The Year in Crypto

Daniel J. Bernstein
University of Illinois at Chicago
Technische Universiteit Eindhoven

Nadia Heninger
University of Pennsylvania

Tanja Lange
Technische Universiteit Eindhoven

Candidate Indistinguishability Obfuscation
and Functional Encryption for all circuits

Sanjam Garg
UCLA
sanjamg@cs.ucla.edu

Craig Gentry
IBM Research
craigbgentry@gmail.com

Shai Halevi
IBM Research
shaih@alum.mit.edu

Mariana Raykova
IBM Research
mariana@cs.columbia.edu

Amit Sahai
UCLA
sahai@cs.ucla.edu

Brent Waters
University of Texas at Austin
bwaters@cs.utexas.edu

July 21, 2013

Abstract

In this work, we study *indistinguishability obfuscation* and *functional encryption* for general circuits:

Indistinguishability obfuscation requires that given any two equivalent circuits C_0 and C_1 of similar size, the obfuscations of C_0 and C_1 should be computationally indistinguishable.

In functional encryption, ciphertexts encrypt inputs x and keys are issued for circuits C. Using the key SK_C to decrypt a ciphertext $CT_x = \mathsf{Enc}(x)$, yields the value $C(x)$ but does not reveal anything else about x. Furthermore, no collusion of secret key holders should be able to learn anything more than the union of what they can each learn individually.

Understanding Cyberspace
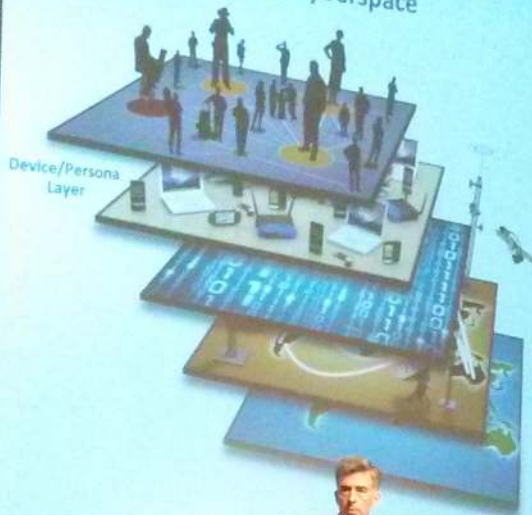
Understanding Cryptography

mathematical problems	factoring, discrete log, . . .
cryptographic primitives	RSA, Diffie-Hellman, DSA, AES, RC4, SHA-1, . . .
protocols	TLS, SSH, PGP, . . .
library implementations	OpenSSL, BSAFE, NaCl, . . .
software applications	Apache, Firefox, Chrome, . . .

The Cryptopocalypse

Math Advances Raise the Prospect of an Internet Security Crisis

Academic advances suggest that the encryption systems that secure online communications could be undermined in just a few years.

By Tom Simonite on August 2, 2013

The encryption systems used to secure online bank accounts and keep critical communications private could be undone in just a few years, security researchers warned at the [Black Hat conference](#) in Las Vegas yesterday. Breakthroughs in math research made in the past six months could underpin practical, fast ways to decode encrypted data that's considered unbreakable today.

A quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristic

Improvements over FFS in small to medium characteristic

Razvan Barbulescu, Pierrick Gaudry, Antoine Joux, Emmanuel Thomé

1 Introduction

The discrete logarithm problem (DLP) was first proposed as a hard problem in cryptography in the seminal article of Diffie and Hellman [DH76]. Since then, together with factorization, it has become one of the two major pillars of public key cryptography. As a consequence, the problem of computing discrete logarithms has attracted a lot of attention. From an exponential algorithm in 1976, the fastest DLP algorithms have been greatly improved during the past 35 years. A first major progress was the realization that the DLP in finite fields can be solved in subexponential time, i.e. $L(1/2)$ where $L_N(\alpha) = \exp\left(O((\log N)^\alpha (\log \log N)^{1-\alpha})\right)$. The next step further reduced this to a heuristic $L(1/3)$ running time in the full range of finite fields, from fixed characteristic finite fields to prime fields [Adl79, Cop84, Gor93, Adl94, JL06, JLSV06].

Recently, practical and theoretical progress have been made [Jou13a, GGMZ13, Jou13b] with an emphasis on small to medium characteristic finite fields and composite degree extensions. The most general and efficient algorithm [Jou13b] gives a complexity of $L(1/4 + o(1))$ when the characteristic is smaller than the square root of the extension degree. Among the ingredients of this approach, we find the use of a very

Fact: All the public-key crypto we use relies on three assumptions:

factoring integers into primes

discrete log modulo primes

discrete log in elliptic curve groups

```
airey:~ nadiah$ gpg --search-keys rivest@csail.mit.edu
gpg: searching for "rivest@csail.mit.edu" from hkp server keys.gnupg.net
(1)     Ronald L Rivest <rivest@csail.mit.edu>
          1024 bit DSA key 567B4BAD, created: 2010-12-19
(2)     Ronald L Rivest <rivest@csail.mit.edu>
          1024 bit DSA key 54BFA094, created: 2004-09-18
Keys 1-2 of 2 for "rivest@csail.mit.edu".  Enter number(s), N)ext, or Q)uit >
```

discrete log modulo primes

Discrete log over small characteristic fields

(Not actually used in any deployed crypto.)

- Factoring, discrete log have subexponential-time algorithms.
- No big algorithmic improvement since 1993.
- All progress has been Moore's law, implementation details, etc.

Discrete log over small characteristic fields

(Not actually used in any deployed crypto.)

- Factoring, discrete log have subexponential-time algorithms.
- No big algorithmic improvement since 1993.
- All progress has been Moore's law, implementation details, etc.

