7-2 Analysis System (XPIA)

Takashi KUROKAWA, Ryo NOJIMA, and Shiho MORIAI

Secure Socket Layer(SSL)/Transport Layer Security(TLS) are cryptographic protocols widely used from e-government systems to network services immediately surrounding people, for example, internet banking and online shopping. At the same time, the cryptographic attacks against SSL/TLS have constantly been evolved and sophisticated in step with the development and popularization of online services.

In this paper, we show an outline of our research and development for the recent attacks.

1 Introduction

Varieties of network-based services, such as e-commerce and online shopping, are widely available on the Internet. SSL/TLS is an underlying technology often used to ensure the information security of such services. When a user needs to authenticate the destination of his/her connection, Public Key Infrastructure (PKI) technology is used to verify the validity of the information stored in the public key certificate. In 2012, two groups of researchers — N. Henninger et.al., and A.K. Lenstra et.al. — independently reported that many public keys generated for use in server certificates might have the same secret key (a prime number) due to several reasons, including a bias in the random numbers used to generate RSA keys [1][2]. The factorization of a large number into prime factors is considered to be difficult to compute, providing the security of RSA. However, to compute the greatest common divisor is an easy task. Thus, if two RSA public keys have the same secret key (a prime number) in common, so-called "collision," the secret key can easily be revealed by calculating their greatest common divisor (Fig.1). Exposing a secret key may lower the hurdle to forgery of server certificates and other fraudulent acts. Several reports have mentioned the number of servers with such vulnerability, but only a few have given detailed information — e.g. the specific location of the servers. Against this background, we made an investigation into the following question: How many vulnerable RSA public keys are present in the .JP domain? The results will be reviewed in Section **2**.

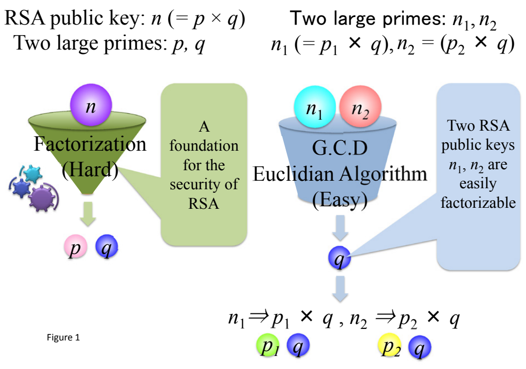
RSA public key: n (= $p \times q$)
Two large primes: p, q
Two large primes: n_1, n_2
n_1 (= p_1 × q), n_2 = (p_2 × q)
n
Factorization (Hard)
A foundation for the security of RSA
p
q
n_1
n_2
G.C.D Euclidian Algorithm (Easy)
Two RSA public keys n_1, n_2 are easily factorizable
q
$n_1 \Rightarrow p_1$ × q , $n_2 \Rightarrow p_2$ × q
p_1
q
p_2
q
Figure 1

Fig. 1 Sharing of secret keys in RSA public keys

SSL/TLS-based secure communication uses a combination of cryptographic primitives such as RSA, DH, AES, RC4, CBC mode, and HMAC. The following examples represent typical vulnerability cases reported in recent years: BEAST attacks and POODLE attacks that take advantage of a CBC mode vulnerability in SSL3.0 and TLS1.0 as well as web browser bugs, and attacks that make use of passwords stolen from encrypted cookies. In the latter case, biases in the RC4 key stream are used to statistically estimate the plaintext, which in turn is used to steal the password.

In the case that a block cipher is used in the encrypted communication, according to the specifications of SSL3.0 and TLS1.0, the CBC mode is used and the initialization vector IV is selected from the last block of the ciphertext encrypted in the last communication (Fig.2). Although this implicates predictability of the vector IV that may undermine indistinguishability, a violation of indistinguishability does not necessarily allow the attackers to restore the original plaintext. However, as T. Duong and J. Rizzo have demonstrated [3], the use of the BEAST attack method enables the attacker to decrypt the original plaintext within reasonable complexity under the condition that the web browser and peripheral software have one or more

