Certificate Reputation: Cryptographic Analysis of Public Keys in Use

Dan Shumow
Microsoft Research
Disclaimer: The views and opinions expressed in this talk are those of the authors and are not necessarily those of Microsoft Corporation.
Joint Work With...

• Microsoft Research (MSR)
  • Kevin Kane
  • Tolga Acar
  • Brian LaMacchia

• Microsoft Trustworthy Computing (TwC)
  • Kelvin Yiu
  • Ben Nick

• Windows Security and Identity (SID)
  • Phil Hallin
  • Anoosh Saboori
Certificate Reputation (CertRep)

• In Windows 8 / Internet Explorer 11, for users who have opted in, Microsoft has begun collecting certificates used by TLS servers as well as certificates used to sign applications (verified by AppVerifier.)
• This effort is motivated by FLAME and other high profile subversions of the Global CA ecosystems.
• Client side components recently released (October/November 2013.)
• Analysis of gathered data is just beginning / in progress.
• Long term / high level goal is to monitor the health of the CA ecosystem.
CertRep Goals

• Detect fraudulent X.509 certificates containing MS domain names issued by public CAs.
• Identify public CAs that do not comply with trusted root program technical requirements.
• Detect widespread man-in-the-middle attacks against popular sites using fraudulent TLS server certificates (attacks that affect at least thousands of IE11 + users.)
• Detect cryptographic attacks such as hash collisions or repeated prime factors in RSA moduli.
• Collect data to inform cryptographic policy decisions.
CertRep Overview

• CAPI2 (Windows Certificate API) modified to record public certificates (Windows)
• These certificates are reported back to Microsoft via SmartScreen. (Trustworthy Computing)
• CertRep data stored through Microsoft internal data storage and computation service COSMOS. (Trustworthy Computing)
• Aggregated certificates are analyzed in an attempt to detect bad cryptography or potentially harmful certificate policy violations (Microsoft Research)
SmartScreen

• Certificate collection is accomplished through SmartScreen
• SmartScreen is an Internet Explorer/Windows feature to protect users from phishing and malware.

• SmartScreen History
  • Internet Explorer 8 (2009): First release, checks for malicious URLs (phishing attacks.)
  • Internet Explorer 9 (2011): Added support for checking downloaded executables.
  • Windows 8.1 (2013): Added support for collecting certificates of all Authenticode signed executables.
  • Internet Explorer 11 (2013): Added support for collecting TLS Certificates of the domain of visited URLs.
Opting Into SmartScreen

Windows 8+

Windows Phone
CertRep Telemetry Data

• SmartScreen samples clients (doesn’t take every cert observed.)
• We are seeing 100k-150k certs coming in per day.
• Coverage currently has holes (but is advancing every day.)
• Potential benefit over surveys of public certificates: Visibility into enterprise that do not have their CAs visible to the public internet (but do opt into SmartScreen.)
COSMOS

• CertRep Data is aggregated and accessed through COSMOS
• COSMOS is Microsoft’s internal storage and computation service for our online services.
• Used by Microsoft online services, including Bing.
• High Level Overview:
  • Petabyte storage
  • Uses Dryad programming model: MapReduce + DAGs.
  • Programmed in SCOPE Language: SQL-ish language, optimized and compiled to Dryad.
  • Virtual Clusters: Logical Unit of Execution.

See: “SCOPE: Easy and Efficient Parallel Processing of Massive Data Sets”
Cryptanalysis of Certificates

• Hash Collision Detection [Stevens’ “Counter Cryptanalysis”]
• Batch Factoring.

• Analysis of Certificate Fields:
  • Cryptographic:
    • Algorithms used
    • Bit Length / Key strength.
  • Noncryptographic fields:
    • Serial Numbers
    • Subject Name
    • Extensions
Hash Collision Detection

• Motivation: Detect and thwart a future FLAME-like attack.
• Hash Collisions against MD5 are feasible.
• SHA-1 Hash Collisions have not been demonstrated yet, but are expected to be forthcoming.
• Microsoft is pushing to move off of SHA-1, however SHA-1 Hashes will be valid on certificates and in use for some time.
Hash Collision Overview

The Merkle-Damgård Construction

• Recall that both the MD5 and SHA-1 hash functions are instances of the generic Merkle-Damgård (MD) construction.
• This is an iterated application of nonlinear round functions on input blocks.
Hash Collision Overview

Message-block Differences

\[ m_1 \rightarrow f_1 \rightarrow y_1 \rightarrow m_i \rightarrow f_i \rightarrow y_i \rightarrow m_i + \Delta_i \rightarrow y_i + \partial_i \rightarrow m_i \rightarrow f_i \rightarrow y_i \rightarrow m_i \rightarrow f_i \rightarrow y_i \rightarrow m_i \rightarrow f_i \rightarrow y_i \]
Hash Collision Detection

Overview

• Algorithm given by Marc Stevens in his PhD thesis to detect collisions in MD5 and SHA-1.

