CWI

New attacks on SHA-1: Optimal joint local collision analysis

Marc Stevens

marc.stevens@cwi.nl

Published at EUROCRYPT 2013

Part I – Introduction

Part II – Optimal joint local collision analysis
Part III – New attacks
To conclude …

Introduction

Digital signature schemes

- One of the pillars for P.K.I.s

- Used to ensure authenticity in/of
  - Browsers
  - Documents
  - Email
  - Software updates
  - Downloadable content
  - Currency transactions

- Hash-Then-Sign:
  {MD5,SHA-1,SHA-2}-{RSA,DSA}

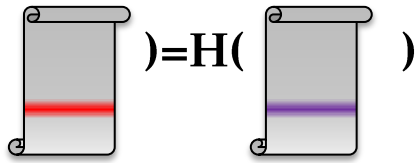
- Hash collision SHA-1(A)=SHA-1(B) ⇒ forgery

Introduction

Collision attacks on MD5 & SHA-1

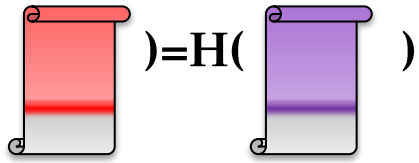
- Distinguish between 2 types
  - Identical prefix

$$H(P \mid C \mid S) = H(P \mid C' \mid S)$$



  - Chosen-prefix

$$H(P \mid C \mid S) = H(P' \mid C' \mid S)$$



  - P, P', S: Free to choose s/t $|P| = |P'|$
  - C, C': Generated based on P and P', $|C| = |C'| \in [64B, 1KB]$

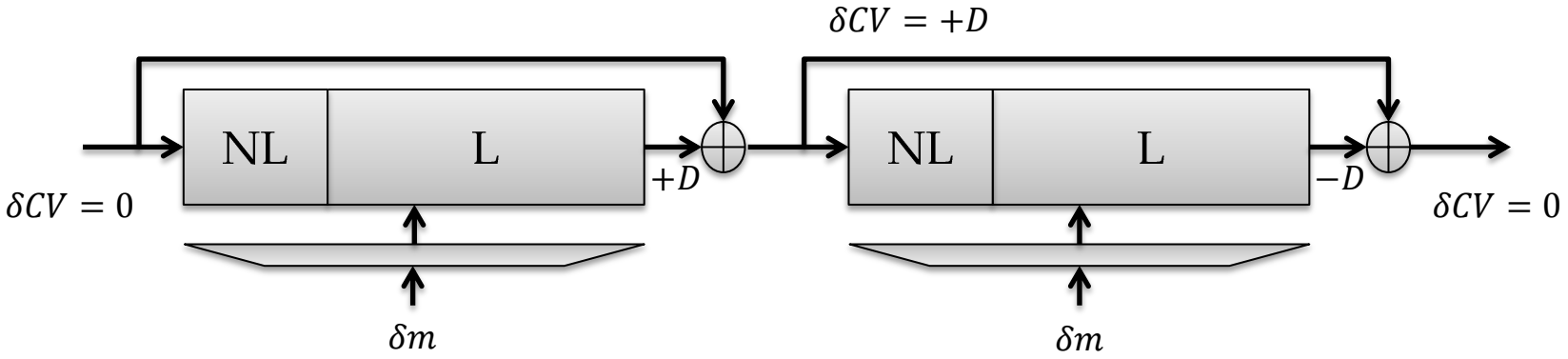
Introduction

	MD5		SHA-1		SHA-256	
	Id.Pr.	Ch.Pr.	Id.Pr.	Ch.Pr.	Id.Pr.	Ch.Pr.
Birthday	$2^{64.3}$	$2^{64.8}$	$2^{80.3}$	$2^{80.8}$	$2^{128.3}$	$2^{128.8}$
2004	2^{40}		2^{69}			
2005	2^{37}		(2^{63})			
2006	2^{32}	2^{49}				
2007	2^{25}	2^{42}	(2^{61})			
2008	2^{21}					
2009	2^{16}	2^{39}				
2010						
2011						
2012			$\underline{2^{61}}$	$\underline{2^{77}}$		
today	$\mathbf{2^{16}}$	$\mathbf{2^{39}}$	$\mathbf{2^{61}}$	$\mathbf{2^{77}}$	$\mathbf{2^{128.3}}$	$\mathbf{2^{128.8}}$

