Backdoors

always

backfire

Tanja Lange

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

21 February 2019

Vodafone Greece

- Vodafone Greece used Ericsson switches to manage phone network.
  - Purchased without legal interception capabilities (=wiretaps).
  - Upgrade in early 2003 included remote-control equipment subsystem (RES).
- Some time after this and before 24 January 2005, "somebody" used the RES capabilities and put wiretaps on >100 phones, including Greek prime minister and his wife, ministers of national defense, foreign affairs, and justice, ...

Vodafone Greece

- Vodafone Greece used Ericsson switches to manage phone network.
  - Purchased without legal interception capabilities (=wiretaps).
  - Upgrade in early 2003 included remote-control equipment subsystem (RES).
- Some time after this and before 24 January 2005, "somebody" used the RES capabilities and put wiretaps on >100 phones, including Greek prime minister and his wife, ministers of national defense, foreign affairs, and justice, ...
- 9 March 2005 Costas Tsalikidis, telecommunications engineer in charge of network planning at Vodafone Greece, found hanged in his apartment, an apparent suicide.

Vodafone Greece

- Vodafone Greece used Ericsson switches to manage phone network.
  - Purchased without legal interception capabilities (=wiretaps).
  - Upgrade in early 2003 included remote-control equipment subsystem (RES).
- Some time after this and before 24 January 2005, "somebody" used the RES capabilities and put wiretaps on >100 phones, including Greek prime minister and his wife, ministers of national defense, foreign affairs, and justice, ...
- 8 March 2005 Vodafone deactivates rogue software.
- 9 March 2005 Costas Tsalikidis, telecommunications engineer in charge of network planning at Vodafone Greece, found hanged in his apartment, an apparent suicide.
- 10 March Vodafone CEO informs prime minister.

Vodafone Greece

- Vodafone Greece used Ericsson switches to manage phone network.
  - Purchased without legal interception capabilities (=wiretaps).
  - Upgrade in early 2003 included remote-control equipment subsystem (RES).
- Some time after this and before 24 January 2005, "somebody" used the RES capabilities and put wiretaps on >100 phones, including Greek prime minister and his wife, ministers of national defense, foreign affairs, and justice, ...
- 8 March 2005 Vodafone deactivates rogue software.
- 9 March 2005 Costas Tsalikidis, telecommunications engineer in charge of network planning at Vodafone Greece, found hanged in his apartment, an apparent suicide.
- 10 March Vodafone CEO informs prime minister.

https://spectrum.ieee.org/telecom/security/the-athens-affair

https://theintercept.com/2015/09/28/death-athens-rogue-nsa-operation/

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.

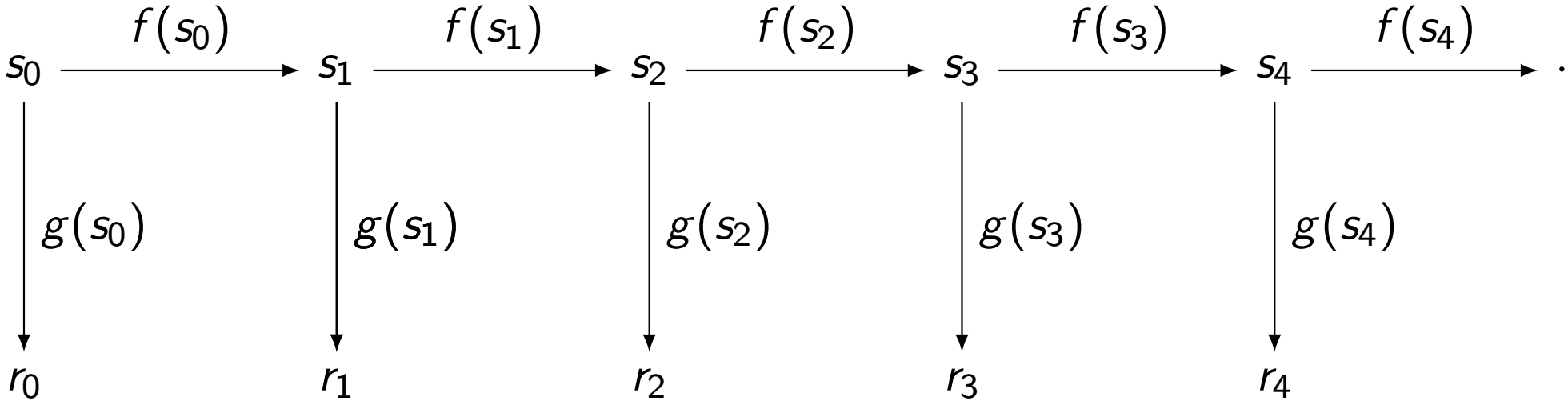
Snowden at SXSW:

> *[..] 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.*

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:



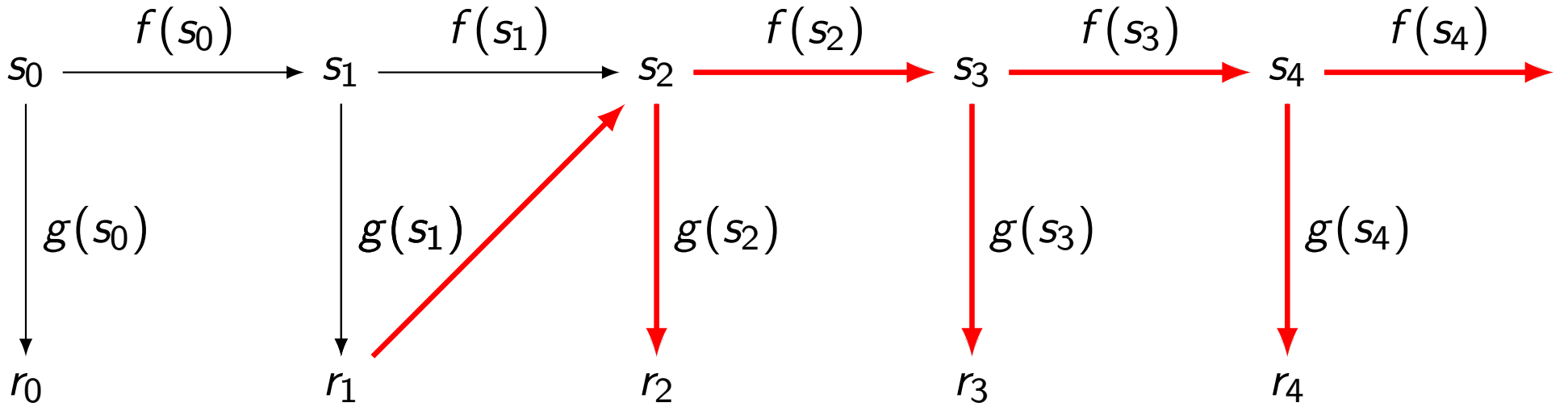
XXX's mission: Predict the "random" output bits.

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:



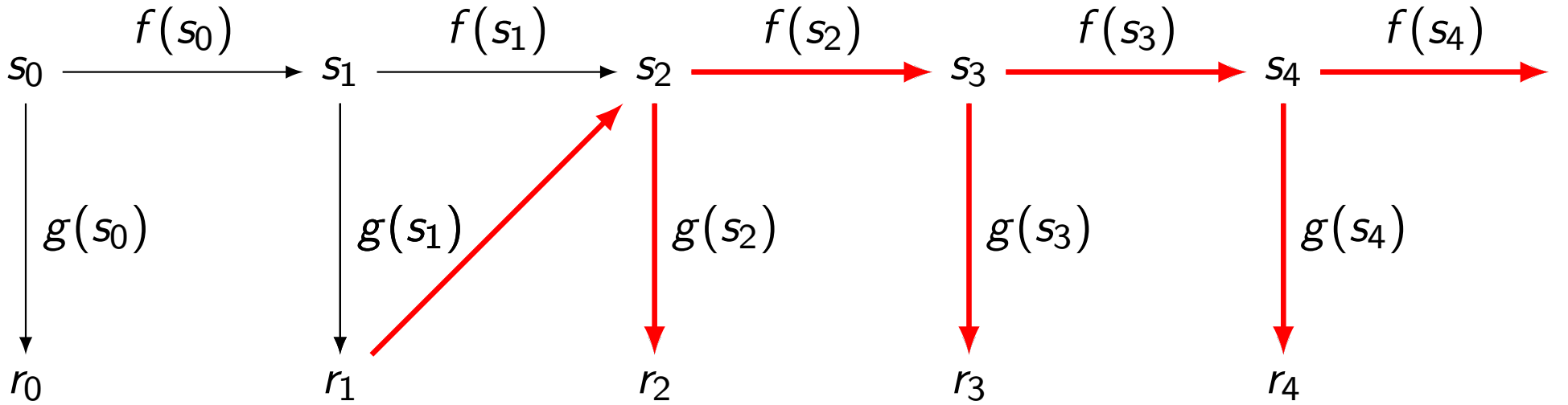
XXX's mission: Predict the "random" output bits.

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:



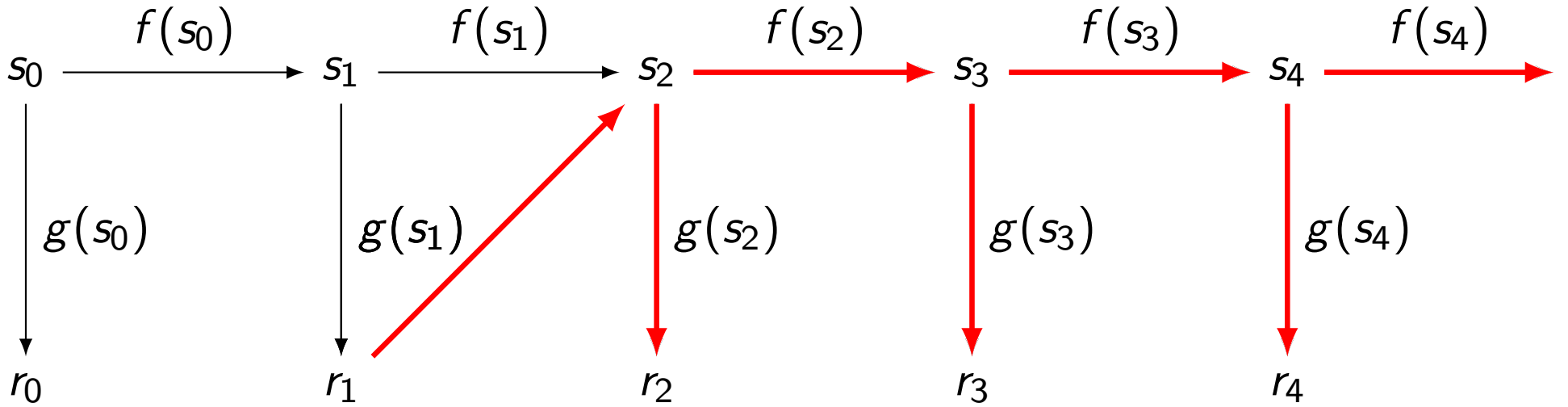
XXX's mission: Predict the "random" output bits.

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.

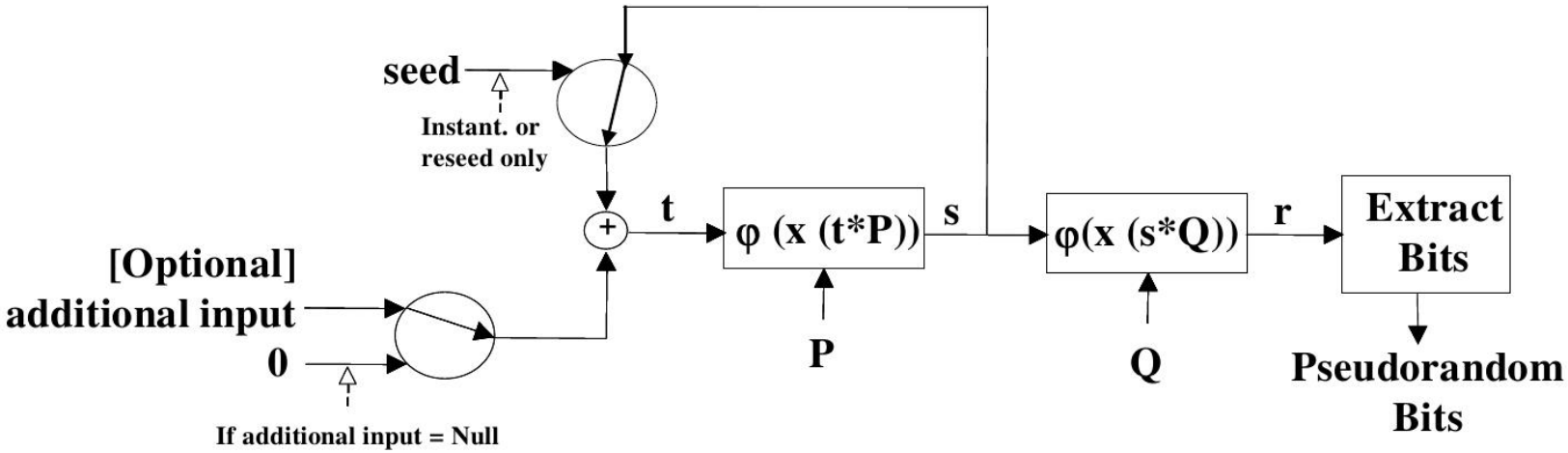
1. Create protocols that directly output r_n for some reason.
2. Design f, g with back door from r_n to s_{n+1}: i.e., get $f(s)$ from $g(s)$.
3. Standardize this design of f, g.
4. Convince users to switch to this design: e.g., publish "security proof".

Full TODO list

- Design: Construct a PRNG that secretly contains a back door.
- Evaluation: Publish statements that the PRNG is secure.
- Standardization: Edit standards to include the PRNG.
- Standardization maintenance: Monitor changes to the PRNG standard, and counteract changes that make the PRNG more difficult to exploit.
- Auxiliary standardization: If necessary modify other standards to make the PRNG easier to exploit.
- Selection, implementation, and deployment: Provide incentives to cryptographic libraries to implement this PRNG.
- Attack optimization: Reduce the cost of computation required to exploit the back door, through algorithmic improvements and through influencing the way the PRNG is used in practice.

DUAL_EC RNG: history part I

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



Extract gives all but the top 16 bits $\Rightarrow$ about 2^{15} points sQ match given string.

Claim:

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

DUAL_EC RNG: common public history part II

Various public warning signals:

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

DUAL_EC RNG: common public history part II

Various public warning signals:

- Gjøsteen (March 2006): output sequence is biased.
- Brown (March 2006): security "proof"
  "This proof makes essential use of Q being random." If d with $dQ = P$ is known then $dR_i = S_{i+1}$. Brown concludes that there might be distinguisher.
- Sidorenko & Schoenmakers (May 2006): output sequence is even more biased. Answer: Too late to change, already implemented.
- Included in standards ISO 18031 (2005), NIST SP 800-90 (2006), ANSI X9.82 (2007).
- 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.

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; recommends against using Dual_EC.
RSA suggests changing default in BSAFE.

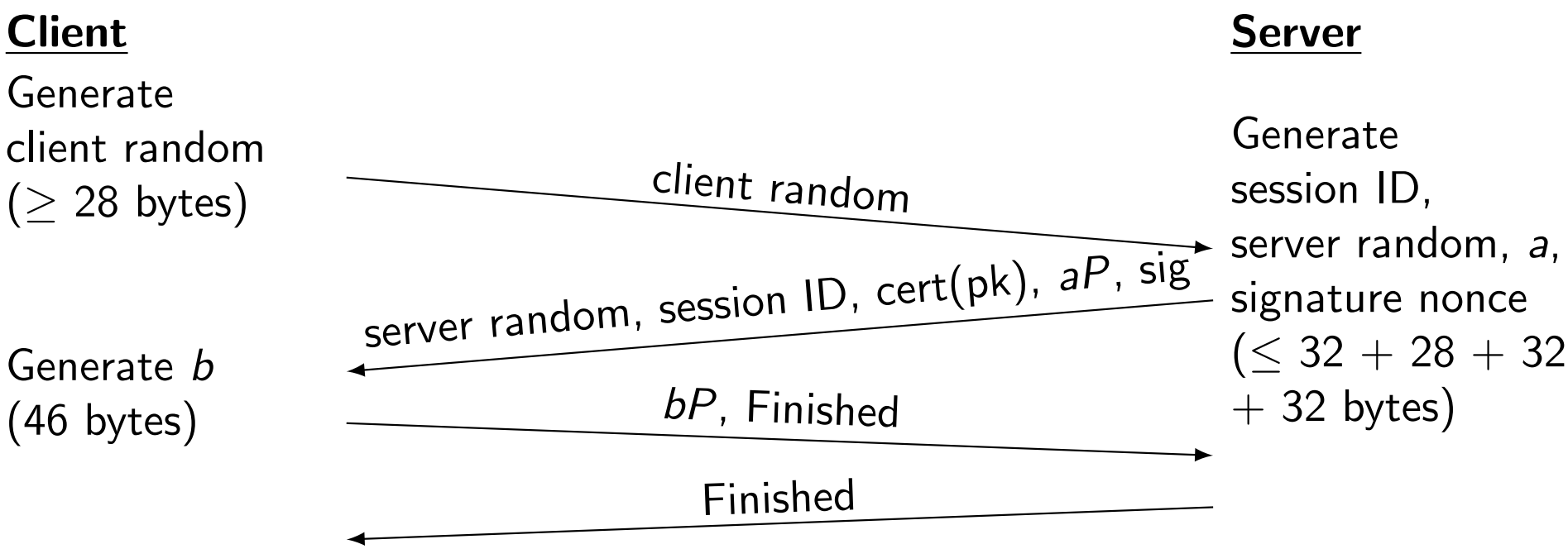
21 April 2014 NIST removes Dual EC from the standard.

SSL/TLS/HTTPS – internet security protocols

How are RNGs actually used? Do implementations actually leak enough of r_n?

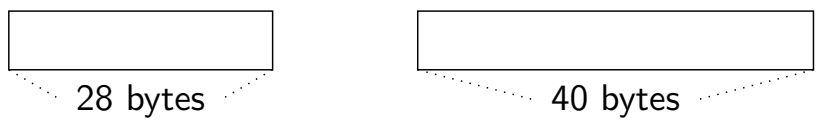
SSL/TLS/HTTPS – internet security protocols

How are RNGs actually used? Do implementations actually leak enough of r_n?



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

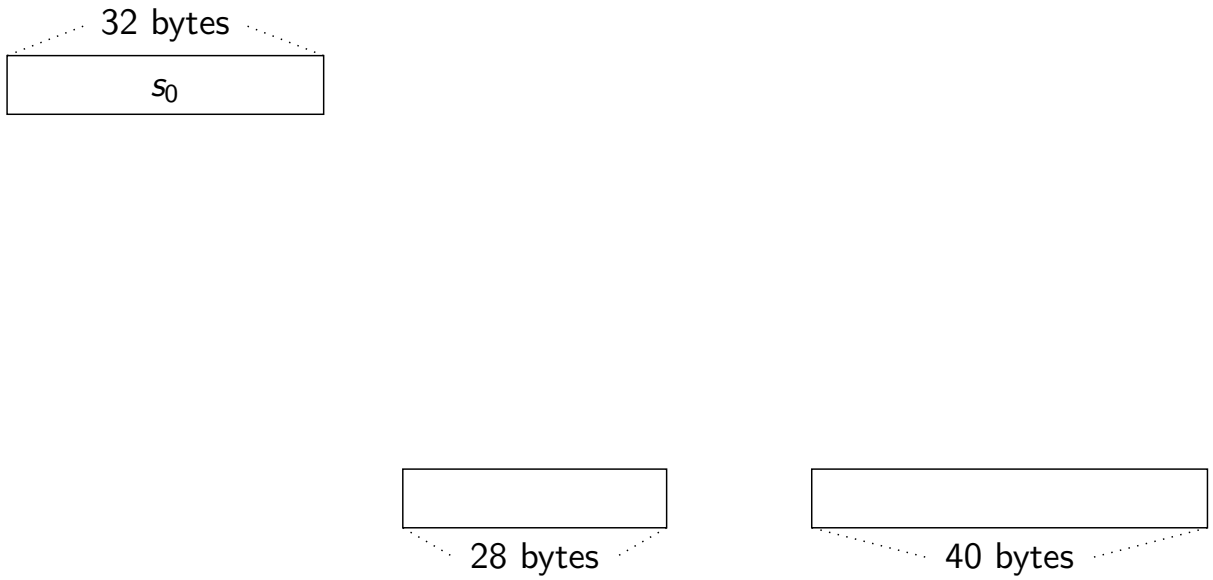
Dual EC in TLS



Graphic thanks to Ruben Niederhagen.

Dual EC in TLS

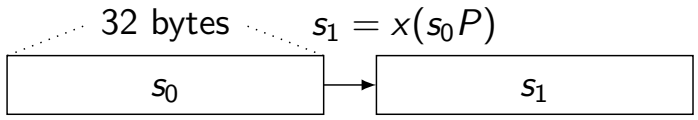
Points Q and P on an elliptic curve.

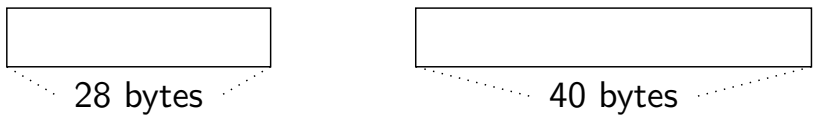


Graphic thanks to Ruben Niederhagen.

Dual EC in TLS

Points Q and P on an elliptic curve.

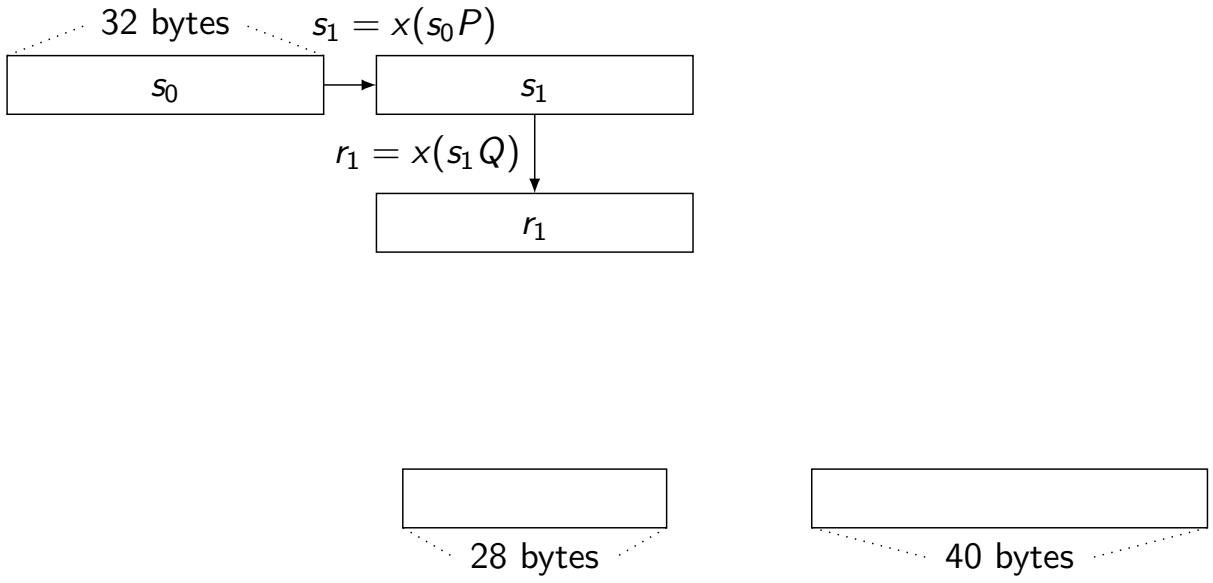




Graphic thanks to Ruben Niederhagen.

Dual EC in TLS

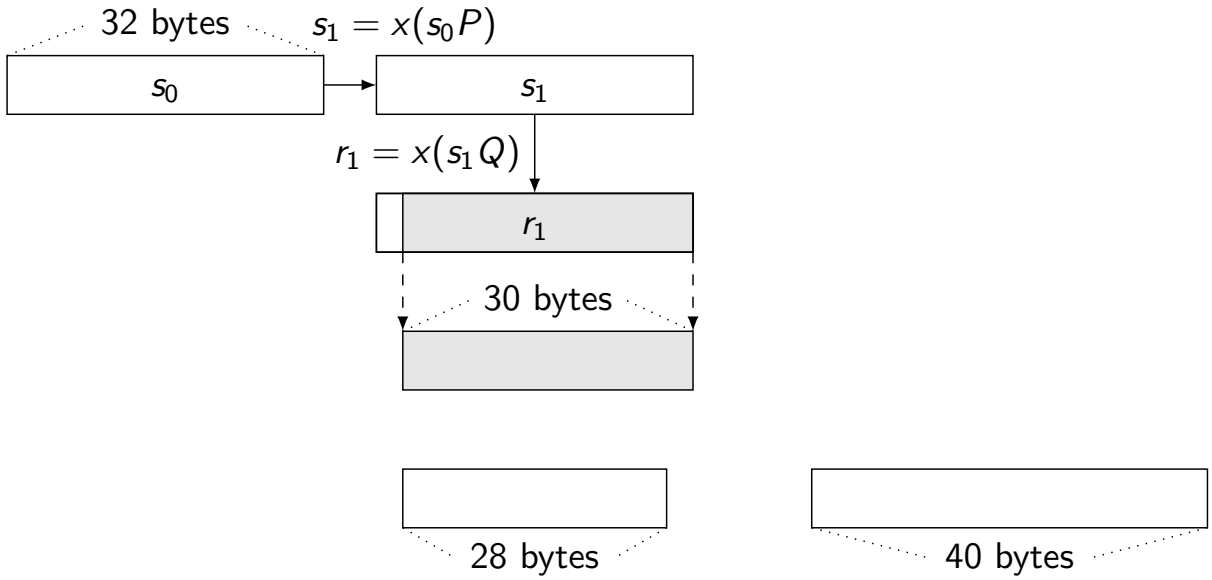
Points Q and P on an elliptic curve.



Graphic thanks to Ruben Niederhagen.

