# MỘT SỐ NGUYÊN NHÂN RÒ RỈ KHÓA RIÊNG CHỨNG THƯ SỐ

**TS. Đỗ Quang Trung, TS.Nguyễn Văn Nghị, Đào Thị Thu Thủy,**

*Học viện Kỹ thuật mật mã*

*Bài viết trình bày về vấn đề để lộ lọt khóa riêng tương ứng chứng thư số người dùng trong hệ thống hạ tầng cơ sở khóa công khai. Cụ thể, bài viết đưa ra các nguyên nhân, tấn công phổ biến thời gian gần đây nhằm khôi phục, đánh cắp khóa riêng tương ứng chứng thư số người dùng trong hạ tầng cơ sở khóa công khai đang được sử dụng rộng rãi trong những năm gần đây.*



Ngày nay, công nghệ thông tin ngày càng phát triển mạnh mẽ ứng dụng vào trong mọi lĩnh vực của đời sống, các thông tin trao đổi trên mạng ngày càng nhiều và đa dạng. Kèm theo sự phát triển của mạng thông tin thì nhu cầu bảo mật thông tin trên mạng trở nên cấp thiết và được quan tâm nhiều hơn. Các giải pháp, phương pháp bảo mật được nhiều tổ chức đưa ra rất nhiều và đa dạng. Trong đó giải pháp chữ ký số, hạ tầng cơ sở khóa công khai (Public Key Infrastructure – PKI) là hệ thống vừa mang tính tiêu chuẩn, vừa mang tính công nghệ cho phép người dùng trong một mạng công cộng không bảo mật (như Internet), có thể trao đổi thông tin một cách an toàn thông qua việc sử dụng kỹ thuật mật mã với một cặp khóa riêng và công khai được chứng nhận bởi một nhà cung cấp chứng nhận số (Certificate Authority - CA) được tín nhiệm. Hệ thống PKI cung cấp một chứng thư số chứa khóa công khai, định danh người dùng, các trường thông tin liên quan và đi kèm là khóa riêng tương ứng cho người dùng. Một số ứng dụng của PKI cung cấp cho người dùng có thể kể đến: Mã hóa, giải mã văn bản; Xác thực người dùng ứng dụng; Mã hóa email hoặc xác thực người gửi email; Tạo chữ ký số trên văn bản điện tử.

Song song, hệ thống này cũng phải đối mặt với nhiều loại hình tấn công. Trong đó, tấn công khôi phục, chiếm đoạt khóa riêng tương ứng chứng thư số người dùng là vấn đề rất nghiêm trọng có thể dẫn tới quá trình hoạt động không đúng đắn, mất an toàn của hệ thống PKI. Một số nguyên nhân chính có thể dẫn tới các tấn công này có thể kể đến như sau:

- Lỗi từ phía người dùng: quá trình lưu trữ khóa riêng của người dùng sử dụng các thiết bị không an toàn như sử dụng chung USB không bảo mật, lưu trên thiết bị di động Android hay dịch vụ lưu trữ miễn phí trên Internet…

- Hệ thống triển khai PKI không an toàn: hệ thống triển khai PKI bao gồm các máy chủ cung cấp dịch vụ PKI, giao thức thực thi PKI, máy tính cài phần mềm thực hiện ký số của người dùng. Những máy tính này có thể bị kẻ tấn công cài cắm mã độc thông qua lỗ hổng bảo mật của hệ điều hành, phần mềm thực hiện PKI để đánh cắp khóa riêng.

- Điểm yếu về mặt toán học hay điểm yếu trong cài đặt hệ mật khóa công khai trong chứng thư số: Hệ mật khóa công khai đi kèm trong chứng thư số thường gặp nhất là RSA và ít hơn là các hệ mật Elliptic. Một người quản trị không có kiến thức về hệ mật khóa công khai khi triển khai hệ thống PKI thường sẽ cài đặt các hệ mật này tồn tại điểm yếu về mặt toán học như độ dài khóa ngắn, các tham số thực hiện không đảm bảo an toàn.

Bài viết này sẽ trình bày những nguyên nhân chính mà kẻ tấn công sử dụng để thực hiện tấn công khôi phục, chiếm đoạt khóa riêng tương ứng chứng thư số của người dùng trong các hệ thống PKI được áp dụng rộng rãi trên hệ thống mạng máy tính trong những năm gần đây.

