Counter-cryptanalysis:
fire retardant for the next
Flame-like attack on MD5 & SHA-1

Marc Stevens
marc.stevens @ cwi.nl

CWI Amsterdam
Part I – Weak signature schemes

Part II – Counter-cryptanalysis
Part III – Flame
To Conclude…
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 $\text{MD5}(\text{A})=\text{MD5}(\text{B}) \Rightarrow$ forgery
Collision attacks on MD5 & SHA-1

- Distinguish between 2 types
  - Identical prefix
    \[ H(P \ | \ C \ | \ S) = H(P \ | \ C' \ | \ S) \]
    \[ H(\ ) = H(\ ) \]

  - Chosen-prefix
    \[ H(P \ | \ C \ | \ S) = H(P' \ | \ C' \ | \ S) \]
    \[ H(\ ) = H(\ ) \]

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

<table>
<thead>
<tr>
<th>Year</th>
<th>MD5</th>
<th>SHA-1</th>
<th>SHA-256</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Id.Pr.</td>
<td>Ch.Pr.</td>
<td>Id.Pr.</td>
</tr>
<tr>
<td>Birthday</td>
<td>$2^{64.3}$</td>
<td>$2^{64.8}$</td>
<td>$2^{80.3}$</td>
</tr>
<tr>
<td>2004</td>
<td>$2^{40}$</td>
<td></td>
<td>$2^{69}$</td>
</tr>
<tr>
<td>2005</td>
<td>$2^{37}$</td>
<td></td>
<td>(2^{63})</td>
</tr>
<tr>
<td>2006</td>
<td>$2^{32}$</td>
<td>$2^{49}$</td>
<td></td>
</tr>
<tr>
<td>2007</td>
<td>$2^{25}$</td>
<td>$2^{42}$</td>
<td>(2^{61})</td>
</tr>
<tr>
<td>2008</td>
<td>$2^{21}$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2009</td>
<td>$2^{16}$</td>
<td>$2^{39}$</td>
<td></td>
</tr>
<tr>
<td>2010</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2011</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2012</td>
<td></td>
<td></td>
<td>$2^{61}$</td>
</tr>
<tr>
<td>today</td>
<td>$2^{16}$</td>
<td>$2^{39}$</td>
<td>$2^{61}$</td>
</tr>
</tbody>
</table>

Published collision attacks on MD5 & SHA-1
Notes

- Generate your own MD5 chosen-prefix collision attack in a day using Project HashClash: https://code.google.com/p/hashclash/

- No publicly known collision for SHA-1 has been found yet

- First SHA-1 collision more likely to be constructed by nation-states than academia due to required resources, see: http://www.schneier.com/blog/archives/2012/10/when_will_we_se.html
Strategies for meaningful colliding files

- Using identical-prefix collisions
  - Meaningful C and C’
    
    \[
    C = \text{"… of money is $10,000.00…"} \\
    C' = \text{"… of money is $20,000.00 …"} \\
    C = \text{"… OFFSET=X …"} \\
    C' = \text{"… OFFSET=Y …"} \\
    \]
    
    Hard
  - IF-THEN-ELSE construct
    
    \[
    \text{IF ( C ==C ) THEN ... ELSE ...} \\
    \text{IF ( C’==C ) THEN ... ELSE ...} \\
    \]
    
    Easy, but requires IF-THEN-ELSE

- Using chosen-prefix collisions
  - Meaningful different P, P’ & hide C and C’ in message
    
    \[
    P = \text{"I owe you $20"} \\
    P' = \text{"You will inherit all my possessions"} \\
    C = \text{<hidden image>} \\
    C' = \text{<hidden image>} \\
    \]
    
    Easy
Introduction
Abuse examples

- **Identical-prefix**
  - Colliding Software [Kam04,Mik04]
  - Colliding PostScript documents [DL05,GIS05]
  - Colliding X.509 certificates (same ID, diff. RSA moduli) [LdW05]

- **Chosen-prefix**
  - Colliding PDF documents [SLdW07]
  - Colliding Software [SLdW07]
  - Colliding X.509 certificates (diff. IDs) [SLdW07]
  - Rogue Certification Authority [SSALMOdW09]
  - Rogue Windows Update signing certificate [Flame12]
Introduction

Solutions for weak signatures schemes

What to do when a signature scheme is broken?

The easy answer: “migrate to a more secure scheme”
i.e., move from MD5-RSA to SHA-2-RSA

Who should migrate?

