Sneaking key escrow in through the back door

Tanja Lange

Technische Universiteit Eindhoven
http://projectbullrun.org/dual-ec/

11 February 2015

Capstone Project

- NSA program, public since 1993.
- Standards for government, also planned for commercial and private use.
- Advertised as making strong cryptography available, no risk to security of country and citizens.

Capstone Project

- NSA program, public since 1993.
- Standards for government, also planned for commercial and private use.
- Advertised as making strong cryptography available, no risk to security of country and citizens.
- New designs (and acronyms):
  - Escrowed Encryption Standard (EES)
  - Law Enforcement Access Field (LEAF)
- Key escrow highly controversial: can be used to spy on citizens and adds weakness to system.

Capstone Project

- NSA program, public since 1993.
- Standards for government, also planned for commercial and private use.
- Advertised as making strong cryptography available, no risk to security of country and citizens.
- New designs (and acronyms):
  - Escrowed Encryption Standard (EES)
  - Law Enforcement Access Field (LEAF)
- Key escrow highly controversial: can be used to spy on citizens and adds weakness to system.
- Most prominent example: Clipper chip.
- Matt Blaze showed how to circumvent escrow part; project stopped.

[Photo by Travis Goodspeed]

Fast forward 10(?) years

Fast forward 10(?) years

Random numbers are important

- Cryptography needs random numbers to generate long-term secret keys for encryption and signatures.
- Many schemes expect random (or pseudorandom) numbers, e.g.
  - ephemeral keys for DH key exchange,
  - nonces for digital signatures,
  - nonces in authenticated encryption.
- Nonce reuse can reveal long-term secret keys (e.g. PlayStation disaster)
- DSA/ECDSA are so touchy that biased nonces are enough to break them.

Random numbers are important to the NSA

- Cryptography needs random numbers to generate long-term secret keys for encryption and signatures.
- Many schemes expect random (or pseudorandom) numbers, e.g.
  - ephemeral keys for DH key exchange,
  - nonces for digital signatures,
  - nonces in authenticated encryption.
- Nonce reuse can reveal long-term secret keys (e.g. PlayStation disaster)
- DSA/ECDSA are so touchy that biased nonces are enough to break them.

Snowden at SXSW:

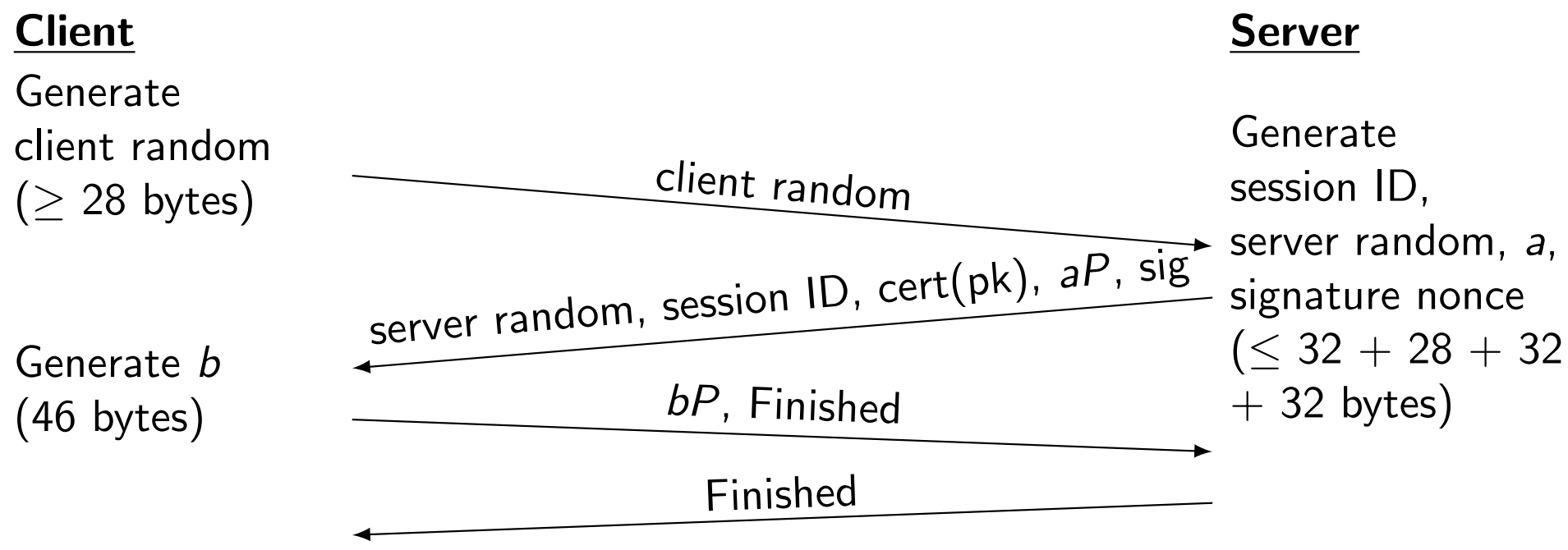
> *[..] we know that these encryption algorithms we are using today work typically it is the random number generators that are attacked as opposed to the encryption algorithms themselves.*

SSL/TLS/HTTPS – internet security protocols

Use of randomness in internet protocols.

SSL/TLS/HTTPS – internet security protocols

Use of randomness in internet protocols.

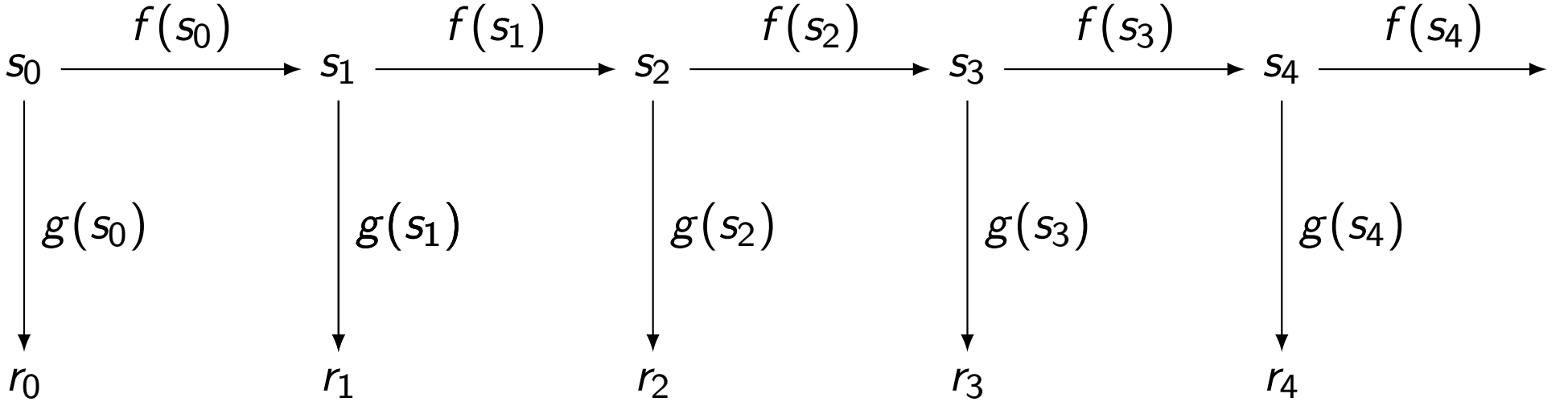


MS = PRF($x(abP)$, "master secret", client random —— server random)

Pseudo-random-number generators

Crypto libraries expand short seed into long stream of random bits.
Random bits are used as secret keys, DSA nonces, ...

The usual structure, starting from short seed s_1:



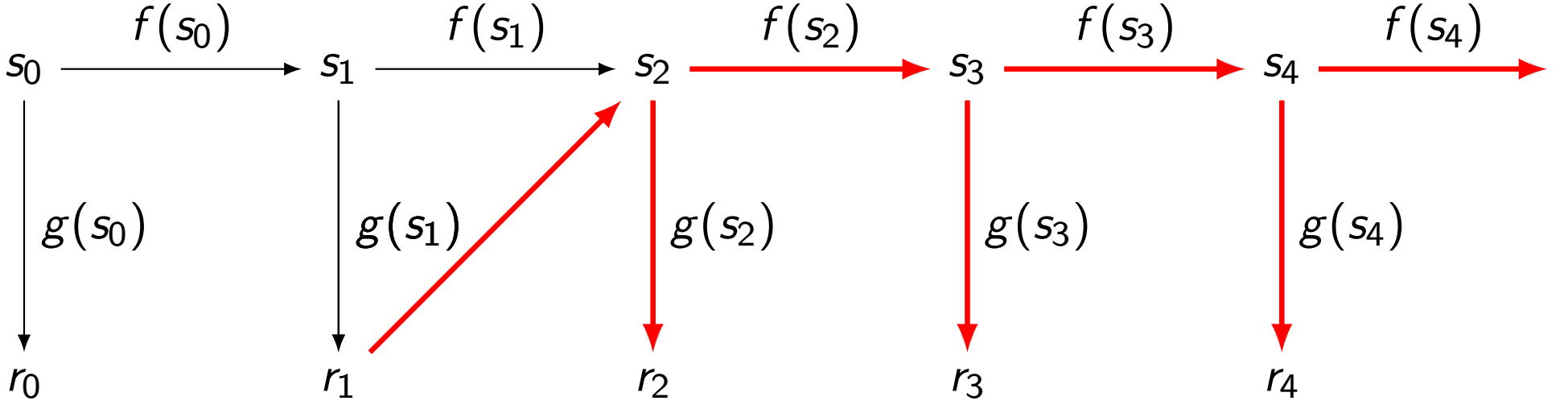
XXX's mission: Predict the "random" output bits.

1. Create protocols that directly output r_n for some reason.

Pseudo-random-number generators

Crypto libraries expand short seed into long stream of random bits.
Random bits are used as secret keys, DSA nonces, ...