## Sử dụng chứng thư số với khóa có độ dài 512 bit

Vào 8/2011, Mikko Hypponen đã có bài viết chi tiết về việc sử dụng những chứng thư số hợp lệ để ký tập tin có chứa mã độc [1]. Những chứng thư số hợp lệ này có những chứng thư số của chính phủ Mỹ được ký bởi các trung tâm chứng thực được chấp thuận bởi Mozilla hay Microsoft (Digisign Server ID (Enrich)/DigiCert Sdn). Ban đầu, Mikko cho rằng các chứng thư số này bị đánh cắp/lấy trộm khóa riêng, nhưng khi đi sâu vào điều tra nhận thấy rằng các chứng thư số này đều có độ dài khóa rất khiêm tốn là 512 bit. Việc sử dụng chứng thư số với độ dài như vậy dẫn tới việc kẻ tấn công tìm ra khóa riêng với các thuật toán phân tích số nguyên có độ dài 512 bit là hoàn toàn khả thi với năng lực của máy tính.

Nguyên nhân chính việc này là do các CA gốc của Mozilla hay Microsoft không đưa ra các luật cấm các chứng thư số có độ dài 512 bits được cấp phát bởi các CA trung gian ủy quyền. Tới tháng 6/2012 [2], Microsoft đã ban hành chính sách cấm tất cả việc chấp thuận chứng thư số có độ dài dưới 1024 bit trên nền tảng từ hệ điều hành (HĐH) Window XP trở đi: cụ thể cấm cài đặt, chấp thuận chứng thư số 1024 bit trên HĐH, trình duyệt Microsoft IE, trình duyệt Microsoft Edge, cấm sử dụng chứng thư số này ký số các Email trong Microsoft Outlook, cấm cài đặt các phần mềm được ký số với chữ ký độ dài 1024 bits. Các CA gốc nổi tiếng khác như Mozillla, Oracle… cũng thực hiện các chính sách tương tự cùng năm đó.

Tuy nhiên, Microsoft hay các nhà cung cấp dịch vụ mật mã vẫn cho phép tạo các khóa công khai RSA có độ dài tối thiểu 512 bit. Điều này, cho phép các người quản trị thiếu kiến thức mật mã vẫn tạo các khóa có độ dài 512 đến 1024 bit để áp dụng vào hệ thống PKI cho riêng tổ chức, cơ quan của mình (với các nguyên nhân như: để cấu hình mặc định hay đáp ứng yêu cầu tốc độ của hệ thống PKI). Việc thiết lập này không xảy ra cảnh bảo hoặc lỗi gì vì hệ thống PKI thiết lập trên nền tảng các HĐH Window cũ (trước 2012 và không cập nhật), HĐH Linux hay các trình duyệt Web, Mail Client phiên bản cũ hoặc phần mềm của các nhà phát triển không có chính sách gì về độ dài khóa trong chứng thư. Không riêng Microsoft mà các hệ thống PKI mã nguồn mở phổ biến như EJBCA, OpenCA, OpenSSL EasyCA đều mặc định sinh chứng thư số RSA có độ dài khóa 1024 bit (thậm chí vẫn hỗ trợ 512 bit).

Ngoài ra, để tăng tốc độ thực thi quá trình ký số, kiểm tra ký số thì một số hệ thống PKI tự phát triển hoặc cung cấp bởi bên thứ ba khác còn thực hiện cài đặt hệ mật RSA với các tham số đầu vào khóa riêng $d$ hoặc khóa công khai $e$ có giá trị rất nhỏ. Điều này dẫn tới các tấn công khôi phục khóa riêng RSA chỉ với xấp xỉ 0,009 giây được trình bày trong [3].

## Chiếm quyền/kiểm soát trái phép máy chủ cung cấp chứng thư số

Vào năm 2011, đã có những báo cáo xác thực rằng DigiNotar một công ty chứng thực số lớn của Hà Lan được chấp thuận bởi Google đã bị

tấn công. Công ty này đã không có công bố gì về các cáo buộc này. Phải đến 7/2011 [4], Fox-IT một công ty chuyên phân tích, đánh giá bảo mật hệ thống đã đưa ra bản thông báo về cuộc tấn công này. Fox-IT đã phân tích các dữ liệu log trên toàn bộ hệ thống của DigiNotar và đưa ra nhận định chính sau: Kẻ tấn công đã xâm nhập vào 8 hệ thống máy chủ CA của DigiNotar, có đầy đủ quyền quản trị và các quyền về các bản ghi cơ sở dữ liệu trên các máy chủ CA này. Qua phân tích log của các máy chủ CA thì thấy rằng kẻ tấn công đã có sự cấp phát chứng thư số mới giống nhau và giống với các chứng thư số trước đó đã có trên cơ sở dữ liệu: gồm số seri, hiệu lực, không bao gồm khóa công khai và mã định danh thực thể được cấp.