Signers: definitively!: generate SHA-2-RSA signatures
Problems: compatibility/deployment issues, risk-cost trade-off, human,…
Result: forgeries can still be constructed till the last signer migrates

Verifiers: definitively!: don’t accept MD5-RSA signatures
Problem: too many old signatures in use to just invalidate them all at once
Result: old and new forgeries can abused against nearly everyone

The easy answer is not a practical solution for the near future
Our answer: detect forged signatures

Verifiers: don’t accept forged MD5-RSA signatures
Results:
- Old legitimate signatures are still valid
- Verifier protected against forgeries
- Independent of migration by signers

How to detect forged signatures?: counter-cryptanalysis!
Part II – Counter-cryptanalysis

Part III – Flame
To Conclude…
New paradigm: counter-cryptanalysis
- Use cryptanalytic techniques of attacks against them
- Detect and/or reconstruct cryptanalytic attacks at cryptographic level

First practical example [Ste12]:
- Detection whether message is constructed with cryptanalytic attack
  - Computational cost: MD5: $\times 224$
    SHA-1: $\times 15$
    (much less using early abort: W.I.P.)
- Works for single given message of collision pair, other sibling not necessary
Counter-cryptanalysis
Cryptanalysis of MD5

- Differential cryptanalysis between 2 messages
  - MD5 performs 64 steps for each 512-bit block
  - Use differential path
    - System of differential equations over these 64 steps
    - Exact description how differences should propagate
  - Solve differential path to find collision
Counter-cryptanalysis is based on that:

- for a given message the entire hash computation is known
- feasible collision attacks ‘require’ trivial differential steps
  - MD5: known attacks use at least 16 trivial differential steps
    either zero difference (0,0,0,0) or MSB difference (2^{31}, 2^{31}, 2^{31}, 2^{31})
  - SHA-1: known approaches use at least 6 trivial differential steps
    zero difference (0,0,0,0,0)
- only very few message block differences may lead to feasible attacks
  - Very limited set of suitable message block differences
  - MD5: 200+ (very few actually used in implemented attacks)
  - SHA-1: 15+ (no actual collisions known yet)
Counter-cryptanalysis
Recovering near-collision blocks

- Guess message block difference & working state difference at step $i$
- Reconstruct computation
- Check whether collision in chaining value is obtained
- For each guess cost is equivalent to compression function call
- Work backwards to recover more near-collision blocks
Counter-cryptanalysis

Reference implementation

Reference implementation to detect collision attacks
- Available at http://marc-stevens.nl/research (at the bottom)

- Library interface to replace existing MD5/SHA-1 implementation
  - MD5Init/MD5Init_unsafe, MD5Update, MD5Final
  - SHA1Init/SHA1Init_unsafe, SHA1Update, SHA1Final
  - \{MD5,SHA1\}Final returns non-zero value if an attack is detected
  - \{MD5,SHA1\}Init_unsafe always results in correct (and possibly unsafe) hash
  - \{MD5,SHA1\}Init results in correct hash if no attack has been detected, otherwise a safe hash is returned

- Command line program
  - detectcollv <files>
Counter-cryptanalysis
Applications

Anomaly detection for digital signatures

° Online: active protection
  – Signer: protection against malicious signature requests
  – Verifier: protection against forged signatures
  – E.g., for TLS/SSL, OSs (drivers, executables, updates), etc.

° Offline: forensic analysis
  – Main example: spyware Flame
  – Confirm use of chosen-prefix collision attack
  – Reconstruction of cryptanalytic attack details
  – Interesting discovery:
    • Variant of our chosen-prefix collision attack [SSA⁺09]
    • Used algorithms and approaches not in scientific literature
Part III – Flame

To Conclude…
Cf. [Kas12,Sot12]
- Highly advanced malware
- Targeting the Middle-East
- Discovered in May 2012
- Active since 2007 or earlier
- Uncharacteristic features for malware
  - Up to 20 modules: each carefully selected prior to infection
  - Almost 20MB: includes Lua VM & libraries for compression, database, …
  - Did not spread wildly & evaded discovery for ~5 years
  - Surgical-precision attacks: each target carefully selected
  - Spread itself illegitimately using the Windows Update platform
  - First cryptanalytic attack on hash function found in the ‘wild’
  - Developed new variant cryptanalytic attack to do so…

Source: Kaspersky Lab
Flame Propagation

- Man-in-the-middle attack on Windows Update
- Local network attack
  - Registers itself as proxy server for **update.microsoft.com** using WPAD (Web Proxy Auto-Discovery)
  - Windows Update falls back to insecure HTTP
    - depends on digital signatures for security
    - no need to subvert TLS/SSL connection
- Propagation
  - Flame serves fake ‘security update’ using Windows Update platform
  - Requires properly-signed ‘security update’
  - **Uses illegitimate sub-CA** valid since Feb 2010
    ⇒ sub-CA invalid before that time
    ⇒ this attack was almost certainly done around Feb 2010 or later
Flame Certificate hierarchy

- Microsoft Root Certificate Authority
  - Microsoft Windows Verification PCA
    - Microsoft Windows
      - Patch_KBxxx.exe
  - Microsoft Enforced Licensing Intermediate PCA
  - Microsoft Enforced Licensing Registration Authority CA
    - Microsoft LSRA PA

