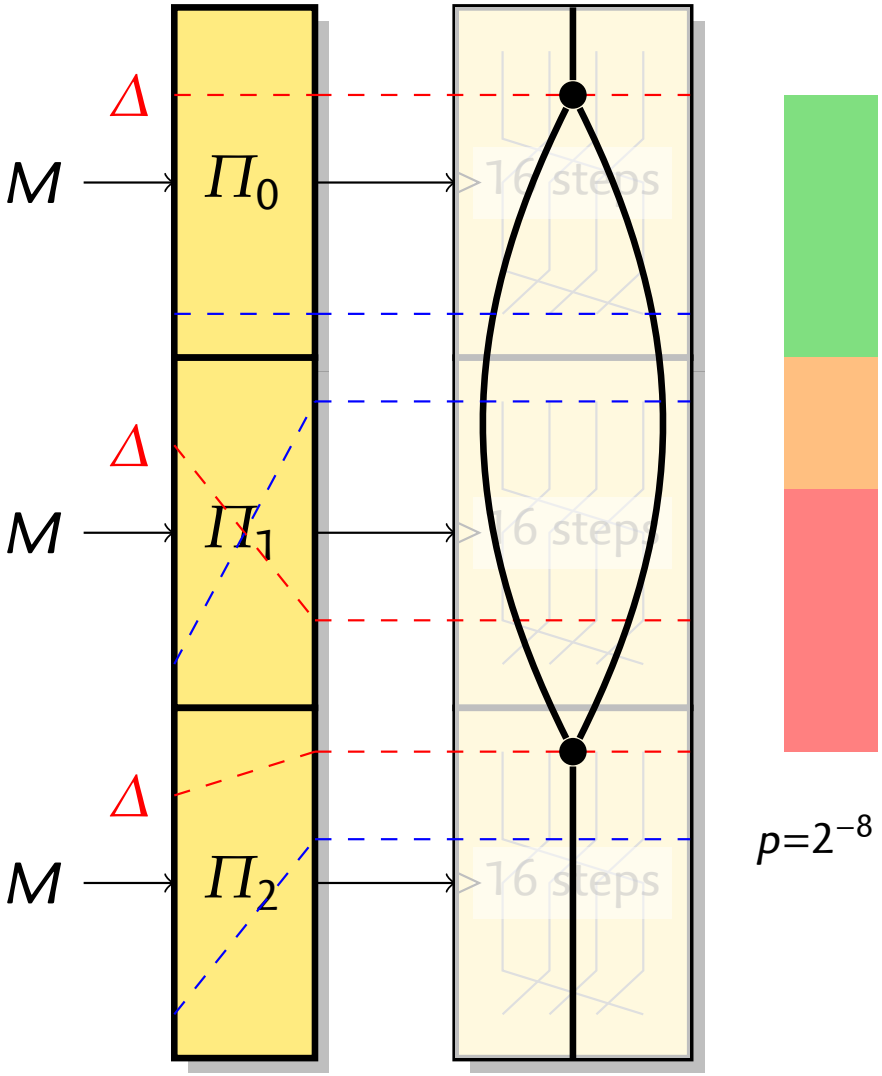
Differential attacks



- Iterated structure

- Take an input pair P, P'
  $C = E_K(P)$, $C' = E_K(P')$
- Look for Δ_P, Δ_C with large p:
  $p = \Pr[\Delta_P \rightsquigarrow \Delta_C]$
  $= \Pr[C' = C + \Delta_C \mid P' = P + \Delta_P]$
- Specify Δ_{X_i} at each step:
  $\Delta_P \rightsquigarrow \Delta_{X_1} \rightsquigarrow \Delta_{X_2} \rightsquigarrow \cdots \rightsquigarrow \Delta_C$
- $\Pr\left[\Delta_{X_0} \rightsquigarrow \Delta_{X_n}\right] \geq \prod_i \Pr\left[\Delta_{X_i} \rightsquigarrow \Delta_{X_{i+1}}\right]$