The usual structure, starting from short seed s_1:



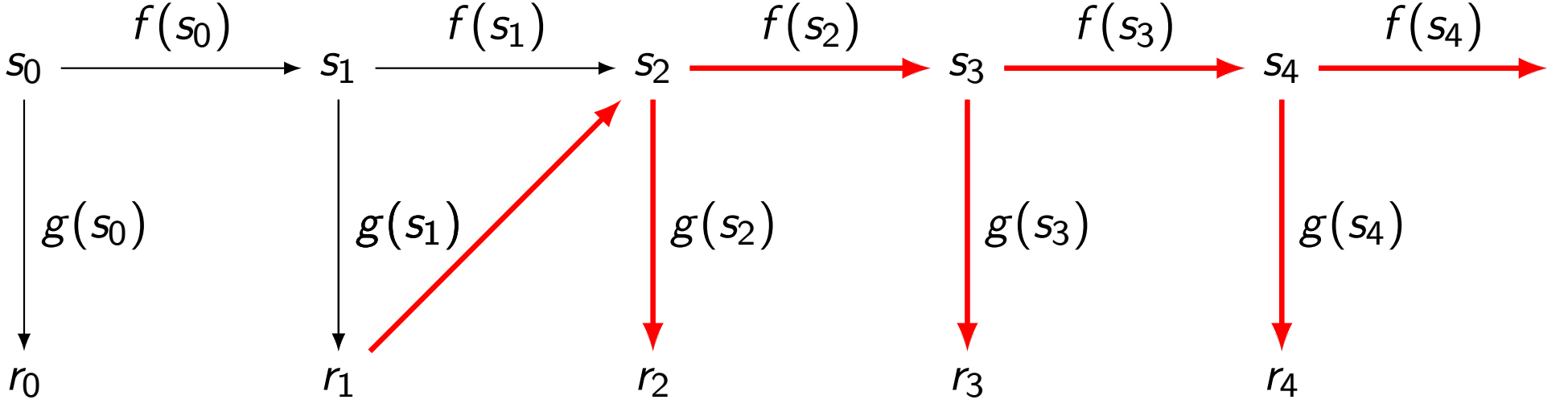
XXX's mission: Predict the "random" output bits.

1. Create protocols that directly output r_n for some reason.
2. Design f, g with back door from r_n to s_{n+1}: i.e., get $f(s)$ from $g(s)$.

Pseudo-random-number generators

Crypto libraries expand short seed into long stream of random bits.
Random bits are used as secret keys, DSA nonces, ...

The usual structure, starting from short seed s_1:



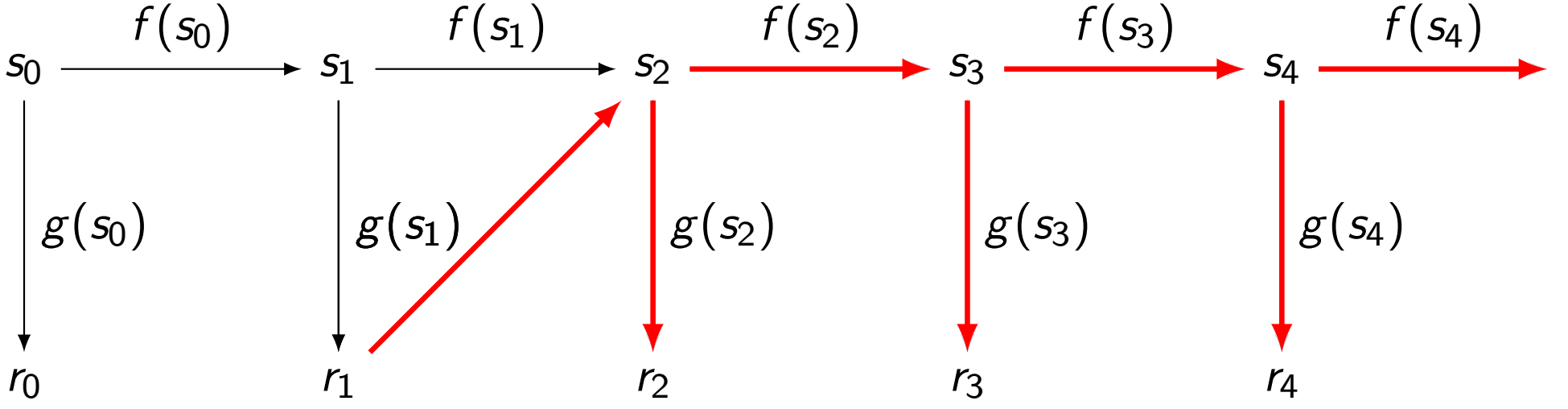
XXX's mission: Predict the "random" output bits.

1. Create protocols that directly output r_n for some reason.
2. Design f, g with back door from r_n to s_{n+1}: i.e., get $f(s)$ from $g(s)$.
3. Standardize this design of f, g.

Pseudo-random-number generators

Crypto libraries expand short seed into long stream of random bits.
Random bits are used as secret keys, DSA nonces, ...

The usual structure, starting from short seed s_1:



XXX's mission: Predict the "random" output bits.

1. Create protocols that directly output r_n for some reason.
2. Design f, g with back door from r_n to s_{n+1}: i.e., get $f(s)$ from $g(s)$.
3. Standardize this design of f, g.
4. Convince users to switch to this design: e.g., publish "security proof".

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.*

Dual EC had been flagged before for being extremely inefficient and possibly backdoored . . . so 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.*

Dual EC had been flagged before for being extremely inefficient and possibly backdoored . . . so 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.*

Dual EC had been flagged before for being extremely inefficient and possibly backdoored . . . so 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; recommends against using Dual_EC.
RSA suggests changing default in BSAFE.

21 April 2014 NIST removes Dual EC from the standard.

ELATI ED VIDEO

bama on surveillance:
here may be another way
skinning the cat"

December 2013

(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 firm in the computer security industry, Reuters has learned.

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

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

December 22, 2013

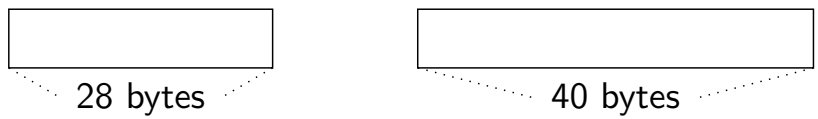
Recent press coverage has asserted that RSA entered into a "secret contract" with the NSA to incorporate a known flawed random number generator into its BSAFE encryption libraries. We categorically deny this allegation.

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.
- When NIST issued new guidance recommending no further use of this algorithm in September 2013, we adhered to that guidance, communicated that recommendation to customers and discussed the change openly in the

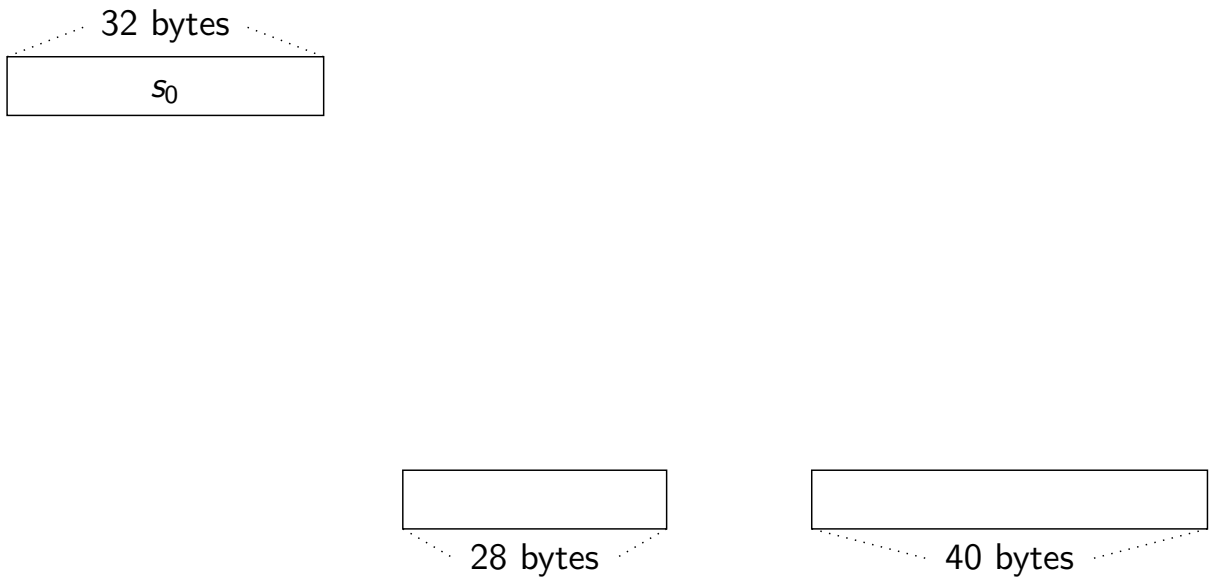
Dual EC in TLS



Graphic thanks to Ruben Niederhagen.

Dual EC in TLS

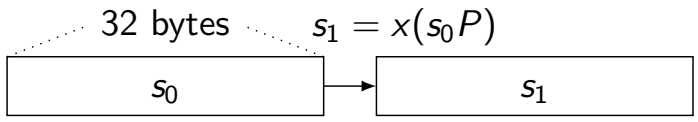
Points Q and P on an elliptic curve.

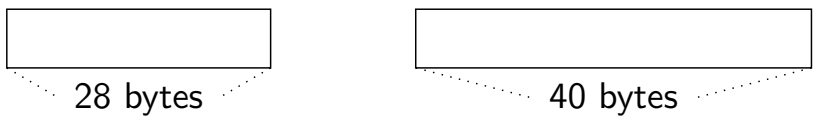


Graphic thanks to Ruben Niederhagen.

Dual EC in TLS

Points Q and P on an elliptic curve.

