Crypto news and views

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

Nadia Heninger
University of Pennsylvania

Tanja Lange
Technische Universiteit Eindhoven

Scope of this Talk

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

The Cryptopocalypse

MIT Technology Review

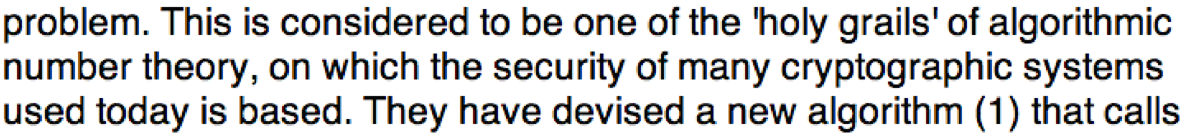
Math Advances Raise the Prospect of an Internet Security Crisis

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

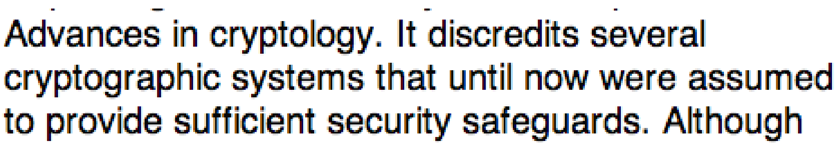
By Tom Simonite on August 2, 2013

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.

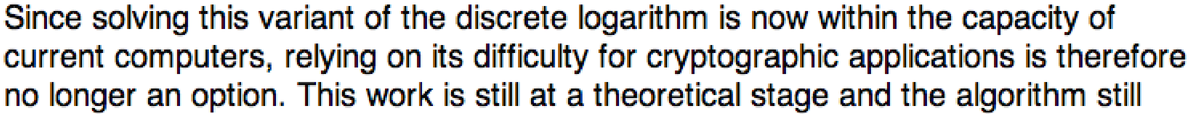
Science Daily May 2014



problem. This is considered to be one of the 'holy grails' of algorithmic number theory, on which the security of many cryptographic systems used today is based. They have devised a new algorithm (1) that calls



Advances in cryptology. It discredits several cryptographic systems that until now were assumed to provide sufficient security safeguards. Although



Since solving this variant of the discrete logarithm is now within the capacity of current computers, relying on its difficulty for cryptographic applications is therefore no longer an option. This work is still at a theoretical stage and the algorithm still

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 $L_N(\alpha) = \exp\big(O((\log N)^\alpha (\log\log N)^{1-\alpha})\big)$. 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].

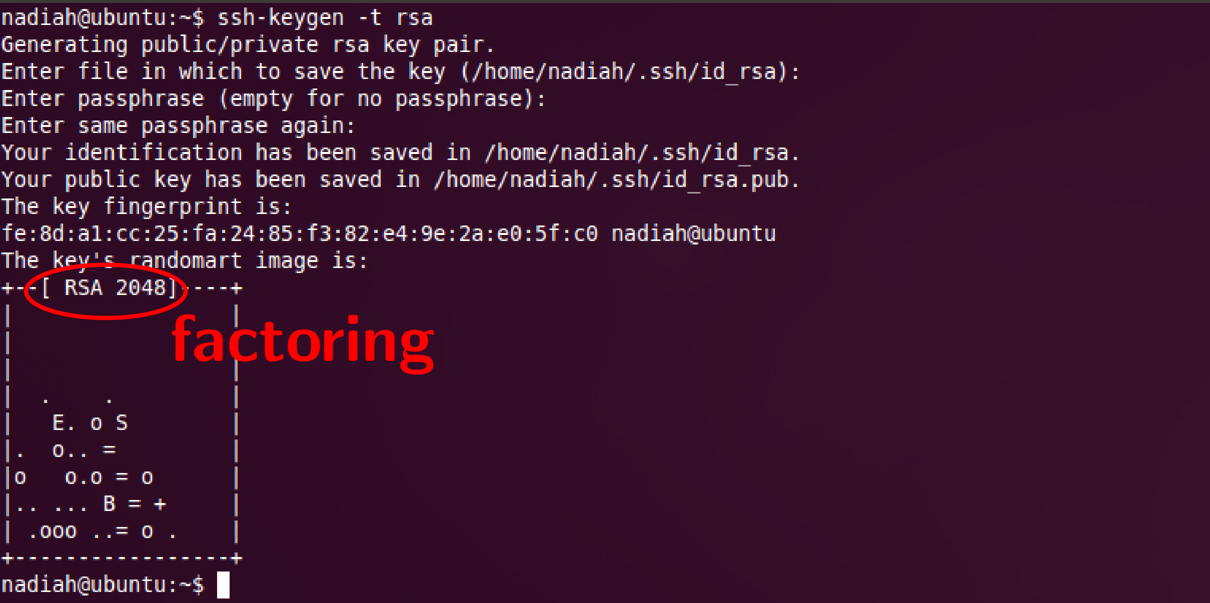
Recently, practical and theoretical progress have been made [Jou13a, GGMZ13, Jou13b] with an emphasis on small to medium characteristic finite fields and composite degree extensions. The most general and efficient algorithm [Jou13b] gives a complexity of $L(1/4 + o(1))$ when the characteristic is smaller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

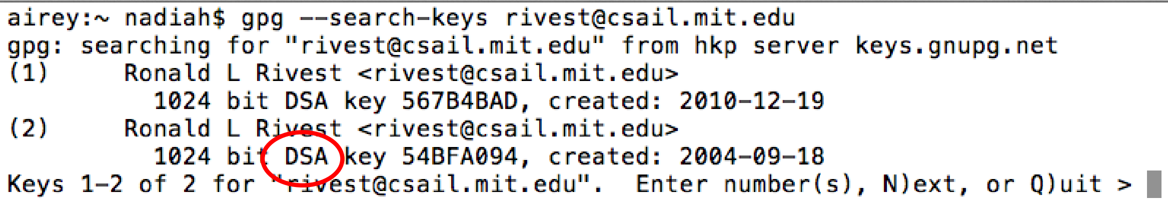
discrete log modulo primes

discrete log in elliptic curve groups



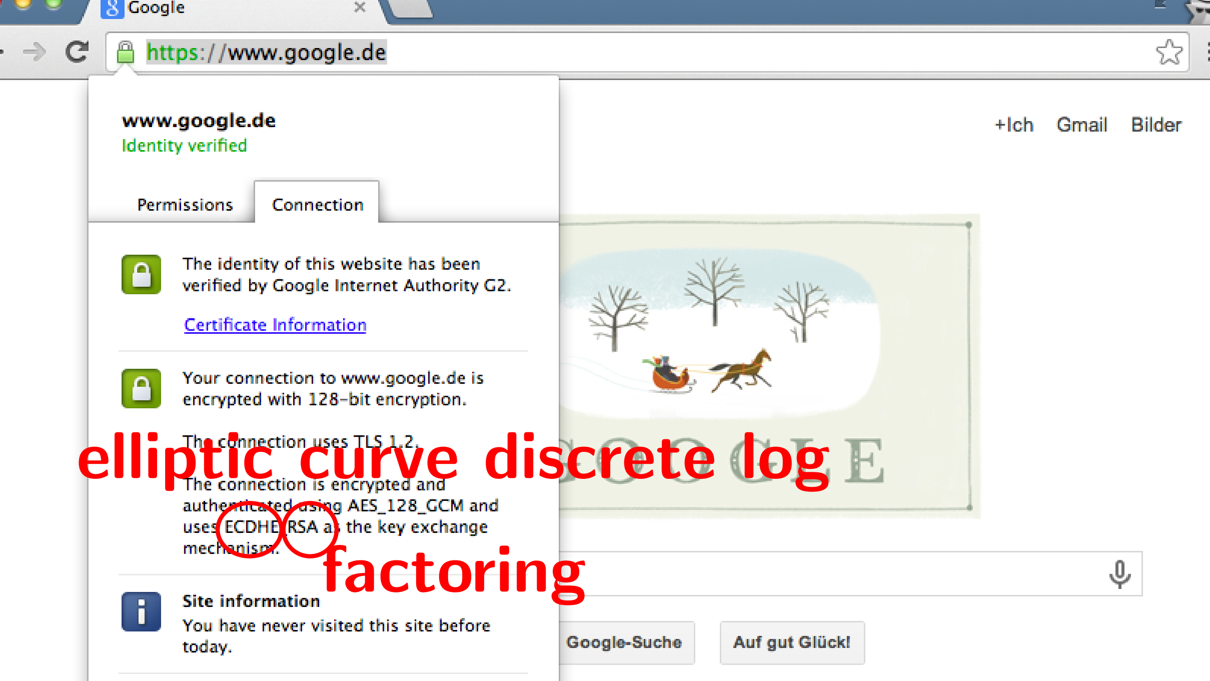
```
nadiah@ubuntu:~$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/nadiah/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/nadiah/.ssh/id_rsa.
Your public key has been saved in /home/nadiah/.ssh/id_rsa.pub.
The key fingerprint is:
fe:8d:a1:cc:25:fa:24:85:f3:82:e4:9e:2a:e0:5f:c0 nadiah@ubuntu
The key's randomart image is:
+--[ RSA 2048]----+
|                 |
|                 |
|                 |
|  .    .         |
|   E. o S        |
|.  o.. =         |
|o   o.o = o      |
|.. ... B = +     |
| .ooo ..= o .    |
+-----------------+
nadiah@ubuntu:~$
```

factoring



```
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



www.google.de

Identity verified

The identity of this website has been verified by Google Internet Authority G2.

Certificate Information

Your connection to www.google.de is encrypted with 128-bit encryption.

The connection uses TLS 1.2.

The connection is encrypted and authenticated using AES_128_GCM and uses ECDHE_RSA as the key exchange mechanism.

Site information

You have never visited this site before today.

elliptic curve discrete log

factoring

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 →	$L((64/9)^{1/3}, 1/3)$	86	116	156
improved constant →	$L((32/9)^{1/3}, 1/3)$	68	92	124
improved exponent →	$L((64/9)^{1/4}, 1/4)$	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 hierarchy 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

Picture credit: Reuters via `www.popularresistance.org`

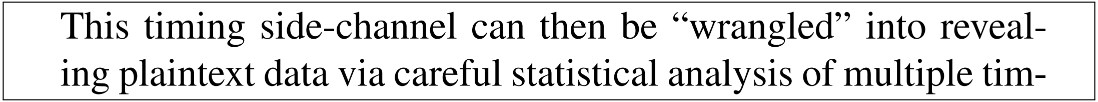
February 2013: timing-padding-oracle attacks against TLS

> `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.

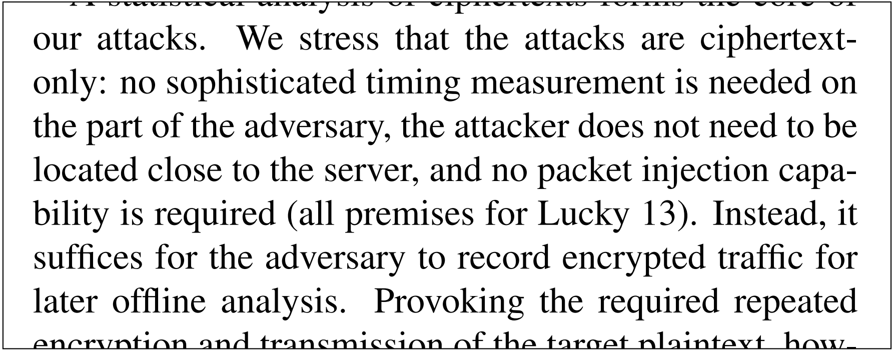
February 2013: TLS algorithm agility to the rescue!

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

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

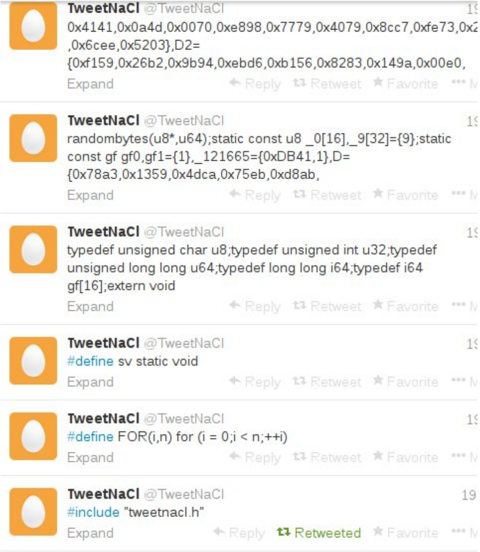
Daniel J. Bernstein, Yun-An Chang, Chen-Mou Cheng, Li-Ping Chou, Nadia Heninger, Tanja Lange, and Nicko van Someren. Asiacrypt 2013.

Taiwanese Citizen Digital Certificate PKI

Factored 180 RSA keys in use using guessing, trial division, and nifty math tricks.

- Deployed crypto can fail catastrophically with bad randomness.
- Faulty hardware RNG in Renesas AE45C1 microcontroller.
- Failure of some Chunghwa Telecom HiCOS PKI smart cards to post-process output.

July 2013: TweetNaCl



TweetNaCl @TweetNaCl 1

```
0x4141,0x0a4d,0x0070,0xe898,0x7779,0x4079,0x8cc7,0xfe73,0x2
,0x6cee,0x5203},D2=
{0xf159,0x26b2,0x9b94,0xebd6,0xb156,0x8283,0x149a,0x00e0,
```

TweetNaCl @TweetNaCl 1

```
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,
```

TweetNaCl @TweetNaCl 1

```
typedef unsigned char u8;typedef unsigned int u32;typedef
unsigned long long u64;typedef long long i64;typedef i64
gf[16];extern void
```

TweetNaCl @TweetNaCl 1

```
#define sv static void
```

TweetNaCl @TweetNaCl 1

```
#define FOR(i,n) for (i = 0;i < n;++i)
```

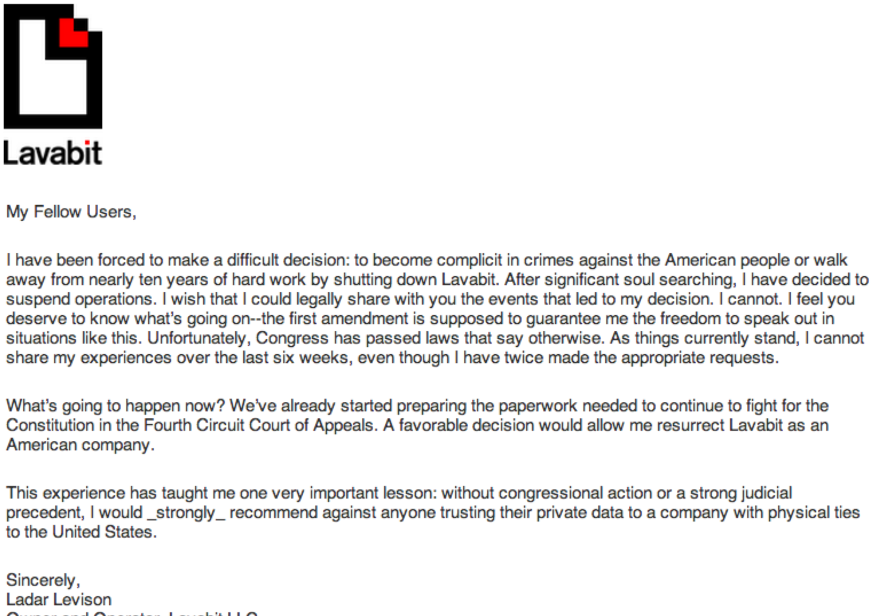
TweetNaCl @TweetNaCl 19