Until December 2012:

2012-12-24	1175-bit and 1425-bit	Joux
2013-02-11	$\mathbb{F}_{2^{1778}}^*$	Joux
2013-02-19	$\mathbb{F}_{2^{1971}}^*$	GGMZ
2013-02-20	$L(1/4 + o(1), c)$	Joux
2013-03-22	$\mathbb{F}_{2^{4080}}^*$	Joux
2013-04-11	$\mathbb{F}_{2^{6120}}^*$	GGMZ
2013-05-21	$\mathbb{F}_{2^{6168}}^*$	Joux
2013-06-18	$n^{O(\log n)}$ algorithm for $\mathbb{F}_{p^n}^*$	Barbulescu, Gaudry, Joux, Thomé

Extrapolated impact of hypothetical factoring algorithm improvements

Current general-purpose factoring running time for integer N:

$$L((64/9)^{1/3}, 1/3) = \exp\left((64/9)^{1/3}(\ln N)^{1/3} * (\ln \ln N)^{2/3}\right)$$

Small-characteristic field DL improvement from $L(1/3) \to L(1/4) \to n^{O(\log n)}$.

		bit length of N		
		1024	2048	4096
current state $\to$	$L((64/9)^{1/3}, 1/3)$	86	116	156
improved constant $\to$	$L((32/9)^{1/3}, 1/3)$	68	92	124
improved exponent $\to$	$L((64/9)^{1/4}, 1/4)$	49	63	81
		bit-security of key		

- Researchers in area agree that small-characteristic techniques can't be adapted to factoring or large primes

- Reminder that sometimes big progress can be made on old problems.

- There is *no proof* that factoring/discrete log are hard. (Polynomial heirarchy would collapse if they were NP-hard.)

- Elliptic curve discrete log totally different story: index calculus unlikely to work. (Already Miller 1986, Koblitz 2000.)

Some recommendations:

- Don't hard-code algorithms or key sizes.[*] If you must, use conservative choices.

- Listen to cryptographers. This is old news.

- Think about adopting elliptic curves. (More on this later.)

January 2013

A *user* actually tries to use crypto!

January 2013

A *user* actually tries to use crypto! . . . and fails.

January 2013

A *user* actually tries to use crypto! ... and fails. Close to #epicfail.

January 2013

A *user* actually tries to use crypto! ... and fails. Close to #epicfail.

"It's really annoying and complicated,
the encryption software.
... He kept harassing me,
but at some point he just got frustrated,
so he went to Laura."

—Glenn Greenwald,
quoted in "How Laura Poitras helped Snowden spill his secrets",
New York Times Magazine, 18 August 2013

February 2013: timing-padding-oracle attacks against TLS

> This leaves a small timing channel, since MAC performance depends to
> some extent on the size of the data fragment, but it is not believed
> to be large enough to be exploitable, due to the large block size of
> existing MACs and the small size of the timing signal.
>
> —RFC 5246, "The Transport Layer Security (TLS) Protocol, Version 1.2", 2008

February 2013: timing-padding-oracle attacks against TLS

```
This leaves a small timing channel, since MAC performance depends to
some extent on the size of the data fragment, but it is not believed
to be large enough to be exploitable, due to the large block size of
existing MACs and the small size of the timing signal.
```

> —RFC 5246, "The Transport Layer Security (TLS) Protocol, Version 1.2", 2008

This timing side-channel can then be "wrangled" into revealing plaintext data via careful statistical analysis of multiple tim-

—AlFardan and Paterson,
"Lucky Thirteen: breaking the TLS and DTLS record protocols",
IEEE Symposium on Security and Privacy 2013

February 2013: TLS algorithm agility to the rescue!

Typical vendor response:

February 2013: TLS algorithm agility to the rescue!

Typical vendor response:

> To mitigate this vulnerability, configure the client-side SSL
> profile to prefer RC4-SHA ciphers.

February 2013: TLS algorithm agility to the rescue!

Typical vendor response:

> To mitigate this vulnerability, configure the client-side SSL
> profile to prefer RC4-SHA ciphers.

Successful upgrade: RC4 was used for >50% of TLS traffic in February 2013.

March 2013: attacks against RC4 in TLS

A statistical analysis of ciphertexts forms the core of our attacks. We stress that the attacks are ciphertext-only: no sophisticated timing measurement is needed on the part of the adversary, the attacker does not need to be located close to the server, and no packet injection capability is required (all premises for Lucky 13). Instead, it suffices for the adversary to record encrypted traffic for later offline analysis. Provoking the required repeated encryption and transmission of the target plaintext, how-

—AlFardan, Bernstein, Paterson, Poettering, Schuldt,
"On the security of RC4 in TLS",
USENIX Security Symposium 2013

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.

As reported at 29C3:

Collected 3 million certificiates with RSA public keys.

Factored 103 keys using GCD algorithm:

$$N_1 = pq_1 \qquad N_2 = pq_2$$

$$\gcd(N_1, N_2) = p$$

Oops, bad RNG. End of story?

Most commonly shared factor appears 46 times

c0000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000002f9

Next most common factor appears 7 times

c924249224929249924949244924492
24929249924949244924249224929249
92494924492424922492924992494924
492424922492924992494924492424e5

Factoring RSA keys from certified smart cards: Coppersmith in the wild

Daniel J. Bernstein, Yun-An Chang, Chen-Mou Cheng, Li-Ping Chou, Nadia Heninger, Tanja Lange, and Nicko van Someren. Asiacrypt 2013.

Factored 80 more keys using guessing, trial division, and nifty math tricks.

- Nontrivial GCD is not the only way RSA can fail with bad randomness.
- Faulty hardware RNG in Renesas AE45C1 microcontroller.
- Failure of some Chunghwa Telecom HiCOS PKI smart cards to post-process output.

June 19, 2013, Meanwhile at the NSA

The SIMON and SPECK Families of
Lightweight Block Ciphers

Ray Beaulieu and Douglas Shors and
Jason Smith and Stefan
Treatman-Clark and Bryan Weeks and
Louis Wingers.

http://eprint.iacr.org/2013/404
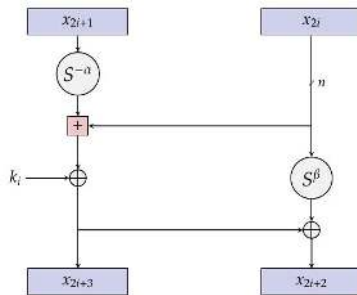
Figure 4.1: Speck round function; (x_{2i+1}, x_{2i}) denotes the subcipher after i steps of encryption.

