PKI Day 2018 事前資料(≠発表資料), Tokyo, Japan

before ROCA / after ROCA / beyond ROCA から見えてくるRSA暗号実装の闇

須賀祐治

2018-04-18

Ongoing Innovation

Introduction

RSA公開鍵暗号

まだまだ圧倒的に利用されている

一方でTLS1.3では排除（鍵交換・署名として）

- 鍵生成
  - どでかい素数 p, q を2つ選択
  - N:= pq, e:= 65537(例えば)　←公開鍵
    - Nを素因数分解することは困難
    - eは$\phi(n)=(p-1)(q-1)$と素
  - $d:= e^{-1} \bmod (p-1)(q-1)$　←秘密鍵
    - p か q が分かると d も導出可能
- 暗号化　$c = m^e \bmod N$
- 復号　$c^d \bmod N = m$
  - 逆に使えば「デジタル署名」にもなる

サーバ証明書もRSAは
大きなシェアを持ってはいるけど

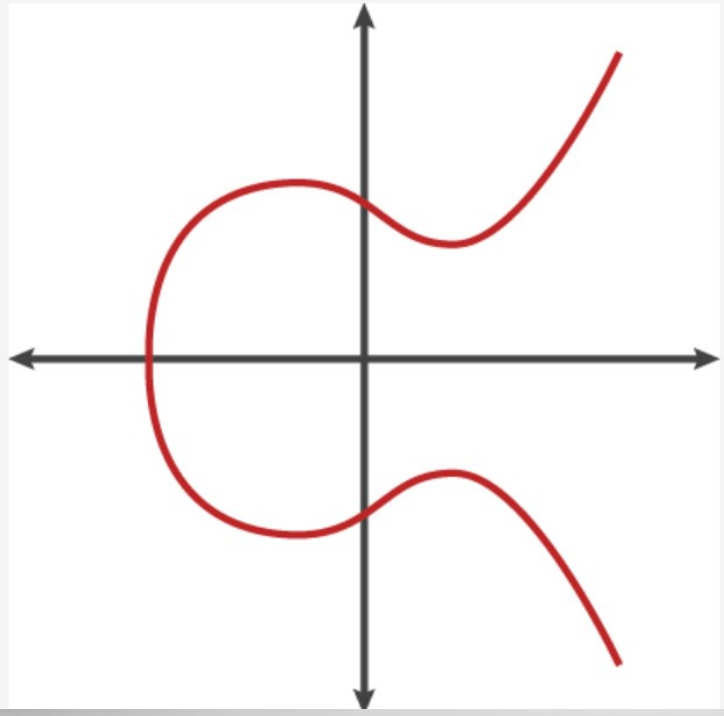
CLOUDFLARE

Blog home | What we do | Support | Community | Login

ECDSA: The digital signature algorithm of a better internet

10 Mar 2014 by Nick Sullivan.

G+ Share 535 Like 1 Tweet



Search the blog

Cloudflare blog

Contact our team

US callers
1 (888) 99-FLARE
UK callers
+44 (0)20 3514 6970
International callers
+1 (650) 319-8930

Full feature list and plan types

Cloudflare provides performance ar
for any website. More than 6 millior
use Cloudflare.

There is no hardware or software. C
works at the DNS level. It takes only
to sign up. To learn more, please vis
website

Cloudflare features

Overview

秘密鍵が漏洩したときの影響

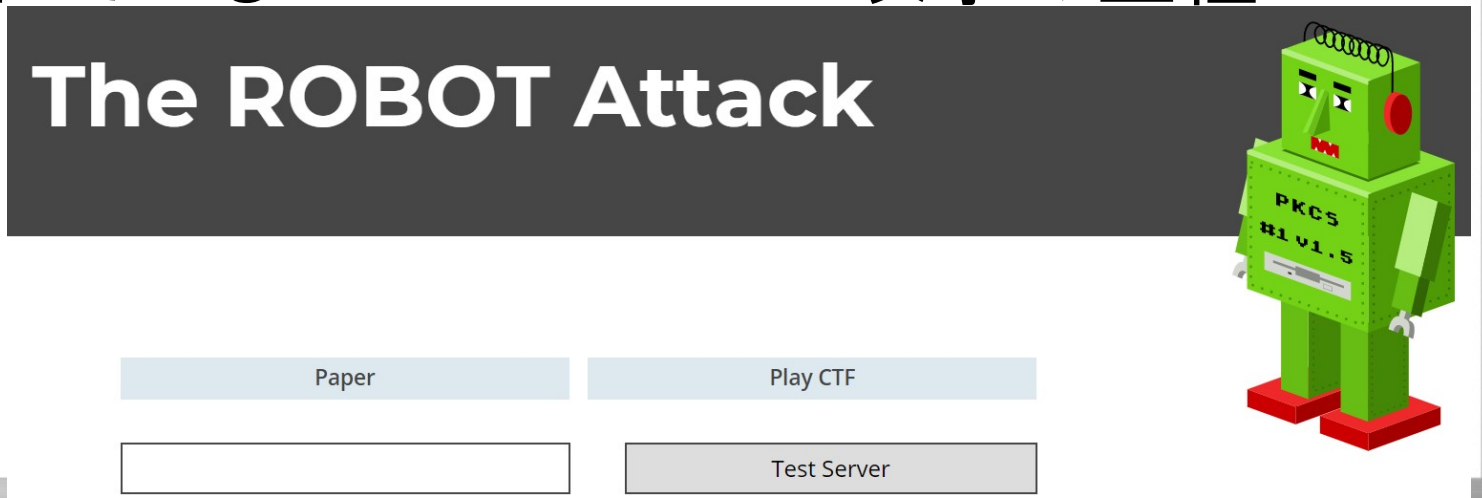
影響の分類	詳細
(1)暗号化された通信の暴露	SSL/TLSやSSHサーバとクライアントの通信を傍受できる環境において，暗号化を行っているにも関わらず通信内容を把握することができる
(2)サーバのなりすまし	DNS詐称やネットワークの乗っ取りが可能な環境において，SSL/TLSやSSHサーバになりすますことができる
(3)不正ログインもしくはクライアントのなりすまし	SSL/TLSでのクライアント認証のように公開鍵証明書を用いたログインが可能になる
(4)不正プログラムへのコード署名	コードサイニング用の公開鍵証明書の秘密鍵が漏洩することで，当該証明書によって保証されたプログラムであるかのように見せることができる

大前提：秘密鍵は漏洩しないという仮定

脱線：これと混同しないように

Return Of Bleichenbacher's Oracle Threat (ROBOT)

- The ROBOT Attack https://robotattack.org
- Padding Oracle攻撃
  - BEAST攻撃とかPOODLE攻撃とかと同様に
    何度もトライ＆エラーを繰り返し
    過去のメッセージを復元するという類の攻撃
  - よく知られているBleichenbacher攻撃の亜種



脱線：これと混同しないように

PKCS#1 v1.5 暗号化の問題

- TLSやS/MIME，JOSE(JSON暗号化), XML暗号化などで利用される暗号化形式で潜在的な脆弱性を抱えてた
- 回避する実装方法がTLS仕様にも書かれているがそれが複雑で分かりにくいから対応しきれなかった
- 古い実装（例えばJSSE: CVE-2012-5081）も露呈
- RSA PKCS #1 v1.5 暗号化はCRYPTRECでは「互換性維持のために継続利用を容認」
  - http://www.cryptrec.go.jp/list/cryptrec-ls-0001-2016.pdf
  - 1軍でも2軍でもなく戦力外通告を出されている方式

