Analyse de primitives symétriques

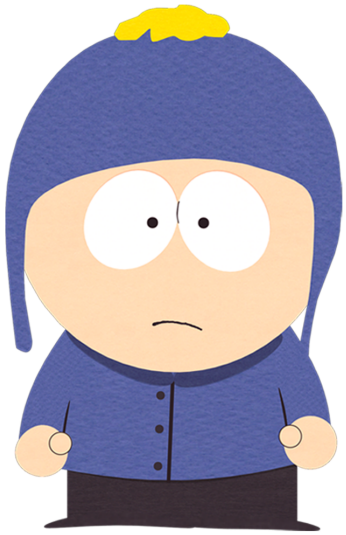
Pierre Karpman

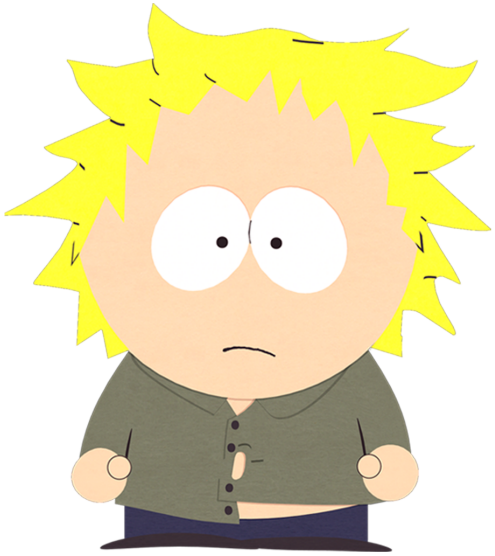
Thèse préparée à l'Inria Saclay & Rennes, l'École polytechnique, et la Nanyang Technological University