Published collision attacks on MD5 & SHA-1

Introduction

- First collision attack on full SHA-1 [WYY05]
- Identical-prefix collision attack
- 2 near-collision blocks



- Linear part: use linear combination of local collisions
- Non-linear part: transition from δCV to linear part

Introduction

- Improvements for full SHA-1
  - **[WYY05]** : $\mathbf{2^{69}}$
  - Wang, Yao, Yao 2005 : $\mathbf{2^{63}}$ (no publication, partially verified)
  - [SKI06] : **??** (**2^{52}** symbolic message modifications x2^{23}?)
  - Mendel et al. 2007 : $\mathbf{2^{60.x}}$ (no publication)
  - [MHP09] : $\mathbf{2^{52}}$ (withdrawn)
  - [Chen11] : $\mathbf{2^{58}}$ (too optimistic by factor $2^{3.5}$?)
- Cryptanalytic advancements not reflected in literature [PCTH11]
- No example collisions & no public implementations
- No chosen-prefix collision attacks

- Our goals:
  - Renew efforts on SHA-1: it is still widely used despite these attacks
  - Rigorously treat Linear part (combinations of local collisions)
  - Public open-source implementation

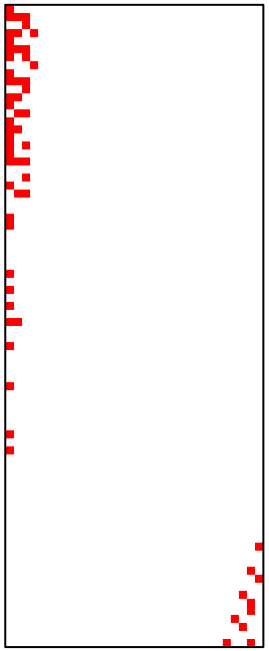
Local-collision analysis

- Local collision
  - 6-steps
  - First step: introduces disturbance
  - Next 5 steps: cancel disturbance

- Analysis of a single local collision
  - Success probability
  - Message bit relations
  - Exhaustively enumerate partial differential paths
  - Exact & optimal

Local-collision analysis

- Analysis of multiple local collisions (so far)
  - Disturbance vector marks local collisions
  - Use analysis of individual local collisions
  - Combine results
  - Make heuristic corrections
    - Prevent impossible situations
    - Local-collision compression $(2^{b+1} - 2^b \Rightarrow 2^b)$
    - Allow extra freedom at last 2 steps of SHA-1
  - Enables good collision attacks
  - What is the best possible?
    - Maximum joint success probability?
    - Maximum amount of freedom?

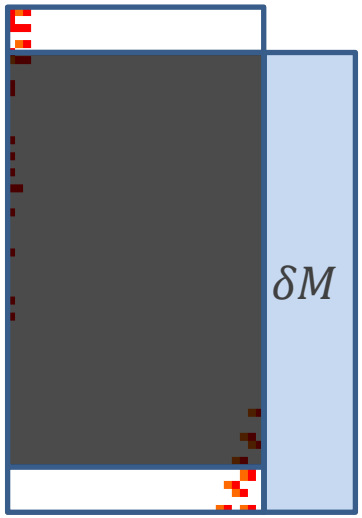


Local-collision analysis

- Analysis of multiple local collisions (so far)
  - Interactions of local collisions not studied thoroughly
  - Possible local-collision dependencies
    - Message differences:
      bit differences XOR'ed: may cancel
    - State differences:
      local collisions starting at same step interact due to carries
    - Boolean Function differences:
      'near' local collisions may cause more than 1 non-zero input difference
      or may interact through common zero-difference input bits
      Effect increased due to carries
  - Higher density L.C.s at start & end ⇒ more dependencies
  - So far only heuristic approximations of these effects on success probability
  - Q: How can we study all these effects in an exact manner?
  - A: Exhaustively enumerate differential paths

Truncated differential paths

- Differential path $\mathcal{P}$ for SHA-1
  - Based on local collisions
  - Exact description
  - Message, state & boolean-function differences
  - Probability can be computed exactly
  - Too restrictive for practical attack