Những nguyên nhân chính mà kẻ tấn công xâm nhập được vào hệ thống của DigiNotar được Fox-IT đưa ra như sau:

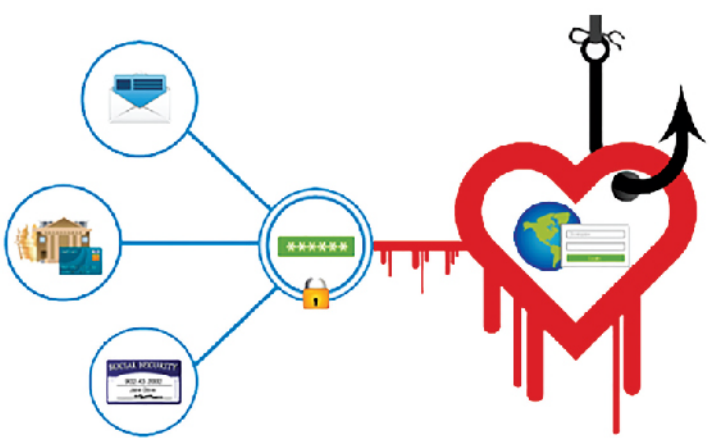
- Tất cả các chứng thư số máy chủ đều thuộc một miền Windows, cho phép sự truy cập của một người quản trị kiểm soát mọi thứ.

- Mật khẩu quản trị được đặt rất đơn giản và có thể dễ dàng bị tấn công vét cạn.

- Có sự phát hiện mã độc và các công cụ được sử dụng trong cuộc tấn công.

- Các phần mềm trên 8 máy chủ đều đã lỗi thời và chưa được vá lỗ hổng.

- Không có sự phân tách hiệu quả giữa các vùng mạng trong hệ thống.




Với các quyền hạn như vậy, kẻ tấn công hoàn toàn biết hết các thông tin chứng thư số, khóa riêng tương ứng đã cung cấp một cách hợp lệ cho người dùng.

### Lỗ hổng Heartbleed

Heartbleed là một lỗ hổng được phát hiện vào tháng 4 năm 2014 [5]. Heartbleed là một lỗ hổng trong OpenSSL, một thư viện mã nguồn mở dùng để triển khai (Transport Layer Security - TLS) và (Secure Sockets Layer - SSL), được Google, Facebook, Yahoo, Amazon và rất nhiều trang web lớn trên thế giới sử dụng để bảo vệ trong việc truyền tải thông tin cá nhân của người dùng.

Việc khai thác lỗ hổng liên quan tới cơ chế Heartbeat trong giao thức SSL/TLS. Cụ thể, một máy người dùng kết nối tới máy chủ dịch vụ được bảo vệ bởi SSL/TLS, thì sau thời gian không sử dụng kết nối người dùng muốn kết nối lại máy chủ và sử dụng các thông số phiên liên lạc trước đó mà không phải thiết lập lại phiên SSL/TLS từ đầu. Người dùng gửi gói Heartbeat request (các request có thể dài tới 64 KB) tới máy chủ dịch vụ với độ dài gói tương ứng, máy chủ dịch vụ sẽ gửi gói phản hồi với nội dung gói tin vừa nhận được lưu trữ trong bộ nhớ chủ động của máy chủ có độ tương ứng độ dài kèm theo gói gửi. Lỗ hổng ở đây là máy chủ không thực hiện kiểm tra độ dài gói tin request có đúng độ dài gói này đã gửi tới không. Trường hợp, người dùng (kẻ tấn công) sẽ gửi gói tin với nội dung ngắn giả sử 40 KB mà độ dài đính kèm là 64 KB tới máy chủ. Khi đó, máy chủ trả phản hồi lại gói tin với nội dung có độ dài 64 KB với 40 KB là nội dung gói tin gửi, còn 20 KB là nội dung khác trong bộ nhớ chủ động của máy chủ. Việc này cho phép kẻ tấn công có thể dễ dàng truy cập vào dữ liệu lưu trữ trong bộ nhớ chủ động của máy chủ dịch vụ. Điều này dẫn tới, kẻ tấn công có thể lấy được thông tin quan trọng như tài khoản/mật khẩu người dùng, hay khóa riêng của chứng thư số người dùng tương ứng.