```
#include "tweetnacl.h"
```

`nacl.cr.yp.to`:
high-speed high-security NaCl
(Networking and Cryptography library).

`https://twitter.com/tweetnacl`:
reimplemented all NaCl functions
in just 100 tweets!

August 2013



Lavabit

My Fellow Users,

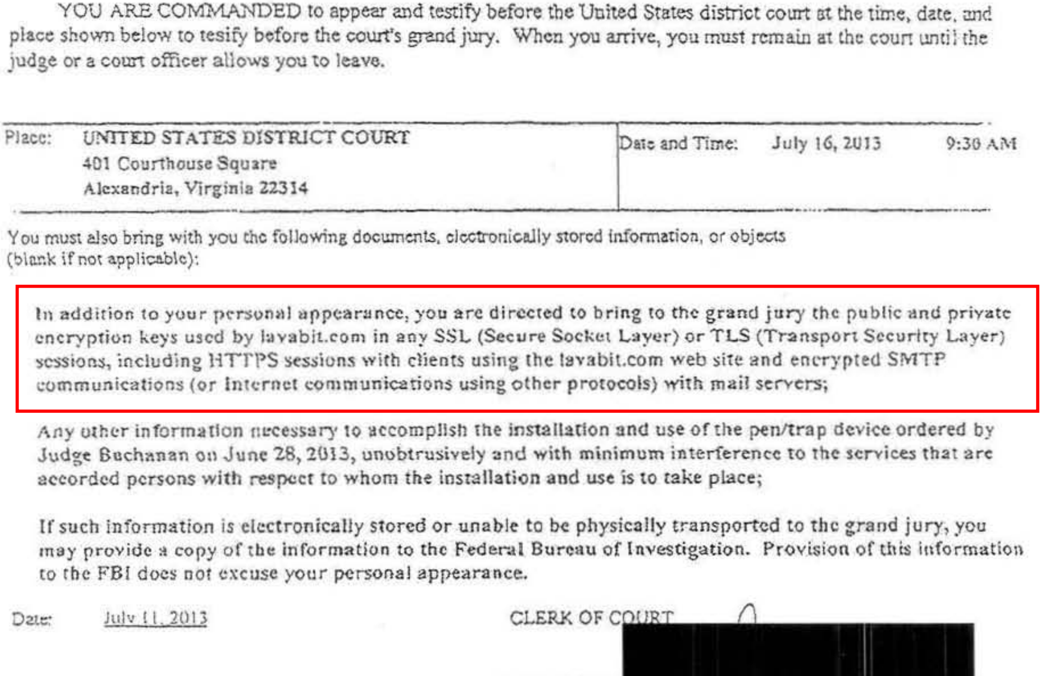
I have been forced to make a difficult decision: to become complicit in crimes against the American people or walk away from nearly ten years of hard work by shutting down Lavabit. After significant soul searching, I have decided to suspend operations. I wish that I could legally share with you the events 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

Lavabit employed two stages of encryption for its paid subscribers: storage encryption and transport encryption. Storage encryption protects emails and other data that rests on Lavabit's servers. Theoretically, no person other than the email user could access the data once it was so encrypted. By using storage encryption, Lavabit held a unique market position in the email industry, as many providers do not encrypt stored data.



YOU ARE COMMANDED to appear and testify before the United States district court at the time, date, and place shown below to tesify 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 [redacted]

UNDER SEAL

UNITED STATES DISTRICT COURT

for the

Eastern District of Virginia

In the Matter of the Search of
(Briefly describe the property to be searched or identify the person by name and address)
INFORMATION ASSOCIATED WITH
[redacted]
THAT IS STORED AT PREMISES
CONTROLLED BY LAVABIT, LLC

Case No. 1:13SW522

SEARCH AND SEIZURE WARRANT

To: Any authorized law enforcement officer

An application by a federal law enforcement officer or an attorney for the government requests the search of the following person or property located in the Northern District of Texas *(identify the person or describe the property to be searched and give its location)*:
See Attachment A

ATTACHMENT B

Particular Things to be Seized

I. Information to be disclosed by Lavabit, LLC (the "Provider")

To the extent that the information described in Attachment A is within the possession, custody, or control of the Provider, including any emails, records, files, logs, or information that has been deleted but is still available to the Provider, the Provider is required to disclose the following information to the government for each account or identifier listed in Attachment A:

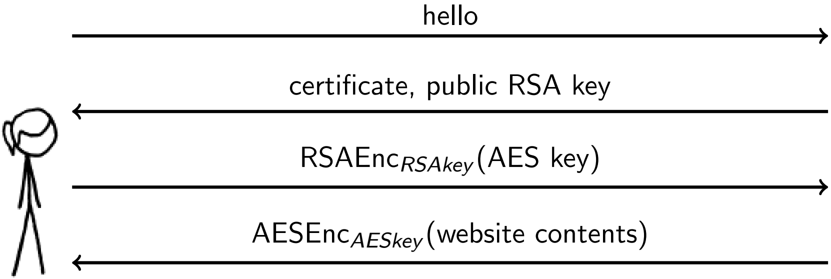
a. All information necessary to decrypt communications sent to or from the Lavabit e-mail account [redacted] including encryption keys and SSL keys;

b. All information necessary to decrypt data stored in or otherwise associated with the Lavabit account [redacted].

Despite the unequivocal language of the August 1 Order, Lavabit dallied and did not comply. Just before the 5:00 pm August 2 deadline, for instance, Levison provided the FBI with an 11-page printout containing largely illegible characters in 4-point type, which he represented to be Lavabit's encryption keys. The Government instructed Lavabit to provide the keys in an industry-standard electronic format by the morning of August 5. Lavabit did not respond.

TLS RSA Key Exchange

Why forward secrecy is important

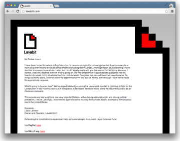


hello

certificate, public RSA key

$\text{RSAEnc}_{RSAkey}(\text{AES key})$

$\text{AESEnc}_{AESkey}(\text{website contents})$



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

Defending the constitution is expensive! Help us by donating to the Lavabit Legal Defense Fund.

Via **PayPal**: here

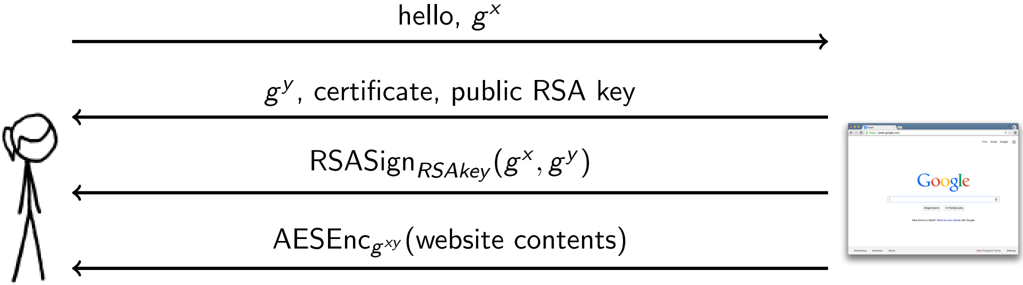