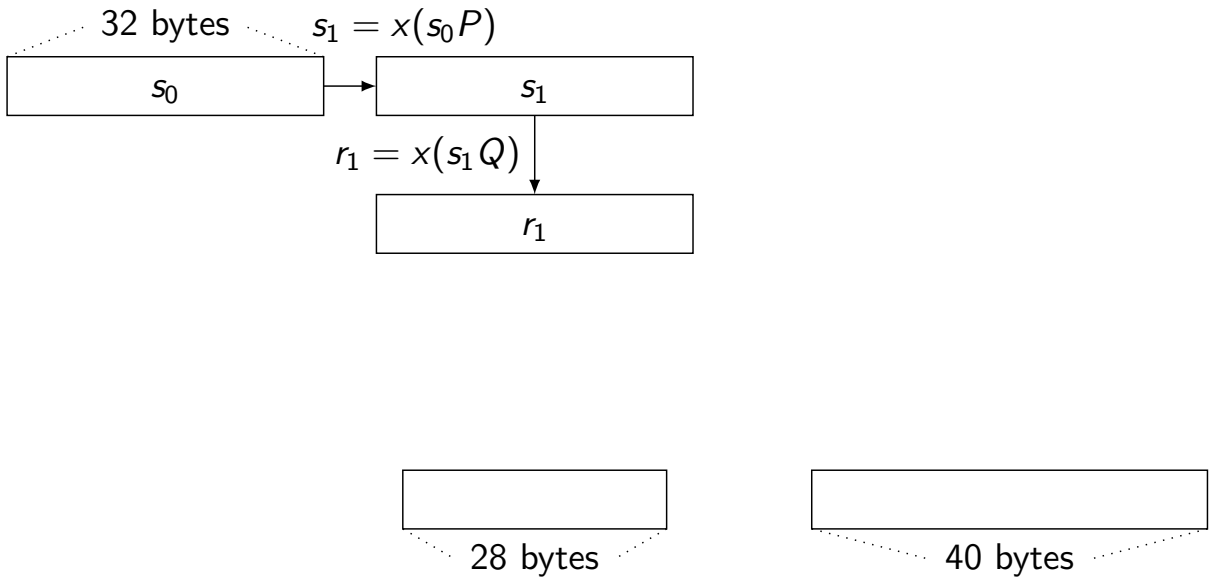




Graphic thanks to Ruben Niederhagen.

Dual EC in TLS

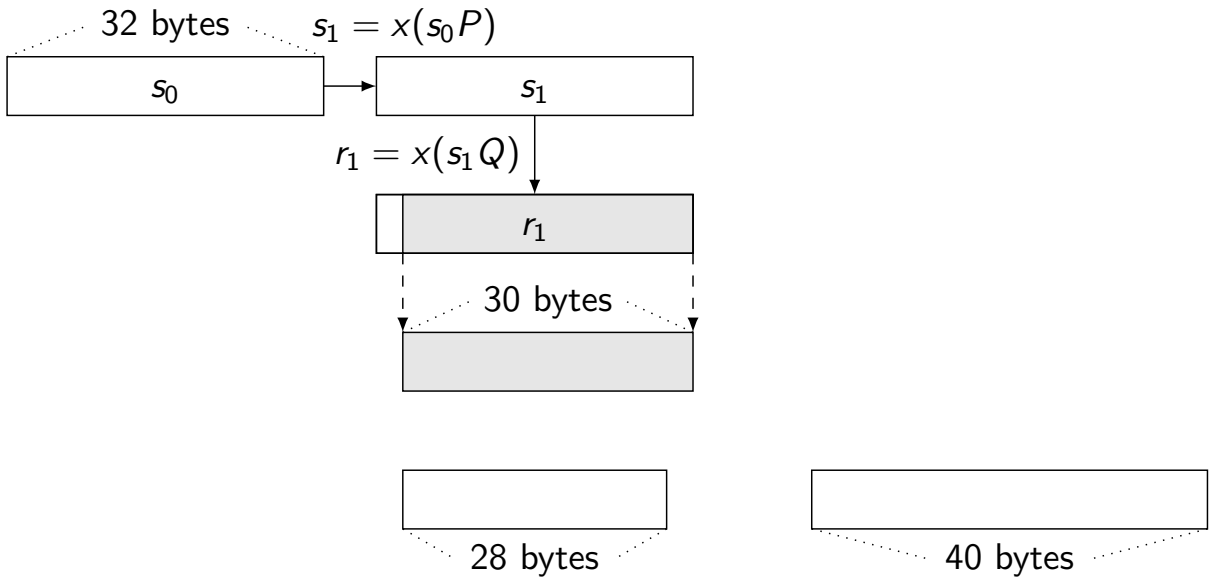
Points Q and P on an elliptic curve.



Graphic thanks to Ruben Niederhagen.

Dual EC in TLS

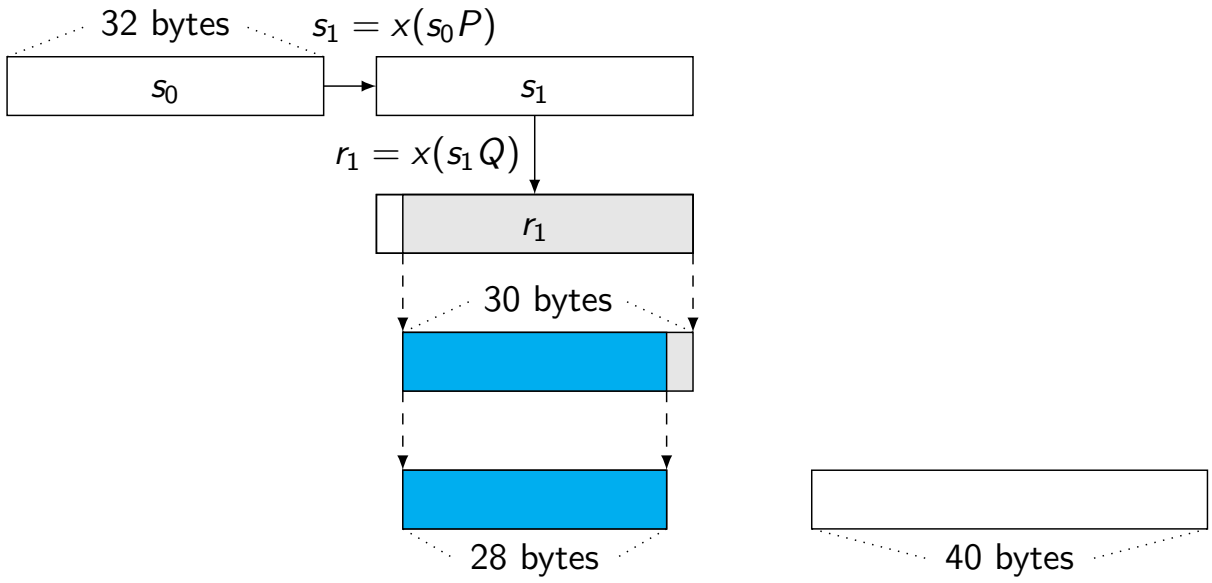
Points Q and P on an elliptic curve.



Graphic thanks to Ruben Niederhagen.

Dual EC in TLS

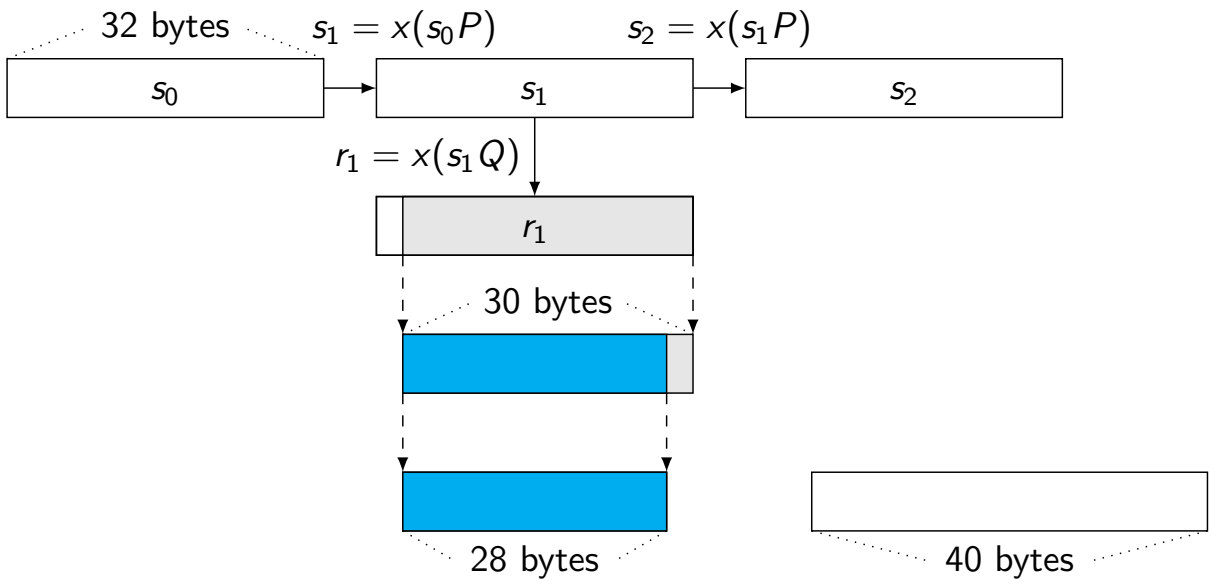
Points Q and P on an elliptic curve.



Graphic thanks to Ruben Niederhagen.

