From Collisions to Chosen-Prefix Collisions

Application to Full SHA-1

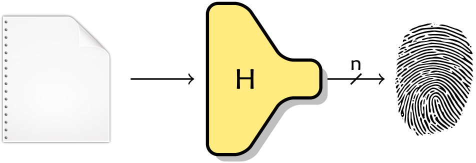
Gaëtan Leurent Thomas Peyrin

Inria, France

NTU, Singapour

Eurocrypt 2019

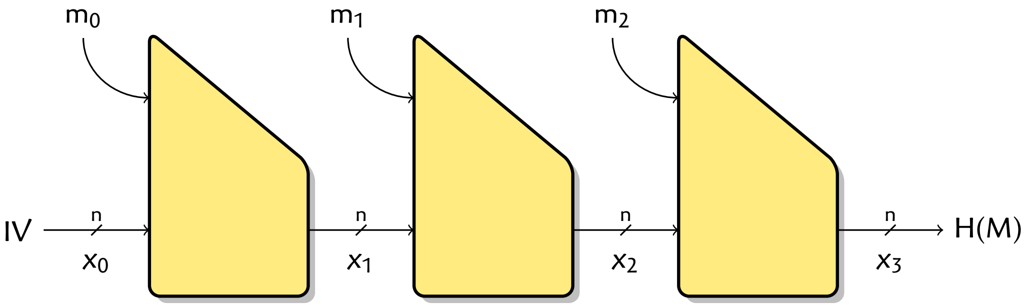
Hash functions



- Hash function: public function $\{0,1\}^* \to \{0,1\}^n$
  - Maps arbitrary-length message to fixed-length hash
- Hash function should behave like a random function
  - Hard to find collisions, preimages
  - Hash can be used as fingerprint, identifier
- Used in many different contexts
  - Signature: hash-and-sign
  - MAC: hash-and-PRF
  - Blockchain: Proof-of-work, ...

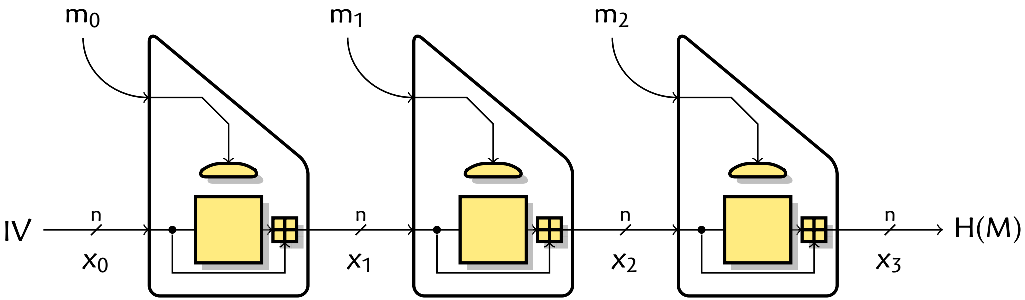
SHA-1

- Designed by NSA: SHA-0 [1993], then SHA-1 [1995]
- Standardized by NIST, ISO, IETF, ... Widely used until quite recently
- State size: $n = 160$
  - Expected collision security 2^{80}
- Iterative structure: Merkle-Damgård construction
- Block cipher-based compression function: Davies-Meyer



SHA-1

- Designed by NSA: SHA-0 [1993], then SHA-1 [1995]
- Standardized by NIST, ISO, IETF, ... Widely used until quite recently
- State size: $n = 160$
  - Expected collision security 2^{80}
- Iterative structure: Merkle-Damgård construction
- Block cipher-based compression function: Davies-Meyer



SHA-1 Cryptanalysis

2005-02 Theoretical collision with 2^{69} operations [Wang & al., Crypto'05]

… Several unpublished collision attacks in the range $2^{51} - 2^{63}$

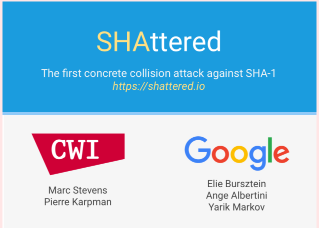
2010-11 Theoretical collision with 2^{61} operations [Stevens, EC'13]

2015-10 Practical freestart collision (on GPU) [Stevens, Karpman & Peyrin, Crypto'15]

2017-02 Practical collision with $2^{64.7}$ operations (on GPU) [Stevens & al., Crypto'17]

SHAttered attack: Colliding PDFs

SHA-1 =
`38762cf7f55934b34d17`
`9ae6a4c80cadccbb7f0a`



SHA-1 today

- Modern web browsers reject SHA-1 certificates since 2017
- SHA-1 certificates still exists
  - CAs still sell legacy SHA-1 certificates

- SHA-1 certificates still accepted by modern non-browser TLS clients
  - Until a few week ago, a mailserver in TU Darmsdtat used a SHA-1 certificate
  - Windows 10 "Mail" app connects without error

```
$ sslscan mail.sim.informatik.tu-darmstadt.de:993
[...]
  SSL Certificate:
Signature Algorithm: sha1WithRSAEncryption
```

- SHA-1 also used in Git, TLS 1.2 handshake, ...