Differential collision attack



[Chabaud & Joux, Crypto 1998]
[Wang & al, Crypto & EC 2005]

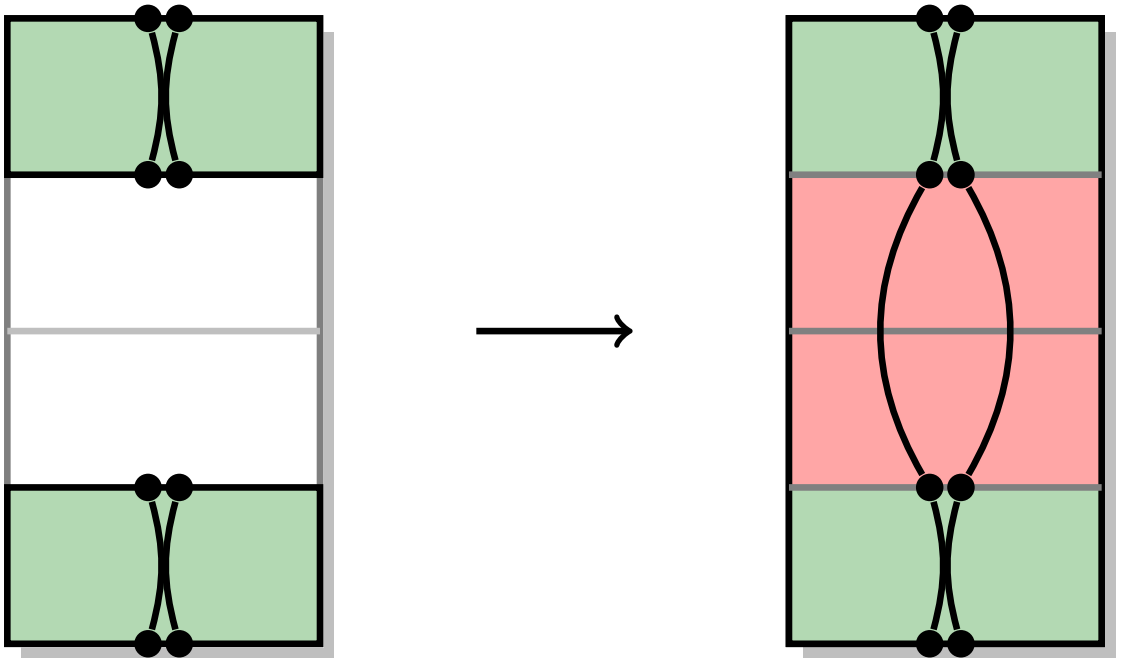
1. Precomputation:
   - Choose a message difference.
   - Build a differential path.
   - Derive a set of sufficient conditions.
2. Collision search:
   - Start with a random message, check the conditions
   - Use message modifications

Differential attacks against ARX

- Most of the cryptanalysis of ARX designs is bit-twiddling
  - As opposed to SBox based designs
- Building/verifying differential trails for ARX designs is hard
  - Many trails built by hand
  - Problems with several attacks
  - Hard to evaluate a design
- Later, automatic search
  - Mostly for MD/SHA designs. Pure ARX harder?
  - Better paths
  - New applications: HMAC attacks, rogue certificates
- Not all tools are public

Main Setting

- We target hash-function attacks
- We aim to connect two high-probability trails
- We will use degrees of freedom on the low probability section



Main Setting

- We target hash-function attacks
- We aim to connect two high-probability trails
- We will use degrees of freedom on the low probability section

Using the algorithm

1. Set input/output difference, and key difference
   - Select simple high probability trails by hand
2. Algorithm find intermediate difference
   - Complex trail in the middle
3. Find a pair of input values
   - Easy using degree of freedom

