Poster: Analysis of Reused Private Keys in the Code Signing PKI

Doowon Kim
University of Maryland
doowon@cs.umd.edu

S. Gokberk Karaca
Bilkent University
gokberk.karaca@ug.bilkent.edu.tr

Tudor Dumitraș
University of Maryland
tdumitra@umiacs.umd.edu

***Abstract*—In Public Key Infrastructures (PKIs) (e.g., the Web's PKI and the code signing PKI), the best practice for certificates to be newly issued from Certificate Authorities (CAs) is that applicants (e.g., publishers) should always generate a new pair of public/private keys for their new certificates. In the Web's PKI (e.g., TLS), the recent measurement studies have reported that public/private keys are commonly reused when new TLS certificates are reissued. However, the bad practice in the code signing PKI is not well understood even though it can result in more critical security threats than in TLS.**

In this paper, we collect a large scale of code signing certificates from one of the largest malware repositories, and analyze how the same public/private keys are reused for new code signing certificates. We find that there exists a third party that has requested code signing certificates for different publishers (with different common names and locality addresses); but the numerous binary samples, signed with the different certificates, are classified as only two labels of Potentially Unwanted Applications (PUAs). It may indicate that the two PUA families are very related to each other, or the PUAs may be controlled by the third party. We also find that the third party has continuously requested code signing certificates from CAs with their single public key, which suggests that the third party has a reliable process to obtain certificates from CAs.

I. Introduction

Public Key Infrastructures (PKIs) help establish trust between two different entities in an untrusted network. The trust between them is built by Certificate Authorities (CAs) who is responsible for issuing digital certificates that bind public keys with respective entities (i.e., organizations or individuals) after carefully verifying their identity. To obtain digital certificates from CAs, the entities need to generate a pair of public and private keys for a Certificate Signing Request (CSR), and submit the CSR to the CAs.

In the PKIs, the best rule of thumb for an applicant is to always generate a new key pair of public and private keys when new certificates are reissued. In other words, the public and private keys that have been used before should not be reused for certificates to be newly issued. However, unfortunately, the best practice is poorly followed in the real world. Specifically, in TLS, recent measurement studies [2] have reported that the same public/private keys are commonly reused when new TLS certificates are re-issued. It can lead to security threats such as a man-in-the-middle attack.

However, in the code signing PKI, little is known about how public/private keys are reused for newly issued certificates although the issue can lead to more critical security threats than in TLS. In particular, in TLS, since a TLS certificate is typically bound to a certain domain, the only one certain domain can be affected and abused. However, in the code signing PKI, a code signing certificate can be used to sign as many binary samples as publishers want so that a compromised certificate can be misused to sign numerous malware samples, which can result in a tremendous number of victims.

In this paper, we conduct the measurement studies on how the same public/private keys are reused when code signing certificates are newly issued. We find that 246 (1.3%) leaf certificates have public keys that have been previously used. The most common case is the simple applicants' mistake; they reused the same CSR files for their new certificates since the common names and locality address are the same or very similar. Interestingly, a single public key is reused for 52 certificates. It indicates that there is a third party that requests certificates for their clients (i.e., publishers) with the same third party's public key. The numerous binary samples signed with the certificates are classified as only two Potentially Unwanted Application (PUAs) labels, which may indicate that the signed binaries are controlled by the third party. We also find that the third party has continuously requested certificates from CAs with their single public key. It indicates that the third party has reliable process to obtain certificates from CAs.

II. Data Collection

To analyze how public/private keys are reused for new code signing certificates, we first need to collect signed PE binary samples and extract only valid code signing certificates from the samples. We then cluster the leaf code signing certificate by public keys, which helps find certificates issued with reused public/private keys.

Data Source. We utilize *VirusShare* to collect code signing certificates since the website is one of the largest repositories of malware samples. We download 319 out of 327 tar files; each tar file has 131,072 or 65,536 malware samples (including PE and non-PE samples).

Certificate Validation. Code signing certificates should be validated because the certificates may be altered by malware authors to bypass Anti-Virus protections for clients [1]. We utilize the Windows tools such as *SigCheck* and *SignTool* for validation.