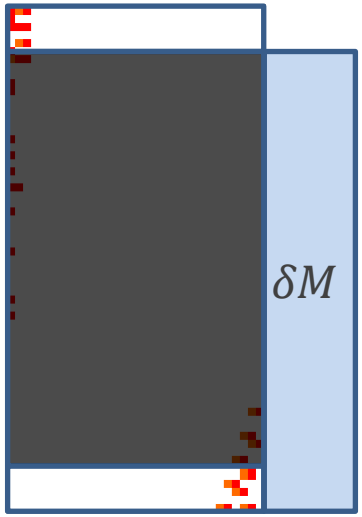
- Truncated differential path $\mathcal{T}$ for SHA-1
  - Exact message differences
  - Exact starting state differences
  - Exact ending state differences
  - Only D.V.-specified disturbances 'in between'
  - Identified with set of all possible differential paths $\mathcal{P} \in \mathcal{T}$ matching truncated differential path

Part II – Optimal joint local collision analysis

Part III – New attacks
To conclude …

Joint local-collision analysis

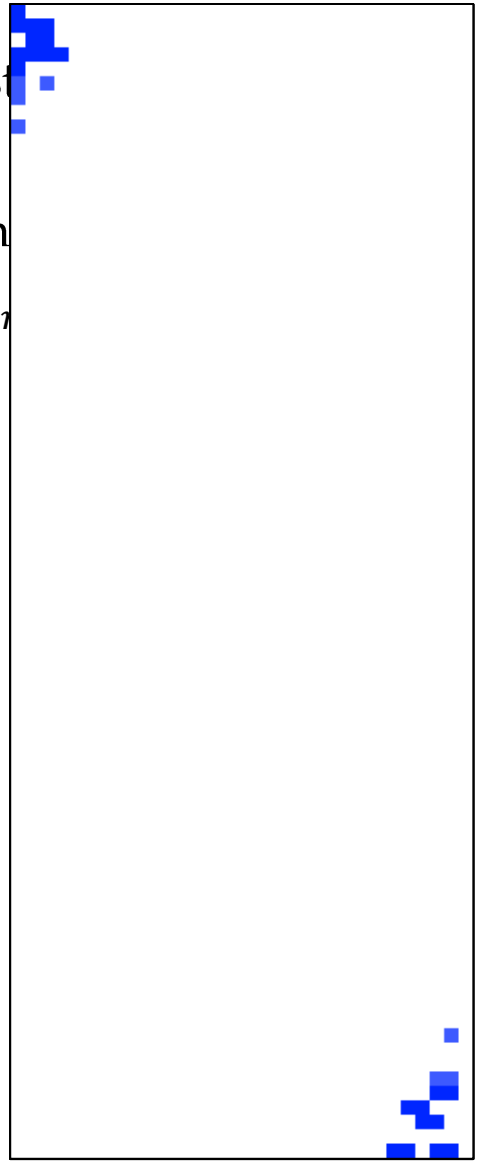
- Optimal joint local-collision analysis over last 60 steps (unpractical)
  - Exhaustively enumerate partial differential paths
  - Partition paths into truncated differential paths $\mathcal{T}$
    - Truncated path probability $p_{\mathcal{T}} = \sum_{\mathcal{P} \in \mathcal{T}} \Pr[\mathcal{P}]$
  - Optimal joint success probability $p_{max} := \max p_{\mathcal{T}}$



  - Problems:
    - # diff. paths grows exponentially in # local collisions
    - # δM grows exponentially in # local collisions
  - Solutions
    - Differential path reduction: exploit similarities
    - Message difference classes: combine δM with same probabilities

Joint local-collision analysis

- Optimal joint local-collision analysis over last 60 st
  - *Differential path reduction*
  - Exploit strong similarities between differential path
  - Split differential paths $\mathcal{P}$ into two parts: $\mathcal{P}_{red}$ & $\mathcal{P}_{elim}$
    - $\mathcal{P} = \mathcal{P}_{red} + \mathcal{P}_{elim}$
    - $\Phi(\mathcal{P}) = (\Delta WS_{start}, \Delta WS_{end}) = \Phi(\mathcal{P}_{red})$
    - $\Pr[\mathcal{P}] = \Pr[\mathcal{P}_{red}] \cdot \Pr[\mathcal{P}_{elim}]$
    - $\forall\ \mathcal{P}_{ext}$: $\Pr[\mathcal{P}_{ext}] = \Pr[\mathcal{P}_{red,ext}] \cdot \Pr[\mathcal{P}_{elim}]$
    - $Reduce(\mathcal{P}) \coloneqq \mathcal{P}_{red}$
  - Efficient algorithm
    1. Copy starting and ending differences to $\mathcal{P}_{red}$
    2. Copy all boolean function differences depending on state differences in $\mathcal{P}_{red}$
    3. Copy all state differences depending on boolean function differences in $\mathcal{P}_{red}$
    4. Repeat 2 & 3 until no new differences are added