Via **RALLY.org**: here

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

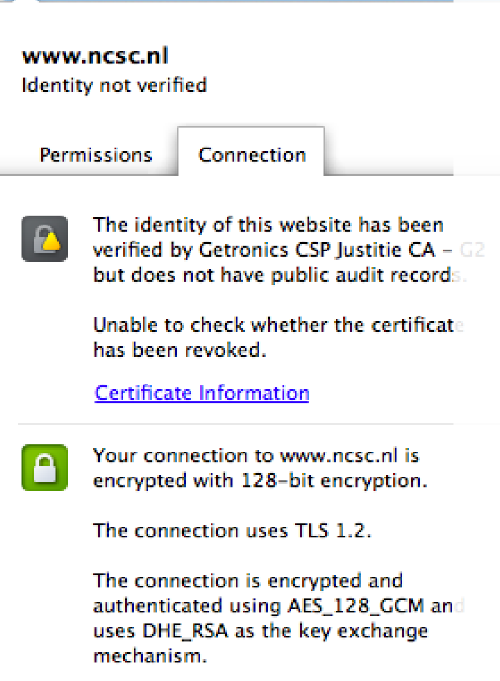
$\mathrm{RSASign}_{RSAkey}(g^x, g^y)$

$\mathrm{AESEnc}_{g^{xy}}(\text{website contents})$

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.ncsc.nl
Identity not verified

Permissions | Connection

The identity of this website has been verified by Getronics CSP Justitie CA – G2 but does not have public audit records.

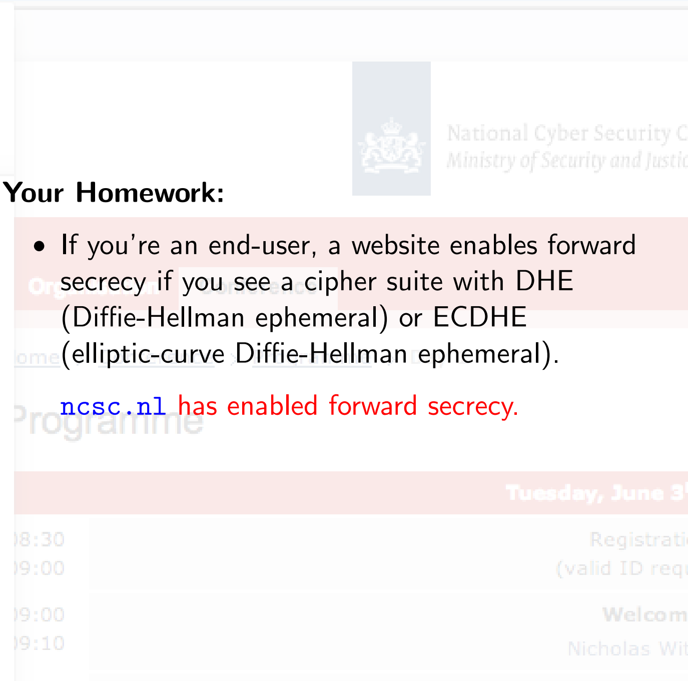
Unable to check whether the certificate has been revoked.

Certificate Information

Your connection to www.ncsc.nl is encrypted with 128-bit encryption.

The connection uses TLS 1.2.

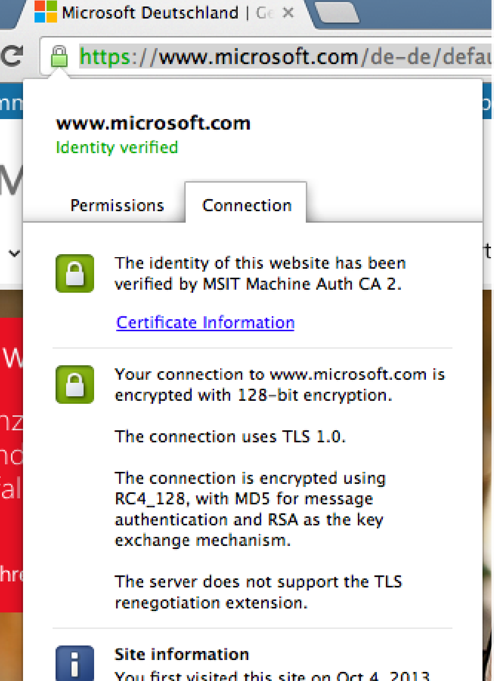
The connection is encrypted and authenticated using AES_128_GCM and uses DHE_RSA as the key exchange mechanism.



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).

`ncsc.nl` has enabled forward secrecy.



Microsoft Deutschland | Ge
https://www.microsoft.com/de-de/default.aspx
Permissions
Connection

www.microsoft.com

Identity verified

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.

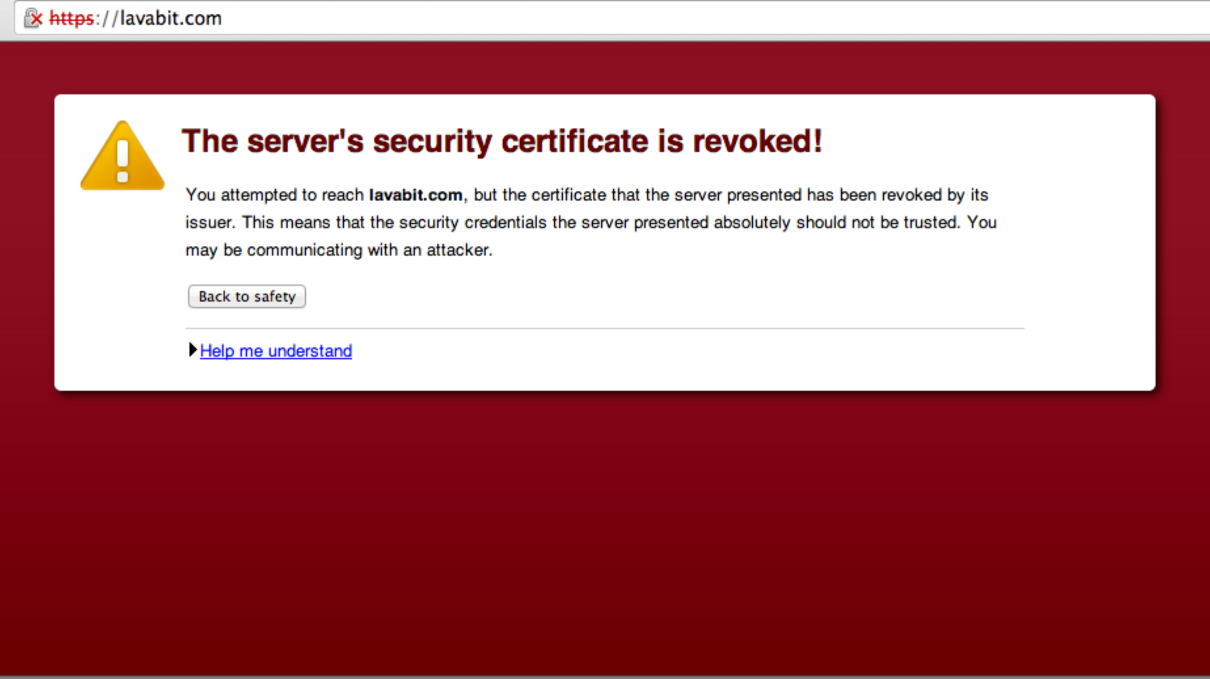
Site information

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.



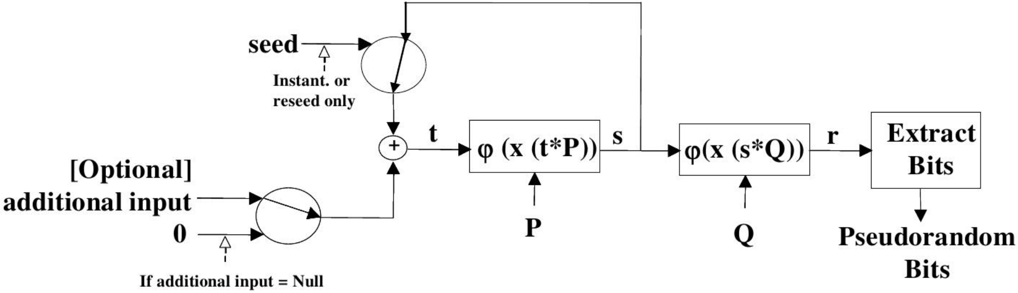
The server's security certificate is revoked!

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.

We reiterate that our review is circumscribed by the arguments that Lavabit raised below and in this Court. We take this narrow course because an appellate court is not a freestanding open forum for the discussion of esoteric hypothetical questions. *See* *Swann v. Charlotte-Mecklenburg Bd.*

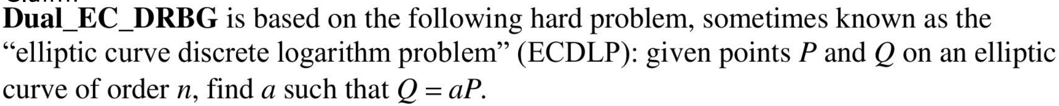
DUAL_EC RNG: history part I

Earliest public source (?) June 2004, draft of ANSI X9.82:



φ gives all but the top 16 bits ⇒ 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 $Q = aP$.

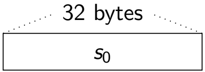
DUAL_EC RNG: common public history part II

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 SP800-90 gets appendix about choosing points verifiably at random,
  but requires use of standardized P, Q for FIPS-140 validation.

Dual EC

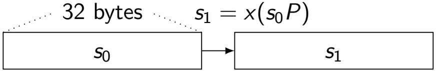
Points Q and P on an elliptic curve.



Graphic based on work by Ruben Niederhagen.

Dual EC

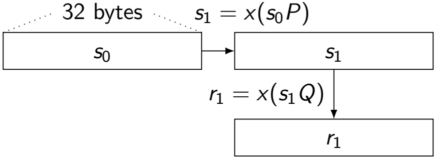
Points Q and P on an elliptic curve.



Graphic based on work by Ruben Niederhagen.

Dual EC

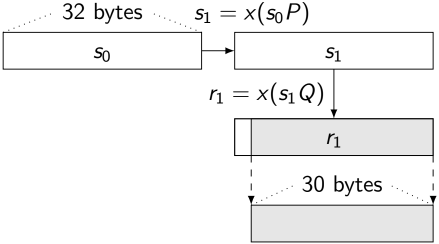
Points Q and P on an elliptic curve.



Graphic based on work by Ruben Niederhagen.

Dual EC

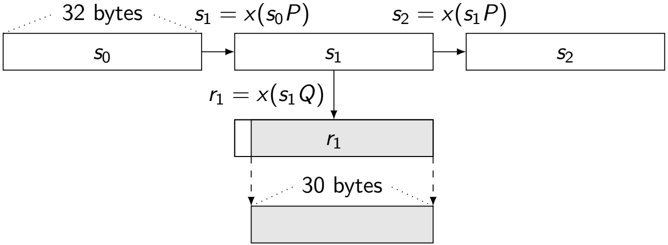
Points Q and P on an elliptic curve.



Graphic based on work by Ruben Niederhagen.

Dual EC

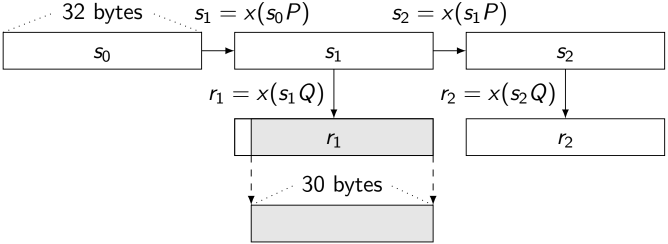
Points Q and P on an elliptic curve.



Graphic based on work by Ruben Niederhagen.

Dual EC

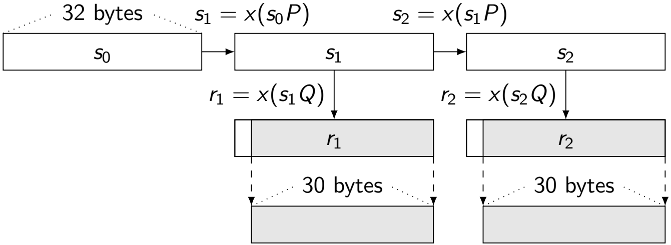
Points Q and P on an elliptic curve.



Graphic based on work by Ruben Niederhagen.

Dual EC

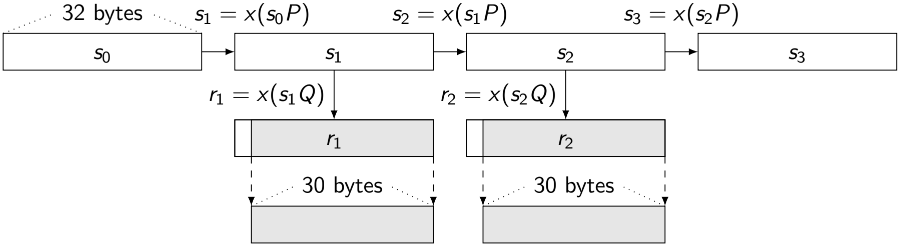
Points Q and P on an elliptic curve.



Graphic based on work by Ruben Niederhagen.

Dual EC

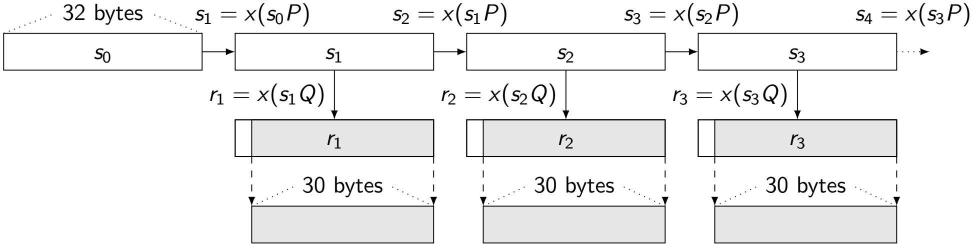
Points Q and P on an elliptic curve.



Graphic based on work by Ruben Niederhagen.

Dual EC

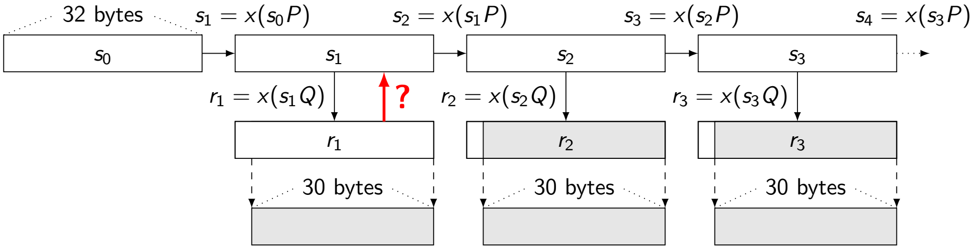
Points Q and P on an elliptic curve.



Graphic based on work by Ruben Niederhagen.

Dual EC

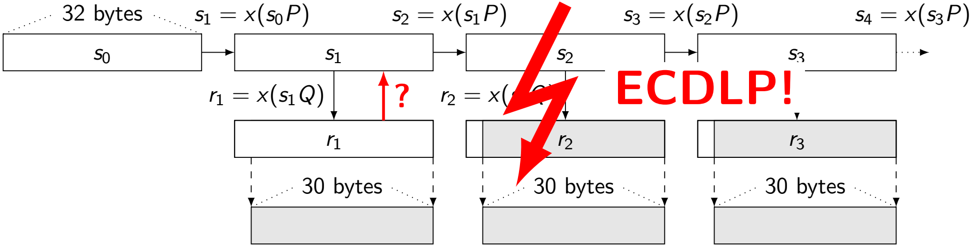
Points Q and P on an elliptic curve.



Graphic based on work by Ruben Niederhagen.

Dual EC

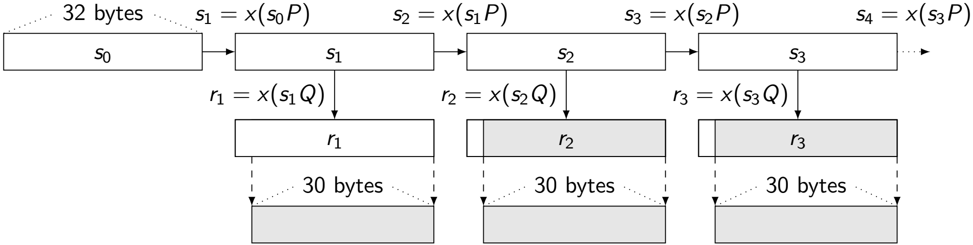
Points Q and $P = dQ$ on an elliptic curve.



Graphic based on work by Ruben Niederhagen.

Basic attack