Dual EC in TLS

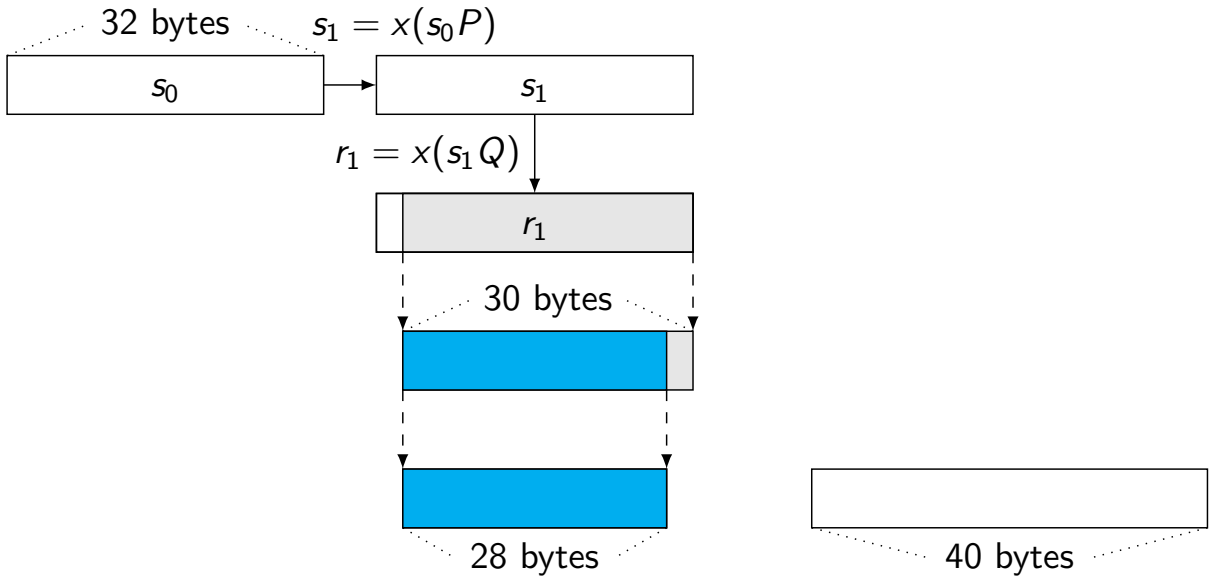
Points Q and P on an elliptic curve.



Graphic thanks to Ruben Niederhagen.

Dual EC in TLS

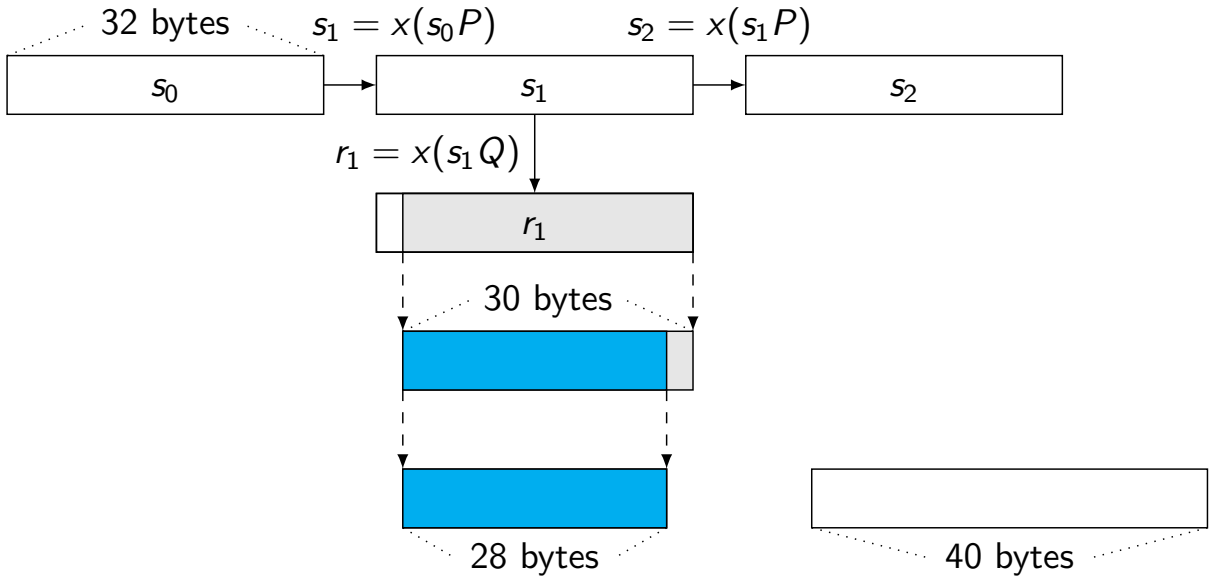
Points Q and P on an elliptic curve.



Graphic thanks to Ruben Niederhagen.

Dual EC in TLS

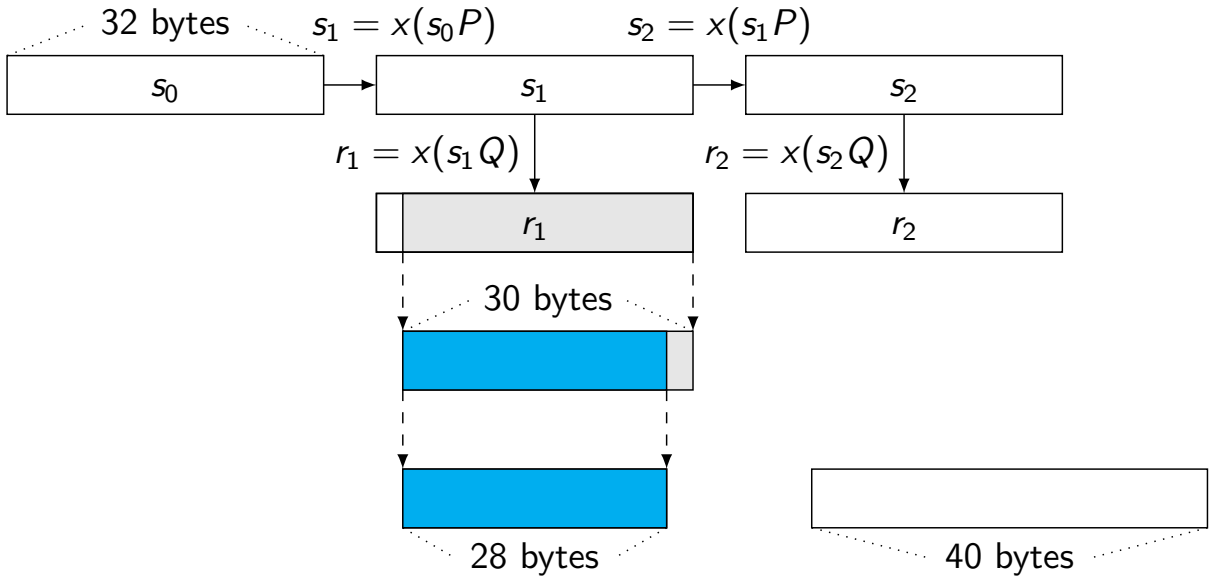
Points Q and P on an elliptic curve.



Graphic thanks to Ruben Niederhagen.

Dual EC in TLS

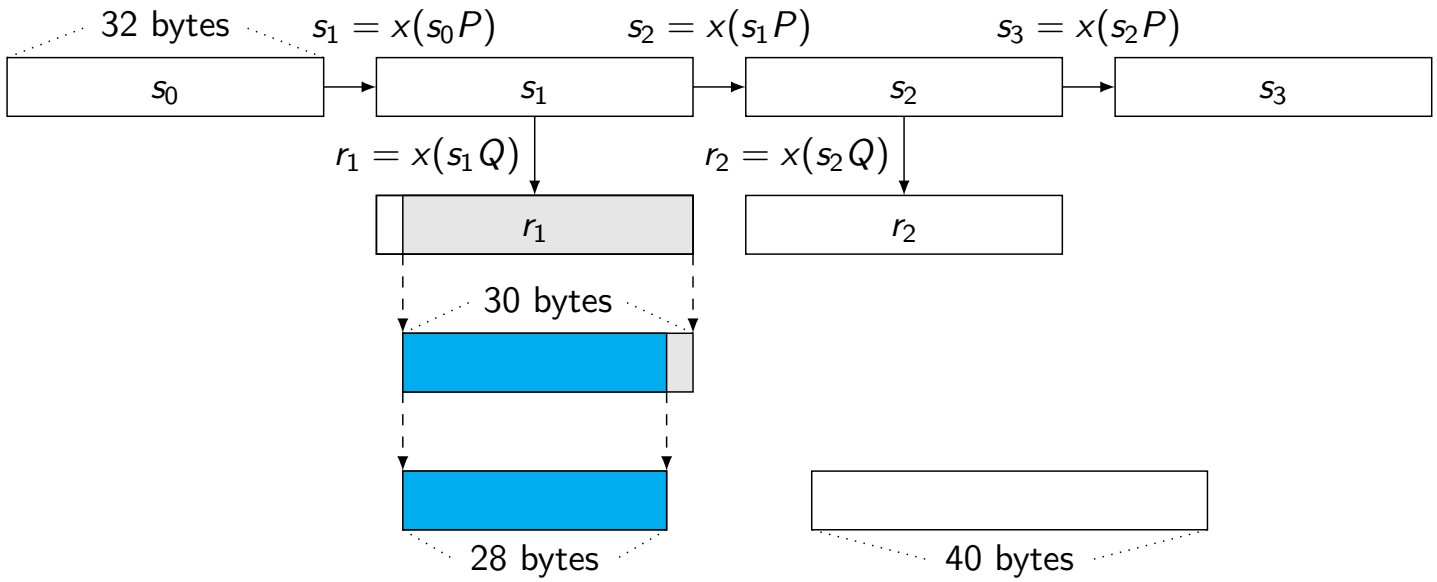
Points Q and P on an elliptic curve.



Graphic thanks to Ruben Niederhagen.

Dual EC in TLS

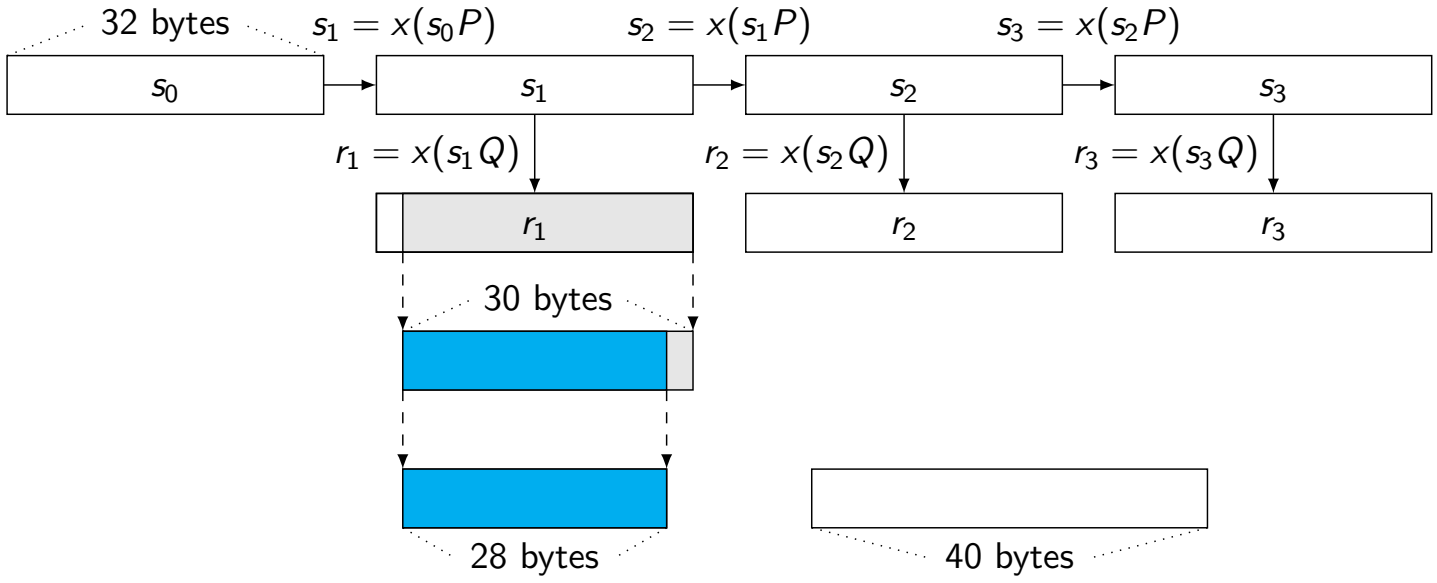
Points Q and P on an elliptic curve.



Graphic thanks to Ruben Niederhagen.

Dual EC in TLS

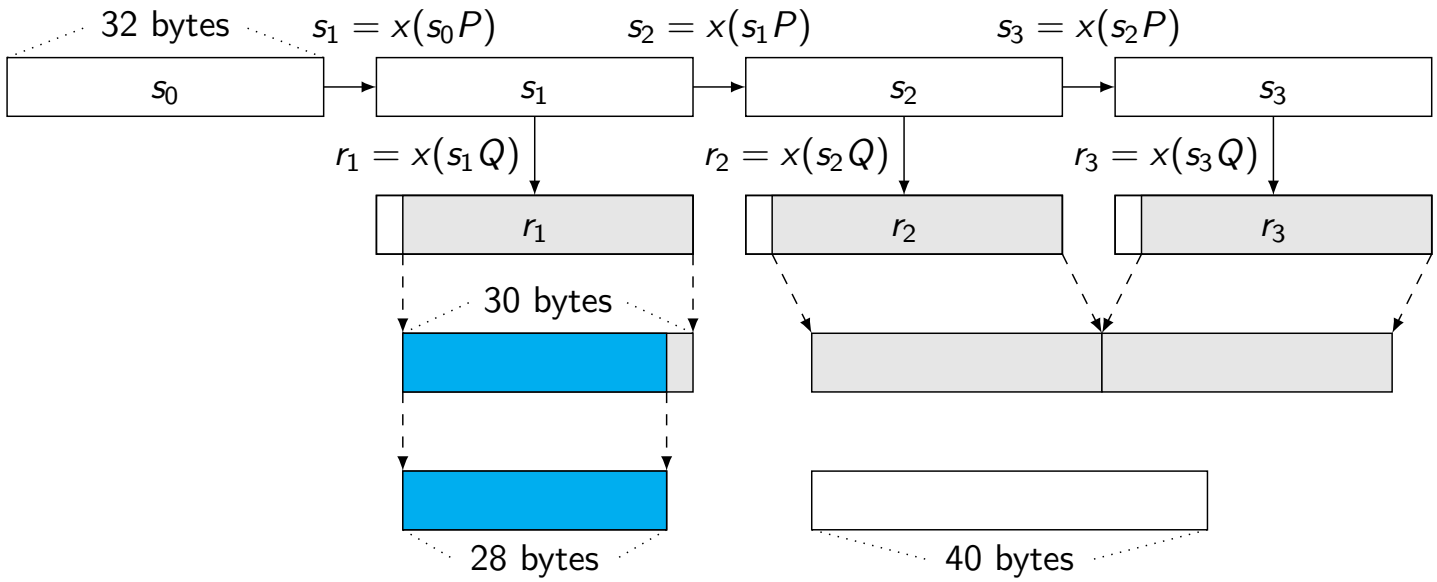
Points Q and P on an elliptic curve.



Graphic thanks to Ruben Niederhagen.

Dual EC in TLS

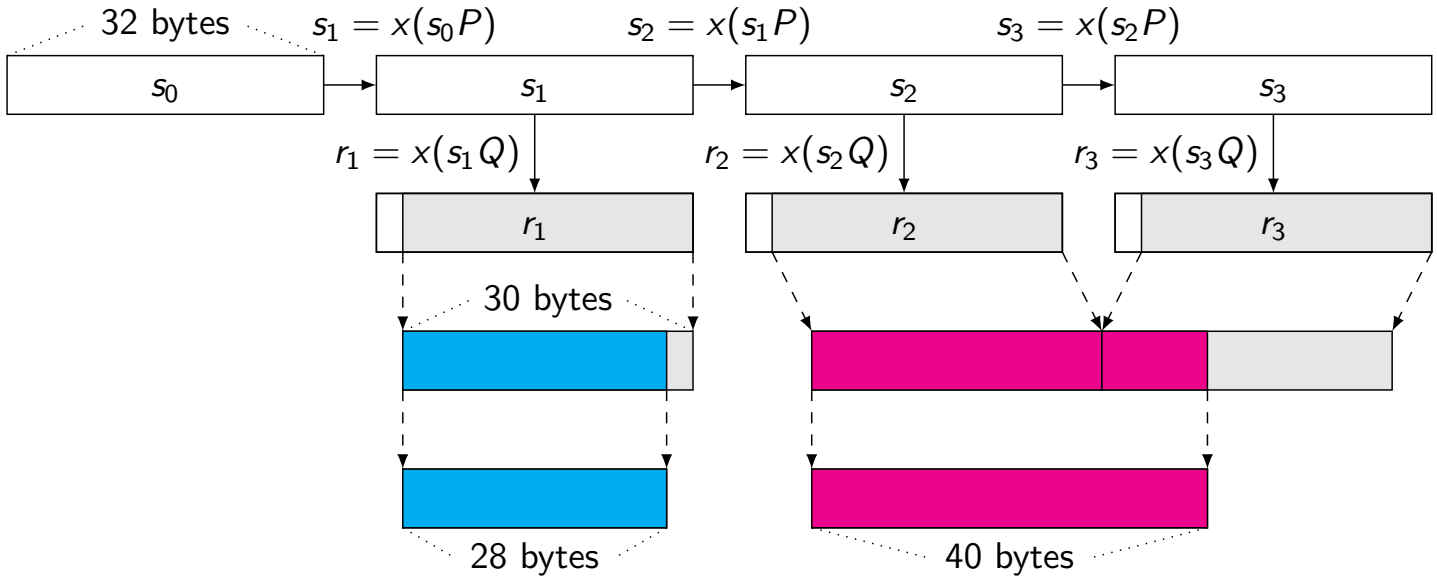
Points Q and P on an elliptic curve.



Graphic thanks to Ruben Niederhagen.

Dual EC in TLS

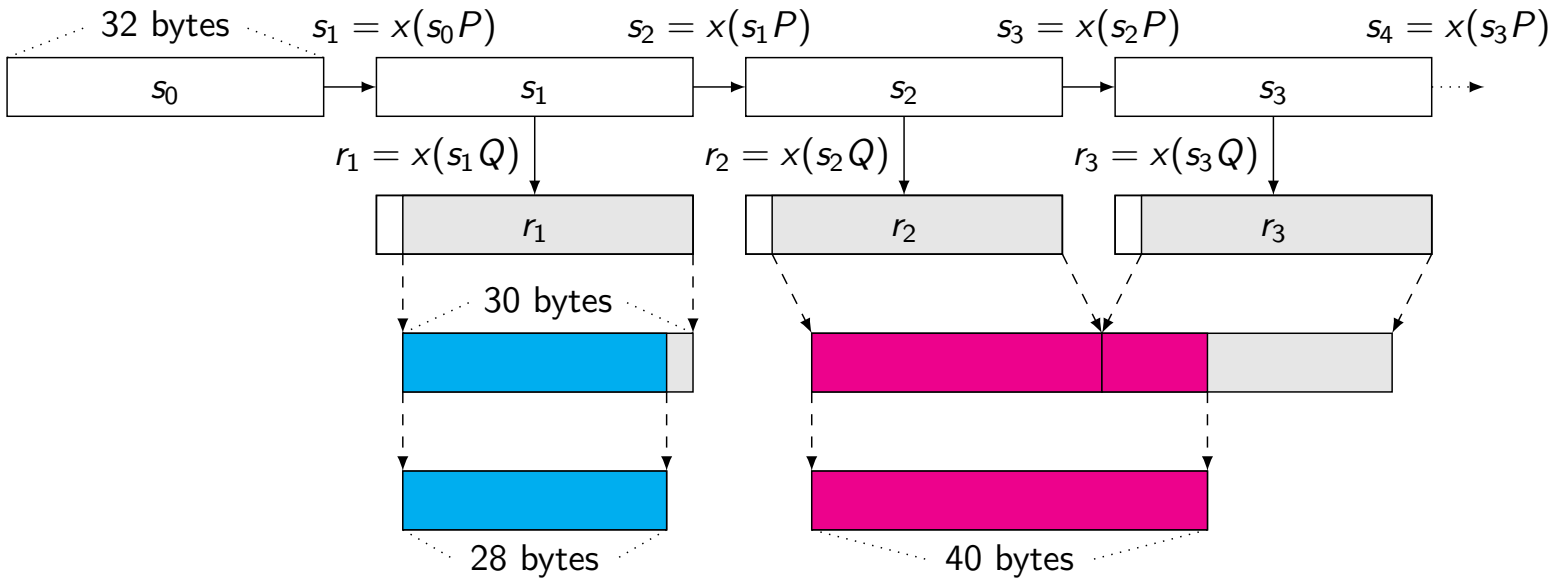
Points Q and P on an elliptic curve.



Graphic thanks to Ruben Niederhagen.

Dual EC in TLS

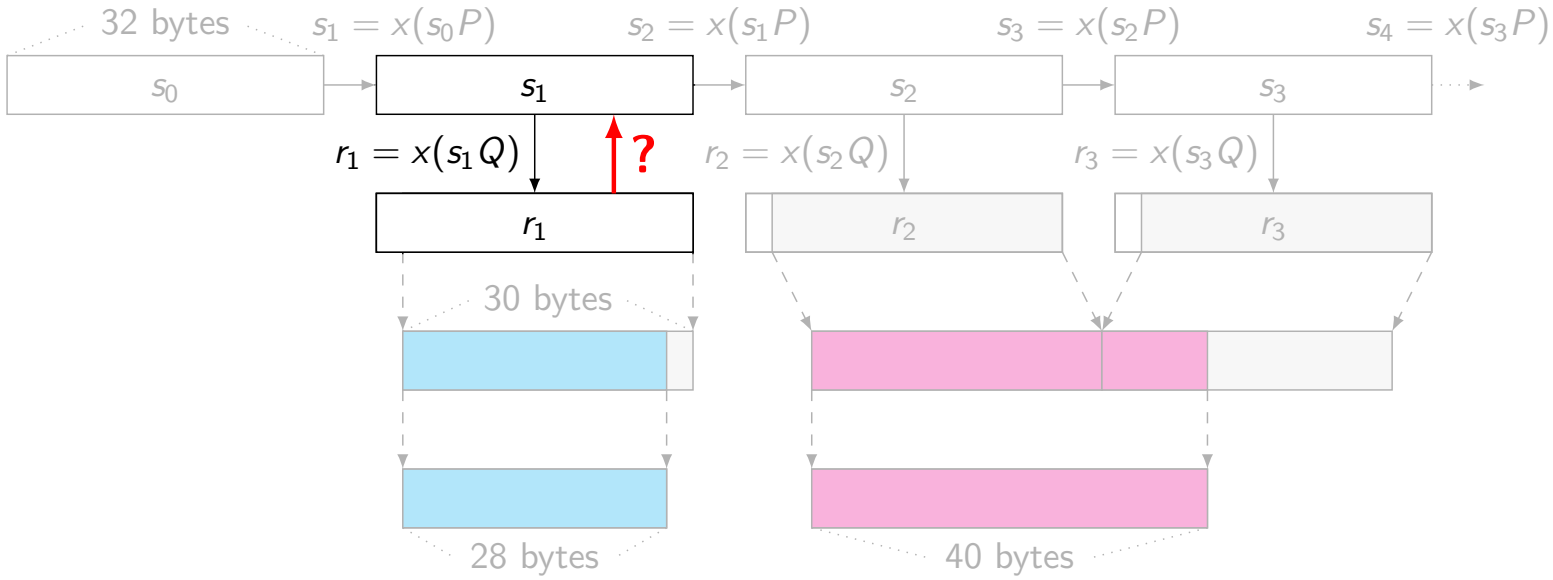
Points Q and P on an elliptic curve.



Graphic thanks to Ruben Niederhagen.

Dual EC in TLS

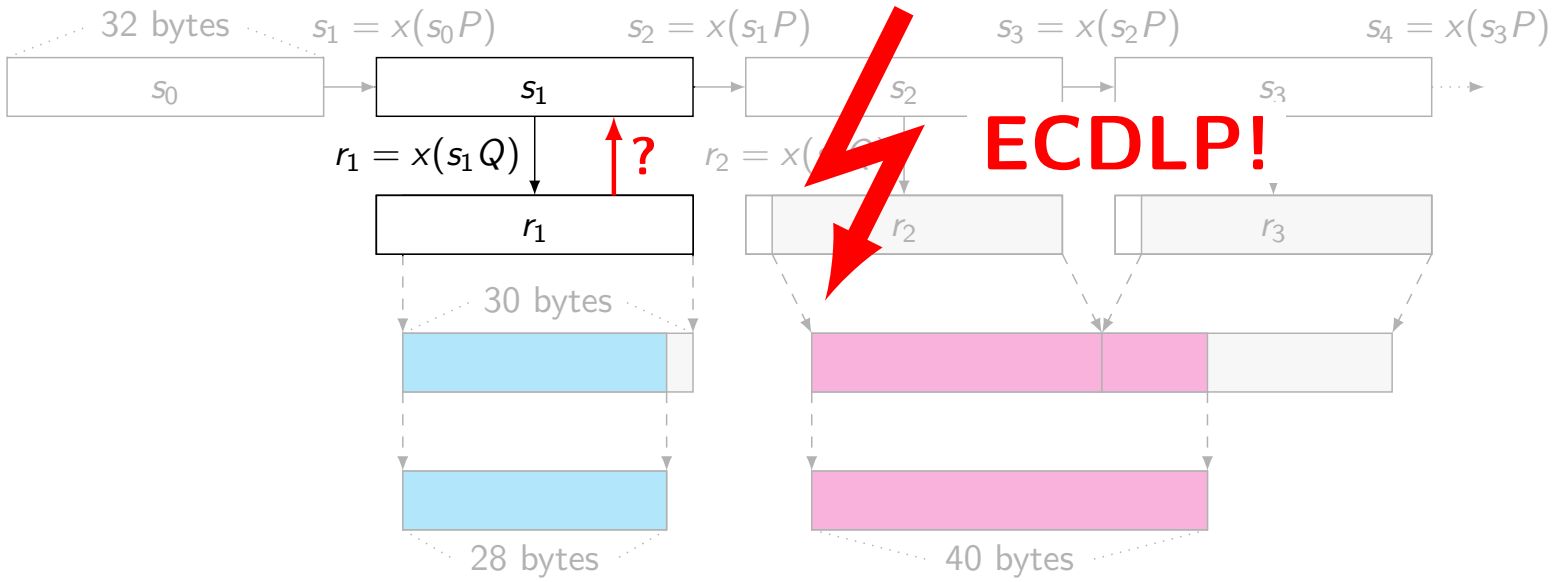
Points Q and P on an elliptic curve.



Graphic thanks to Ruben Niederhagen.