Dual EC in TLS

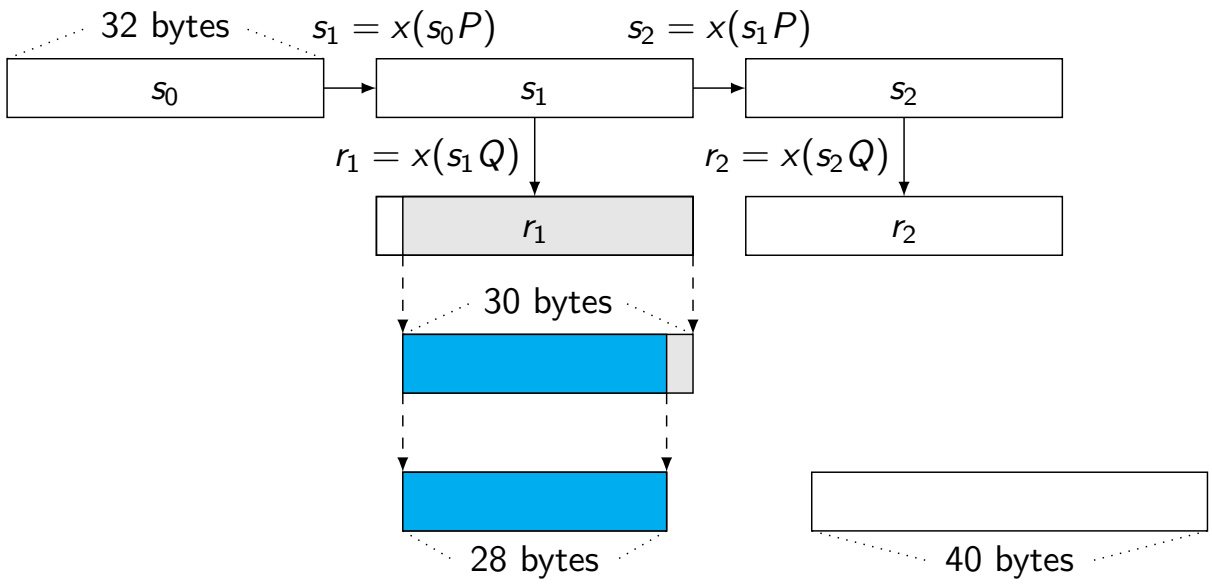
Points Q and P on an elliptic curve.



Graphic thanks to Ruben Niederhagen.

Dual EC in TLS

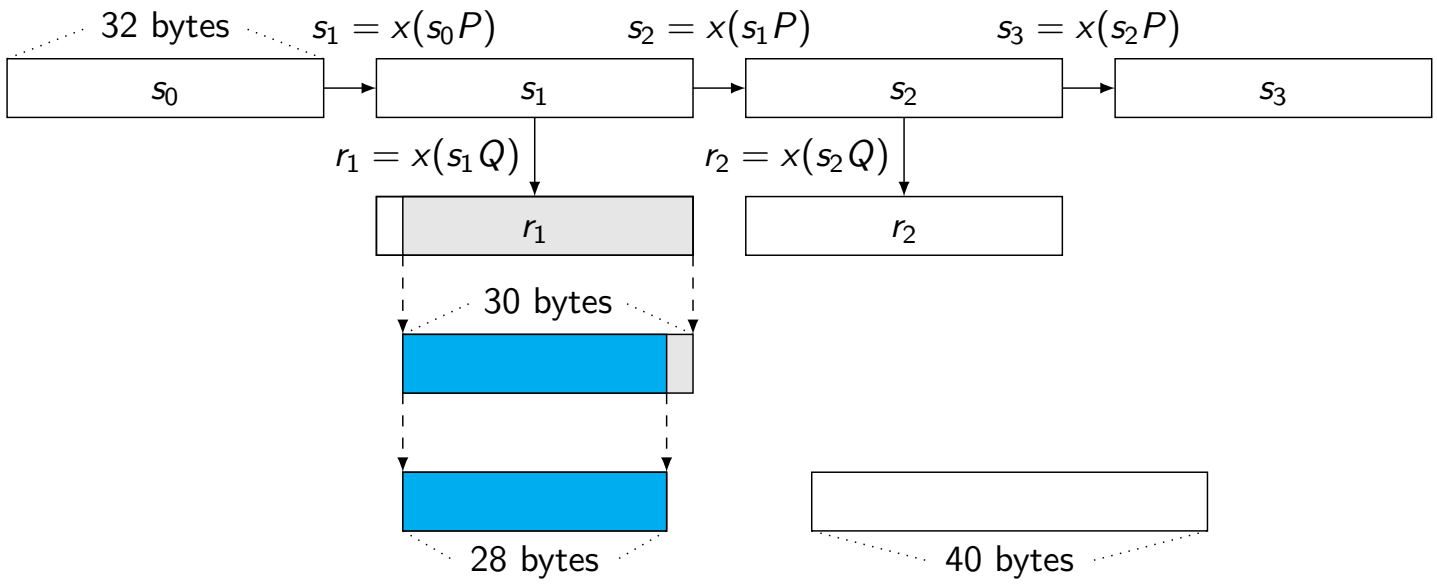
Points Q and P on an elliptic curve.



Graphic thanks to Ruben Niederhagen.

Dual EC in TLS

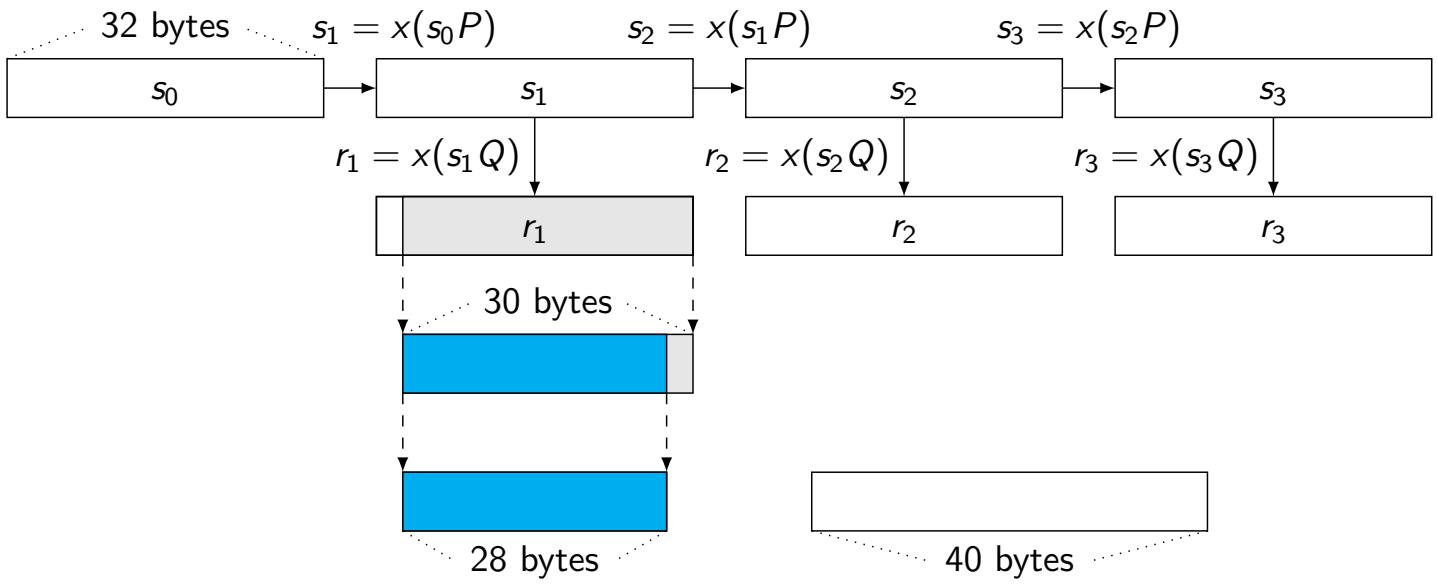
Points Q and P on an elliptic curve.



Graphic thanks to Ruben Niederhagen.

Dual EC in TLS

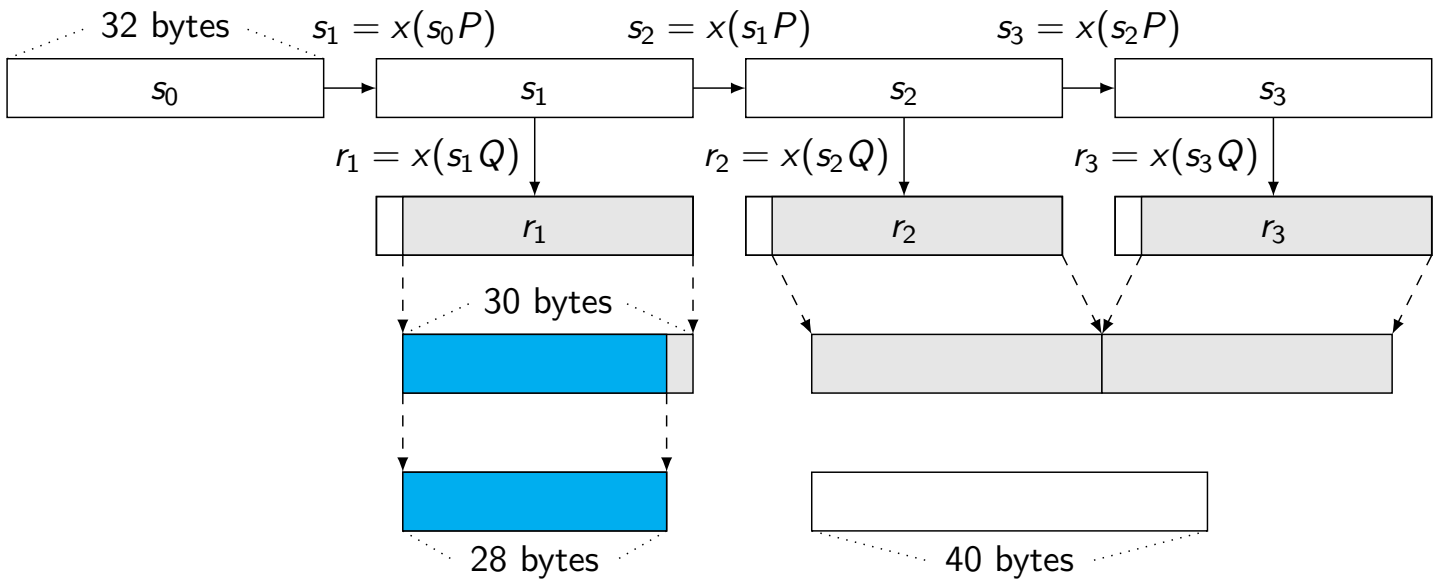
Points Q and P on an elliptic curve.



Graphic thanks to Ruben Niederhagen.