Points Q and $P = dQ$ on an elliptic curve.

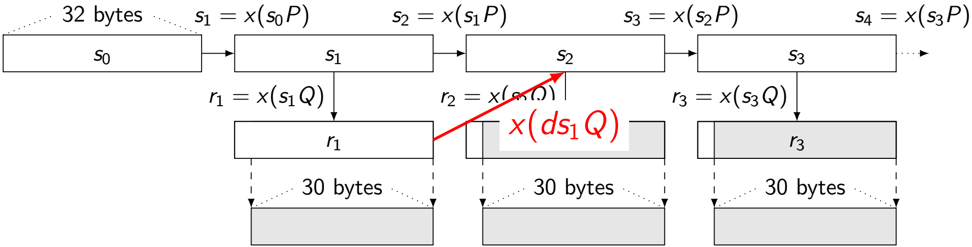


Graphic based on work by Ruben Niederhagen.

Basic attack

Points Q and $P = dQ$ on an elliptic curve.

$$s_2 = x(s_1 P) = x(s_1 dQ)$$

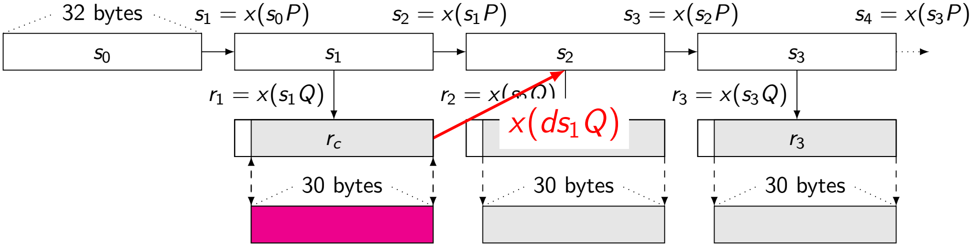


Graphic based on work by Ruben Niederhagen.

Basic attack

Points Q and $P = dQ$ on an elliptic curve.

$$s_2 = x(s_1 P) = x(s_1 dQ)$$

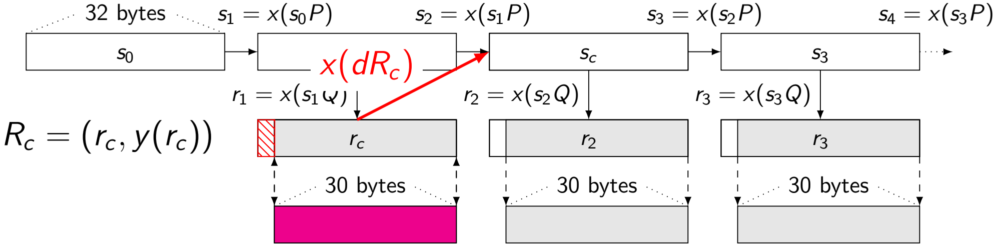


Graphic based on work by Ruben Niederhagen.

Basic attack

Points Q and $P = dQ$ on an elliptic curve.

$$s_2 = x(s_1 P) = x(s_1 dQ)$$

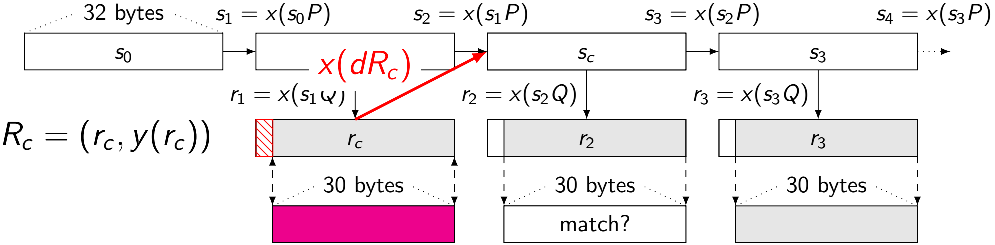


Graphic based on work by Ruben Niederhagen.

Basic attack

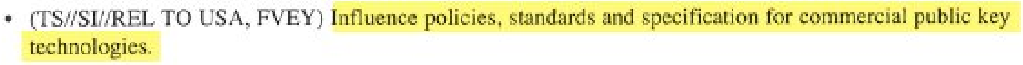
Points Q and $P = dQ$ on an elliptic curve.

$$s_2 = x(s_1 P) = x(s_1 dQ)$$



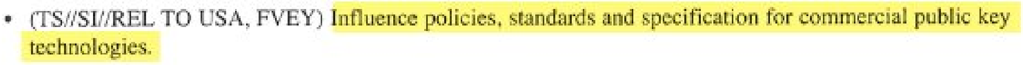
Graphic based on work by Ruben Niederhagen.

September 2013: NSA Bullrun program



- (TS//SI//REL TO USA, FVEY) Influence policies, standards and specification for commercial public key technologies.

September 2013: NSA Bullrun program

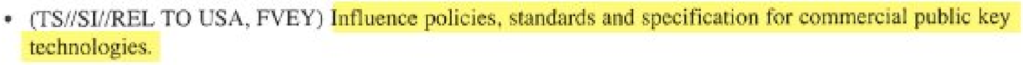


- (TS//SI//REL TO USA, FVEY) Influence policies, standards and specification for commercial public key technologies.

NYT:

> *the NSA had inserted a back door into a 2006 standard adopted by NIST [..] called the Dual EC DRBG standard.*

September 2013: NSA Bullrun program



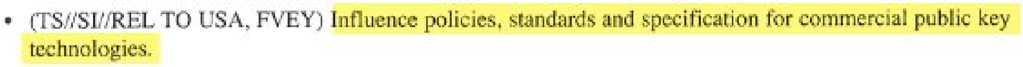
- (TS//SI//REL TO USA, FVEY) Influence policies, standards and specification for commercial public key technologies.

NYT:

> *the NSA had inserted a back door into a 2006 standard adopted by NIST [..] called the Dual EC DRBG standard.*

. . . but surely nobody uses that!?!

September 2013: NSA Bullrun program



- (TS//SI//REL TO USA, FVEY) Influence policies, standards and specification for commercial public key technologies.

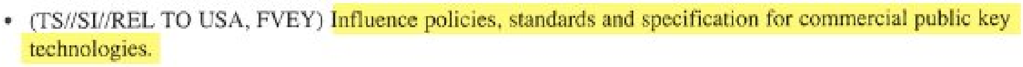
NYT:

> *the NSA had inserted a back door into a 2006 standard adopted by NIST [..] called the Dual EC DRBG standard.*

. . . but surely nobody uses that!?!

NIST's DRBG Validation List: more than 70 validations of Dual_EC_DRBG; RSA's BSAFE has Dual_EC_DRBG enabled as default.

September 2013: NSA Bullrun program



- (TS//SI//REL TO USA, FVEY) Influence policies, standards and specification for commercial public key technologies.

NYT:

> *the NSA had inserted a back door into a 2006 standard adopted by NIST [..] called the Dual EC DRBG standard.*

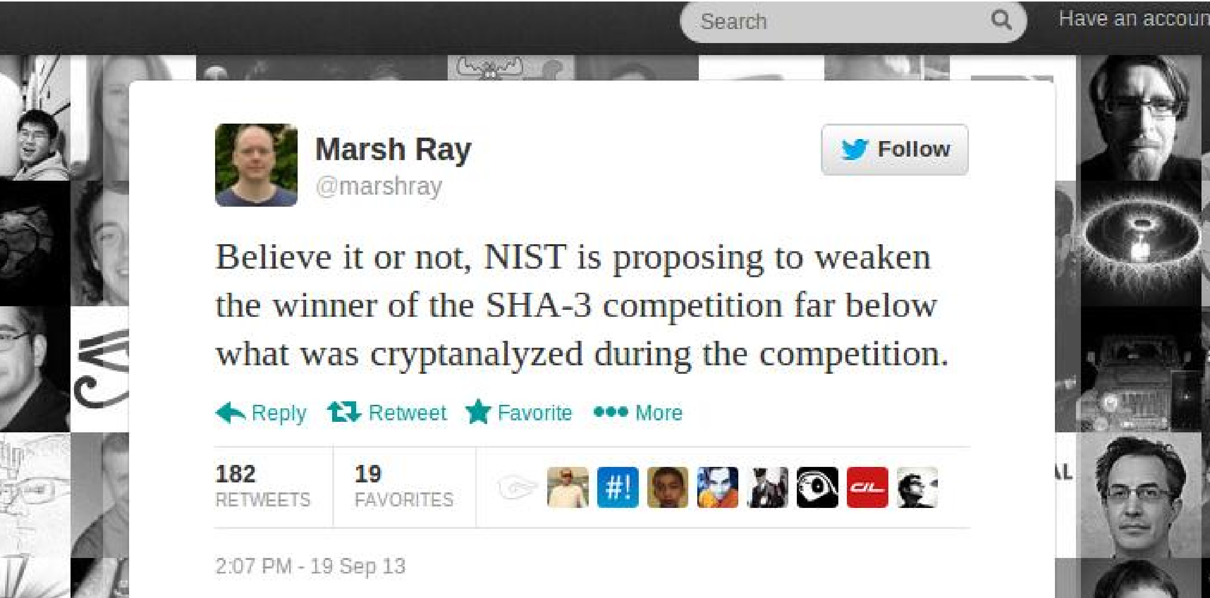
. . . but surely nobody uses that!?!

NIST's DRBG Validation List: more than 70 validations of Dual_EC_DRBG; RSA's BSAFE has Dual_EC_DRBG enabled as default.

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

21 April 2014 NIST removes Dual EC from the standard.

September 2013: SHA-3 controversy erupts



Marsh Ray
@marshray

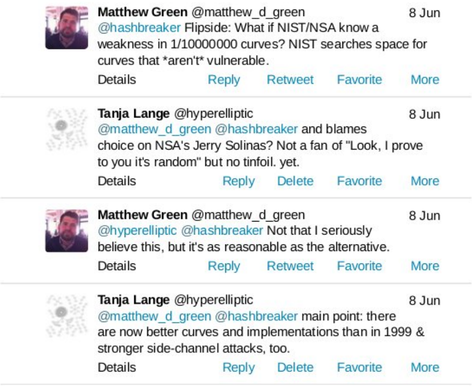
Believe it or not, NIST is proposing to weaken the winner of the SHA-3 competition far below what was cryptanalyzed during the competition.

182 RETWEETS 19 FAVORITES

2:07 PM - 19 Sep 13

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.

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.

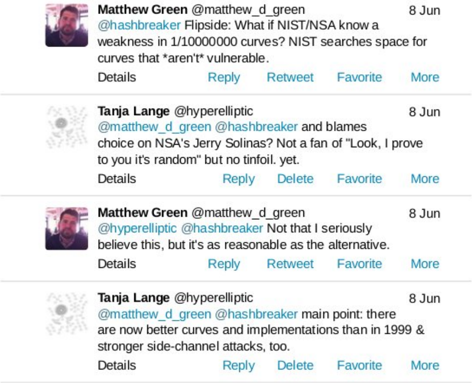
Matthew Green @matthew_d_green 8 Jun
@hyperelliptic @hashbreaker Not that I seriously believe this, but it's as reasonable as the alternative.

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.

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

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.

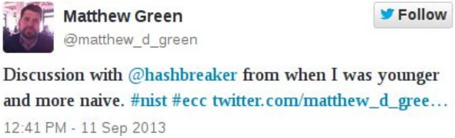
Tanja Lange @hyperelliptic 8 Jun