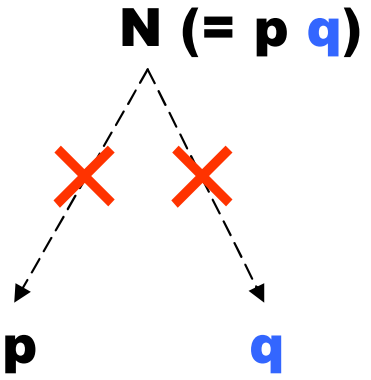
before ROCA

Debian OpenSSL（2008）

- Debianの特定バージョンにおけるOpenSSLを使って鍵生成を行った場合，極端に少ない鍵空間からしか秘密鍵を導出していないという問題.
- 2008年にアドバイザリが発行されていたにも関わらず，現在でもその脆弱な鍵を利用しているサイトが存在している.

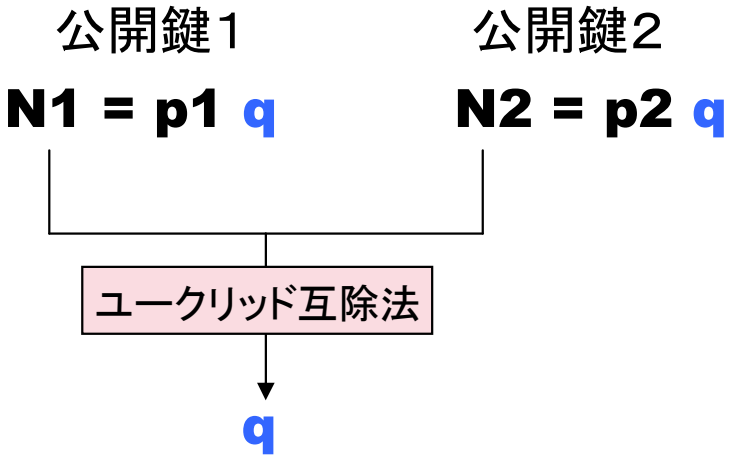
http://jvn.jp/vu/JVNVU925211/

ある意味RSAの致命的な問題（Ron was Wrong 2012）

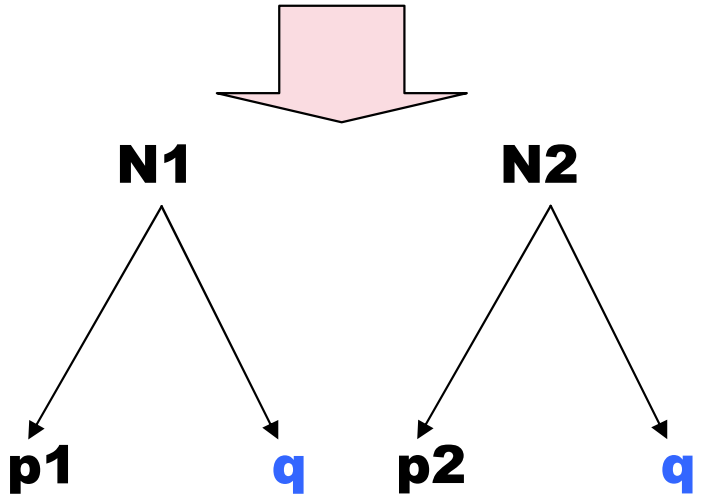


素因数分解は困難

（RSAはこの困難性に
安全の根拠を置いている）



因数を見つけるのは容易

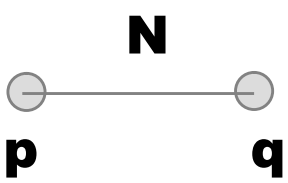


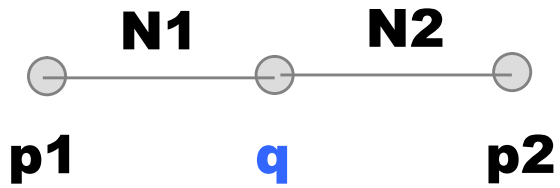
2つの公開鍵が素因数分解できてしまう

https://eprint.iacr.org/2012/064.pdf

この状況をグラフ表現すると…

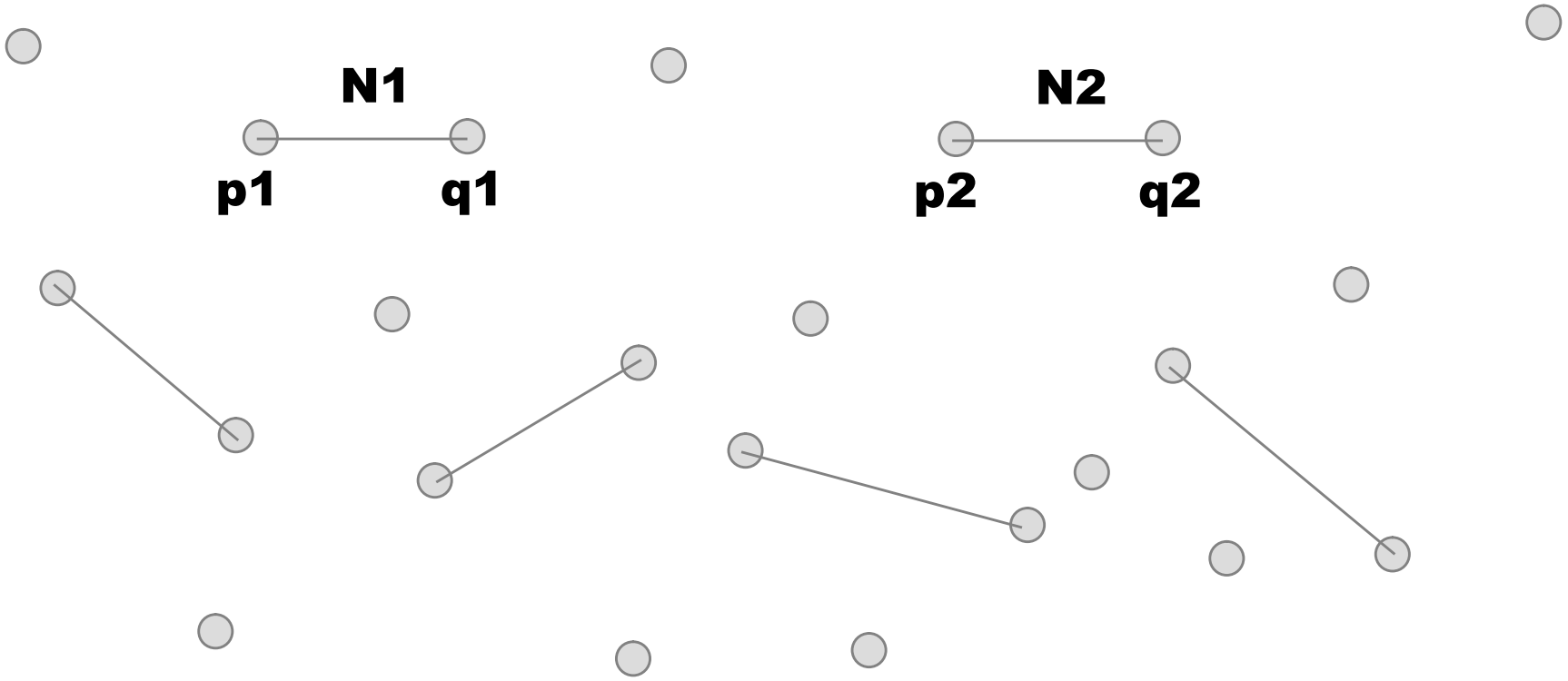
- グラフ表現
  - 頂点：素数
  - 辺：2素数を結ぶ辺の存在＝公開鍵の存在