Dual EC in TLS

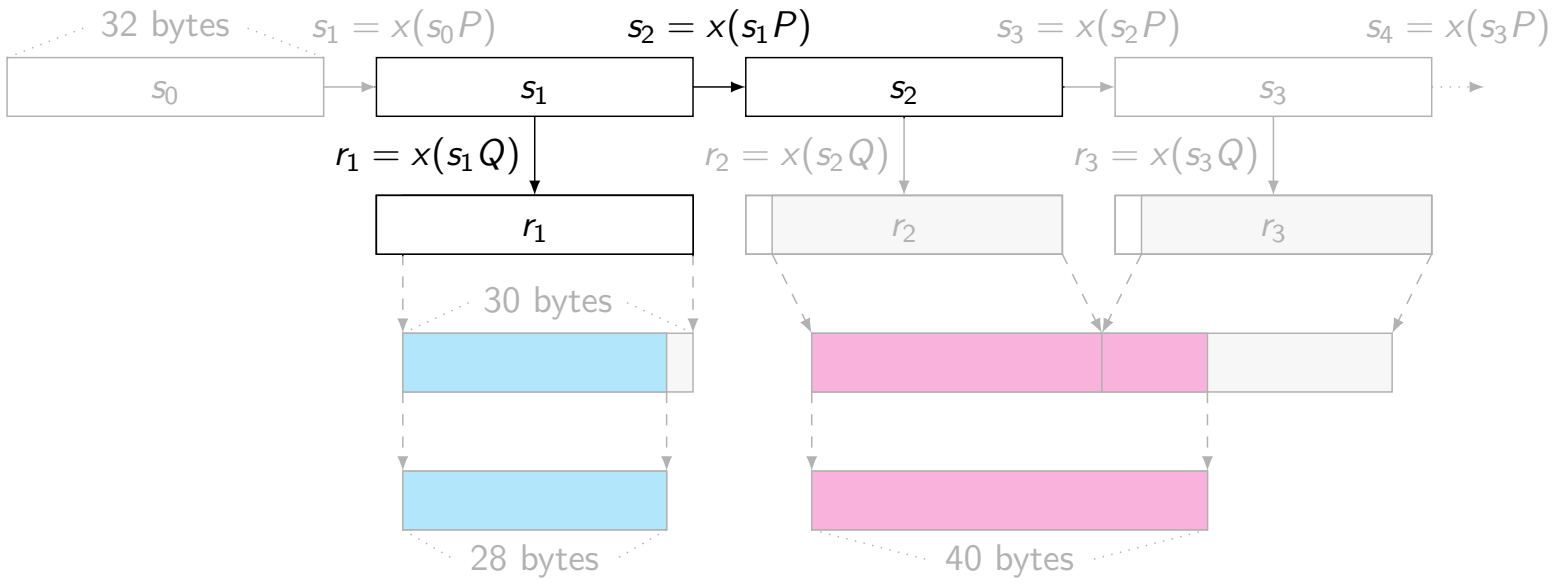
Points Q and P on an elliptic curve.



Graphic thanks to Ruben Niederhagen.

Basic attack

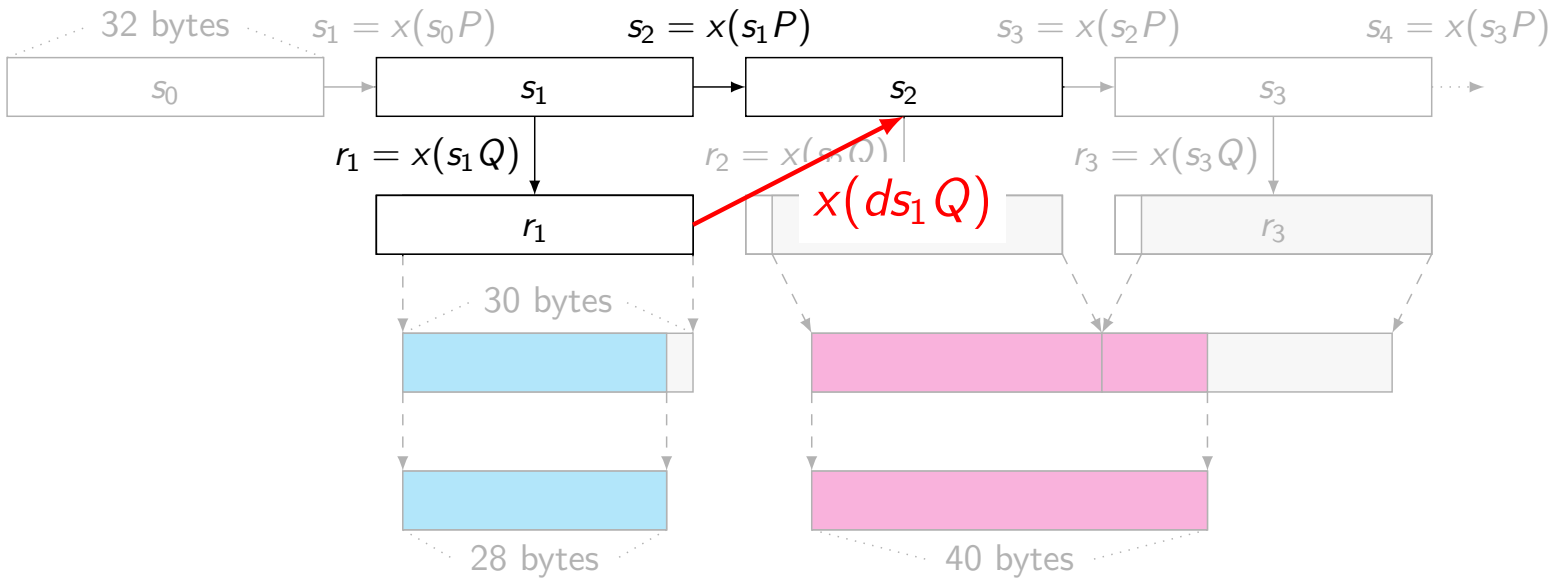
Points Q and $P = dQ$ on an elliptic curve.



Graphic thanks to Ruben Niederhagen.

Basic attack

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

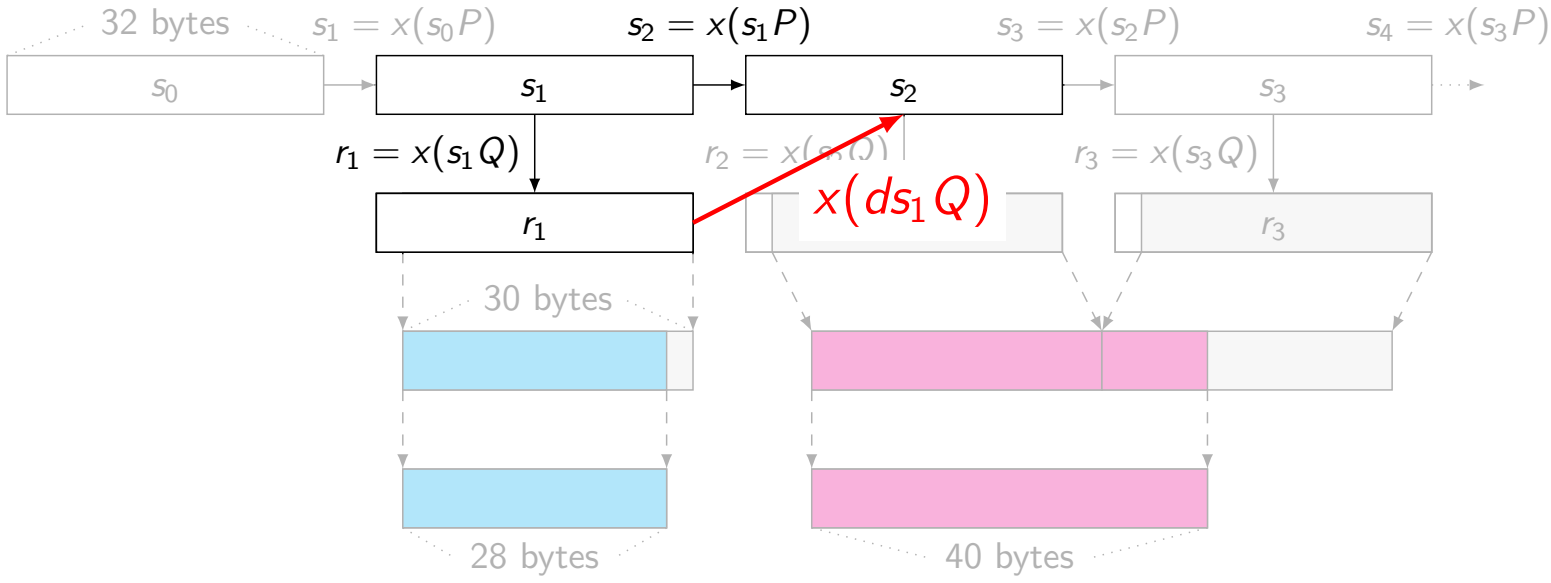


Graphic thanks to Ruben Niederhagen.

Basic attack

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

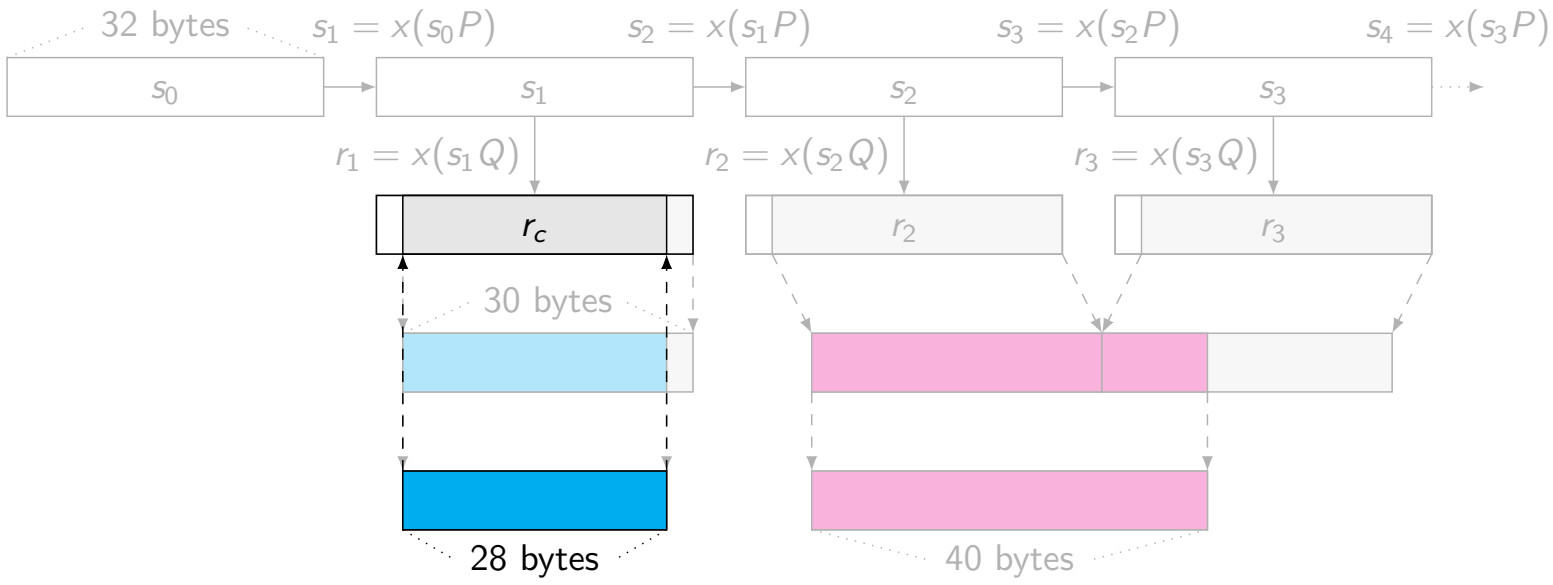
$$s_2 = x(s_1 P) = x(s_1 dQ)$$



Graphic thanks to Ruben Niederhagen.

Basic attack

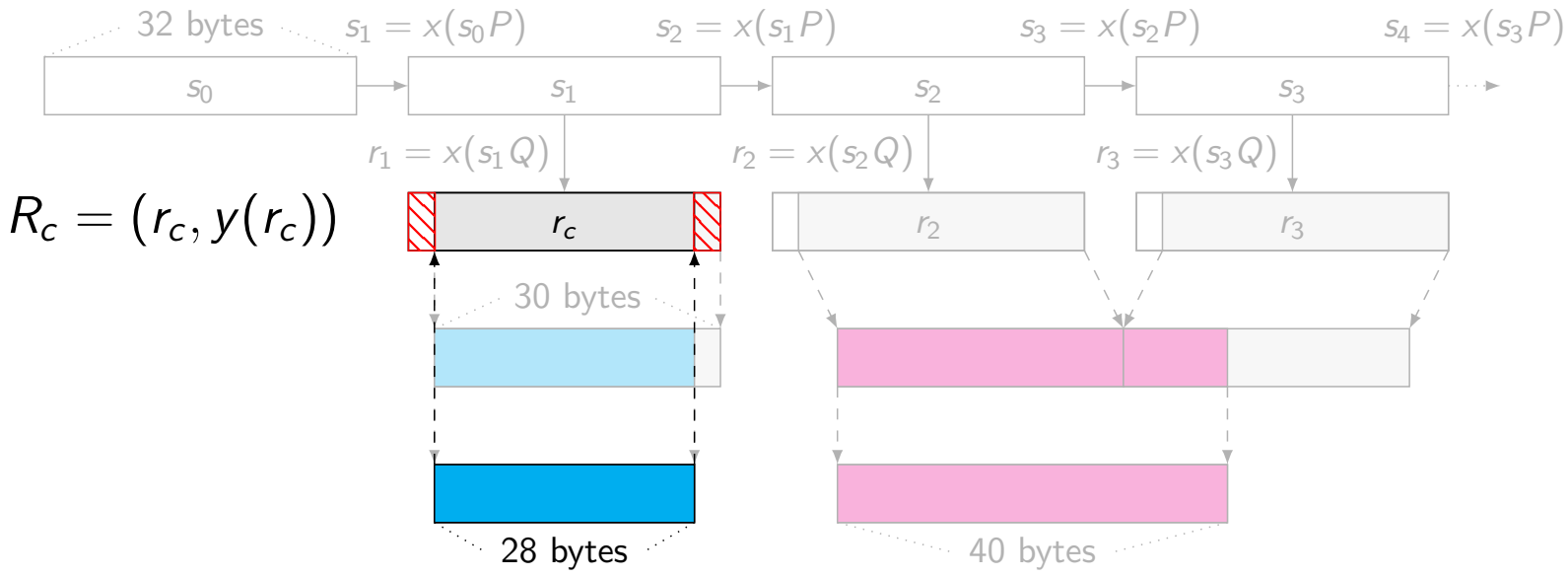
Points Q and $P = dQ$ on an elliptic curve.



Graphic thanks to Ruben Niederhagen.

Basic attack

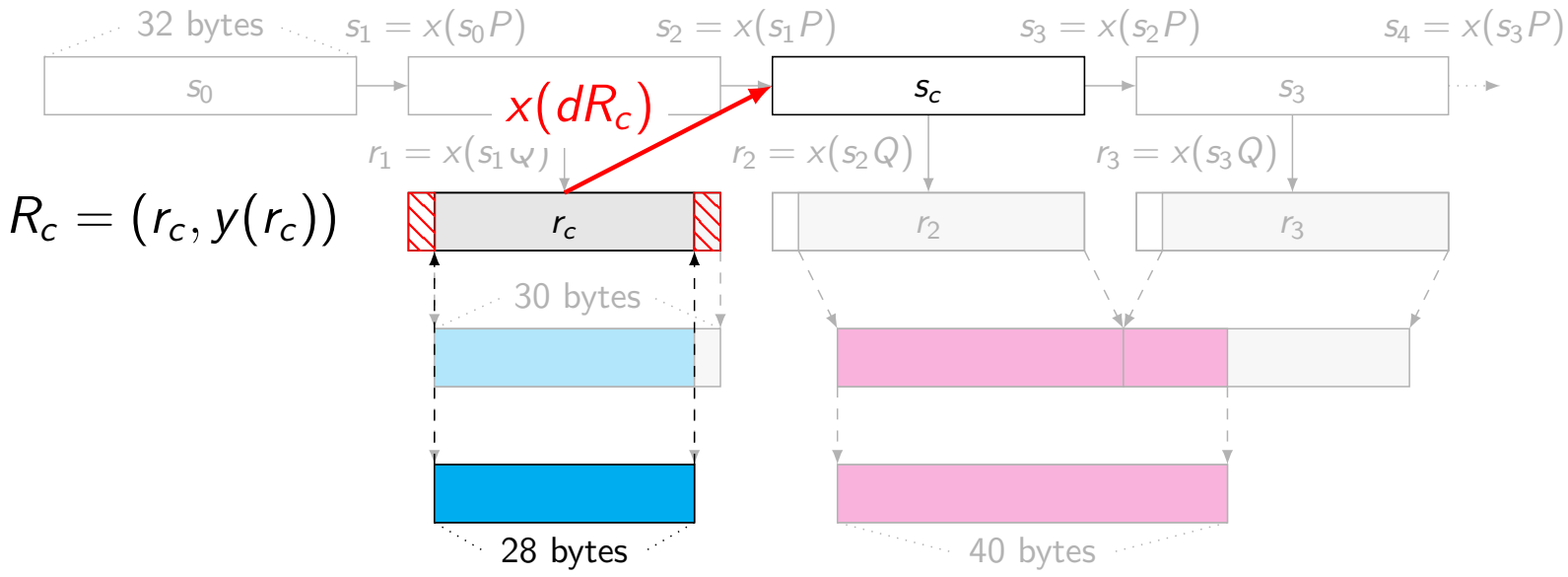
Points Q and $P = dQ$ on an elliptic curve.



Graphic thanks to Ruben Niederhagen.

Basic attack

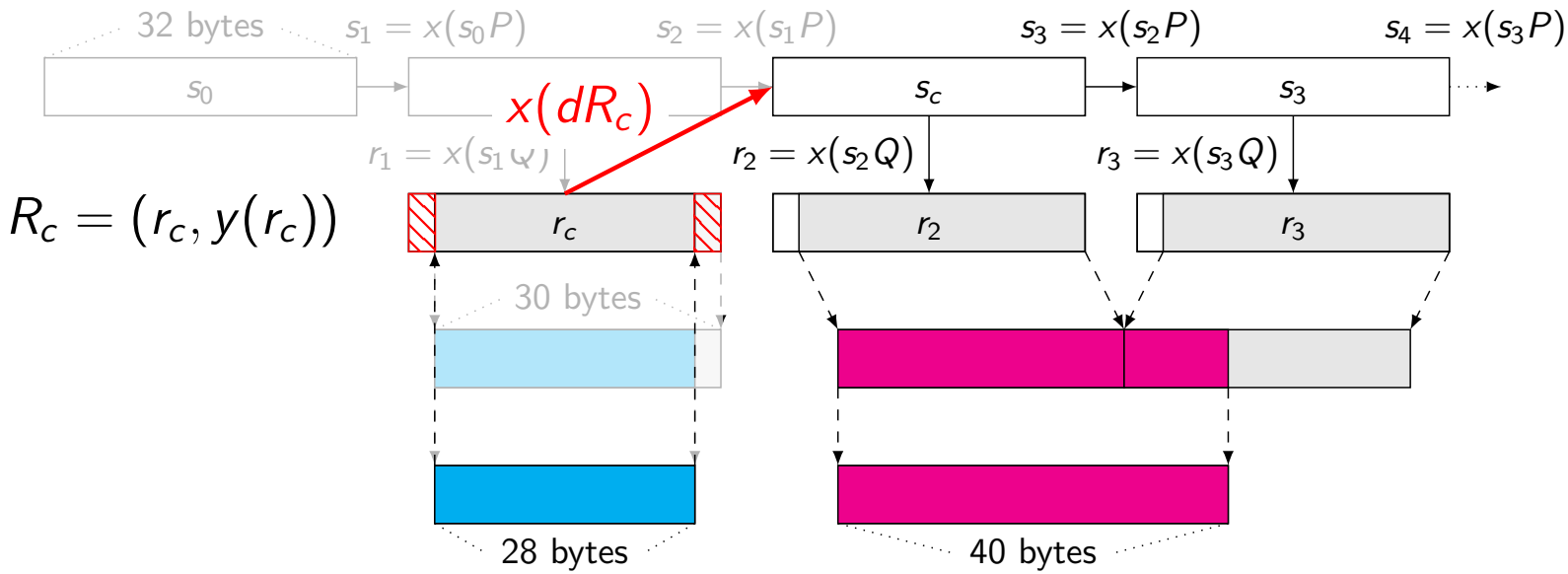
Points Q and $P = dQ$ on an elliptic curve.



Graphic thanks to Ruben Niederhagen.

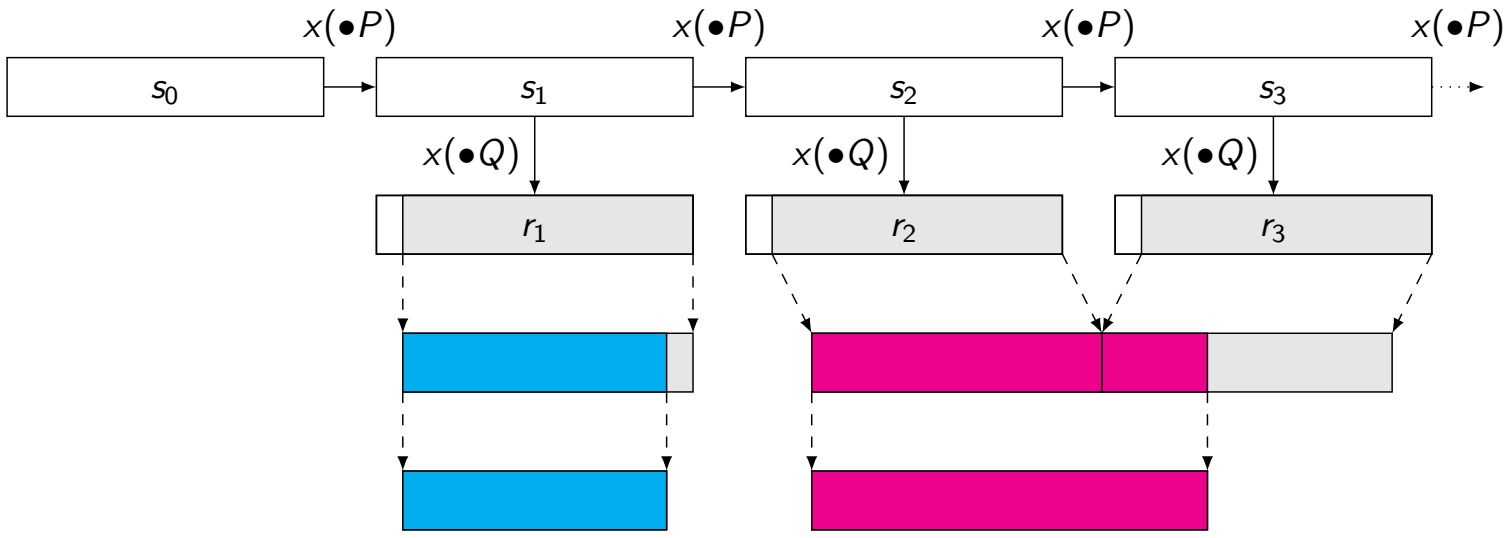
Basic attack

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



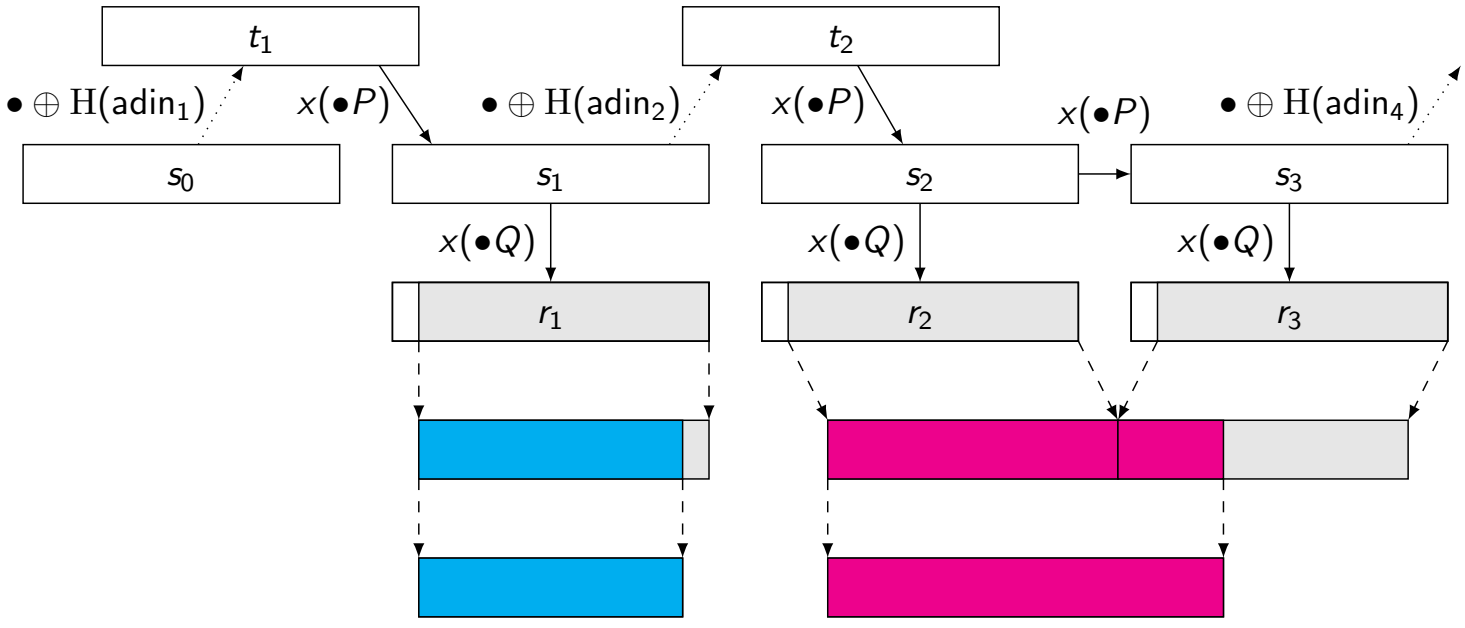
Graphic thanks to Ruben Niederhagen.