@matthew_d_green @hashbreaker and blames choice on NSA's Jerry Solinas? Not a fan of "Look, I prove to you it's random" but no tinfoil. yet.

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

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.

Green: “Flipside: What if NIST/NSA know a weakness in 1/10000000 curves? NIST searches space for curves that *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 Curve41417.

SafeCurves: choosing safe curves for elliptic-curve cryptography

All known security criteria for elliptic curves, machine verified.

Elligator: undetectable curve points.

New Curve41417.

Also: can the curve be backdoored?

http://safecurves.cr.yp.to

SafeCurves: choosing safe curves for elliptic-curve cryptography

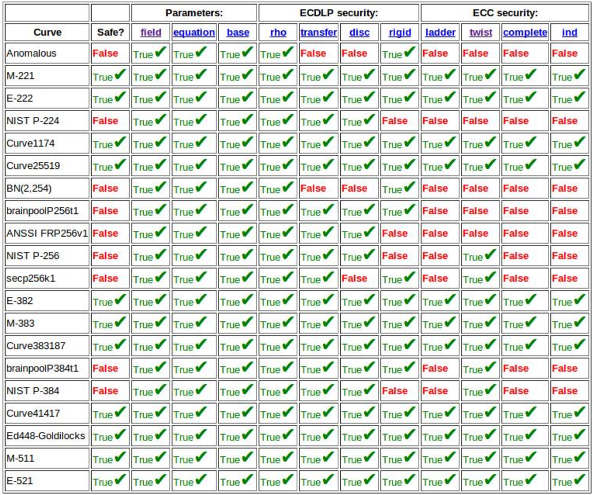
All known security criteria for elliptic curves, machine verified.

Elligator: undetectable curve points.

New Curve41417.

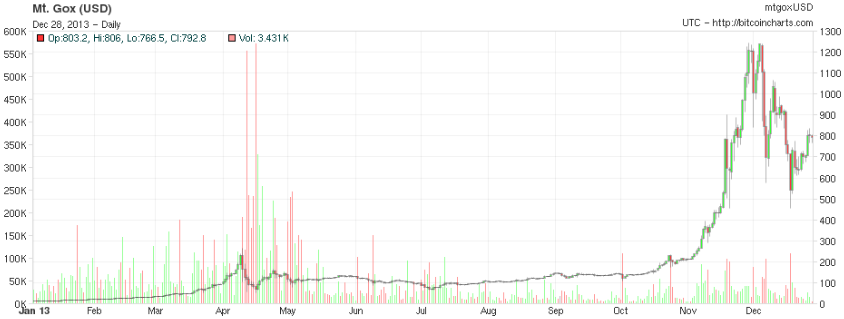
Also: can the curve be backdoored?

http://safecurves.cr.yp.to



		Parameters:			ECDLP security:				ECC security:			
Curve	Safe?	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
ANSSI FRP256v1	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
E-382	True ✓	True ✓	True ✓	True ✓	True ✓	True ✓	True ✓	True ✓	True ✓	True ✓	True ✓	True ✓
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
Curve41417	True ✓	True ✓	True ✓	True ✓	True ✓	True ✓	True ✓	True ✓	True ✓	True ✓	True ✓	True ✓
Ed448-Goldilocks	True ✓	True ✓	True ✓	True ✓	True ✓	True ✓	True ✓	True ✓	True ✓	True ✓	True ✓	True ✓
M-511	True ✓	True ✓	True ✓	True ✓	True ✓	True ✓	True ✓	True ✓	True ✓	True ✓	True ✓	True ✓
E-521	True ✓	True ✓	True ✓	True ✓	True ✓	True ✓	True ✓	True ✓	True ✓	True ✓	True ✓	True ✓

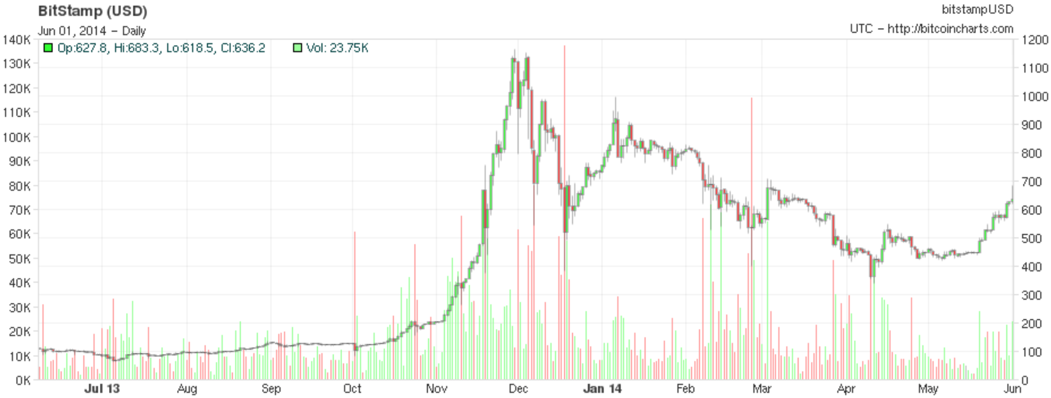
Bitcoin goes mainstream, bringing ECDSA with it



August 2013: Android Java RNG vulnerability blamed for bitcoin thefts

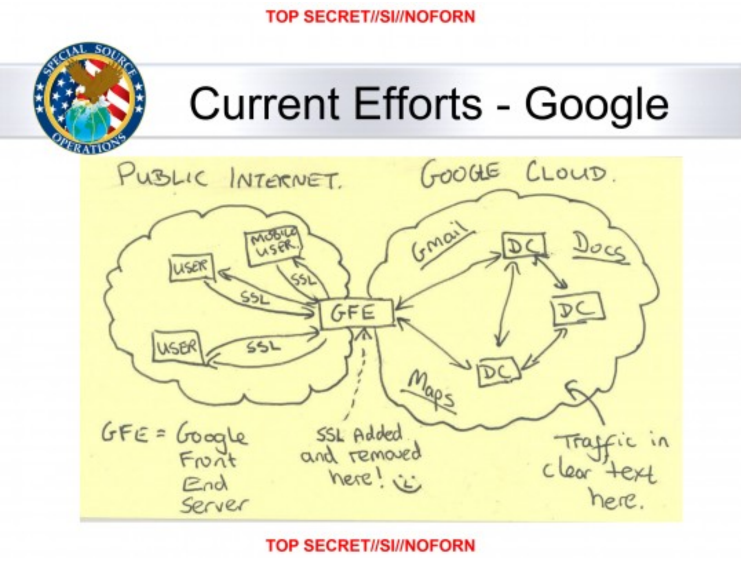
`1HKywxiL4JziqXrzLKhmB6a74ma6kxbSDj` has stolen 59 bitcoin from addresses using repeated ECDSA signature randomness.

Bitcoin goes mainstream, bringing ECDSA with it



February 7 2014: Mt. Gox suspends trading, possibly because of transaction malleability.

October 2013: MUSCULAR



TOP SECRET//SI//NOFORN

Current Efforts - Google

GFE = Google Front End Server

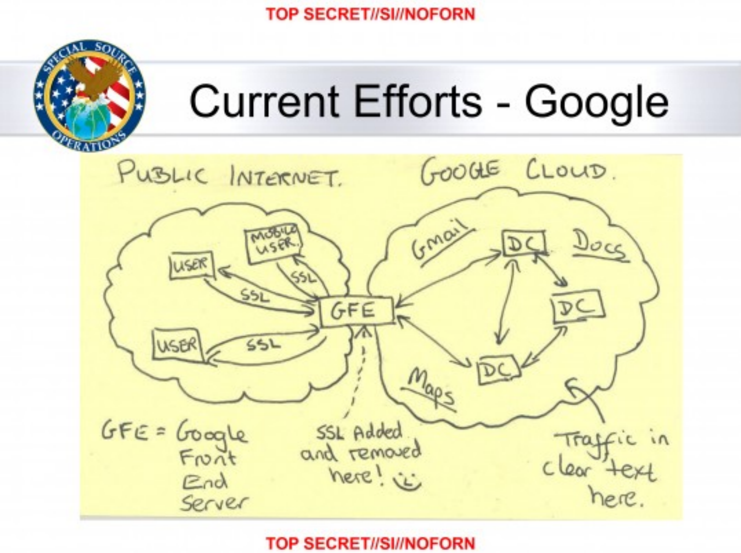
SSL Added and removed here! :)

Traffic in clear text here.

TOP SECRET//SI//NOFORN

Official Google statement:
"We are outraged"

October 2013: MUSCULAR



TOP SECRET//SI//NOFORN

Current Efforts - Google

GFE = Google Front End Server

SSL Added and removed here! ☺

Traffic in clear text here.

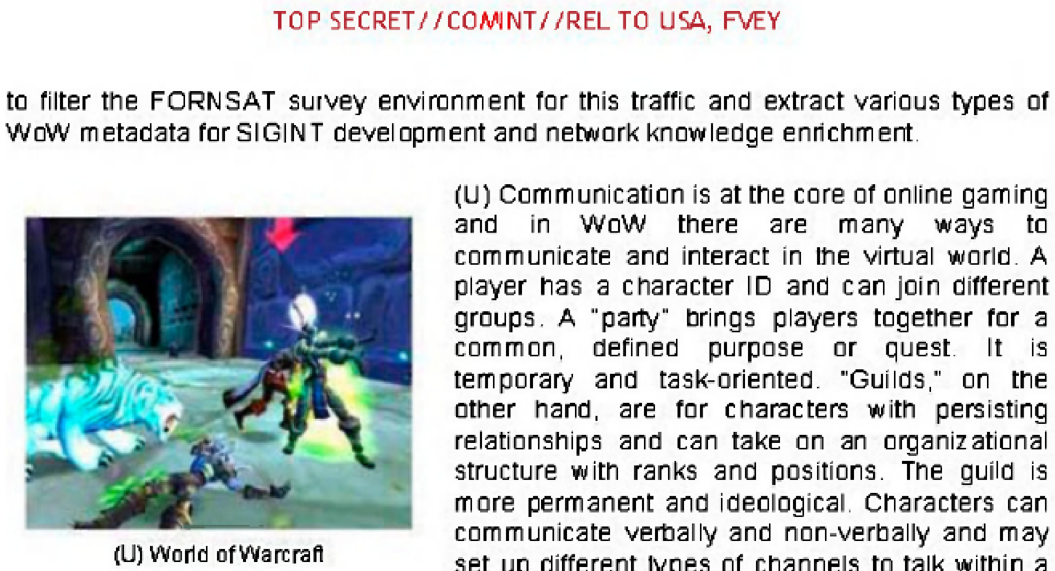
TOP SECRET//SI//NOFORN

Official Google statement:
“We are outraged”

Unofficial Google statement:
“Fuck these guys.”

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

Meanwhile at the NSA . . .



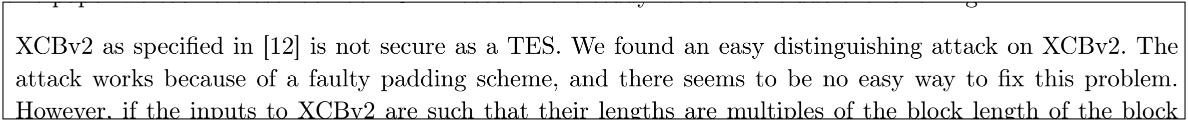
TOP SECRET//COMINT//REL TO USA, FVEY

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

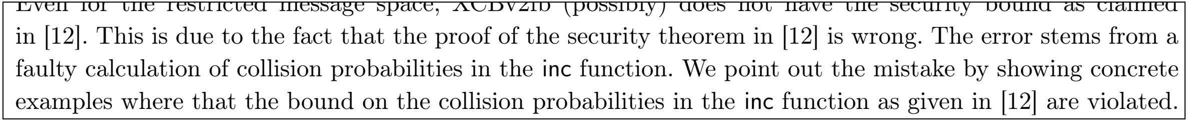
(U) World of Warcraft

(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



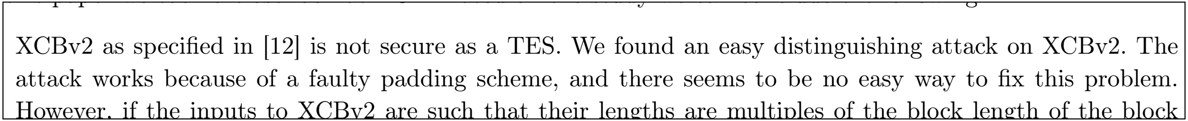
XCBv2 as specified in [12] is not secure as a TES. We found an easy distinguishing attack on XCBv2. The attack works because of a faulty padding scheme, and there seems to be no easy way to fix this problem. However, if the inputs to XCBv2 are such that their lengths are multiples of the block length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



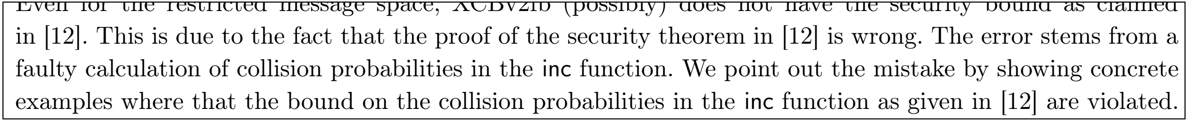