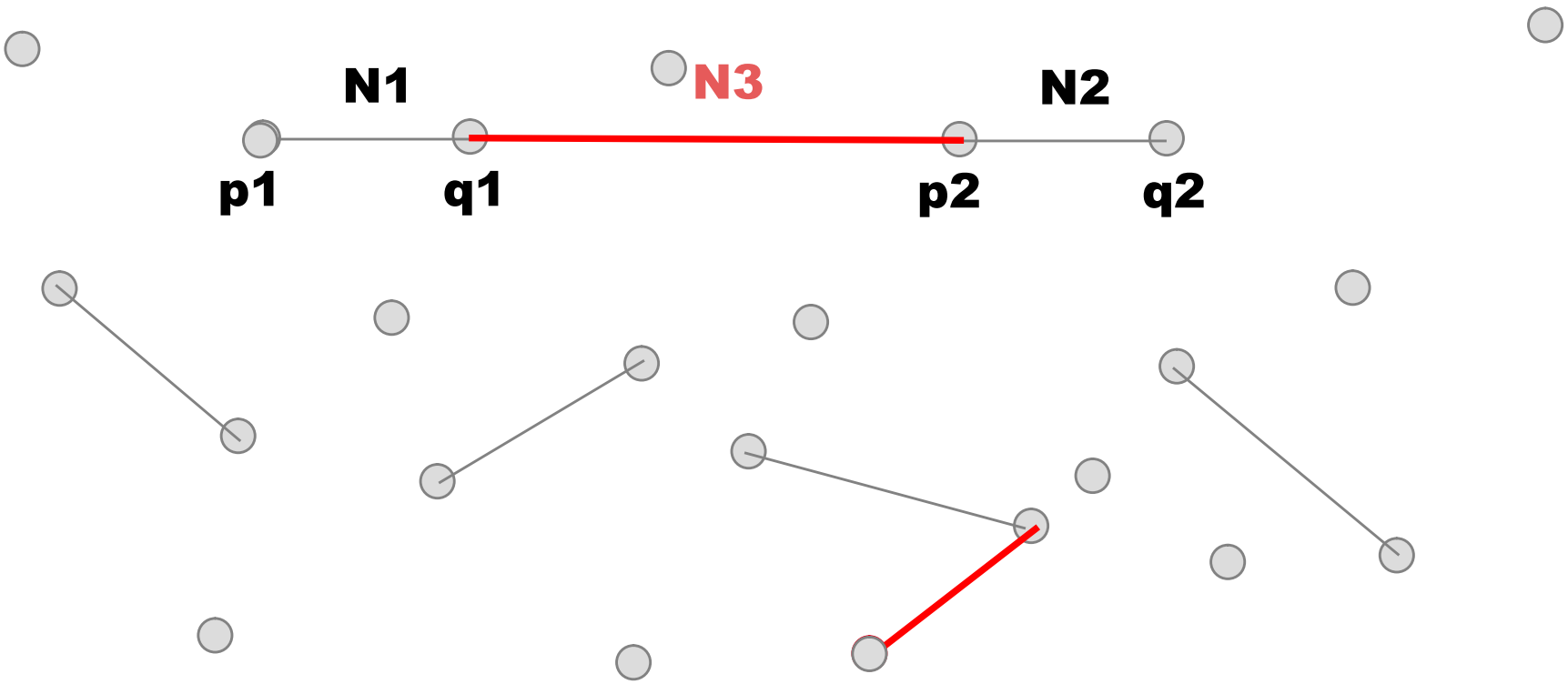
あるべき姿

- 2素数を選んで辺を結んだら（Nを計算したら）ほかのグラフと頂点を共有していない



しかし急にこういうことに…

- ほかのグラフと頂点を共有していなかったのに勝手に結ばれてしまうこともありうる

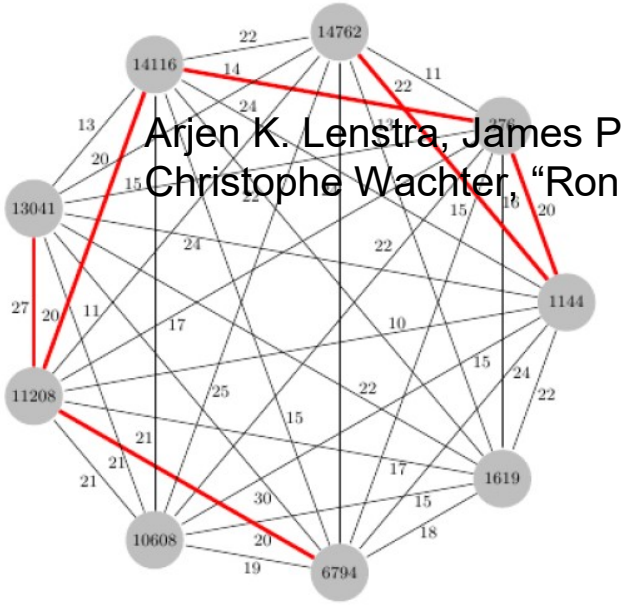


選んだ素数が偶然重なる可能性

- ランダムに選択したつもりの素数が
  ほかと偶然重なる可能性が0ではないことは
  RSAアルゴリズムの根本的な問題
- 素数定理
  - 素数がどのくらい存在するか知る指標
- 2048ビットRSAで用いられる1024ビット素数の候補は素数定理から約$2^{1014.53}$個も存在
  - 容易に重複するものではない
- しかし素数生成時のアルゴリズムに偏りがある
  ＝$2^{1014.53}$個の全空間から素数を抽出していない場合にこの問題が起き得る

PRNGの問題

- 素数生成時に用いられる擬似乱数生成モジュールがまずいと問題が起きる
  - 9つの素数しか生成しないデバイスの存在
  - 36（$=_9C_2$）個の公開鍵は異なる9個の秘密鍵の組み合わせで構成



Arjen K. Lenstra, James P. Hughes, Maxime Augier, Joppe W. Bos, Thorsten Kleinjung, Christophe Wachter, “Ron was wrong, Whit is right”, http://eprint.iacr.org/2012/064

A shorter version of this paper will appear in *Proc. 21st USENIX Security Symposium*, Aug. 2012. Rev. 2; July 11, 2012.
For the newest revision of this paper, partial source code, and our online key-check service, visit **https://factorable.net**.

Mining Your Ps and Qs: Detection of Widespread Weak Keys in Network Devices

Nadia Heninger[†*] Zakir Durumeric[‡*] Eric Wustrow[‡] J. Alex Halderman[‡]
[†] *University of California, San Diego* nadiah@cs.ucsd.edu
[‡] *The University of Michigan* {zakir, ewust, jhalderm}@umich.edu

https://factorable.net/weakkeys12.extended.pdf

PKI Day 2012での私案

http://www.jnsa.org/seminar/pki-day/2012/data/PM02_suga.pdf

Internet Initiative Japan Inc.

