

# Dual EC

a standardized back door

Ruben Niederhagen

Joint work with Stephen Checkoway<sup>1</sup>, Matthew Fredrikson<sup>2</sup>,  
Matthew Green<sup>1</sup>, Tanja Lange<sup>3</sup>, Thomas Ristenpart<sup>2</sup>,  
Daniel J. Bernstein<sup>3,5</sup>, Jake Maskewicz<sup>4</sup>, and Hovav Shacham<sup>4</sup>

<sup>1</sup>Johns Hopkins University, <sup>2</sup>University of Wisconsin,  
<sup>3</sup>Technische Universiteit Eindhoven, <sup>4</sup>UC San Diego,  
<sup>5</sup>University of Illinois at Chicago

**TU** / **e**

Technische Universiteit  
**Eindhoven**  
University of Technology

April 28th, 2014

## Random numbers are crucial for cryptography:

- ▶ generation of **private keys** for authentication
- ▶ generation of **secret keys** for encryption
- ▶ generation of **secret nonces** for digital signatures
- ▶ generation of **ephemeral keys** for perfect-forward secrecy
- ▶ ...

## Random numbers are crucial for cryptography:

- ▶ generation of **private keys** for authentication
- ▶ generation of **secret keys** for encryption
- ▶ generation of **secret nonces** for digital signatures
- ▶ generation of **ephemeral keys** for perfect-forward secrecy
- ▶ ...

Must be impossible for an attacker to predict!

## Challenges of random number generation:

- ▶ computers are built to be deterministic
- ▶ “real” randomness is rare

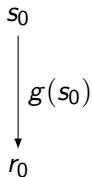
## Challenges of random number generation:

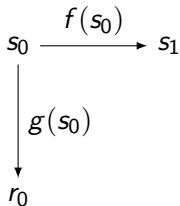
- ▶ computers are built to be deterministic
- ▶ “real” randomness is rare

## Common approach:

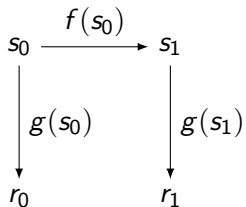
- ▶ use *pseudo* random numbers
- ▶ start with a random *seed*
- ▶ compute subsequent values deterministically  
⇒ update a secret internal state

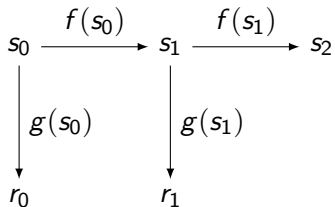
50

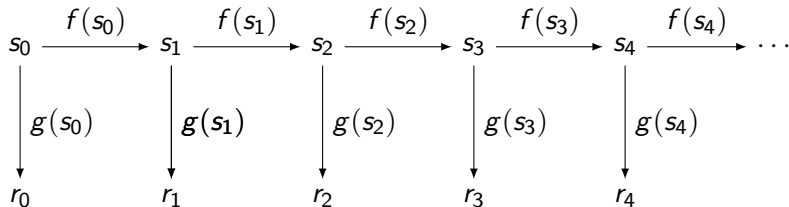


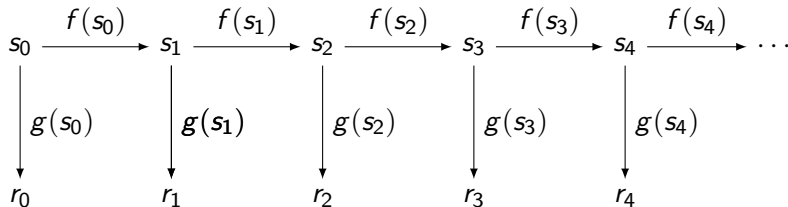




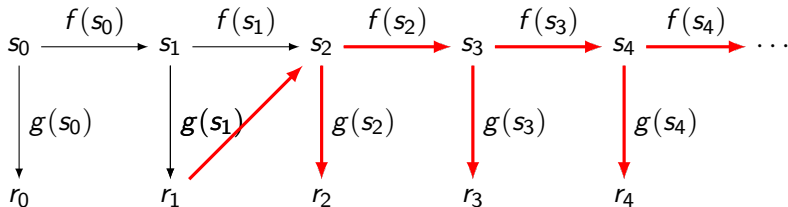








Broken if attacker learns internal state!



Broken if attacker learns internal state!

## Topic of this talk:

The “potential” back door in the  
*Dual Elliptic Curve Deterministic Random Bit Generator* (Dual EC)  
standardized by ANSI, ISO, and NIST.

June 2004 Dual EC appears in an ANSI draft.

June 2004    Dual EC appears in an ANSI draft.  
in 2004        RSA makes Dual EC the default RNG in BSAFE.



- June 2004 Dual EC appears in an ANSI draft.
- in 2004 RSA makes Dual EC the default RNG in BSAFE.
- in 2005 An ISO standard is published including Dual EC.

- June 2004 Dual EC appears in an ANSI draft.
- in 2004 RSA makes Dual EC the default RNG in BSAFE.
- in 2005 An ISO standard is published including Dual EC.
- Dec. 2005 A draft is released by NIST including Dual EC.

- June 2004 Dual EC appears in an ANSI draft.
- in 2004 RSA makes Dual EC the default RNG in BSAFE.
- in 2005 An ISO standard is published including Dual EC.
- Dec. 2005 A draft is released by NIST including Dual EC.
- early 2006 Several researchers, e.g., Schoenmakers and Sidorenko at TU/e, point out cryptographic weaknesses in Dual EC.

- June 2004 Dual EC appears in an ANSI draft.
- in 2004 RSA makes Dual EC the default RNG in BSAFE.
- in 2005 An ISO standard is published including Dual EC.
- Dec. 2005 A draft is released by NIST including Dual EC.
- early 2006 Several researchers, e.g., Schoenmakers and Sidorenko at TU/e, point out cryptographic weaknesses in Dual EC.
- June 2006 NIST SP 800/90A is published including Dual EC, ignoring the warnings.

- June 2004 Dual EC appears in an ANSI draft.
- in 2004 RSA makes Dual EC the default RNG in BSAFE.
- in 2005 An ISO standard is published including Dual EC.
- Dec. 2005 A draft is released by NIST including Dual EC.
- early 2006 Several researchers, e.g., Schoenmakers and Sidorenko at TU/e, point out cryptographic weaknesses in Dual EC.
- June 2006 NIST SP 800/90A is published including Dual EC, ignoring the warnings. This includes Dual EC in FIPS 140-2, the typical certification for RNGs.

- June 2004 Dual EC appears in an ANSI draft.
- in 2004 RSA makes Dual EC the default RNG in BSAFE.
- in 2005 An ISO standard is published including Dual EC.
- Dec. 2005 A draft is released by NIST including Dual EC.
- early 2006 Several researchers, e.g., Schoenmakers and Sidorenko at TU/e, point out cryptographic weaknesses in Dual EC.
- June 2006 NIST SP 800/90A is published including Dual EC, ignoring the warnings. This includes Dual EC in FIPS 140-2, the typical certification for RNGs.
- Aug. 2007 Shumow and Ferguson demonstrate the basic back door.

5 Sept. 2013      NSA's "Project Bullrun" is revealed by documents from Edward Snowden with the purpose "to covertly introduce weaknesses into the encryption standards followed by hardware and software developers around the world."

5 Sept. 2013 NSA's "Project Bullrun" is revealed by documents from Edward Snowden with the purpose "to covertly introduce weaknesses into the encryption standards followed by hardware and software developers around the world." The New York Times writes that "the NSA had inserted a back door into a 2006 standard adopted by NIST [...] called the Dual EC DRBG standard."



- 5 Sept. 2013 NSA's "Project Bullrun" is revealed by documents from Edward Snowden with the purpose "to covertly introduce weaknesses into the encryption standards followed by hardware and software developers around the world." The New York Times writes that "the NSA had inserted a back door into a 2006 standard adopted by NIST [...] called the Dual EC DRBG standard."
- 19 Sept. 2013 RSA advises not to use Dual EC.