June 19, 2013, Meanwhile at the NSA

The SIMON and SPECK Families of
Lightweight Block Ciphers

Ray Beaulieu and Douglas Shors and
Jason Smith and Stefan
Treatman-Clark and Bryan Weeks and
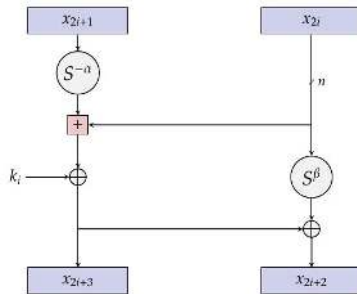Louis Wingers.

http://eprint.iacr.org/2013/404

Figure 4.1: Speck round function; (x_{2i+1}, x_{2i}) denotes the subcipher after i steps of encryption.

4 follow-up papers on ePrint $\Rightarrow$ success on distracting the cryptographers.

July 2013: TweetNaCl

The NaCl library in 100 tweets!
https://twitter.com/tweetnacl

July 2013: TweetNaCl

TweetNaCl @TweetNaCl · 19
0x4141,0x0a4d,0x0070,0xe898,0x7779,0x4079,0x8cc7,0xfe73,0x2
,0x6cee,0x5203},D2=
{0xf159,0x26b2,0x9b94,0xebd6,0xb156,0x8283,0x149a,0x00e0,
Expand ← Reply ⟲ Retweet ★ Favorite ••• M

TweetNaCl @TweetNaCl · 19
randombytes(u8*,u64);static const u8 _0[16],_9[32]={9};static
const gf gf0,gf1={1},_121665={0xDB41,1},D=
{0x78a3,0x1359,0x4dca,0x75eb,0xd8ab,
Expand ← Reply ⟲ Retweet ★ Favorite ••• M

TweetNaCl @TweetNaCl · 19
typedef unsigned char u8;typedef unsigned int u32;typedef
unsigned long long u64;typedef long long i64;typedef i64
gf[16];extern void
Expand ← Reply ⟲ Retweet ★ Favorite ••• M

TweetNaCl @TweetNaCl · 19
#define sv static void
Expand ← Reply ⟲ Retweet ★ Favorite ••• M

TweetNaCl @TweetNaCl · 19
#define FOR(i,n) for (i = 0;i < n;++i)
Expand ← Reply ⟲ Retweet ★ Favorite ••• M

TweetNaCl @TweetNaCl 19
#include "tweetnacl.h"
Expand ← Reply ⟲ Retweeted ★ Favorite ••• M

The NaCl library in 100 tweets!
https://twitter.com/tweetnacl

Advertisement:
Hear more about NaCl tomorrow at
You-Broke-The-Internet assembly
Operating systems session.

2013-12-29 13:00 Hall E

Lavabit

My Fellow Users,

I have been forced to make a difficult decision: to become complicit in crimes against the American people or walk away from nearly ten years of hard work by shutting down Lavabit. After significant soul searching, I have decided to suspend operations. I wish that I could legally share with you the events that led to my decision. I cannot. I feel you deserve to know what's going on--the first amendment is supposed to guarantee me the freedom to speak out in situations like this. Unfortunately, Congress has passed laws that say otherwise. As things currently stand, I cannot share my experiences over the last six weeks, even though I have twice made the appropriate requests.

What's going to happen now? We've already started preparing the paperwork needed to continue to fight for the Constitution in the Fourth Circuit Court of Appeals. A favorable decision would allow me resurrect Lavabit as an American company.

This experience has taught me one very important lesson: without congressional action or a strong judicial precedent, I would _strongly_ recommend against anyone trusting their private data to a company with physical ties to the United States.

Sincerely,
Ladar Levison
Owner and Operator, Lavabit LLC

YOU ARE COMMANDED to appear and testify before the United States district court at the time, date, and place shown below to testify before the court's grand jury. When you arrive, you must remain at the court until the judge or a court officer allows you to leave.

| Place: | UNITED STATES DISTRICT COURT
401 Courthouse Square
Alexandria, Virginia 22314 | Date and Time: | July 16, 2013 | 9:30 AM |

You must also bring with you the following documents, electronically stored information, or objects (blank if not applicable):

In addition to your personal appearance, you are directed to bring to the grand jury the public and private encryption keys used by lavabit.com in any SSL (Secure Socket Layer) or TLS (Transport Security Layer) sessions, including HTTPS sessions with clients using the lavabit.com web site and encrypted SMTP communications (or Internet communications using other protocols) with mail servers;

Any other information necessary to accomplish the installation and use of the pen/trap device ordered by Judge Buchanan on June 28, 2013, unobtrusively and with minimum interference to the services that are accorded persons with respect to whom the installation and use is to take place;

If such information is electronically stored or unable to be physically transported to the grand jury, you may provide a copy of the information to the Federal Bureau of Investigation. Provision of this information to the FBI does not excuse your personal appearance.

Date: July 11, 2013   CLERK OF COURT

TLS RSA Key Exchange
Why forward secrecy is important

hello →

← certificate, public RSA key

RSAEnc$_{RSAkey}$(AES key) →

← AESEnc$_{AESkey}$(website contents)
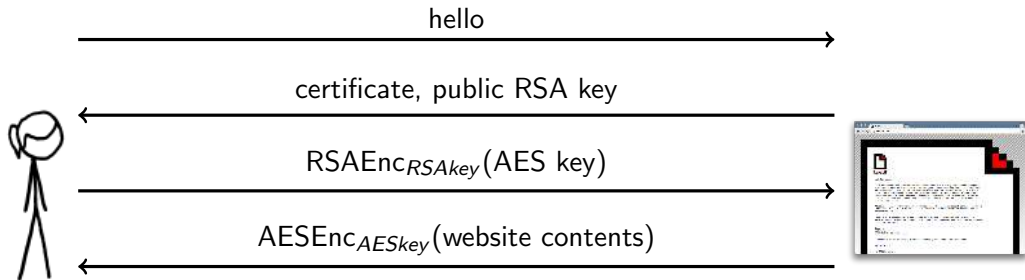
An adversary with Lavabit's private key can

- impersonate Lavabit.com to anyone
- decrypt traffic from now on *and from any point in the past*.

