Counter-cryptanalysis
analyzing Flame's new collision attack

Marc Stevens
marc.stevens@cwi.nl

CWI Amsterdam
Overview

- Introduction
  - What if?
  - New paradigm: counter-cryptanalysis
  - Applications
- Counter-cryptanalysis
- Flame: background
- Flame: counter-cryptanalysis
- (Preliminary) conclusions
What if a cryptographic primitive gets broken?

Symmetric encryption schemes
- Only two parties using a communications channel
- Together: negotiate more secure encryption schemes
- New communication secure
- Old communications: no sense in re-encrypting older communications
- Impossible to detect (passive) cryptanalytic attacks at cryptographic level

Digital signature schemes
- Up to three parties: requester, signer, verifier
- Signer can switch to more secure scheme
- New signatures secure
- Old signatures: signer may not be able to replace them
- Verifier vulnerable while older weak scheme is still supported
- Is it possible to detect dubious signatures at cryptographic level?
  Answer: YES
Example: MD5-based signatures
- MD5 practically broken since 2004 [WY05]
- Shown to be a realistic threat to TLS/SSL in 2008 [SSA⁺09]
- MD5-based signatures prohibited to public CA’s from 31 Dec 2010
- Yet ‘older’ MD5-based signatures still supported everywhere
- Abuse still possible for applications not using public CA’s
Introduction

New paradigm: counter-cryptanalysis

- 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: \( \times 15 \)
  - Works for single given message of collision pair, other sibling not necessary

![Diagram](image)
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]
    - But: used algorithms and approaches not in scientific literature
Overview

- Introduction
- Counter-cryptanalysis
  - Cryptanalysis of MD5
  - Cryptanalytic tell-tales
  - Recovering near-collision blocks
- Flame: background
- Flame: counter-cryptanalysis
- (Preliminary) conclusions
Collision attacks on MD5 & SHA-1 based on near-collision attacks

Near-collision attack:
- Given input chaining value pair
- Computes message block pair
- Such that output chaining value pair has ‘desired’ differences
- Based on differential path
Differential path

- Precise description of how differences propagate through compression function
- Last 40 steps determine most of attack’s complexity

⇒ trivial differential steps \textit{required} for feasible attacks

⇒ very limited set of suitable message differences (MD5: 200+)
Counter-cryptanalysis

Based on:

- that for given message the entire hash computation is known
- that cryptanalytic collision attacks are highly specialized
  - Very limited set of suitable message differences
  - MD5: 200+ (very few actually used in implemented attacks)
  - SHA-1: 15+ (no actual collisions known yet)
- that cryptanalytic collision attacks ‘require’ trivial differential steps
  - MD5: known attacks use at least 16 trivial differential steps
    either zero difference state (0,0,0,0) or MSB difference state \( (2^{31}, 2^{31}, 2^{31}, 2^{31}) \)
  - SHA-1: known approaches use at least 6 trivial differential steps
    zero difference state (0,0,0,0,0)
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
Overview

- Introduction
- Counter-cryptanalysis
- Flame: background
  - Overview of Flame
  - Propagation
  - Certificate hierarchy
  - ‘Free’ signatures
  - Chosen-prefix collision attack
  - Predicting variable certificate fields
- Flame: counter-cryptanalysis
- (Preliminary) conclusions
Flame: background
Overview of Flame

- Malware
- Discovered in May 2012
- Active since 2010 or earlier
- Complex
  - Almost 20MB
  - Up to 20 modular components
  - Advanced info-stealing
  - Advanced propagation
- Surgical-precision attacks
- Initial response: “Flame is lame”
  most aspects have been seen before
- Until discovery in June 2012:
  Flame is the first to use a cryptanalytic attack in the wild…

[Kas12,Sot12]
Flame: background
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’ through Windows Update
  - Requires properly-signed ‘security update’
  - **Uses illegitimate sub-CA** (valid since Feb 2010)
Flame: background Certificate hierarchy
Flame: background 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

