“Mining Your Ps and Qs” のその後

黒川 貴司† 野島 良† 盛合 志帆†

†情報通信研究機構 ネットワークセキュリティ研究所
〒184-8795 東京都小金井市貫井北町 4-2-1
{blackriver, ryo-no, shiho.moriai}@nict.go.jp

あらまし 2012 年に Heninger らと Lenstra らが, それぞれ独立に公開鍵証明書に含まれている多くの公開鍵に脆弱性があることを報告した. 具体的には, RSA の公開鍵の素因数が多数の公開鍵において共有されており, 最大公約数を計算することによって, 素因数分解ができてしまうといったものであった. 彼らの報告は大きな衝撃を与えたが, 例えば JP ドメインに属する証明書がどうなっているか等, 詳細は不明であった. そこで本稿では, 彼らが報告した問題の再調査を行い, その脆弱性が JP ドメインに含まれる証明書にどの程度存在するか, また, どのように改善されたかなどを報告する.

After the “Mining Your Ps and Qs”

Takashi Kurokawa† Ryo Nojima† Shiho Moriai†

†Network Security Research Institute, NICT
4-2-1 Nukuikitamachi, Koganei, Tokyo 184-8795, JAPAN
{blackriver, ryo-no, shiho.moriai}@nict.go.jp

Abstract In 2012, Heninger et al. and Lenstra et al. independently found that vulnerable public keys in SSL certificates are widely spread all over the world. They pointed out that the problem lies in RSA public keys: many prime factors are shared among distinct public keys and then can be easily extracted by computing GCDs. Their research results had a huge impact on the security community. However, as far as we know, the details such as how many vulnerable keys belong to the JP domain have not been reported so far. In this paper, we investigate the problem they pointed out and report the present status of those vulnerable keys.

1 はじめに

電子商取引, インターネットショッピングなどのオンラインサービスが広く普及したことにより, 膨大な量の情報がネットワークを介して行き交うようになった. こういったサービスを提供する多くのサイトにおいて, SSL/TLS (Secure Socket Layer, Transport Leyer Security) と組み合わせたプロトコル HTTPS を使うことが主流となっている.

SSL/TLS は最も広く利用されているセキュリティプロトコルである一方, 軽微なものまで含めると, 多くの脆弱性が報告されている. 近年の代表的なものに, SSL/TLS サーバからの応答時間を計測することにより秘密鍵を導出してしまう Brumley らのタイミング攻撃 [3], CBC モードの弱点とブラウザのバグを利用した BEAST 攻撃 [6], 圧縮アルゴリズムのメッセージ毎の圧縮率の違いを利用した CRIME 攻撃 [7], また, CRIME 攻撃を発展させた BREACH 攻撃があ

る [8]. さらには RC4 のキーストリームの統計的偏りを利用し, 暗号化されたクッキーに含まれるパスワードを盗み出す攻撃手法 [10, 1] まで報告されている. 本稿では, こういった多くの脆弱性報告の中でも, 特に [9, 11] に注目する.

2012 年, Heninger らと Lenstra らは, それぞれ独立にインターネット上で公開されている RSA 暗号の公開鍵が無視できない割合で破れること, すなわち素因数分解ができるいう衝撃的な報告を行った [9, 11]. その原理は非常に単純であり, インターネットで公開されている多くの RSA 暗号の公開鍵が共通の素数を共有しているというものである. 素数定理より暗号用途の素数は十分に多いと考えられることから, 彼らの報告は想定外であり, 社会に大きな衝撃と不安を与えた.

[9, 11] の発表から約一年が経過し, 彼らの成果に刺激され, いくつかの研究 [12] が報告されている. しかしそれは, 「素数 (乱数) の生成」という根本的な問題を解決することを目的にしたものであり, 彼らが破った公開鍵はどこで利用されているものなのか, またその鍵は現在も使われているのかなどの報告は少ない [4].

そこで本研究では, 下記の調査を行った.