SHA-1 today

- Modern web browsers reject SHA-1 certificates since 2017
- SHA-1 certificates still exists
  - CAs still sell legacy SHA-1 certificates

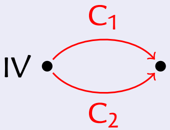
- SHA-1 certificates still accepted by modern non-browser TLS clients
  - Until a few week ago, a mailserver in TU Darmsdtat used a SHA-1 certificate
  - Windows 10 “Mail” app connects without error

```
$ sslscan mail.sim.informatik.tu-darmstadt.de:993
[...]
  SSL Certificate:
Signature Algorithm: sha1WithRSAEncryption
```

- SHA-1 also used in Git, TLS 1.2 handshake, ...

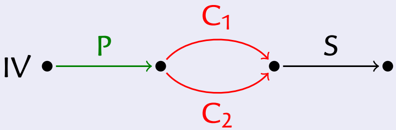
Exploiting collisions

Collision attack



- Start from IV
- C_1 and C_2 collide

Adding prefix and suffix



- Add identical prefix and suffix using iterative structure
- Usually same difficulty (just a different IV)

- Issue: C_1 and C_2 look random (not controlled)
  - Solution: hide in some ignored sections of the file (*e.g.* comment)
- Issue: collision is not meaningful
  - Solution: many file formats (*e.g.* PDF) allow conditional branches

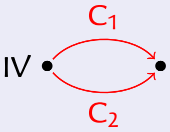
$M_1 = $ "`if` (C_1 == C_1) `{ good } else { evil }`"

$M_2 = $ "`if` (C_2 == C_1) `{ good } else { evil }`"

(prefix: `if`; suffix: `== C1) { good } else { evil }`)

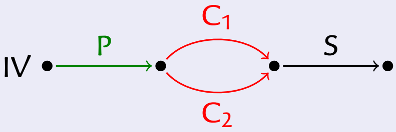
Exploiting collisions

Collision attack



- Start from IV
- C_1 and C_2 collide

Adding prefix and suffix



- Add identical prefix and suffix using iterative structure
- Usually same difficulty (just a different IV)

- Issue: C_1 and C_2 look random (not controlled)
  - Solution: hide in some ignored sections of the file (*e.g.* comment)
- Issue: collision is not meaningful
  - Solution: many file formats (*e.g.* PDF) allow conditional branches

$$M_1 = \text{“}\underbrace{\texttt{if (}}_{\text{prefix}}C_1\ \underbrace{\texttt{== } C_1\texttt{) \{ good \} else \{ evil \}}}_{\text{suffix}}\text{”}$$

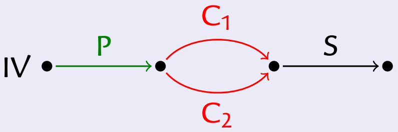
$$M_2 = \text{“}\underbrace{\texttt{if (}}_{\text{prefix}}C_2\ \underbrace{\texttt{== } C_1\texttt{) \{ good \} else \{ evil \}}}_{\text{suffix}}\text{”}$$

Chosen-Prefix Collisions [Stevens, Lenstra & de Weger, EC'07]

- Even with a prefix and prefix, many protocol seem unaffected by collision attacks

Identical-prefix collision

- Given IV, find $M_1 \neq M_2$ s. t. $H(M_1) = H(M_2)$



- Arbitrary common prefix/suffix, random collision blocks
- Breaks integrity verification
- Breaks signatures (in theory)

Chosen-prefix collision

- Given P_1, P_2, find $M_1 \neq M_2$ s. t. $H(P_1 \| M_1) = H(P_2 \| M_2)$

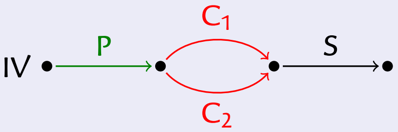
- Breaks certificates [Stevens & al, Crypto'09]
- Breaks TLS, IKE, SSH [Bhargavan & L, NDSS'16]

Chosen-Prefix Collisions [Stevens, Lenstra & de Weger, EC'07]

- Even with a prefix and prefix, many protocol seem unaffected by collision attacks

Identical-prefix collision

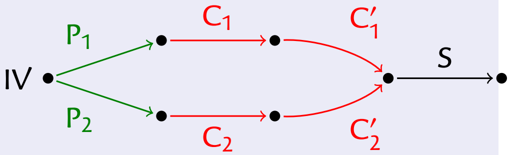
- Given IV, find $M_1 \neq M_2$ s. t. $H(M_1) = H(M_2)$



- Arbitrary common prefix/suffix, random collision blocks
- Breaks integrity verification
- Breaks signatures (in theory)

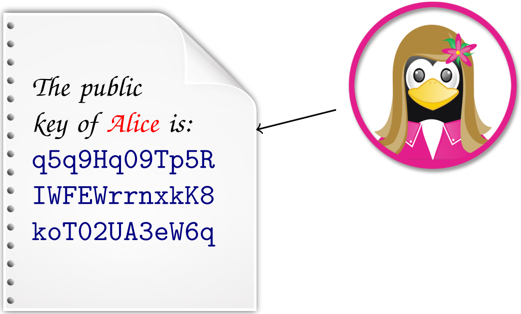
Chosen-prefix collision

- Given P_1, P_2, find $M_1 \neq M_2$ s. t. $H(P_1 \parallel M_1) = H(P_2 \parallel M_2)$