- MS
- Terminal Services LS

MD5 collision attack to forge signature
License management system for Terminal Services clients

- Based on X.509 certificate hierarchy under “Microsoft Root CA”
- “Microsoft LSRA PA” provides signed certificates to servers
- Fully automated process
- Fixed fields:
  - CommonName: “Terminal Services LS”
  - X.509 extensions
- Fields chosen by signing server
  - validity period
  - serial number
  - sufficiently predictable
- Free to choose fields
  - public key
  - used to hide collision bits
**Flame: background**

**Chosen-prefix collision attack**

Uses chosen-prefix collision attack [SLdW07]:

<table>
<thead>
<tr>
<th>Flame’s certificate</th>
<th>Standard TLS certificate</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Serial number, validity</strong></td>
<td><strong>Serial number, validity</strong></td>
</tr>
<tr>
<td>+229</td>
<td><strong>CN=Terminal Services LS</strong></td>
</tr>
<tr>
<td><strong>CN=MS</strong></td>
<td>+259</td>
</tr>
<tr>
<td>2048-bit RSA key (271 bytes)</td>
<td><strong>birthday bits</strong></td>
</tr>
<tr>
<td>+500</td>
<td>+504</td>
</tr>
<tr>
<td>+504</td>
<td>+512</td>
</tr>
<tr>
<td>+512</td>
<td><strong>4 near collisions blocks (computed)</strong></td>
</tr>
<tr>
<td><strong>issuerUniqueID data</strong></td>
<td><strong>RSA key (509 bytes?)</strong></td>
</tr>
<tr>
<td>+768</td>
<td>+768</td>
</tr>
<tr>
<td></td>
<td>+786</td>
</tr>
<tr>
<td></td>
<td><strong>Identical bytes (copied from signed cert)</strong></td>
</tr>
<tr>
<td></td>
<td>+1392</td>
</tr>
<tr>
<td></td>
<td><strong>X509 extensions</strong></td>
</tr>
<tr>
<td></td>
<td><strong>MD5 signature</strong></td>
</tr>
<tr>
<td>+1392</td>
<td></td>
</tr>
</tbody>
</table>

| publicly available                                      | lost!?                                |

Source: AlexSotirov
Predicting variable certificate fields [Sot12]

- Validity period
  - Fully automated ⇒ accurately predictable
  - 1 second window
  - ‘Easy’: previously done by hand [SSA+09]

- Serial number
  - Based on sequential number and current time (ms since boot)
  - Certificate request acts as Query & Increment operation
  - 1 millisecond window
  - Significantly harder ⇒ probably needs many attempts
Overview

- Introduction
- Counter-cryptanalysis
- Flame: background
- Flame: counter-cryptanalysis
  - Analyzing Flame’s attack
  - Recovered differential paths
  - First observations
- (Preliminary) conclusions
° Only Flame’s “MS” sub-CA certificate public
° The colliding “TSLS” certificate is not public (lost?)
° Use of chosen-prefix collision is exposed through identical suffix
  – TLS X.509 extensions visible in “MS” IssuerUniqueID-field

° 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 2009)
    • chosen-prefix collision detected
    • 4 near-collision blocks recovered
    • all differential paths reconstructed
    • <0.03 seconds
  – Big surprise looking at differential paths
 Flame: counter-cryptanalysis
Recovered differential paths

dm4=[!31!] dm11=[!15!]
dm14=[!31!]
Q64: 
Q34: 
Q25
Q15: 
Q13: 
Q12: 
Q11: 
Q9: 
Q7: 
Q5: 
Q4: 
Q3: 
Q2: 
Q1:  
Q0: 

Q15: 
Q14: 
Q13: 
Q12: 
Q11: 
Q10: 
Q9: 
Q8: 
Q7: 
Q6: 
Q5: 
Q4: 
Q3: 
Q2: 
Q1:  
Q0: 