- 5 Sept. 2013 NSA's "Project Bullrun" is revealed by documents from Edward Snowden with the purpose "to covertly introduce weaknesses into the encryption standards followed by hardware and software developers around the world." The New York Times writes that "the NSA had inserted a back door into a 2006 standard adopted by NIST [...] called the Dual EC DRBG standard."
- 19 Sept. 2013 RSA advises not to use Dual EC.
- 20 Dec. 2013 Reuters reports that NSA paid RSA \$10 million to use Dual EC as their default RNG.

- 5 Sept. 2013 NSA's "Project Bullrun" is revealed by documents from Edward Snowden with the purpose "to covertly introduce weaknesses into the encryption standards followed by hardware and software developers around the world." The New York Times writes that "the NSA had inserted a back door into a 2006 standard adopted by NIST [...] called the Dual EC DRBG standard."
- 19 Sept. 2013 RSA advises not to use Dual EC.
- 20 Dec. 2013 Reuters reports that NSA paid RSA \$10 million to use Dual EC as their default RNG.
- 21 Apr. 2014 NIST removes Dual EC from the standard.

## Kelsey, in December 2013 slides:

- ▶ 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”.

## Transport Layer Security (TLS)

- ▶ Used in the Internet for encryption of communication.  
Examples:
  - ▶ eMail transport
  - ▶ online banking
  - ▶ online shopping
  - ▶ ...
- ▶ Standard covers a vast amount of protocols and optional features.
- ▶ Client and server agree on what parameters to use.
- ▶ Client and server agree on a **random secret key**.

## Client

generate  
client random

client random

server random, session ID, cert(pk), aP, sig

generate  $b$

$bP$ , Finished

Finished

## Server

generate  
session ID,  
server random,  $a$ ,  
signature nonce



## Common TLS implementations:

- ▶ RSA's BSAFE
  - ▶ RSA BSAFE Share for Java (BSAFE Java)
  - ▶ RSA BSAFE Share for C and C++ (BSAFE C)
- ▶ Microsoft's SChannel
- ▶ OpenSSL

All of these offer Dual EC.

## Common TLS implementations:

- ▶ **RSA's BSAFE**
  - ▶ RSA BSAFE Share for Java (BSAFE Java)
  - ▶ RSA BSAFE Share for C and C++ (BSAFE C)
- ▶ Microsoft's SChannel
- ▶ OpenSSL

Remember: NSA paid RSA Security \$10 million to use Dual EC as the default RNG!

All of these offer Dual EC.



## Arithmetic on Elliptic Curves

Operate on points  $P = (x_P, y_P)$  on an elliptic curve.

- ▶ addition:  $A + B = C$
- ▶ scalar mul.:  $4 \cdot A = A + A + A + A$

## Arithmetic on Elliptic Curves

Operate on points  $P = (x_P, y_P)$  on an elliptic curve.

- ▶ addition:  $A + B = C$
- ▶ scalar mul.:  $4 \cdot A = A + A + A + A$

## Useful in Cryptography:

It is easy to compute  $k \cdot A$ , e.g.:

$$B = 243 \cdot A = A + 2A + 16A + 32A + 64A + 128A$$

Cost: 5 additions and 7 doublings

## Arithmetic on Elliptic Curves

Operate on points  $P = (x_P, y_P)$  on an elliptic curve.

- ▶ addition:  $A + B = C$
- ▶ scalar mul.:  $4 \cdot A = A + A + A + A$

## Useful in Cryptography:

It is *easy* to compute  $k \cdot A$ , e.g.:

$$B = 243 \cdot A = A + 2A + 16A + 32A + 64A + 128A$$

Cost: 5 additions and 7 doublings

It is *hard* to find  $k$  such that  $B = k \cdot A!$

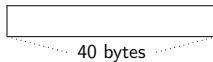
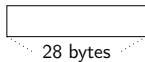
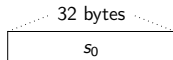


28 bytes

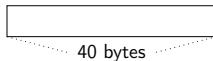
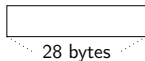
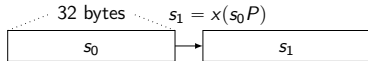


40 bytes

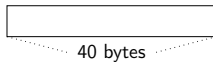
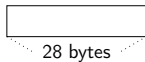
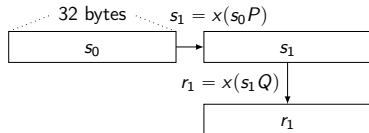
Points  $Q$  and  $P$  on an elliptic curve.



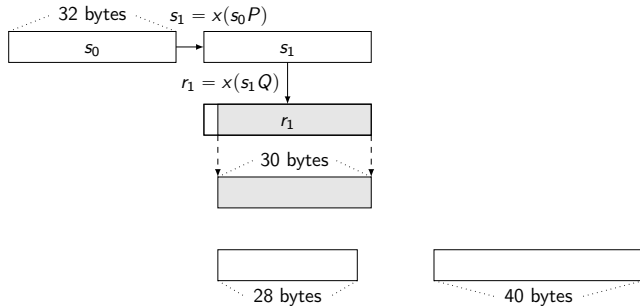
Points  $Q$  and  $P$  on an elliptic curve.



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

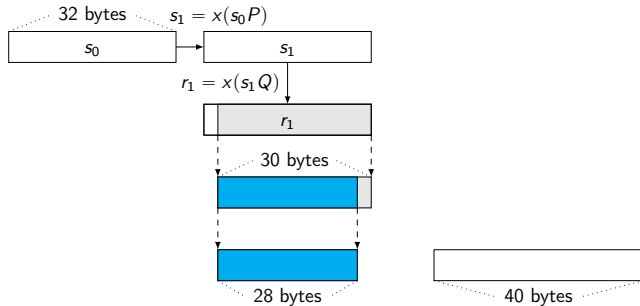


Points  $Q$  and  $P$  on an elliptic curve.

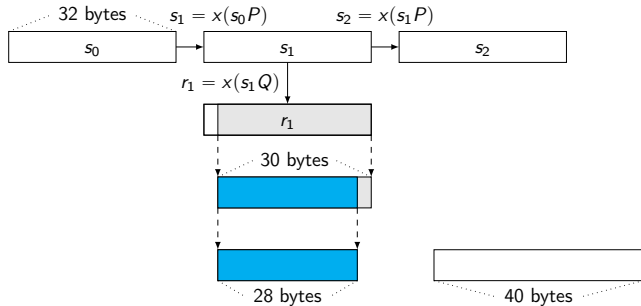




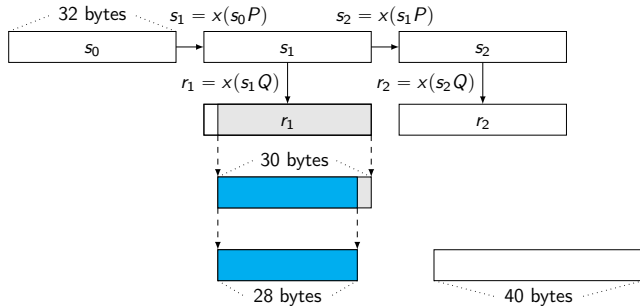
Points  $Q$  and  $P$  on an elliptic curve.



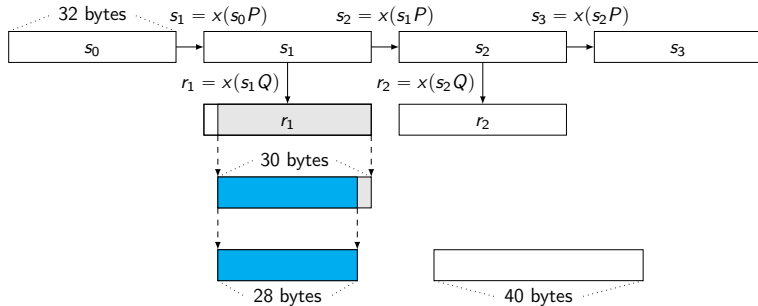
Points  $Q$  and  $P$  on an elliptic curve.