Lỗ hổng này tồn tại trên các phiên bản OpenSSL 1.0.1 đến 1.0.1f. Các phiên bản trước đó không có cơ chế Heartbeat, còn các phiên bản sau đã thực hiện cơ chế vá lỗ hổng. Tuy nhiên, hiện nay vẫn tồn tại các máy chủ sử dụng các phiên bản OpenSSL có lỗ hổng này.

## Lỗ hổng CloudBleed

Cloudflare là một công ty cung cấp dịch vụ chứng thư số, tên miền, dịch vụ an toàn cho các máy chủ web của khách hàng [6]. Cloudflare thực thi một Proxy ngược ở giữa máy chủ dịch vụ và người dùng truy cập để thêm các tính năng bảo mật và cải thiện hiệu suất. Đầu năm 2017, một lỗi trong mã nguồn phía máy chủ Cloudflare đã được phát hiện làm rò rỉ các thông tin trong bộ nhớ chủ động máy chủ thông qua các thông điệp phản hồi HTML.

Theo công bố của Cloudflare, vấn đề bắt nguồn từ việc công ty quyết định sử dụng trình phân tích cú pháp HTML mới có tên cf-html. Trình phân tích cú pháp HTML là một ứng dụng quét mã để lấy ra thông tin có liên quan như thẻ bắt đầu và thẻ kết thúc. Điều này giúp người quản trị dễ dàng sửa đổi mã đó hơn.

Cloudflare đã gặp sự cố khi đưa định dạng mã nguồn của cf-html và trình phân tích cú pháp cũ Ragel của nó để hoạt động với phần mềm của riêng mình. Một lỗi trong mã đã tạo ra một thứ gọi là lỗ hổng chạy tràn bộ đệm. Lỗi liên quan đến tổ hợp ký tự "==" trong mã mà lẽ ra phải là tổ hợp ">=". Điều này có nghĩa là khi dịch vụ Cloudflare ghi dữ liệu người dùng vào bộ đệm, có kích thước hạn chế cho dữ liệu tạm thời, nó sẽ lấp đầy bộ đệm và sau đó tiếp tục ghi dữ liệu ở một bộ đệm khác. Mà bộ đệm khác này lại quy định để lưu trữ dữ liệu của một người dùng khác. Điều này cho phép người dùng khai thác một kỹ thuật nhất định để lấy một phần ngẫu nhiên thông tin cá nhân của một người ngẫu nhiên.

Lỗ hổng này tương tự như Heartbleed năm 2014, có thể làm rò rỉ bất kỳ dữ liệu bộ nhớ máy chủ Cloudflare tại thời điểm đó: internal headers, cookies, HTML POST bodies, mật khẩu và thậm chí cả khóa riêng tương ứng chứng thư số người dùng đã cấp.

## Lộ khóa riêng chứng thư số với cơ chế RSA-CRT

Hệ mật RSA, có số nguyên modulo $n = q * p$, với $q, p$ là hai số nguyên tố lớn có kích cỡ $\sqrt{n}$. Chi phí thực hiện phép toán trong hệ mật là $O((\log n)^2)$ sẽ tăng trưởng lên rất nhều khi n tăng kích cỡ. Để giảm chi phí, cần thực hiện tính toán hiệu quả riêng biệt phép toán trên các trường $Z(p)$ và $Z(q)$, cần thiết tối ưu đầu vào và kết quả đầu ra. Quá trình tối ưu này được thực hiện qua thực hiện tối ưu định lý phần dư Trung Hoa (Chinese Remainder Theorem - CRT). Với quá trình tối ưu hóa này thì Arjen Lenstra đưa ra một cuộc tấn công kênh kề bổ sung [7].

Cụ thể với tối ưu hóa CRT, phần lớn tính toán xảy ra trong vành $Z(p) \oplus Z(q)$. Nếu một lỗi xảy ra trong quá trình tính toán chữ ký số RSA ở chính xác một trong các trường, ví dụ trong $Z(p)$ và kẻ tấn công nắm giữ chữ ký lỗi $y$ tương ứng bản rõ $x$ được ký đó thì $1<gcd(y^e-x,n) < n$ trên các số nguyên, với e là thành phần khóa công khai RSA và đại diện cho số nguyên phù hợp $y^e$ mod $n$ đã được chọn. Điều này có nghĩa ước số chung lớn