Dual EC in TLS



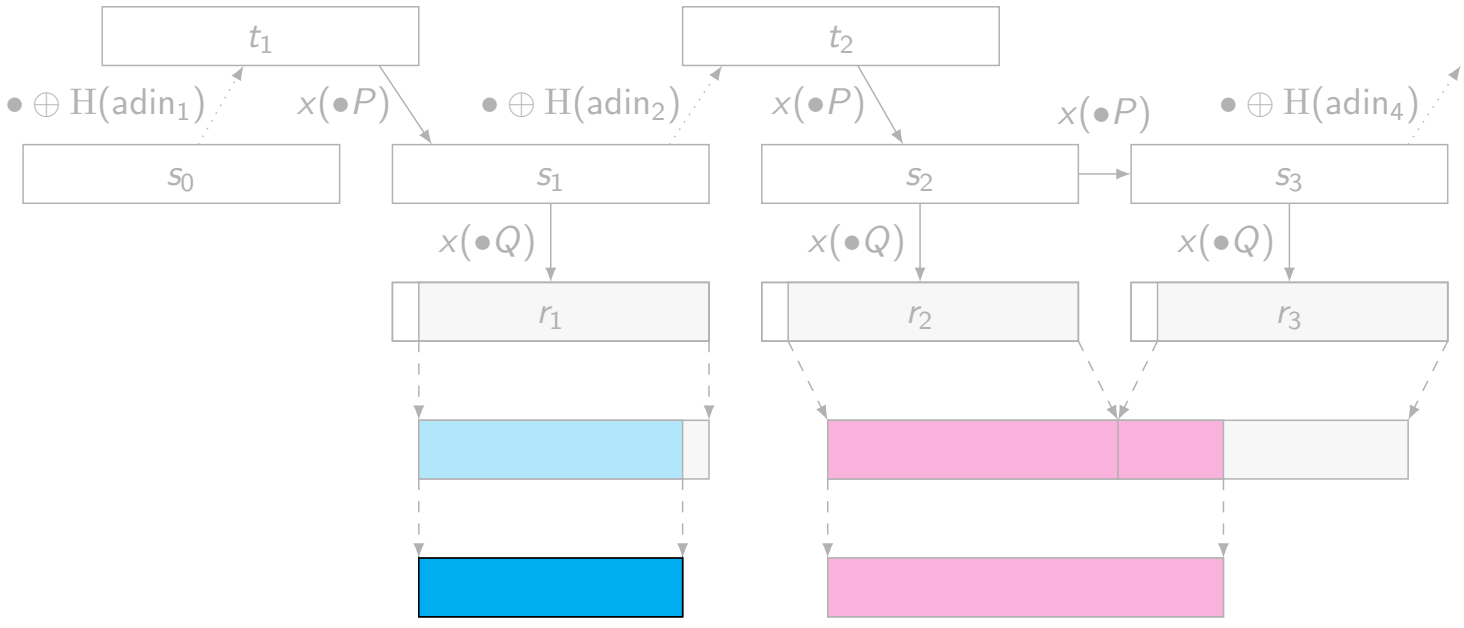
Graphic thanks to Ruben Niederhagen.

NIST SP800-90 in June 2006



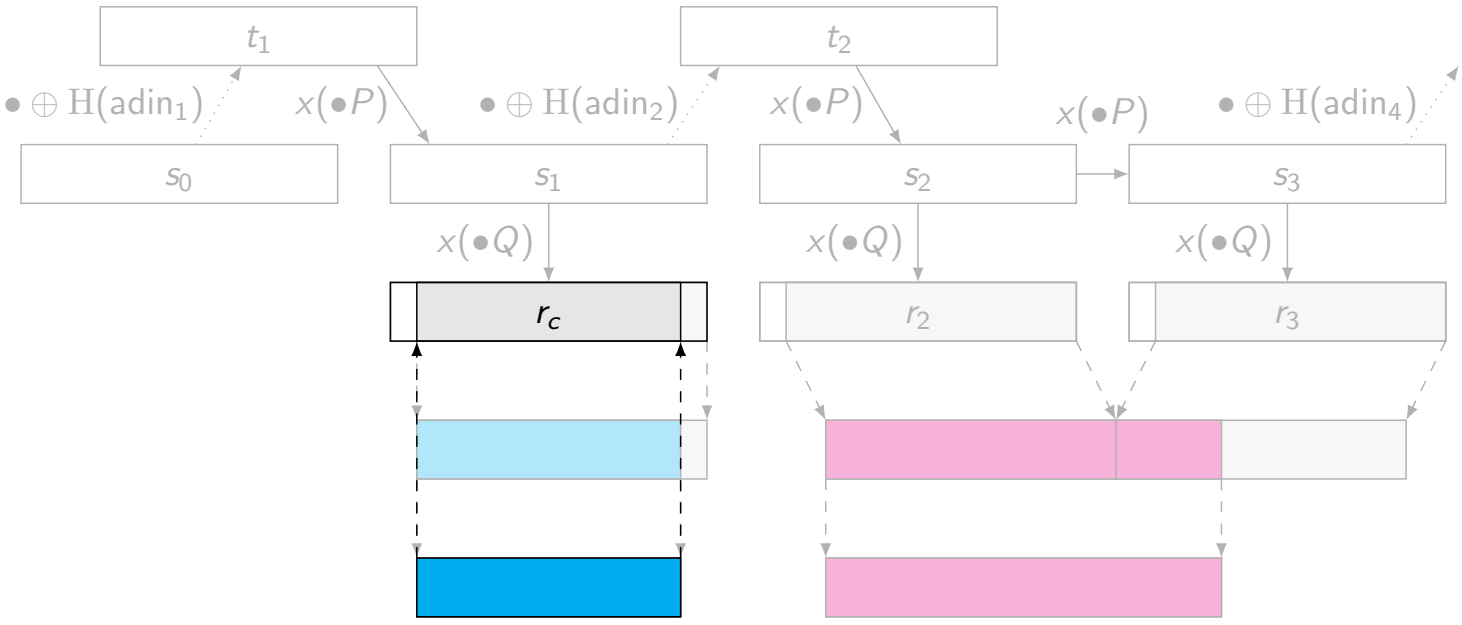
Graphic thanks to Ruben Niederhagen.

NIST SP800-90 in June 2006



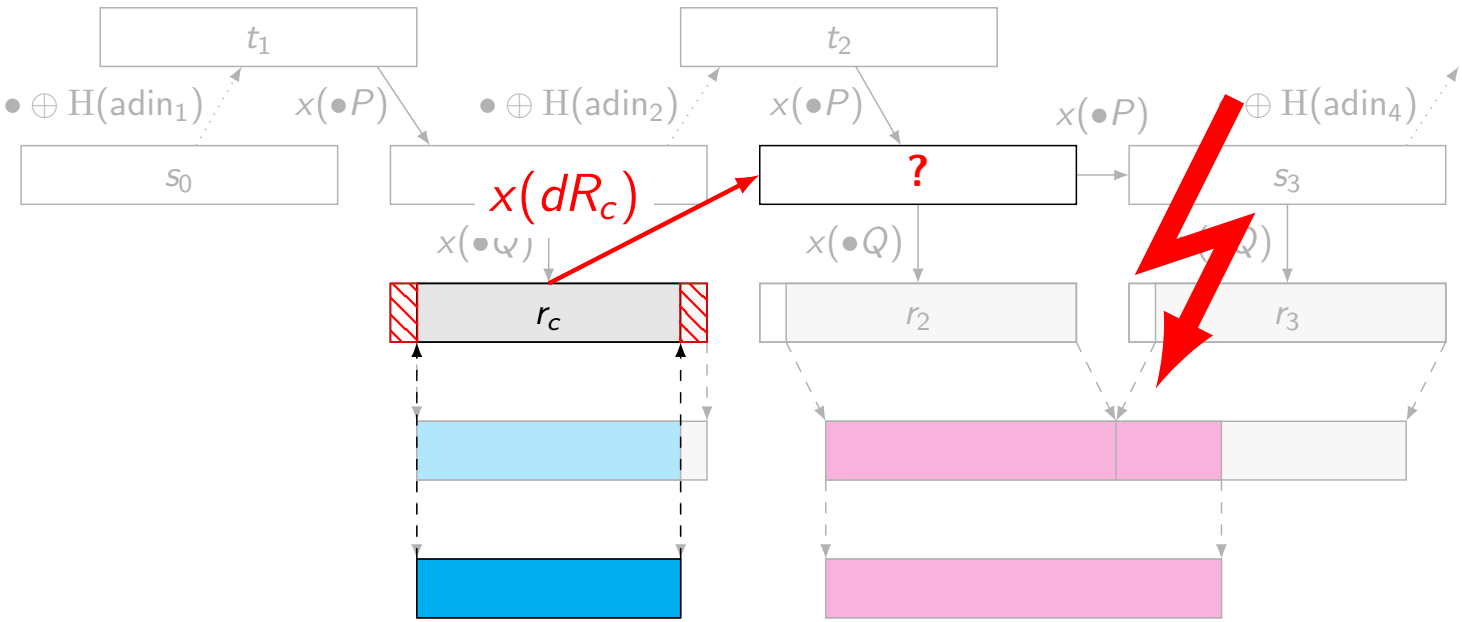
Graphic thanks to Ruben Niederhagen.

NIST SP800-90 in June 2006



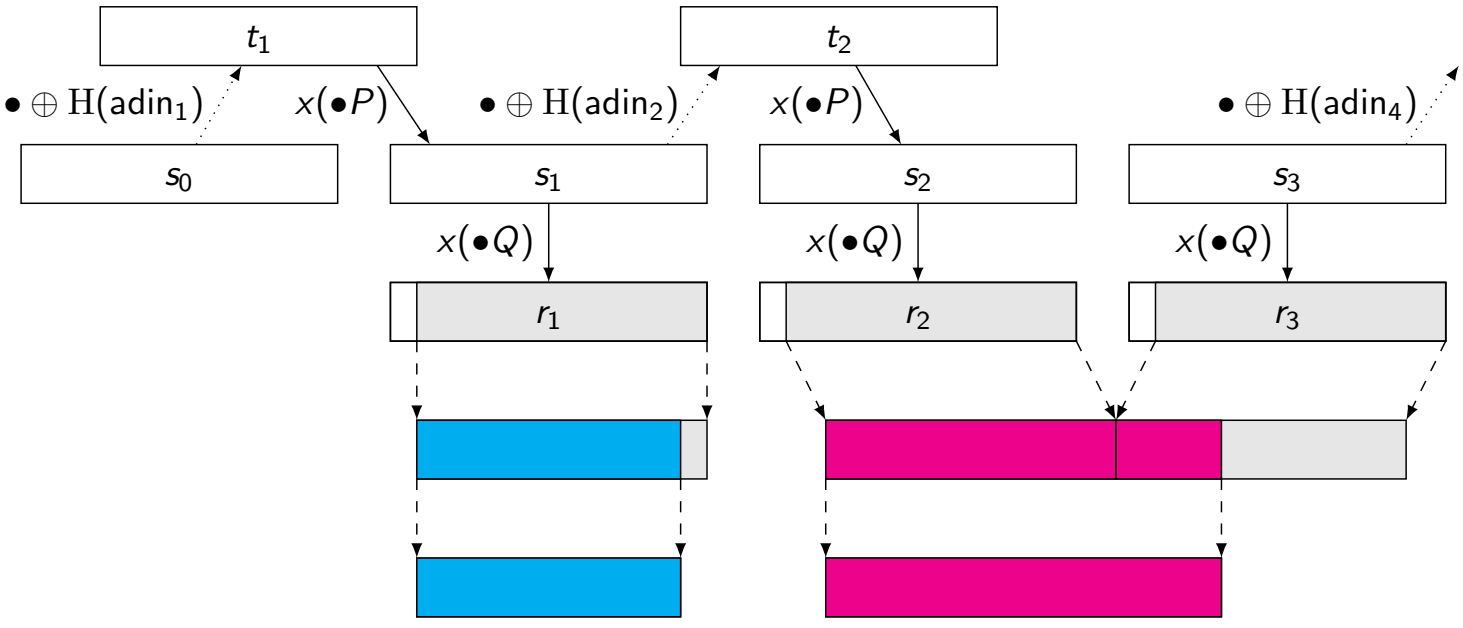
Graphic thanks to Ruben Niederhagen.

NIST SP800-90 in June 2006



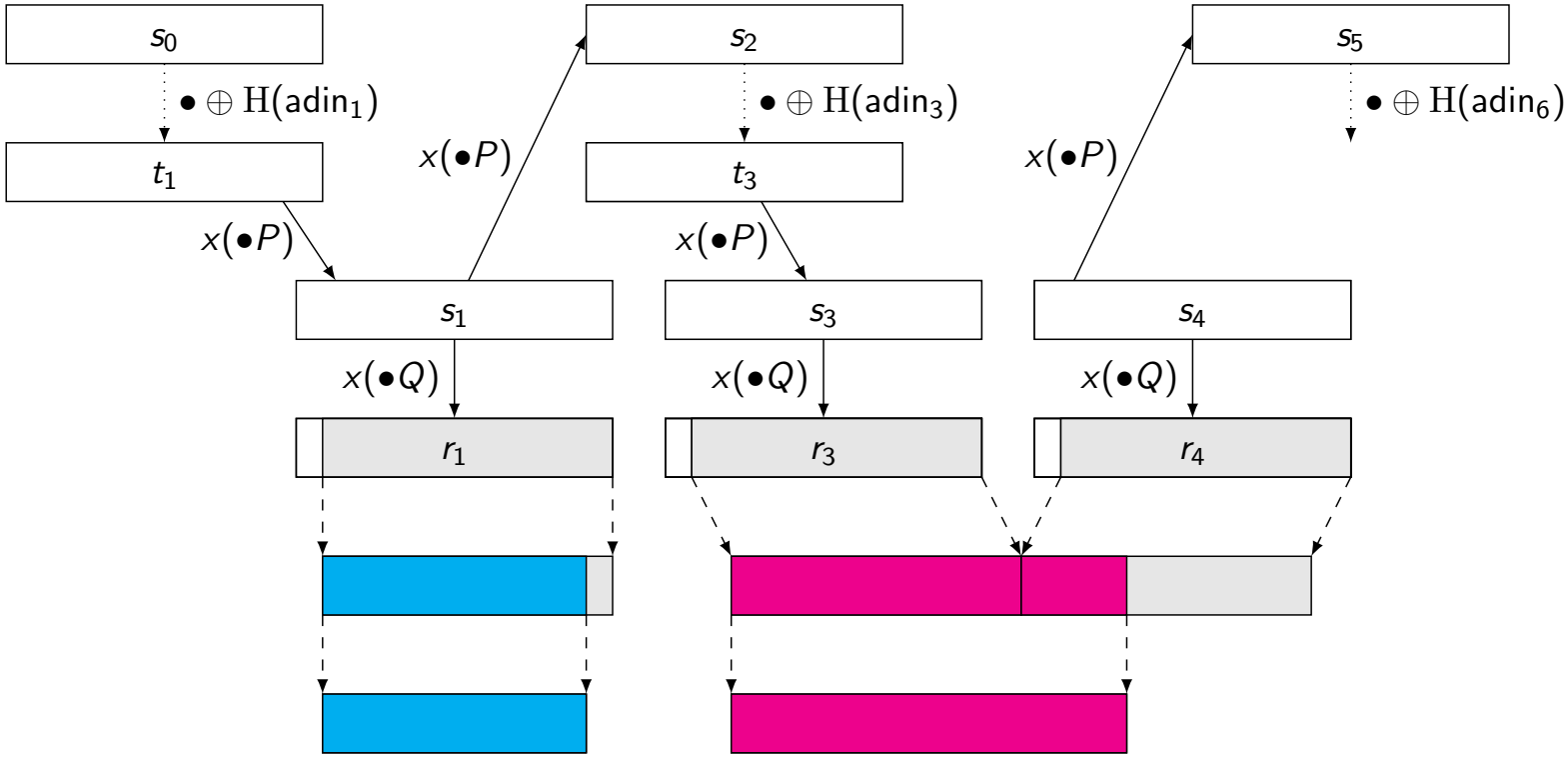
Graphic thanks to Ruben Niederhagen.

Dual EC in TLS



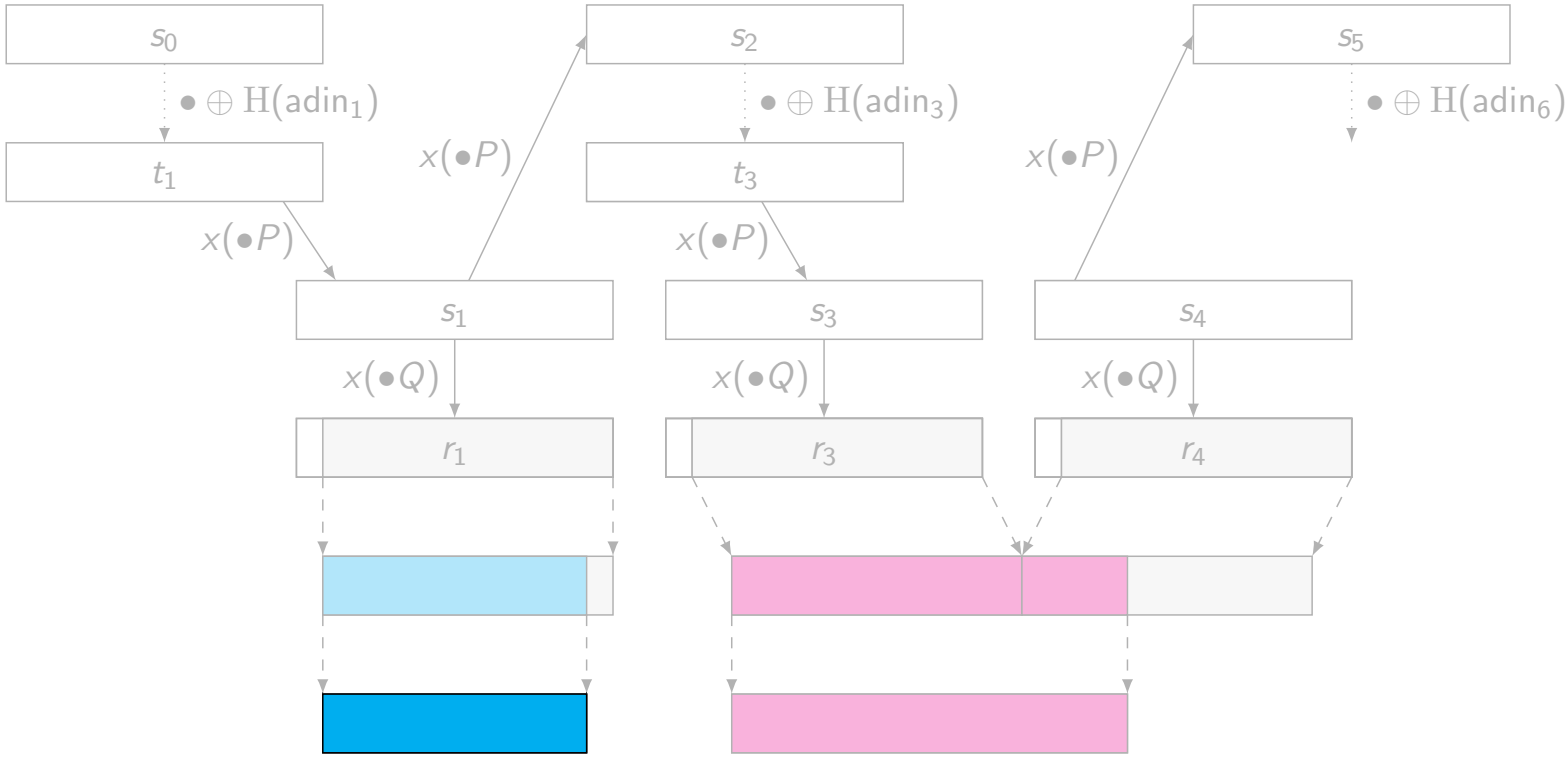
Graphic thanks to Ruben Niederhagen.

NIST SP800-90 in March 2007



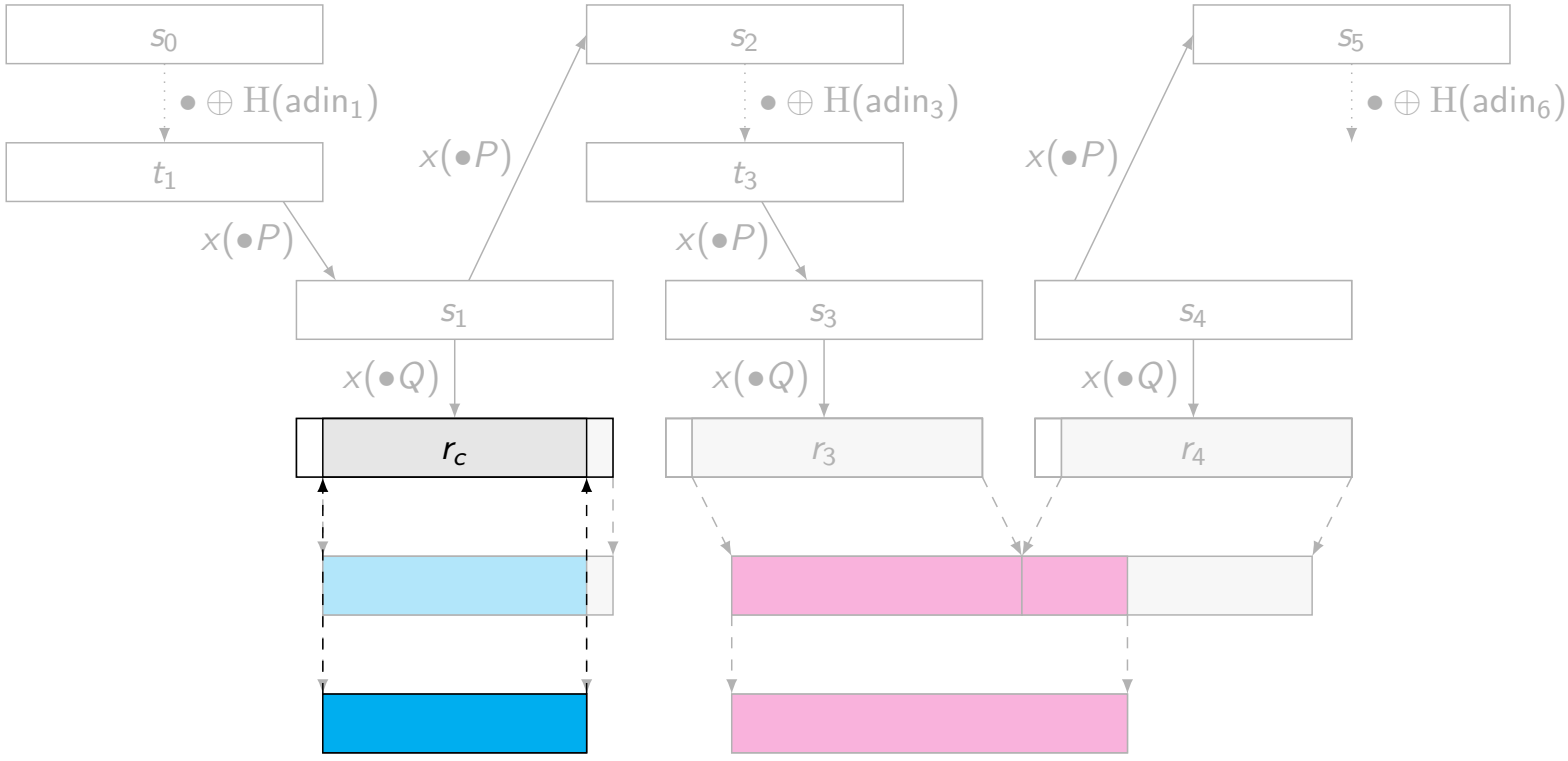
Graphic thanks to Ruben Niederhagen.

NIST SP800-90 in March 2007



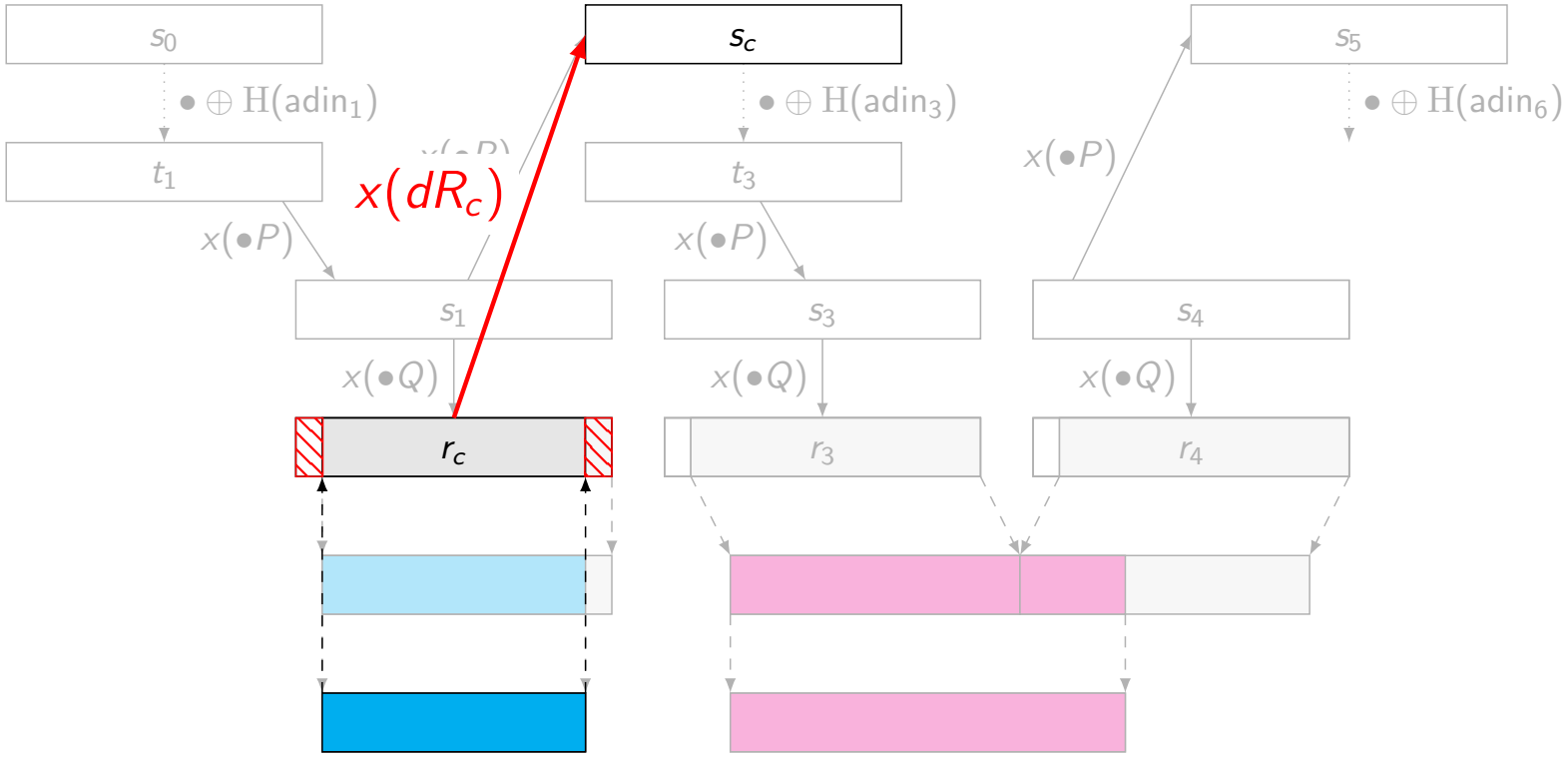
Graphic thanks to Ruben Niederhagen.