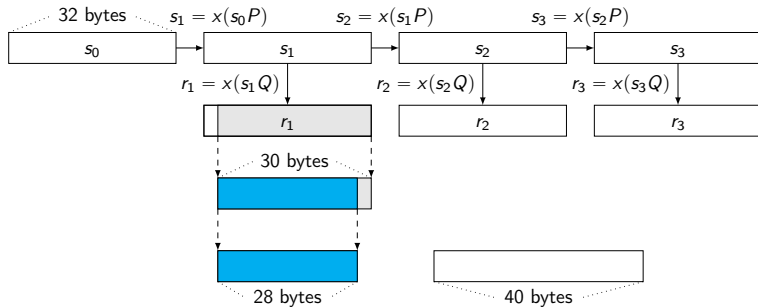
Points  $Q$  and  $P$  on an elliptic curve.



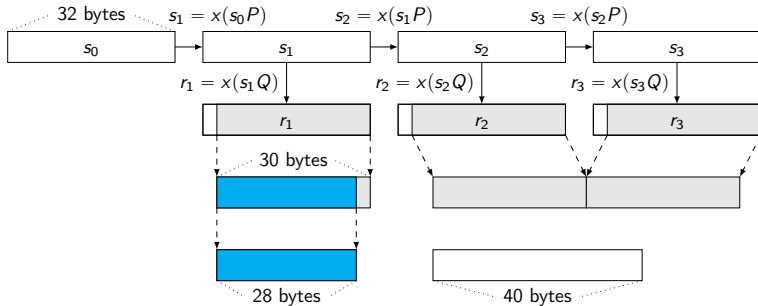
Points  $Q$  and  $P$  on an elliptic curve.



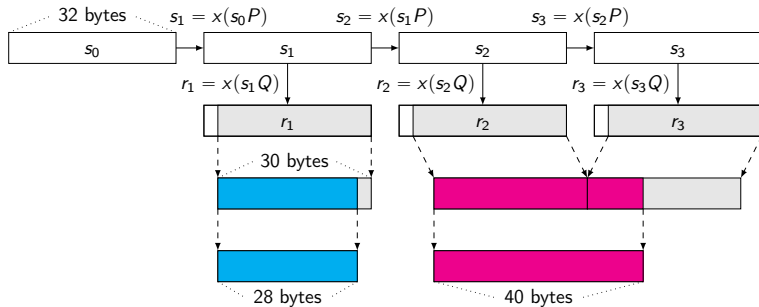
Points  $Q$  and  $P$  on an elliptic curve.



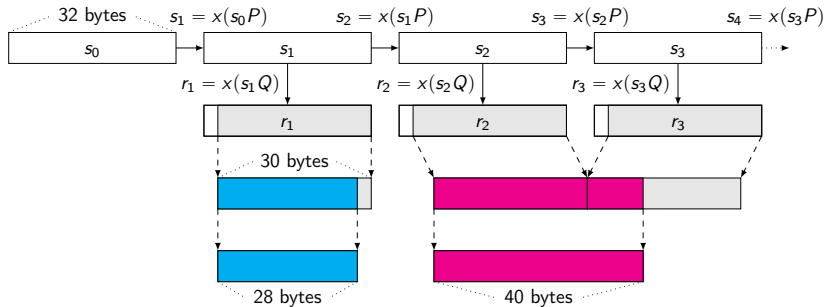
Points  $Q$  and  $P$  on an elliptic curve.



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

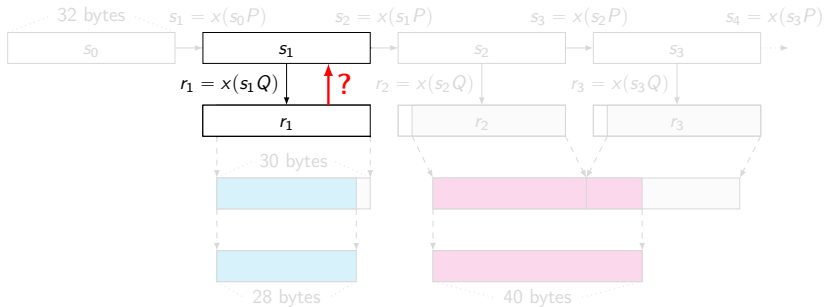


Points  $Q$  and  $P$  on an elliptic curve.

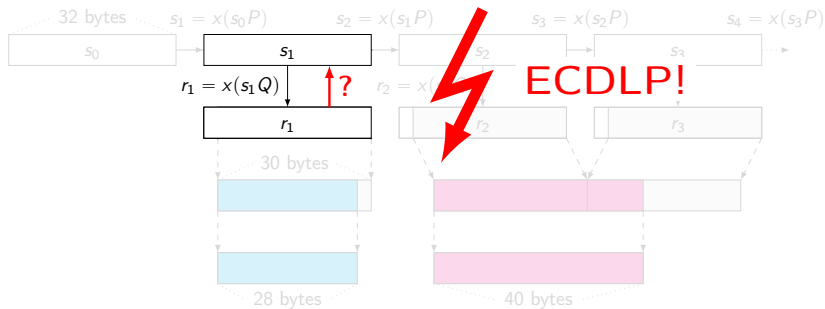




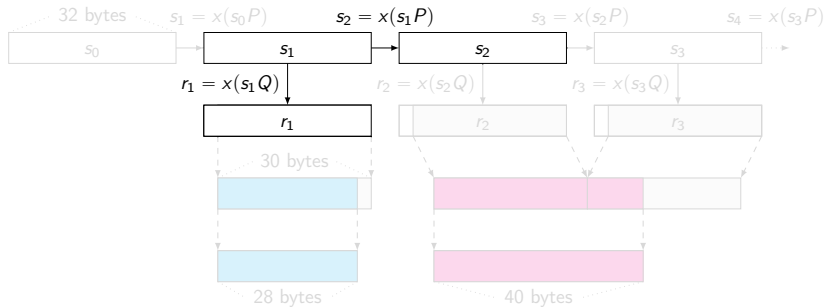
Points  $Q$  and  $P$  on an elliptic curve.



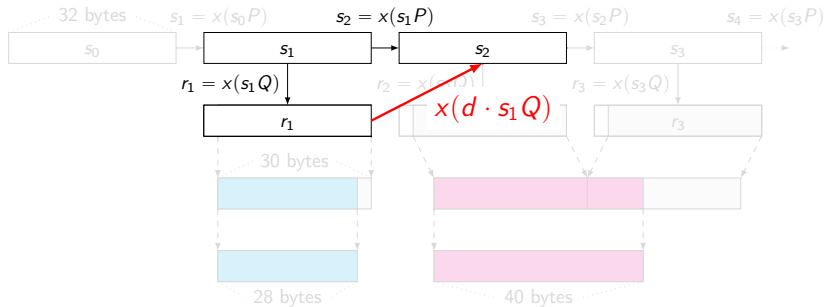
Points  $Q$  and  $P$  on an elliptic curve.



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

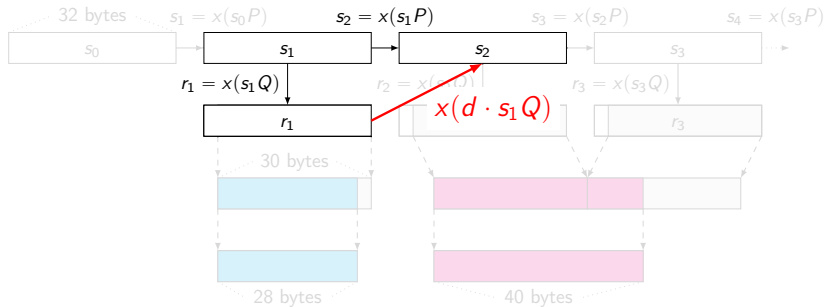


Points  $Q$  and  $P = dQ$  on an elliptic curve.