- Breaks certificates [Stevens & al, Crypto'09]
- Breaks TLS, IKE, SSH [Bhargavan & L, NDSS'16]

Attacking key certification [Stevens, Lenstra & de Weger, EC'07]



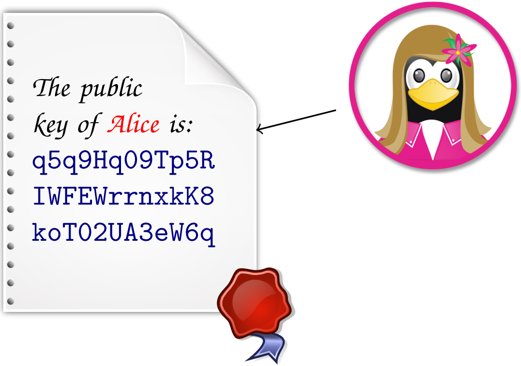
PKI Infrastructure

- Alice generates key
- Ask PKI to sign
- Certificate proves ID

Impersonation attack

1. Bob creates keys s.t. $H(\text{Alice}\|k_A) = H(\text{Bob}\|k_B)$
2. Bob asks CA to certify his key k_B
3. Bob copies the signature to k_A, impersonates Alice

Attacking key certification [Stevens, Lenstra & de Weger, EC'07]



PKI Infrastructure

- Alice generates key
- Ask PKI to sign
- Certificate proves ID

Impersonation attack

1. Bob creates keys s.t. $H(\text{Alice}\|k_A) = H(\text{Bob}\|k_B)$
2. Bob asks CA to certify his key k_B
3. Bob copies the signature to k_A, impersonates Alice

Attacking key certification [Stevens, Lenstra & de Weger, EC'07]

The public
key of Alice is:
Z0t226BvLI05
seJ+L6NRaT49
0E6p9TY2sW74

The public
key of Bob is:
7+zvZNcjdxXx
YRfYal4ZFmiY
E70hkirqNyfm

prefix

collision

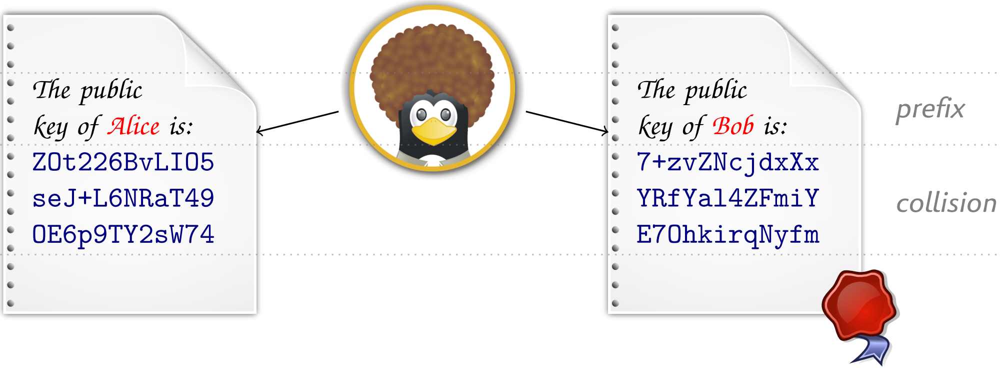
PKI Infrastructure

- Alice generates key
- Ask PKI to sign
- Certificate proves ID

Impersonation attack

1. Bob creates keys s.t. $H(\text{Alice}\|k_A) = H(\text{Bob}\|k_B)$
2. Bob asks CA to certify his key k_B
3. Bob copies the signature to k_A, impersonates Alice

Attacking key certification [Stevens, Lenstra & de Weger, EC'07]



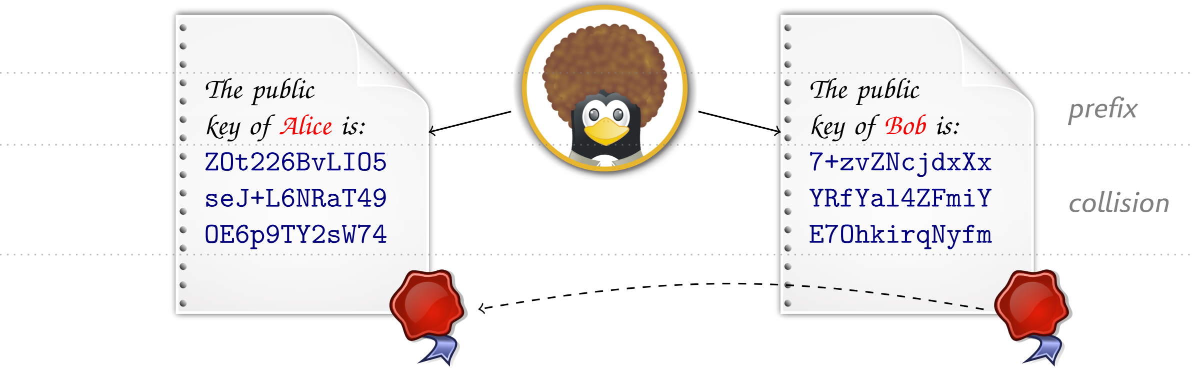
PKI Infrastructure

- Alice generates key
- Ask PKI to sign
- Certificate proves ID

Impersonation attack

1. Bob creates keys s.t. $H(\text{Alice}\|k_A) = H(\text{Bob}\|k_B)$
2. Bob asks CA to certify his key k_B
3. Bob copies the signature to k_A, impersonates Alice