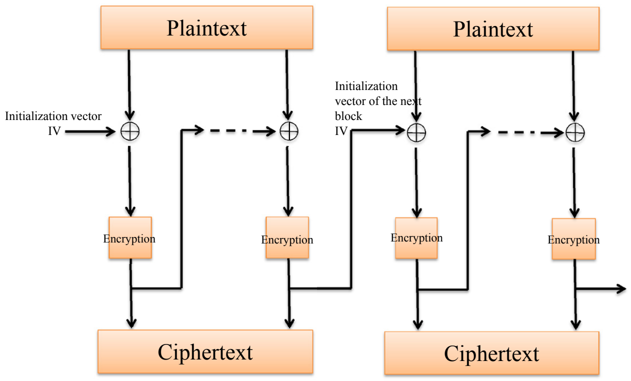

Fig. 2 CBC mode in SSL3/0 and TLS1.0

bugs related to SOP (Same Origin Policy). Alarmed by this, browser vendors have released CBC mode security patches in view of coping with the possibility that there may be many other unknown SOP-related bugs. These security patches, endorsed to apply from the viewpoint of maintaining interconnectivity, are collectively called the *1/n-1 Record Splitting Patch*. We demonstrated the validity of this splitting patch from the point of security, i.e. it satisfies IND-CPA security. The results are reviewed in Section **3**.

2 Vulnerability of RSA public keys

In this section, a review is made on the contents we reported in reference [4].

2.1 Collection of public key certificates and extraction of RSA public keys

To collect public key certificates, we took advantage of those that had been already collected by SSL Observatory [5], rather than collecting them anew by scrawling all IPv4 address space. The descriptions in the public key certificates are based on the X.509 specification. In the case of RSA, public keys are stored in the modulus field, which is contained under the *subjectPublicKey* field, after being converted using DER encoding.

2.2 Analysis of RSA public keys

In the search for weak RSA public keys from a vast number of public key certificates, the use of simple GCD (Greatest Common Divisor) calculations between the RSA public keys could take a very long time — for n RSA public keys, the amount of calculation would be of the order of n^2. However, using a pair-by-pair binary tree approach can reduce the amount of calculation to the order of $n \log n$. At the time of this study, we extracted 2,742,833 public key certificates that each contained a 1,024-bit RSA public key from those collected by SSL Observatory (as of 2010). We checked the occurrence of common factors among them and succeeded in detecting 8,703 weak RSA public keys.

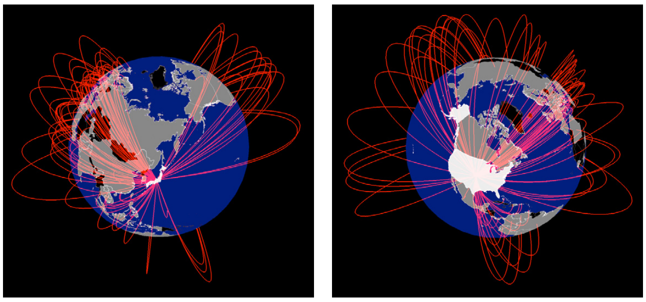

Fig. 3 Showing of common factor sharing (Left: viewed from Japan, Right: viewed from USA)

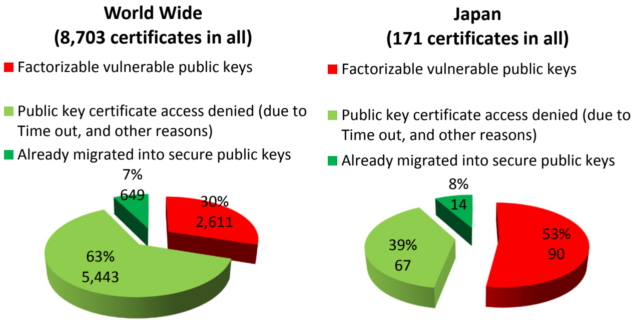

Fig. 4 Ratio of public key certificates with weak public keys (as of around Oct. 2013)

2.3 Visualization

Normally, certain types of useful information (such as nation and host name) are obtainable from the fields in a public key certificate that help identify the user who owns a weak RSA public key. In many cases, however, the *subject* field in the public key certificates in which a weak RSA key was detected did not even contain the nation information. To overcome this problem, we made reference to the IP address allocation information published by regional Internet Registry (AfriNIC, ARIN, APNIC, LACNIC, RIPE NCC) to assign the nation information(Fig. 3).

2.4 Crawling and log information analysis

Because of the possibility that the vulnerable public key certificates may have been updated, a crawler was made to take possession of the latest version. At the time of this investigation, we could not reach 5,443 public key certificates. Examination of the residual 3,260 public key certificates that we could access revealed that 2,611 hosts still used the type of RSA public keys that could be factored into prime numbers. Among the 171 public key certificates used in the .JP domain, 90 hosts still used factorizable RSA public keys (Fig.4).

To investigate the types of hosts that still possess factorizable public keys, attempts were made to connect to the 2,611 hosts to review their top pages. At the time of this investigation, 2,233 hosts (top pages) out of 2,611 were available to us. See reference [4] for detailed descriptions.