調査 1 [9, 11] において素因数分解された公開鍵の中で, JP ドメインで利用されている公開鍵

調査 2 上記公開鍵の中で, 現在も利用されている公開鍵

本調査を実施するにあたり, 本研究では, [11] において利用された公開鍵を, [9, 11] と同様の手法で解析した. また, それに加えて公開鍵証明書をダウンロードするクローラを作成し, [9, 11] で素因数分解された公開鍵が現在も使われているかなどの調査を行った. 本稿では, その詳細を紹介する.

2 Heninger らと Lenstra らの研究概要

Heninger らと Lenstra らの報告 [9, 11] の再調査を行うため, その概要を紹介する.

Step 1: クローラなどで公開鍵証明書 $C_1, \ldots, C_m$ を収集する.

Step 2: 公開鍵証明書 $(C_1, \ldots, C_m)$ の中から RSA 公開鍵暗号の公開鍵 $(N_1, e_1), \ldots, (N_n, e_n)$ を抽出する. ただし, $n \leq m$, N_i は合成数, e_i はべき指数とする.

Step 3: 全ての N_i, N_j $(N_i \neq N_j)$ について, 最大公約数を計算する.

Heninger らと Lenstra らが収集した公開鍵証明書は, 基本的には, IPv4 のアドレス空間全体からダウンロードしたものである. Heninger らは, クローラの作成を自ら行い, Amazon EC2 上で実行した. 一方, Lenstra らが調査した公開鍵証明書の大部分は, SSL Observatory [13] が収集したものである.

SSL/TLS に限定した場合, Heninger らは, クローリングにより約 580 万本の異なる公開鍵証明書を収集し, 23,576 個の RSA 公開鍵を素因数分解することに成功した. 同じ証明書 (合成数) を利用したホストが多数存在することから, 結局, これは 64,081 台の SSL/TLS ホストの鍵が解読できたことになる. 一方, Lenstra らは, 約 620 万本の公開鍵証明書を収集した. その中で, 異なる RSA 公開鍵は合計 5,989,523 個であった. 最終的には, 12,934 本の RSA 公開鍵を素因数分解することに成功した.

3 実施事項

3.1 解析方法

本研究では, 図 1 に従い公開鍵の解析を実施した.

公開鍵証明書の収集: SSL Observatory が収集した公開鍵証明書を利用する.

RSA 公開鍵の抽出: 全ての公開鍵証明書から RSA 暗号の公開鍵を抽出する.

RSA 公開鍵の解析: RSA 公開鍵の合成数 $(N_1, \ldots, N_n)$ について, $\{\gcd(N_i, N_j) \mid 1 \leq i \neq j \leq n\}$ を計算する. $\gcd(N_i, N_j) \neq 1$ であ

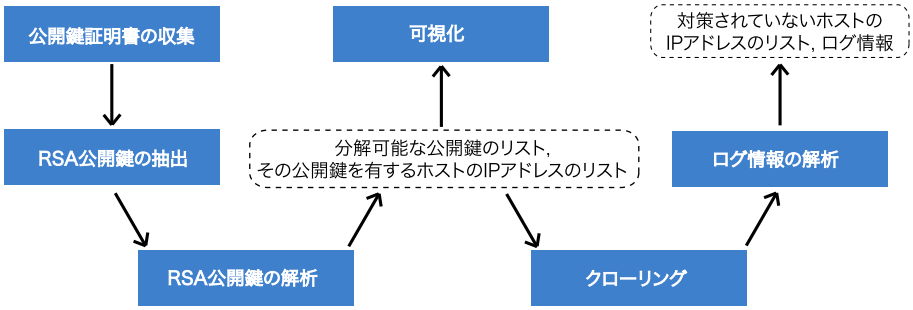

図 1: 本研究における公開鍵解析方法の概要

る場合, N_i, N_j を素因数分解できることを意味している. ただし, 処理時間の短縮のために実際の計算はこれとは異なる.

本処理により, 素因数分解可能な RSA 公開鍵とそれに対応する公開鍵証明書とホストの IP アドレスが得られる.

可視化: 「**RSA 公開鍵の解析**」により得られた情報を元に, 可視化を行う.