Attacking key certification [Stevens, Lenstra & de Weger, EC'07]



PKI Infrastructure

- Alice generates key
- Ask PKI to sign
- Certificate proves ID

Impersonation attack

1. Bob creates keys s.t. $H(\text{Alice}||k_A) = H(\text{Bob}||k_B)$
2. Bob asks CA to certify his key k_B
3. Bob copies the signature to k_A, impersonates Alice

Outline

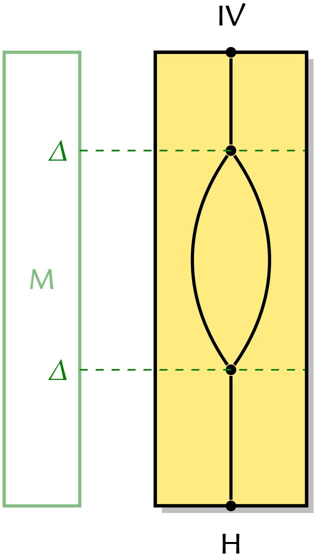
- Chosen-prefix collisions are more dangerous than identical-prefix collisions
  - Creation of a rogue CA with MD5 CPC [SSALMO, Crypto'09]
  - Abused in the wild: Flame malware (MD5 CPC)
- Generic attacks require $2^{n/2}$ operations in both cases
- Cryptanalytic attack harder for chosen-prefix collisions

	Identical-Prefix Collisions	Chosen-Prefix Collisions
MD5	2^{16} [SSALMO C'09]	$2^{39.1}$ [SSALMO C'09]
SHA-1	$2^{64.7}$ [Stevens EC'13, SBKAM C'17]	$2^{77.1}$ [Stevens EC'13]

Goal of this work

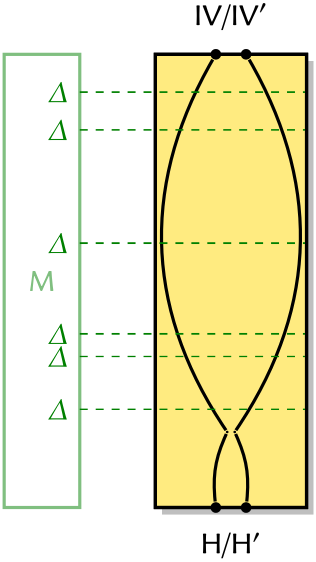
- Improve SHA-1 chosen-prefix collision attacks
- Reduce the gap between Identical-Prefix and Chosen-Prefix Collisions

Differential collision attacks



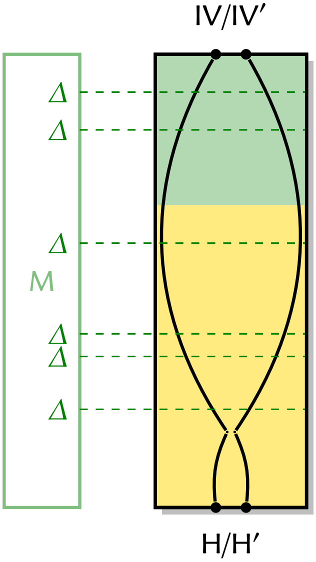
1. Differential cryptanalysis
   - Find a high probability trail $0 \to 0$
   - Find a conforming message
2. Linearized trails [Chabaud & Joux, C'98]
   - Linear combinations of local collisions
   - High probability, but non-zero input / output diff.
3. Message modification [BC04, WYY05]
   - Satisfy first rounds without paying probability
4. Non-linear trails [Wang & *al.*, C'05]
   - Modify trail in first rounds using non-linearity
   - Can start from arbitrary difference
     $\Rightarrow$ near-collision
5. Multi-block technique [CJ98, WYY05]
   - Two trails with same linear core: $0 \to \delta$ and $\delta \to \delta$
     $\Rightarrow$ collision

