Need Security Against Digital Signature Forgeries

Marc Stevens
Cryptology Group
CWI Amsterdam

joint work with Max Fillinger (CWI) & Dan Shumow (Microsoft Research)
Talk overview

• First part:
  • Supermalware Flame used digital signature forgery
  • Reconstruction of cryptanalytic forgery attack
  • New insights into cryptanalytic resources of secret agencies

• Second part:
  • How can we trust old digital signatures?
  • Counter-cryptanalysis: forgery detection
  • New improved forgery detection
  • New improved release forgery detection library
Snowden files

- Lot of news about activities of Security Agencies
  
  “Collect it all, know it all, exploit it all”

- Cryptography is circumvented, not broken
  - Get the plaintext at server or client end
  - Subpoena the keys (Lavabit, CryptoSeal VPN, SSL, ...)
  - Use kleptography to backdoor key generation:
    DUAL ECC random number generator
  - Weaken crypto standards and implementations
Snowden files

- Cryptography itself seems to work, hard to break:
  - End-to-end
  - RSA, Diffie-Hellman, ECDH and AES

- Little news related to actual cryptanalysis:
  no known ‘head on‘ attacks to break crypto primitives

- Nevertheless some insight into capabilities
  due to exposed cryptanalytic work on MD5
  in the supermalware Flame discovered in 2012
Known MD5 collision attacks

- Cryptographic hash function: $MD5: M \rightarrow \{0,1\}^{128}$
- Digital signatures: hash-then-sign paradigm
  hash collision $\Rightarrow$ digital signature collision/forgery
- Merkle-Damgard construction:
  chaining value updated iteratively using compression function

- Breakthrough collision attacks by [Wang et al. 2004]

- Limited form of collisions, no direct impact on cybersecurity
  $\Rightarrow$ little to no response in industry to migrate from MD5
Known MD5 collision attacks

- [2007 & 2009 Stevens et al.]
Theoretical: more versatile collision attacks

• Practical: realistic abuse scenario with high impact on cybersecurity
Known MD5 collision attacks

- [2009 Stevens et al.] Rogue Certification Authority

- MD5-based signatures not allowed for public CA’s since end 2010
- MD5-based signatures still in use for legacy platforms
- MD5-based signatures still ubiquitously supported
Known MD5 collision attacks

- Attacks on MD5 (& SHA-1) based on near-collision attacks
Known MD5 collision attacks

- Attacks on MD5 (& SHA-1) based on near-collision attacks
- Near-collision attack on compression function:
  - Given input chaining value pair
  - Compute message block pair
  - To achieve ‘desired’ difference between output chaining values

Based on a differential path
U.S., Israel developed Flame computer virus to slow Iranian nuclear efforts, officials say

By Ellen Nakashima, Greg Miller

The United States and Israel developed a computer virus known as Flame that could penetrate and sabotage computer networks, according to Western officials.

The massive piece of malware, nicknamed Flame that can steal data from networked computer systems, is one of the most sophisticated cyberwarfare campaigns ever, the officials said.

Flame malware wielded rare "collision" crypto attack against Microsoft

Such real-world exploits are almost unheard of, underscoring the significance of the attack, which was first reported by the cybersecurity company Kaspersky Labs.

Microsoft Security Research and Defense Blog

Flame malware collision attack explained

Since our last MSRC blog post, we've received questions on the nature of the cryptographic attack we saw in the complex, targeted malware known as Flame. This blog summarizes what our research revealed and why we made the decision to release Security Advisory 2718704 on Sunday night PDT. In short, by default the attacker's certificate would not work on Windows Vista or more recent versions of Windows. They had to perform a collision attack to forge a certificate that would be valid for code signing on Windows Vista or more recent versions of Windows. On systems that pre-date Windows Vista, an attack is possible without an MD5 hash collision. This certificate and all certificates from the involved certificate authorities were invalidated in Security Advisory 2718704. We continue to encourage all customers who are not installing updates automatically to do so immediately.
Supermalware Flame

- US/Israel espionage on Middle-East
- Discovered in May 2012

Highly advanced malware
- Surgical-precision attacks: each target carefully selected
- 20MB in up to 20 modules: each carefully selected prior to infection
- Spread itself illegitimately through Windows Update protocol
Supermalware Flame

• June 3: MS: Windows Update digital signature forgery!