• Uses a list of $L$ message block / working state differences to scan block compression for colliding message blocks.

• Total cost is $L$ hash computations.
Source: Marc Stevens’ PhD Thesis, Chapter 8, Figure 9
Hash Collision Detection
High Level Algorithm Description -- MD5

• There are a small number of working state differences corresponding to potentially successful differential paths.

• Differential paths require that at some step working states be the same, or all off in the top bit of the working state.

• Given the message, round functions can be run forward and backwards to recreate hash record

• Reconstruction takes same time as an MD5 block compression.

• False positive rate estimate $\frac{2^{22}}{2^{128}}$
Hash Collision Detection in CertRep

• Currently we have MD5 Collision Detection Implemented.
  • Cheap and easy to run in CertRep
    (already fast and not many MD5 certs remaining.)
• Planning to implement SHA-1 collision detection.
  • Begin scanning for reduced round collisions (any full collisions are likely to follow from known collisions.)
  • Update message block differences once actual SHA-1 collisions are found.
• Works very well in the COSMOS MapReduce programming model.
Factoring by Batch GCD

• Popular and high profile attacks on Weak RSA Keys: “Mining Your Ps and Qs: Detection of Widespread Weak Keys in Network Devices” and “Ron was wrong, Whit is right”

• Given a set of RSA moduli $\{N_i\}_{i=1}^m$ determines if any two distinct moduli share a prime factor, hence factoring both.

• If poor random number generation was used in RSA key generation this will detect weak keys.

• Tolga Acar implemented this approach at Microsoft Research.
Factoring by Batch GCD

Algorithm due to Bernstein:

• Uses the fact that:
  \[ \gcd(N_1, N_2N_3 \cdots N_m) = \gcd(N_1, N_1N_2N_3 \cdots N_m \mod N_1^2)/N_1 \]

Given the set of RSA moduli \( \{N_i\}_{i=1}^m \)

1. Compute \( P = \prod_{i=1}^{m} N_i \) using a product tree.
2. Compute \( z_i = (P \mod N_i^2) \) for each \( i \).
3. Compute \( g_i = \gcd(N_i, z_i) \mod N_i \) for each \( i \).
4. If any \( g_i \neq 1 \), \( N_i \) has been factored.
Factoring by Batch GCD

• CertRep provides a large set of RSA keys to feed into this algorithm.
• Does not lend itself to the Dryad computational model.
  • Keys need to be transferred out of COSMOS data store.
  • Batch factoring must be run on separate computational resources.
• Goal is to detect if a software bug is causing bad RSA keys to propagate (after initial research was done), or if weak keys are deployed in a customer enterprise CA that is not publicly visible.
Analysis of Certificates

Gather information about the public keys that are being used:
• Keep track of key strengths / bit lengths.
• Track algorithms in use, adoption of ECC.
Monitor “noncryptographic” fields in the certificates.
• Estimate bits of entropy in Serial Numbers.
• Subject name (look for close misspellings, etc.)
• Evaluate if field values are out of line with policy, statistically aberrant from other certs issued by the same CA.
• Monitor Certificate Extensions (e.g. Extended Key Usage.)
Future Directions

• Continue to work with Microsoft products to add sources of certificates to telemetry data.
• Extend analysis of public keys to other algorithms (e.g. ECDSA.)
• Automated detection / machine learning applied to certificate fields to search for potential cryptographic attacks or otherwise fraudulently issued certificates.
• Incorporate advances in public cryptanalysis in an ongoing fashion.
• Publicly share data gathered by CertRep to help inform industry wide certificate policy decisions.
Conclusion

• We are using recent advances in cryptanalysis to analyze the health of the CA ecosystem to protect our users.

• Individually any of the cryptographic analysis may not uncover an attack. However, this is a mitigation against the long time it takes to migrate from weakened cryptography.