Differential collision attacks



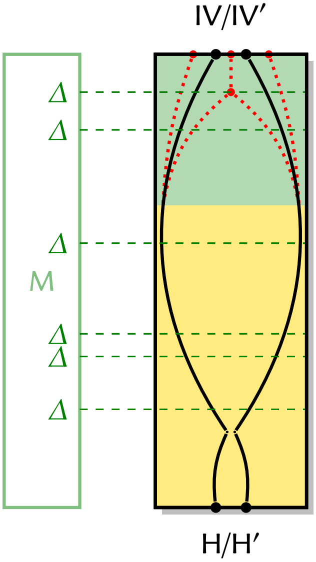
1. Differential cryptanalysis
   - Find a high probability trail $0 \to 0$
   - Find a conforming message
2. Linearized trails [Chabaud & Joux, C'98]
   - Linear combinations of local collisions
   - High probability, but non-zero input / output diff.
3. Message modification [BC04, WYY05]
   - Satisfy first rounds without paying probability
4. Non-linear trails [Wang & *al.*, C'05]
   - Modify trail in first rounds using non-linearity
   - Can start from arbitrary difference
     $\Rightarrow$ near-collision
5. Multi-block technique [CJ98, WYY05]
   - Two trails with same linear core: $0 \to \delta$ and $\delta \to \delta$
     $\Rightarrow$ collision

Differential collision attacks



1. Differential cryptanalysis
   - Find a high probability trail $0 \to 0$
   - Find a conforming message
2. Linearized trails [Chabaud & Joux, C'98]
   - Linear combinations of local collisions
   - High probability, but non-zero input / output diff.
3. Message modification [BC04, WYY05]
   - Satisfy first rounds without paying probability
4. Non-linear trails [Wang *& al.*, C'05]
   - Modify trail in first rounds using non-linearity
   - Can start from arbitrary difference
     $\Rightarrow$ near-collision
5. Multi-block technique [CJ98, WYY05]
   - Two trails with same linear core: $0 \to \delta$ and $\delta \to \delta$
     $\Rightarrow$ collision

Differential collision attacks

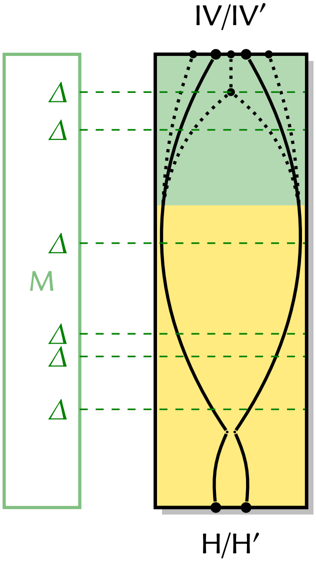


1. Differential cryptanalysis
   - Find a high probability trail $0 \to 0$
   - Find a conforming message
2. Linearized trails [Chabaud & Joux, C'98]
   - Linear combinations of local collisions
   - High probability, but non-zero input / output diff.
3. Message modification [BC04, WYY05]
   - Satisfy first rounds without paying probability
4. Non-linear trails [Wang & *al.*, C'05]
   - Modify trail in first rounds using non-linearity
   - Can start from arbitrary difference

     ⇒ near-collision
5. Multi-block technique [CJ98, WYY05]
   - Two trails with same linear core: $0 \to \delta$ and $\delta \to \delta$

     ⇒ collision

Differential collision attacks



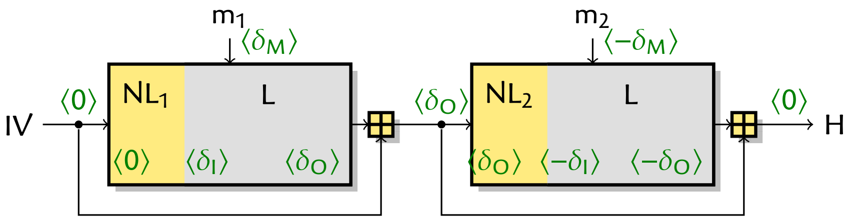
1. Differential cryptanalysis
   - Find a high probability trail $0 \to 0$
   - Find a conforming message
2. Linearized trails [Chabaud & Joux, C'98]
   - Linear combinations of local collisions
   - High probability, but non-zero input / output diff.
3. Message modification [BC04, WYY05]
   - Satisfy first rounds without paying probability
4. Non-linear trails [Wang & *al.*, C'05]
   - Modify trail in first rounds using non-linearity
   - Can start from arbitrary difference

     $\Rightarrow$ near-collision
5. Multi-block technique [CJ98, WYY05]
   - Two trails with same linear core: $0 \to \delta$ and $\delta \to \delta$

     $\Rightarrow$ collision

MD5/SHA-1 collision attack [Wang & al.]

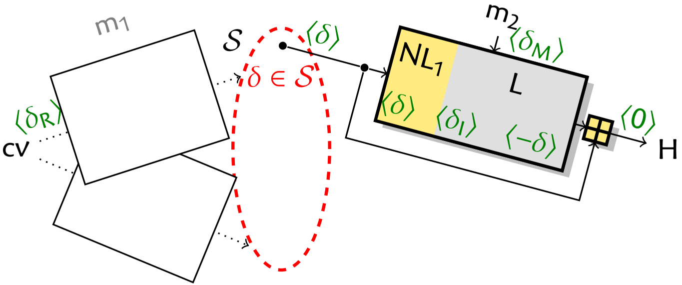
- Multi-block technique
  - Start from a good core linear trail $\delta_I \to \delta_O$
  - Build two non-linear trails $0 \to \delta_I$, $\delta_O \to -\delta_I$
  - Differences cancel due to feed-forward