MD5 collision attack to forge signature

- MS
- Terminal Services LS
- WuSetupV.exe
Uses chosen-prefix collision attack [SLdW07]:

<table>
<thead>
<tr>
<th>Flame’s certificate</th>
<th>Standard TLS certificate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Serial number, validity</td>
<td>Serial number, validity</td>
</tr>
<tr>
<td>CN=MS</td>
<td>CN=Terminal Services LS</td>
</tr>
<tr>
<td>2048-bit RSA key (271 bytes)</td>
<td>birthday bits</td>
</tr>
<tr>
<td></td>
<td>4 near collisions blocks (computed)</td>
</tr>
<tr>
<td>issuerUniqueId data</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Identical bytes (copied from signed cert)</td>
</tr>
<tr>
<td></td>
<td>X509 extensions</td>
</tr>
<tr>
<td></td>
<td>MD5 signature</td>
</tr>
</tbody>
</table>

Source: Alex Sotirov
Flame
Analyzing Flame’s attack

- Only Flame’s “MS” sub-CA certificate public
- The colliding “TSLS” certificate is not public (lost?)

- First example for counter-cryptanalysis
  - Assumed chosen-prefix collision attack
  - Only 1 of the 2 colliding certificates available to us
  - Ran proof-of-concept implementation (from 2008)
    - chosen-prefix collision detected
    - 4 near-collision blocks recovered
    - all differential paths reconstructed
    - <0.03 seconds
Yet unknown chosen-prefix collision attack

- Other differential path family
  - Same message differences for all 4 near-collision attacks (up to sign)
    \[ \delta m_4 = 2^{31}, \quad \delta m_{11} = \pm 2^{15}, \quad \delta m_{14} = 2^{31} \]
  - No systematic elimination using \( \delta m_{11} = \pm 2^b \) as in [SSA+09]

- Yet unknown differential path construction algorithm
  - Differences in all bit positions of Q6 in all 4 near-collision attacks
  - Not a characteristic of known construction algorithms
  - Initial tests show this approach to be significantly slower than our approach

- Yet unknown birthday search
  - Birthday search is preprocessing phase to find differences that can be cancelled by differential path family
  - Less flexible differential path family \( \Rightarrow \) Higher complexity birthday search
  - Approx. \( 2^{49} \) MD5 compression function calls
  - To compare: our attack in total has average complexity of \( 2^{44.6} \) MD5 calls

For a more extensive analysis, see [http://eprint.iacr.org/2013/358](http://eprint.iacr.org/2013/358)
To conclude…
Conclusions

- Migrate away from MD5 and SHA-1 based signature schemes

- The easy answer of "migrate to more secure signature schemes" is not a practical solution for the near future

- Instead allow old signatures, but protect verifiers against forgeries

- Real-time signature forgery detection possible
  - works for collision attacks on MD5 & SHA-1 [Ste12]
  - recovers full differential paths

- Reference implementation available
  - Feedback requested!
  - Let me know where it is used!
Conclusions

- Flame uses chosen-prefix collision attack ‘in the wild’
  - But an entirely new variant!
  - Different differential path family than [SSA⁺09]
  - Yet unknown birthday search
  - Yet unknown block-wise elimination procedure
  - Yet unknown differential path construction algorithm
  - New attack has higher complexity than [SSA⁺09]
Open questions

- Who made Flame?
  - Evidence points to world-class cryptanalysts, not just hackers
  - Adds to predominant speculation of nation-state behind Flame

- Why develop a new variant attack before Feb 2010?
  - But our attack implementation is public since June 2009 (see [Ste12])
  - Requires large effort, done in parallel
  - Nevertheless: exposes their cryptanalytic knowledge

- Will the first successful SHA-1 attack be due to scientific efforts or not?
  - Recent years have shown almost no public efforts on SHA-1.
  - Flame’s attack on MD5 was developed independently and in parallel to public scientific efforts.
  - Perhaps attacks on SHA-1 are as well…
  - Nation-states have more computing resources than academics…
Thank you for your attention

Questions?
References


[Kam04] MD5 considered to be harmful someday, Dan Kaminsky, 2004.


[LdW05] On the possibility of constructing meaningful hash collisions for public keys, A.K. Lenstra, B. de Weger


[SLdW07] Chosen-prefix collisions and colliding X.509 certificates for different identities, M. Stevens, A.K. Lenstra, B. de Weger,

[Sot12] Analyzing the MD5 collision in Flame, Alex Sotirov, SummerCon conference, New York, June 2012.


[Ste12] Attacks on Hash Functions and Applications, Marc Stevens, PhD thesis, Leiden University. (See also the open-source project at: http://code.google.com/p/hashclash/ )

[WY04] How to break MD5 and other hash functions, X. Wang, H. Yu,