NIST SP800-90 in March 2007



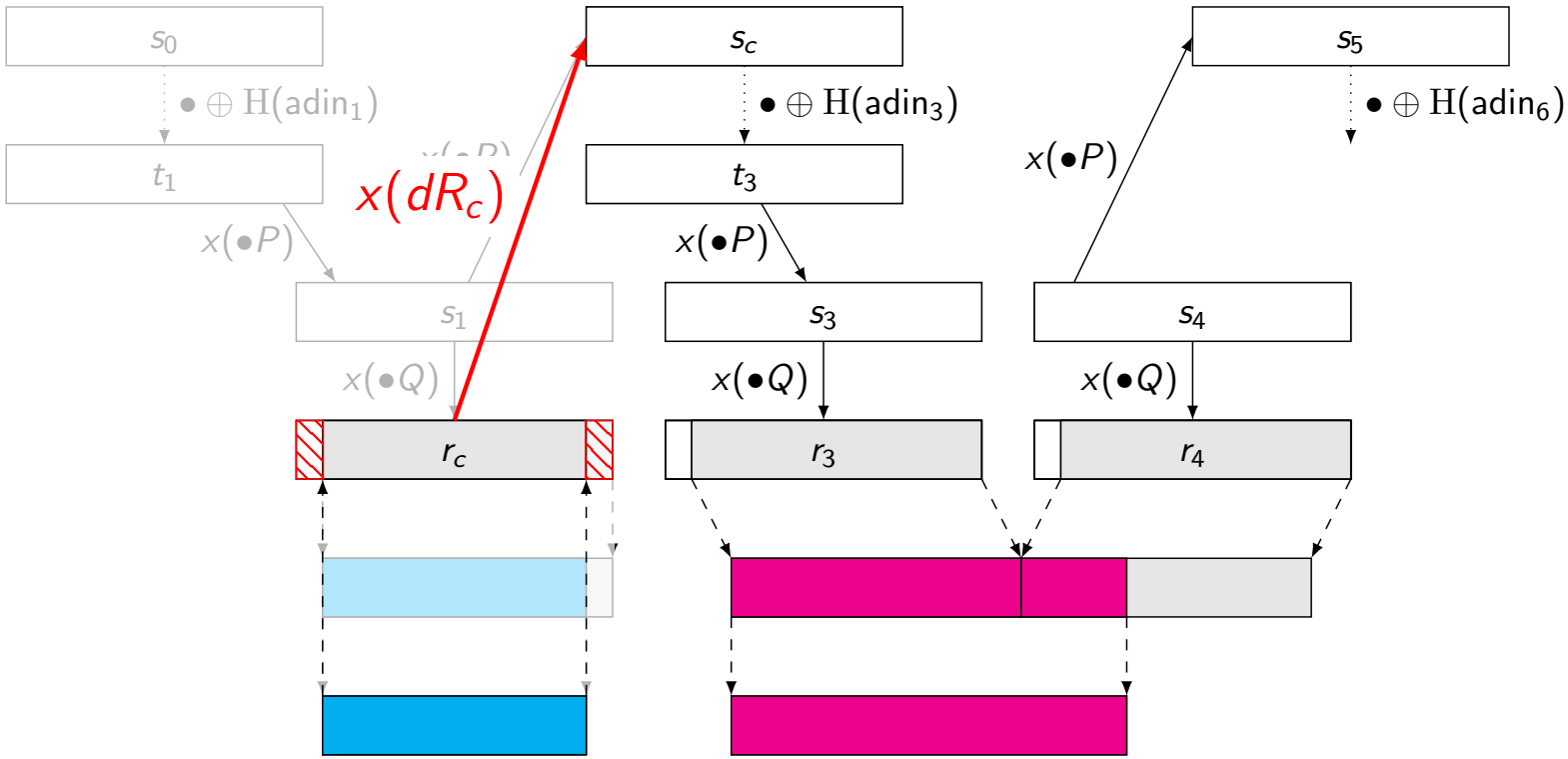
Graphic thanks to Ruben Niederhagen.

NIST SP800-90 in March 2007



Graphic thanks to Ruben Niederhagen.

NIST SP800-90 in March 2007



Graphic thanks to Ruben Niederhagen.

Changelog to NIST SP800-90

Appendix I : (Informative) Revisions

This original version of this Recommendation was completed in June, 2006. In March 2007, the following changes were made (note that the changes are indicated in italics):

...

4. In Section 10.3.1.4, a step was inserted that will provide backtracking resistance (step 14 of the pseudocode). The same change was made to the example in Appendix F.5.3 (step 19.1).

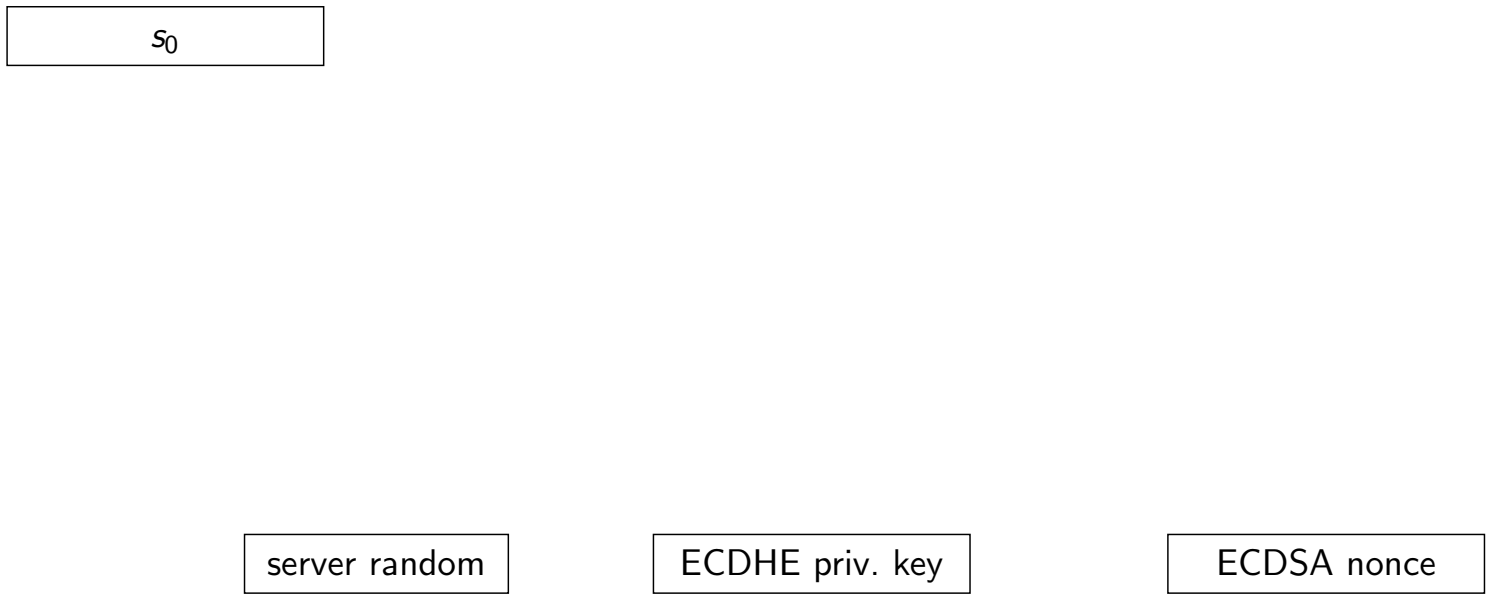
Attack – Example: BSAFE-Java

server random

ECDHE priv. key

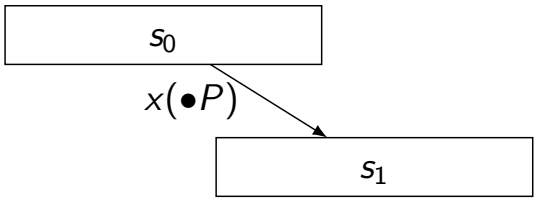
ECDSA nonce

Attack – Example: BSAFE-Java



Graphics: Ruben Niederhagen.

Attack – Example: BSAFE-Java



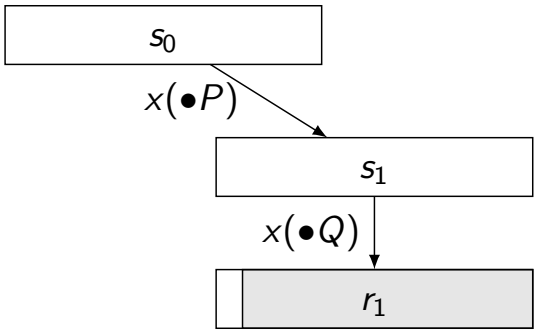
server random

ECDHE priv. key

ECDSA nonce

Graphics: Ruben Niederhagen.

Attack – Example: BSAFE-Java



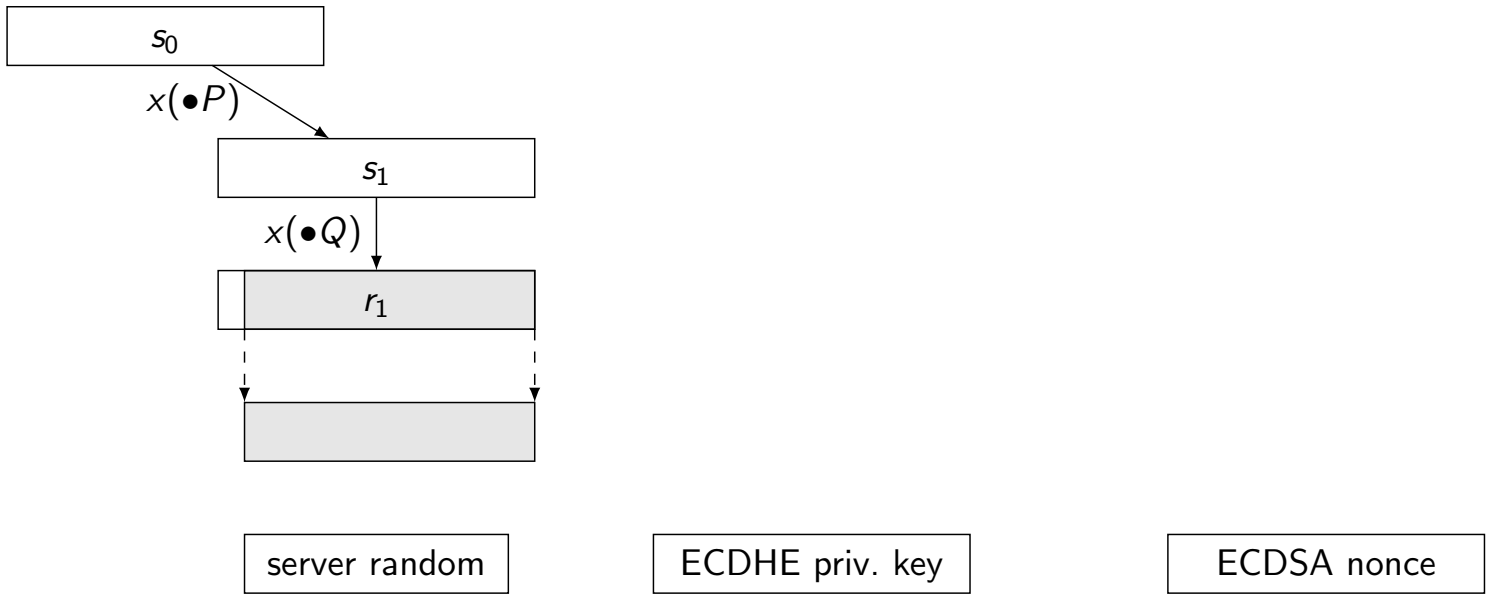
server random

ECDHE priv. key

ECDSA nonce

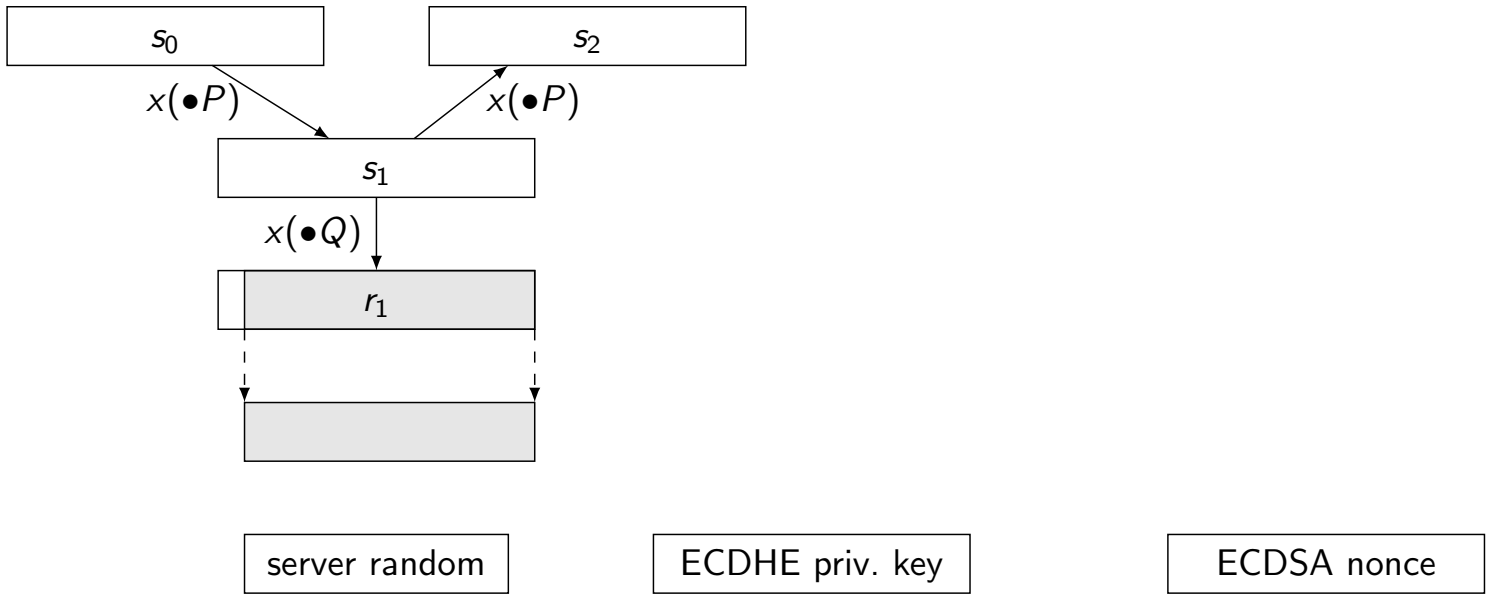
Graphics: Ruben Niederhagen.

Attack – Example: BSAFE-Java



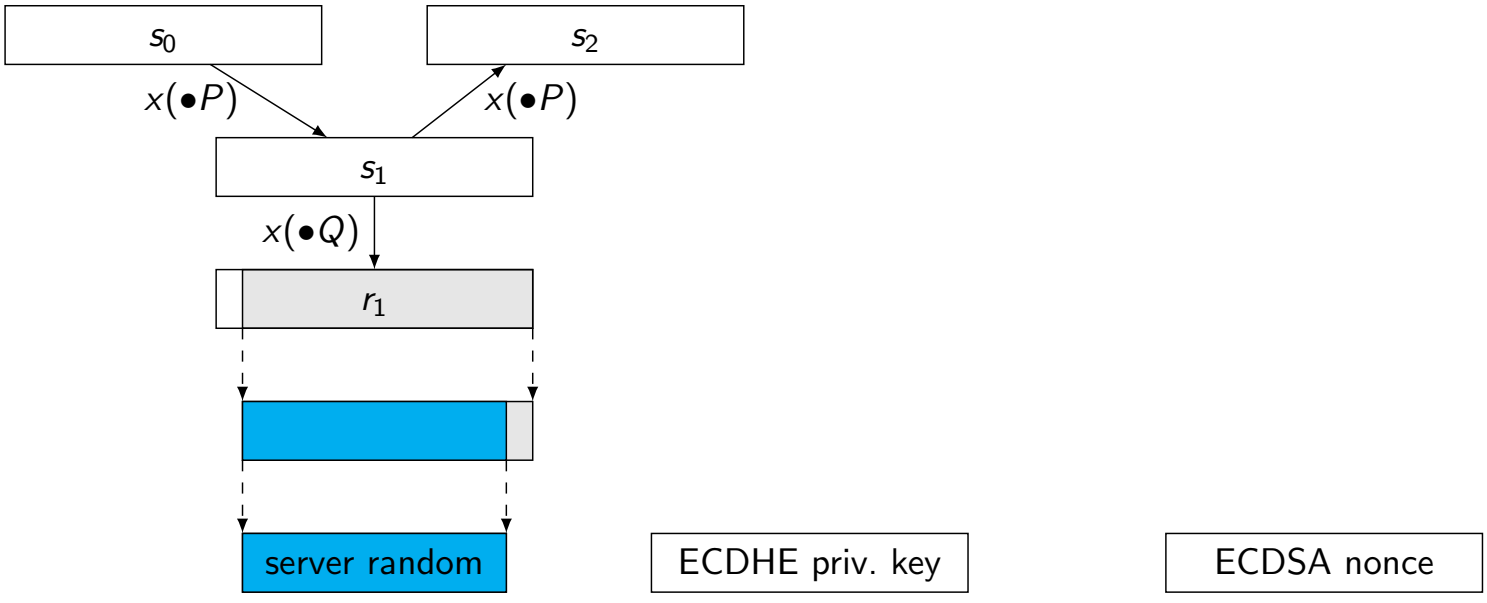
Graphics: Ruben Niederhagen.

Attack – Example: BSAFE-Java



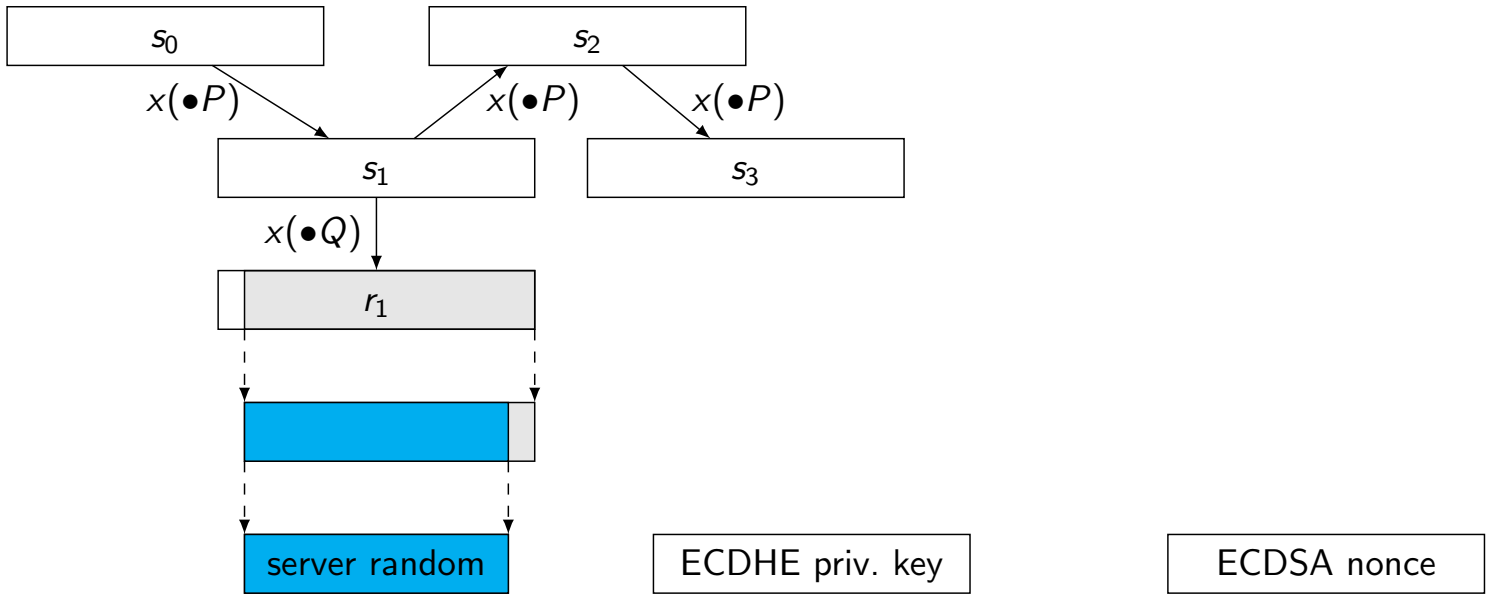
Graphics: Ruben Niederhagen.

Attack – Example: BSAFE-Java



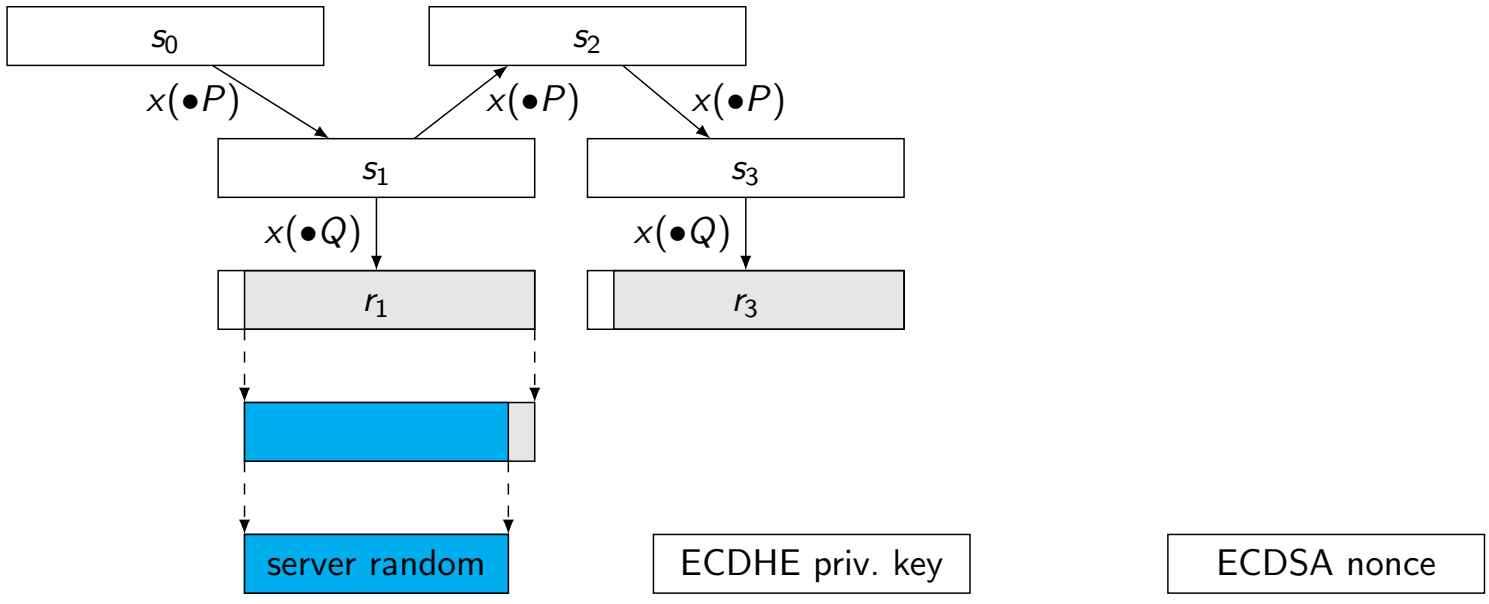
Graphics: Ruben Niederhagen.

Attack – Example: BSAFE-Java



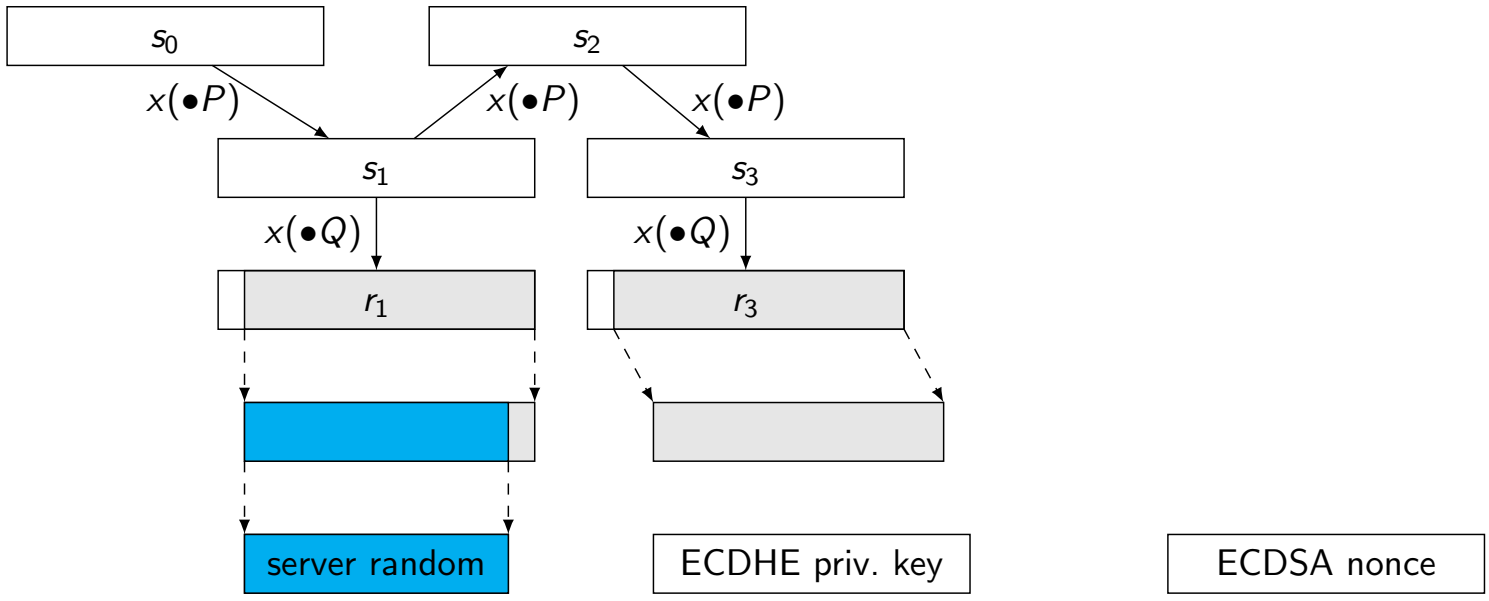
Graphics: Ruben Niederhagen.