対策

- 十分なエントロピーを確保できる環境かどうか確認
  - とはいうものの… それを実現するのは難しい
- 鍵データベースを構築して，これまでに利用された「使い回し鍵」を使っていないか確認できる仕組み
  - factorable.net と同様のサービス
  - 事例：Debian OpenSSL鍵生成問題時に配布されたDB
- 鍵長を特殊なものにする
  - 例えば N が 2012 ビットになるような合成数など
    - この鍵を受け入れないデバイスもあるかもしれないので注意

Rump session at CRYPTO2013

- Heningerらが報告
  - “Factoring RSA keys from certified smart cards: Coppersmith in the wild”
  - “Factoring as a service”

Taiwan Citizen Digital Certs

- 300万枚の証明書を国営LDAPサーバから入手
  - うち230万枚は 1024-bit RSA, 70万枚は2048-bit

Taiwan Citizen Digital Certificate

Government-issued smart cards allow citizens to

- file income taxes,
- update car registrations,
- transact with government agencies,
- interact with companies (e.g. Chunghwa Telecom) online.

FIPS-140 and Common Criteria Level 4+ certified.

素数の特徴がひどい...

Most commonly shared factor appears 46 times

c0000000000000000000000000000000
0000000000000000000000000000000000
0000000000000000000000000000000000
00000000000000000000000000002f9

Next most common factor appears 7 times

c92424922492924992492449242492
24929249924949244924249224929249
92494924492424922492924992494924
49242492249292499249249244924e5

Some more prime factors

```
c0000000000000000000000000000000
0000000000000000000000000000000000
0000000000000000000000000000000000
00000000000000000000000000101ff

c0000000000000000000000000000000
0000000000000000000000000000000000
0000000000000000000000000000000000
00000000000000000000100000177

ffffaa55fffffffff3cd9fe3ffff676
fffffffffffe0000000000000000000
0000000000000000000000000000000000
00000000000000000000000000009d

c000b8000000000000000000000000
0000000000000000000000000000000000
0000068000000000000000000000000
00000000000000000000000000000251
```

ここまでRSAをディスってきたけど

- 間違えてはいけないのは
  暗号アルゴリズムそのものとしてのRSAは
  まだ大丈夫

- 死亡：DES/MD4/MD5
- 環境によっては利用可：RC4/SHA-1

技術的要因以外の問題

- 問題の本質は技術的なことだけではない
  - コスト負担の問題
    - お互いに押し付けあわないで
      それぞれの立ち位置でカバーしあうことが大切
  - まだまだ啓蒙活動が必要なのかもしれない
    - 一例）暗号学者と実装者との理解の乖離

“ツール”の使用者：使い方が　まずい
“ツール”の実装者：実装が　まずい
“ツール”の設計者：設計・仕様が　まずい

- PRNG
- 暗号
- CBC

Predictions (December 2012)

- ほかのアルゴリズム・プロトコルへの横展開
  - 楕円暗号などほかのプリミティブでも起こる
  - POP/SMTP/FTP over SSL etc…
    - こっちの方が検証系はやばそうな気も…
- Low entropy を利用した攻撃
  - DSA署名生成時のように一時鍵・セッション鍵の鍵空間の狭さを利用した脆弱性の発見
    - 公開鍵系に依存しないはず（ハードコーディング系）
  - CAや主要サイトの鍵生成時の環境を特定し鍵生成アルゴリズムをぶん回して鍵を特定

DSAでも同じことが起こる

- 秘密鍵生成時ではなく署名生成時に発生

- 共通パラメータ (p, q, g, y)
- 秘密鍵 x, 公開鍵 $y = g^x \bmod p$
- 署名生成
  - 一時鍵(ephemeral key) k をランダムに選択
  - $r := (g^k \bmod p) \bmod q$
  - $s := k^{-1}(H(m)+xr) \bmod q$ (r,s):署名

DSAにおけるk選択時の問題

- もし k がばれると署名(r,s)から秘密鍵xが暴露
  - $x= r^{-1}(ks-H(m)) \bmod q$
    - cf) $s:= k^{-1}(H(m)+xr) \bmod q$ を式変形したら上が得られる

- 同じk を用いて2つの署名を生成すると…

  $r:= (g^k \bmod p) \bmod q$ ← これが共通

  $s1:= k^{-1}(H(m1)+xr) \bmod q$

  $s2:= k^{-1}(H(m2)+xr) \bmod q$

  → $k= (H(m1)-H(m2))(s1-s2)^{-1} \bmod q$

富樫風で申し訳ないが

ECDSAでも同じ問題がある

2. (署名者 Alice による署名生成)
   (a) 乱数 r を生成し, スカラー倍 $U = r \times P = (x_U, y_U)$ を計算する.
   (b) メッセージ m のハッシュ値 $H(m)$ を計算する.
   (c) $u = x_U \bmod \ell$, $v = (H(m) + u \times d_A)/r \bmod \ell$ を計算する.
   (d) Bob に署名 (u, v) を送る.
3. (検証者 Bob による署名検証)
   (a) Alice の公開鍵 P_A をもとに, $d = 1/v \bmod \ell$ と点 $V = d \times H(m) \times P + d \times u \times P_A = (x_V, y_V)$ を計算する.
   (b) $u = x_V \bmod \ell$ ならば署名を受理する.

DSA) 乱数 k, $r \equiv (g^k \bmod p) \bmod q$ $s \equiv (H(m) + xr)/k$

ECDSA) r $rP = (x_u, y_u)$ $v \equiv (H(m) + rx_u)/r$

$\downarrow \bmod \ell$

u

DSA) k がばれると s から $x = (ks - H(m))/r$ が出てきる

ECDSA) r がばれると v から $x = (vr - H(m))/u$ が出てきる

$v_1 = (H(m_1) + xu)/r$

$v_2 = (H(m_2) + xu)/r$ $\Rightarrow$

$H(m_1) = v_1 r - xu$

$-)\ H(m_2) = v_2 r - xu$

$H(m_1) - H(m_2) = (v_1 - v_2) r$

$\rightarrow r$ がばれる

Bitcoin developers say critical Android flaw leaves digital wallets vulnerable to theft

By Ellis Hamburger on August 11, 2013 04:06 pm Email @hamburger

http://www.theverge.com/2013/8/11/4611770/bitcoin-critical-android-flaw-digital-wallets

Android Security Vulnerability

http://bitcoin.org/en/alert/2013-08-11-android

11 August 2013

What happened

We recently learned that a component of Android responsible for generating secure random numbers contains critical weaknesses, that render all Android wallets generated to date vulnerable to theft. Because the problem lies with Android itself, this problem will affect you if you have a wallet generated by any Android app. An incomplete list would be Bitcoin Wallet, blockchain.info wallet, BitcoinSpinner and Mycelium Wallet. Apps where you don't control the private keys at all are not affected. For example, exchange frontends like the Coinbase or Mt Gox apps are not impacted by this issue because the private keys are not generated on your Android phone.

2013年12月

Asiacrypt 2013 rump session

The Asiacrypt 201 erved as conch player.