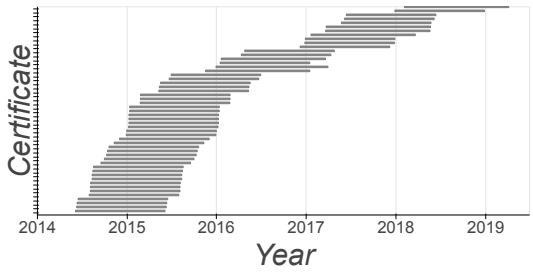


Fig. 1. **Certificates Timeline** – Each line corresponds to a different certificate that has the same public key (52 certificates in total). A third party continuously obtained certificates for their customers from CAs without any problems.

Malware Labeling. To better understand what kind of malware families are signed with the certificates issued with the reused public/private keys, we use ReversingLabs [1] to classify our collected binary samples. However, we first need to re-scan and re-analyze our collect binary samples because all binary samples collected in VirusShare cannot be always considered as malware due to false positives. VirusShare has collected binary samples just because the sample was classified as malware by only one AV engine.

III. Measurement Results

Summary of Data Collection. We obtain 30,146,559 malware samples from 319 tar files of *VirusShare*. Of these files, 5,114,527 (16.9%) are signed PE files. We first check if the signed PE files are properly signed using the Windows *SigCheck* and *SignTool* tools, which remains 4,696,140 (91.8%, out of 5,114,527) PE binary samples. We then extract 19,475 unique leaf code signing certificates from the properly signed PE files. On average, each certificate is used to sign about 241.1 PE files (σ: 3,230.3, min: 1, and max: 152,883) in our dataset.

Reused Public/Private Keys. We cluster our collected code signing certificates with each public keys. The most of certificates have their own unique public keys. In other words, when a new certificate was issued, a new pair of public/private keys were generated and used for the new certificate.

However, we find that 246 (1.3%, out of 19,475) unique leaf code signing certificates were issued with 90 reused public/private keys. A single public key is averagely reused for 2.7 certificates. Typically, the certificates have the exact same subject fields or slightly different subject fields such as common name (publisher name) and locality address. It indicates that applicants (publishers) used the same CSR files to request new code signing certificates from the same CAs that the applicants previously requested, and CAs never check if the provided public key was used before.

Surprisingly, we find that a certain public key was reused for 52 code signing certificates with very different common names, which indicates that there would exist a third party that requests code signing certificates for different companies (different common names and locality addresses) from CAs with the third party's same public key. We label the binary samples signed with the 52 certificates using ReversingLabs. The binary samples are classified as PUA, and labeled as *InstallCore* or *Dealply*; *InstallCore* is an installation and content distribution platform and *Dealply* is an adware that installs a browser extension to display advertisements in the browser. We can know that the two PUAs are very related to each other (The PUAs were developed or maintained by the third party). As shown in Figure 1, there is the issuance time of the 52 certificates. The third party has continuously requested certificates from CAs with their single public key. It has happened without any further investigations of CAs. It can indicates that the third party has a reliable process to obtain certificates from CAs.

The 246 certificates are used to sign in a total of 279,458 binary samples. Averagely, each certificate is used to sign 1,136 binary samples (σ: 10,760.1, min: 1, and max: 152,883). It indicates that the even small number of the certificates with the reused public/private keys have been used to sign numerous binary samples, which means that the owner of the certificates have been unaware of this security issue and have used the certificates to sign their binary samples.

Recommendations. Applicants who request code signing certificates should generate a new pair of public/private keys for their new certificates. In turn, CAs also should check the public keys given by applicants to see if the keys have been previously used. If previously used, CAs should ask them to regenerate keys.

IV. Conclusion

When certificates are newly issued, the best security rule of thumb is to generate a new pair of public/private keys. However, in the code signing PKI, the best practice is poorly followed in the wild. Specifically, a small number of code signing certificates were issued with reused public/private keys. We find that a single public key was reused for 52 different certificates for different publishers. It indicates that there would exist a third-party who requests code signing certificates for their customers (publishers) with the third party's same public key.

Acknowledgment

We thank ReversingLabs and VirusShare for access to their service. This research was partially supported by the National Science Foundation (award CNS-1564143) and the Department of Defense.

References

[1] D. Kim, B. J. Kwon, and T. Dumitraş, "Certified malware: Measuring breaches of trust in the windows code-signing pki," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '17, 2017.