Attack – Example: BSAFE-Java



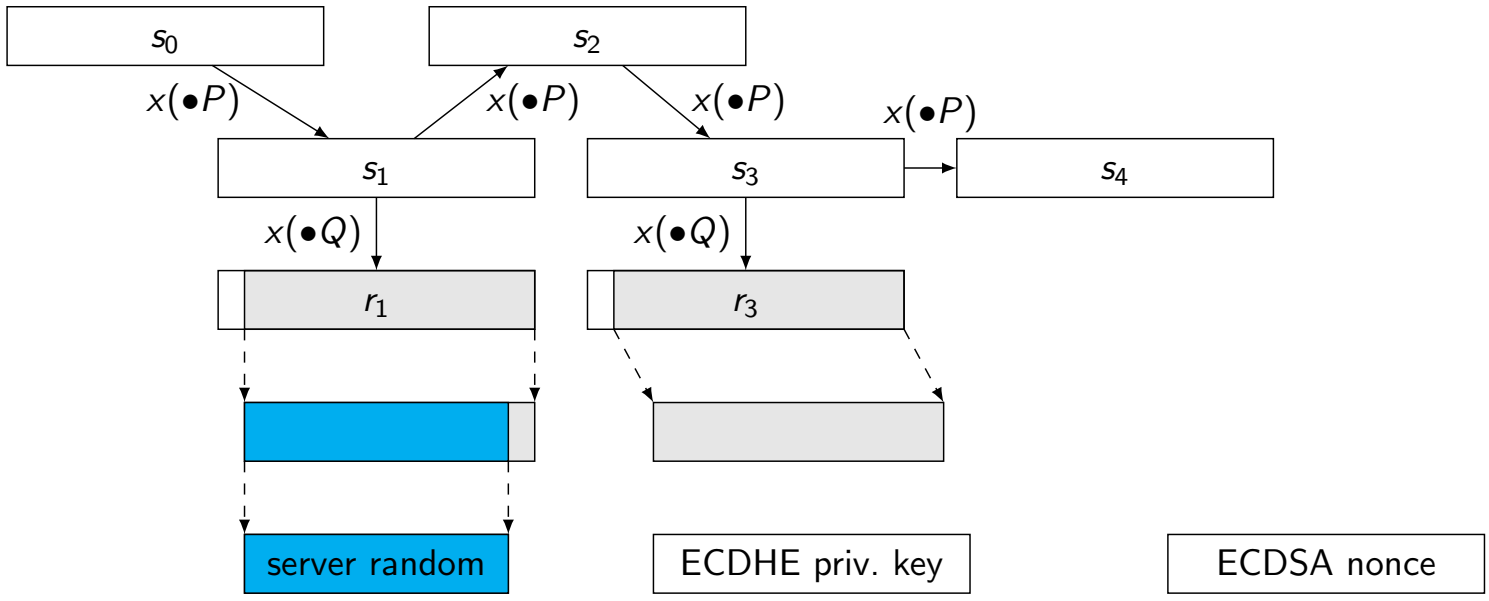
Graphics: Ruben Niederhagen.

Attack – Example: BSAFE-Java



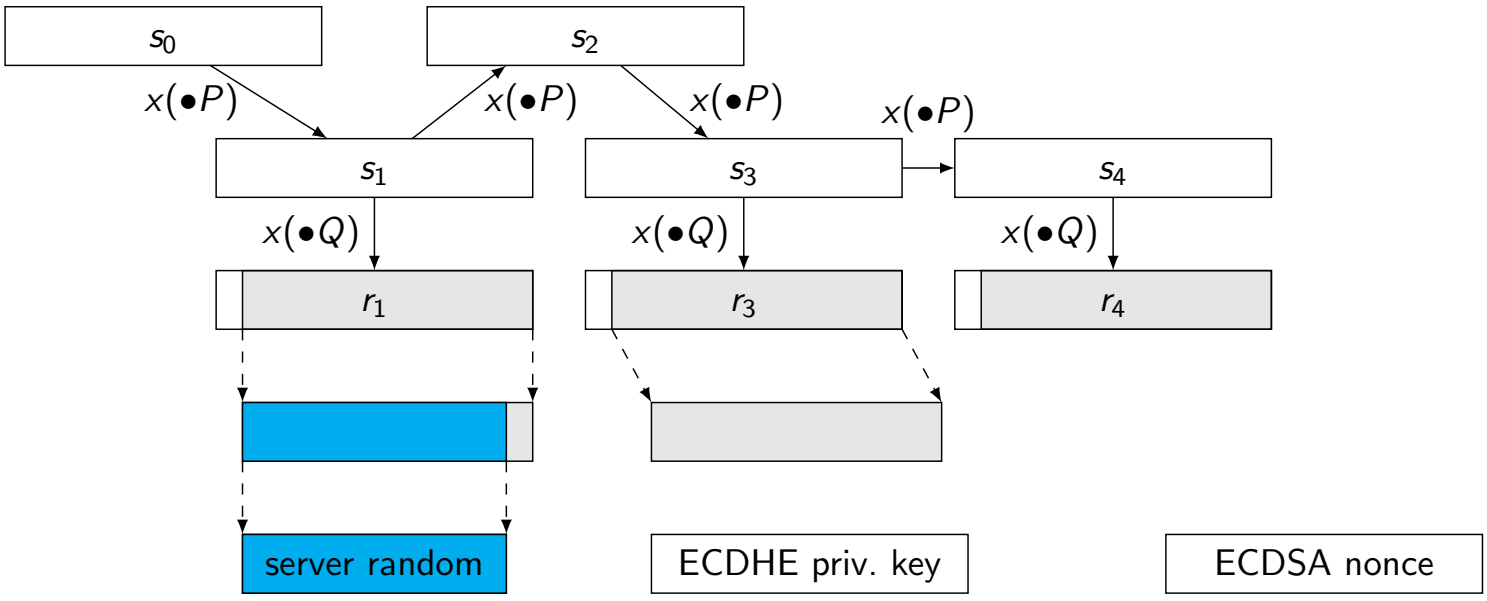
Graphics: Ruben Niederhagen.

Attack – Example: BSAFE-Java



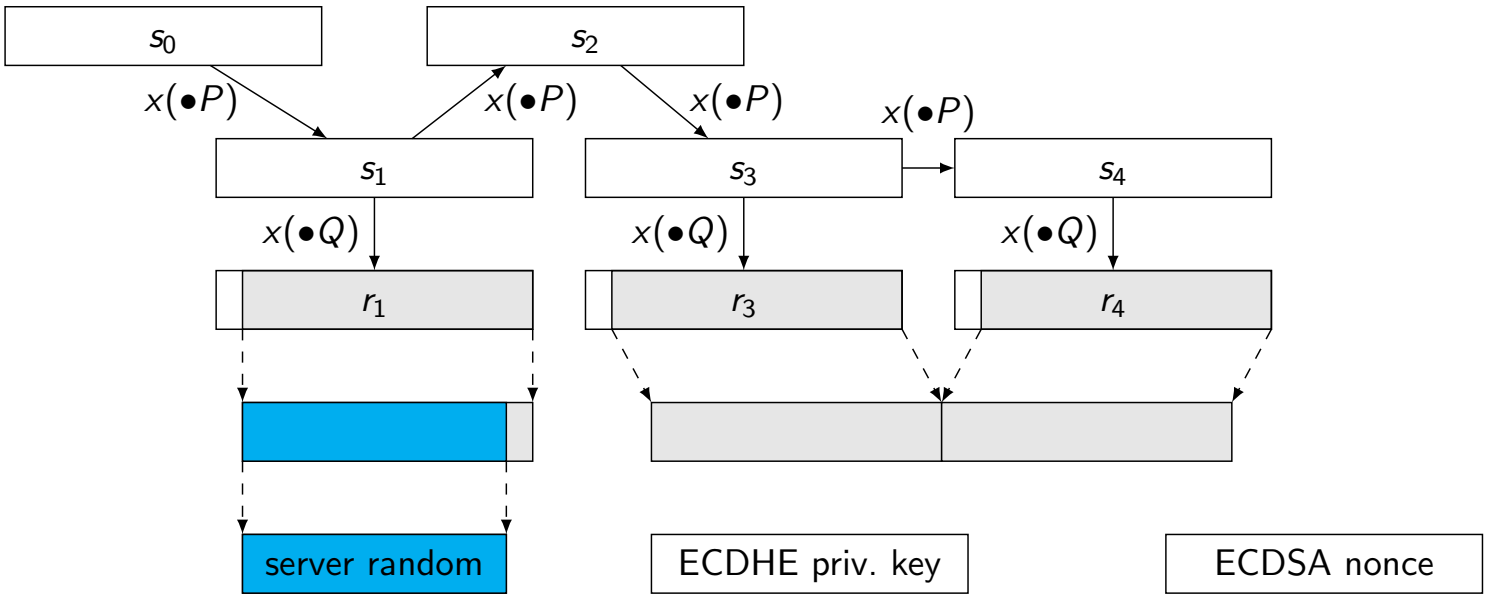
Graphics: Ruben Niederhagen.

Attack – Example: BSAFE-Java



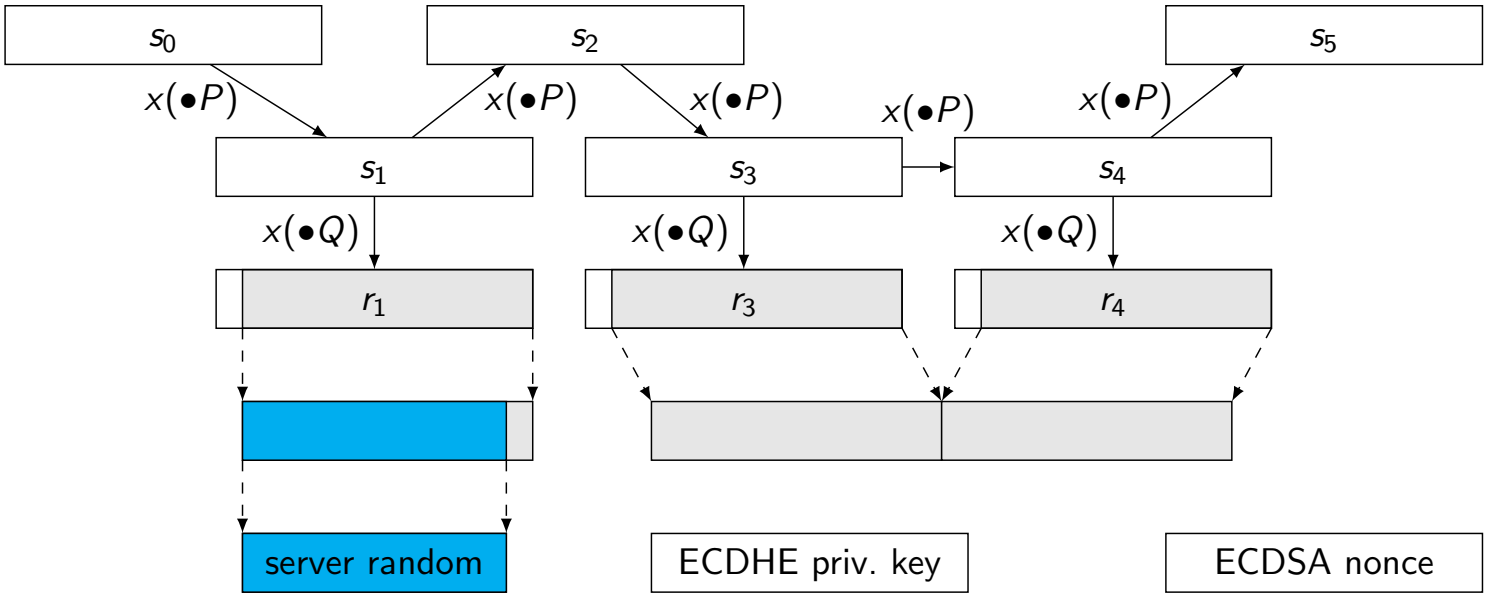
Graphics: Ruben Niederhagen.

Attack – Example: BSAFE-Java



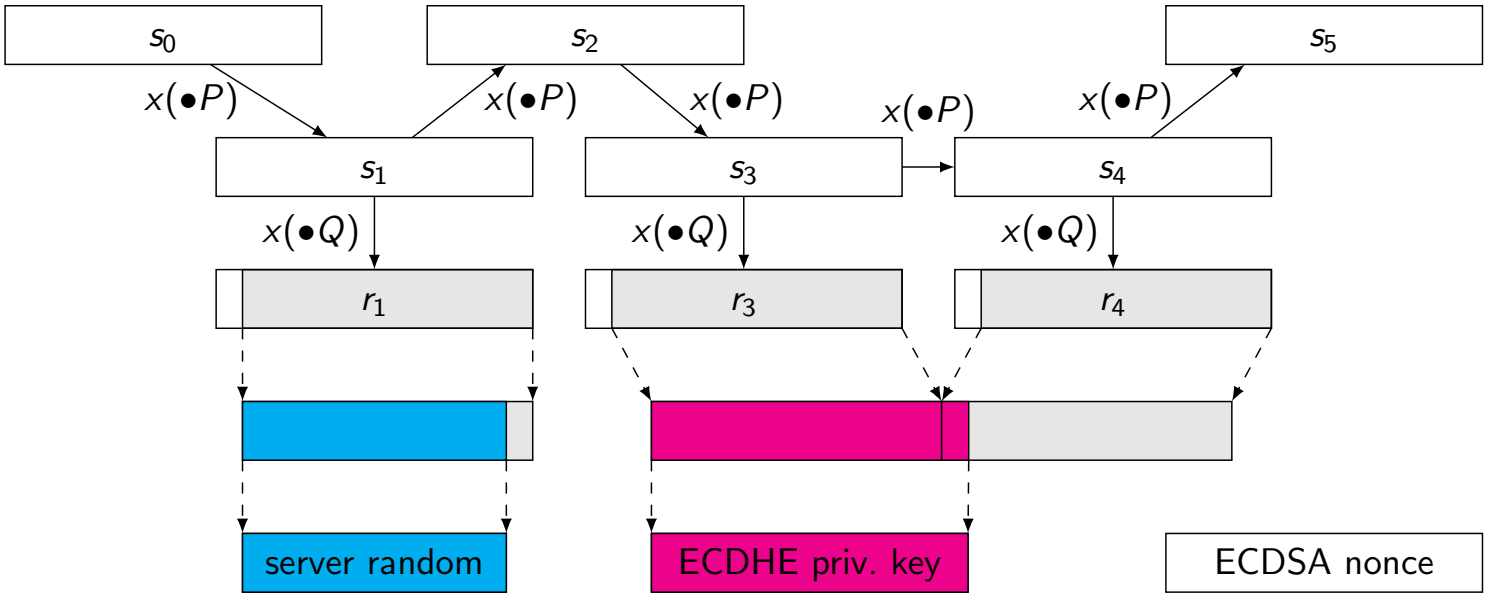
Graphics: Ruben Niederhagen.

Attack – Example: BSAFE-Java



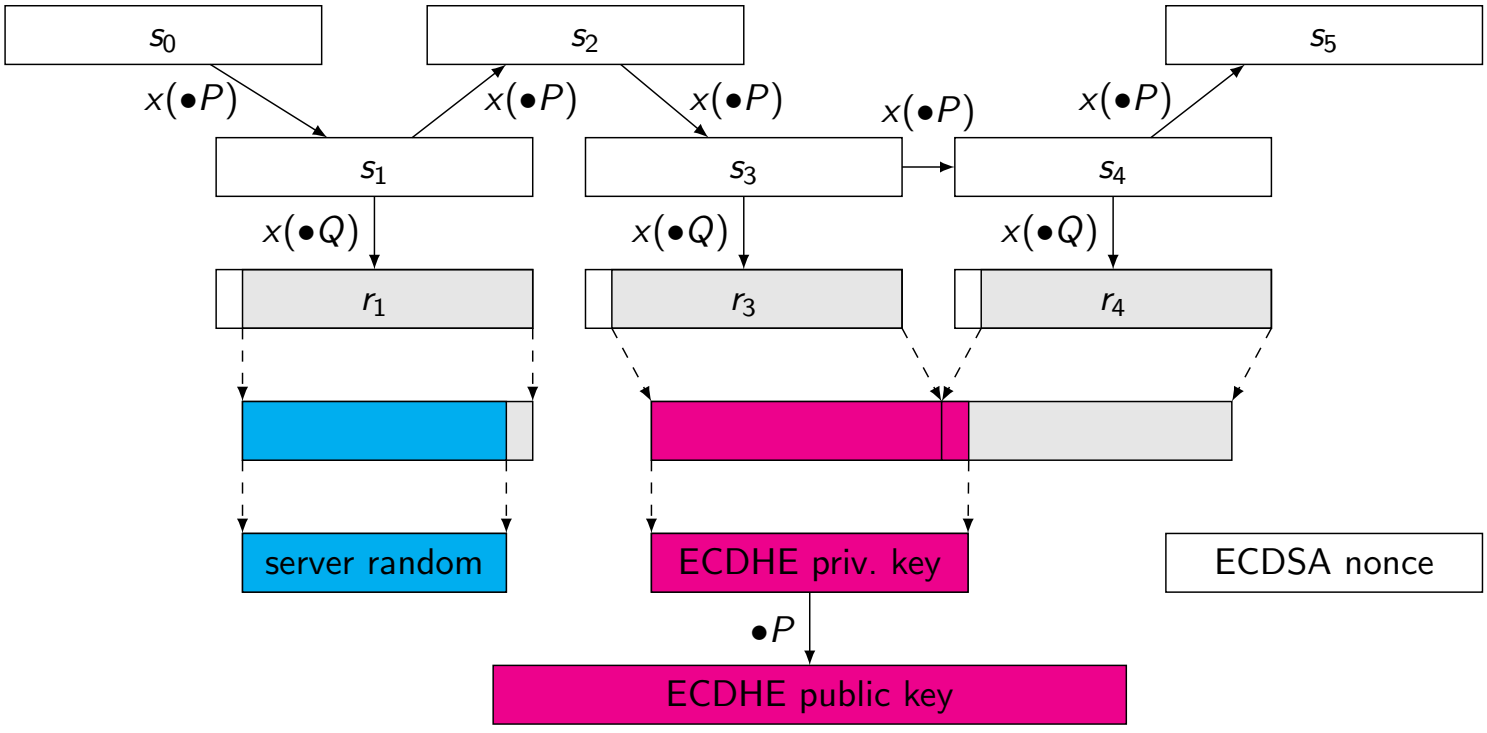
Graphics: Ruben Niederhagen.

Attack – Example: BSAFE-Java



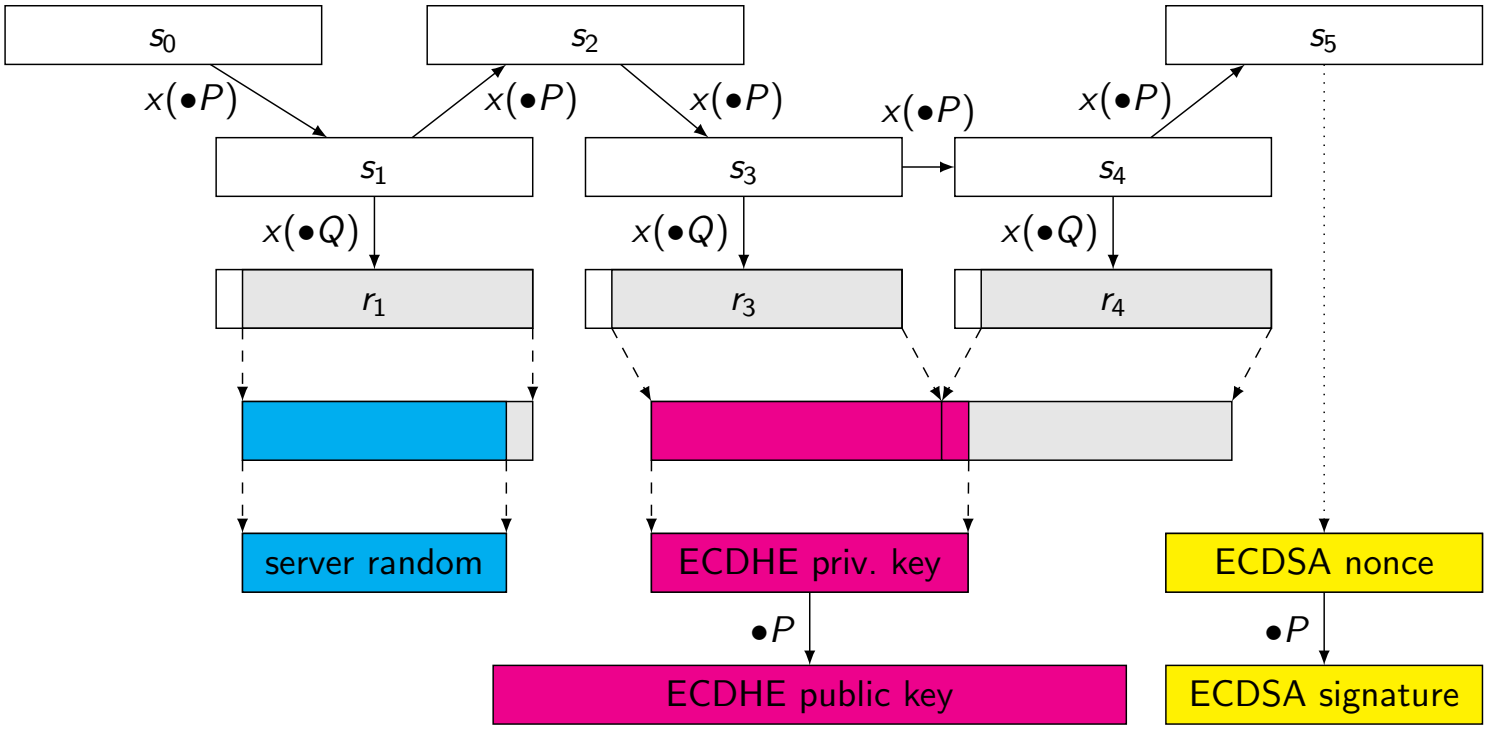
Graphics: Ruben Niederhagen.

Attack – Example: BSAFE-Java



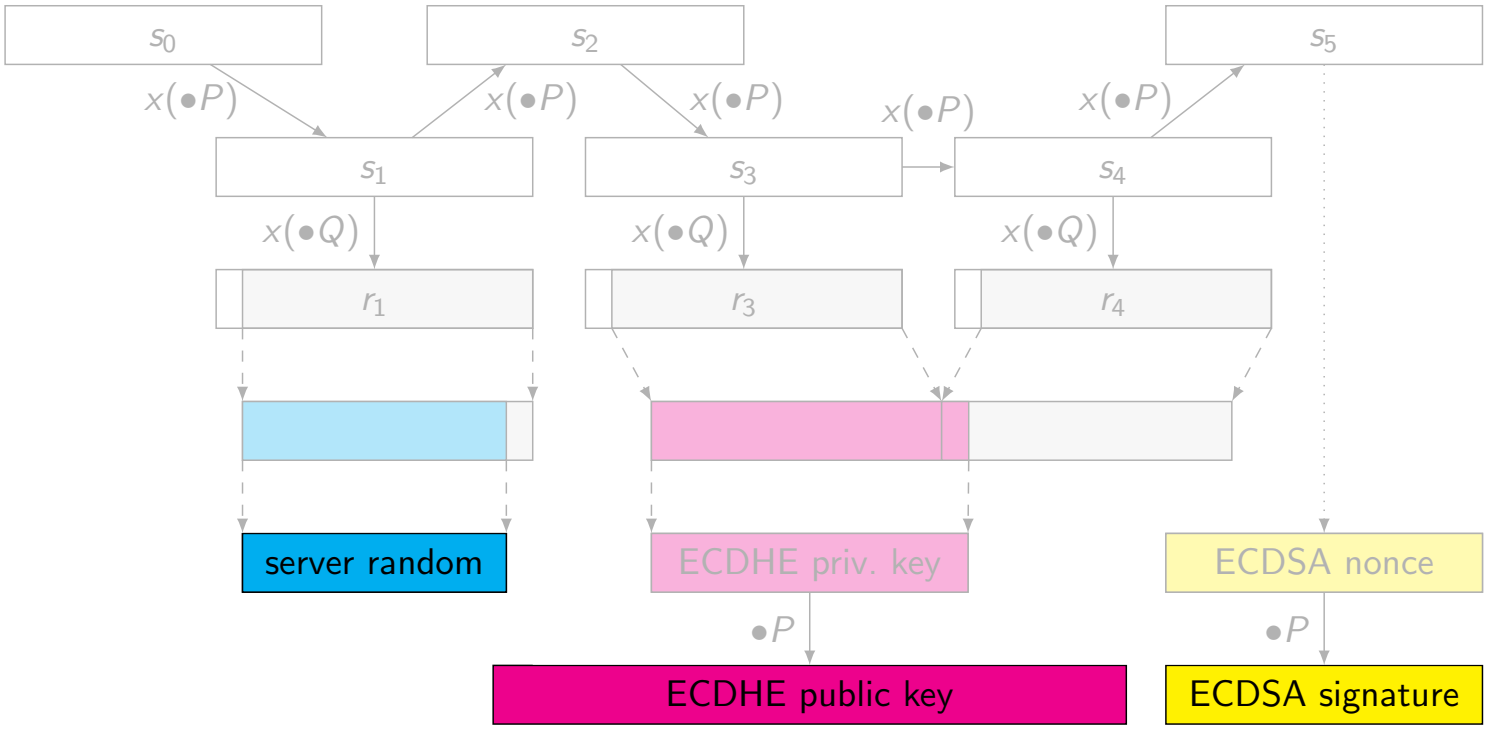
Graphics: Ruben Niederhagen.