Sous la direction de Daniel Augot, Pierre-Alain Fouque et Thomas Peyrin

Palaiseau
2016–11–18

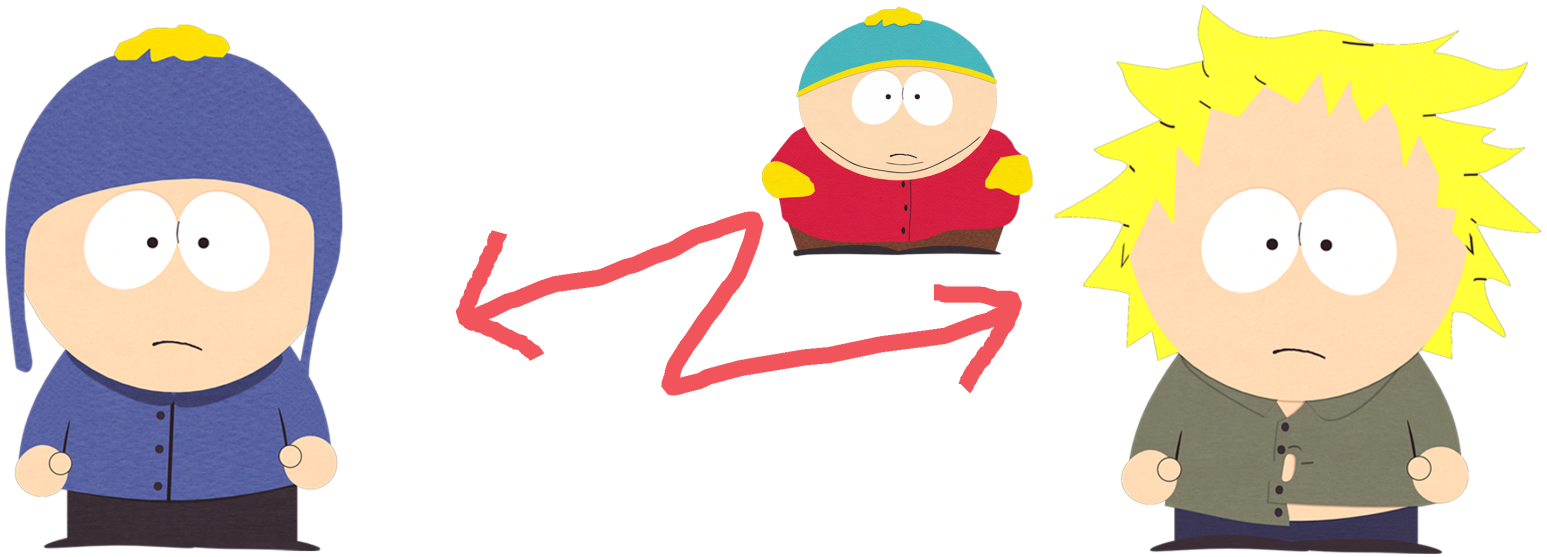
Motivating cryptography



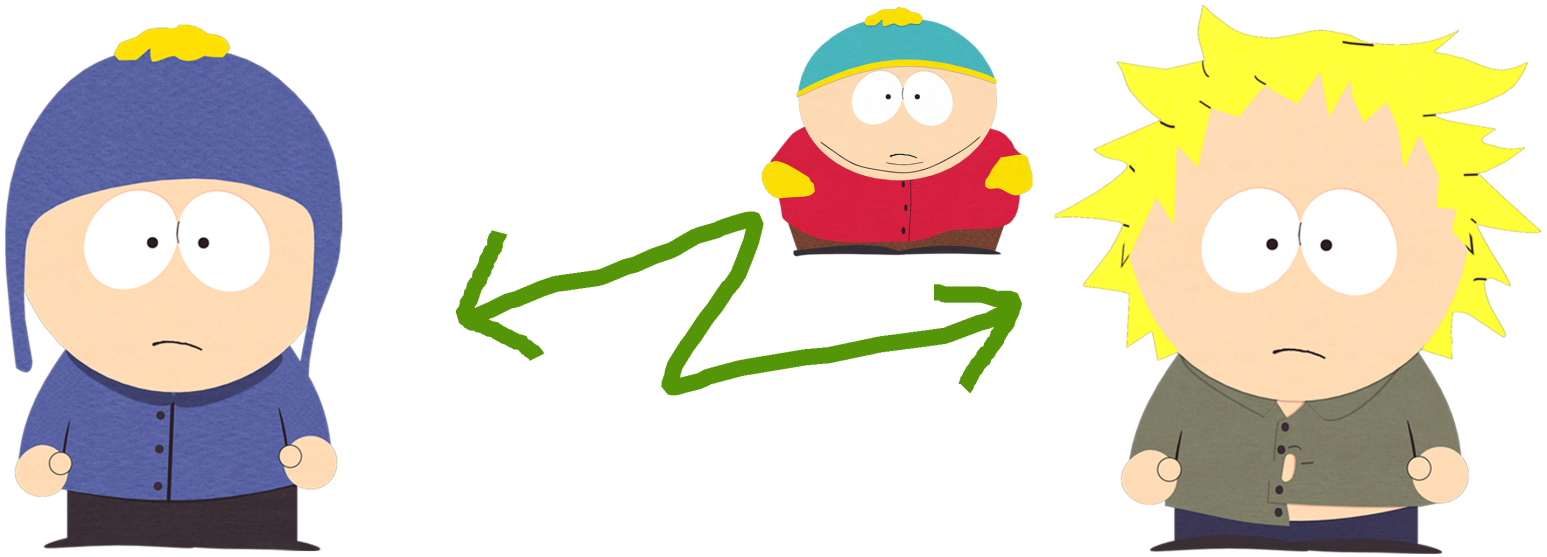


Motivating cryptography

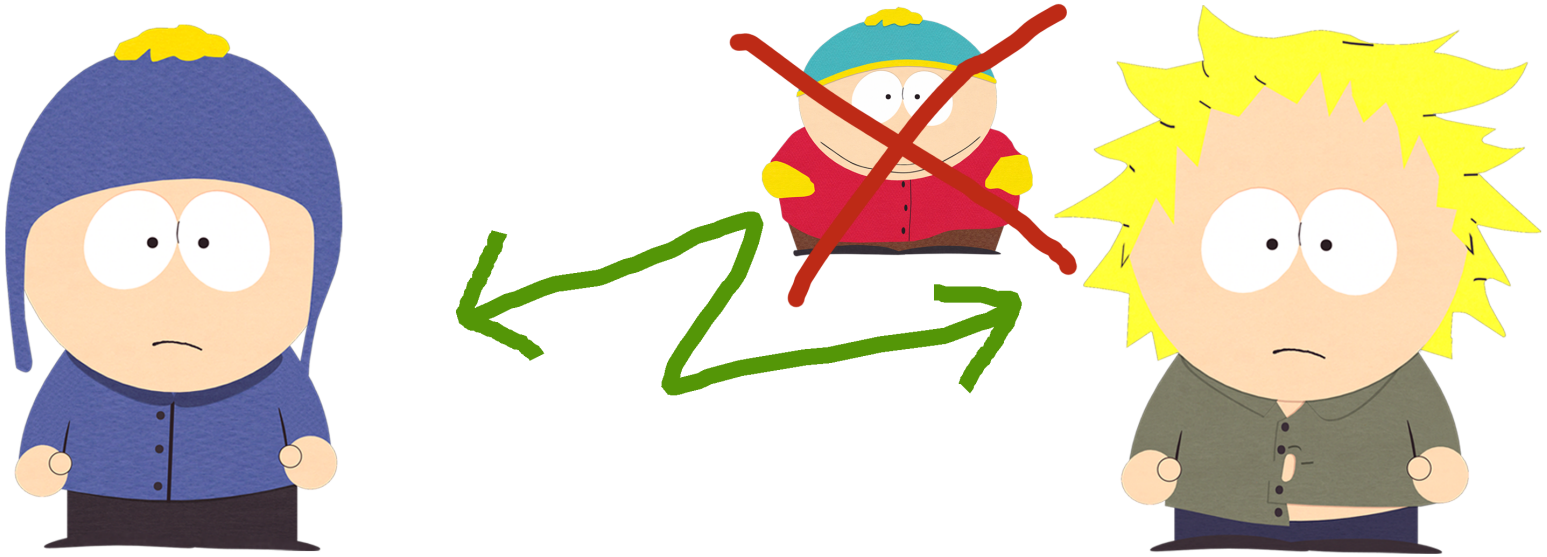
Motivating cryptography



Motivating cryptography



Motivating cryptography



A hierarchy of cryptographic components

A protocol (e.g. TLS) uses among others

- A key exchange algorithm (e.g. Diffie-Hellman)
  — "public-key" cryptography
  - instantiated with a secure group (e.g. ANSSI FRP256V1)
- An authenticated-encryption mode of operation (e.g. GCM)
  — "symmetric-key" cryptography
  - instantiated with a secure block cipher (e.g. the AES)
- A digital signature algorithm (e.g. ECDSA)
  — "public-key" + "symmetric-key" cryptography
  - instantiated with a secure group and a secure hash function (e.g. SHA-3)

Primitive-centered crypto in a nutshell

- Design new primitives
  - Fast, lightweight, quantum-resistant (isogeny-based, etc.), ...
- Analyse new proposals
- Analyse standards
  - AES, SHA-{1,2,3}, ...

Focus of this thesis

We studied various aspects of

- design
- analysis
- implementation

of block ciphers and hash functions

List of publications

Efficient and Provable White-Box Primitives
with Pierre-Alain Fouque, Paul Kirchner and Brice Minaud
ASIACRYPT 2016

Freestart collision for full SHA-1
with Thomas Peyrin and Marc Stevens
EUROCRYPT 2016

Key-Recovery Attacks on ASASA
with Patrick Derbez, Pierre-Alain Fouque and Brice Minaud
ASIACRYPT 2015

From Distinguishers to Key Recovery: Improved Related-Key Attacks on Even-Mansour
ISC 2015

Practical Free-Start Collision Attacks on 76-step SHA-1
with Thomas Peyrin and Marc Stevens
CRYPTO 2015

Higher-Order Differential Meet-in-The-Middle Preimage Attacks on SHA-1 and BLAKE
with Thomas Espitau and Pierre-Alain Fouque
CRYPTO 2015

Diffusion Matrices from Algebraic-Geometry Codes with Efficient SIMD Implementation
with Daniel Augot and Pierre-Alain Fouque
SAC 2014