TLS Diffie-Hellman Key Exchange

Why forward secrecy is important

hello, g^x

g^y, certificate, public RSA key

$\text{RSASign}_{RSAkey}(g^x, g^y)$

$\text{AESEnc}_{g^{xy}}(\text{website contents})$
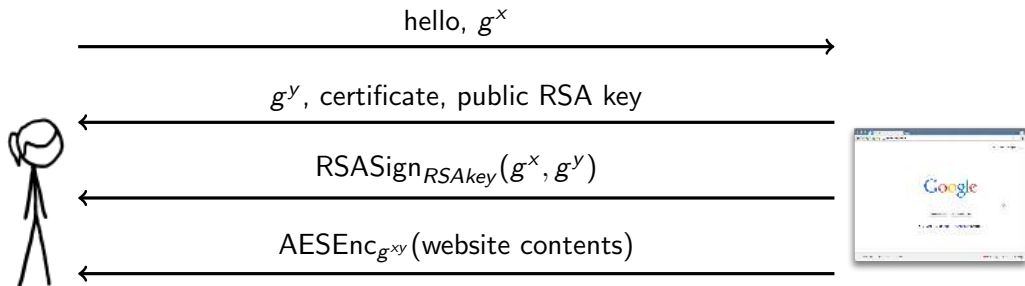
An adversary with Lavabit's private key can

- impersonate Lavabit.com to anyone

Forward secrecy: cannot retroactively decrypt historical traffic if the private keys were forgotten.

www.ccc.de
Identity verified

Permissions   Connection

🔒 The identity of this website has been verified by CAcert Class 3 Root.

Certificate Information

🔒 Your connection to www.ccc.de is encrypted with 256-bit encryption.

The connection uses TLS 1.0.

The connection is encrypted using AES_256_CBC, with SHA1 for message authentication and DHE_RSA as the key exchange mechanism.

ℹ️ **Site information**
You have never visited this site before today.

What do these mean?

Your Homework:

- If you're an end-user, a website enables forward secrecy if you see a cipher suite with DHE (Diffie-Hellman ephemeral) or ECDHE (elliptic-curve Diffie-Hellman ephemeral).

  ccc.de has enabled forward secrecy.

www.microsoft.com
Identity verified

Permissions | Connection

The identity of this website has been verified by MSIT Machine Auth CA 2.

Certificate Information

Your connection to www.microsoft.com is encrypted with 128-bit encryption.

The connection uses TLS 1.0.

The connection is encrypted using RC4_128, with MD5 for message authentication and RSA as the key exchange mechanism.

The server does not support the TLS renegotiation extension.

Site information
You first visited this site on Oct 4, 2013.

- If you run a website, enable forward secrecy! See e.g. https://bettercrypto.org

  microsoft.com does not offer forward secrecy.

- If you build a privacy tool, use end-to-end crypto.

The server's security certificate is revoked!

You attempted to reach **lavabit.com**, but the certificate that the server presented has been revoked by its issuer. This means that the security credentials the server presented absolutely should not be trusted. You may be communicating with an attacker.

[Back to safety]

▶ Help me understand

August 2013: MEGAMOS crypto

At VW's request, English court censors Usenix Security presentation on keyless entry systems for luxury cars

Cory Doctorow at 7:43 pm Sat, Jul 27, 2013

Baris Ege, Flavio Garcia, Roel Verdult break VW car immobilizers.

Paper stopped from being published since it contained "secret" crypto algorithm.

August 2013: CRYPTO Rump session

Using full-disk encryption
Email with PGP
Elliptic curves in your browser
for forward secrecy

Hardware tokens for crypto
Using bitcoins to pay
Everybody use **CRYPTO**
Screw the NSA

Full song: http://www.youtube.com/watch?v=0ricox_ozb4

Scary Paper of the Year: *Stealthy Dopant-Level Hardware Trojans*

by Becker, Regazzoni, Paar, and Burleson, CHES 2013
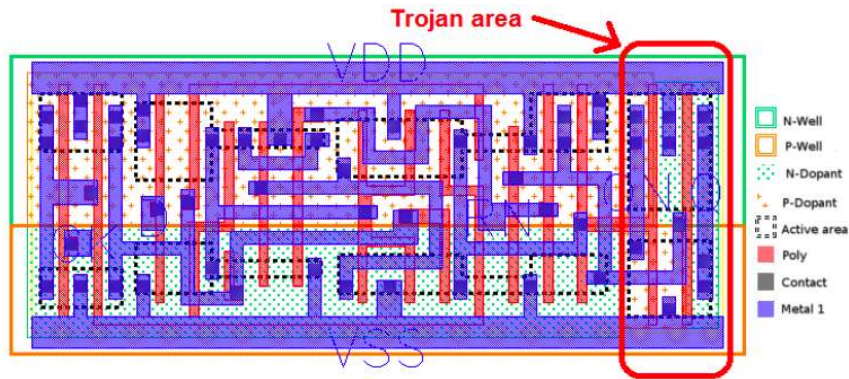
Fig. 2. Layout of the Trojan DFFR_X1 gate. The gate is only modified in the highlighted area by changing the dopant mask. The resulting Trojan gate has an output of $Q = V_{DD}$ and $QN = GND$.

DUAL_EC RNG: history part I

Earliest public source (?) June 2004, draft of ANSI X9.82:
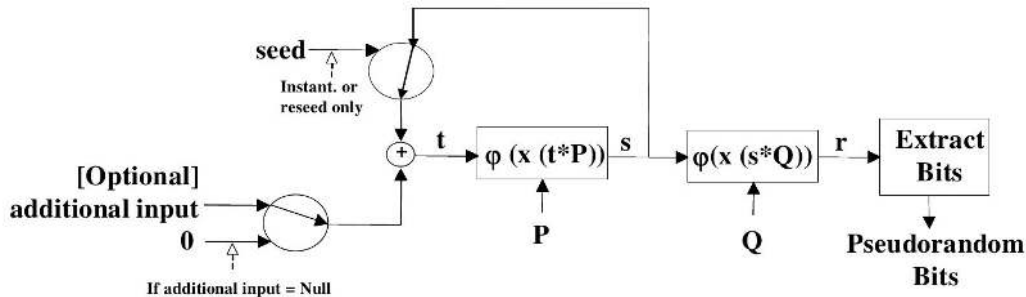
φ gives all but the top 16 bits $\Rightarrow$ about 2^{15} points sQ match given string.