Even for the restricted message space, XCBv2fb (possibly) does not have the security bound as claimed in [12]. This is due to the fact that the proof of the security theorem in [12] is wrong. The error stems from a faulty calculation of collision probabilities in the inc function. We point out the mistake by showing concrete examples where that the bound on the collision probabilities in the inc function as given in [12] are violated.

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

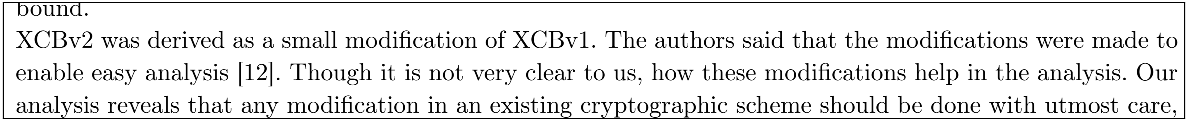
December 2013: trouble with XCB disk-encryption standard



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, XCBv2fb (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

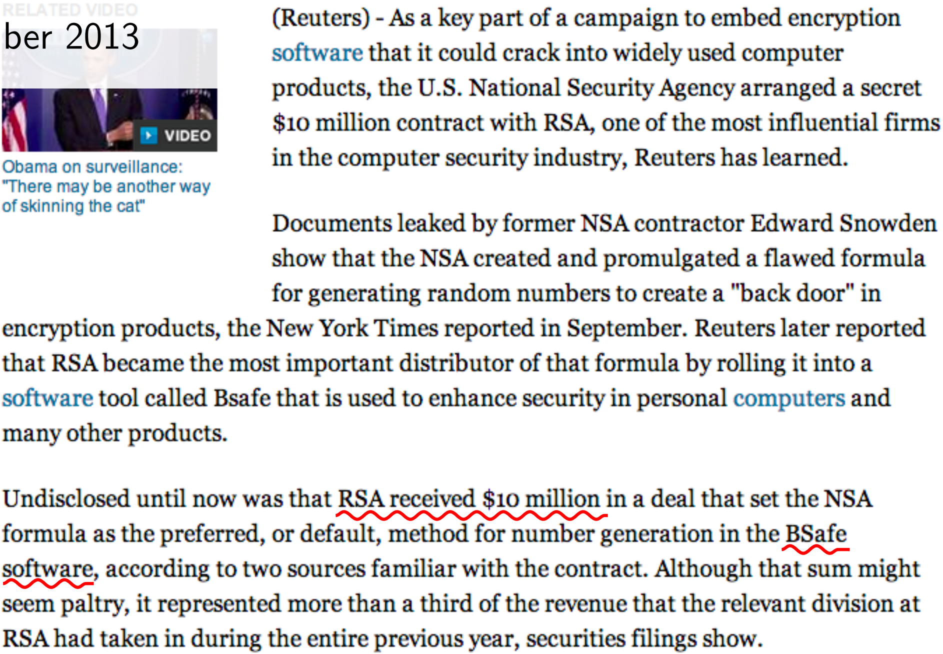
December 2013: Obama's NSA review panel report

and hence that some commercially available software is not trustworthy today.

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.

December 2013

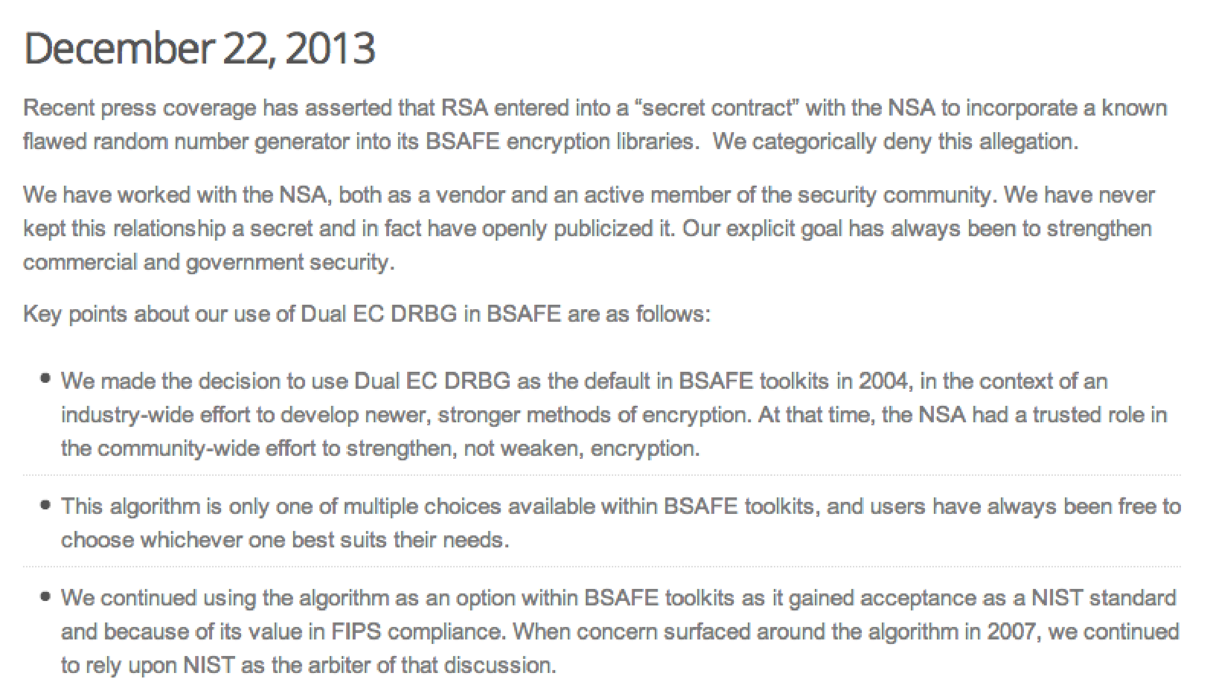


Obama on surveillance: "There may be another way of skinning the cat"

(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.

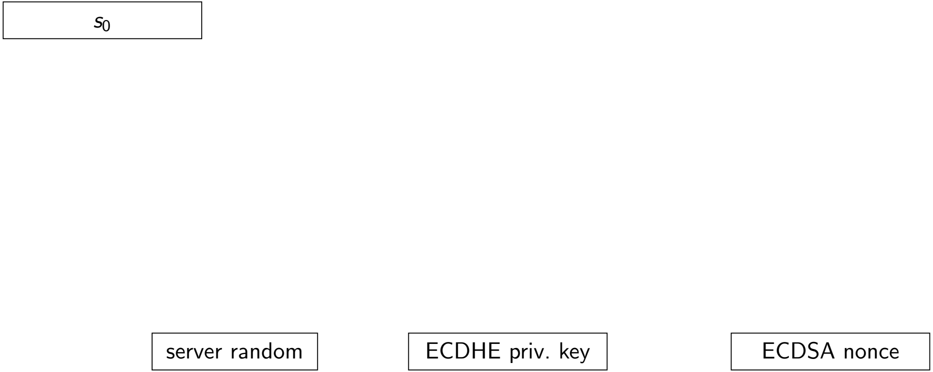
Attacking Dual EC in TLS – Example: BSAFE-Java

server random

ECDHE priv. key

ECDSA nonce

Attacking Dual EC in TLS – Example: BSAFE-Java



s_0

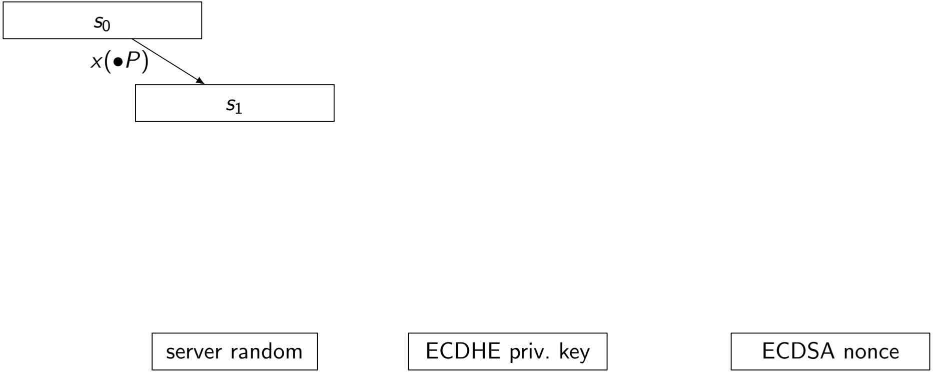
server random

ECDHE priv. key

ECDSA nonce

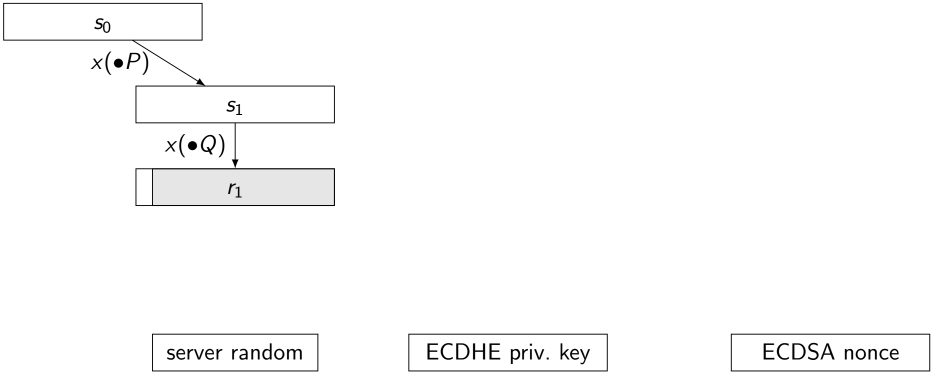
Graphics: Ruben Niederhagen.

Attacking Dual EC in TLS – Example: BSAFE-Java



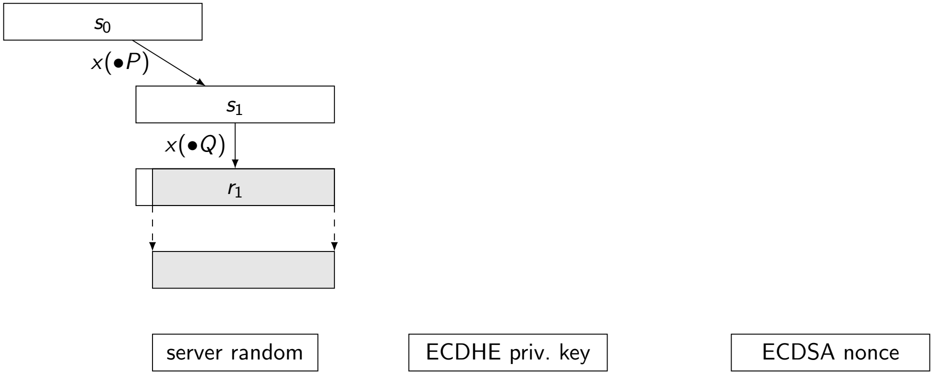
Graphics: Ruben Niederhagen.

Attacking Dual EC in TLS – Example: BSAFE-Java



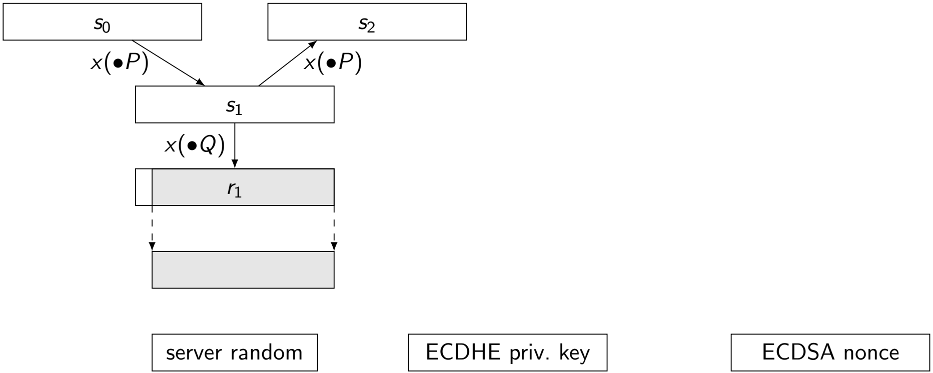
Graphics: Ruben Niederhagen.

Attacking Dual EC in TLS – Example: BSAFE-Java



Graphics: Ruben Niederhagen.

Attacking Dual EC in TLS – Example: BSAFE-Java



s_0

s_2

$x(\bullet P)$

$x(\bullet P)$

s_1

$x(\bullet Q)$

r_1

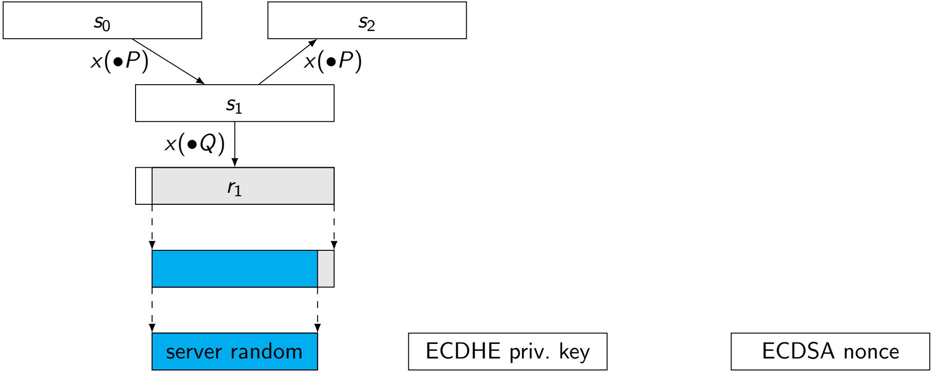
server random

ECDHE priv. key

ECDSA nonce

Graphics: Ruben Niederhagen.

Attacking Dual EC in TLS – Example: BSAFE-Java



s_0

s_2

$x(\bullet P)$

$x(\bullet P)$

s_1

$x(\bullet Q)$

r_1

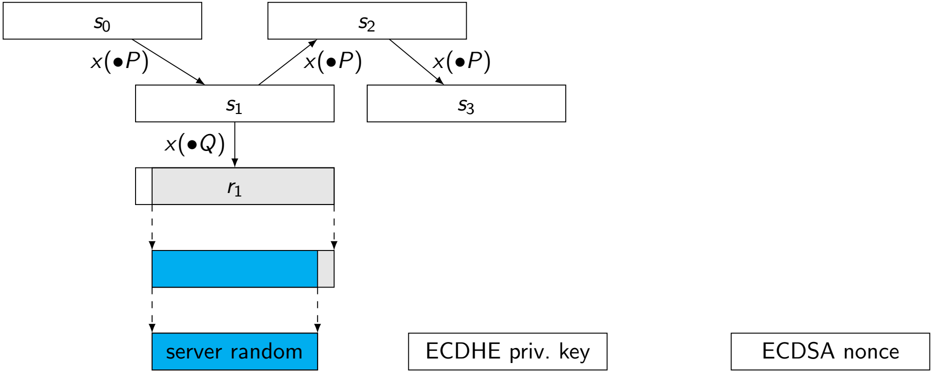
server random

ECDHE priv. key

ECDSA nonce

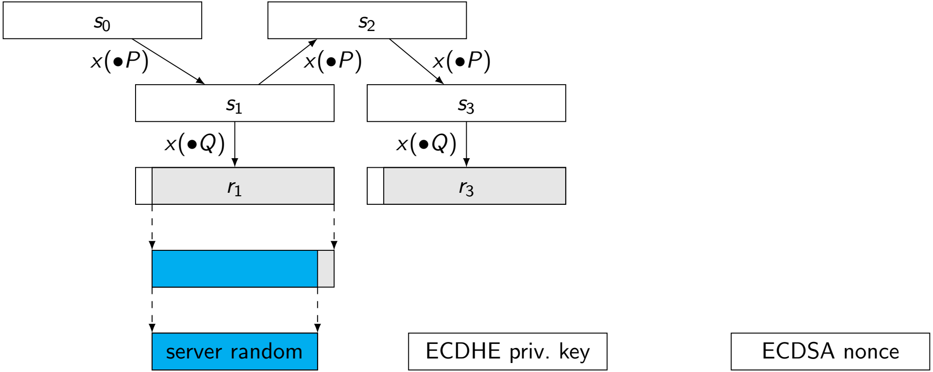
Graphics: Ruben Niederhagen.

Attacking Dual EC in TLS – Example: BSAFE-Java



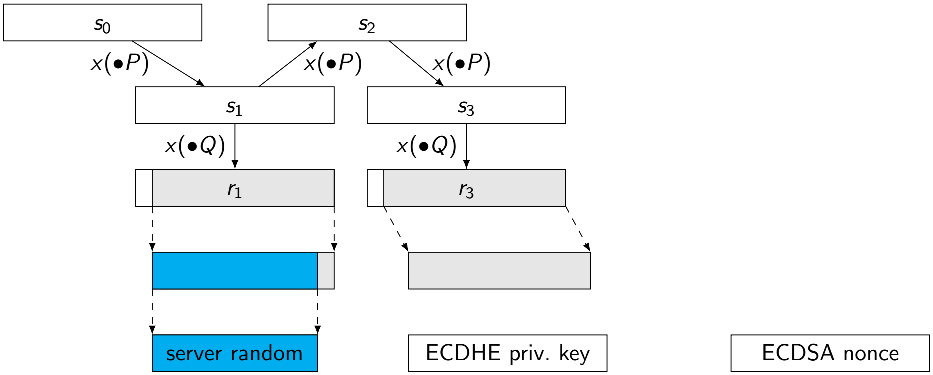
Graphics: Ruben Niederhagen.

Attacking Dual EC in TLS – Example: BSAFE-Java



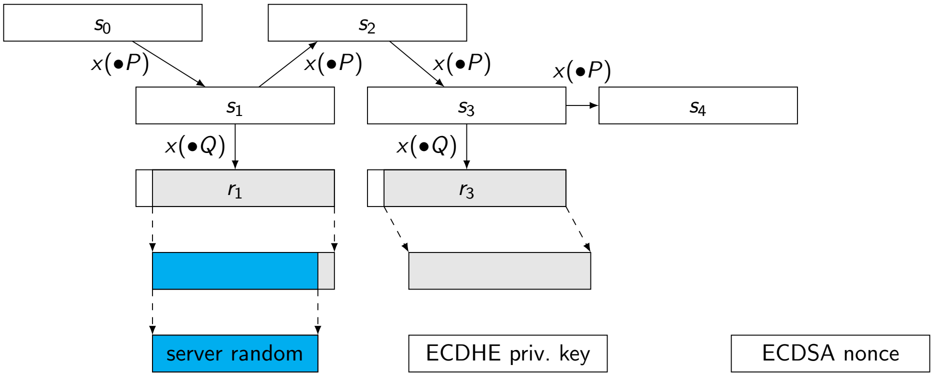
Graphics: Ruben Niederhagen.

Attacking Dual EC in TLS – Example: BSAFE-Java



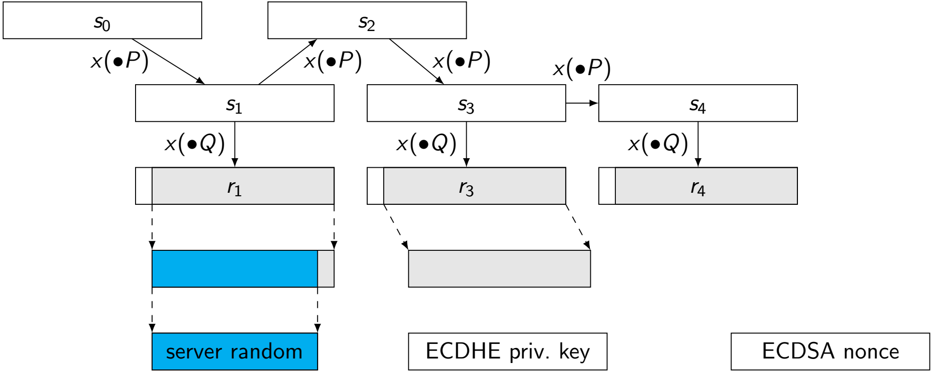
Graphics: Ruben Niederhagen.

Attacking Dual EC in TLS – Example: BSAFE-Java



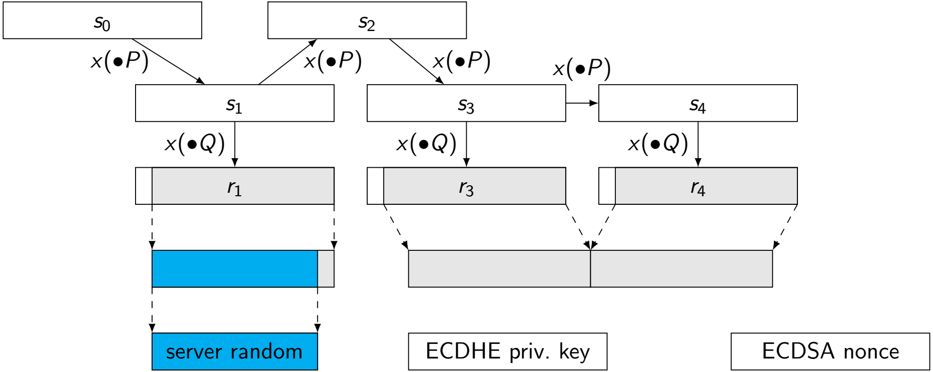
Graphics: Ruben Niederhagen.

Attacking Dual EC in TLS – Example: BSAFE-Java



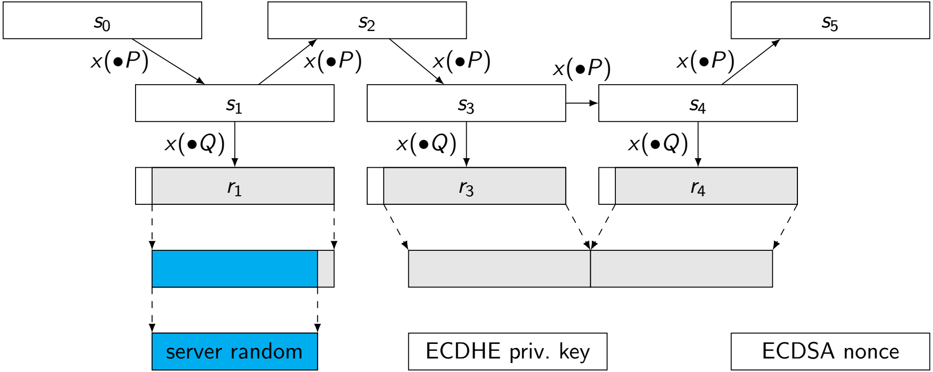
Graphics: Ruben Niederhagen.

Attacking Dual EC in TLS – Example: BSAFE-Java



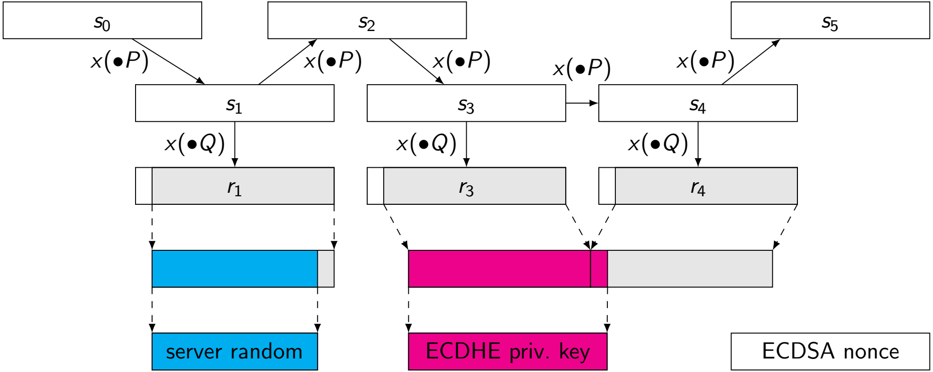
Graphics: Ruben Niederhagen.

Attacking Dual EC in TLS – Example: BSAFE-Java



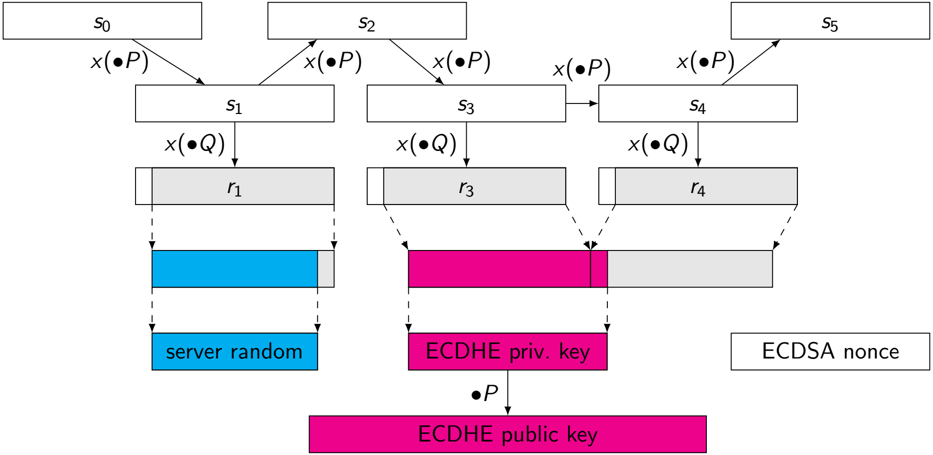
Graphics: Ruben Niederhagen.

Attacking Dual EC in TLS – Example: BSAFE-Java



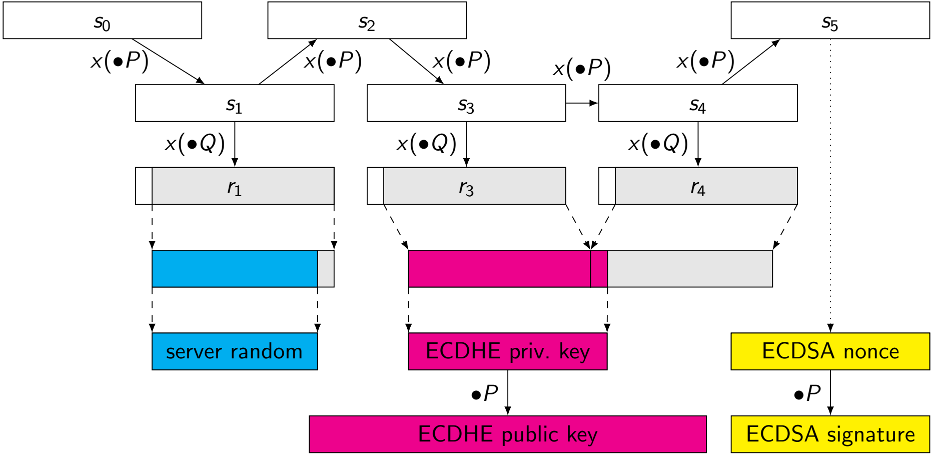
Graphics: Ruben Niederhagen.

Attacking Dual EC in TLS – Example: BSAFE-Java



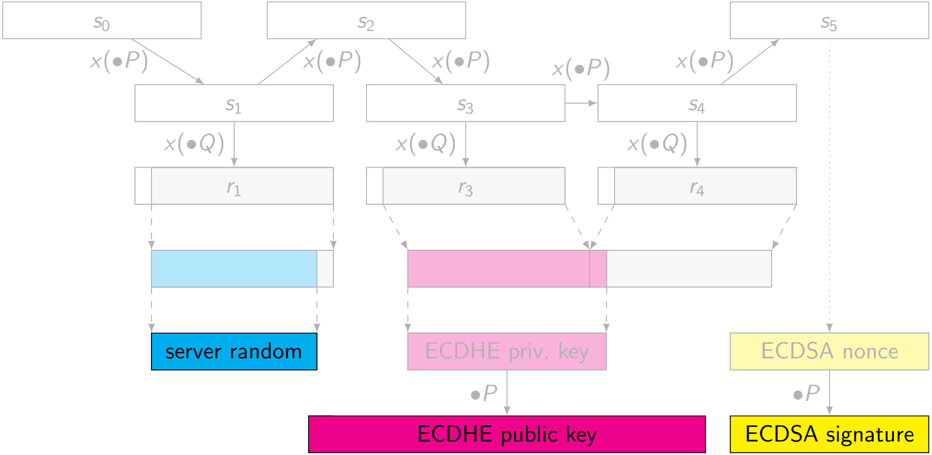
Graphics: Ruben Niederhagen.

Attacking Dual EC in TLS – Example: BSAFE-Java



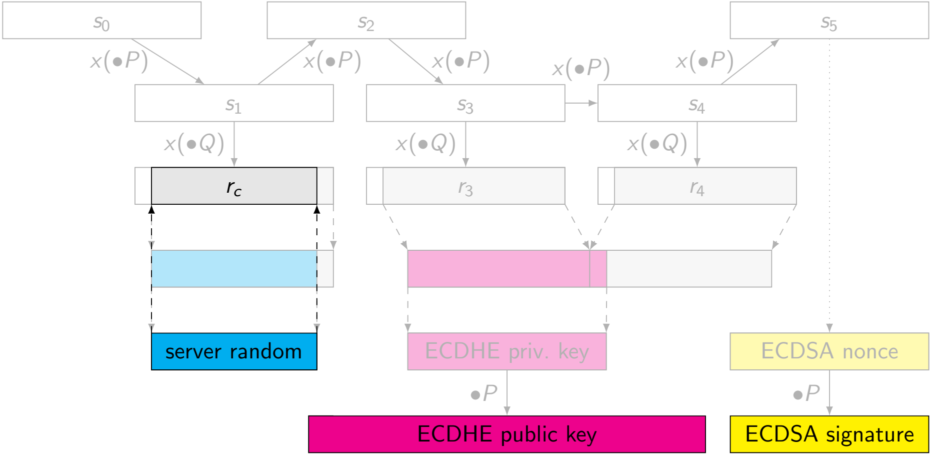
Graphics: Ruben Niederhagen.

Attacking Dual EC in TLS – Example: BSAFE-Java



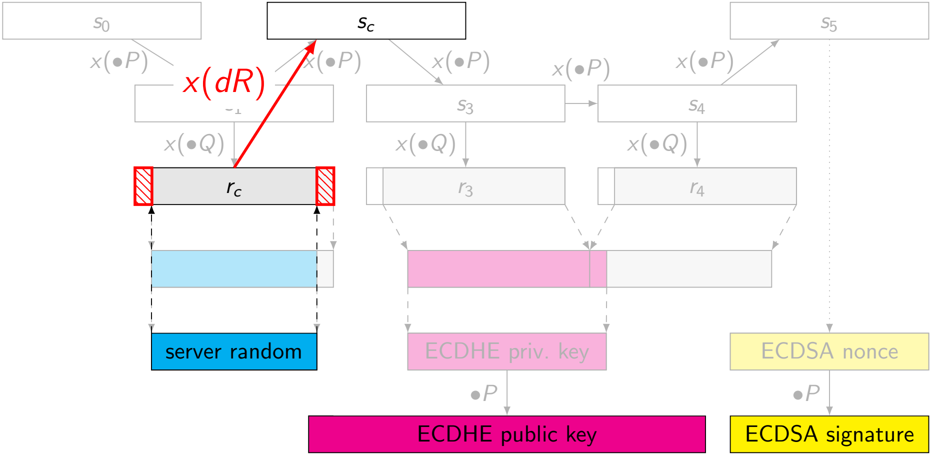
Graphics: Ruben Niederhagen.

Attacking Dual EC in TLS – Example: BSAFE-Java



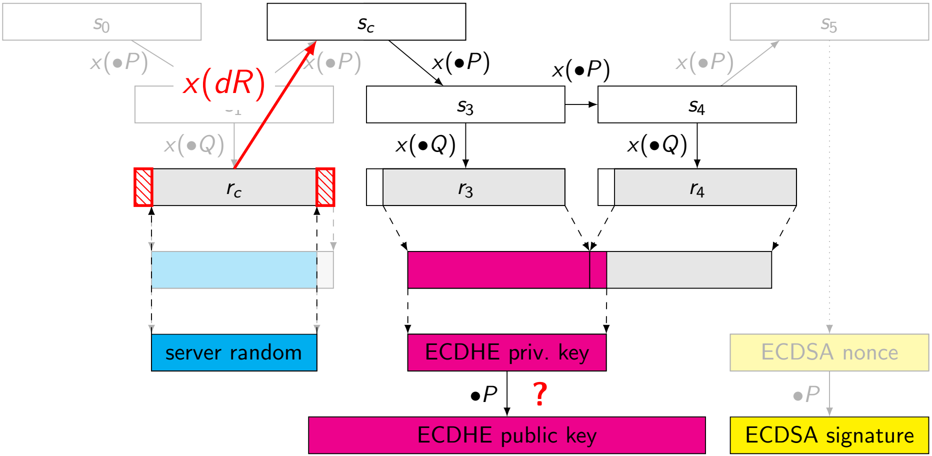
Graphics: Ruben Niederhagen.

Attacking Dual EC in TLS – Example: BSAFE-Java