List of publications

Efficient and Provable White-Box Primitives
with Pierre-Alain Fouque, Paul Kirchner and Brice Minaud
ASIACRYPT 2016

Freestart collision for full SHA-1
with Thomas Peyrin and Marc Stevens
EUROCRYPT 2016

Key-Recovery Attacks on ASASA
with Patrick Derbez, Pierre-Alain Fouque and Brice Minaud
ASIACRYPT 2015

From Distinguishers to Key Recovery: Improved Related-Key Attacks on Even-Mansour
ISC 2015

Practical Free-Start Collision Attacks on 76-step SHA-1
with Thomas Peyrin and Marc Stevens
CRYPTO 2015

Higher-Order Differential Meet-in-The-Middle Preimage Attacks on SHA-1 and BLAKE
with Thomas Espitau and Pierre-Alain Fouque
CRYPTO 2015

Diffusion Matrices from Algebraic-Geometry Codes with Efficient SIMD Implementation
with Daniel Augot and Pierre-Alain Fouque
SAC 2014

Introduction

New linear mappings for block ciphers

Practical freestart collisions for the full SHA-1

Conclusion

Block ciphers

Block cipher

A block cipher is a mapping $\mathcal{E} : \mathcal{K} \times \mathcal{M} \to \mathcal{C}$ s.t. for all $k \in \mathcal{K}$, $\mathcal{E}(k, \cdot)$ is invertible and $\mathcal{E}(k, \cdot)$, $\mathcal{E}^{-1}(k, \cdot)$ are efficiently computable.

In most cases:

- $\mathcal{M} = \mathcal{C} = \{0, 1\}^n$, $n \in \{64, 128\}$
- $\mathcal{K} = \{0, 1\}^\kappa$, $\kappa \in \{64, 80, 128, 256\}$

Security of block ciphers

Ideal block cipher

Key-recovery security

Differential cryptanalysis (Biham and Shamir, 1990)

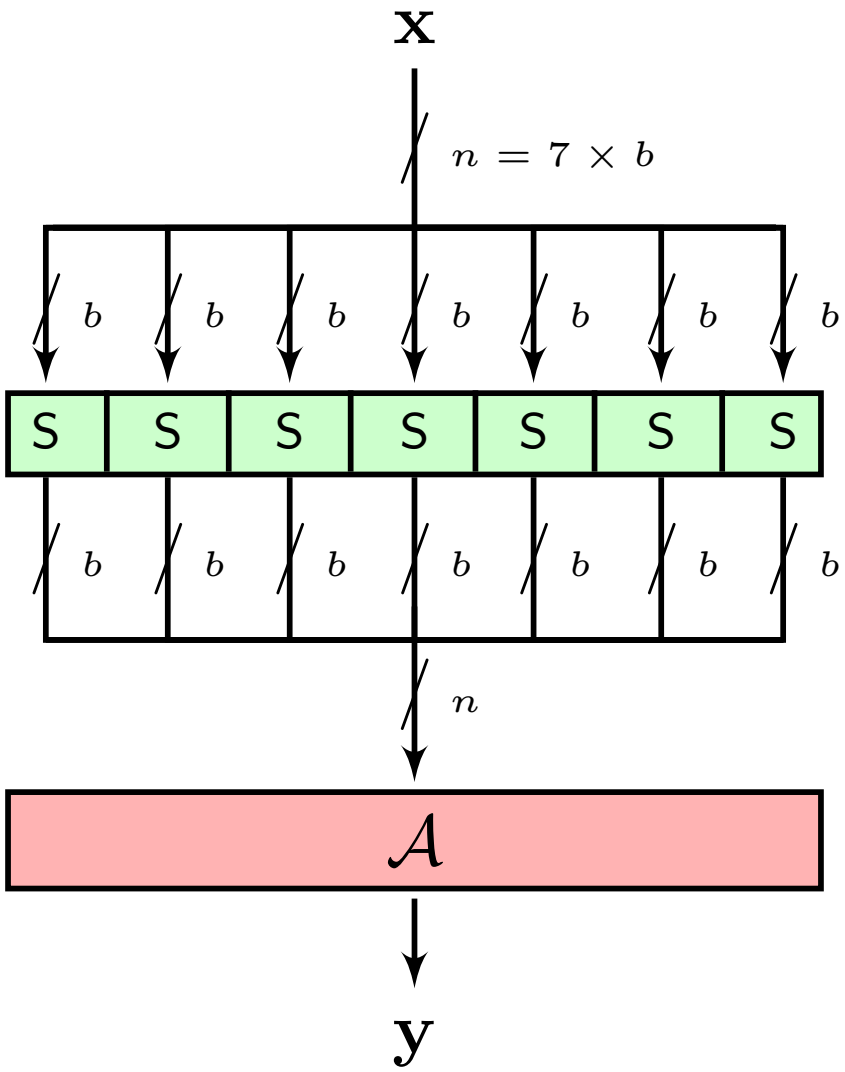
Exploit statistical properties of $x \mapsto \mathcal{E}(k, x) \oplus \mathcal{E}(k, x \oplus \Delta)$

Substitution-Permutation-Network block ciphers

SPN round function: $\mathcal{R} = \mathcal{A} \circ \mathcal{S}$

- $\mathcal{S}$ is the parallel application of s b-bit *S-boxes*
- $\mathcal{A}$ is an affine transformation over $\mathbf{F}_2^n$, $n = s \times b$

An SPN round function in a picture



The wide-trail strategy (Daemen, 1995)

- Use structure in $\mathcal{A}$ and $\mathcal{S}$ to lower-bound the number of *active* S-boxes in a differential characteristic or linear approximation
- Introduce a notion of *diffusion* for a round function
- Canonical example: the AES (Daemen & Rijmen, 2002)

New linear mappings for block ciphers

Diffusion Matrices from Algebraic-Geometry Codes with Efficient SIMD Implementation,
with Daniel Augot and Pierre-Alain Fouque,
SAC 2014

The objective

- Find $\mathcal{A}$ over $\mathbf{F}_{2^b}^s$ with very good *diffusion*, with b small (i.e. 4)

A diffusion measure: the branch number (Daemen 1995)