Dual EC in TLS

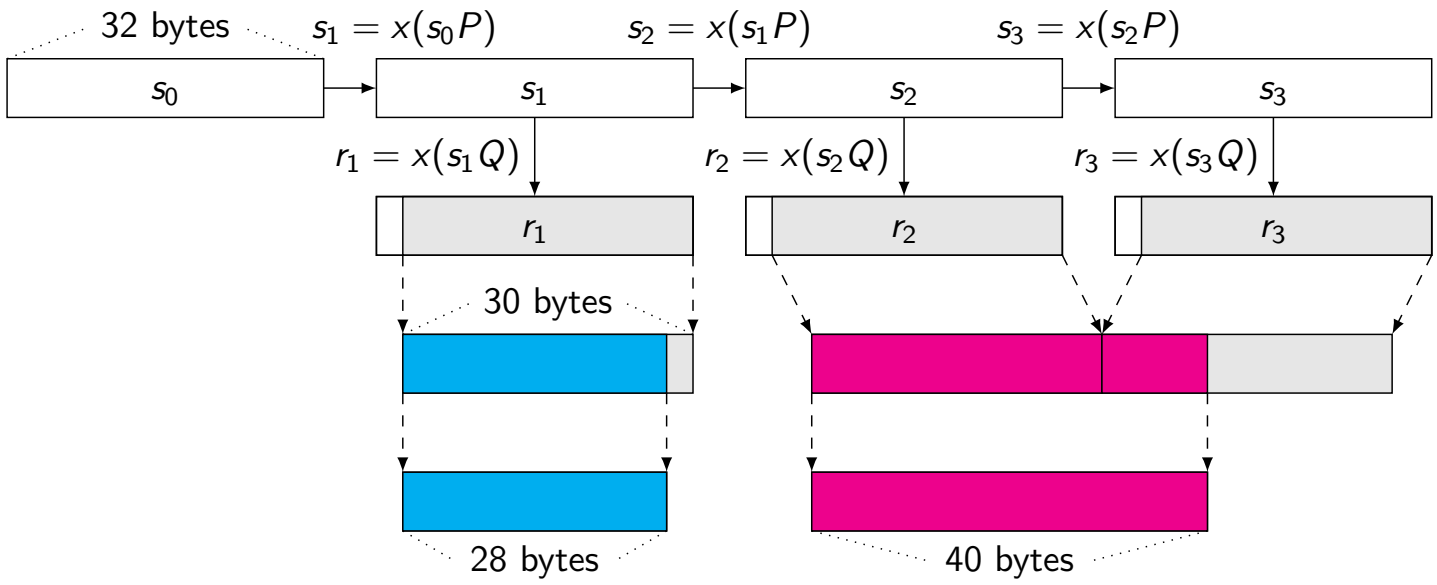
Points Q and P on an elliptic curve.



Graphic thanks to Ruben Niederhagen.

Dual EC in TLS

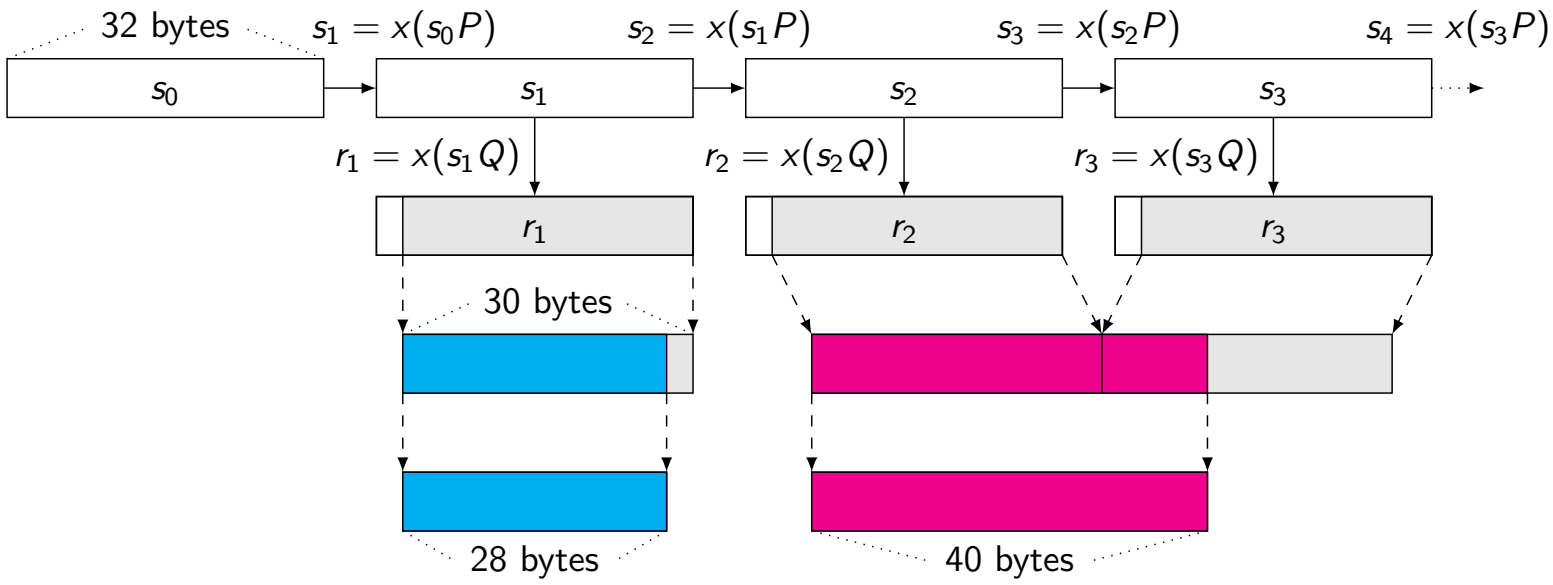
Points Q and P on an elliptic curve.



Graphic thanks to Ruben Niederhagen.

Dual EC in TLS

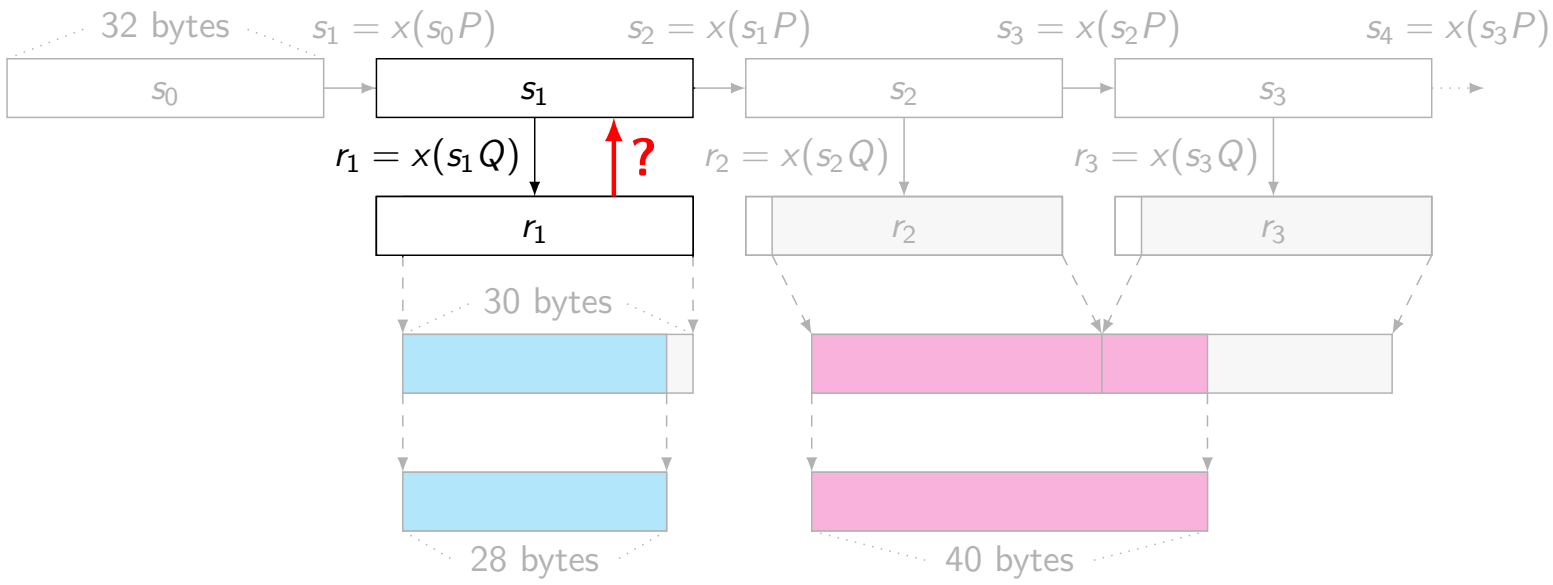
Points Q and P on an elliptic curve.



Graphic thanks to Ruben Niederhagen.

Dual EC in TLS

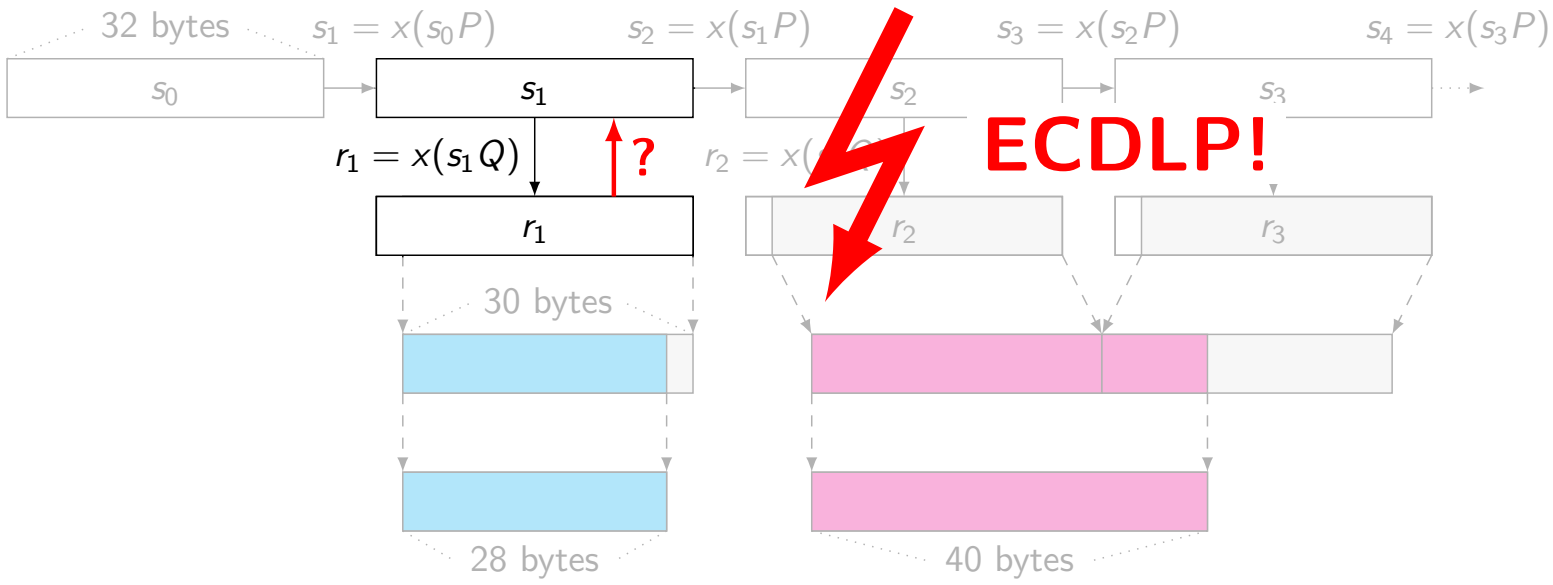
Points Q and P on an elliptic curve.