nhất vừa tính tiết lộ một thừa số nguyên tố của $n$. Đây chính là kết quả mong muốn của cuộc tấn công, giả sử một lỗi như vậy xảy ra: vành đẳng cấu $\varphi$: $Z(n) \rightarrow Z(p) \oplus Z(q)$ với tối ưu hóa CRT có thể được thực hiện dưới dạng mô-đun giảm $p$ và $q$ trong các trường tương ứng. Giả sử $\varphi(y)=(y_p,y_q)$ cho một chữ ký $y$ bị lỗi trong thành phần $y_p$, thì lỗi $\varphi(y^e)=(y_p^e, y_q^e)$ vẫn bị hạn chế thành phần đầu tiên. Viết thông báo $x$ dưới dạng $\varphi(x)=(x_p^e, x_q^e)$, chúng ta thấy rằng $\varphi(y^e-x)=(a,0)$ đối với một vài giá trị $a \in Z(p)$. Điều này có nghĩa bất kỳ đại diện $(y^e-x)$ mod n đều là một bội số của $q$. Nếu đại diện nhỏ hơn $n$ (luôn có thể sắp xếp), ước số chung lớn nhất của $(y^e-x)$ mod $n$ là $q$.

Các nguyên nhân có thể gây lỗi trong quá trình ký số RSA:

- Sử dụng các thư viện cũ hoặc tồn tại lỗ hổng dẫn tới các phép toán thực hiện sai trên trường số nguyên. Ví dụ, lỗ hổng CVE-2014-3570 là một sự cố khiến các phép toán bình phương của OpenSSL không hoạt động bình thường đối với một vài đầu vào.

- Việc chạy đua tốc độ ký số của các chương trình ký số bởi các nhà cung cấp, dẫn tới thực thi đa luồng và có thể gây lỗi.

- Đơn vị số học của CPU hỏng hoặc bị quá tải.

- Lỗi trong bộ nhớ CPU, các bộ nhớ đệm khác hoặc bộ nhớ chính.

- Các phần quan trọng của khóa riêng có thể đã bị hỏng sau khi việc kiểm tra tính toàn vẹn (nếu có) đã được thực hiện, khiến tất cả các chữ ký trong tương lai tiết lộ khóa riêng.

Các máy chủ TLS sử dụng chứng thư số RSA có thực thi RSA-CRT để tối ưu hóa là mục tiêu tấn công. Kẻ tấn công sử dụng các thuật toán Lenstra này để khai thác lỗ hổng trong thực thi RSA-CRT và tìm được khóa riêng chứng thư số.

Trên đây là những nguyên nhân chính gây ra việc bị lộ khóa riêng tương ứng chứng thư số người dùng. Việc lộ khóa riêng đối với người dùng mạng thông thường sẽ dẫn tới nguy cơ bị giả mạo bởi kẻ tấn công: ví dụ thực hiện các giao dịch tiền tệ hay liên lạc đối tác kinh doanh trên danh nghĩa người dùng. Đối với bên máy chủ cung cấp dịch vụ như Web, Email thì có thể dẫn tới thêm các nguy cơ về việc sử dụng khóa riêng để ký vào văn bản, phần mềm có chứa mã độc từ kẻ tấn công.

## Kết luận

Việc lộ khóa riêng người dùng từ điểm yếu hệ thống PKI hay điểm yếu quá trình cài đặt các hệ mật khóa công khai trong PKI là tồn tại và là phổ biến trong các hệ thống PKI tự phát triển hoặc cung cấp bởi bên thứ ba có mức uy tín thấp. Hậu quả các cuộc tấn công này dẫn tới là nghiêm trọng, gây thiệt hại lớn. Yêu cầu đặt ra cần có sự hiểu biết thấu đáo về các tấn công này từ phía người quản trị viên CA để có thể thực hiện, triển khai PKI được hiệu quả và đúng đắn.❖


**TÀI LIỆU THAM KHẢO**

1. Michael Sandee, RSA-512 Certificates abused in the wild, Fox-IT, 2011.
2. Kurt L Hudson MSFT, Blocking RSA Keys less than 1024 bits, 2012. https://www.sysadmins.lv/retired-msft-blogs/pki/rsa-keys-under-1024-bits-are-blocked.aspx
3. Filipe Boucinha: A Survey of Cryptanalytic Attacks on RSA, The thesis for the Degree of Master in Mathematics and Fundamental Applications, University of Technical Lisboa, 2011.
4. Fox-IT, Black Tulip: Report of the investigation into the DigiNotar Certificate Authority breach, 2011.
5. Biggs, John, Heartbleed, The First Security Bug With A Cool Logo, TechCrunch, 2014.
6. John Graham-Cumming, Incident report on memory leak caused by Cloudflare parser bug, cloudflare.com, 2017.
7. Florian Weimer, Factoring RSA Keys With TLS Perfect Forward Secrecy, Red Hat Product Security, 2015.