Let $M \in \mathcal{M}_s(\mathbf{F}_{2^b})$, $\mathbf{x} \in \mathbf{F}_{2^b}^s$ and $\mathrm{wt}(\mathbf{x})$ be the number of non-zero coordinates of $\mathbf{x}$.
The *differential branch number* of M is

$$\min_{\mathbf{x} \neq \mathbf{0}} (\mathrm{wt}(\mathbf{x}) + \mathrm{wt}(M\mathbf{x}))$$

The *linear branch number* of M is

$$\min_{\mathbf{x} \neq \mathbf{0}} (\mathrm{wt}(\mathbf{x}) + \mathrm{wt}(M^t\mathbf{x}))$$

Branch number & minimal distance

- $M \in \mathcal{M}_s(\mathbf{F}_{2^b})$ has differential branch number d
  $\Leftrightarrow [I_s \; M]$ generates a $[2s, s, d]_{\mathbf{F}_{2^b}}$ code
- Singleton bound: d is at most $s+1$; equality $\Rightarrow$ MDS code, MDS matrix
- MDS conjecture: there is no $[2s, s, s+1]_{\mathbf{F}_{2^b}}$ MDS code with $2s > 2^b$

The idea

- Take for $\mathcal{A}$ a single M with high branch number (SHARK structure, Rijmen et al., 1996)
- For $b = 4$, $s \geq 16$ (block size ≥ 64), M cannot be MDS
- $\Rightarrow$ Use *Algebraic geometry codes*: trade length for minimal distance (Goppa, 1981), (Tsfasman,Vlăduț,Zink, 1982)

AG codes

AG codes as evaluation codes:

- Say we want an $[n, k]_{\mathbf{F}_q}$ code

1. Let $\mathcal{X}$ be a smooth plane curve of genus g with $\#\mathcal{X}(\mathbf{F}_q) > n$
2. Inject $\mathbf{F}_q^k$ to $\mathcal{L}(rP)$ of dim. k with $P \in \mathcal{X}(\mathbf{F}_q)$
3. $\mathcal{C}(m)$ is the evaluation of m on n distinct points of $\mathcal{X} \backslash \{P\}$
4. For well-chosen n and k, $r = k - 1 + g$ (Riemann, Roch)
$$\Rightarrow \mathrm{wt}(\mathcal{C}(m \neq 0)) \geq n - (k - 1 + g)$$

- $\Rightarrow$ the min. distance is g less than MDS, but $\mathcal{X}$ can be chosen s.t. $\#\mathcal{X}(\mathbf{F}_q) > q + 1$

A concrete AG code

- Let $\mathcal{X}$ be of equation $x^5 = y^2z^3 + yz^4$ in $\mathbf{P}^2(\mathbf{F}_{2^4})$
- It is a maximal curve of genus 2 and has 33 points
- $\Rightarrow$ We can define a $[32, 16, 15]_{\mathbf{F}_{2^4}}$ code $\mathcal{C}$
- This gives many (up to 32!) $M \in \mathcal{M}_{16}(\mathbf{F}_{2^4})$ of diff. branch number 15
- The dual also has min. distance 15 $\Rightarrow$ M has lin. branch number 15

Implementation matters

For M to be used as $\mathcal{A}$ in a block cipher, we need:

- Efficient implementations of multiplication by M...
- ... That are "constant-time", to protect against side-channel attacks

Thus we:

- Defined good *vectorized* algorithms for $\times M$ using `pshufb`
- Optimized the structure of M for faster multiplication

A cost function for $\times M$: #`pshufb`

- Combinatorial in nature
- Low cost can be obtained if M is generated by a single row

Tuning M for fast implementations

- Idea: use the automorphisms of $\mathcal{C}$ to find a circulant M
- Result: no luck, but still got M generated by 8 rows (cost 52)
- Second idea: randomly sample many generating matrices ($\approx 2^{38}$)
- Result: found many matrices of cost 43

Structured block M (cost 52)

$$\begin{pmatrix}
5 & 2 & 1 & 3 & 8 & 5 & 1 & 5 & 12 & 10 & 14 & 6 & 7 & 11 & 4 & 11 \\
2 & 2 & 4 & 1 & 5 & 12 & 2 & 1 & 9 & 15 & 8 & 11 & 7 & 6 & 9 & 3 \\
1 & 4 & 4 & 3 & 1 & 2 & 15 & 4 & 5 & 13 & 10 & 12 & 9 & 6 & 7 & 13 \\
3 & 1 & 3 & 3 & 5 & 1 & 4 & 10 & 14 & 2 & 14 & 8 & 15 & 13 & 7 & 6 \\
8 & 5 & 1 & 5 & 5 & 2 & 1 & 3 & 7 & 11 & 4 & 11 & 12 & 10 & 14 & 6 \\
5 & 12 & 2 & 1 & 2 & 2 & 4 & 1 & 7 & 6 & 9 & 3 & 9 & 15 & 8 & 11 \\
1 & 2 & 15 & 4 & 1 & 4 & 4 & 3 & 9 & 6 & 7 & 13 & 5 & 13 & 10 & 12 \\
5 & 1 & 4 & 10 & 3 & 1 & 3 & 3 & 15 & 13 & 7 & 6 & 14 & 2 & 14 & 8 \\
12 & 9 & 5 & 14 & 7 & 7 & 9 & 15 & 7 & 6 & 11 & 3 & 15 & 5 & 13 & 7 \\
10 & 15 & 13 & 2 & 11 & 6 & 6 & 13 & 6 & 6 & 7 & 9 & 5 & 10 & 2 & 14 \\
14 & 8 & 10 & 14 & 4 & 9 & 7 & 7 & 11 & 7 & 7 & 6 & 13 & 2 & 8 & 4 \\
6 & 11 & 12 & 8 & 11 & 3 & 13 & 6 & 3 & 9 & 6 & 6 & 7 & 14 & 4 & 12 \\
7 & 7 & 9 & 15 & 12 & 9 & 5 & 14 & 15 & 5 & 13 & 7 & 7 & 6 & 11 & 3 \\
11 & 6 & 6 & 13 & 10 & 15 & 13 & 2 & 5 & 10 & 2 & 14 & 6 & 6 & 7 & 9 \\
4 & 9 & 7 & 7 & 14 & 8 & 10 & 14 & 13 & 2 & 8 & 4 & 11 & 7 & 7 & 6 \\
11 & 3 & 13 & 6 & 6 & 11 & 12 & 8 & 7 & 14 & 4 & 12 & 3 & 9 & 6 & 6
\end{pmatrix}$$