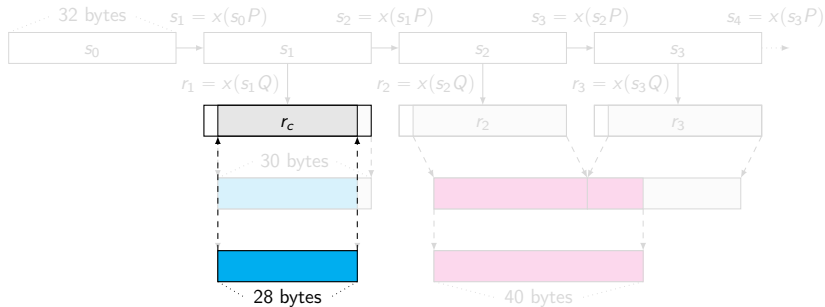


Points  $Q$  and  $P = dQ$  on an elliptic curve.

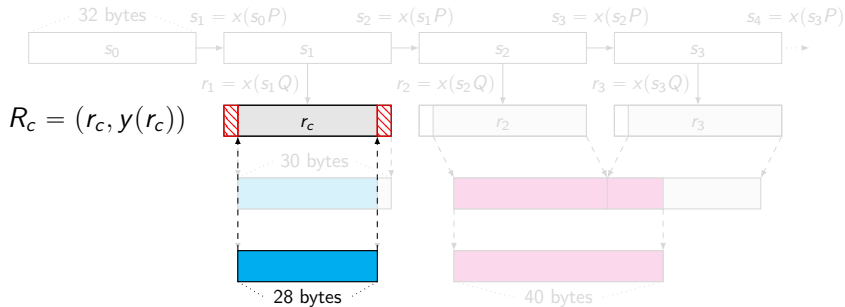
$$s_2 = x(s_1 P) = x(s_1 \cdot dQ)$$



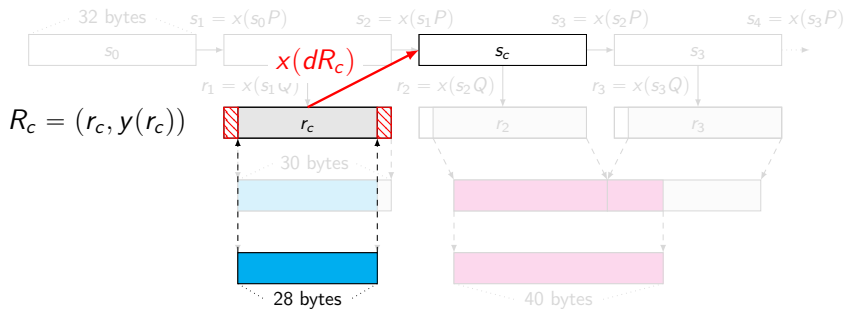
Points  $Q$  and  $P = dQ$  on an elliptic curve.



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

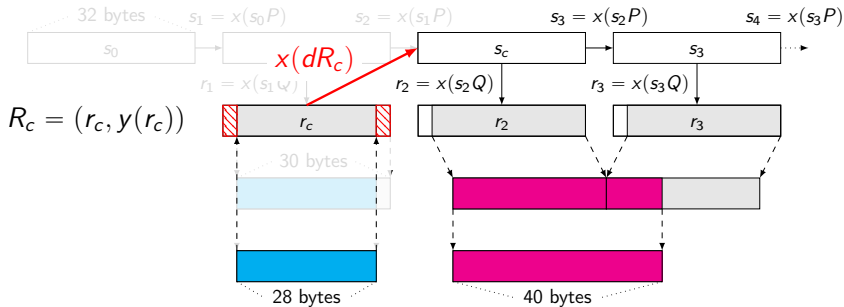


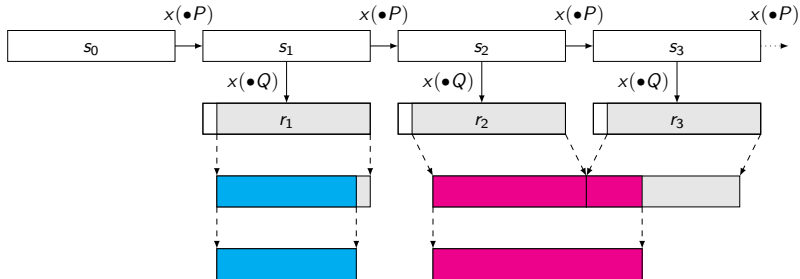
Points  $Q$  and  $P = dQ$  on an elliptic curve.

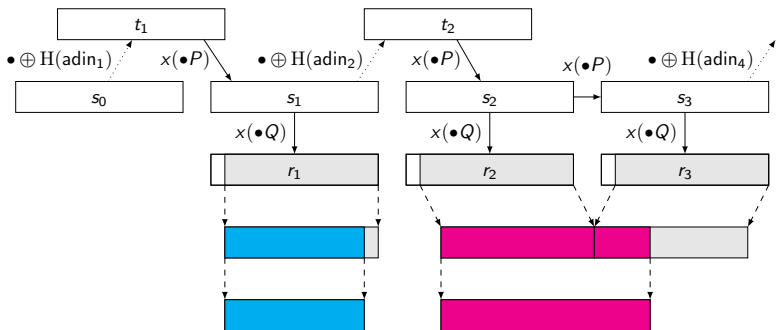


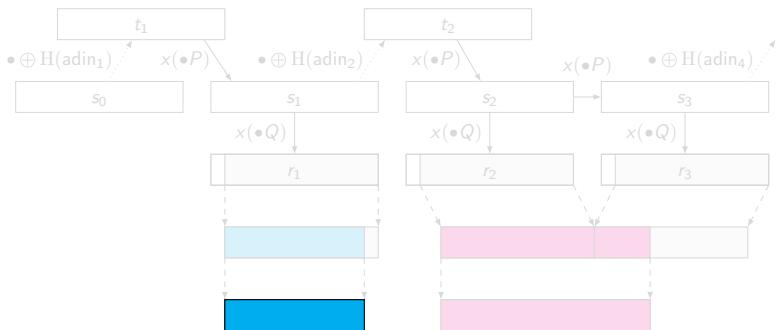


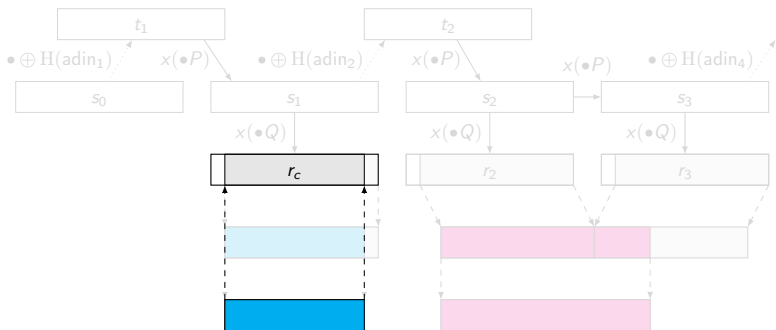
Points  $Q$  and  $P = dQ$  on an elliptic curve.