Graphic thanks to Ruben Niederhagen.

Dual EC in TLS

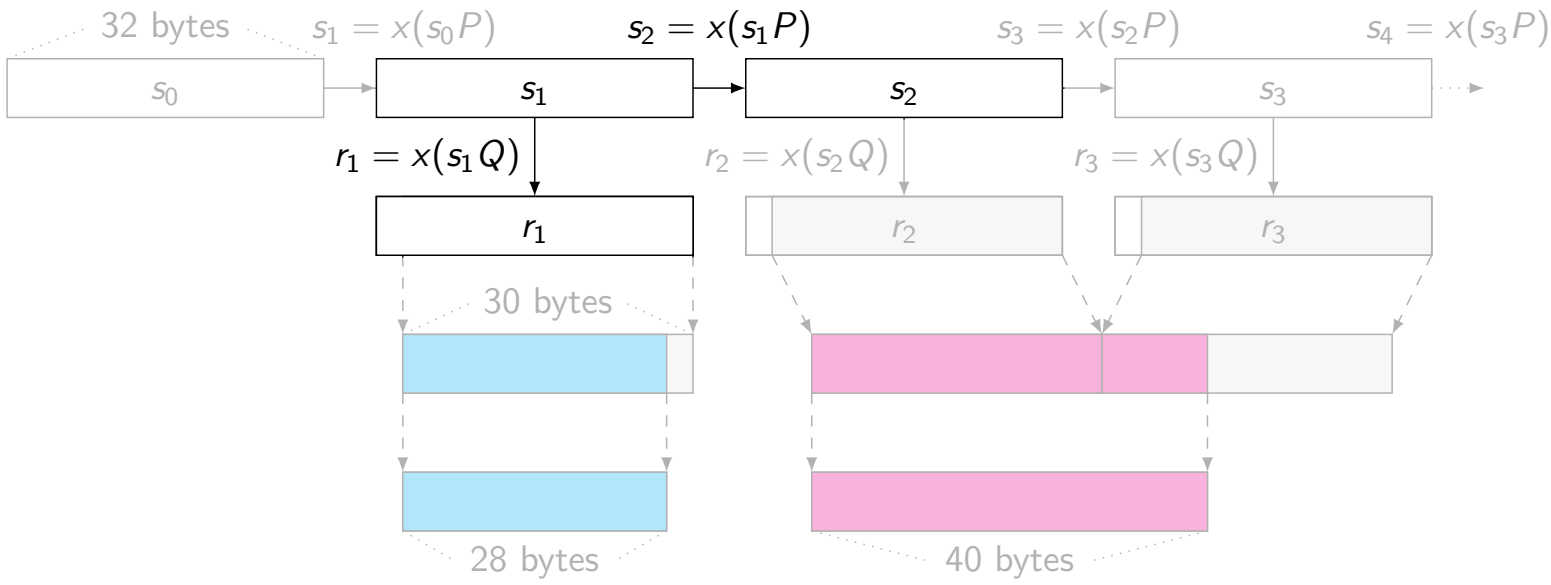
Points Q and P on an elliptic curve.



Graphic thanks to Ruben Niederhagen.

Basic attack

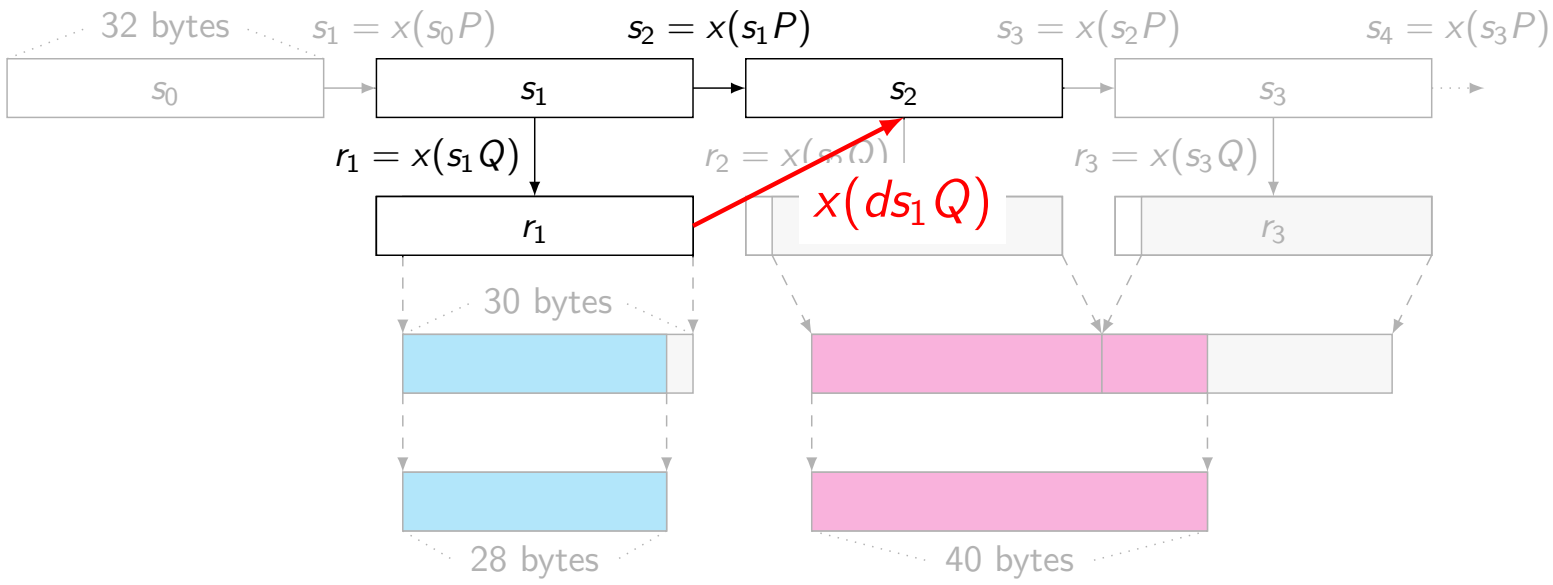
Points Q and $P = dQ$ on an elliptic curve.



Graphic thanks to Ruben Niederhagen.

Basic attack

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

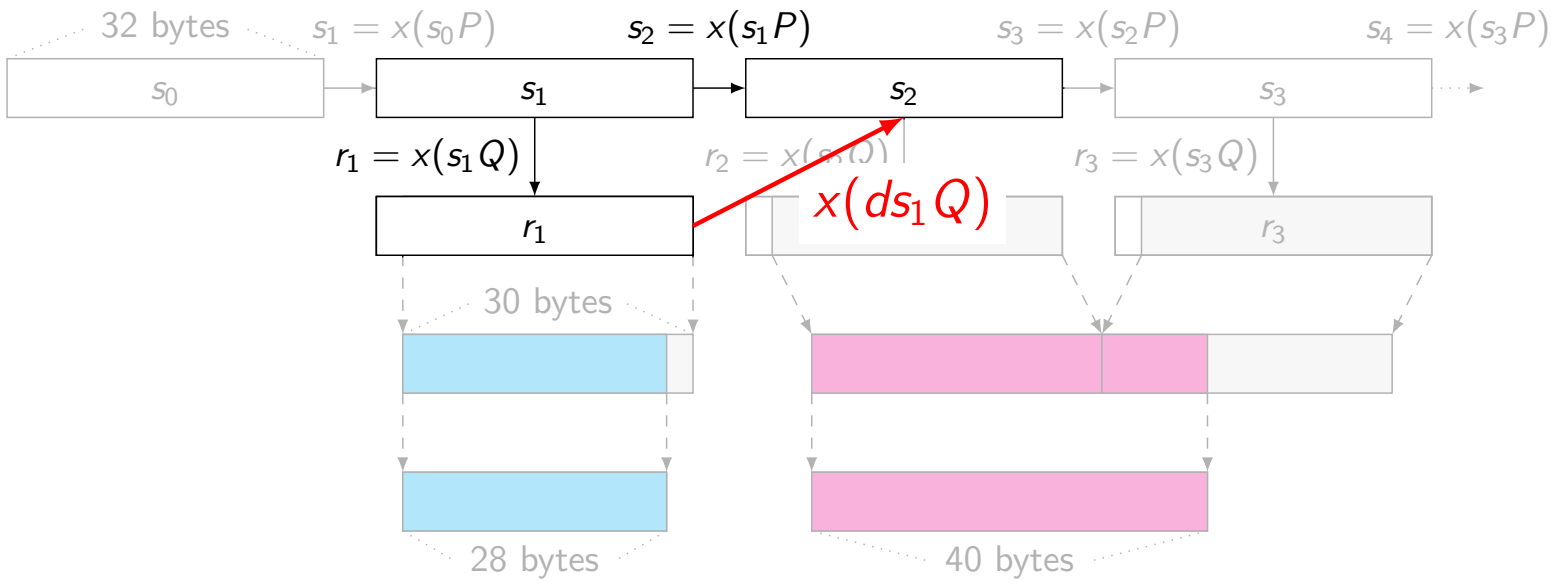


Graphic thanks to Ruben Niederhagen.