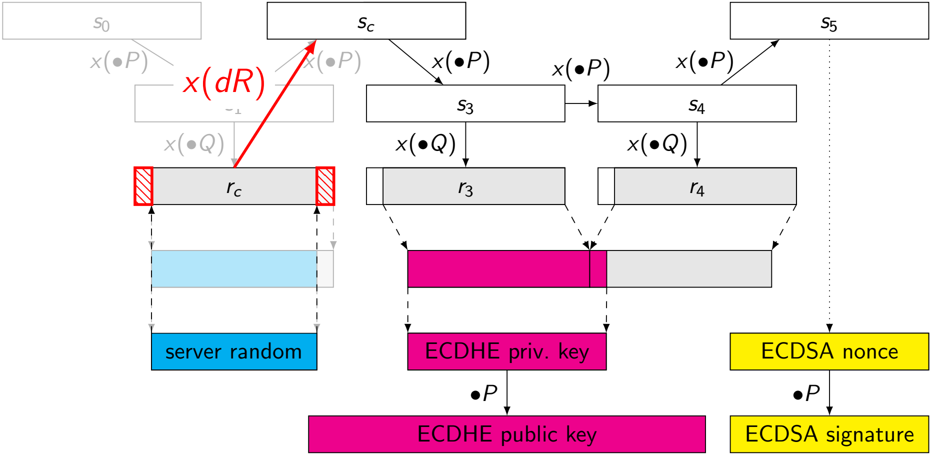
Graphics: Ruben Niederhagen.

Attacking Dual EC in TLS – Example: BSAFE-Java



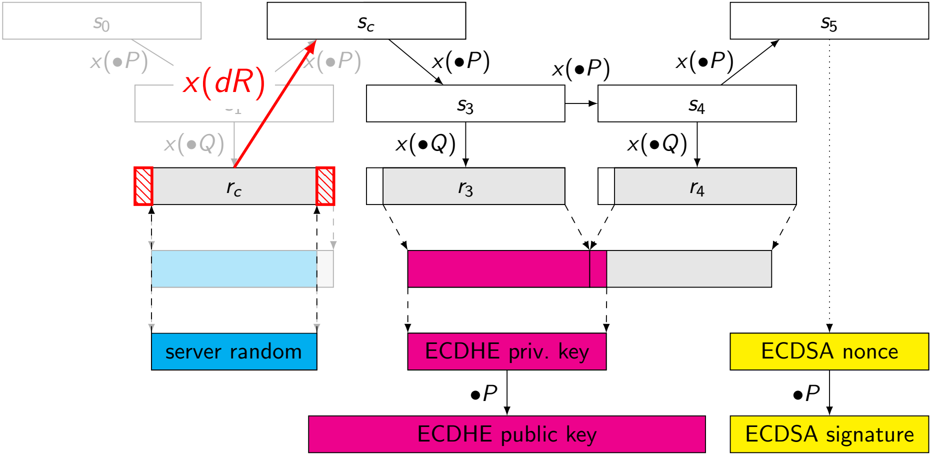
Graphics: Ruben Niederhagen.

Attacking Dual EC in TLS – Example: BSAFE-Java



Graphics: Ruben Niederhagen.

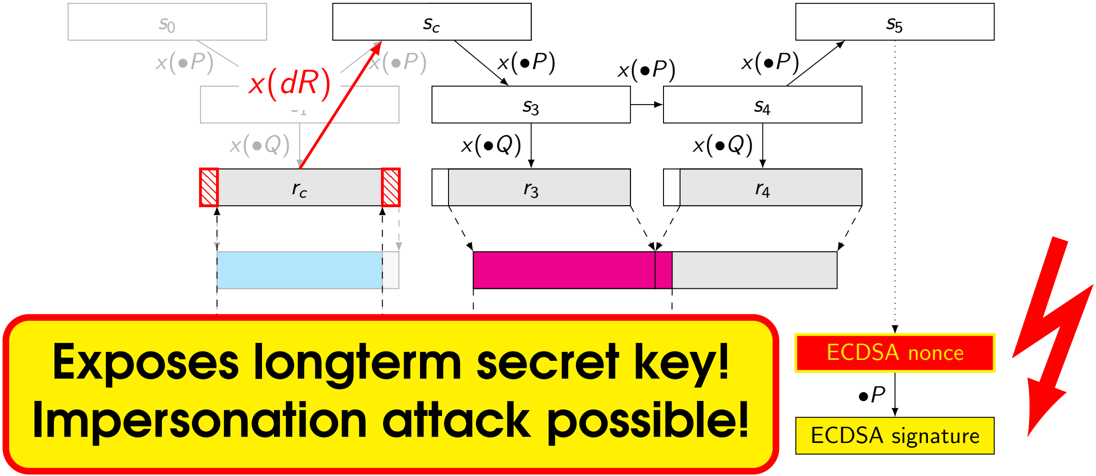
Attacking Dual EC in TLS – Example: BSAFE-Java



Graphics: Ruben Niederhagen.

average cost: $2^{31}(C_v + 5C_f)$

Attacking Dual EC in TLS – Example: BSAFE-Java

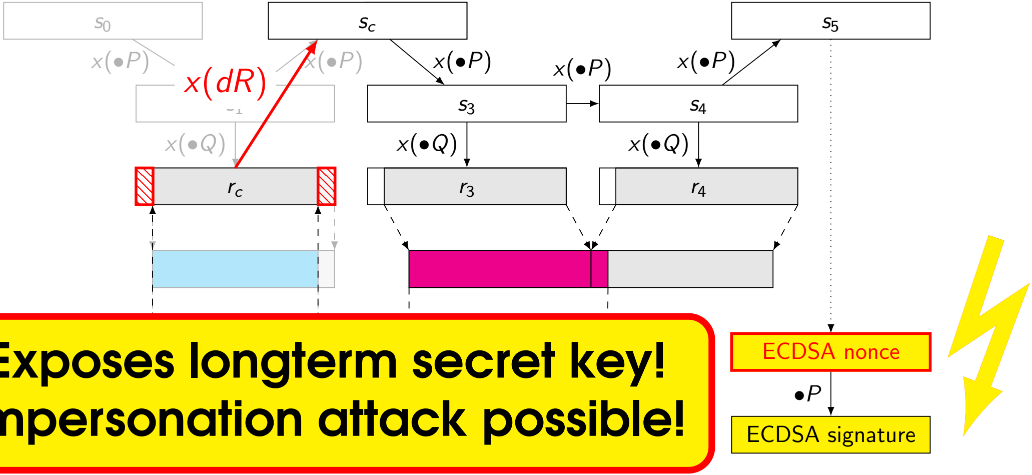


Exposes longterm secret key!
Impersonation attack possible!

Graphics: Ruben Niederhagen.

average cost: $2^{31}(C_v + 5C_f)$

Attacking Dual EC in TLS – Example: BSAFE-Java



Exposes longterm secret key!
Impersonation attack possible!

Graphics: Ruben Niederhagen.

average cost: $2^{31}(C_v + 5C_f)$

Timings

Attack	Intel Xeon Reference System		16-CPU AMD Cluster
	2^{22} Candidates (s)	Expected Runtime (min)	Total Runtime (min)
BSAFE-C v1.1	–	0.26	0.04*
BSAFE-Java v1.1	75.08*	641	63.96*
SChannel I	72.58*	619	62.97*
SChannel II	62.79*	1,760	182.64*
OpenSSL-fixed I	–	0.04	0.02*
OpenSSL-fixed II	–	707	83.32*
OpenSSL-fixed III	–	$2^k \cdot 707$	$2^k \cdot 83.32$

*measured

See much more at http://projectbullrun.org/dual-ec/.

Details on Intel's RNG

Details on Intel’s RNG

[7] D. J. Johnston, ”Mircoarchitecture Specification (MAS) for PP-DRNG,” Intel Corporation (unpublished), V1.4, 2009.

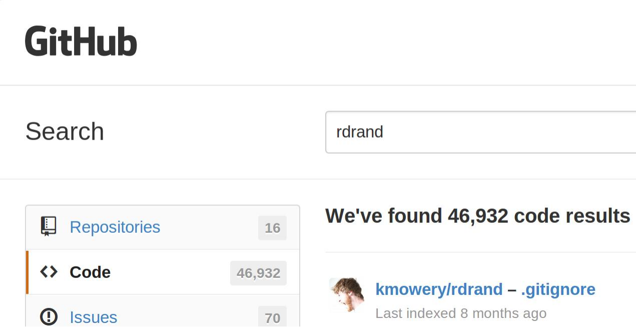
[8] C. E. Dike, ”3 Gbps Binary RNG Entropy Source,” Intel Corporation (unpublished), 2011.

[9] C. E. Dike and S. Gueron, ”Digital Symmetric Random Number Generator Mathematics,” Intel Corporation (unpublished), 2009.

(References from “Analysis of Intel’s Ivy Bridge Digital Random Number Generator Prepared for Intel” by Mike Hamburg, Paul Kocher, and Mark E. Marson. Cryptography Research, Inc.)

Intel recommendations

David Johnston (RDRAND designer), 2012: "It provides both the entropy, the seeds and the PRNG in hardware. So you can replace the whole shebang and eliminate software PRNGs. Just use the output of the RDRAND instruction wherever you need a random number."



GitHub

Search

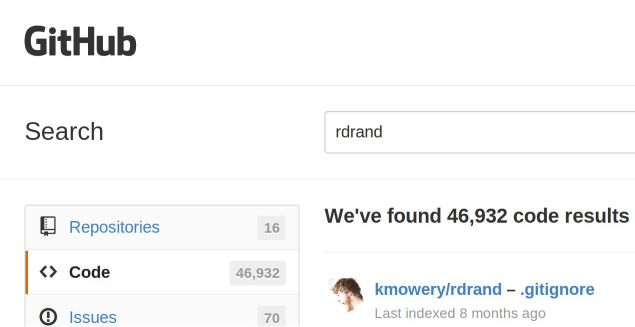
rdrand

We've found 46,932 code results

kmowery/rdrand – .gitignore
Last indexed 8 months ago

Intel recommendations

David Johnston (RDRAND designer), 2012: “It provides both the entropy, the seeds and the PRNG in hardware. So you can replace the whole shebang and eliminate software PRNGs. Just use the output of the RDRAND instruction wherever you need a random number.”



GitHub

Search

rdrand

Repositories 16

Code 46,932

Issues 70

We've found 46,932 code results

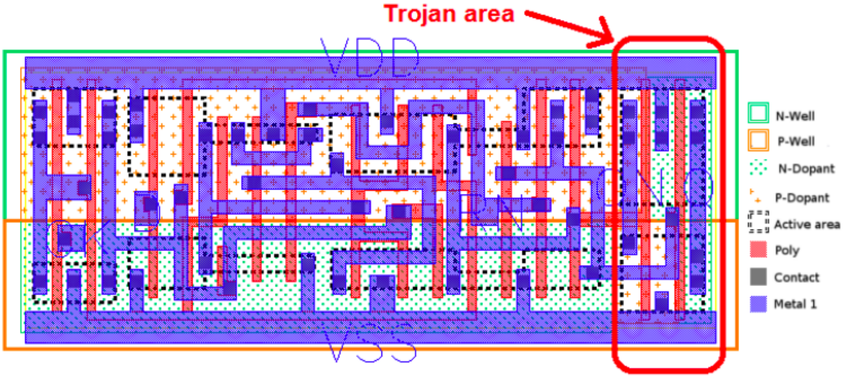
kmowery/rdrand – .gitignore

Last indexed 8 months ago

Snowden at SXSW: “*. . . we know that these encryption algorithms we are using today work; typically it is the random number generators that are attacked as opposed to the encryption algorithms themselves.*”

Scary Paper: *Stealthy Dopant-Level Hardware Trojans*

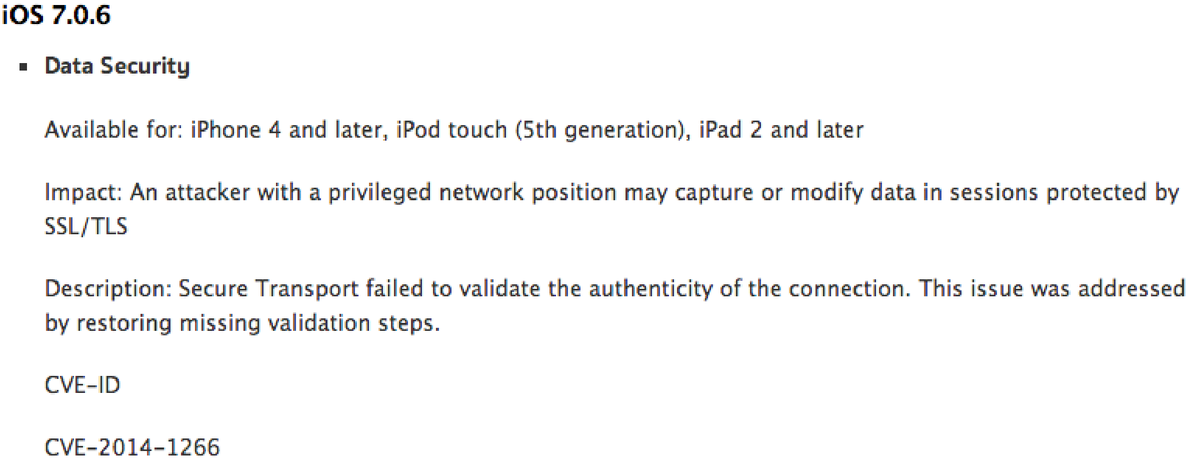
by Becker, Regazzoni, Paar, and Burleson, CHES 2013



Trojan area

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 $QN = GND$.

February 2014



iOS 7.0.6

- **Data Security**

  Available for: iPhone 4 and later, iPod touch (5th generation), iPad 2 and later

  Impact: An attacker with a privileged network position may capture or modify data in sessions protected by SSL/TLS

  Description: Secure Transport failed to validate the authenticity of the connection. This issue was addressed by restoring missing validation steps.

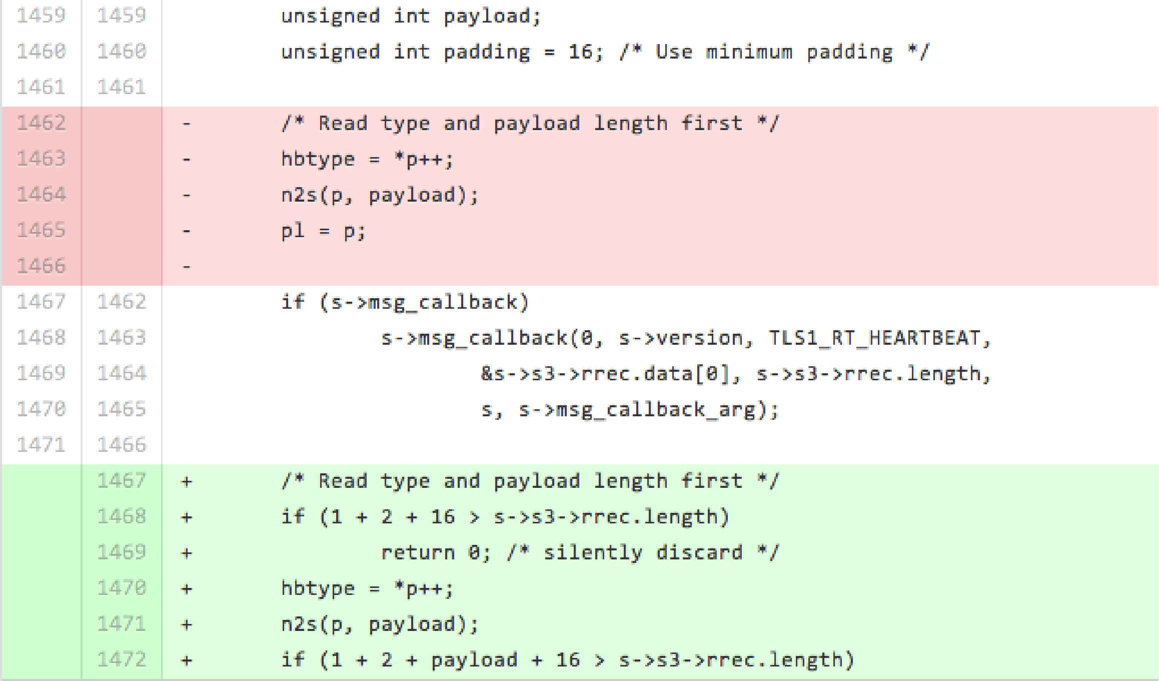
  CVE-ID

  CVE-2014-1266