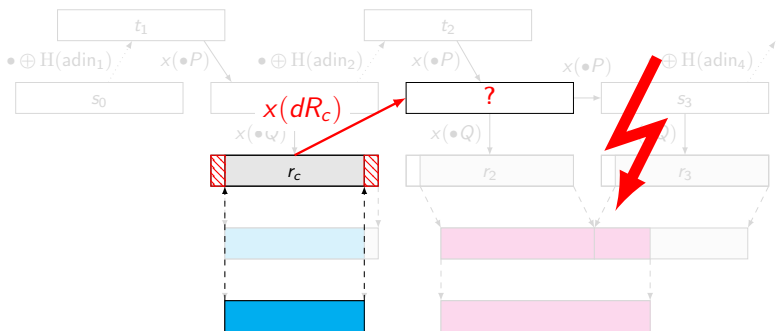


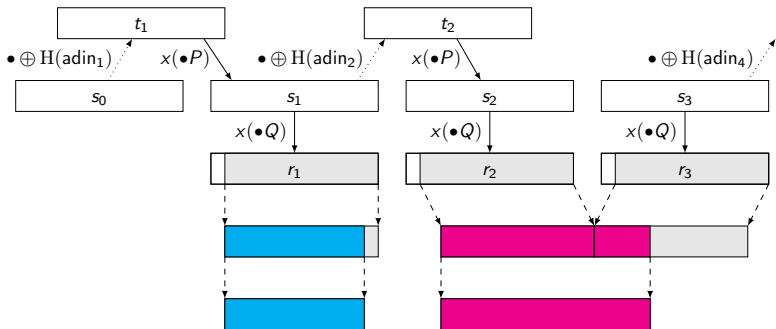


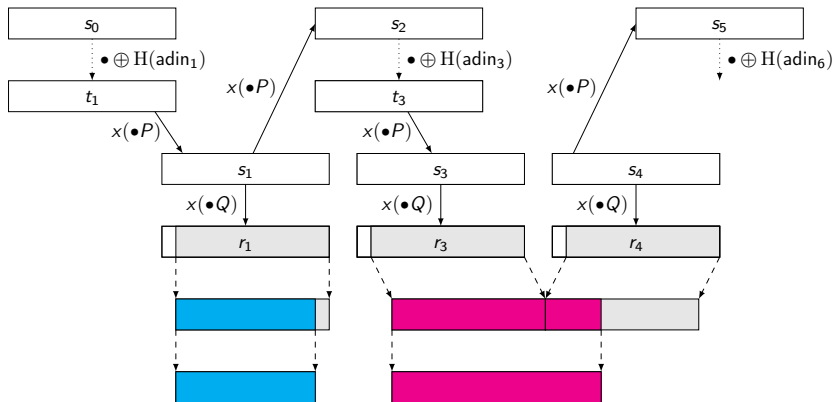




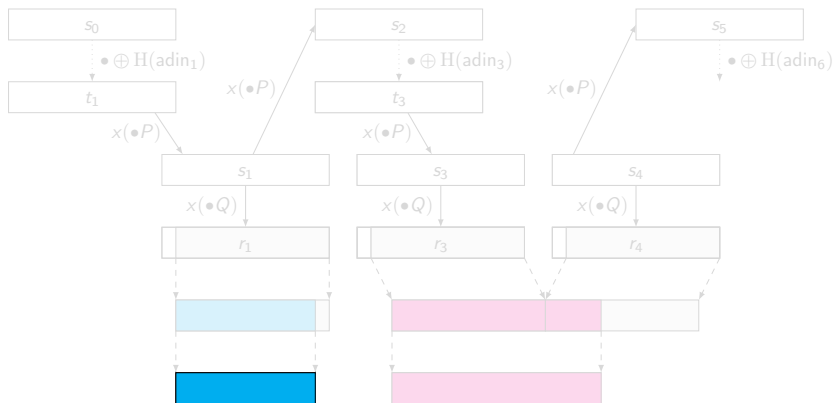


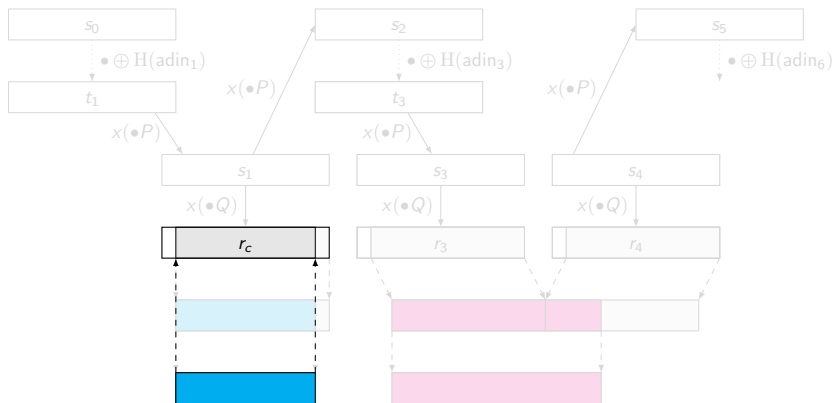


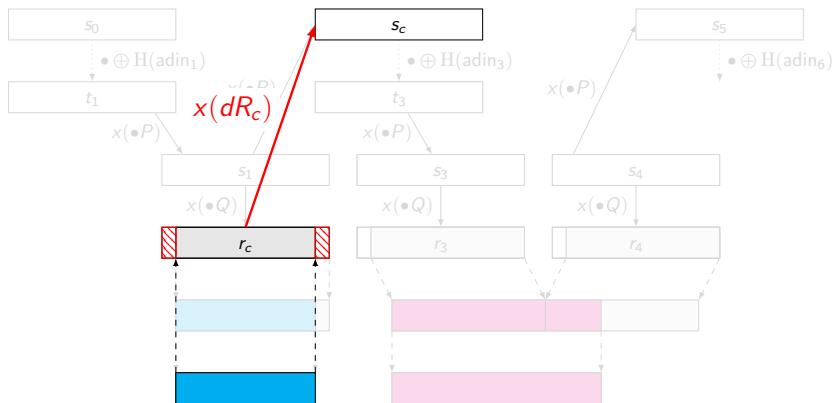


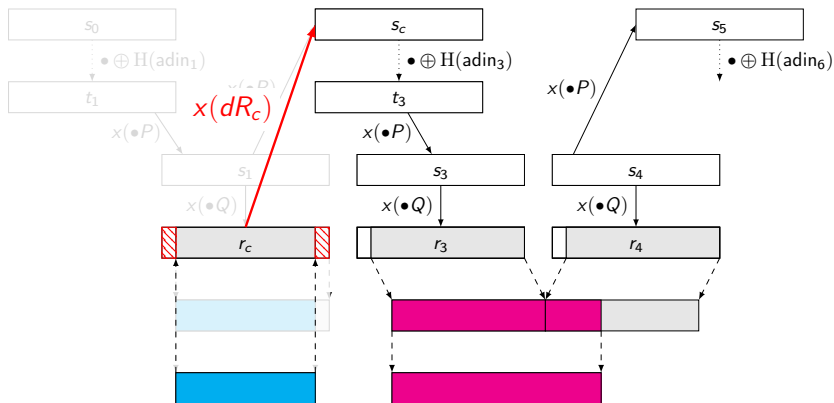












## Attack targets in our analysis:

- ▶ RSA's BSAFE
  - ▶ RSA BSAFE Share for Java (BSAFE Java)
  - ▶ RSA BSAFE Share for C and C++ (BSAFE C)
- ▶ Microsoft's SChannel
- ▶ OpenSSL

We replaced the points  $P$  and  $Q$  with points where we know the back-door computation.