Unstructured fast M (cost 43)

$$\begin{pmatrix}
11 & 6 & 1 & 6 & 10 & 14 & 10 & 9 & 13 & 3 & 3 & 12 & 9 & 15 & 2 & 9 \\
6 & 12 & 0 & 4 & 2 & 8 & 9 & 2 & 5 & 11 & 9 & 5 & 4 & 1 & 15 & 6 \\
9 & 11 & 2 & 2 & 1 & 11 & 13 & 15 & 13 & 3 & 2 & 1 & 14 & 1 & 3 & 10 \\
0 & 0 & 9 & 8 & 11 & 6 & 2 & 1 & 11 & 10 & 15 & 10 & 10 & 15 & 1 & 14 \\
13 & 13 & 3 & 15 & 3 & 1 & 11 & 2 & 9 & 2 & 10 & 14 & 1 & 11 & 1 & 2 \\
1 & 9 & 8 & 4 & 14 & 10 & 2 & 5 & 15 & 2 & 12 & 12 & 9 & 10 & 1 & 9 \\
5 & 9 & 11 & 2 & 15 & 1 & 12 & 4 & 6 & 0 & 6 & 4 & 5 & 8 & 2 & 9 \\
1 & 4 & 14 & 9 & 13 & 2 & 10 & 12 & 0 & 6 & 6 & 9 & 2 & 0 & 11 & 10 \\
13 & 10 & 3 & 9 & 2 & 15 & 6 & 6 & 11 & 1 & 9 & 9 & 12 & 14 & 10 & 3 \\
0 & 10 & 6 & 12 & 11 & 0 & 4 & 9 & 1 & 14 & 10 & 2 & 9 & 2 & 13 & 6 \\
2 & 0 & 5 & 6 & 9 & 0 & 1 & 5 & 15 & 12 & 13 & 15 & 1 & 11 & 13 & 11 \\
11 & 2 & 10 & 1 & 1 & 15 & 0 & 8 & 0 & 9 & 14 & 10 & 10 & 6 & 11 & 15 \\
12 & 14 & 10 & 11 & 3 & 10 & 6 & 0 & 5 & 11 & 1 & 8 & 2 & 9 & 2 & 3 \\
15 & 2 & 2 & 5 & 1 & 10 & 9 & 4 & 1 & 8 & 9 & 9 & 12 & 10 & 14 & 12 \\
15 & 1 & 12 & 5 & 13 & 11 & 0 & 6 & 2 & 5 & 11 & 1 & 15 & 0 & 9 & 13 \\
5 & 6 & 11 & 0 & 2 & 9 & 14 & 11 & 12 & 10 & 3 & 2 & 8 & 10 & 3 & 1
\end{pmatrix}$$

Performance and applications

- Eight $(\mathcal{A} \circ \mathcal{S})$ rounds should resist attacks
- $\Rightarrow$ On Sandy Bridge, ≈ 30 cycles per byte for good M (AVX assembly)
- Not so fast for general-purpose...
- ... But only needs 128 S-box applications (≈ 512 N.L. gates)

- Potential application: cipher suitable for masking at very high order, e.g. beat Mysterion (Journault, Standaert, Varici, 2016)

Introduction

New linear mappings for block ciphers

Practical freestart collisions for the full SHA-1

Conclusion

Hash functions

Hash function

A hash function is a mapping $\mathcal{H} : \mathcal{M} \to \mathcal{D}$

In practice:

- $\mathcal{M} = \bigcup_{\ell < N} \{0,1\}^\ell$, $\mathcal{D} = \{0,1\}^n$, $N \gg n$
- Typically $N = 2^{64}$, $n \in \{128, 160, 224, 256, 384, 512\}$
- It is a keyless primitive
- What's a good hash function?

Three security notions

First preimage resistance

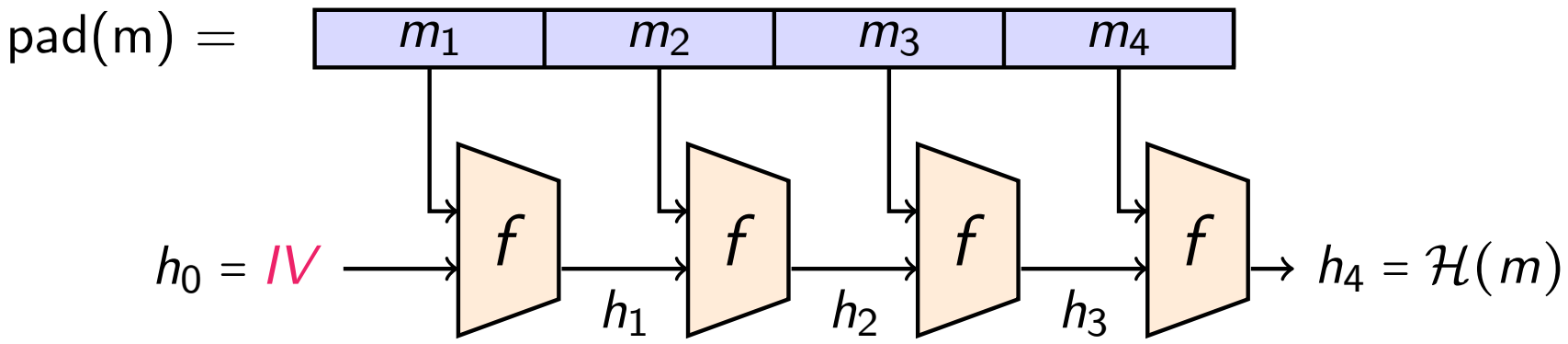
Second preimage resistance

Collision resistance

Find $m, m' \neq m$ s.t. $\mathcal{H}(m) = \mathcal{H}(m')$
Best generic attack is in $\mathcal{O}(2^{\frac{n}{2}})$