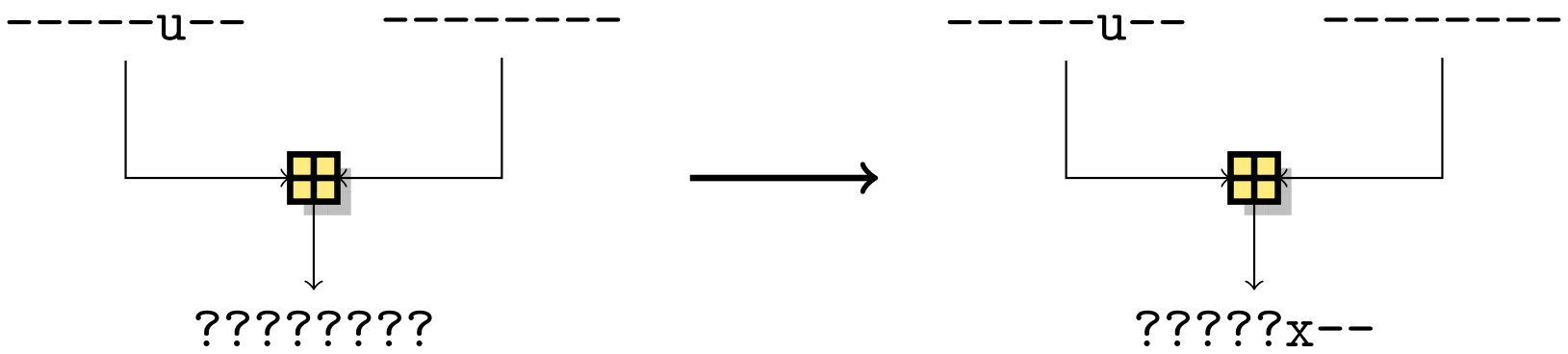
Propagation

- We want to propagate information:

- Input difference given
- Goal: infer output difference

Propagation

- We want to propagate information:



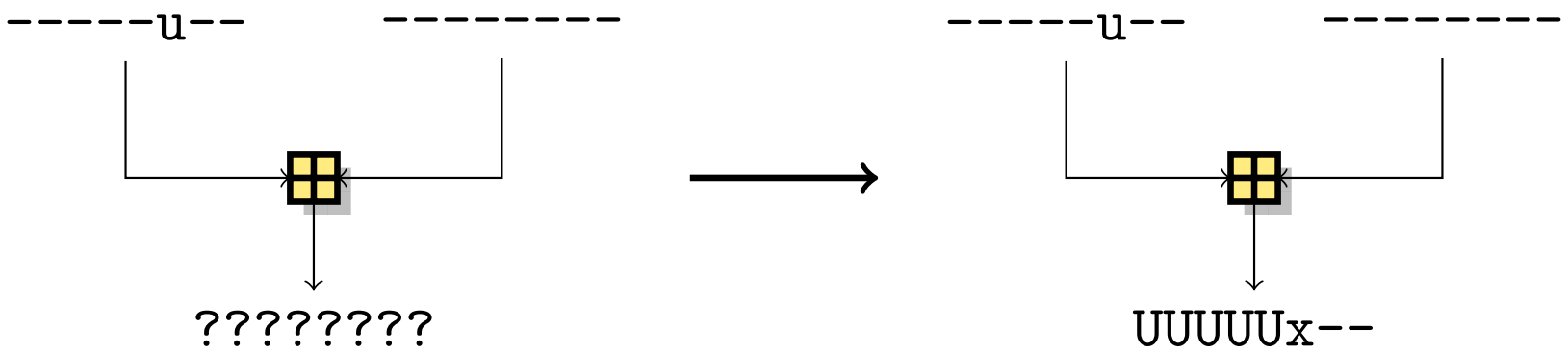
- Input difference given
- Goal: infer output difference

With single-bit constraints:

- We don't know if there is a carry
- Output bits can be active or inactive

Propagation

- We want to propagate information:



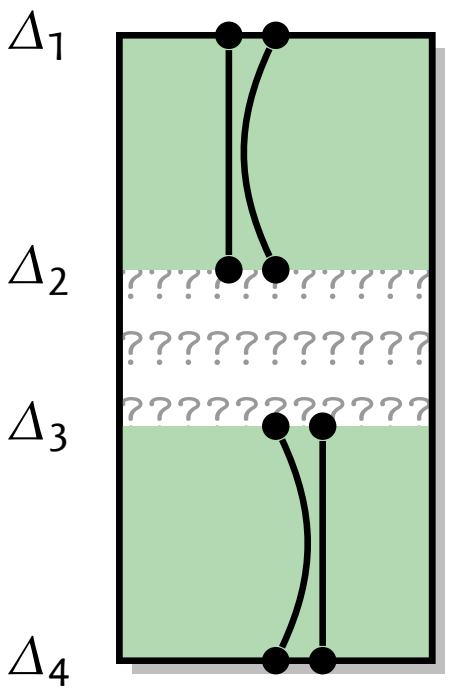
- Input difference given
- Goal: infer output difference

With multi-bit constraints:

- Carry bit can be active only if previous bit is active:
  - `x` if previous bit is `n`
  - `-` if previous bit is `-` or `u`

Algorithm

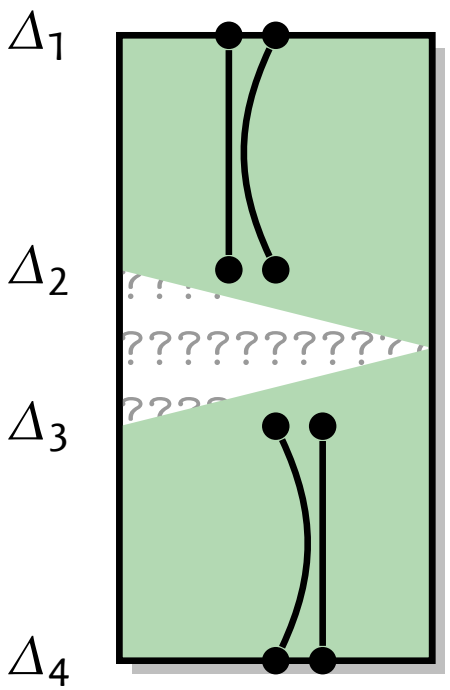
- Guess active bits in the middle and propagate
- Propagation will add necessary constraints (forced guess)



1. Initial characteristic
2. Propagation
3. Guessing
4. Propagation
5. ...
6. Final characteristic

Algorithm

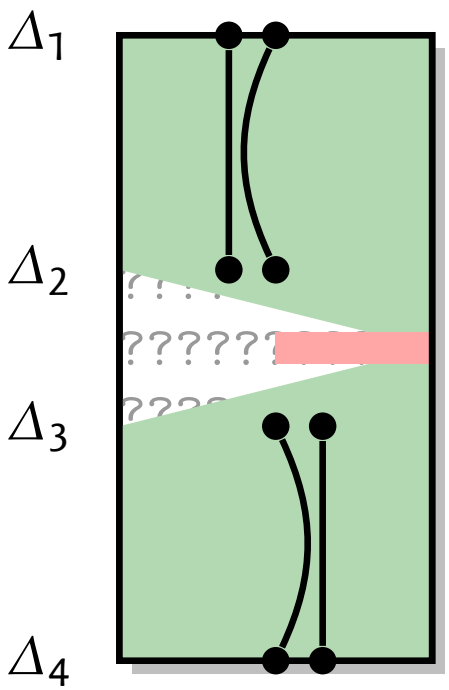
- Guess active bits in the middle and propagate
- Propagation will add necessary constraints (forced guess)