server random

ECDHE priv. key

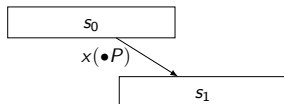
ECDSA nonce

$s_0$

server random

ECDHE priv. key

ECDSA nonce

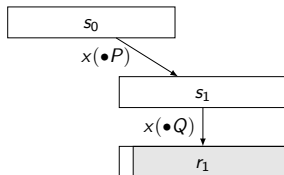


server random

ECDHE priv. key

ECDSA nonce

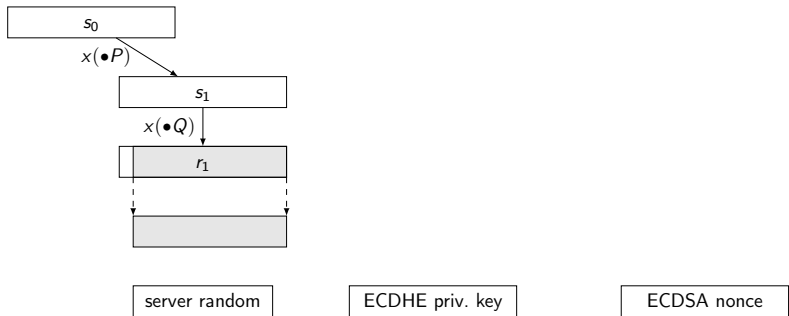


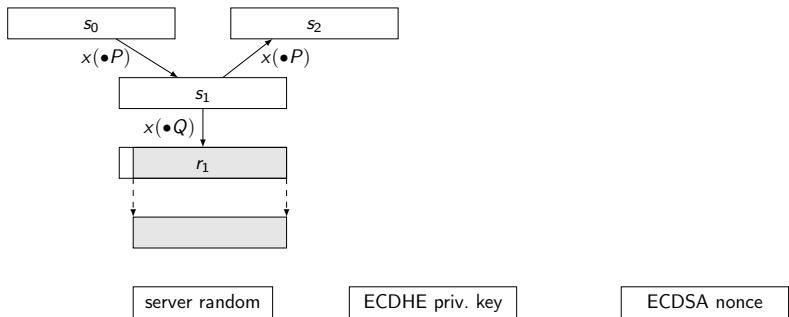


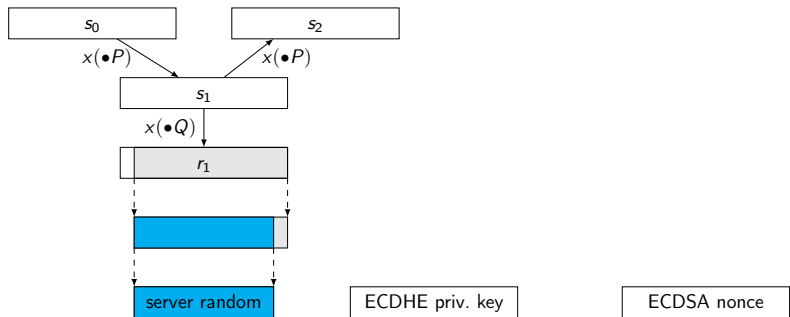
server random

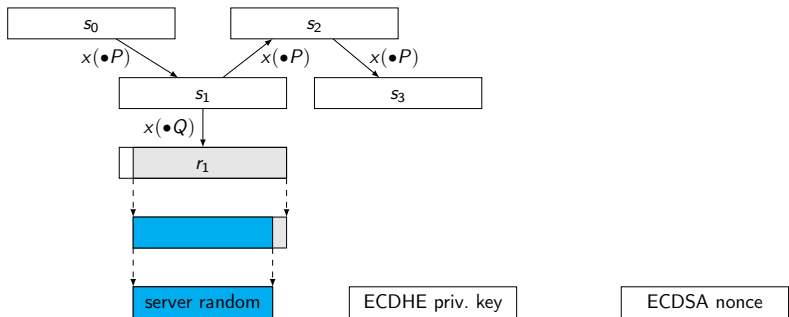
ECDHE priv. key

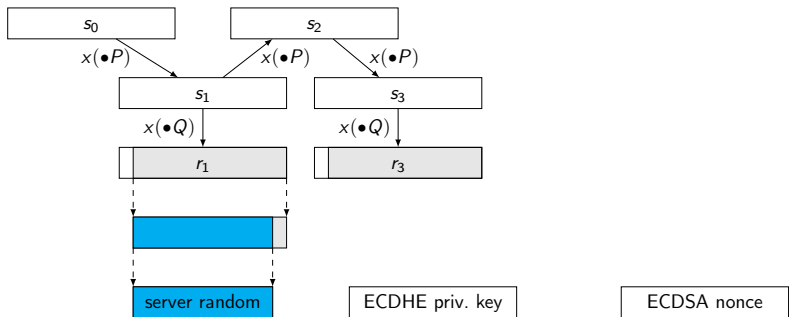
ECDSA nonce

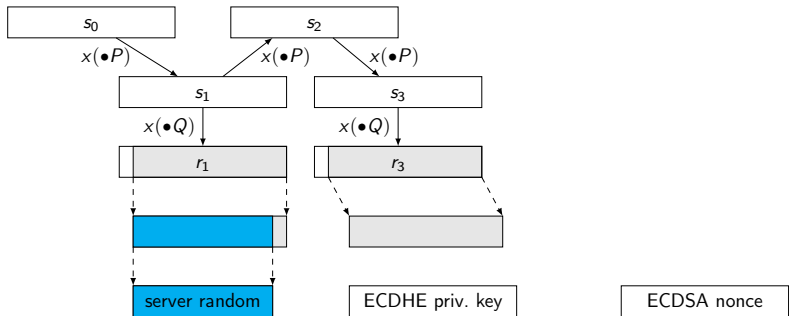


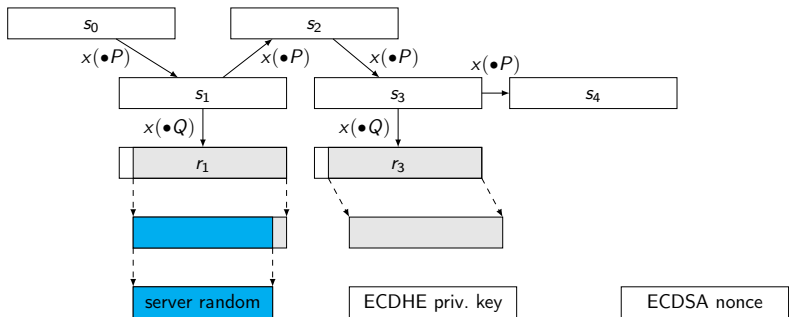




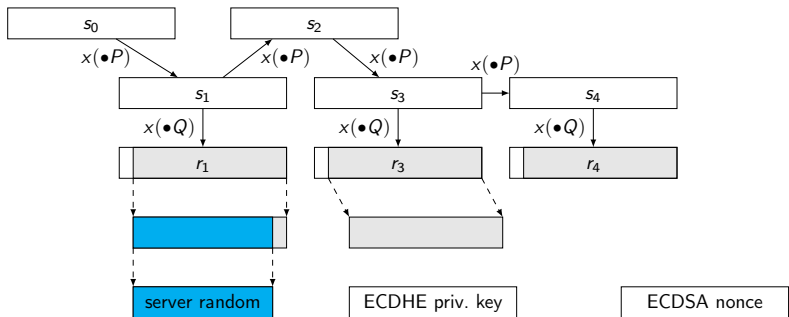


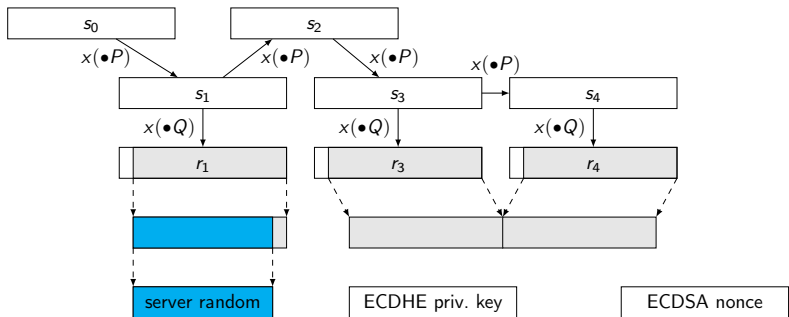


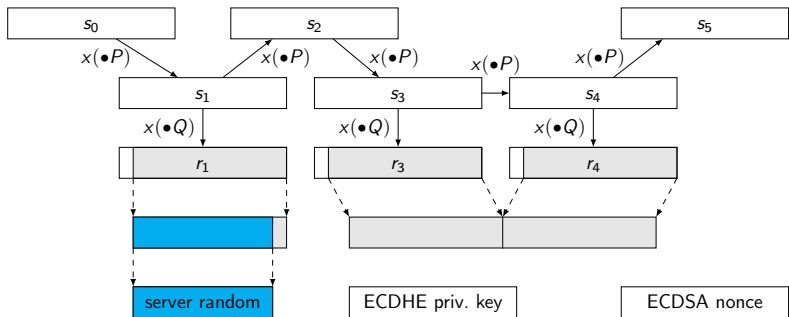


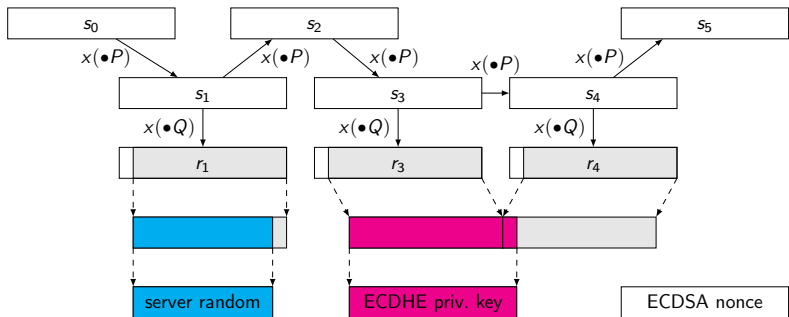


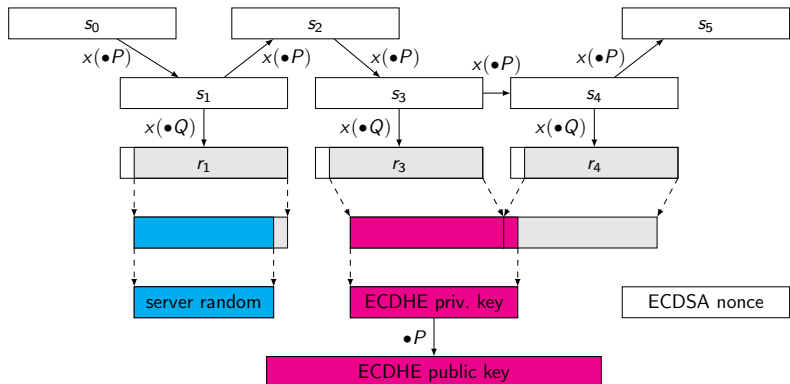


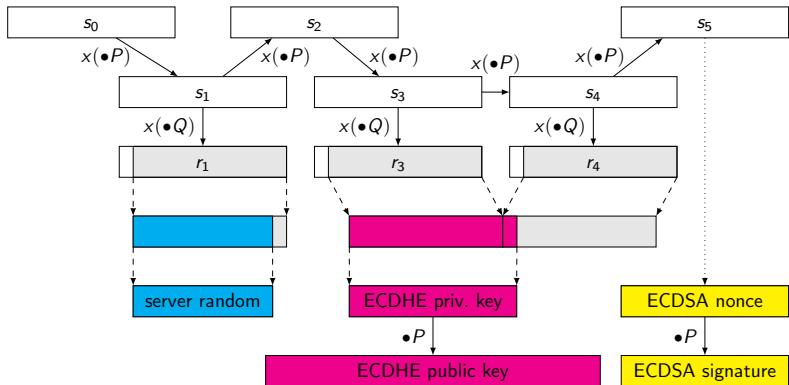


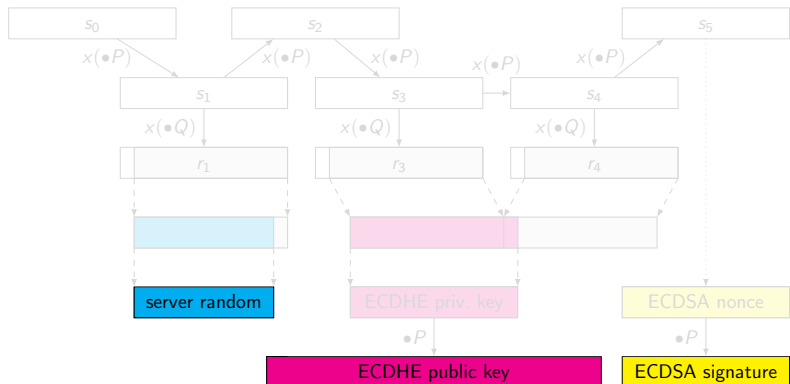


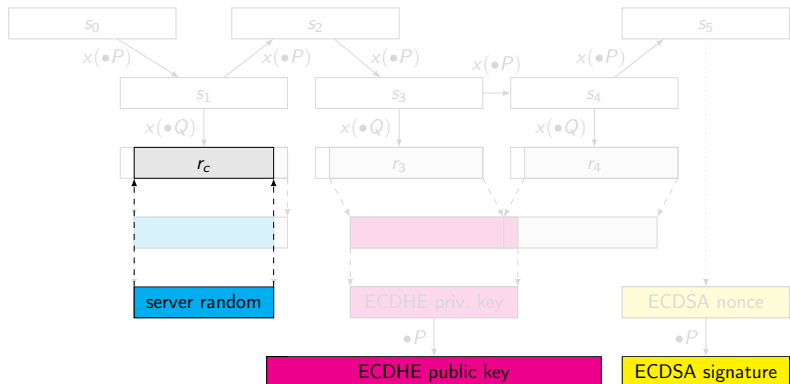




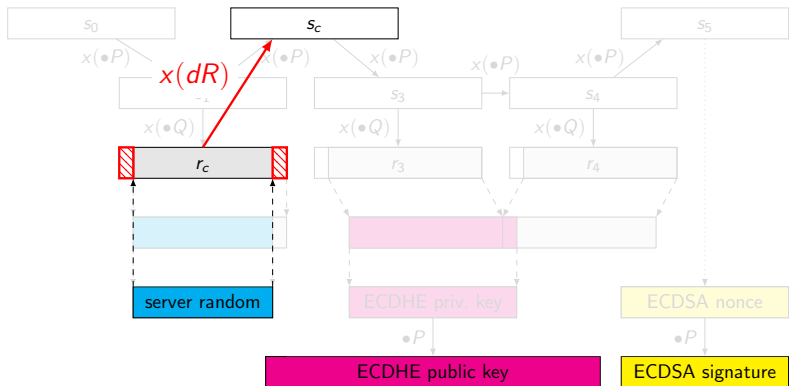


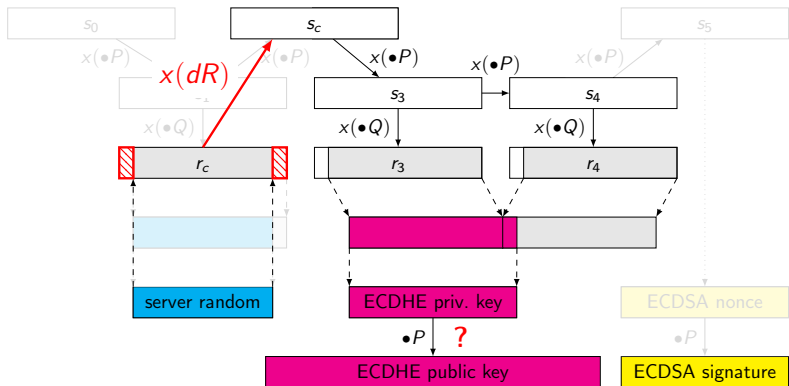


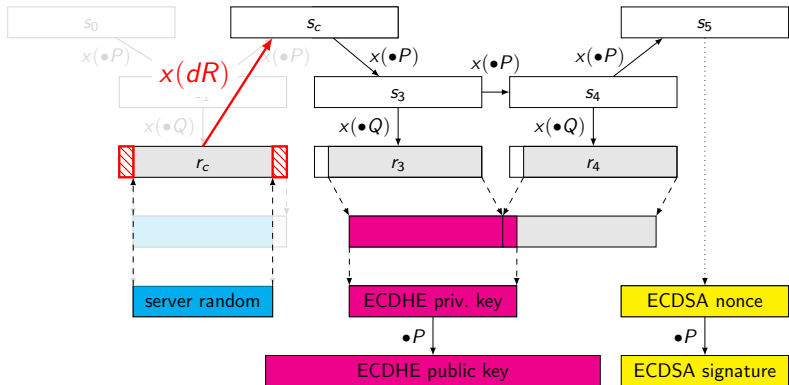


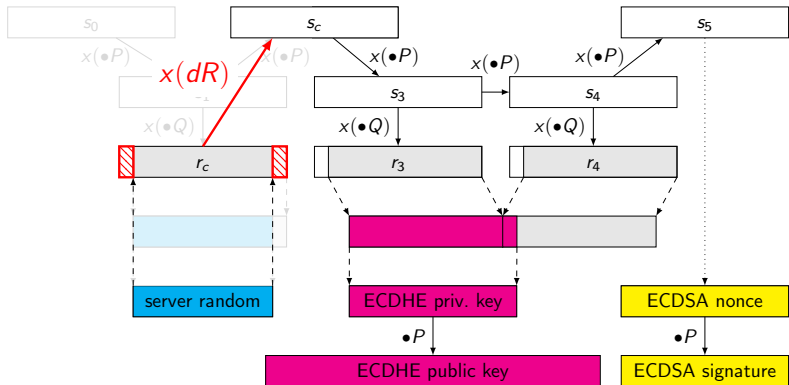




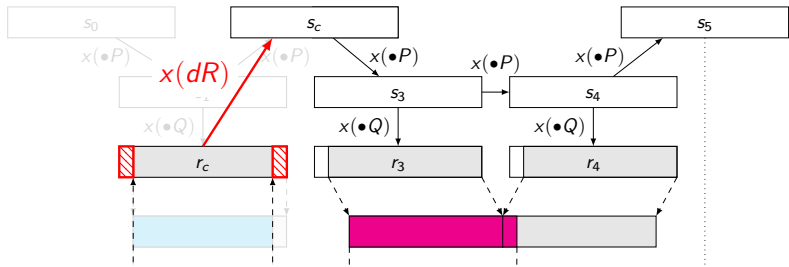




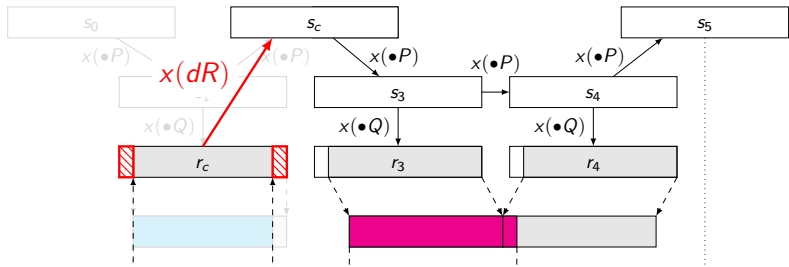




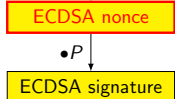
average cost:  $2^{31}(C_v + 5C_f)$



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



**Exposes longterm secret key!  
Impersonation attack possible!**



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

Attack	Intel Xeon CPU		AMD Cluster
	Avg. Time (min)	# for 1s	Tot. Time (min)
BSAFE-C v1.1	0.26	16	0.04
BSAFE-Java v1.1	641	38,500	63.96
SChannel I	619	37,100	62.97
SChannel II	1,760	106,000	182.64
OpenSSL-fixed I	0.04	3	0.02
OpenSSL-fixed II	707	44,200	83.32
OpenSSL-fixed III	$2^k \cdot 707$	$2^k \cdot 44,200$	$2^k \cdot 83.32$