Slides are now ava

			Slides
	Asiacrypt 2		
	Session 1		
19:30	Daniel J. B		slides
19:32	Dominique	nown Recipient Encryption	slides
19:36	Jian Guo, Y	HMAC and NMAC with	slides
19:39	Shiho Mori		
19:41	Javad Aliza Somitra Sa	alysis	slides
19:44	Debrup Cha		slides
19:50	Sanjay Bha		slides
19:57	Aggelos Ki		slides
19:59	Rosario Ge	and Efficient Hash	
20:06	Jeremiah B		slides
20:13	Aarhus Cry	crypt 2014)	
20:15	Break		

Elliptic Curve Cryptography in Practice

Joppe W. Bos, J. Alex Halderman,
Nadia Heninger, Jonathan Moore,
Michael Naehrig, and Eric Wustrow

実際に盗難にあっている

Repeated ECDSA Signature Nonces

158 bitcoin addresses repeated signature nonces.

Address `1HKywxiL4JziqXrzLKhmB6a74ma6kxbSDj` has stolen 59 **BTC** ≈ 3.6 **million rupees** from these addresses.

3 of these repeats due to Android Java RNG vulnerability.

ROCA

The Return of Coppersmith's Attack: Practical Factorization of Widely Used RSA Moduli*

Matus Nemec
Masaryk University,
Ca' Foscari University of Venice
mnemec@mail.muni.cz

Marek Sys†
Masaryk University
syso@fi.muni.cz

Petr Svenda
Masaryk University
svenda@fi.muni.cz

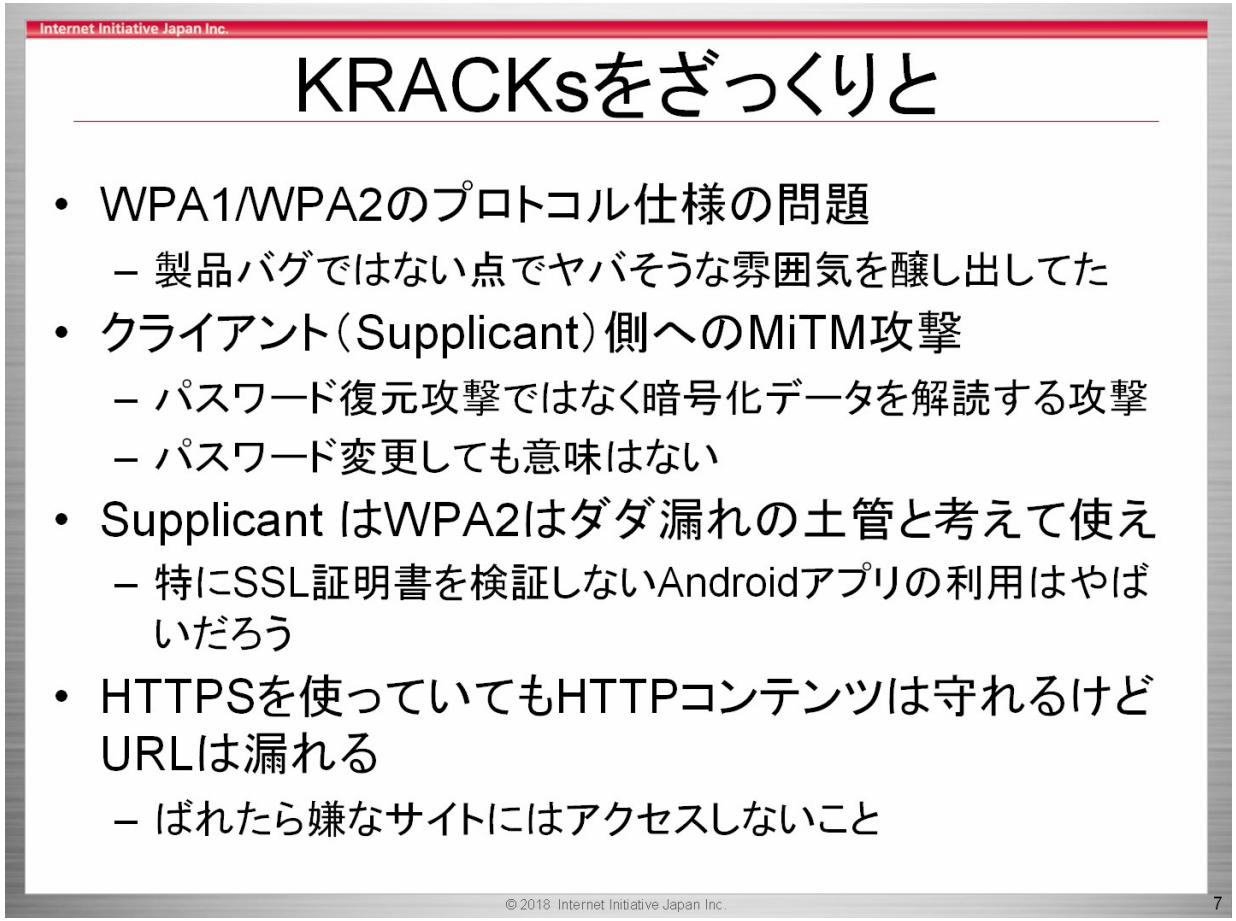
Dusan Klinec
EnigmaBridge, Masaryk University
dusan@enigmabridge.com

Vashek Matyas
Masaryk University
matyas@fi.muni.cz

https://crocs.fi.muni.cz/public/papers/rsa_ccs17

The Return of Coppersmith's Attack

- ROCA: Vulnerable RSA generation (CVE-2017-15361)
- KRACKsと同じくACMCCS2017で発表



ROCA

- RSA鍵生成モジュール（Infineon Technologies AG製）の問題を指摘
- 750,000のエストニアIDカードに影響
  - 実は1ヶ月前の9月に既報だった

http://estonianworld.com/technology/possible-security-risk-affects-750000-estonian-id-cards/

影響度（論文で指摘されているもののみ）

Domain name	Analyzed datasets	# Vuln. keys/devices	% Vulnerable
Complete/larger-scale datasets			
Certification authorities	all browser-trusted roots (173), level ⩽ 3 intermediates (1,869)	0 keys	0
ePass signing certificates	ICAO Document Signing Certificates, CSCA Master Lists	0 keys	0
Estonian eID	sample of 130,152 randomly selected citizens	71,417 keys	54.87
Estonian mobile eID	sample of 30,471 randomly selected citizens	0 keys	0
Estonian e-residents	sample of 4,414 e-residents	4,414 keys	100
Message security (PGP)	complete PGP key server dump (9 M)	2,892 keys	0.03
Software signing (GitHub)	SSH keys for GitHub developers (4.7 M)	447 keys	0.01
Software signing (Maven)	signing keys for all public Maven artifacts	5 keys	0.003
TLS/HTTPS	complete IPv4 scan, Certificate Transparency	15 keys	<0.001
Trusted boot (TPM)	41 laptops with different chips by 6 TPM manufacturers	10 devices	24.39
Limited, custom-collected datasets			
Payment cards (EMV)	13 cards from 4 EU countries, 6 with *Manufacturer* chip	0 keys	0
Programmable smartcard	25 cards from JCAlgTest.org database, 6 with *Manufacturer* chip	2 cards	8.67
Software signing (Android)	1,080 top ranking applications and games	0 keys	0

Table 4: The summary of the number and fraction of vulnerable keys detected in different domains. The domains are ordered lexicographically and separated into two groups based on the representativeness of inspected datasets.