クローリング: 「**RSA 公開鍵の解析**」の結果を使い, 脆弱な公開鍵を有する, あるいは有していたホストから最新の公開鍵証明書をダウンロードする.

ログ情報の解析: 「**RSA 公開鍵の解析**」で得られた脆弱な公開鍵と「**クローリング**」により得られた最新の公開鍵が同じか否かを検証する.

3.2 SSL Observatory について

SSL Observatory [13] は, 世界中で利用されている公開鍵証明書の状況を調査することを目的として, IPv4 アドレス空間において入手可能な公開鍵証明書を収集している. 抽出により得られた公開鍵の内訳は表 1, 2 の通りである.

表 1: SSL Observatory の RSA と DSA の内訳

	RSA	DSA
個数	4,019,596	1907

表 2: SSL Observatory の RSA 公開鍵のサイズ別の内訳

ビット	512	1024	2048
個数	71,315	2,783,867	1,064,038

3.3 RSA 公開鍵の解析について

SSL/TLS では, ハンドシェイク時に公開鍵証明書のやり取りを行っている. 公開鍵証明書は ITU 及び IETF において標準化された X.509 に基づいて記述されている. 公開鍵証明書の構造を図 2 に示す. RSA の場合, 公開鍵証明書における subjectPublicKey フィールド内の modulus フィールドに DER エンコーディングにより変換されて格納された RSA 公開鍵を抽出する.

数百万もの公開鍵証明書の中から脆弱な RSA 公開鍵を抽出する場合, 上記 3.1 節の Step 3 で述べたような単純に合成数同士の GCD を計算する方法では, 対象とする鍵の数 n^2 のオーダーに比例するために非常に多くの時間を要することになる. 実際, 16 コア, 2.3 GHz の AMD Opteron 6276 を 4 CPU 搭載するサーバーにおいて, SSL Observatory が収集した公開鍵証明書 (2010 年) のうち, 2,783,867 個の 1,024 ビット合成数同士の GCD を計算するのに, GMP ライブラリを利用して, 62 スレッドで約 5.5ヶ月を要した. つまり, 約 28.4 CPU year である.

整数 N_i $(1 \leq i \leq n)$ の GCD を同時に計算する方法としては Heninger らが行った方法と Lenstra らが行った方法が知られている.

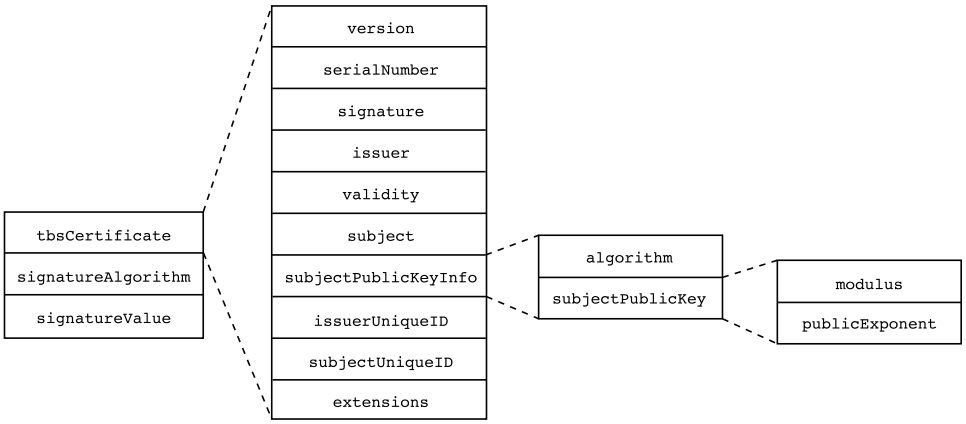

図 2: X.509 公開鍵証明書の構造 (RSA の場合)