Chosen-prefix collision attack [Stevens, Lenstra & de Weger, EC'07]

Main idea

Find a set of "nice" chaining value differences $\mathcal{S}$



1. Birthday phase
   - Find m_1, m'_1 such that $H(P_1 \| m_1) - H(P_2 \| m'_1) \in \mathcal{S}$
   - Complexity about $\sqrt{2^n/|\mathcal{S}|}$
2. Near-collision phase
   - Adjust non-linear trail
   - Erase the state difference, using near-collision blocks

How to build $\mathcal{S}$: previous works

MD5 [SLW07]

- Family of core trails, output on different bits
- Several near-collision blocks, erase differences bit by bit
- Very structured set $\mathcal{S}$

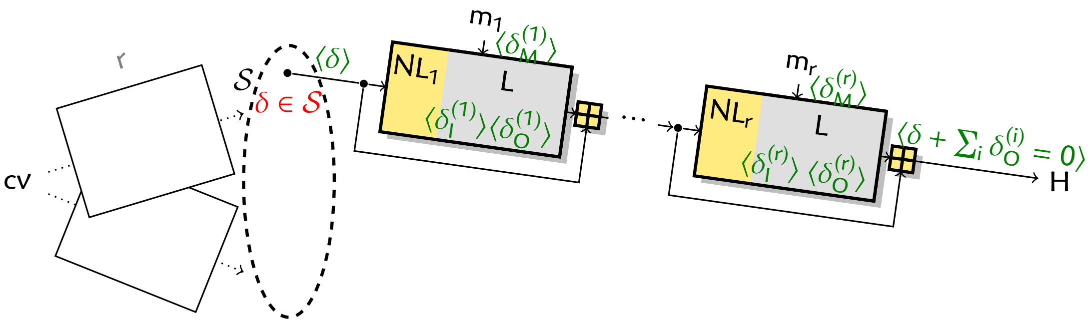
SHA-1 [S13]

- Single core trail, vary the last rounds
- Single near-collision block
- Small set $\mathcal{S}$, no structure

Our work

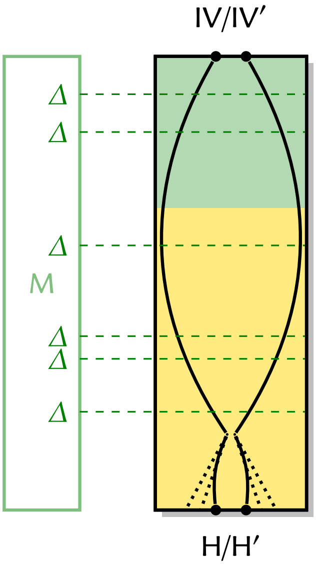
- The bottleneck of the SHA-1 attack is the birthday phase
  - Complexity around $\sqrt{2^n/|\mathcal{S}|}$
  - We need a larger set $\mathcal{S}$
- Can we combine those ideas and improve them?

New techniques



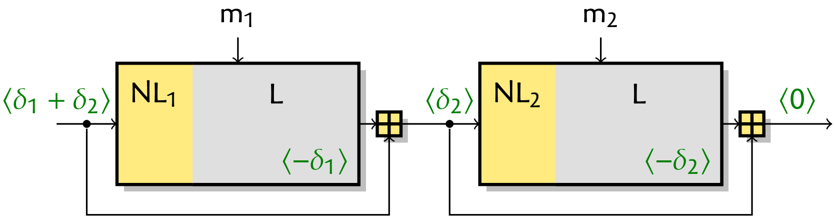
1. Larger set of output differences for the compression function (192 → 8768)
2. Multi-block technique using a single core trail $|\mathcal{S} \approx 2^{30}|$
3. Dynamic selection of near-collision targets (clustering)

Relaxing the final rounds



- Start from a core linear trail
- Modify last rounds to reach new difference
- Previous work: [Stevens, EC'13]
  192 differences with optimal probability
- Our work:
  8768 differences with non-optimal probability
- Reduce the complexity from $2^{77.1}$ to $2^{74.3}$

Multi-block technique with unstructured set



- Assume we reach a set of output differences $\mathcal{D}$ with one block
- With two blocks, we can reach a set of output differences:
  $\mathcal{S} := \{\delta_1 + \delta_2 \mid \delta_1, \delta_2 \in \mathcal{D}\}$
- With n blocks:
  $\mathcal{S} := \{\delta_1 + \delta_2 + \cdots \delta_n \mid \delta_1, \delta_2, \ldots \delta_n \in \mathcal{D}\}$
- Reduce the complexity from $2^{74.3}$ to $2^{68.6}$

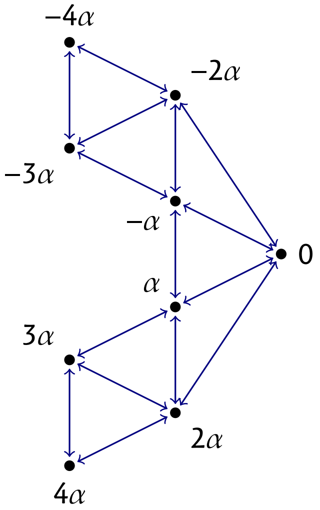
Clustering

Observation

A value in S can be reached in many different ways

$\delta_1 + \delta_2 + \delta_3 = \delta_1 + \delta_3 + \delta_2 = \delta_2 + \delta_1 + \delta_3 = \cdots$