Claim:

Dual_EC_DRBG is based on the following hard problem, sometimes known as the "elliptic curve discrete logarithm problem" (ECDLP): given points P and Q on an elliptic curve of order n, find a such that $Q = aP$.

DUAL_EC RNG: common public history part II

Various public warning signals:

- Gjøsteen (March 2006): output sequence is biased.
  "While the practical impact of these results are modest, it is hard to see how these flaws would be acceptable in a pseudo-random bit generator based on symmetric cryptographic primitives. They should not be accepted in a generator based on number-theoretic assumptions."

- Brown (March 2006): security "proof"
  "This proof makes essential use of Q being random." If d with $dQ = P$ is known then $dR_i = S_{i+1}$, concludes that there might be distinguisher.

- Sidorenko & Schoenmakers (May 2006): output sequence is even more biased. Answer: Too late to change, already implemented.

- Shumow & Ferguson (August 2007): Backdoor if d is known.

- NIST standard gets appendix about choosing points verifiably at random, continues to recommend fixed P and Q.

September 2013: NSA Bullrun program

- (TS//SI//REL TO USA, FVEY) Insert vulnerabilities into commercial encryption systems, IT systems, networks, and endpoint communications devices used by targets.
- (TS//SI//REL TO USA, FVEY) Collect target network data and metadata via cooperative network carriers and/or increased control over core networks.
- (TS//SI//REL TO USA, FVEY) Leverage commercial capabilities to remotely deliver or receive information to and from target endpoints.
- (TS//SI//REL TO USA, FVEY) Exploit foreign trusted computing platforms and technologies.
- (TS//SI//REL TO USA, FVEY) Influence policies, standards and specification for commercial public key technologies.
- (TS//SI//REL TO USA, FVEY) Make specific and aggressive investments to facilitate the development of a robust exploitation capability against Next-Generation Wireless (NGW) communications.

September 2013: NSA Bullrun program

- (TS//SI//REL TO USA, FVEY) Insert vulnerabilities into commercial encryption systems, IT systems, networks, and endpoint communications devices used by targets.
- (TS//SI//REL TO USA, FVEY) Collect target network data and metadata via cooperative network carriers and/or increased control over core networks.
- (TS//SI//REL TO USA, FVEY) Leverage commercial capabilities to remotely deliver or receive information to and from target endpoints.
- (TS//SI//REL TO USA, FVEY) Exploit foreign trusted computing platforms and technologies.
- (TS//SI//REL TO USA, FVEY) Influence policies, standards and specification for commercial public key technologies.
- (TS//SI//REL TO USA, FVEY) Make specific and aggressive investments to facilitate the development of a robust exploitation capability against Next-Generation Wireless (NGW) communications.

Later NYT names Dual_EC_DRBG. . .

September 2013: NSA Bullrun program

- (TS//SI//REL TO USA, FVEY) Insert vulnerabilities into commercial encryption systems, IT systems, networks, and endpoint communications devices used by targets.
- (TS//SI//REL TO USA, FVEY) Collect target network data and metadata via cooperative network carriers and/or increased control over core networks.
- (TS//SI//REL TO USA, FVEY) Leverage commercial capabilities to remotely deliver or receive information to and from target endpoints.
- (TS//SI//REL TO USA, FVEY) Exploit foreign trusted computing platforms and technologies.
- (TS//SI//REL TO USA, FVEY) Influence policies, standards and specification for commercial public key technologies.
- (TS//SI//REL TO USA, FVEY) Make specific and aggressive investments to facilitate the development of a robust exploitation capability against Next-Generation Wireless (NGW) communications.

Later NYT names Dual_EC_DRBG. . . but surely nobody uses that piece of shit?!

September 2013: NSA Bullrun program

- (TS//SI//REL TO USA, FVEY) Insert vulnerabilities into commercial encryption systems, IT systems, networks, and endpoint communications devices used by targets.
- (TS//SI//REL TO USA, FVEY) Collect target network data and metadata via cooperative network carriers and/or increased control over core networks.
- (TS//SI//REL TO USA, FVEY) Leverage commercial capabilities to remotely deliver or receive information to and from target endpoints.
- (TS//SI//REL TO USA, FVEY) Exploit foreign trusted computing platforms and technologies.
- (TS//SI//REL TO USA, FVEY) Influence policies, standards and specification for commercial public key technologies.
- (TS//SI//REL TO USA, FVEY) Make specific and aggressive investments to facilitate the development of a robust exploitation capability against Next-Generation Wireless (NGW) communications.

Later NYT names Dual_EC_DRBG. . . but surely nobody uses that piece of shit?!

NIST's DRBG Validation List: RSA's BSAFE has Dual_EC_DRBG enabled and default.

September 2013: NSA Bullrun program

- (TS//SI//REL TO USA, FVEY) Insert vulnerabilities into commercial encryption systems, IT systems, networks, and endpoint communications devices used by targets.
- (TS//SI//REL TO USA, FVEY) Collect target network data and metadata via cooperative network carriers and/or increased control over core networks.
- (TS//SI//REL TO USA, FVEY) Leverage commercial capabilities to remotely deliver or receive information to and from target endpoints.
- (TS//SI//REL TO USA, FVEY) Exploit foreign trusted computing platforms and technologies.
- (TS//SI//REL TO USA, FVEY) Influence policies, standards and specification for commercial public key technologies.
- (TS//SI//REL TO USA, FVEY) Make specific and aggressive investments to facilitate the development of a robust exploitation capability against Next-Generation Wireless (NGW) communications.