Attack	Intel Xeon CPU		AMD Cluster
	Avg. Time (min)	# for 1s	Tot. Time (min)
BSAFE-C v1.1	0.26	16	0.04
BSAFE-Java v1.1	641	38,500	63.96
SChannel I	619	37,100	62.97
SChannel II	1,760	106,000	182.64
OpenSSL-fixed I	0.04	3	0.02
OpenSSL-fixed II	707	44,200	83.32
OpenSSL-fixed III	$2^k \cdot 707$	$2^k \cdot 44,200$	$2^k \cdot 83.32$



## Draft for a proposed TLS extension named “Extended Random”:

- ▶ allows client to request up to  $2^{16}$  random bytes
- ▶ has a weak motivation:  
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 insufficient.
- ▶ was co-authored by an employee of NSA

## Draft for a proposed TLS extension named “Extended Random”:

- ▶ allows client to request up to  $2^{16}$  random bytes
- ▶ has a weak motivation:  
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 insufficient.
- ▶ was co-authored by an employee of NSA

Makes Dual EC even more vulnerable!

## Dual EC — a standardized back door:

- ▶ (co-)authored by NSA

## Dual EC — a standardized back door:

- ▶ (co-)authored by NSA
- ▶ may contain a back door (can neither be proven nor disproven)

## Dual EC — a standardized back door:

- ▶ (co-)authored by NSA
- ▶ may contain a back door (can neither be proven nor disproven)
- ▶ allows the back-door owner to compute all future random outputs

## Dual EC — a standardized back door:

- ▶ (co-)authored by NSA
- ▶ may contain a back door (can neither be proven nor disproven)
- ▶ allows the back-door owner to compute all future random outputs
- ▶ makes flaw in DSS a back door, allows impersonation

## Dual EC — a standardized back door:

- ▶ (co-)authored by NSA
- ▶ may contain a back door (can neither be proven nor disproven)
- ▶ allows the back-door owner to compute all future random outputs
- ▶ makes flaw in DSS a back door, allows impersonation