Attack – Example: BSAFE-Java



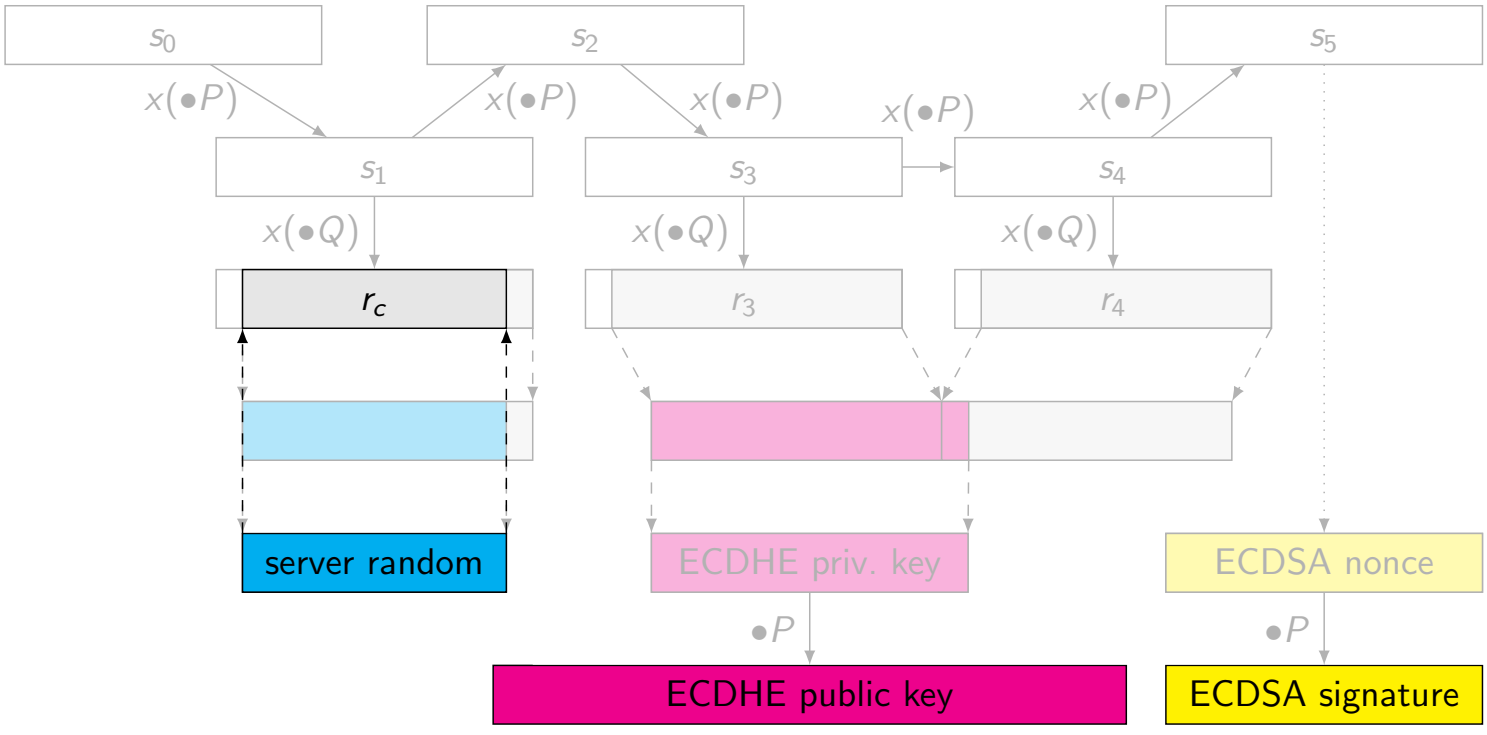
Graphics: Ruben Niederhagen.

Attack – Example: BSAFE-Java



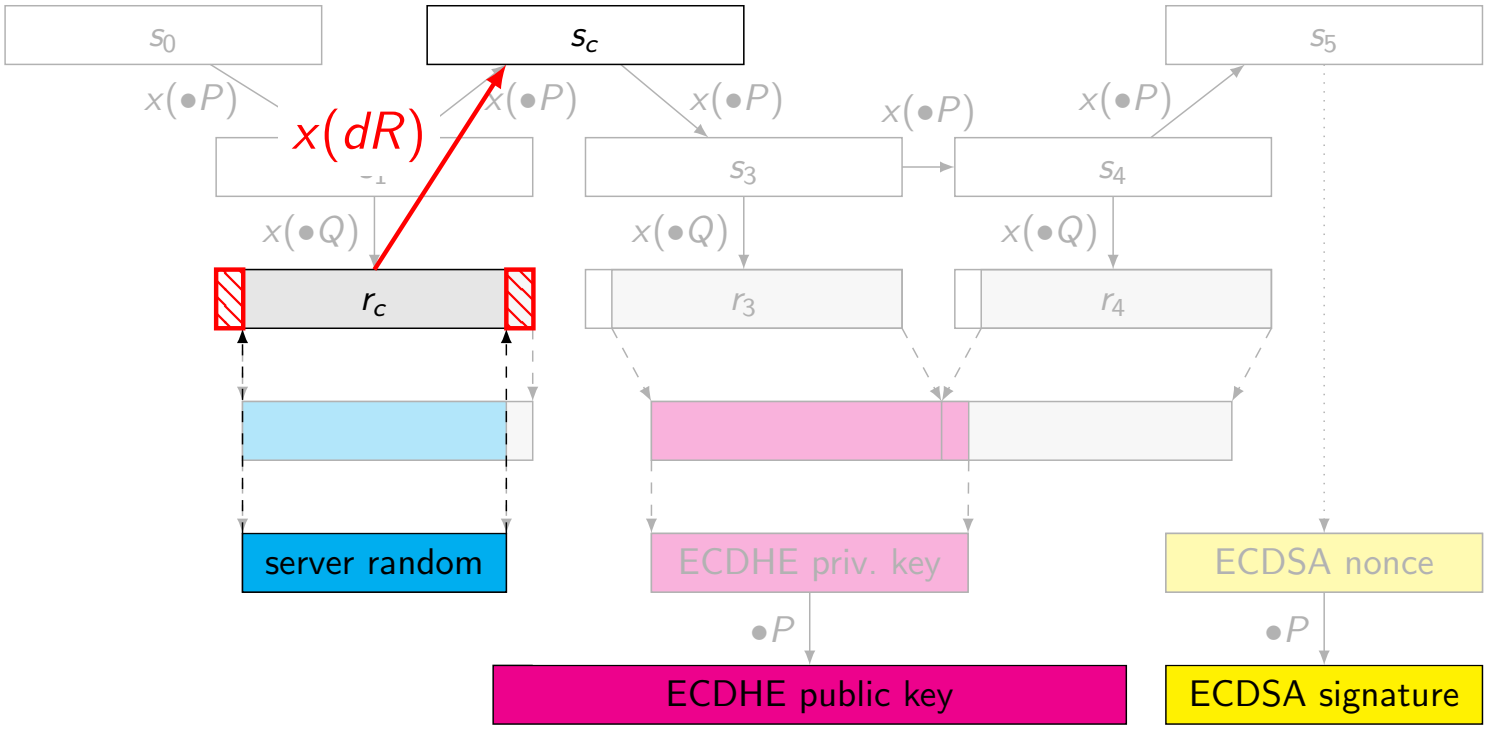
Graphics: Ruben Niederhagen.

Attack – Example: BSAFE-Java



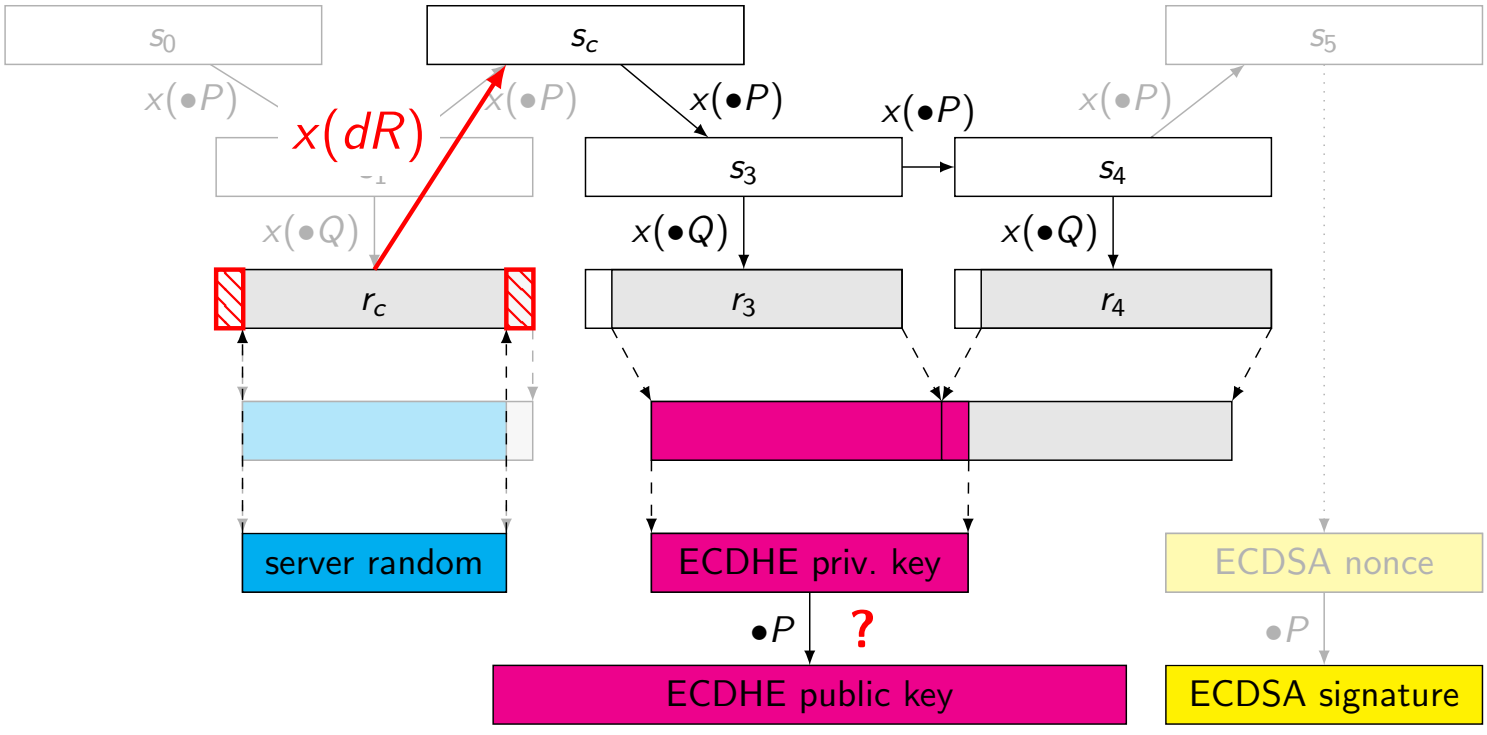
Graphics: Ruben Niederhagen.

Attack – Example: BSAFE-Java



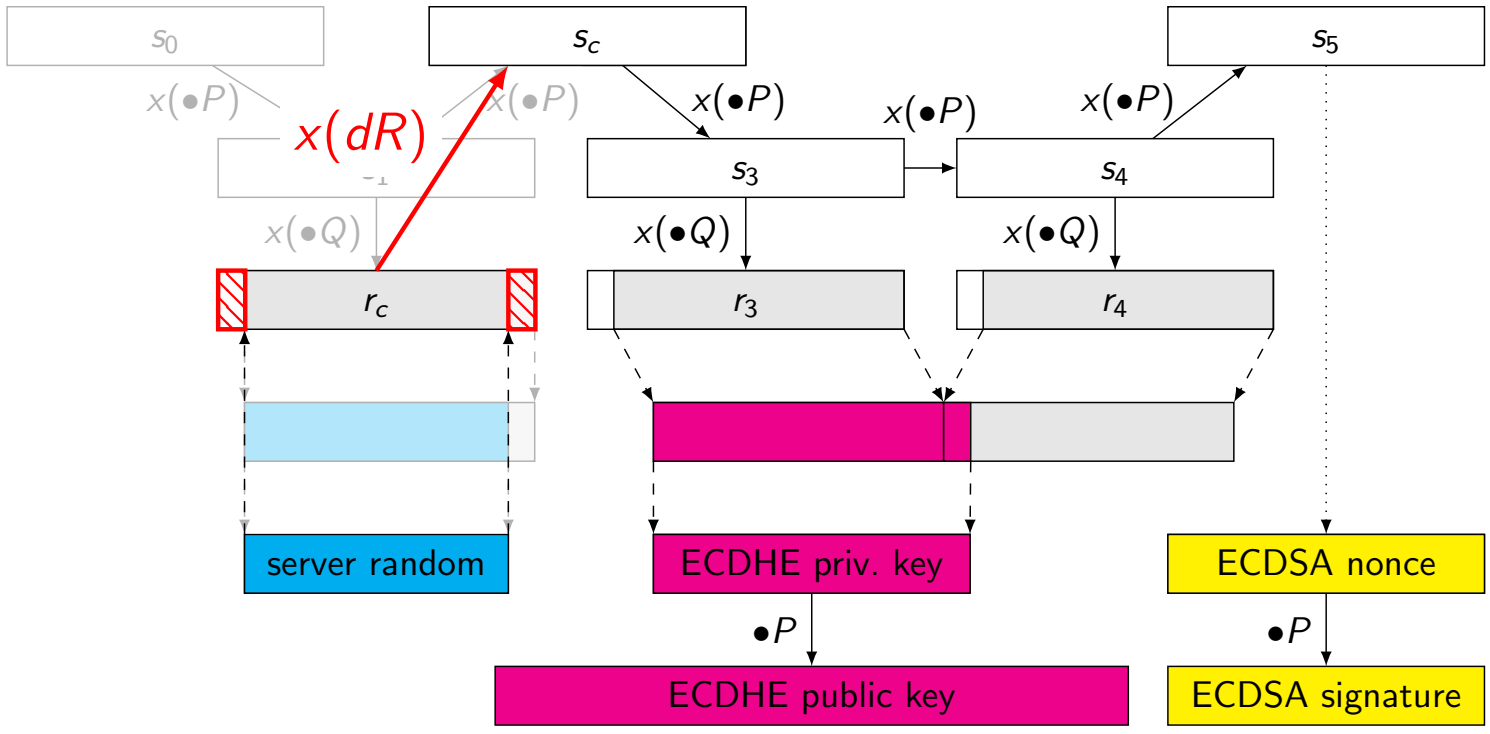
Graphics: Ruben Niederhagen.

Attack – Example: BSAFE-Java



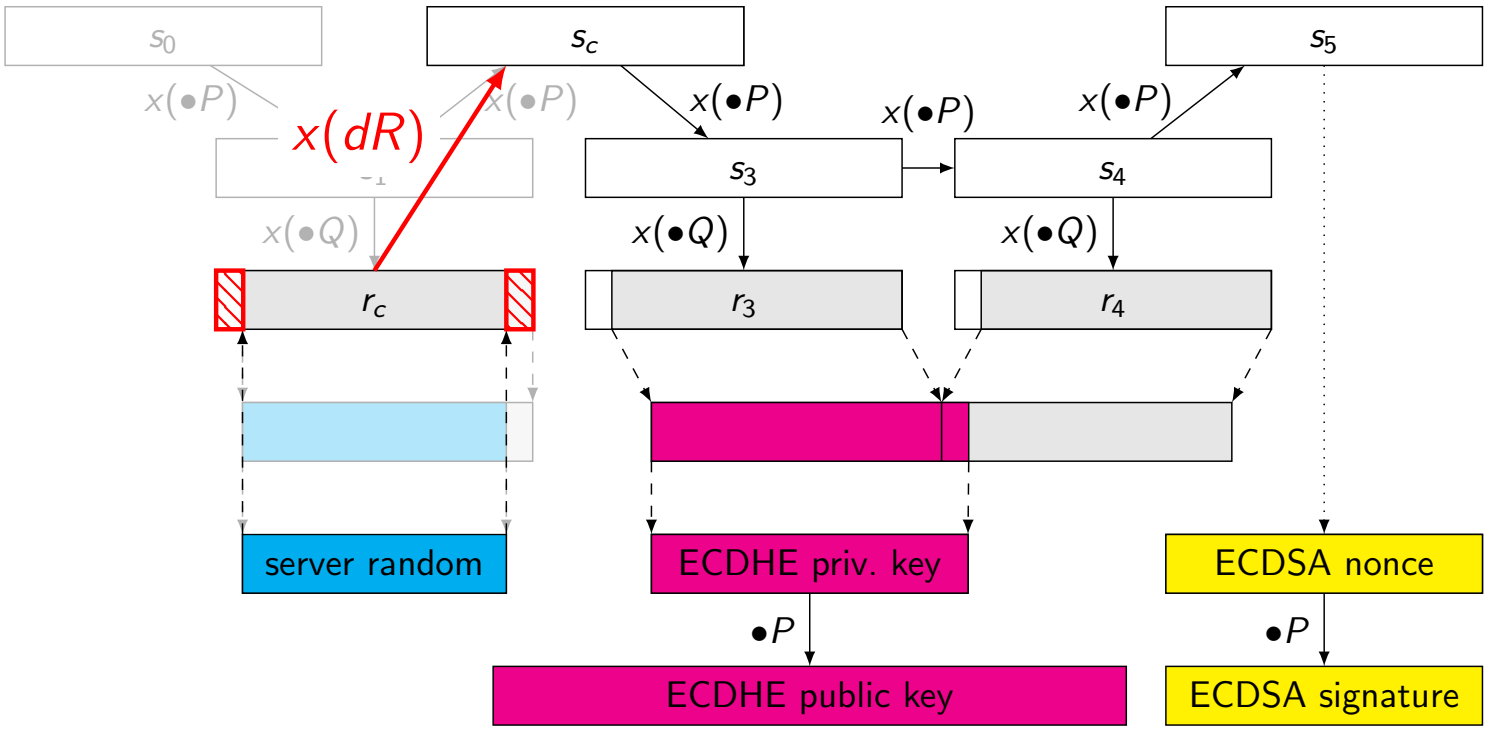
Graphics: Ruben Niederhagen.

Attack – Example: BSAFE-Java



Graphics: Ruben Niederhagen.

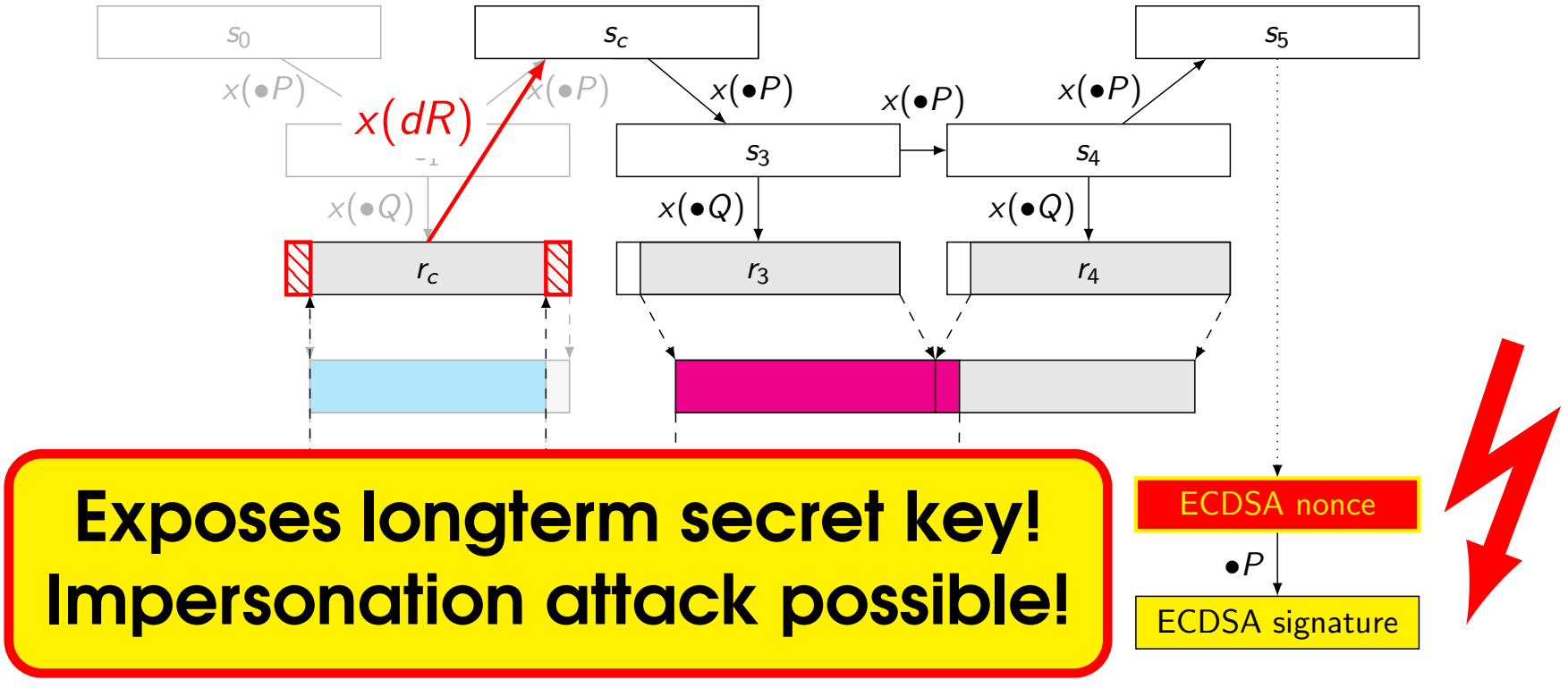
Attack – Example: BSAFE-Java



Graphics: Ruben Niederhagen.

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

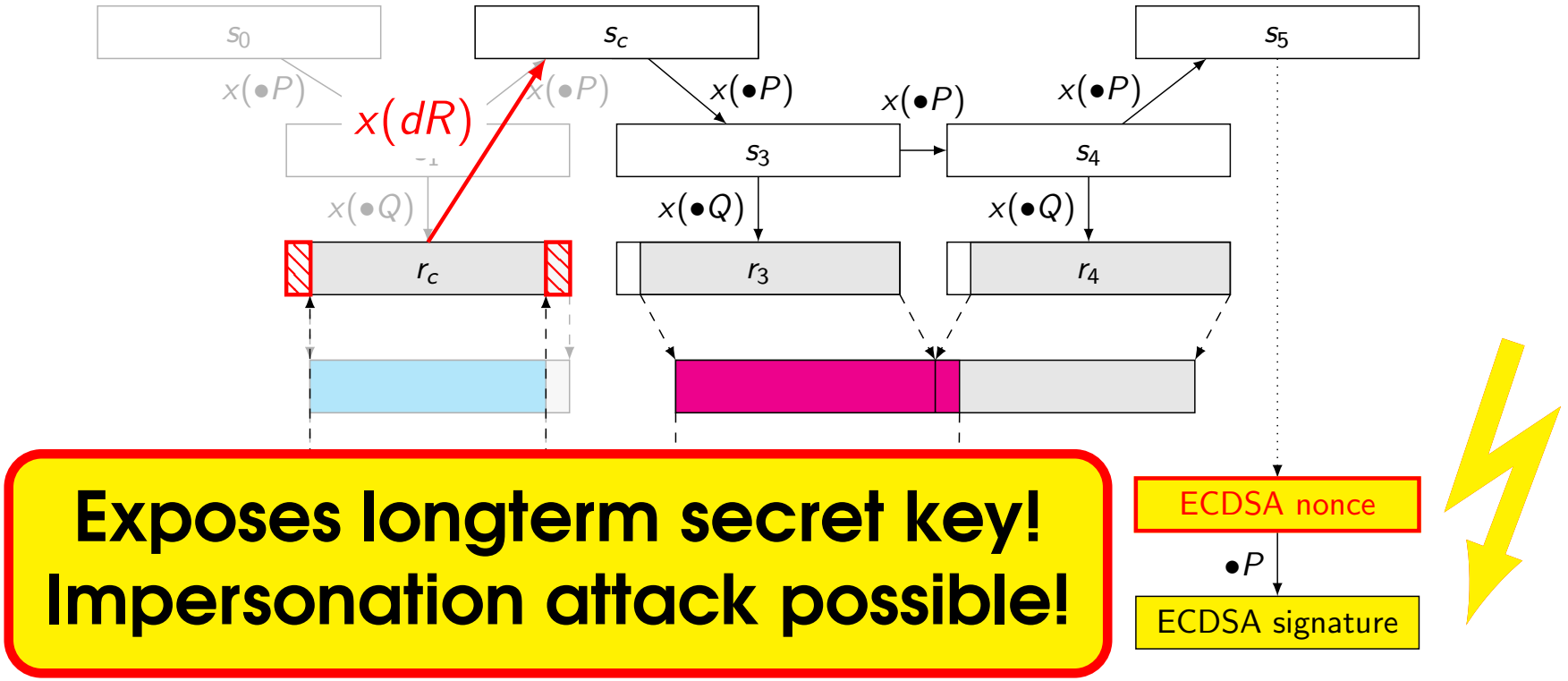
Attack – Example: BSAFE-Java



Graphics: Ruben Niederhagen.

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

Attack – Example: BSAFE-Java



Graphics: Ruben Niederhagen.

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

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

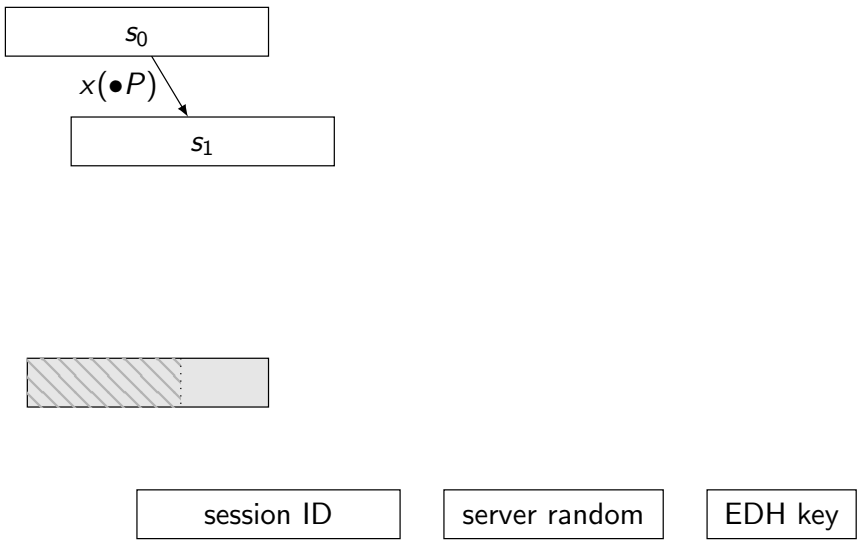
Attack – Example: BSAFE-C

Graphic thanks to Ruben Niederhagen.

Attack – Example: BSAFE-C

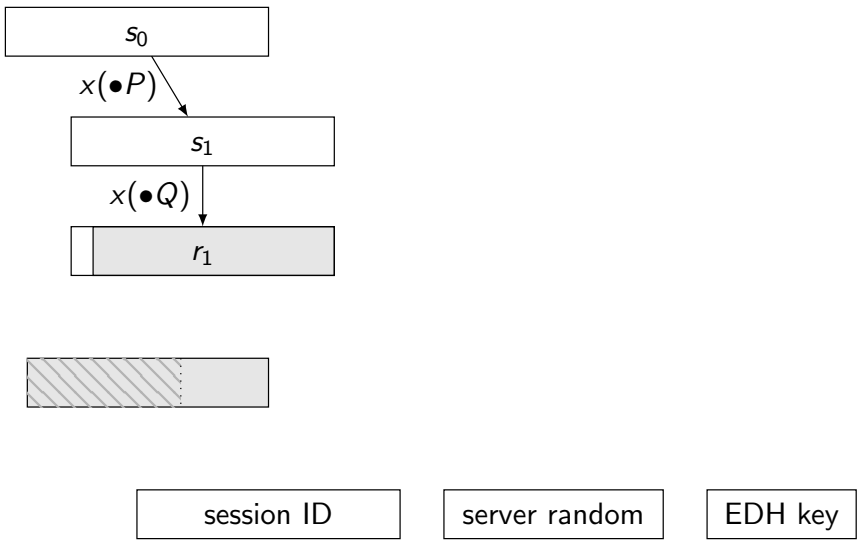
Graphic thanks to Ruben Niederhagen.