2.5 Cooperation with JIPDEC

The information described above was transferred from NICT to the Japan Information Processing and Development Center (JIPDEC) for analysis. The public keys contained in the self-signed certificates, which are treated by the accreditation of specified certification business under the Japanese law on electronic signatures, were examined at JIPDEC (August 2014) and verified to pose no vulnerability risks [6]. Note that the public key certificates used in this investigation were from the data set collected by https://scans.io/, and not from those collected by SSL Observatory.

3 Provable security of patched CBC mode in TLS1.0

In this section, a review is made on the contents of reference [7]–[9].

3.1 Patches in TLS10

According to reference [10], TLS1.0 is currently most widely supported among SSL/TLS, and the CBC mode is the mode of choice in many ciphersuites. Because the message authentication code (MAC) is not applied to padding in the CBC mode of TLS1.0, a vulnerability may become present if the padding error and MAC error are distinguishable. Existence of a type of attack, called a padding oracle attack, was known to take advantage of this [11]. Several methods were proposed to avoid this flaw, including one that added a blank fragment. However, these were not adopted formally as a patch in view of their possible interference with interconnectivity [12]. As the need for measures to fix the flaw became more pressing due to the discovery of the BEAST attack, the *1/n-1* record splitting patch was adopted, which was proposed by reference [13] and verified to have no adverse impact on interconnectivity. This method, i.e. *1/n-1* record splitting, first splits the plaintext into two chunks (the leading byte and remaining bytes) and encrypts each independently (Table 1). Let us define this approach collectively – CBC mode encryption after an application of *1/n-1* record splitting, and original message authentication codes of TLS1.0 - as SplTLS1.0.

Table 1 An example of 1/n-1 record splitting

The numbers in parentheses: bytes, C>S: transmission from client to server, S>C transmission from server to client.

In the case of AES, applying the patch converts the first application data into a 32-byte ciphertext. In this case, therefore, the patch is applied only to the client.

C>S	V3.1	(1)	ChangeCipherSpec
C>S	V3.1	(48)	Handshake
S>C	V3.1	(170)	Handshake
S>C	V3.1	(1)	ChangeCipherSpec
S>C	V3.1	(48)	Handshake
C>S	V3.1	(32)	application_data
C>S	V3.1	(80)	application_data
S>C	V3.1	(328)	application_data
S>C	V3.1	(608)	application_data

3.2 Security of patched CBC mode

The proof of Theorem 1 below is given in reference [7] (refer to it for the details). A probabilistic polynomial time algorithm $\mathcal{K}$ was used to generate the key K. For evaluation, a deterministic polynomial time algorithm $\mathcal{F}$ was used, which accepts the key K and x as input, and outputs $\mathcal{F}\ (K, x)$.

Definition 1. $\mathfrak{P} = (\mathcal{K}, \mathcal{F})$ is a pseudorandom function (PRF) if the difference in probability is negligible* for the following two probability outputs: the probability an arbitrary probabilistic polynomial time algorithm $\mathcal{A}$ outputs the value of 1, when $\mathcal{F}(K,\cdot)$is executed using a randomly selected key K, and the probability $\mathcal{A}$ outputs the value of *1*, when another function $\mathcal{F}'$ (selected arbitrarily from those functions that have the same domain of definition and range of values with $\mathcal{F}(K,\cdot)$) is executed. If $\mathcal{F}(K,\cdot)$ is a permutation, then $\mathfrak{P}$ is called a pseudorandom permutation (PRP).

If $\mathcal{E}$ is a cipher algorithm, the following relation is assumed: $\mathrm{LR}_{K,\,b}(M_0, M_1) = \mathcal{E}(K, M_b)(b \in \{0,1\})$.

Definition 2. Symmetric key cryptography $\mathcal{SE} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ is IND-CPA secure if: a key K, $b \in \{0,1\}$, and $\mathrm{LR}_{K,\,b}$ are executed against an arbitrary probabilistic polynomial time algorithm $\mathcal{A}$, producing output b', and the absolute

* A function ε is "negligible" if: a certain value k exists for arbitrary value $c>0$, and $\varepsilon(n)<1/n^c(n \geq k)$ holds.

value of the difference between the following two values is negligible – the probability of being b=b' for a randomly selected $b' \in \{0,1\}$, and ½.

Definition 3. $\mathcal{MA} = (\mathcal{K}, \mathcal{T}, \mathcal{V})$ is a message certification code if: the verification algorithm $\mathcal{V}$, a deterministic polynomial time algorithm, outputs *0* or *1* for the set of inputs K, M, t , where the key K is produced by a probabilistic polynomial time algorithm (key generating algorithm), and the tag t is produced by a deterministic polynomial time algorithm (tag generation algorithm) using the key K and a plaintext M as input. $\mathcal{MA}$ is complete if $\mathcal{V}(K, M, t) = 1$ and $t = \mathcal{T}(K, M)$ are equivalent. Note that $\mathcal{MA}$ is called a pseudorandom function if $\mathcal{T}(K, \cdot)$ is a pseudorandom function.