Q64: 
Q34: 
Q25
Q15: 
Q14: 
Q13: 
Q12: 
Q11: 
Q10: 
Q9: 
Q8: 
Q7: 
Q6: 
Q5: 
Q4: 
Q3: 
Q2: 
Q1:  
Q0: 

dm4=[!31!] dm11=[!15!]
dm14=[!31!]
Q64: 
Q34: 
Q25
Q15: 
Q13: 
Q12: 
Q11: 
Q9: 
Q7: 
Q5: 
Q4: 
Q3: 
Q2: 
Q1:  
Q0: 

differential path
near-collision block 1

Q25-32: 
Q33: 
Q34: 
Q35-59: 
Q60:
Q61:
Q62:
Q63:
Q64:

Q25-32: 
Q33: 
Q34: 
Q35-58: 
Q60:
Q61:
Q62:
Q63:
Q64:

differential path
near-collision block 2
Differential path family

- Based on differential path of first MD5 collision attack [WY05]
  - Same message differences for all 4 near-collision attacks
    \[ \delta m_4 = 2^{31}, \quad \delta m_{11} = \pm 2^{15}, \quad \delta m_{14} = 2^{31} \]
  - Variations: uses other differences in last 4 steps
    \[
    \begin{align*}
    &\text{WY ( [31], [31,25], [31,25], [31,25] )} \\
    &1. \quad ( [31], [31,25], [31,25,-14,-12,9], [31,25,-18,-15,-12,9,1] ) \\
    &2. \quad ( [31,5], [31,-25,-9,5], [31,26,24,20,-9,5], [-26,24,21,-14,-9,5,0] ) \\
    &3. \quad ( [31], [31,25,9], [31,25,24,-14,9], [30,26,-24,20,-17,15,9,-3] ) \\
    &4. \quad ( [31], [31,-25,-9], [31,-25,14,-9], [-25,14,-9,-5,3,0] ) \\
    \end{align*}
    \]
  - No apparent systematic approach: difficult analysis

- Different from known chosen-prefix collision attack [SSA+09]
  - Family of message differences: \( \delta m_{11} = 2^b \)
  - Allows systematic elimination of differences: \( ([], [b + 10], [b + 10], [b + 10] + r) \)
  - Uses birthday search that results in \((0, x, x, y)\) for arbitrary \(x\) and \(y\)
Differential path construction

- Uses differences in *all* bit positions of Q6 in *all* 4 near-collision attacks:
  - 1st Q6: |+---------+---+--------+---|  
  - 2nd Q6: |+--------+-----+--------+---|  
  - 3rd Q6: |+--------+-----+--------+---|  
  - 4th Q6: |+--------+-----+--------+---|  

- Not a characteristic of known construction algorithms
  - Known construction algorithms aim at a low number of bit conditions  
  - Differences in all bit positions for any Qi may occur, but unlikely  

- Seems ‘by design’
- Does not aid in collision search $\Rightarrow$ must aid in path construction
Overview

- Introduction
- Counter-cryptanalysis
- Flame: background
- Flame: counter-cryptanalysis
- (Preliminary) conclusions
(Preliminary) conclusions

- Real-time signature forgery detection possible
  - works for collision attacks on MD5 & SHA-1 [Ste12]
  - recovers full differential paths
  - protects verifiers against malicious signatures
    when still supporting MD5 & SHA-1 based signatures

- 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
Open questions

- Who did it?
  - Evidence points to world-class cryptanalysts, not hackers
  - Adds to speculation of nation-state behind Flame

- Why develop a new variant attack?
  - Requires large effort
  - But our attack implementation is public since 2009 (see [Ste12])
  - Also preliminary results indicates new attack not faster than [SSA+09]
  - (but may be more cost effective on massively parallel architectures)
  - Nevertheless: exposes their cryptanalytic knowledge

- How does it compare to our attack [SSA+09]
Thank you for your attention

Questions?
References


[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/)