Later NYT names Dual_EC_DRBG. . . but surely nobody uses that piece of shit?!

NIST's DRBG Validation List: RSA's BSAFE has Dual_EC_DRBG enabled and default.

NIST re-opens discussions on SP800.90; recommmends against using Dual_EC.
RSA suggests changing default in BSAFE.

How expensive is using the backdoor?

Rereading the standard:

" $x(A)$ is the x-coordinate of the point A on the curve, given in affine coordinates. An implementation may choose to represent points internally using other coordinate systems; for instance, when efficiency is a primary concern. In this case, a point shall be translated back to affine coordinates before $x()$ is applied."

How expensive is using the backdoor?

Rereading the standard:

" $x(A)$ is the x-coordinate of the point A on the curve, given in affine coordinates. An implementation may choose to represent points internally using other coordinate systems; for instance, when efficiency is a primary concern. In this case, a point shall be translated back to affine coordinates before $x()$ is applied."

Given $r_i = \varphi(x(s_i Q))$, $r_{i+1} = \varphi(x(s_{i+1} Q))$, and NSA backdoor $d = log_P(Q)$.

1. Expand r_i to candidate $Q_i = s_i Q$, [50% chance; if fail move on to next candidate]
2. compute candidate $P_{i+1} = dQ_i$ and candidate $s_{i+1} = \varphi(x(P_{i+1}))$
3. check, $\varphi(x(s_{i+1} Q))$ against r_{i+1}. [if fail, goto 1.; else most likely done!]

How expensive is using the backdoor?

Rereading the standard:

" $x(A)$ is the x-coordinate of the point A on the curve, given in affine coordinates. An implementation may choose to represent points internally using other coordinate systems; for instance, when efficiency is a primary concern. In this case, a point shall be translated back to affine coordinates before $x()$ is applied."

Given $r_i = \varphi(x(s_i Q))$, $r_{i+1} = \varphi(x(s_{i+1} Q))$, and NSA backdoor $d = log_P(Q)$.

1. Expand r_i to candidate $Q_i = s_i Q$, [50% chance; if fail move on to next candidate]
2. compute candidate $P_{i+1} = d Q_i$ and candidate $s_{i+1} = \varphi(x(P_{i+1}))$
3. check, $\varphi(x(s_{i+1} Q))$ against r_{i+1}. [if fail, goto 1.; else most likely done!]

Timings on i7 M620 Core

missing	16 bits	24 bits	32 bits
1 core	20s	85m	15d4h

How expensive is using the backdoor?

Rereading the standard:

" $x(A)$ is the x-coordinate of the point A on the curve, given in affine coordinates. An implementation may choose to represent points internally using other coordinate systems; for instance, when efficiency is a primary concern. In this case, a point shall be translated back to affine coordinates before $x()$ is applied."

Given $r_i = \varphi(x(s_i Q))$, $r_{i+1} = \varphi(x(s_{i+1} Q))$, and NSA backdoor $d = log_P(Q)$.

1. Expand r_i to candidate $Q_i = s_i Q$, [50% chance; if fail move on to next candidate]
2. compute candidate $P_{i+1} = dQ_i$ and candidate $s_{i+1} = \varphi(x(P_{i+1}))$
3. check, $\varphi(x(s_{i+1} Q))$ against r_{i+1}. [if fail, goto 1.; else most likely done!]

Timings on i7 M620 Core

missing	16 bits	24 bits	32 bits
1 core	20s	85m	15d4h
64k cores			20s

How expensive is using the backdoor?

Rereading the standard:

" $x(A)$ is the x-coordinate of the point A on the curve, given in affine coordinates. An implementation may choose to represent points internally using other coordinate systems; for instance, when efficiency is a primary concern. In this case, a point shall be translated back to affine coordinates before $x()$ is applied."

Given $r_i = \varphi(x(s_i Q))$, $r_{i+1} = \varphi(x(s_{i+1} Q))$, and NSA backdoor $d = log_P(Q)$.

1. Expand r_i to candidate $Q_i = s_i Q$, [50% chance; if fail move on to next candidate]
2. compute candidate $P_{i+1} = dQ_i$ and candidate $s_{i+1} = \varphi(x(P_{i+1}))$
3. check, $\varphi(x(s_{i+1} Q))$ against r_{i+1}. [if fail, goto 1.; else most likely done!]

Timings on i7 M620 Core

	missing	16 bits	24 bits	32 bits
1 core		20s	85m	15d4h
64k cores				20s

From the standard:

"For performance reasons, the value of outlen should be set to the maximum value as provided in Table 4."

Don't give us fewer bits!

September 2013: SHA-3 controversy erupts

How about the NIST curves?

May 2013, Bernstein & Lange: "Security dangers of the NIST curves"

Matthew Green @matthew_d_green — 8 Jun
@hashbreaker Flipside: What if NIST/NSA know a weakness in 1/10000000 curves? NIST searches space for curves that *aren't* vulnerable.
Details   Reply   Retweet   Favorite   More

Tanja Lange @hyperelliptic — 8 Jun
@matthew_d_green @hashbreaker and blames choice on NSA's Jerry Solinas? Not a fan of "Look, I prove to you it's random" but no tinfoil. yet.
Details   Reply   Delete   Favorite   More

Matthew Green @matthew_d_green — 8 Jun
@hyperelliptic @hashbreaker Not that I seriously believe this, but it's as reasonable as the alternative.
Details   Reply   Retweet   Favorite   More

Tanja Lange @hyperelliptic — 8 Jun
@matthew_d_green @hashbreaker main point: there are now better curves and implementations than in 1999 & stronger side-channel attacks, too.
Details   Reply   Delete   Favorite   More

Green: "Flipside: What if NIST/NSA know a weakness in 1/10000000 curves? NIST searches space for curves at 'arent' vulnerable."

How about the NIST curves?

May 2013, Bernstein & Lange: "Security dangers of the NIST curves"

Green: "Flipside: What if NIST/NSA know a weakness in 1/10000000 curves? NIST searches space for curves at 'arent' vulnerable."