- Near-collision block search:
  1. Choice of δ gives message conditions
  2. Search for message reaching δ
- Target δ values with same conditions simultaneously!
  - Eg. half work with two δ with similar cost
- With weights: $w_N = \min\left\{\left(1 + \sum(w_j/c_j^\beta)\right) / \sum(1/c_j^\beta)\right\}$
- Reduce the complexity from $2^{68.6}$ to $2^{66.9}$



Graph $\mathcal{G}$: transitions in $\mathcal{S}$

Ex: $\mathcal{D} := \{-2\alpha, -\alpha, \alpha, 2\alpha\}$

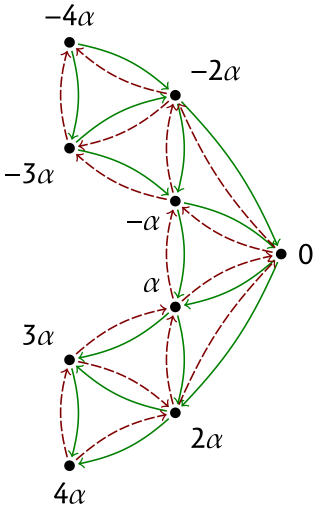
Clustering

Observation

A value in S can be reached in many different ways

$\delta_1 + \delta_2 + \delta_3 = \delta_1 + \delta_3 + \delta_2 = \delta_2 + \delta_1 + \delta_3 = \cdots$

- Near-collision block search:
  1. Choice of δ gives message conditions
  2. Search for message reaching δ
- Target δ values with same conditions simultaneously!
  - Eg. half work with two δ with similar cost
- With weights: $w_N = \min\left\{\left(1 + \sum(w_j/c_j^{\beta})\right) / \sum(1/c_j^{\beta})\right\}$
- Reduce the complexity from $2^{68.6}$ to $2^{66.9}$



Graph $\mathcal{G}$: transitions in $\mathcal{S}$

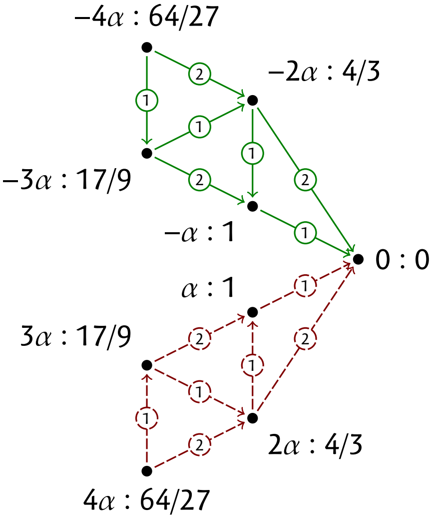
Ex: $\mathcal{D} := \{-2\alpha, -\alpha, \alpha, 2\alpha\}$

Clustering

Observation

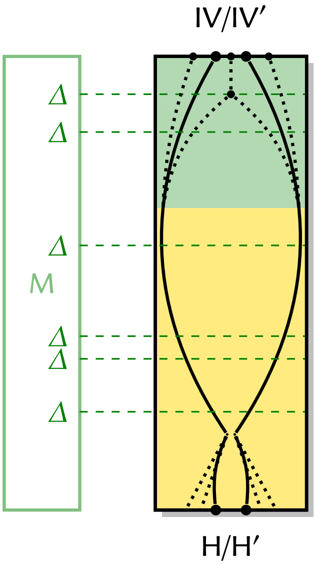
A value in S can be reached in many different ways
$\delta_1 + \delta_2 + \delta_3 = \delta_1 + \delta_3 + \delta_2 = \delta_2 + \delta_1 + \delta_3 = \cdots$

- Near-collision block search:
  1. Choice of δ gives message conditions
  2. Search for message reaching δ
- Target δ values with same conditions simultaneously!
  - Eg. half work with two δ with similar cost
- With weights: $w_N = \min\left\{\left(1 + \sum(w_j/c_j^\beta)\right) / \sum(1/c_j^\beta)\right\}$
- Reduce the complexity from $2^{68.6}$ to $2^{66.9}$



Graph $\mathcal{G}$: transitions in $\mathcal{S}$
Ex: $\mathcal{D} := \{-2\alpha, -\alpha, \alpha, 2\alpha\}$

Application to SHA-1: low-level details



- Start from the SHAttered collision attack
  - Proven to work
  - Complexity $2^{64.7}$ on GPU
- Relax the last rounds
  - 8768 possible output differences
- Assume that we can build trails in the first rounds
  - More constrained than IPC attack
  - C_{block} between $2^{64.7}$ (optimistic) and $2^{67.7}$ (conservative), depending on degrees of freedom
- Build set $\mathcal{S}$ and graph $\mathcal{G}$
  - Large computational effort
  - $|\mathcal{S}| = 2^{33.7}$, iterations for clustering