Merkle-Damgård construction in a picture

(Merkle, 1989), (Damgård, 1989)
Define variable-input-length $\mathcal{H}$ from fixed-input-length f:



Security reduction of $\mathcal{H}$ to f

Additional security notions for MD

$\mathcal{H}_i$: $\mathcal{H}$ with IV set to i

Freestart collisions

A freestart collision is a pair $((i, m), (i', m'))$ s.t. $\mathcal{H}_i(m) = \mathcal{H}_{i'}(m')$

Freestart collisions (variant)

Attack f instead of $\mathcal{H}$

Practical freestart collisions for SHA-1

Freestart collision for full SHA-1,
with Thomas Peyrin and Marc Stevens,
EUROCRYPT 2016

The objective

- Show that SHA-1 is *really* not secure
- Find a practical attack on the full SHA-1 for a well-defined security notion
- And implement it

The idea

- Collision attacks are near-practical (Wang, Yin, Yu, 2005)
- $\Rightarrow$ Move to a freestart model to make the attack faster

The SHA-1 hash function

- Designed by the NSA in 1995
- Hash size is 160 bits $\Rightarrow$ collision security should be 80 bits
- Compression function in Merkle-Damgård mode

SHA-1 compression function

Block cipher in Davies-Meyer mode

Block cipher: 5-branch ARX Feistel

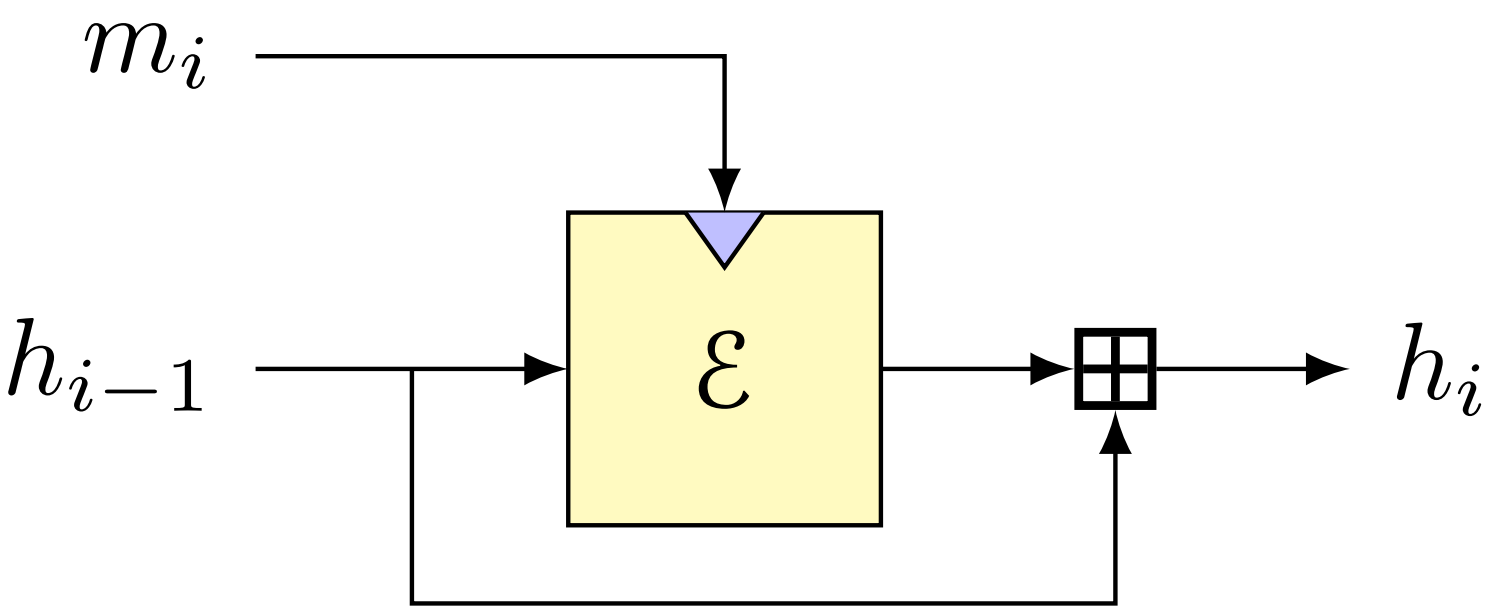
$$A_{i+1} = A_i^{\circlearrowleft 5} + \phi_{i \div 20}(A_{i-1}, A_{i-2}^{\circlearrowleft 2}, A_{i-3}^{\circlearrowleft 2}) + A_{i-4}^{\circlearrowleft 2} + W_i + K_{i \div 20}$$

with a linear message (key) expansion:

$$W_{0 \ldots 15} = M_{0 \ldots 15}, \; W_{i \geq 16} = (W_{i-3} \oplus W_{i-8} \oplus W_{i-14} \oplus W_{i-16})^{\circlearrowleft 1}$$

80 steps in total

Davies-Meyer construction in a picture



Collision attacks for the hash function

SHA-1 is not collision-resistant (Wang, Yin, Yu, 2005)