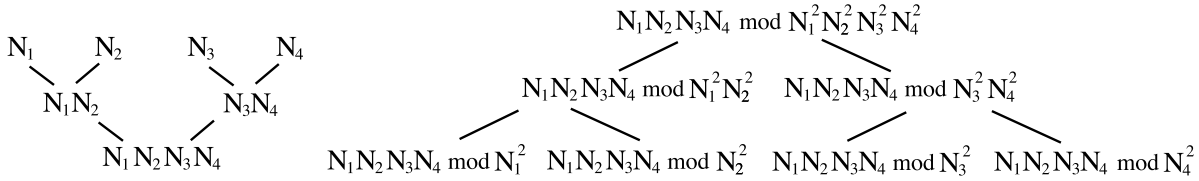

図 3: Product Tree(左) と Remainder Tree(右)　($n = 4$ の場合)

Heninger らは Product Tree 及び Remainder Tree という計算手法を利用している [2, 5]. すなわち, 合成数 $N_i\,(1 \le i \le n)$ のペアごとの積を 2 分木状に計算していって総積 $\Pi = \Pi_{1 \le i \le n} N_i$ を求め, 次に, $\Pi \bmod \Pi^2$ からはじめて, 2 分木状に剰余を計算していって $R_i = \Pi \bmod N_i^2\ (1 \le i \le n)$ を求めている. 2 分木状にしたおかげで, 演算回数のオーダーを $n \log n$ に押さえることができる. 両 Tree の適用の様子を図 3 に示す.

$$\begin{aligned}\gcd(N_i, R_i/N_i) &= \gcd(N_i, (\prod_{j \ne i} N_i) \bmod N_i) \\ &= \gcd(N_i, \prod_{j \ne i} N_i)\end{aligned}$$

なので, N_i とそれ以外の数の共通因数が求められる. なお, 前半の Product Tree における各ノードの値は後半の Remainder Tree において用いるため一時的に格納しておく必要がある.

Lenstra らは Heninger らとは異なり, 総積の代わりに合成数の最小公倍数 LCM を 2 分木状に計算していく手法 (以下, LCM Tree と呼ぶ) を採用している. Remainder Tree に相当するステップの代わりに, 合成数 N_i と Π' との GCD を個々に計算することになる. なお, Π' に比べて合成数 N_i は非常に小さいため, 共通因数を抽出するのための GCD の処理はそれほど重くないことが期待できる.

上記の 2 つの手法と異なる第 3 の手法として, 共通因数の総積を求めるために, 上記の LCM Tree の計算過程において各ノードにおける GCD を記憶しておき, 次に, それらの GCD の GCD を 2 分木状に計算していく手法 (以下, GCD Tree と呼ぶ) もある. 両 Tree の様子を図 4 に示す.

Lenstra らの手法及び第 3 の手法における欠点は, 2 分木の root に近い部分において, GMP ライブラリをそのまま利用している限りは, CPU のコア数を有効に活用できず, 非常に大きな整数同士の GCD の計算時間を削減することが難しいことである. 実際, 上述の 2,783,867 個の 1,024 ビット合成数を例に取ると, ノードの数が多い部分は並列化を行って CPU のコア数に応じて処理時間を短縮できるが, 根に近い部分の計算時間は全体の 1/2 程度を占める.

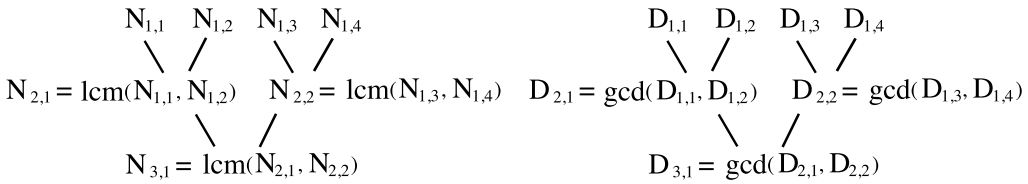

図 4: LCM Tree(左) と GCD Tree(右)　($n = 4$ の場合)

本調査時においては第 3 の手法を利用して, SSL Observatory が収集した公開鍵証明書 (2010 年) の中から, 1,024 ビットの RSA 公開鍵を格納している公開鍵証明書を 2,742,833 本を取り出して, 共通因数の有無の調査を行い, 8,703 個の脆弱な RSA 公開鍵を検出できた. 全体の処理は数時間で完了できる.