[2] L. Zhang, D. Choffnes, D. Levin, T. Dumitraş, A. Mislove, A. Schulman, and C. Wilson, "Analysis of ssl certificate reissues and revocations in the wake of heartbleed," in *Proceedings of the 2014 Conference on Internet Measurement Conference*, ser. IMC '14. New York, NY, USA: ACM, 2014, pp. 489–502. [Online]. Available: http://doi.acm.org/10.1145/2663716.2663758

[1] It provides cyber threat detection and analysis solutions. https://www.reversinglabs.com

Analysis of Reused Private keys in the Code Signing PKI

Doowon Kim[1], S. Gokberk Karaca[2], and Tudor Dumitraș[1]
doowon@cs.umd.edu
University of Maryland, College Park[1], BilKent University[2]

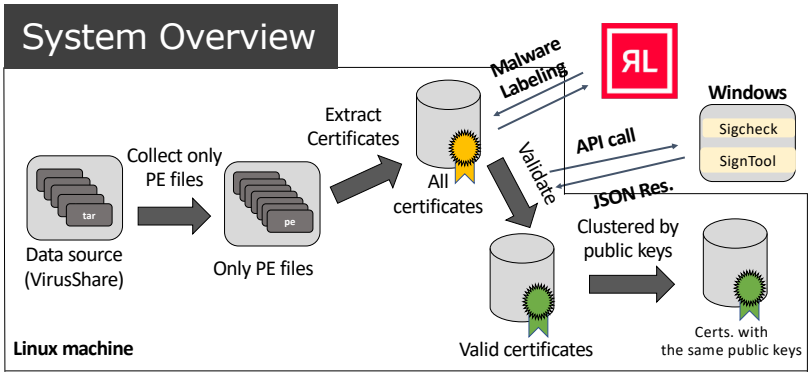
Research Questions

1. **Are public and private keys reused when certificates are newly issued?**
2. **How/why are they reused for the new certificates?**

Motivations

- Best practice when new certificates are issued:
  - Applicants (publishers) should generate a new pair of public and private keys.
  - CAs should check if public keys have been previously used.
- Potential security threats in the reused private keys:
  - If private keys are compromised and reused for new certificates, the new certificates becomes no longer valid.
  - The new certificates can be misused. (e.g., signing malware)
- In code signing PKI, **little** is known about the security issue
  - In TLS, this security issue has been well studied.

System Overview



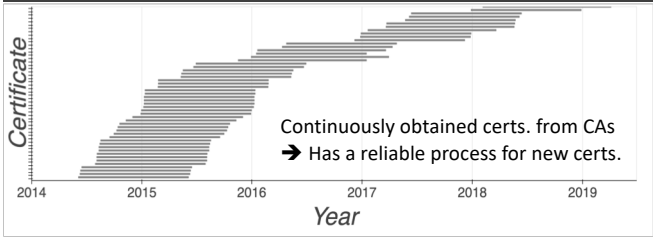
Summary of Data Collection

- 30,146,559 malware samples from *VirusShare.*
  - 5,114,527 (16.9%) signed PE samples.
  - 4,696,140 (91.8%) samples were properly signed.
- 19,475 unique leaf code-signing certificates.
  - Averagely, each certificate was used to sign 241.1 PE files
  - (σ : 3,230.3, min: 1, and max: 152,883)

Key Findings

- 1.3% certs. **reused** public/private keys.
- Used to sign 279,458 binary samples.
  - Each cert was averagely used to sign 1,136 samples.
    - (σ: 10,760.1, min: 1, and max:152,883)
  - Only 6% certs are explicitly revoked.
  - Publishers are **unaware** of this issue.
- Can be categorized into **two** groups.
  - **Reused the same CSR file.**
    - Issued with same common names and locality address.
  - **Third party.**
    - A public key reused for 52 certificates with different common names and locality address.
    - Malware Labels: *PUA.InstallCore, PUA.Dealply*
      - The two PUAs are controlled by the third party.
    - c.f., Timeline of 52 certificates.

Timeline of certs. by the third party.



Acknowledgment

- We thank ReversingLabs and VirusShare for access to their service. This research was partially supported by the National Science Foundation (award CNS-1564143) and the Department of Defense.