- もっと精査すると，他にも出そうな感じ

※TPM (Trusted Platform Module) : 基本的なセキュリティ関連の機能 (主に暗号化キー) を提供するように設計されたマイクロチップ

https://crocs.fi.muni.cz/_media/public/papers/nemec_roca_ccs17_preprint.pdf

日本のベンダーでも影響あり

著者によるツールの公開

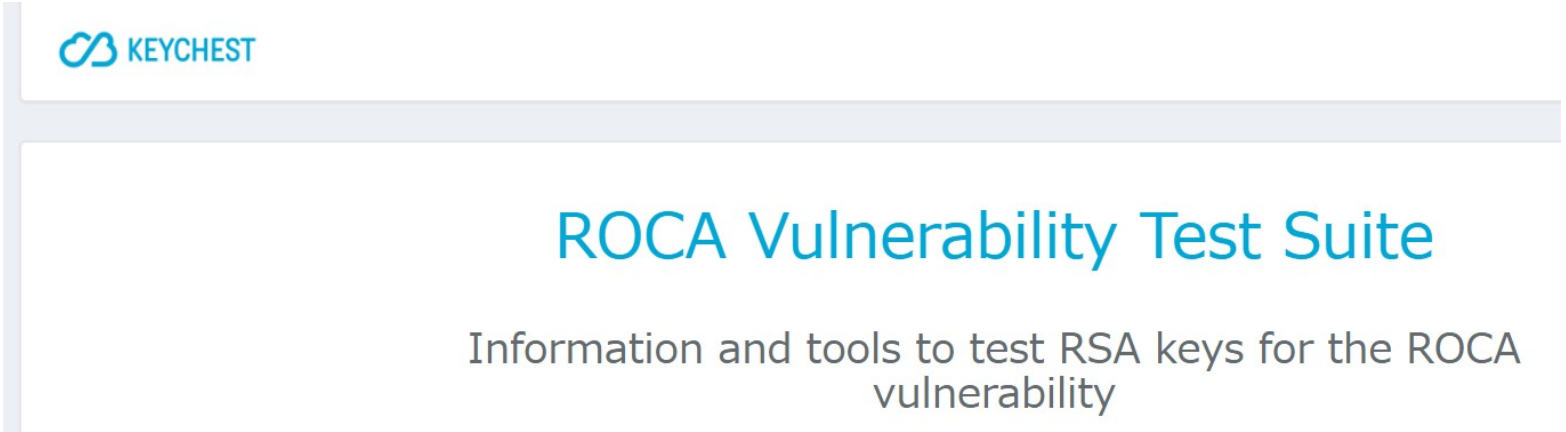
https://github.com/crocs-muni/roca

ROCA detection tool

build passing

This tool is related to ACM CCS 2017 conference paper #124 Return of the Coppersmith's Attack: Practical Factorization of Widely Used RSA Moduli.

- RSAlib（今回問題となったモジュール）で生成されたものかどうかをチェック可能
- オンライン版やS/MIME版（roca@keychest.net）もある



https://keychest.net/roca

Fingerprinting

- 驚くほどシンプル（38個の素数でチェック）

```
self.primes = [3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, 101,
               103, 107, 109, 113, 127, 131, 137, 139, 149, 151, 157, 163, 167]

self.prints = [6, 30, 126, 1026, 5658, 107286, 199410, 8388606, 536870910, 2147483646, 67109890, 2199023255550,
               8796093022206, 140737488355326, 5310023542746834, 576460752303423486, 1455791217086302986,
               14757395258967641292, 20052041432995567486, 60413881392493789203302, 207530445072488465666,
```

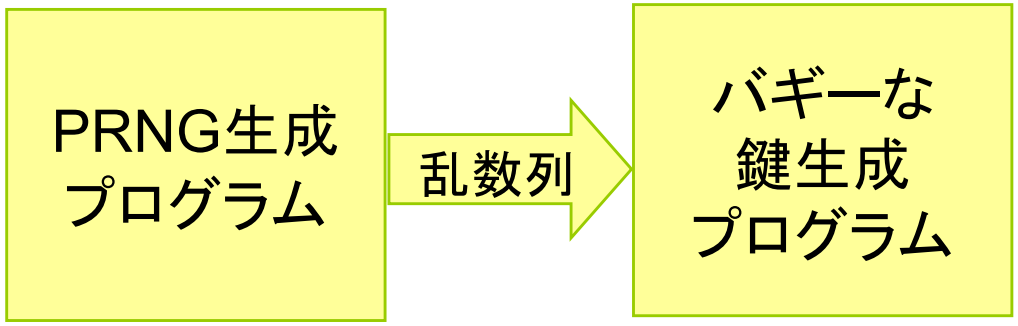
```
    self.tested += 1
    for i in range(0, len(self.primes)):
        if (1 << (modulus % self.primes[i])) & self.prints[i] == 0:
            return False

    self.found += 1
    return True
```