Attack complexity $\equiv 2^{69}$

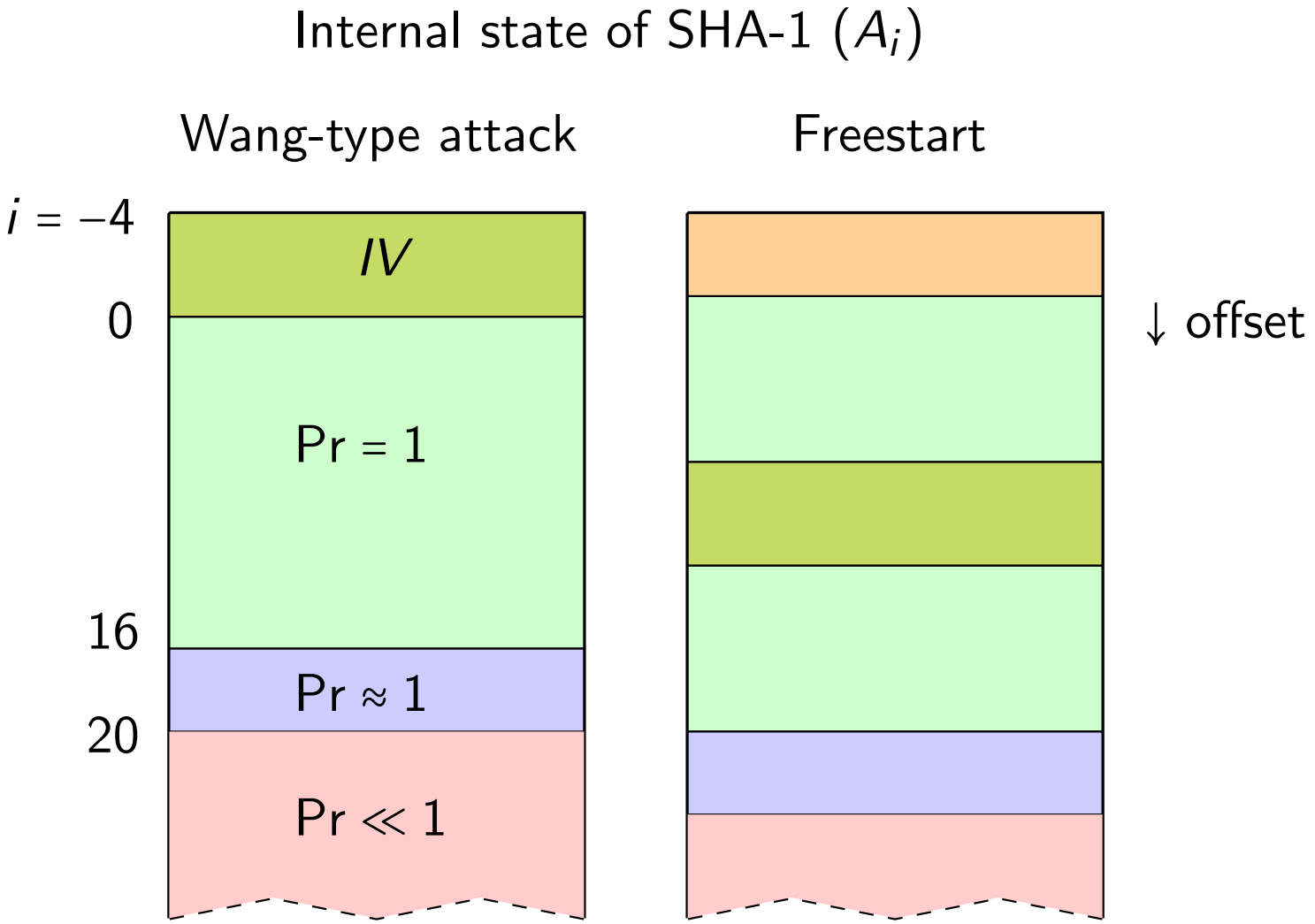
Eventually improved to $\equiv 2^{61}$ (Stevens, 2013)

Differential collision attack (Wang et al., 2005)

1. Find a *good linear* differential path for the message
2. Construct a *non-linear* diff. path to connect the *IV* to the linear path

Time for an attack

The point of freestart (in a picture)



The steps of the attack (Wang-type)

1. Find a good linear part (for the differential path)
2. Construct a good shifted non-linear part (for the diff. path)
3. Instantiate accelerating techniques

Let's do this for 80 steps!

Linear part selection

Criteria:

- High overall probability
- No (or few) differences in last five steps (= differences in IV)
- Few differences in early message words

$\Rightarrow$ Not many candidates

We picked II(59,0) (Manuel notation, 2011)

Linear path in a picture (last 20 steps)

i	A_i	W_i
61	●○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○	○○○○○○○○○○○○○○○○○○○○○○○○○○○●○○○○
62	○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○	○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○
63	●○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○	○○●○○○○○○○○○○○○○○○○○○○○○○○○○○○●○○○○
64	○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○	●○●○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○
65	○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○	○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○
66	○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○	○○●○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○
67	○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○	○○●○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○
68	○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○	○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○
69	○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○	○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○
70	○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○	○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○
71	○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○	○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○
72	○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○	○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○
73	○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○	○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○
74	○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○	○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○●○
75	○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○●○	○○○○○○○○○○○○○○○○○○○○○○○○○●○○○○○○
76	○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○	○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○●○
77	○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○	○●○○○○○○○○○○○○○○○○○○○○○○○○○○○○○●○
78	○○○○○○○○○○○○○○○○○○○○○○○○○○○○○●○○	○●○○○○○○○○○○○○○○○○○○○○○○○●○○○○○●○
79	○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○●○	○●○○○○○○○○○○○○○○○○○○○○○○○○○●○○○●○
80	○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○	

Non-linear part construction

- Start with prefix of high backward probability for the first 4 steps
- Use a mix of automated search for the rest