Joint local-collision analysis

- Optimal joint local-collision analysis over last 60 steps (practical)
  - Definitions
    - $\mathcal{D}_{[b,e]}$:= set of all differential paths over steps $b, \ldots, e$ that follow D.V.
    - $\mathcal{R}_{[b,e]} := Reduce\left(\mathcal{D}_{[b,e]}\right) = \{Reduce(P) | P \in \mathcal{D}_{[b,e]}\}$
    - For $R \in \mathcal{R}_{[b,e]}$ and δM:
      $p_{(R,\delta M)}$:= ''cumulative probability of all P_{elim} that may complete R using δM''
      $$p_{(R,\delta M)} := \sum_{\substack{P \in \mathcal{D}_{[b,e]} | \delta M \\ R = Reduce(P)}} \frac{\Pr[P]}{\Pr[R]} = \sum_{P_{elim} + R \in \mathcal{D}_{[b,e]} | \delta M} \Pr[P_{elim}]$$
  - Rewrite $p_{\mathcal{T}}$ for truncated path $\mathcal{T}$ using δM:

$$p_{\mathcal{T}} = \sum_{P \in \mathcal{T}} \Pr[P] = \sum_{R \in Reduce(\mathcal{T})} \Pr[R] \cdot \left(\sum_{P \in \mathcal{T}, Reduce(P) = R} \frac{\Pr[P]}{\Pr[R]} \right) = \sum_{R \in Reduce(\mathcal{T})} \Pr[R] \cdot p_{(R,\delta M)}$$

  - $p_{\mathcal{T}}$ now only depends on reduced paths R and $p_{(R,\delta M)}$'s
  - Exhaustively enumerate *reduced* partial differential paths
    (practical: # *reduced* paths related to # local collisions at start & end)

Joint local-collision analysis

- Optimal joint local-collision analysis over last 60 steps (practical)
  - *Message difference classes*
  - Message difference class $\bar{M}$ over steps $[b, e]$

$$\forall m_1, m_2 \in \bar{M}\colon\ \forall R \in \mathcal{R}_{[b,e]}\colon\ p_{(R,m_1)} = p_{(R,m_2)}$$

  - Message difference classes are monotone
    - Let $m_1, m_2 \in \bar{M}$ over steps $[b, e]$
    - Backward extensions $a|m_1, a|m_2$ are in the same class over steps $[b-1, e]$
    - Forward extensions $m_1|a, m_2|a$ are in the same class over steps $[b, e+1]$
    - Only need to process 1 representative of each class

Joint local-collision analysis

- Optimal joint local-collision analysis over last 60 steps (practical)
  - Efficiently compute $\mathcal{R}_{[20,79]}$, classes $\bar{M}$ & probabilities $p_{R,\bar{M}}$'s iteratively

    1. Start with trivial $\mathcal{R}_{[t,t]}$
    2. Compute probabilities over steps $[t, t]$
    3. Determine message classes over steps $[t, t]$ & keep 1 representative each
    4. Extend $\mathcal{R}_{[b,e]}$with one step (forward/backward)
       1) Exhaustively extend all paths in $\mathcal{R}_{[b,e]}$ and reduce them
       2) Determine $\mathcal{R}_{[b,e+1]}/\mathcal{R}_{[b-1,e]}$ and cumulative probabilities
       3) Determine message classes & keep 1 representative each
    5. Repeat 4. until $[b, e] = [20,79]$

  - Determine truncated path probabilities $p_{\mathcal{T}}$
  - Determine $p_{max} := \max_{\mathcal{T}} p_{\mathcal{T}}$

Joint local-collision analysis

- Optimal joint local-collision analysis over last 60 steps (practical)
  - Implemented, C++ code will be released as open-source
  - Optional bound u on maximum number of carries
    - Reduces memory and runtime complexity
    - Determines lower-bound for p_{max}
    - Rapidly converging for increasing # carries
  - Analyzed many disturbance vectors
    - Classification by Manuel
    - Only two interesting classes I(K,b) & II(K,b)
    - K – shift (K=40,…,60)
    - b – rotation (b=0,2)