1. Initial characteristic
2. Propagation
3. Guessing
4. Propagation
5. ...
6. Final characteristic

Algorithm

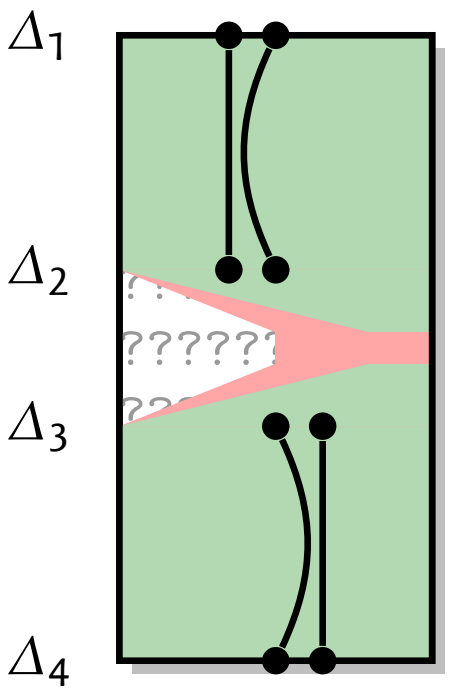
- Guess active bits in the middle and propagate
- Propagation will add necessary constraints (forced guess)



1. Initial characteristic
2. Propagation
3. Guessing
4. Propagation
5. ...
6. Final characteristic

Algorithm

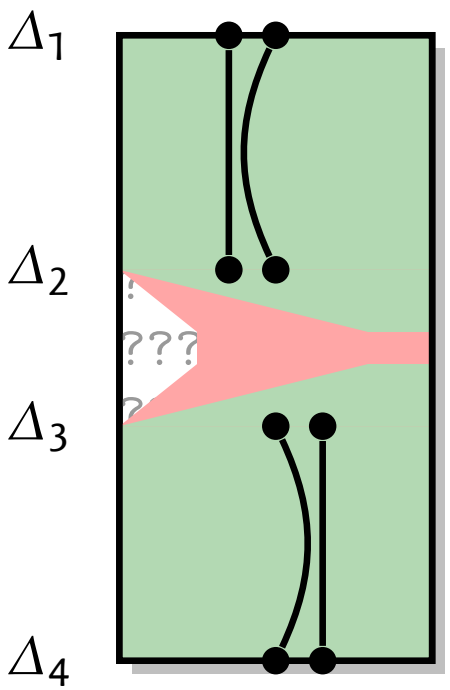
- Guess active bits in the middle and propagate
- Propagation will add necessary constraints (forced guess)



1. Initial characteristic
2. Propagation
3. Guessing
4. Propagation
5. ...
6. Final characteristic

Algorithm

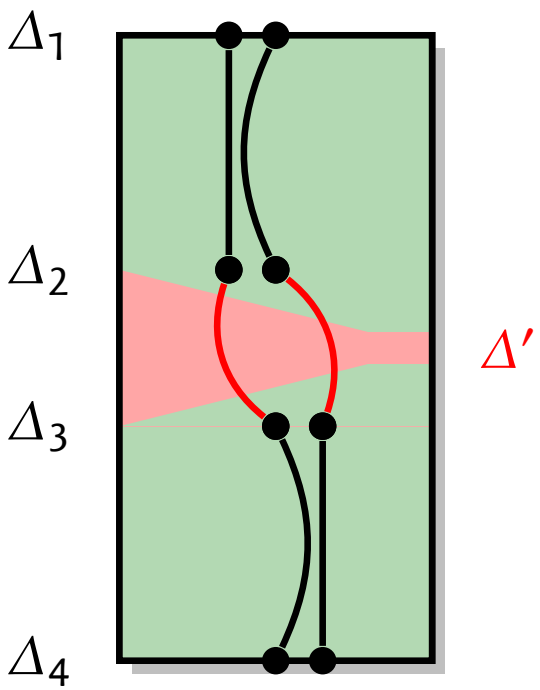
- Guess active bits in the middle and propagate
- Propagation will add necessary constraints (forced guess)



1. Initial characteristic
2. Propagation
3. Guessing
4. Propagation
5. ...
6. Final characteristic

Algorithm

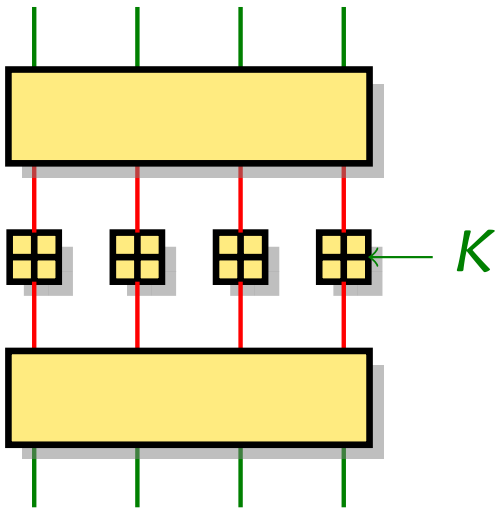
- Guess active bits in the middle and propagate
- Propagation will add necessary constraints (forced guess)