3.4　可視化について

本研究では, 脆弱な RSA 鍵の全世界での分布状況を把握するため可視化することとした. 図 2 の通り, 公開鍵証明書のフィールド中には, 格納されている公開鍵に対応する秘密鍵を所有しているユーザーを表すための subject というフィールドが存在する. このフィールドから脆弱な RSA 鍵を有しているユーザーに関する何らかの情報 (国情報, ホスト名など) を明らかにできる場合がある. しかしながら, 今回, 脆弱な RSA 鍵を検出した公開鍵証明書の subject フィールドの多くは国情報すら記されていないものであった. 従って, 各地域インターネットレジストリ (AfriNIC, ARIN, APNIC, LACNIC, RIPE NCC) が公開している IP アドレスの割り当てに関する情報を参照して SSL Observatory が収集していた IP アドレスから国情報を判別した. その様子を図 5, 6 に示す.

3.5　クローリングについて

SSL Observatory で公開されている公開鍵証明書は 2010 年に収集されたもので若干古いため, 3.1 節の「**RSA 公開鍵の解析**」で見つけた脆弱な公開鍵証明書は更新されている可能性がある. そこで本研究では, 最新の公開鍵証明書をクローラを作成し入手した.

クローラを作成するため, Python の ssl ライブラリに含まれる `wrap_socket` 関数を利用した. 今回の調査では, `wrap_socket` 関数のオプションとして, SSLv3 と TLSv1 を利用した. また, 応答しないホストが多数存在していたため, 応答までのタイムアウト時間を 5 秒に設定した.

4　解析結果

4.1　世界の傾向

図 1 の方針に従い解析を行った結果, SSL Observatory が収集した公開鍵証明書 (2010 年) のうち, 素因数分解が可能な RSA 公開鍵の総数は, 8,703 個であった. 今回のクローリングで, それらの鍵を有していた 8,703 台のホストに接続を試みたが, 約 62%に相当する 5,443 台からは証明書自体を得る事ができなかった (表 3).

表 3: クローリングによる公開鍵証明書の入手状況 (世界)

状況	ホスト数	割合
成功	3,260	38%
失敗	5,443	62%

入手できなかった原因の内訳は, 表 4 の通りである. ほとんどのホストが Timed out, Connection refused, Network is unreachable であった. すなわち, 何らかの対策が施されたか, あるいは当該ホストの電源が切れている状態などであると考えられる.

8,703 本の公開鍵証明書中, 残りの 3,260 本については入手することができた. その中で素因数分解可能な RSA 公開鍵を未だに利用してい

図 5: 日本と他国との間の共通素因子の共有状況

図 6: 米国と他国との間の共通素因子の共有状況

表 4: 入手できなかった原因の内訳 (世界)

原因	ホスト数
Timed out	4,635
Connection refused	333
No route to host	400
Network is unreachable	10
その他	65

るホスト数は 2,611 台であった. 従って, 少なくとも約 30%に相当する 2,611 個の脆弱な公開鍵が現時点でも利用されていることになる.

4.2 日本の傾向

SSL Observatory が収集した公開鍵証明のうち, 8,703 本の公開鍵証明書が素因数分解できる RSA 公開鍵を含んでいた. その中で, JP ドメインで利用されているものは 171 本であった.

171 本の公開鍵証明書の中で, 今回のクローリングで公開鍵証明書自体を入手できなかったのは, 約 39%に相当する 67 本であった (表 5).

表 5: クローリングによる公開鍵証明書の入手状況 (日本)

状況	ホスト数	割合
成功	104	61%
失敗	67	39%

また, 入手できなかった原因の内訳を表 6 に示した.