https://github.com/crocs-muni/roca/blob/master/roca/detect.py

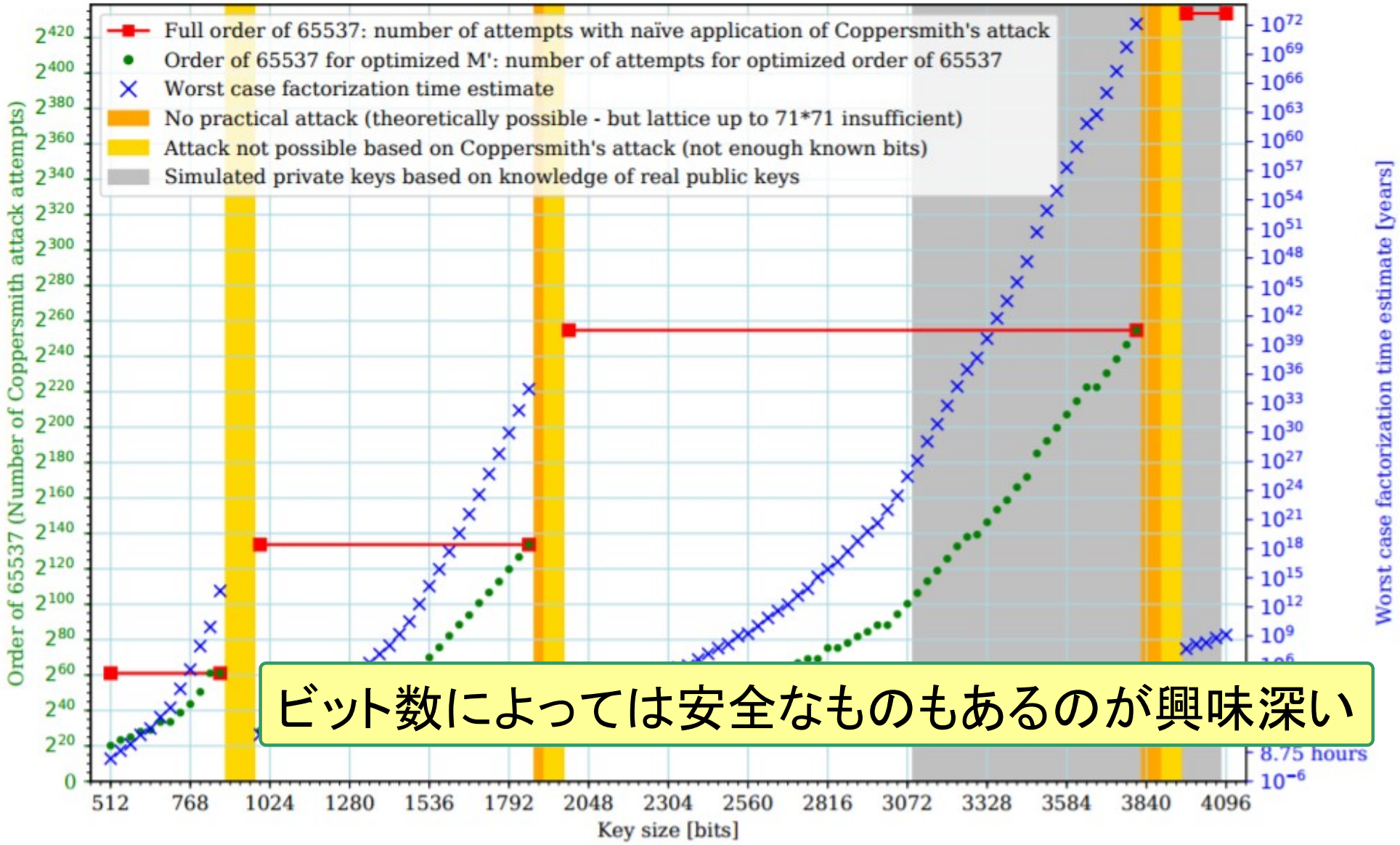
ROCAの本質

（ここは論文中の主張を尊重）

- PRNGの問題ではない
- 鍵生成アルゴリズムのバグである



もうひとつ面白い事実



ビット数によっては安全なものもあるのが興味深い

https://crocs.fi.muni.cz/_media/public/papers/nemec_roca_ccs17_preprint.pdf

再掲：PKI Day 2012での私案

http://www.jnsa.org/seminar/pki-day/2012/data/PM02_suga.pdf

Internet Initiative Japan Inc.

対策

- 十分なエントロピーを確保できる環境かどうか確認
  - とはいうものの… それを実現するのは難しい
- 鍵データベースを構築して，これまでに利用された「使い回し鍵」を使っていないか確認できる仕組み
  - factorable.net と同様のサービス
  - 事例：Debian OpenSSL鍵生成問題時に配布されたDB
- 鍵長を特殊なものにする
  - 例えば N が 2012 ビットになるような合成数など
    - この鍵を受け入れないデバイスもあるかもしれないので注意

beyond ROCA

PKI Day 2012での予言

http://www.jnsa.org/seminar/pki-day/2012/data/PM02_suga.pdf

Internet Initiative Japan Inc.

Predictions

- アルゴリズムの横展開
  - 楕円暗号などほかのプリミティブでも起こる
- Low entropy を利用した攻撃
  - DSA署名生成時のように一時鍵・セッション鍵の鍵空間の狭さを利用した脆弱性の発見
  - CAや主要サイトの鍵生成時の環境を特定し鍵生成アルゴリズムをぶん回して鍵を特定

公開鍵から「鍵生成ライブラリ」を推定

CROCS
Centre for Research on Cryptography and Security

- ACSAC2017で
  ROCAの次を発表

Measuring Popularity of Cryptographic Libraries in Internet-Wide Scans

Matus Nemec
Masaryk University,
Ca' Foscari University of Venice
mnemec@mail.muni.cz

Dusan Klinec
EnigmaBridge, Masaryk University
dusan@enigmabridge.com

Petr Svenda
Masaryk University
svenda@fi.muni.cz

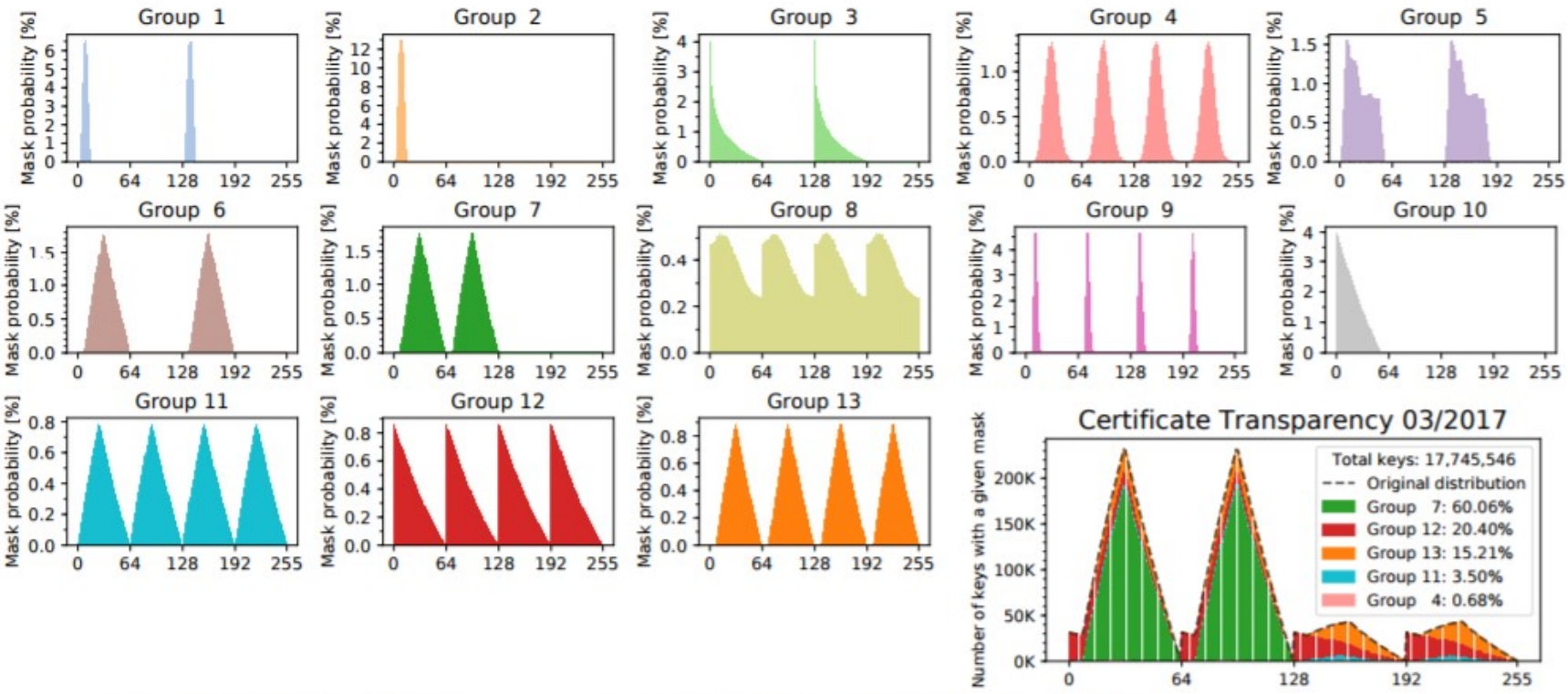
Peter Sekan
Masaryk University
peter.sekan@mail.muni.cz