September 2013

Matthew Green
@matthew_d_green

Discussion with @hashbreaker from when I was younger and more naive. #nist #ecc twitter.com/matthew_d_gree…
12:41 PM - 11 Sep 2013

SafeCurves: choosing safe curves for elliptic-curve cryptography

All known security
criteria for elliptic
curves, machine verified.

Elligator: undetectable
curve points.

New Curve3617.

SafeCurves: choosing safe curves for elliptic-curve cryptography

All known security
criteria for elliptic
curves, machine verified.

Elligator: undetectable
curve points.

New Curve3617.

Also: can the curve be
backdoored?

http://safecurves.
cr.yp.to

SafeCurves: choosing safe curves for elliptic-curve cryptography

All known security criteria for elliptic curves, machine verified.

Elligator: undetectable curve points.

New Curve3617.

Also: can the curve be backdoored?

http://safecurves.cr.yp.to

Curve	Safe?	Parameters:			ECDLP security:				ECC security:			
		field	equation	base	rho	transfer	disc	rigid	ladder	twist	complete	ind
Anomalous	False	True ✔	True ✔	True ✔	True ✔	False	False	True ✔	False	False	False	False
M-221	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔
E-222	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔
NIST P-224	False	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	False	False	False	False	False
Curve1174	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔
Curve25519	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔
BN(2,254)	False	True ✔	True ✔	True ✔	True ✔	False	False	True ✔	False	False	False	False
brainpoolP256t1	False	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	False	False	False	False
ANSSIFRP256v1	False	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	False	False	False	False	False
NIST P-256	False	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	False	False	True ✔	False	False
secp256k1	False	True ✔	True ✔	True ✔	True ✔	True ✔	False	True ✔	False	True ✔	False	False
M-383	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔
Curve383187	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔
brainpoolP384t1	False	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	False	True ✔	False	False
NIST P-384	False	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	False	False	True ✔	False	False
Curve3617	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔	True ✔

Bitcoin goes mainstream, bringing ECDSA with it

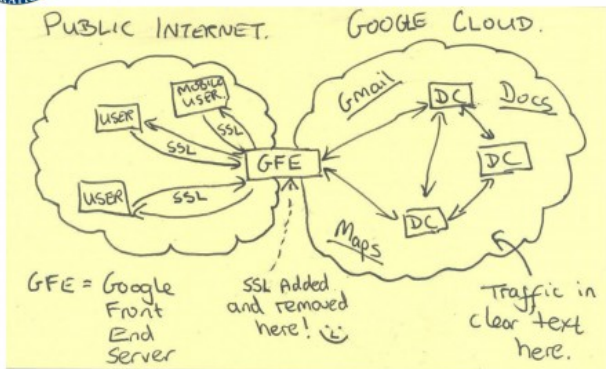August 2013: Android Java RNG vulnerability blamed for bitcoin thefts

1HKywxiL4JziqXrzLKhmB6a74ma6kxbSDj has stolen 59 bitcoin from addresses using repeated ECDSA signature randomness.
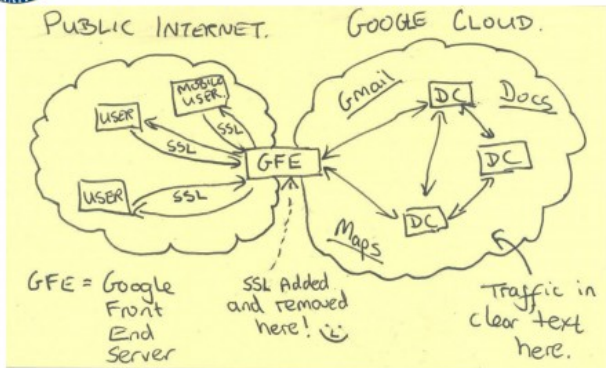
October 2013: MUSCULAR

Official Google statement:
"We are outraged"

October 2013: MUSCULAR

TOP SECRET//SI//NOFORN

Current Efforts - Google

Official Google statement:
"We are outraged"

Unofficial Google statement:
"Fuck these guys."

TOP SECRET//SI//NOFORN

SSL crypto not great – but even worse when it's circumvented.

Meanwhile at the NSA II

to filter the FORNSAT survey environment for this traffic and extract various types of WoW metadata for SIGINT development and network knowledge enrichment.

(U) World of Warcraft

(U) Communication is at the core of online gaming and in WoW there are many ways to communicate and interact in the virtual world. A player has a character ID and can join different groups. A "party" brings players together for a common, defined purpose or quest. It is temporary and task-oriented. "Guilds," on the other hand, are for characters with persisting relationships and can take on an organizational structure with ranks and positions. The guild is more permanent and ideological. Characters can communicate verbally and non-verbally and may set up different types of channels to talk within a

December 2013: trouble with XCB disk-encryption standard

XCBv2 as specified in [12] is not secure as a TES. We found an easy distinguishing attack on XCBv2. The attack works because of a faulty padding scheme, and there seems to be no easy way to fix this problem. However, if the inputs to XCBv2 are such that their lengths are multiples of the block length of the block

Even for the restricted message space, XCBv2b (possibly) does not have the security bound as claimed in [12]. This is due to the fact that the proof of the security theorem in [12] is wrong. The error stems from a faulty calculation of collision probabilities in the inc function. We point out the mistake by showing concrete examples where that the bound on the collision probabilities in the inc function as given in [12] are violated.

<div align="right">

—Chakraborty, Hernandez-Jimenez, Sarkar,
"Another look at XCB",
4 December 2013

</div>

December 2013: trouble with XCB disk-encryption standard

XCBv2 as specified in [12] is not secure as a TES. We found an easy distinguishing attack on XCBv2. The attack works because of a faulty padding scheme, and there seems to be no easy way to fix this problem. However, if the inputs to XCBv2 are such that their lengths are multiples of the block length of the block

Even for the restricted message space, XCBv2b (possibly) does not have the security bound as claimed in [12]. This is due to the fact that the proof of the security theorem in [12] is wrong. The error stems from a faulty calculation of collision probabilities in the inc function. We point out the mistake by showing concrete examples where that the bound on the collision probabilities in the inc function as given in [12] are violated.

bound.