1. Initial characteristic
2. Propagation
3. Guessing
4. Propagation
5. ...
6. Final characteristic

Degrees of freedom

- Without degree of freedom, connecting trails does not make sense
  - For a fixed permutation, one pair on average with a given input/output difference
- Use key addition as the meeting point:



Fixed sparse difference (input)

Guessed
Fixed sparse difference (input)
Guessed

Fixed sparse difference (input)

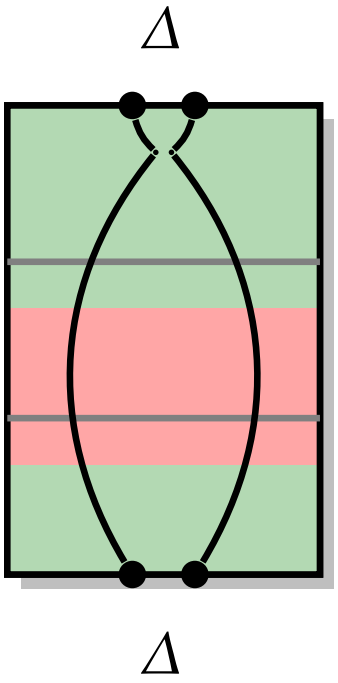
General results

- Some tweaking necessary
  - Number of rounds in the search section
  - Search parameters

Extra tricks

- We specify in advance the words to be guessed
- We guess from LSB to MSB
- Use backtracking, stop after some time
- When it fails, remember the best guess and restart
  - simulated annealing

Semi-free-start Collision Attack

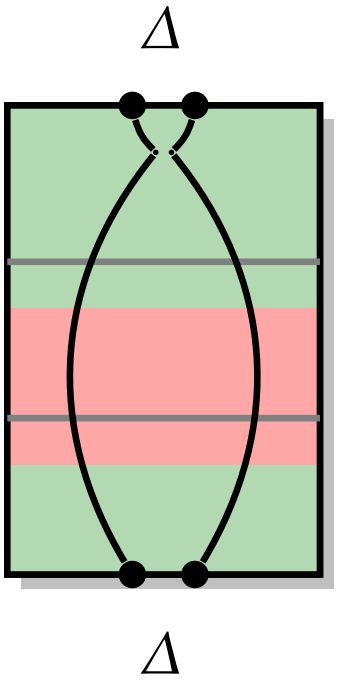


- Trails with no key difference
- Select a small difference Δ in the state
  - Build a trail $\Delta \rightarrow \Delta$
  - Collisions with the feed-forward
- Algorithm finds 12-round characteristics
- Practical attack

Limitations

- Dense path: low probability
- Many key conditions
  - Only valid for some IVs.
  - Semi-free-start collision.

Semi-free-start Collision Attack



- Trails with no key difference
- Select a small difference Δ in the state
  - Build a trail $\Delta \rightarrow \Delta$
  - Collisions with the feed-forward
- Algorithm finds 12-round characteristics
- Practical attack