```
static OSStatus
SSLVerifySignedServerKeyExchange(SSLContext *ctx, bool isRsa, SSLBuffer signedParams,
                                 uint8_t *signature, UInt16 signatureLen)
{
    OSStatus        err;
...
    if ((err = SSLHashSHA1.update(&hashCtx, &serverRandom)) != 0)
        goto fail;
    if ((err = SSLHashSHA1.update(&hashCtx, &signedParams)) != 0)
        goto fail;
        goto fail;
    if ((err = SSLHashSHA1.final(&hashCtx, &hashOut)) != 0)
        goto fail;

err = sslRawVerify(ctx, ...
                       signature,
                       signatureLen);
...
fail:
    SSLFreeBuffer(&signedHashes);
    SSLFreeBuffer(&hashCtx);
```

April 2014

```
OpenSSL Security Advisory [07 Apr 2014]
========================================

TLS heartbeat read overrun (CVE-2014-0160)
==========================================

A missing bounds check in the handling of the TLS heartbeat extension can be
used to reveal up to 64k of memory to a connected client or server.

Only 1.0.1 and 1.0.2-beta releases of OpenSSL are affected including
1.0.1f and 1.0.2-beta1.

Thanks for Neel Mehta of Google Security for discovering this bug and to
Adam Langley <agl@chromium.org> and Bodo Moeller <bmoeller@acm.org> for
preparing the fix.

Affected users should upgrade to OpenSSL 1.0.1g. Users unable to immediately
upgrade can alternatively recompile OpenSSL with -DOPENSSL_NO_HEARTBEATS.

1.0.2 will be fixed in 1.0.2-beta2.
```



```
        unsigned int payload;
        unsigned int padding = 16; /* Use minimum padding */

-       /* Read type and payload length first */
-       hbtype = *p++;
-       n2s(p, payload);
-       pl = p;
-
        if (s->msg_callback)
                s->msg_callback(0, s->version, TLS1_RT_HEARTBEAT,
                        &s->s3->rrec.data[0], s->s3->rrec.length,
                        s, s->msg_callback_arg);

+       /* Read type and payload length first */
+       if (1 + 2 + 16 > s->s3->rrec.length)
+               return 0; /* silently discard */
+       hbtype = *p++;
+       n2s(p, payload);
+       if (1 + 2 + payload + 16 > s->s3->rrec.length)
```

"Optic Nerve" – aka Terrorists just wanna have fun

"Unfortunately . . . it would appear that a surprising number of people use webcam conversations to show intimate parts of their body to the other person. Also, the fact that the Yahoo software allows more than one person to view a webcam stream without necessarily sending a reciprocal stream means that it appears sometimes to be used for broadcasting pornography."

"3% and 11% of the Yahoo webcam imagery harvested by GCHQ contains 'undesirable nudity'."

Picture credit: Jeff Dunham via ngwanehansel.blogspot.nl

“Optic Nerve” – aka Terrorists just wanna have fun

“Unfortunately . . . it would appear that a surprising number of people use webcam conversations to show intimate parts of their body to the other person. Also, the fact that the Yahoo software allows more than one person to view a webcam stream without necessarily sending a reciprocal stream means that it appears sometimes to be used for broadcasting pornography.”

“3% and 11% of the Yahoo webcam imagery harvested by GCHQ contains ‘undesirable nudity’.”

But they have more problems:

Picture credit: Jeff Dunham via `ngwanehansel.blogspot.nl`

"Optic Nerve" – aka Terrorists just wanna have fun

"Unfortunately . . . it would appear that a surprising number of people use webcam conversations to show intimate parts of their body to the other person. Also, the fact that the Yahoo software allows more than one person to view a webcam stream without necessarily sending a reciprocal stream means that it appears sometimes to be used for broadcasting pornography."

"3% and 11% of the Yahoo webcam imagery harvested by GCHQ contains 'undesirable nudity'."

But they have more problems:

"We use face detection to try to censor material which may be offensive ..."

Picture credit: Jeff Dunham via ngwanehansel.blogspot.nl

“Optic Nerve” – aka Terrorists just wanna have fun

“Unfortunately . . . it would appear that a surprising number of people use webcam conversations to show intimate parts of their body to the other person. Also, the fact that the Yahoo software allows more than one person to view a webcam stream without necessarily sending a reciprocal stream means that it appears sometimes to be used for broadcasting pornography.”

“3% and 11% of the Yahoo webcam imagery harvested by GCHQ contains ‘undesirable nudity’.”

But they have more problems:

“We use face detection to try to censor material which may be offensive ...”

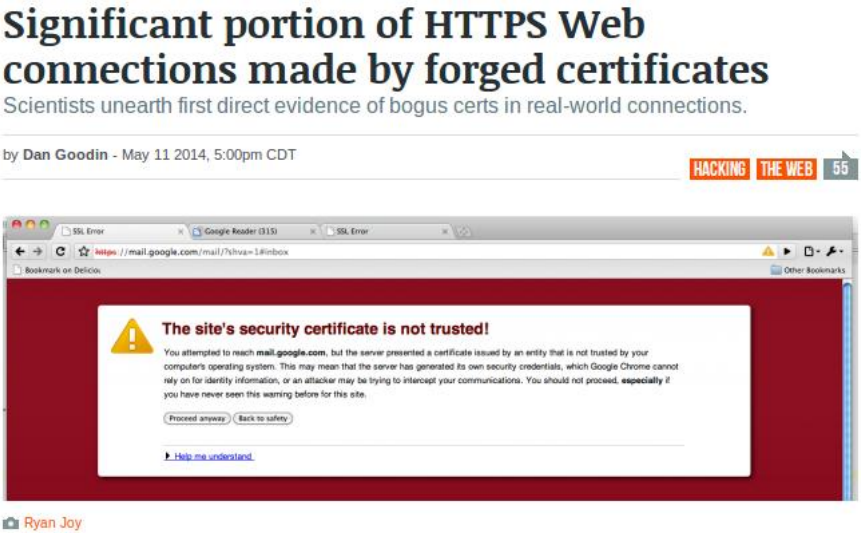
Picture credit: Jeff Dunham via ngwanehansel.blogspot.nl

Meanwhile at the NSA … “Watching Every Word in Snitch City”

Illustration by Intercept staff.

“If you are bothered by snitches in your office, whether of the unwilling or voluntary variety, the best solution is to keep your behavior above reproach. Be a good performer, watch what you say and do, lock your screen when you step away from your workstation, and keep fodder for wagging tongues (your Viagra stash, photos of your wild-and-crazy girls’ weekend in Atlantic City) at home or out of sight.”

11 May 2014: 0.2% of Facebook HTTPS connections are MiTMed



Significant portion of HTTPS Web connections made by forged certificates

Scientists unearth first direct evidence of bogus certs in real-world connections.

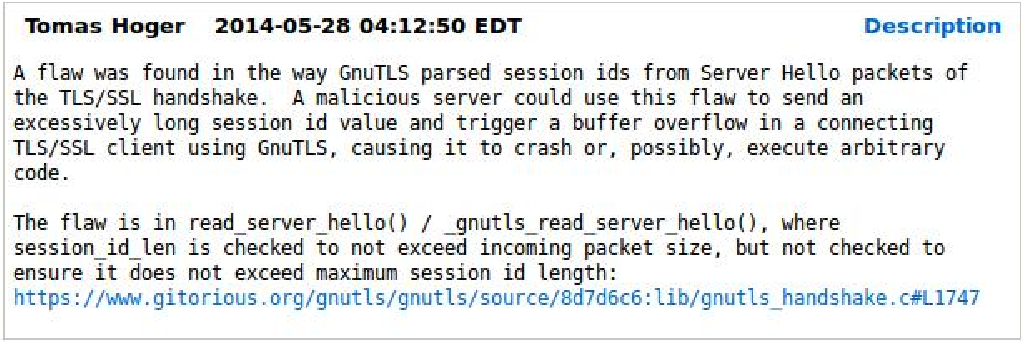
by **Dan Goodin** - May 11 2014, 5:00pm CDT

The site's security certificate is not trusted!

You attempted to reach **mail.google.com**, but the server presented a certificate issued by an entity that is not trusted by your computer's operating system. This may mean that the server has generated its own security credentials, which Google Chrome cannot rely on for identity information, or an attacker may be trying to intercept your communications. You should not proceed, **especially** if you have never seen this warning before for this site.

Ryan Joy

28 May 2014: Remote code execution in GnuTLS



Tomas Hoger **2014-05-28 04:12:50 EDT** **Description**

```
A flaw was found in the way GnuTLS parsed session ids from Server Hello packets of
the TLS/SSL handshake.  A malicious server could use this flaw to send an
excessively long session id value and trigger a buffer overflow in a connecting
TLS/SSL client using GnuTLS, causing it to crash or, possibly, execute arbitrary
code.

The flaw is in read_server_hello() / _gnutls_read_server_hello(), where
session_id_len is checked to not exceed incoming packet size, but not checked to
ensure it does not exceed maximum session id length:
https://www.gitorious.org/gnutls/gnutls/source/8d7d6c6:lib/gnutls_handshake.c#L1747
```

June 2014: A new hope

The New York Times

June 2014: A new hope

The New York Times

Humanity invents crypto programming language that isn't C
JOHN MARKOFF June 4, 2014

THE HAGUE, NETHERLANDS, June 4—The C language is no longer the only possible way to explain encryption methods to a computer, experts announced at a conference here today.

"It was previously believed that crypto could be implemented only in languages with dangerously sharp edges," Dr. Cynthia Solomon said. "But our new language shows that this isn't the case."

"Sure, as if you'll actually convince anyone to use a new language," said Vanee Vines, an NSA spokeswoman, while trying not to laugh.