Definition 4. $\mathcal{MA} = (\mathcal{K}, \mathcal{T}, \mathcal{V})$ is IND-CPA secure if: a key K, $b \in \{0,1\}$, and $\mathrm{LR}_{K,b}$ are executed against an arbitrary probabilistic polynomial time algorithm $\mathcal{A}$, producing output b', and the absolute value of the difference between the following two values is negligible – the probability of being b=b' for a randomly selected $b' \in \{0,1\}$, and ½.

Theorem 1. SplTLS1.0 satisfies IND-CPA if the following conditions hold: $\mathfrak{P}$ is a random permutation, and $\mathcal{MA} = (\mathcal{K}, \mathcal{T}, \mathcal{V})$ is a pseudorandom function and complete.

4 Conclusions

We originally started a project to construct a system to verify the vulnerability of RSA public keys used in SSL, and named it "XPIA (X.509 certificate Public-key Investigation and Analysis system)." However, the scope of the research was later expanded to include the security proof of 1/n-1 record splitting patch as applied to TLS1.0. Therefore, the name XPIA has become a generic name to represent research covering the security of SSL/TLS. We continue to direct our attention to the research trends on SSL/TLS, and hope that the research results will be beneficially utilized for the security and reliability of the cryptographic technology used in the operation of e-Government and other fields (such as the accreditation of the specified certification business in Japan).

Acknowledgments

We would like to take this opportunity to express our appreciation to all those who helped us in many aspects of the XPIA project, including preparation of press releases and technology transfer.

References

1 N. Heninger, Z. Durumeric, E. Wustrow, and J.A. Halderman, "Mining Your Ps and Qs:Detection of Widespread Weak Keys in Network Devices," USENIX Security 2012, 2012.

2 A.K. Lenstra, J.P. Hughes, M. Augier, J.W. Bos, T. Kleinjung, and C. Wachter, "Pub-lic Keys," CRYPTO 2012, LNCS 7417, pp.626–642, 2012.

3 T. Duong and J. Rizzo. "Here Come The ⊕ Ninjas," http://netifera.com/research/beast/beast_DRAFT_0621.pdf , May 2011.

4 T. Kurokawa, R. Nojima, and S. Moriai, "After the "Mining Your Ps and Qs," CSS2013, 2013. (In Japanse)

5 The SSL Observatory, Available from https://www.eff.org/observatory/ (2013-08-26)

6 National Institute of Information and Communications Technology, Japan Information Processing Development Center,"Security validation of designated certification businesses that supports such electronic systems as electronic biddings, electronic filings and electronic contracts," December 17th 2014, NICT press release, http://www.nict.go.jp/press/2014/12/17-1.html (In Japanse)

7 T. Kurokawa, R. Nojima, and S. Moriai, "On the Security of TLS1.0 CBC Mode," SCIS2014, 2014. (In Japanse)

8 T. Kurokawa, R. Nojima, and S. Moriai, "Can We Securely Use CBC Mode in TLS1.0?," ICT-EurAsia/CONFENIS 2015: 151–160.

9 T. Kurokawa, R. Nojima, and S. Moriai, "On the security of CBC Mode in SSL3.0 and TLS1.0," Journal of Internet Services and Information Security, 6(1): 2-19, Feb. 2016.

10 SSL Pulse, "Survey of the SSL Implementation of the Most Popular Web Sites," https://www.trustworthyinternet.org/ssl-pulse/

11 S. Vaudenay, "Security Flaws Induced by CBC Padding - Applications to SSL, IPSEC, WTLS ...," EUROCRYPT 2002, pp.534–546.

12 Bodo Moeller, "Security of CBC Ciphersuites in SSL/TLS:Problems and Countermeasures," http://www.openssl.org/~bodo/tls-cbc.txt

13 X. Su,"Bugzilla Bug 665814 Comment 59," https://bugzilla.mozilla.org/show_bug.cgi?id=665814#c59 , July 2011.

Takashi KUROKAWA

Technical Expert, Security Fundamentals Laboratory, Cybersecurity Research Institute
Security Evaluations of Cryptographic Technologies

Ryo NOJIMA, Ph.D.

Senior Researcher, Security Fundamentals Laboratory, Cybersecurity Research Institute
Cryptography, Cryptographic Protocol

Shiho MORIAI, Ph.D.

Director of Security Fundamentals Laboratory, Cybersecurity Research Institute
Cryptography, Information Security