Basic attack

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

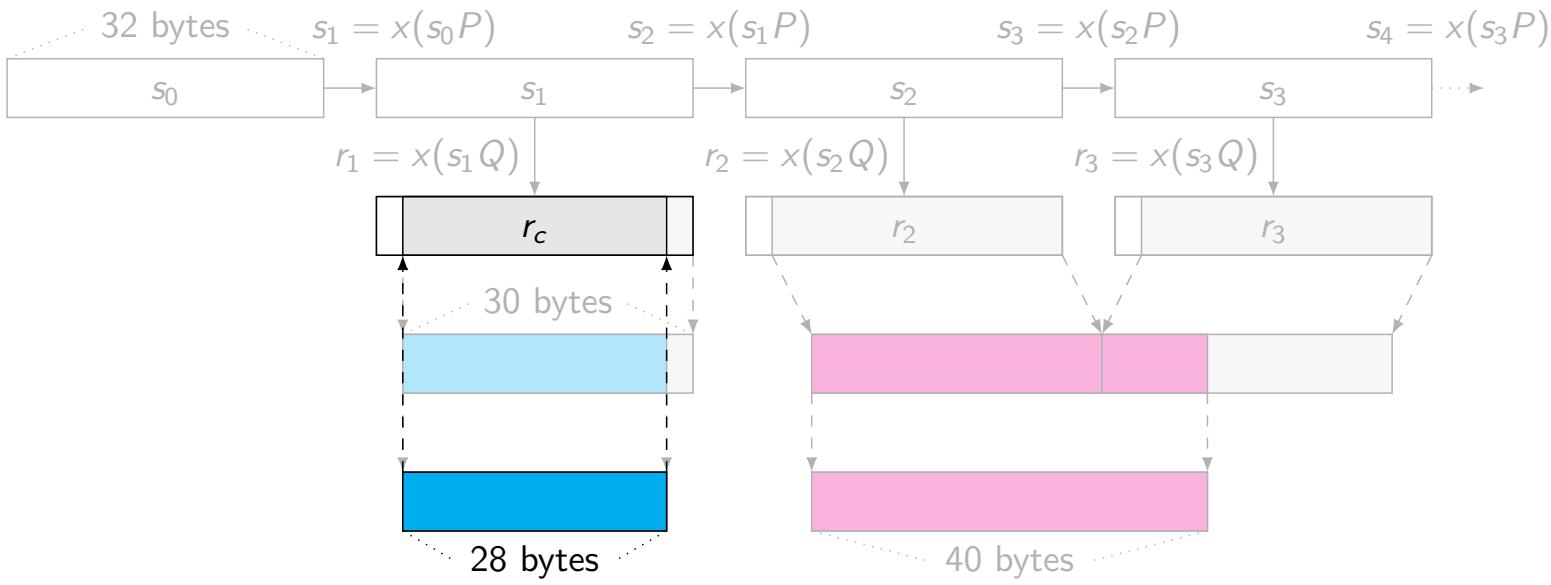
$$s_2 = x(s_1 P) = x(s_1 dQ)$$



Graphic thanks to Ruben Niederhagen.

Basic attack

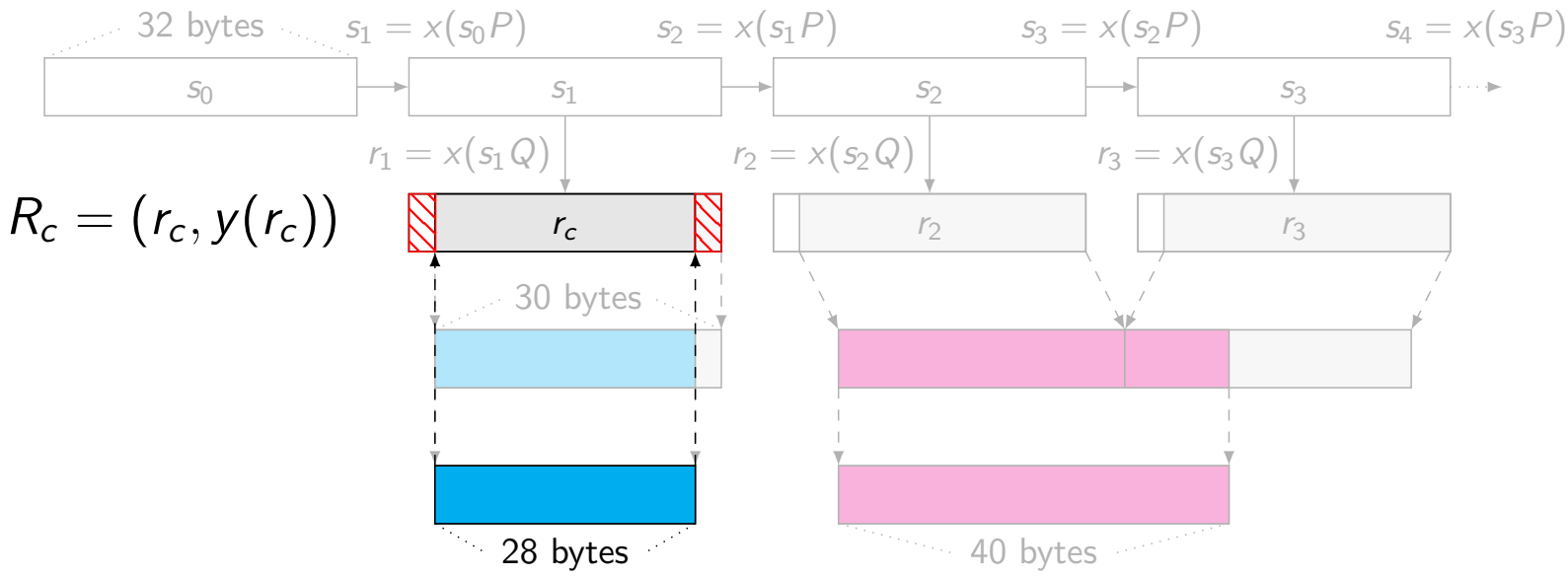
Points Q and $P = dQ$ on an elliptic curve.



Graphic thanks to Ruben Niederhagen.

Basic attack

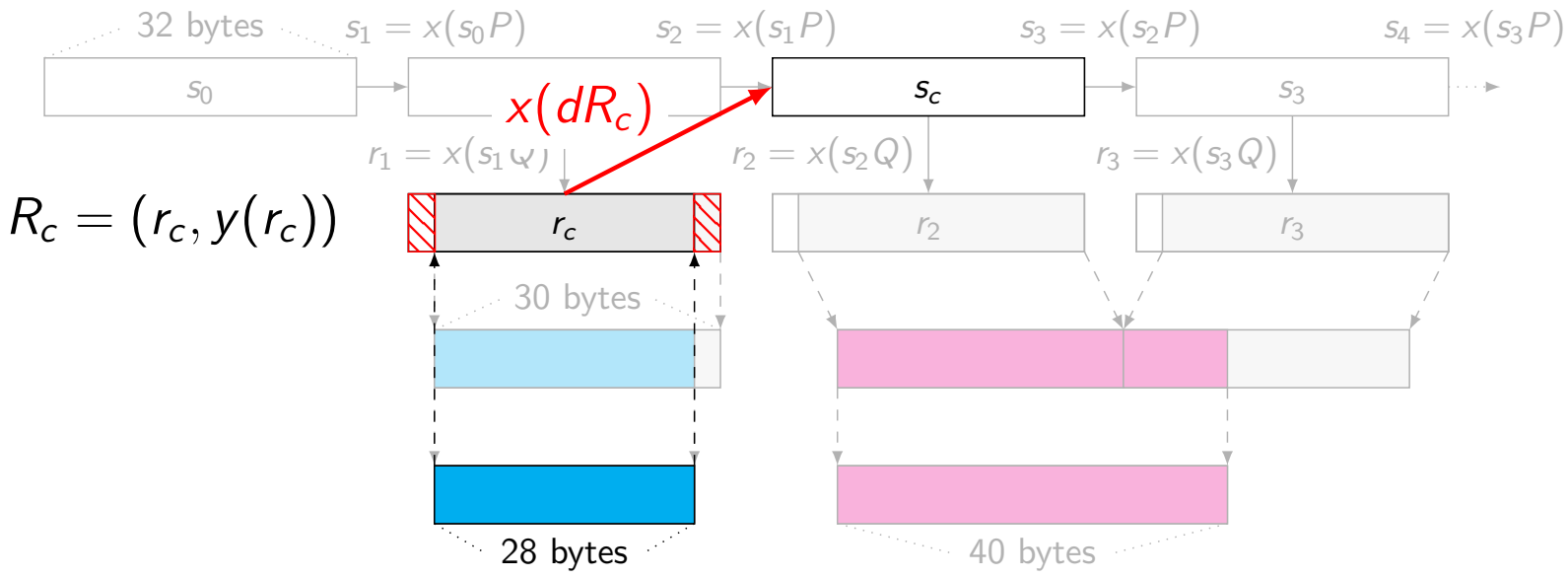
Points Q and $P = dQ$ on an elliptic curve.



Graphic thanks to Ruben Niederhagen.