Attack parameters

Set $\mathcal{S}$		Birthday parameters					
Max cost	Size	Mask	Proba	# coll.	Ch. len.	# chain	Attack cost
$2.0 \cdot C_{block}$	$2^{24.66}$	106 bits	0.71	$2^{30.83}$	2^{34}	$2^{34.74}$	$2^{68.74} + 2^{65.83} + 2.0 \cdot C_{block}$
$2.5 \cdot C_{block}$	$2^{28.59}$	102 bits	0.65	$2^{31.03}$	2^{32}	$2^{34.84}$	$2^{66.84} + 2^{64.03} + 2.5 \cdot C_{block}$
$3.0 \cdot C_{block}$	$2^{30.95}$	98 bits	0.76	$2^{32.44}$	2^{31}	$2^{34.55}$	$2^{65.55} + 2^{64.44} + 3.0 \cdot C_{block}$
$3.5 \cdot C_{block}$	$2^{32.70}$	98 bits	0.76	$2^{30.70}$	2^{30}	$2^{34.68}$	$2^{64.68} + 2^{61.70} + 3.5 \cdot C_{block}$
$4.0 \cdot C_{block}$	$2^{33.48}$	98 bits	0.74	$2^{29.95}$	2^{30}	$2^{34.30}$	$2^{64.30} + 2^{60.95} + 4.0 \cdot C_{block}$
$4.5 \cdot C_{block}$	$2^{33.66}$	98 bits	0.74	$2^{29.77}$	2^{30}	$2^{34.21}$	$2^{64.21} + 2^{60.77} + 4.5 \cdot C_{block}$

Optimal parameters

- Optimistic estimate: $2^{66.9}$ ($C_{block} = 2^{64.7}$, max cost of $3.5 \cdot C_{block}$)
- Conservative estimate: $2^{69.4}$ ($C_{block} = 2^{67.7}$, max cost of $2.5 \cdot C_{block}$)

Results

- Generic framework to turn collision attacks into chosen-prefix collision attacks

Function	Collision type	Complexity (GPU)	Ref.
SHA-1	collision	2^{69}	[Wang & *al.*, C'05]
		$2^{64.7}$	[Stevens, EC'13], [Stevens & *al.*, C'17]*
	chosen-prefix collision	$2^{77.1}$	[Stevens, EC'13]
		$2^{66.9} - 2^{69.4}$	New
MD5	collision	2^{40}	[Wang & *al.*, EC'05]
		2^{16}	[Stevens & *al.*, C'09]
	chosen-prefix collision (9 blocks)	$2^{39.1}$	[Stevens & *al.*, C'09]
	(3 blocks)	2^{49}	[Stevens & *al.*, C'09]
	(1 block)	$2^{53.2}$	[Stevens & *al.*, C'09]
	(2 blocks)	$2^{46.3}$	New

- Small gap between SHA-1 Identical-Prefix and Chosen-Prefix collisions ($\times 4.6 - \times 26$)
- Improvement for MD5 CPC limited to two blocks

*The attack has a complexity of 2^{61} on CPU, and $2^{64.7}$ on GPU

Attack cost and future work

- We are now looking more closely at the low-level details
  - We believe we can keep two boomerangs
  - This gives $C_{block} = 2^{65.1}$, and the total cost is around $2^{67.2}$
- Cost estimation by renting GPUs:
  - About 2.6M$ on Amazon's AWS (using spot `p3.16xlarge` instances @7.5$/hr)
  - Around 540 000$ renting GPU (former mining farms?)
  - Affordable for state-level adversaries
- Security advice: retire SHA-1 NOW!

On-going work

- New ideas for small improvements of various parts of attacks
- Get the cost below 100 000$
- We hope to build a practical chosen-prefix collision in 2019...

Attack cost and future work

- We are now looking more closely at the low-level details
  - We believe we can keep two boomerangs
  - This gives $C_{block} = 2^{65.1}$, and the total cost is around $2^{67.2}$
- Cost estimation by renting GPUs:
  - About 2.6M$ on Amazon's AWS (using spot `p3.16xlarge` instances @7.5$/hr)
  - Around 540 000$ renting GPU (former mining farms?)
  - Affordable for state-level adversaries
- Security advice: retire SHA-1 NOW!

On-going work

- New ideas for small improvements of various parts of attacks
- Get the cost below 100 000$
- We hope to build a practical chosen-prefix collision in 2019...

Attack cost and future work

- We are now looking more closely at the low-level details
  - We believe we can keep two boomerangs
  - This gives $C_{block} = 2^{65.1}$, and the total cost is around $2^{67.2}$
- Cost estimation by renting GPUs:
  - About 2.6M$ on Amazon's AWS (using spot `p3.16xlarge` instances @7.5$/hr)
  - Around 540 000$ renting GPU (former mining farms?)
  - Affordable for state-level adversaries
- Security advice: retire SHA-1 NOW!

On-going work

- New ideas for small improvements of various parts of attacks
- Get the cost below 100 000$
- We hope to build a practical chosen-prefix collision in 2019...