Non-linear path in a picture

i	A_i	W_i
-4	○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○	
-3	○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○	
-2	○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○☆▼○	
-1	△○○○△○○○○○○○○○○○○○○○○○○○○○○○○▽○○○○○▲	
0	▽△○○▽○○○○○○○○○○○○○○○○○○○○○○○○△○○○○○○	●○▲○○○▲○○○○○○○○○○○○○○○○○○○○○○○○▲○○○○
1	△△▲☆○○▲○○○○○○○○○○○○○○○○☆○○○○○○○▲○○○○	○○▼○○▼○○○○○○○○○○○○○○○○○○○○○○○○▼▲▲○○
2	○○▼△△▼△○△○○○○○○○☆○○○○○○△▲△△▽○△○▽○○	○○▲○○▼▼○○○○○○○○○○○○○○○○○○○○○○○○▼○▲○○
3	○▽○▽▼▽▽△△○☆○△▽○○○▲▽△○▽△△△△☆▽○△○△	○○▼○○▼▼○○○○○○○○○○○○○○○○○○○○○○○○▼▲○▼○
4	○△○△△▲▼△▲☆☆☆▲△☆☆☆▽△△☆☆○▼▲▲▲▲▲▼○▲	○○○○○○○○○○○○○○○○○○○○○○○○○○○○▲○○○○
5	○▲○▲○▼▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲▲○▲▽▼△△△△	○○○○○▼○○○○○○○○○○○○○○○○○○○○○○○○▲▲▲○○
6	○▽○▽○△○▽△△○△△△○△△△△▽▼▽△▽▽▼△○△▽▼▲	●▲○○▲▲○○○○○○○○○○○○○○○○○○○○○○○○▼○▲○○
7	△▼○▲○△○▽△▽△▽▽▽△▽▽▽▽▽▽▽△△△▲○▼○▽○▲	○○○○▼▲○○○○○○○○○○○○○○○○○○○○○○○○○○▲○
8	▽▲○▽○▽○○○○○○○○○○○○○○○○○○▽○○▲○▼○▽○△	●▼○○○○○○○○○○○○○○○○○○○○○○○○○○○○▲○○○○
9	○▲○▽○▽○○○○○○○○○○○○○○○○○○○○▽○▲○○○☆	●○▼▲○▼○○○○○○○○○○○○○○○○○○○○○○○○▼▲▲○○
10	○▲○○○○○○○○○○○○○○○○○○○○○○○○○○▲○▽○○	○○▼▲▲▲○○○○○○○○○○○○○○○○○○○○○○○○○○▼○○
11	○○○▼○○○○○○○○○○○○○○○○○○○○○○○○○○○○○	●○▲▲▲▲○○○○○○○○○○○○○○○○○○○○○○○○▼▲○▲○
12	○○○▽○△○○○○○○○○○○○○○○○○○○○○○○○○△○○	○○▼○○○○○○○○○○○○○○○○○○○○○○○○○○▼○○○○
13	○△○○○▽○○○○○○○○○○○○○○○○○○○○○○○○○★☆	○○▲○○▲○○○○○○○○○○○○○○○○○○○○○○○○▼▲▲○○
14	▲▼○○○○○○○○○○○○○○○○○○○○○○○○○○○○○○	●▲▲○▲▼○○○○○○○○○○○○○○○○○○○○○○○○▼○▲○○
15	△○△▼○○○○○○○○○○○○○○○○○○○○○○○○○○○★○	○○○○▲▼○○○○○○○○○○○○○○○○○○○○○○○○○○▲○
16	▲○△▽○△○○○○○○○○○○○○○○○○○○○○○○○○○○○	●▲○○○○○○○○○○○○○○○○○○○○○○○○○○○○▼○○○○
17	△○▼○○▽○○○○○○○○○○○○○○○○○○○○○○○○○○☆	●○▲▲○▲○○○○○○○○○○○○○○○○○○○○○○○○▲▼▼○○
18	○▲▼○▽○○○○○○○○○○○○○○○○○○○○○○○○○○○★	○○▲○▼▼○○○○○○○○○○○○○○○○○○○○○○○○○○▼○○
19	○▲○□○○○○○○○○○○○○○○○○○○○○○○○○○○○○○	●○▲▼▼▼○○○○○○○○○○○○○○○○○○○○○○○○▼▲○○○
20	▼○○○◆○○○○○○○○○○○○○○○○○○○○○○○○○○○○	●○▲▲○○○○○○○○○○○○○○○○○○○○○○○○○○▲○○○○

Accelerating techniques

Attack process:

- Generate many "partial solutions": message pairs following the diff. path up to some step
- Hope that one yields a collision

To make this efficient, use:

- Message modification (Wang et al., 2005)
- Neutral bits (Biham and Chen, 2004):
  Generate more good instances when one's found
- We choose neutral bits because:
  - Easy to find
  - Easy to implement

Neutral bits (with an offset)

- We start with an offset
- $\Rightarrow$ Use neutral bits with an offset too
- In our attack, offset $= 5$
  - free message words $=$ W5...20 instead of W0...15

Let's sum up

- Initialize the state with an offset
- Initialize message words with an offset
- Use neutral bits with an offset
- $\Rightarrow$ many neutral bits up to late steps (good!)
- $\Rightarrow$ don't know the IV in advance (well...)

If it's practical you must run it

- Attack expected to be practical, but still expensive
- Why not using GPUs?
- One main challenge: how to deal with the branching?

Architecture imperatives

Target platform: Nvidia GTX-970

- Execution is bundled in warps of 32 threads
- Single Instruction Multiple Threads:
  Control-flow divergence is serialized ⇒ minimize branching
- Hide latency by grouping warps into blocks

Our snippet-based approach

1. Store partial solutions up to some step in shared buffers
2. Every thread of a block loads one solution
3. ... tries all neutral bits for this step
4. ... stores successful candidates in next step buffer

GPU results

- Hardware: 64 GTX-970
- $\Rightarrow$ Expected time to find a collision $\lessapprox$ 10 days
  - Energy cost $\equiv$ boil 50 kL of 20°C water
- Complexity $\equiv 2^{57.5}$ SHA-1 compression function
- Conjectured 250-500 days for hash function collision with the same cluster

Summary of this thesis

The work done in this thesis included:

- Design of block ciphers
  - Fly (lightweight), PuppyCipher & CoureurDesBois (whitebox), (Samneric (large matrices))
- Analysis of encryption schemes
  - Prøst-OTR (related-key), ASASA family
- And of hash functions
  - SHA-1, BLAKE(2)

List of publications

Efficient and Provable White-Box Primitives
with Pierre-Alain Fouque, Paul Kirchner and Brice Minaud
ASIACRYPT 2016

Freestart collision for full SHA-1
with Thomas Peyrin and Marc Stevens
EUROCRYPT 2016

Key-Recovery Attacks on ASASA
with Patrick Derbez, Pierre-Alain Fouque and Brice Minaud
ASIACRYPT 2015

From Distinguishers to Key Recovery: Improved Related-Key Attacks on Even-Mansour
ISC 2015

Practical Free-Start Collision Attacks on 76-step SHA-1
with Thomas Peyrin and Marc Stevens
CRYPTO 2015

Higher-Order Differential Meet-in-The-Middle Preimage Attacks on SHA-1 and BLAKE
with Thomas Espitau and Pierre-Alain Fouque
CRYPTO 2015

Diffusion Matrices from Algebraic-Geometry Codes with Efficient SIMD Implementation
with Daniel Augot and Pierre-Alain Fouque
SAC 2014