Basic attack

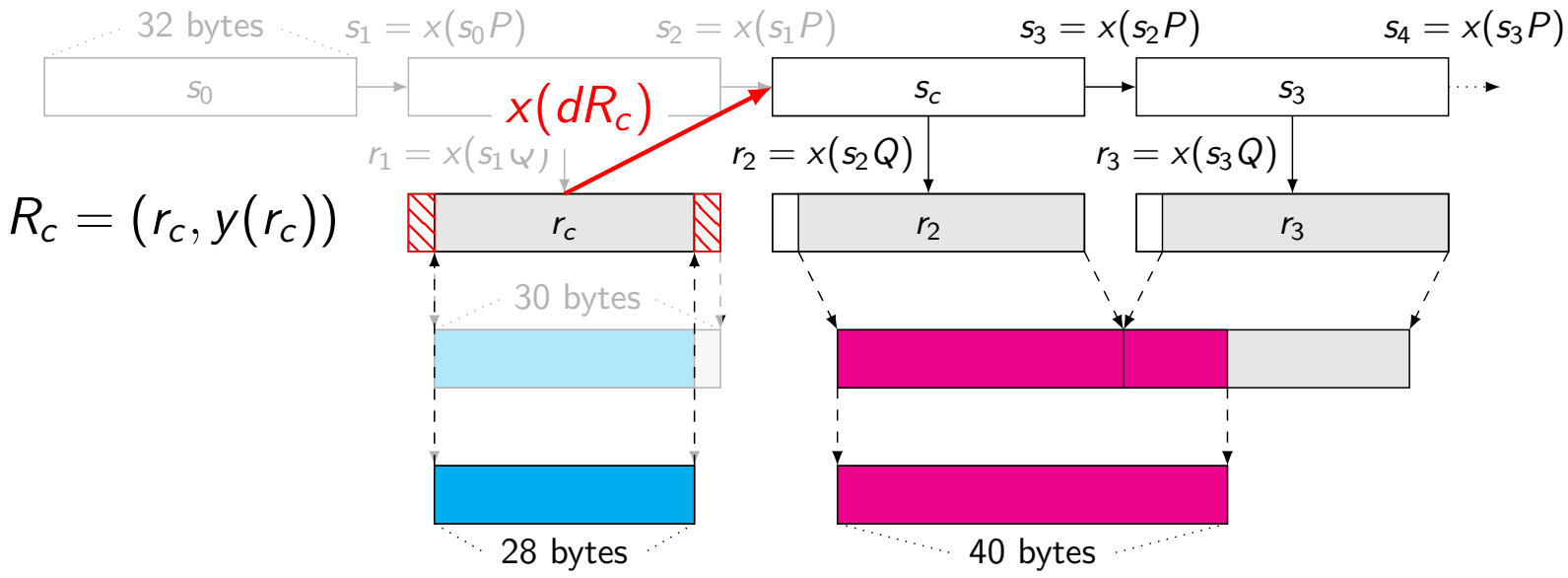
Points Q and $P = dQ$ on an elliptic curve.



Graphic thanks to Ruben Niederhagen.

Basic attack

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



Graphic thanks to Ruben Niederhagen.

Timings

Attack	Bytes per session	Additional entropy (bits)	Time (min)
BSAFE-C v1.1	31–60		0.04*
BSAFE-Java v1.1	28		63.96*
SChannel I	28		62.97*
SChannel II	30		182.64*
OpenSSL-fixed I	32	20	0.02*
OpenSSL-fixed II	32	35	83.32*
OpenSSL-fixed III	32	$35+k$	$2^k \cdot 83.32$

*measured on 16 core cluster

Some more fun with RSA's BSAFE-Java

No additional input,

Some more fun with RSA's BSAFE-Java

No additional input, **explicit watermark** in handshake $\Rightarrow$ easy recognition.

Some more fun with RSA's BSAFE-Java

No additional input, **explicit watermark** in handshake ⇒ easy recognition.

Alas, BSAFE does not give fresh randomness in session ID, so attack costs roughly 2^{32}.

Network Working Group — E. Rescorla
Internet-Draft — RTFM, Inc.
Intended status: Informational — M. Salter
Expires: September 3, 2009 — National Security Agency
March 02, 2009

Extended Random Values for TLS
draft-rescorla-tls-extended-random-02.txt

[..] The rationale for this as stated by DoD is that the public randomness for each side should be at least twice as long as the security level for cryptographic parity, which makes the 224 bits of randomness provided by the current TLS random values

How did we get here . . .

Official editors of SP800-90 are Elaine Barker and John Kelsey.

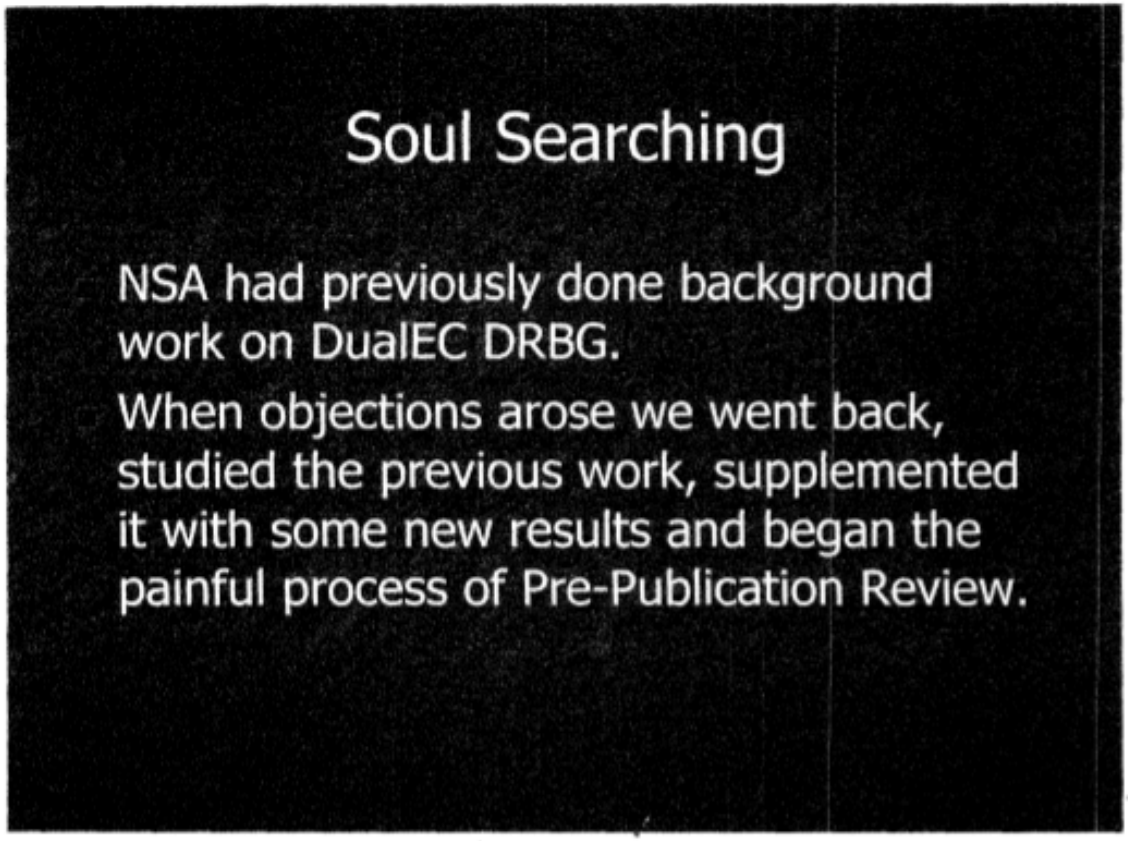
No editors stated for ANSI X9.82 nor for ISO 18031.

Interesting Dec 2013 slide deck by John Kelsey 800 – 90 and Dual EC DRBG.

- Standardization effort by NIST and NSA, with some participation from CSE.
- Most of work on standards done by US federal employees (NIST and NSA, with some help from CSE).
- The standard Dual EC parameters P and Q come ultimately from designers of Dual EC DRBG at NSA.

NIST FOIA

Two FOIA requests by Andrew Crocker and Nate Cardozo of EFF and Matthew Stoller and Rep. Alan Grayson. Files hosted by Matt Green at https://github.com/matthewdgreen/nistfoia. Interesting documents, e.g.



This is most likely a reaction to the research on biases.

From 011 – 9.12 Choosing a DRBG Algorithm.pdf

9.12 Choosing a DRBG Algorithm

Almost no system designer starts out with the idea that he's going to generate good random bits. Instead, he typically starts with some goal he wishes to accomplish, then decides on

X.2 DRBGs Based on Block Ciphers

[[This is all assuming my block cipher based schemes are acceptable to the NSA guys doing the review.--JMK]]

X.3 DRBGs Based on Hard Problems

[[Okay, so here's the limit of my competence. Can Don or Dan or one of the NSA guys with some number theory/algebraic geometry background please look this over? Thanks! --JMK]]

[[I'm really blowing smoke here. Would someone with some actual understanding of these attacks please save me from diving off a cliff right here? --JMK]]

US 20070189527A1

(19) **United States**

(12) **Patent Application Publication**

(10) **Pub. No.: US 2007/0189527 A1**

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

Brown et al.

(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)

(21) Appl. No.: **11/336,814**

(22) Filed: **Jan. 23, 2006**

Related U.S. Application Data

(60) Provisional application No. 60/644,982, filed on Jan. 21, 2005.

Publication Classification

(51) **Int. Cl.**
H04L 9/00 (2006.01)

(52) **U.S. Cl.** **380/44**

(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.

Certicom patents

The Canadian company Certicom (now part of Blackberry) has patents in multiple countries on

- Dual EC exploitation: the use of Dual EC for key escrow (i.e., for a deliberate back door)
- Dual EC escrow avoidance: modifying Dual EC to avoid key escrow.

The patent filing history also shows that

- Certicom knew the Dual EC back door by 2005;
- NSA was informed of the Dual EC back door by 2005, even if they did not know it earlier;
- the patent application, including examples of Dual EC exploitation, was publicly available in July 2006, just a month after SP800-90 was standardized.

http://projectbullrun.org/dual-ec/patent.html

References

Many more results and much more background is provided at http://projectbullrun.org/dual-ec/.

The research on breaking TLS by using the back door in Dual EC is joint work with Stephen Checkoway, Matthew Fredrikson, Ruben Niederhagen, Adam Everspaugh, Matthew Green, Tanja Lange, Thomas Ristenpart, Daniel J. Bernstein, Jake Maskiewicz, Hovav Shacham reported in "On the Practical Exploitability of Dual EC DRBG in TLS Implementations" published at USENIX Security 2014.