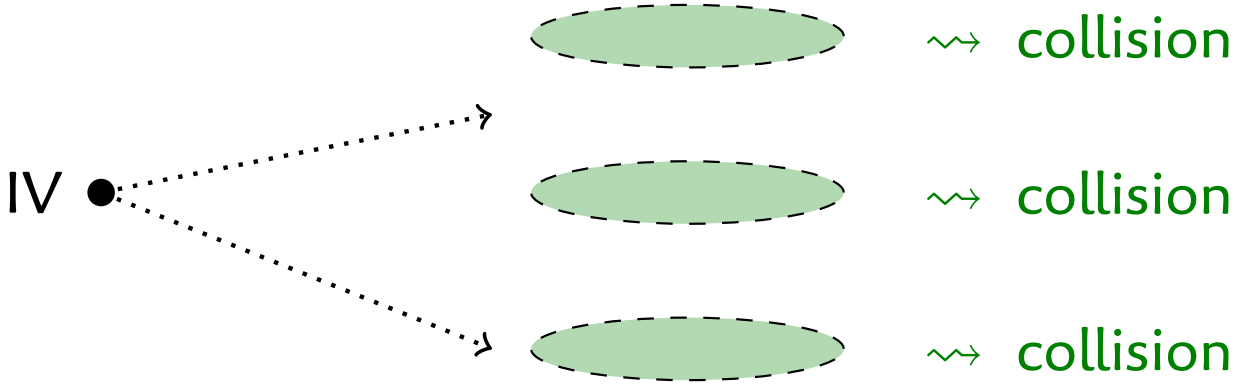
Limitations

- Dense path: low probability
- Many key conditions
  - Only valid for some IVs.
  - Semi-free-start collision.

Full Collision Attack

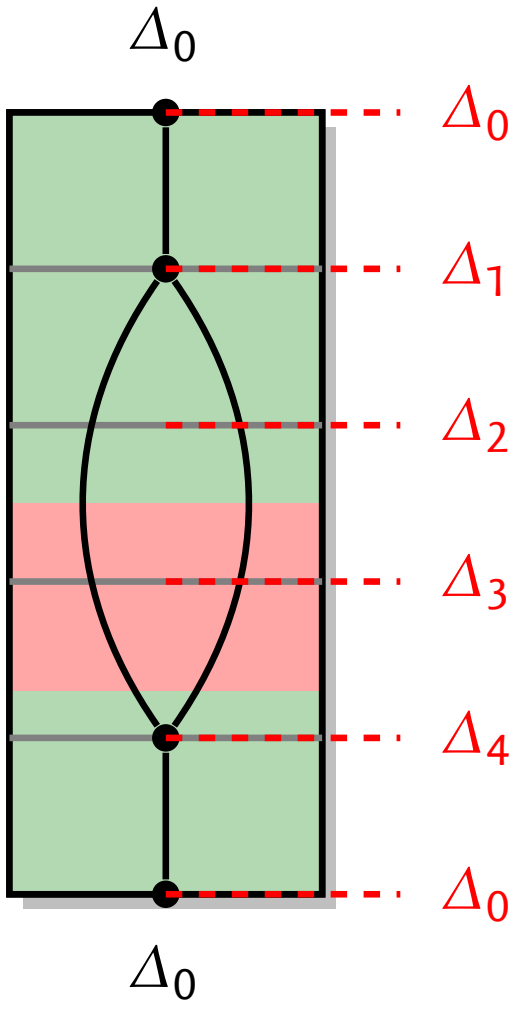
- We build a collision characteristic valid for 2^{106} keys for a cost of $\approx 2^{50}$

1. Build many characteristics (2^{50})
2. Use random message blocks to reach a valid CV for one path.



- Collision attack for 12-round Skein-256 with complexity $\approx 2^{100}$

Free-start Collision Attack



- Trails with small key difference
- This allows inactive rounds
- The key schedule repeats after 5 block
  - Collisions with the feed-forward

- Algorithm finds 20-round characteristics
- Practical attack

Our results

1. Automatic search of differential trails for ARX functions
   - Multi-bit constraints
   - Guess in the middle
   - Simulated annealing

2. Application to Skein-256
   - Collisions for 12 rounds: complexity $\approx 2^{100}$
   - Semi-free-start collisions for 12 rounds: practical
   - Free-start collisions for 20 rounds: practical

   - Huge security margin: 72 rounds for full version

3. Code available:
   `http://www.di.ens.fr/~leurent/arxtools.html`

Thanks

Questions?

With the support of ERC project CRASH

European Research Council
Established by the European Commission
Supporting top researchers from anywhere in the world