表 6: 入手できなかった原因の内訳 (日本)

原因	ホスト数
Timed out	65
Connection refused	2
No route to host	0
Network is unreachable	0
その他	0

SSL Observatory が収集した証明書で JP ドメインに含まれる素因数分解可能な公開鍵証明書 171 本中, JP ドメインで現在も利用されているものは, 90 本であった (表 7). (日本が全体の平均以上になっているのは, クローリングした

表 7: 世界と日本の傾向

	SSL Observatory における素因数分解可能な RSA 鍵数 (2010 年)	残された素因数分解可能な RSA 鍵数 (2013 年)	割合
世界	8,703	2,611	30%
日本	171	90	52%

表 8: トップページの内容についての内訳

内容	ホスト数	割合
ログイン	1,887	84.5%
アクセス拒否	277	12.4%
プリンター管理ツール	28	1.3%
他のサイトへのリンク	17	0.8%
セキュリティツール	13	0.6%
リモート電源管理ツール	11	0.5%

時間帯と時差が関係している可能性もある.)

4.3 脆弱な公開鍵を保持しているホストの正体

どういったホストが素因数分解可能な公開鍵を保持しているかを調査するため, 2,611 台のホストに接続し, そのトップページを調べた. 本調査では, 2,611 台のホスト中, 2,233 台のホストのトップページを入手することができた. なお, 利用したツールは Linux 標準付属の curl である.

入手した HTML ファイルの内訳を表 8 に示す. 1,887 件と大多数となっている「ログイン」に関しては, 単純なログイン画面が表示されるのみであり, そのほとんどが同じ画面となっている. その原因は, 同じ HTML ファイルが利用されていることにあり, 1,887 件中 1,567 件が, また残りの 320 件に関しても 264 件が同じものを利用している.

5 おわりに

本稿では, 公開鍵認証基盤 (PKI) 等において, 最も利用されている公開鍵暗号 RSA に新たな脅威が指摘されたことを受け, この問題に関する再調査を行い, 世界や日本における傾向やこの問題への対応状況を調べた. 今回の我々の調査では, 日本での対応状況が世界に比べて遅れている, すなわち, 素因数分解可能な RSA 公開鍵がそのまま放置されているケースが多く発見された. 我々は今後も定期的に調査を行い, 注意喚起を行っていくとともに, 本問題の解決に向けた取り組みを行っていきたいと考えている.

参考文献

[1] N. AlFardan, D.J. Bernstein, K.G. Paterson, J. Schuldt, On the Security of RC4 in TLS, USENIX Security 2013, 2013.

[2] D.J. Bernstein, Fast multiplication and its applications, Algorithmic Number Theory, MSRI Publications, Volume 44, 2008.

[3] D. Brumley, D. Boneh, Remote timing attacks are practical, Computer Networks 48(5), pp.701–716, 2005.

[4] Z. Durumeric, E. Wustrow, J.A. Halderman, ZMap: Fast Internet-wide Scanning and Its Security Applications, USENIX Security 2013, 2013.

[5] Fast pairwise GCD computation: fastgcd-1.0.tar.gz Available from `https://factorable.net/resources.html` (2013-08-26)

[6] D. Goodin, Hackers break SSL encryption used by millions of sites. Available from `http://www.theregister.co.uk/2011/09/19/beast_exploits_paypal_ssl/` (2013-08-26)

[7] D. Goodin, Crack in Internet's foundation of trust allows HTTPS session hijacking. `http://arstechnica.com/security/2012/09/crime-hijacks-https-sessions/` (2013-08-26)

[8] D. Goodin, Gone in 30 seconds: New attack plucks secrets from HTTPS-protected pages. Available from `http://arstechnica.com/security/2013/08/gone-in-30-seconds-new-attack-plucks-secrets-from-https-protected-pages/` (2013-08-26)

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

[10] T. Isobe, T. Ohigashi, Y. Watanabe, Masakatu Morii, Full Plaintext Recovery Attack on Broadcast RC4, Pre-proceeding of FSE 2013, 2013.

[11] A.K. Lenstra, J.P. Hughes, M. Augier, J.W. Bos, T. Kleinjung, C. Wachter, Public Keys, CRYPTO 2012, LNCS 7417, pp.626–642, 2012.

[12] K. Mowery, M. Yung, C. Wei, D. Kohlbrenner, H. Shacham, S. Swanson, Welcome to the Entropics: Boot-Time Entropy in Embedded Devices, IEEE Symposium on Security and Privacy 2013, pp.589–603, 2013.

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