- ▶ proven to be practical in various TLS libraries

## Dual EC — a standardized back door:

- ▶ (co-)authored by NSA
- ▶ may contain a back door (can neither be proven nor disproven)
- ▶ allows the back-door owner to compute all future random outputs
- ▶ makes flaw in DSS a back door, allows impersonation
- ▶ proven to be practical in various TLS libraries
- ▶ was default RNG in RSA's BSAFE library



## Dual EC — a standardized back door:

- ▶ (co-)authored by NSA
- ▶ may contain a back door (can neither be proven nor disproven)
- ▶ allows the back-door owner to compute all future random outputs
- ▶ makes flaw in DSS a back door, allows impersonation
- ▶ proven to be practical in various TLS libraries
- ▶ was default RNG in RSA's BSAFE library
- ▶ back door becomes even stronger with proposed Extended Random

## Dual EC — a standardized back door:

- ▶ (co-)authored by NSA
- ▶ may contain a back door (can neither be proven nor disproven)
- ▶ allows the back-door owner to compute all future random outputs
- ▶ makes flaw in DSS a back door, allows impersonation
- ▶ proven to be practical in various TLS libraries
- ▶ was default RNG in RSA's BSAFE library
- ▶ back door becomes even stronger with proposed Extended Random

How to fix it?

## Dual EC — a standardized back door:

- ▶ (co-)authored by NSA
- ▶ may contain a back door (can neither be proven nor disproven)
- ▶ allows the back-door owner to compute all future random outputs
- ▶ makes flaw in DSS a back door, allows impersonation
- ▶ proven to be practical in various TLS libraries
- ▶ was default RNG in RSA's BSAFE library
- ▶ back door becomes even stronger with proposed Extended Random

**Don't use Dual EC!**

## Additional information:

<https://projectbullrun.org/dual-ec/>