Joint local-collision analysis

- Optimal joint local-collision analysis over last 60 steps (practical)

DV	u				
	0	1	2	3	7
I(45, 0)	81.00	76.91	76.66	76.54	76.45
I(46, 0)	79.00	75.02	74.92	74.84	74.83
I(47, 0)	79.00	75.15	74.83	74.71	74.61
I(**48**, **0**)	75.00	71.84	71.61	71.51	71.42
I(**49**, **0**)	76.00	72.59	72.34	72.24	72.15
I(**50**, **0**)	75.00	72.02	71.95	71.93	71.92
I(51, 0)	77.00	73.76	73.53	73.43	73.34
I(52, 0)	79.00	76.26	76.24	76.24	76.24
I(47, 2)	79.68	77.01	76.68	76.56	76.47
I(48, 2)	76.68	74.27	73.99	73.88	73.79
I(49, 2)	77.00	74.30	74.02	73.92	73.83
I(50, 2)	77.00	74.74	74.63	74.61	74.60

DV	u				
	0	1	2	3	7
II(45, 0)	83.00	75.45	74.82		
II(**46**, **0**)	76.00	71.85	71.83		
II(47, 0)	81.42	76.23	75.87		
II(48, 0)	80.00	76.11	75.89	75.79	
II(49, 0)	80.00	75.04	74.72	74.60	74.51
II(**50**, **0**)	78.00	73.52	73.23	73.12	73.02
II(**51**, **0**)	77.00	72.55	72.18	72.02	71.88
II(**52**, **0**)	75.00	71.88	71.87	71.76	71.75
II(53, 0)	76.96	73.65	73.34	73.23	73.14
II(54, 0)	77.96	73.97	73.74	73.64	73.55
II(55, 0)	77.96	75.22	74.99	74.89	74.80
II(56, 0)	76.96	74.48	74.18	74.07	73.97
II(46, 2)	82.00	77.51			
II(50, 2)	81.00	76.51	76.16	76.03	

Part III – New attacks

To conclude …

Optimal choices

- Chosen disturbance vector: II(52,0)
- To construct attack we need for the last 60 steps:
  - List of target δCVs (only 1 for the last near-collision attack)
  - ΔWS: state differences after 1st round (to construct 1st round diff. path)
  - Message bit relations
- Determine $p_{\mathcal{T}}$ for all truncated diff. paths $\mathcal{T} \equiv (\Delta WS, \delta CV, \delta M)$
- Select only δCV (with max. prob.)
  - Optimizes success probability of last 60 steps of last near-collision attack
- Fix ΔWS (with max. prob.) and construct 1st round differential path
- Select δM with 6 (max.) truncated diff. paths (given $\Delta WS,\ \delta CVs$)
  - Optimizes success probability of last 60 steps of first near-collision attack
- Set of $\delta M \Rightarrow$ message bit relations
  - Optimizes amount of freedom under above choices
  - Use freedom for message modification techniques

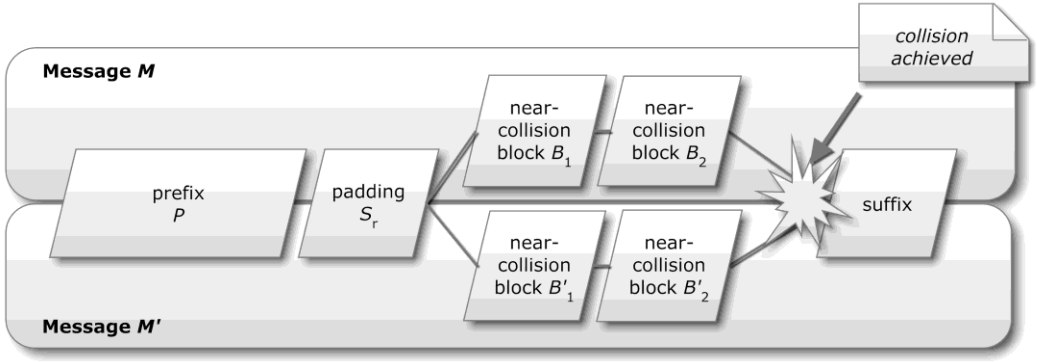
Near-collision attack

- Implementation of first near-collision attack
  - Optimized for last 60 steps
  - 50+ bits of freedom in message space: room for improvement in 1st round
  - Open-source: `http://code.google.com/p/hashclash` 