Vashek Matyas
Masaryk University
matyas@fi.muni.cz

https://crocs.fi.muni.cz/public/papers/acsac2017

すごいいいいというか怖い

https://crocs.fi.muni.cz/_media/public/papers/acsac2017_nemec_rsa_fingerprints.pdf



Group 1: G&D SmartCafe 3.2
Group 2: G&D SmartCafe 4.x & 6.0
Group 3: GNU Crypto 2.0.1
Group 4: Gemalto GXP E64
Group 5: NXP J2A080 & J2A081 & J3A081 & JCOP 41 V2.2.1
Group 6: Oberthur Cosmo Dual 72K
Group 7: **OpenSSL** 0.9.7 & 1.0.2g & 1.0.2k & 1.1.0e
Group 8: PGPSDK 4 FIPS
Group 9: Infineon JTOP 80K, YubiKey 4 & 4 Nano
Group 10: NXP J2D081 & J2E145G, YubiKey NEO
Group 11: **BouncyCastle** 1.54 (Java), **Crypto**++ 5.6.0 & 5.6.3 & 5.6.5, **Libgcrypt** 1.7.6 FIPS, **Microsoft** CryptoAPI & CNG & .NET
Group 12: **BouncyCastle** 1.53 (Java), Cryptix JCE 20050328, FlexiProvider 1.7p7, HSM Utimaco Security Server Se50, **Nettle** 2.0, PolarSSL 0.10.0, PuTTY 0.67, SunRsaSign **OpenJDK** 1.8.0, **mbedTLS** 1.3.19 & 2.2.1 & 2.4.2
Group 13: **Botan** 1.5.6 & 1.11.29 & 2.1.0, Feitian JavaCOS A22 & A40, Gemalto GCX4 72K, HSM SafeNet Luna SA-1700, **LibTomCrypt** 1.17, **Libgcrypt** 1.6.0 & 1.6.5 & 1.7.6, **Libgcrypt** 1.6.0 FIPS & 1.6.5 FIPS, **Nettle** 3.2 & 3.3, Oberthur Cosmo 64, **OpenSSL** FIPS 2.0.12 & 2.0.14, PGPSDK 4, **WolfSSL** 2.0rc1 & 3.9.0 & 3.10.2, **cryptlib** 3.4.3 & 3.4.3.1

まとめ

参考

beyond

ROSCA

https://www.iij.ad.jp/dev/report/iir/031/01_04.html

RSAはどうなるん？

- 量子計算機で素因数分解
  - 1994年：Shorアルゴリズム
    - RSAは素因数分解の困難性，（EC）DHは離散対数問題の困難性が前提
    - 量子計算機を用いることにより，これら2つの問題が多項式時間で解けることが示された
  - 1996年：Groverアルゴリズム
- この2つの事実で暗号屋さんが動（働）いてる
  - もちろんお金も（例：EU Horizon 2020 Projects）

見積もり

- 現在のアーキテクチャの計算機でnビット安全性を有していた暗号アルゴリズムは、量子計算機の解読能力によるとn/2ビット安全性しか確保できない

Table 1 - Comparison of conventional and quantum security levels of some popular ciphers.

Algorithm	Key Length	Effective Key Strength / Security Level	
		Conventional Computing	Quantum Computing
RSA-1024	1024 bits	80 bits	0 bits
RSA-2048	2048 bits	112 bits	0 bits
ECC-256	256 bits	128 bits	0 bits
ECC-384	384 bits	256 bits	0 bits
AES-128	128 bits	128 bits	64 bits
AES-256	256 bits	256 bits	128 bits

Note : Effective key strength for conventional computing derived from NIST SP 800-57 "Recommendation for Key Management"

http://www.etsi.org/images/files/ETSIWhitePapers/QuantumSafeWhitepaper.pdf

耐量子（Post Quantum）暗号

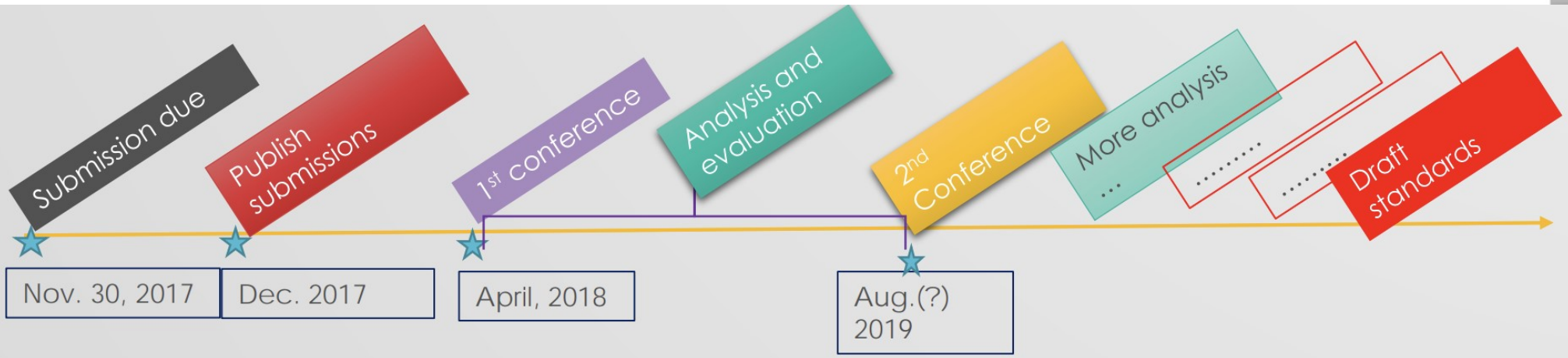
- 量子計算機が完成したらどうなるの？
  - 量子計算機ができても大丈夫なようにしよう

https://csrc.nist.gov/Projects/Post-Quantum-Cryptography

NIST PQC Standardization

- 2025年くらいを目処に標準化

Timeline	
Nov. 30, 2017	Submission deadline
April 2018	Workshop – Submitters' presentations
3-5 years	Analysis phase - NIST reports on findings and more workshops/conferences
2 years later	Draft standards available for public comments



https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/asiacrypt-2017-moody-pqc.pdf

NIST曰く

- コンペティションじゃない
- ポートフォリオの作成だけ
  - 理論ベースが違うものがたくさん並んでてよい

- 一方で軽量暗号（80ビット安全性）
  - 自動車? IoTデバイス向け？

これ本当に要るのか議論してなくない？

CRYPTRECにも調査WGあり

IIJ Internet Initiative Japan

Lead Initiative

日本のインターネットは1992年、IIJとともにはじまりました。以来、IIJグループはネットワーク社会の基盤をつくり、技術力でその発展を支えてきました。インターネットの未来を想い、新たなイノベーションに挑戦し続けていく。それは、つねに先駆者としてインターネットの可能性を切り拓いてきたIIJの、これからも変わることのない姿勢です。IIJの真ん中のIはイニシアティブ――IIJはいつもはじまりであり、未来です。

Ongoing Innovation

お問い合わせ先 IIJインフォメーションセンター
TEL：03-5205-4466（9:30～17:30 土/日/祝日除く）
info@iij.ad.jp
http://www.iij.ad.jp/