XCBv2 was derived as a small modification of XCBv1. The authors said that the modifications were made to enable easy analysis [12]. Though it is not very clear to us, how these modifications help in the analysis. Our analysis reveals that any modification in an existing cryptographic scheme should be done with utmost care,

—Chakraborty, Hernandez-Jimenez, Sarkar,
"Another look at XCB",
4 December 2013

December 2013: acoustic attacks against GnuPG

Acoustic cryptanalysis = power analysis with acoustic transmission of power signal.
News: **4096-bit GnuPG RSA keys extracted in one hour.**

—Genkin, Shamir, Tromer,
"RSA key extraction via low-bandwidth acoustic cryptanalysis",
18 December 2013

December 2013: acoustic attacks against GnuPG

Acoustic cryptanalysis = power analysis with acoustic transmission of power signal.
News: **4096-bit GnuPG RSA keys extracted in one hour.**

—Genkin, Shamir, Tromer,
"RSA key extraction via low-bandwidth acoustic cryptanalysis",
18 December 2013

December 2013: Obama's NSA review panel report

Upon review, however, we are unaware of any vulnerability created by the US Government in generally available commercial software that puts users at risk of criminal hackers or foreign governments decrypting their data. Moreover, it appears that in the vast majority of generally used, commercially available encryption software, there is no vulnerability, or "backdoor," that makes it possible for the US Government or anyone else to achieve unauthorized access.[174]

---

[174] Any cryptographic algorithm can become exploitable if implemented incorrectly or used improperly.

Some wild speculation left undenied by the previous denial:

The NSA could have

- backdoored the Dual-EC DRBG and only they have the secret key.

- backdoored the NIST curves and only they have the secret key and computational power needed in the backdoor.

- introduced vulnerabilities or backdoors into cryptographic software such as OpenSSL which are free software and thus not commercially available.

- introduced vulnerabilities or backdoors into Windows, OS X, and Red Hat, only three commerically available OSes out of hundreds on the market.

- introduced backdoors into cryptographic hardware such as the Intel hardware RNG or crypto instructions.

- modified 100% of generally available commercial software to disable encryption whenever possible.

- a backdoor/"key escrow" feature allowing "lawful access" to any AES-encrypted data.

December 2013

Obama on surveillance: "There may be another way of skinning the cat"

(Reuters) - As a key part of a campaign to embed encryption software that it could crack into widely used computer products, the U.S. National Security Agency arranged a secret $10 million contract with RSA, one of the most influential firms in the computer security industry, Reuters has learned.

Documents leaked by former NSA contractor Edward Snowden show that the NSA created and promulgated a flawed formula for generating random numbers to create a "back door" in encryption products, the New York Times reported in September. Reuters later reported that RSA became the most important distributor of that formula by rolling it into a software tool called Bsafe that is used to enhance security in personal computers and many other products.

Undisclosed until now was that RSA received $10 million in a deal that set the NSA formula as the preferred, or default, method for number generation in the BSafe software, according to two sources familiar with the contract. Although that sum might seem paltry, it represented more than a third of the revenue that the relevant division at RSA had taken in during the entire previous year, securities filings show.

December 22, 2013

Recent press coverage has asserted that RSA entered into a "secret contract" with the NSA to incorporate a known flawed random number generator into its BSAFE encryption libraries. We categorically deny this allegation.

We have worked with the NSA, both as a vendor and an active member of the security community. We have never kept this relationship a secret and in fact have openly publicized it. Our explicit goal has always been to strengthen commercial and government security.

Key points about our use of Dual EC DRBG in BSAFE are as follows:

- We made the decision to use Dual EC DRBG as the default in BSAFE toolkits in 2004, in the context of an industry-wide effort to develop newer, stronger methods of encryption. At that time, the NSA had a trusted role in the community-wide effort to strengthen, not weaken, encryption.

- This algorithm is only one of multiple choices available within BSAFE toolkits, and users have always been free to choose whichever one best suits their needs.

- We continued using the algorithm as an option within BSAFE toolkits as it gained acceptance as a NIST standard and because of its value in FIPS compliance. When concern surfaced around the algorithm in 2007, we continued to rely upon NIST as the arbiter of that discussion.

(19) **United States**

(12) **Patent Application Publication**   (10) Pub. No.: **US 2007/0189527 A1**

Brown et al.   (43) **Pub. Date:**   **Aug. 16, 2007**

(54) **ELLIPTIC CURVE RANDOM NUMBER GENERATION**

(76) Inventors: **Daniel R. L. Brown**, Mississauga (CA); **Scott A. Vanstone**, Campbellville (CA)

Correspondence Address:
Blake, Cassels & Graydon LLP
Commerce Court West
P.O. Box 25
Toronto, ON M5L 1A9 (CA)

Publication Classification

(57) **ABSTRACT**

An elliptic curve random number generator avoids escrow keys by choosing a point Q on the elliptic curve as verifiably random. An arbitrary string is chosen and a hash of that string computed. The hash is then converted to a field element of the desired field, the field element regarded as the x-coordinate of a point Q on the elliptic curve and the x-coordinate is tested for validity on the desired elliptic curve. If valid, the x-coordinate is decompressed to the point Q, wherein the choice of which is the two points is also derived from the hash value. Intentional use of escrow keys can provide for back up functionality. The relationship between P and Q is used as an escrow key and stored by for a security domain. The administrator logs the output of the generator to reconstruct the random number with the escrow key.

Hat tip @nymble.

Snippets from the patent

can provide for back up functionality. The relationship between P and Q is used as an escrow key and stored by for a security domain. The administrator logs the output of the generator to reconstruct the random number with the escrow key.

accounts. A more seamless method may be applied for cryptographic applications. For example, in the SSL and TLS protocols, which are used for securing web (HTTP) traffic, a client and server perform a handshake in which their first actions are to exchange random values sent in the clear.

[0054] Many other protocols exchange such random values, often called nonces. If the escrow administrator observes these nonces, and keeps a log of them 508, then later it may be able to determine the necessary r value. This

IN CASE OF REVOLUTION
BREAK GLASS