  - Publicly verifiable: complexity & correctness
  - Full disclosure: resource for researchers
- Complexity estimate: $2^{57.5}$ SHA-1 compression equiv.
  - Steps [0,32]: $C = 2^{11.97}$ (measured performance, not purely theoretical cost model)
  - Steps [33,52]: $p = 2^{-20.91}$ (exact theoretical probability, implementation verified)
  - Steps [53,60]: $p = 2^{-8}$ "
  - Steps [61,79]: $p = 2^{-16.65}$ "
  - Runtime estimate: 2000 core-years
  - Amazon EC2 cost estimate: $2,000,000

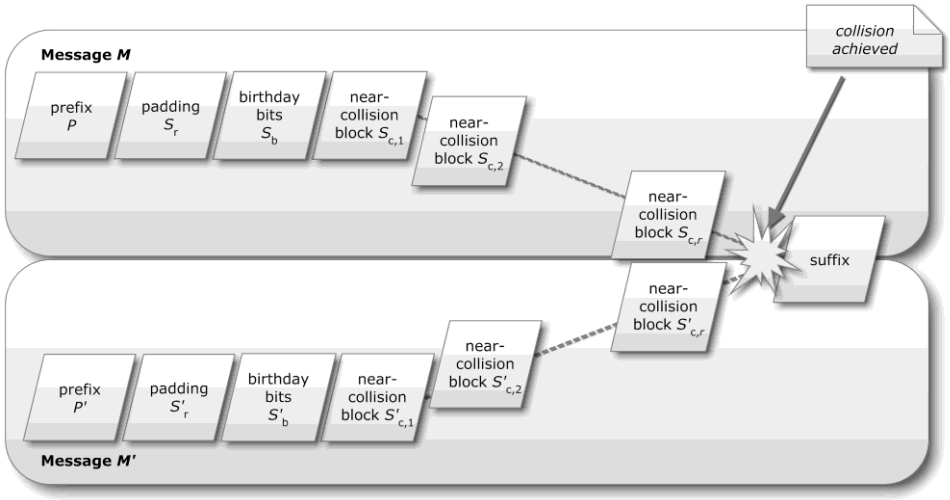
Identical-prefix collision attack

- Two near-collision block attack
- First near-collision attack
  - Introduces $\delta CV \neq 0$
  - Speedup factor 6 due to freedom in δCV
  - Open-source implementation
  - Complexity $2^{57.5}$ SHA-1 compression equiv.
- Second near-collision attack
  - Cancels δCV (depends on first N.C. block)
  - No freedom in δCV
  - Complexity
    optimistic estimate : $2^{60.1} = 2^{57.5} \cdot 6$
    realistic estimate : $\approx 2^{61}$
- Total complexity: $\approx 2^{61}$



Chosen-prefix collision attack

- Birthday search + single near-collision block attack
- Birthday search
  - Targets same set of δCV as 1st N.C. from identical-prefix collision attack
  - 192 possible δCV
  - Tailored birthday search
  - Complexity $2^{77.06} \approx \sqrt{\pi/192} \cdot 2^{160/2}$ (cost of generating pseudo-random walks)
- Near-collision attack
  - Same as 2nd N.C. from identical-prefix collision attack
  - Cancels found δCV
  - Complexity realistic estimate : $\approx 2^{61}$
- Total complexity: $\approx 2^{77.1}$



To conclude …

To conclude…

Academic efforts

- Renewed efforts at constructing better collision attacks on SHA-1
- Goal was to find optimal solution for the last 60 steps in a rigorous manner
- First open-source implementation
  - Publicly verifiable: correctness & complexity
  - Resource for researchers
- So far 1st round part 'first generation' implementation

- Work in progress
  - W.I.P.: extend rigorous methods into 1st round
    (Maximize space for message modification techniques)
  - Exploit more message modification techniques
  - Implementation on graphic cards (more cost effective)
  - Efforts at finding first near-collision block using current implementation

To conclude…

What about possible non-academic efforts?

- Very little academic research on SHA-1 collisions last few years
- Actual SHA-1 collision search has very high cost for an academic effort
- Nation-States have the knowledge, resources & motivation to attack hash functions in order to forge signatures (see the super malware Flame)
- ‘Malicious’ SHA-1 collisions in the ‘wild’ closer than thought?

Thank you for your attention

Questions?