• June 6: MS: Used MD5 chosen-prefix collision attack ?!

• June 7: Stevens: counter-cryptanalysis:

  Recovered cryptanalytic details,
  exposed new variant MD5 CPC attack!

• June 9: Sotirov: millisecond window for successful forgery

  ⇒ 10 to 100 forgery attempts
  ⇒ only a few days per attempt
Supermalware Flame

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

WuSetupV.exe
Supermalware Flame

Flame’s certificate

- Serial number, validity
  - CN=MS
- 2048-bit RSA key (271 bytes)
- issuerUniqueID data
- MD5 signature

Publicly known

Standard TLS certificate

- Serial number, validity
  - CN=Terminal Services LS
- Chosen prefix (difference)
- birthday bits
- 4 near collisions blocks (computed)
- Identical bytes (copied from signed cert)
- X509 extensions
- MD5 signature

Source: Alex Sotirov
Flame’s unknown collision attack

- CRYPTO 2013: Published initial attack analysis
  - Chosen-prefix collision attack

- Uses other ‘differential path’ family

- Unknown differential path construction algorithm
  (observed artifacts not present for known algorithms)

- Unknown birthday search

- Weak lower-bound cost: $2^{44.3}$
  (compared to avg cost of $2^{44.55}$ for known attack with similar params)
Flame’s unknown collision attack

• Upcoming paper jointly with Max Fillinger (CWI):
  • Reverse engineered attack
  • Reconstructed differential path family and likely parameters
  • Determined matching birthday search

• Complexity analysis for various parameter choices

• More precise lower-bound: $2^{46.6}$

• Best-fit parameters: cost $2^{49.3}$
  (compared to avg cost of $2^{44.55}$
  for known attack with similar params)
Insights

Comparison:

- Novel approach to ‘count down’ to zero difference

- Overall cost:
  Expected cost $2^{49.3}$ (ca. 40,000 CPUcore hours)
  worse than [SSA+09] $2^{44.55}$ (ca. 1500 CPUcore hours)
  For 3-day attempts requires equiv. to 560 CPUcores

- More suited for special hardware: GPUs etc
  For 3-day attempts requires about 8 high-end GPUs

- Differential path construction:
  Open-source project HashClash finds
  significantly sparser paths in only 15 seconds

- Speed-up techniques (advanced message modification):
  Not maximized
  Indicates lack of use of ‘rotation conditions’
In conclusion:

• No indication of superior techniques

• Various parts sub-optimal
  • sub-optimal parts should have little effect on total cost
  • as-long-as-it-works approach?
Legacy digital signatures

- Well known that MD5 is broken since 2004, 2007, 2009, ...
- Many legacy MD5-based signatures
- MD5-based signatures trusted almost ubiquitously still today

- Flame’s attack likely to be launched in certificate validity period of Feb 2010 and Feb 2012
  - Forged Certificate not usable before or after
- Proves it is hard to migrate away from MD5
- MD5’s successor SHA-1 also broken

- How can we trust legacy MD5 & SHA-1 based signatures today?
Counter-cryptanalysis

- Counter-cryptanalysis [Stevens 2013]
  - Detect cryptanalytic attacks at the cryptographic level
  - Exploits unavoidable anomalies caused by active attacks
  - Covers entire classes of attacks with identical anomalies

- Collision detection
  - Application to MD5 & SHA-1
  - Single message of collision pair sufficient

- Digital signature forgery detection
  - Apply collision detection
  - Signature is marked as invalid when a collision is detected
  - Invalidates both the innocuous-looking and the malicious message
  - Current release used by e.g.:
    Microsoft (IE SmartScreen), FOX-IT, CAcert, ...
Collision detection

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+)
  (SHA-1: 15+)
Collision detection

- Basic algorithm: detect last near-collision block
  - Guess message block difference & difference at trivial step $i$
  - Determine $B_k'$ from $B_k$ and $WS_i'$ from $WS_i$
  - Reconstruct computation
  - Check whether collision in chaining value is obtained
Collision detection

Practical guarantees

1. False positives occur with negligible probability
   Conjectured $\approx 2^{-128}$ (MD5) / $\approx 2^{-160}$ (SHA-1)

2. No false negatives
   - Assuming list of message block differences is ‘sufficiently complete’
   - Control of achieved security level $\leq N$ by selection of
     $\delta M$ / D.V. based on estimated lower bound attack cost
   - With current knowledge up to approx. 50-bit security for MD5
     and 70-bit security for SHA-1.
Collision detection

- Currently collision detection has high cost: Every $\delta B / D.V.$ costs an additional hash operation.

- SHA-1 is weak: $\geq 15 \ D.V.\ ^\wedge \ s$

- MD5 is very weak: $\geq 224 \ \delta M\ ^\wedge \ s$
Improved collision detection

• Currently each \((\delta B, \delta WS)\)-guess costs 1 full compression

• Speed up collision detection using unavoidable bitconditions: Bit conditions necessary for all possible feasible attacks for a given \(\delta B/D.V:\)

• Verify unavoidable bit conditions quickly and do full work only with low probability

• Does not introduce possible false negatives

• MD5: difficult to find: requires case by case study

• SHA-1: easy to find using powerful tool (joint local collision analysis [Stevens 2013b])
SHA-1 Unavoidable bitconditions

SHA-1: finding unavoidable conditions per D.V.:
- Analyze critical range of steps ([35,65] out of [0,79])
- Enumerate all possible differential paths over that range
- Determine linear span covering $\delta B$ from all possible paths (thus having non-zero probability)
- $\delta B$ outside span implies zero probability
- Linear span $\Rightarrow$ linear equations $= \text{unavoidable bitconditions}$
SHA-1 Unavoidable bitconditions

SHA-1: finding unavoidable conditions

- Per D.V.: 7 to 15 unavoidable bitconditions
  32 D.V.’s totalling 373 UBCs that are overlapping!

- Greedy selection:
  1. Start with spans of equations $U_i$ for each $DV_i$
  2. Let $V_i$ be an empty span for each $DV_i$
  3. Determine set of equations that are elements of the most # sets $U_i \setminus V_i$
  4. Select an equation that is the simplest: lowest weight, small gaps
  5. Add that equation to the span basis of the respective $V_i$’s
  6. Repeat until $U_i = V_i$ for all $DV_i$

- Reduction to 156 unique UBCs, each related to 1 to 7 DVs
  (all of the form: $M_i[a] \oplus M_j[b] = 0/1$)
SHA-1 Unavoidable bitconditions

- Various implementations verifying 156 UBC:
  - Straightforward per D.V.: 2.09 SHA-1 computations
  - Constant-time: 1.33 SHA-1 computations
  - Fastest: 0.82 SHA-1 computation

- UBCs reduce cost of 32 DVs:
  from 32 SHA-1 computations
  to 0.049 SHA-1 computations on average

- Total on average cost:
  \[1 + 0.8 + 0.049 = 1.87\] SHA-1 computations
Improved collision detection library

• New release collision detection library
  
  Check out: https://marc-stevens.nl

• Uses unavoidable bitconditions for SHA-1
• Tests twice as many DVs & 9 times faster than previous version
• Speed is 1.87 times SHA-1

• Includes a special *reduced-round* SHA-1 collision detection for reduced-round SHA-1 example collisions

• Upcoming paper jointly with Dan Shumow (Microsoft Research)
Thank you!