Attack – Example: BSAFE-C



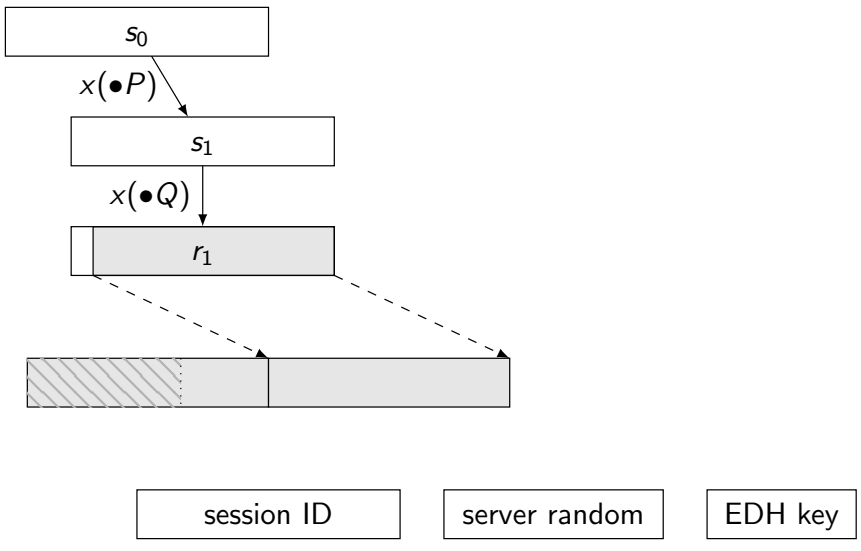
Graphic thanks to Ruben Niederhagen.

Attack – Example: BSAFE-C



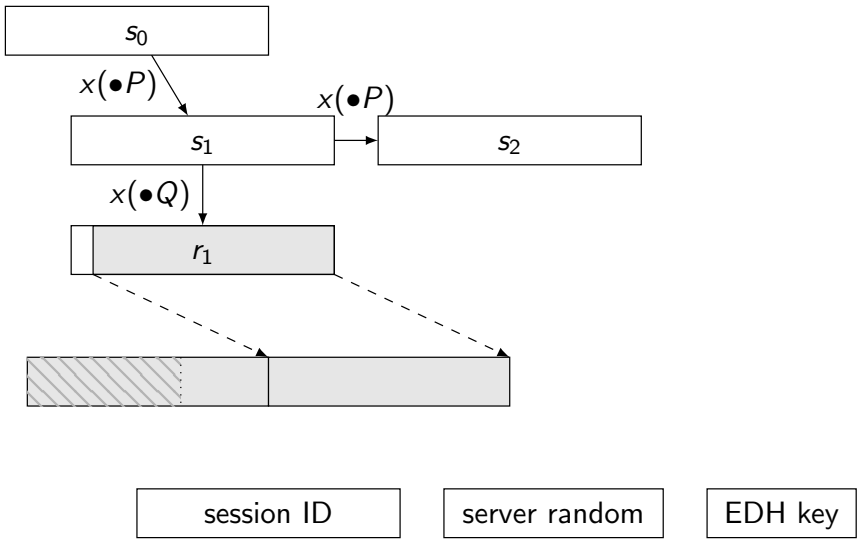
Graphic thanks to Ruben Niederhagen.

Attack – Example: BSAFE-C



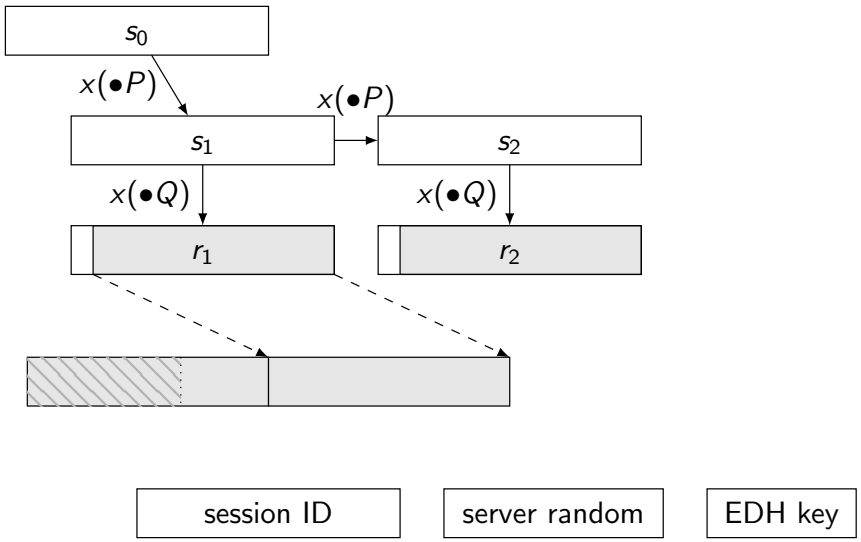
Graphic thanks to Ruben Niederhagen.

Attack – Example: BSAFE-C



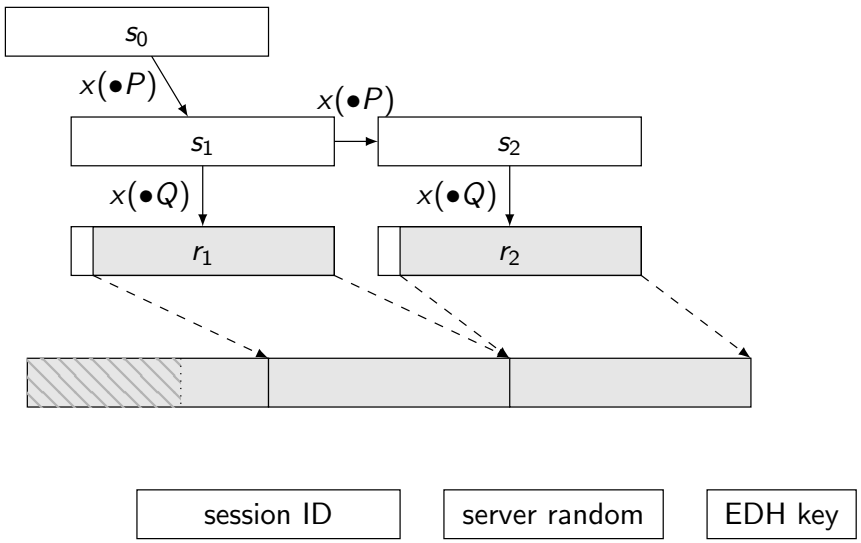
Graphic thanks to Ruben Niederhagen.

Attack – Example: BSAFE-C



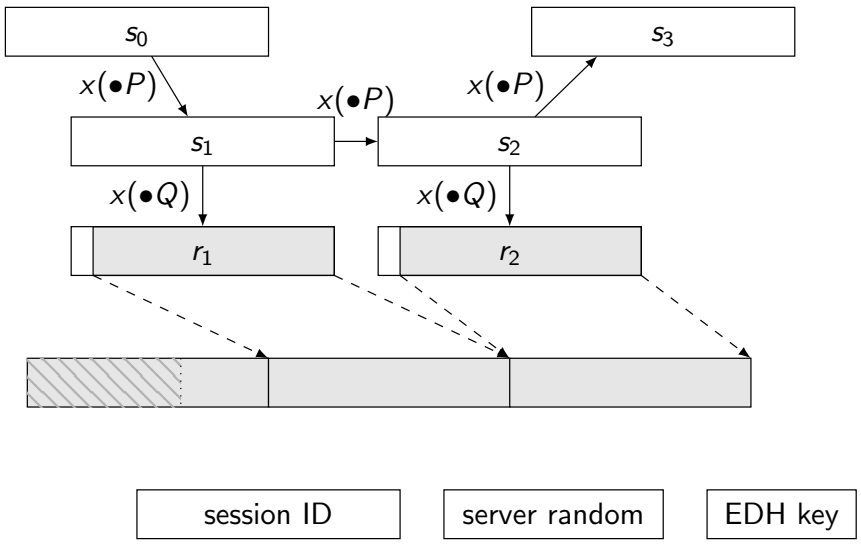
Graphic thanks to Ruben Niederhagen.

Attack – Example: BSAFE-C



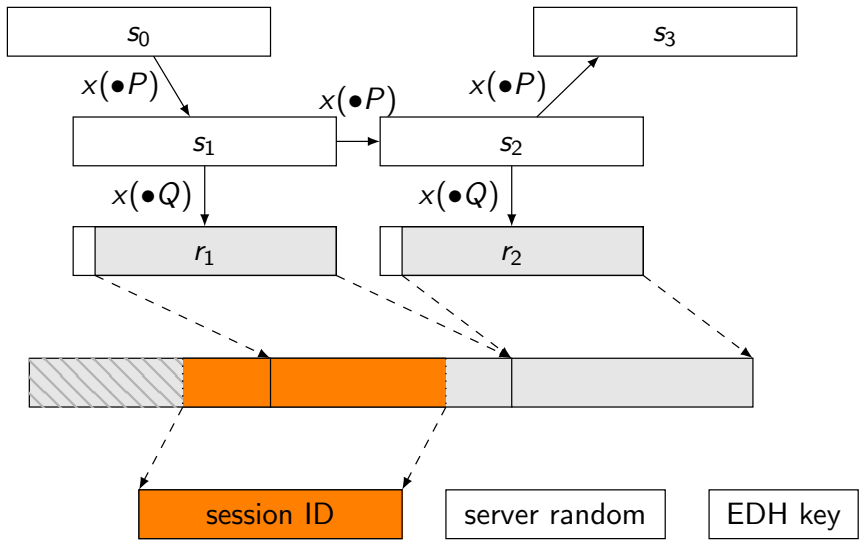
Graphic thanks to Ruben Niederhagen.

Attack – Example: BSAFE-C



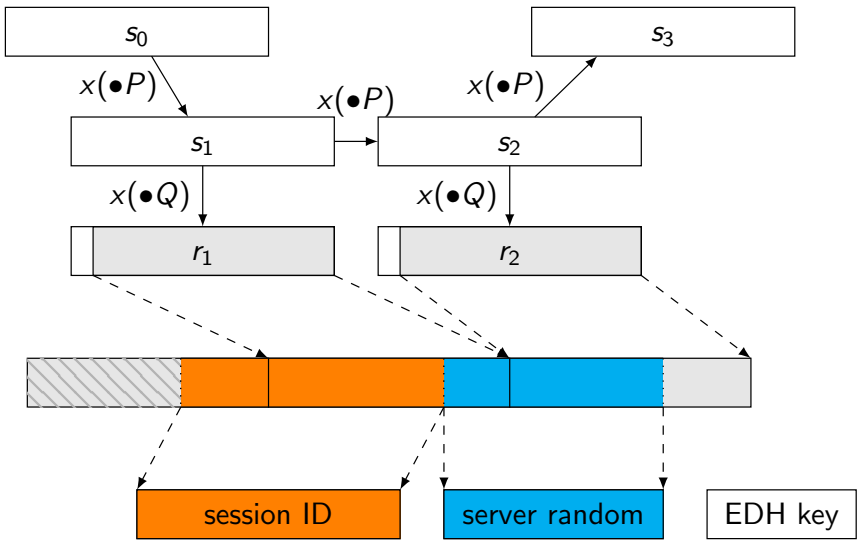
Graphic thanks to Ruben Niederhagen.

Attack – Example: BSAFE-C



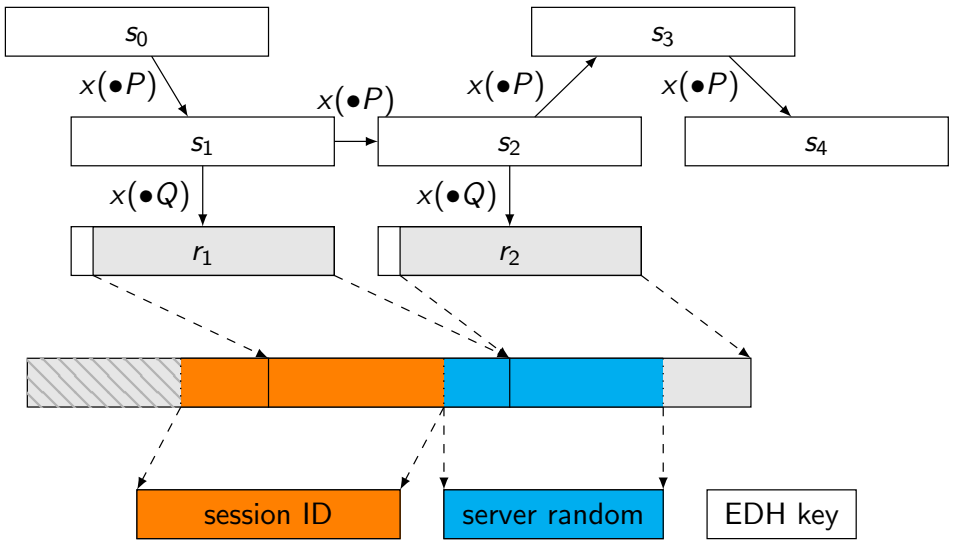
Graphic thanks to Ruben Niederhagen.

Attack – Example: BSAFE-C



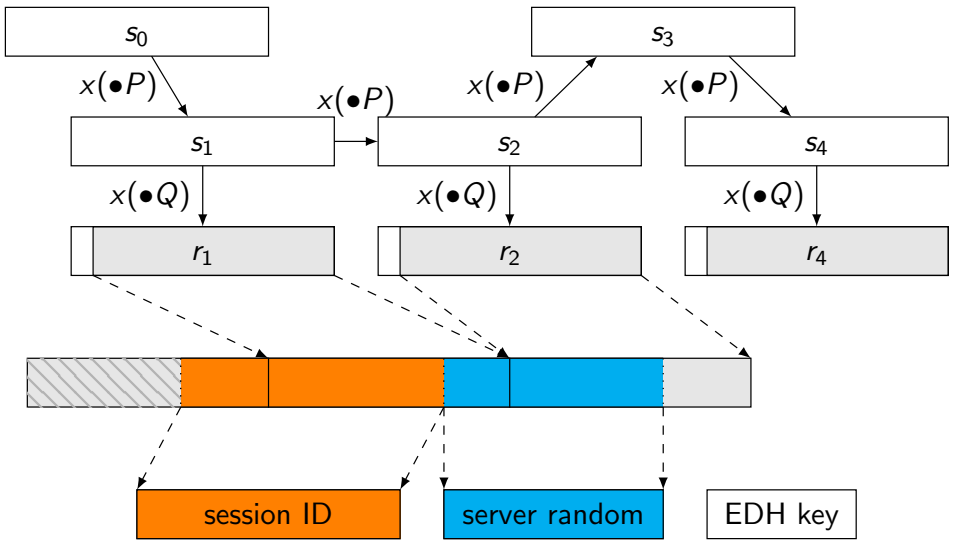
Graphic thanks to Ruben Niederhagen.

Attack – Example: BSAFE-C



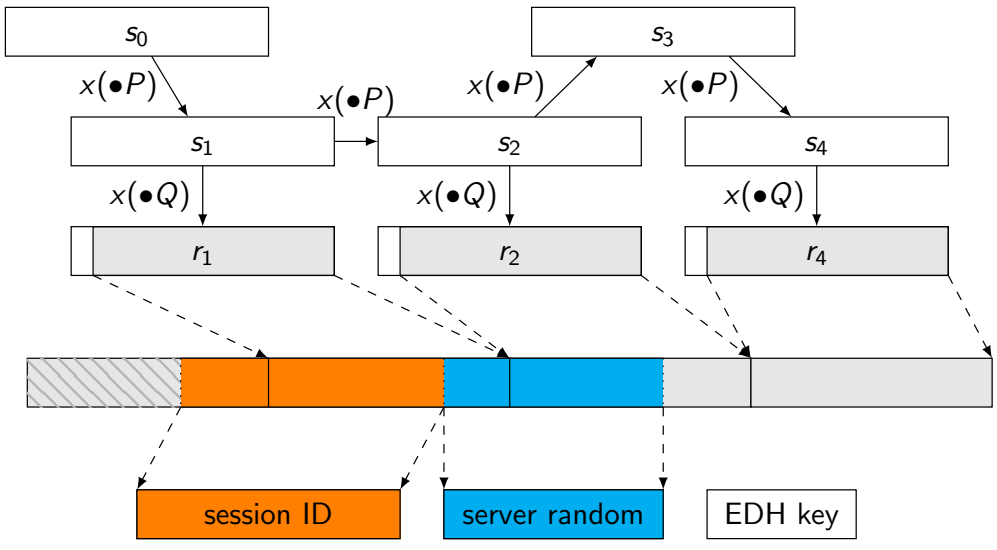
Graphic thanks to Ruben Niederhagen.

Attack – Example: BSAFE-C



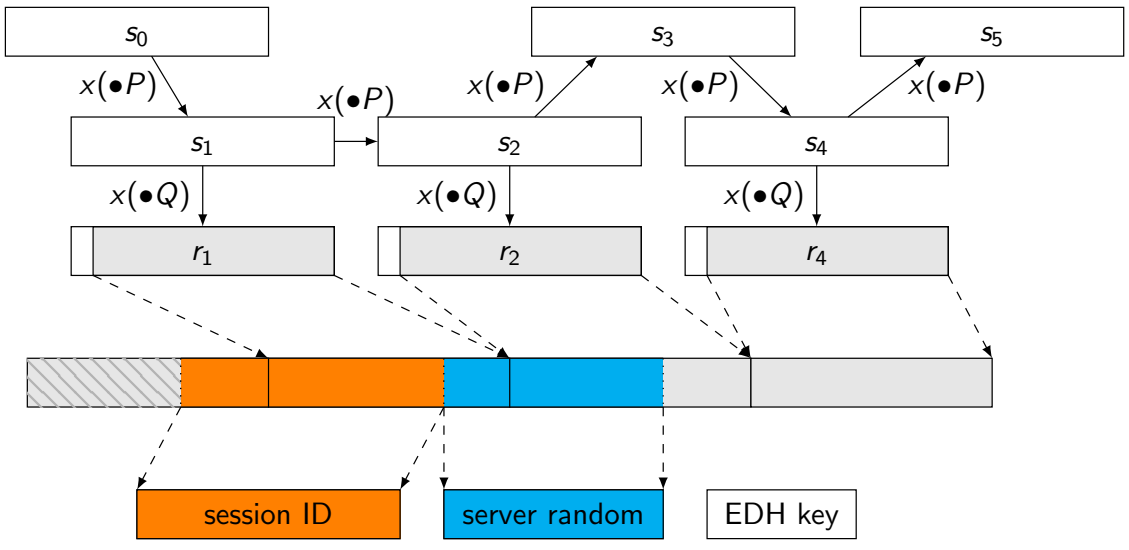
Graphic thanks to Ruben Niederhagen.

Attack – Example: BSAFE-C



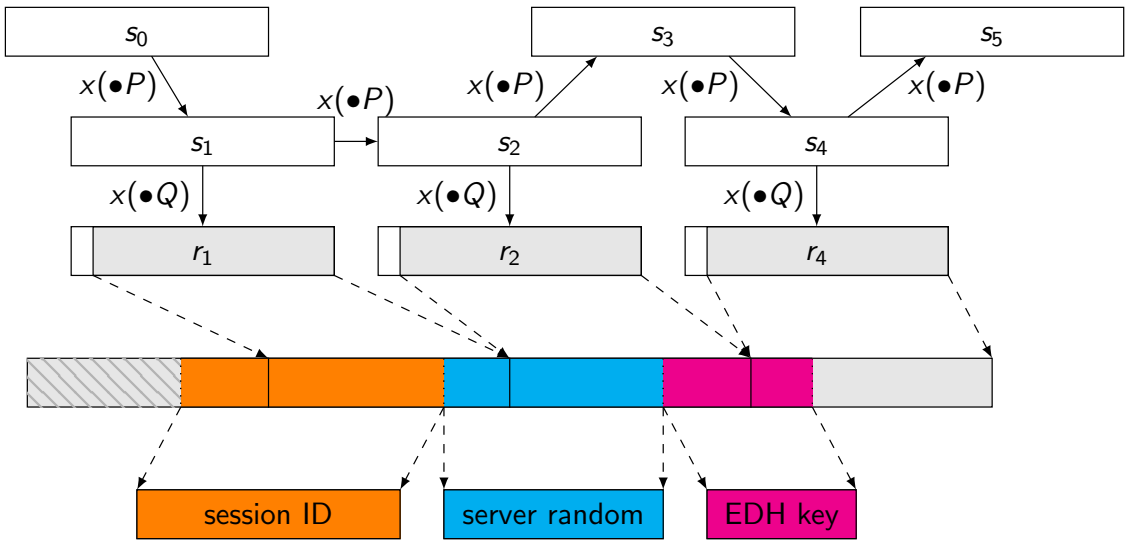
Graphic thanks to Ruben Niederhagen.

Attack – Example: BSAFE-C



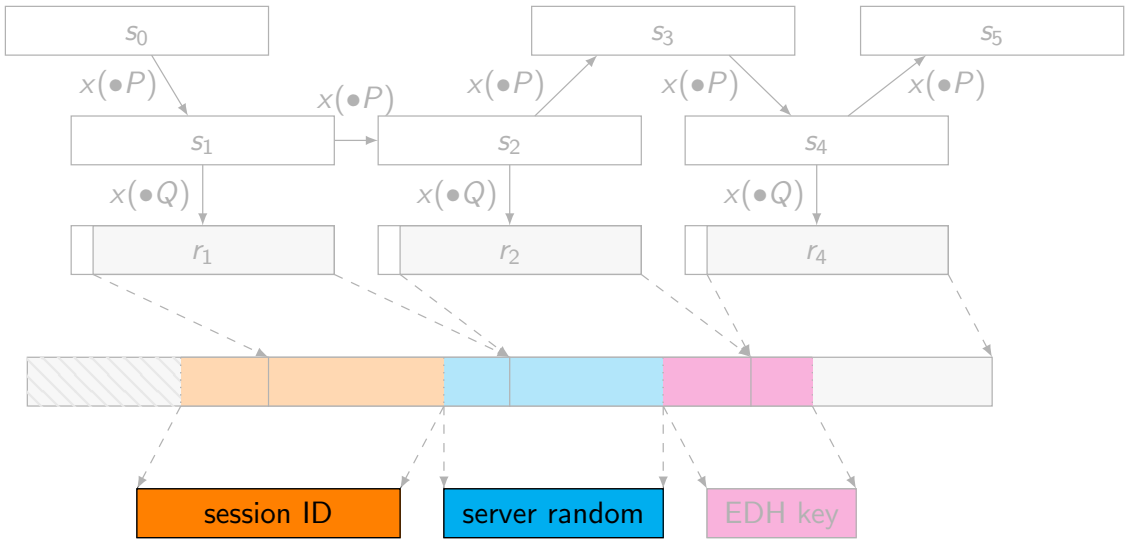
Graphic thanks to Ruben Niederhagen.

Attack – Example: BSAFE-C



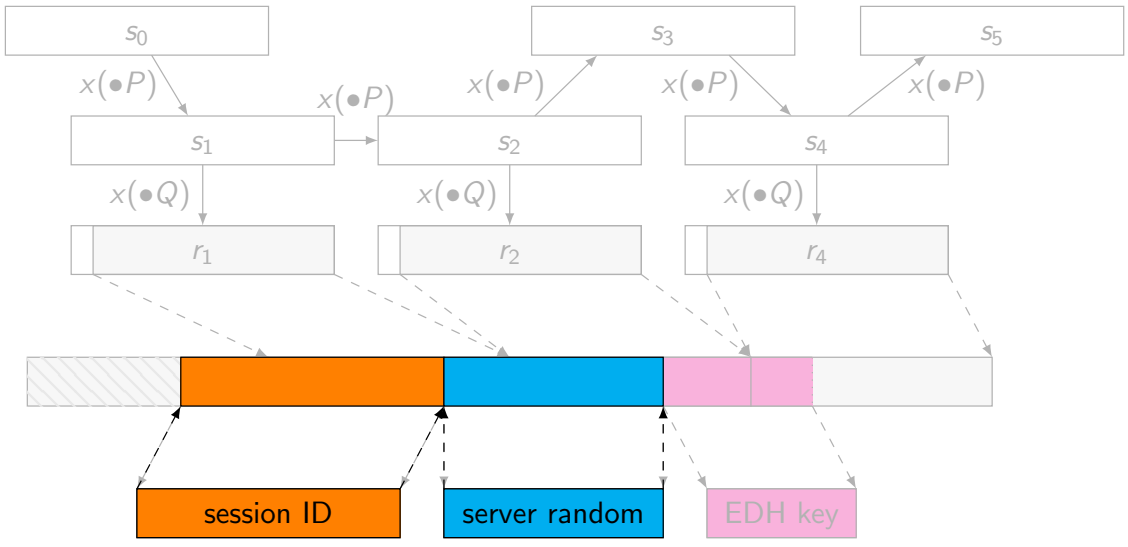
Graphic thanks to Ruben Niederhagen.

Attack – Example: BSAFE-C



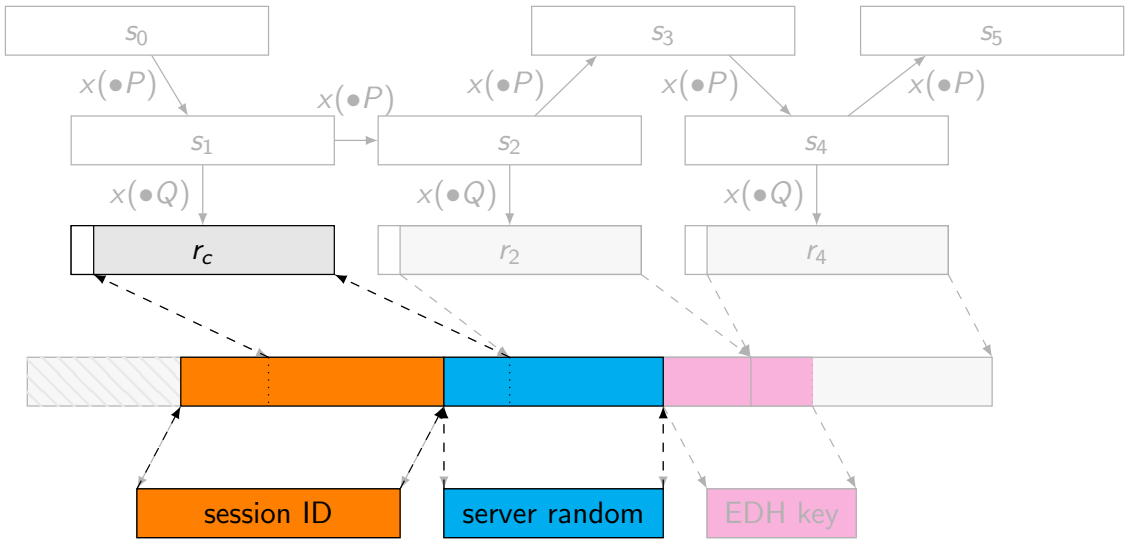
Graphic thanks to Ruben Niederhagen.

Attack – Example: BSAFE-C



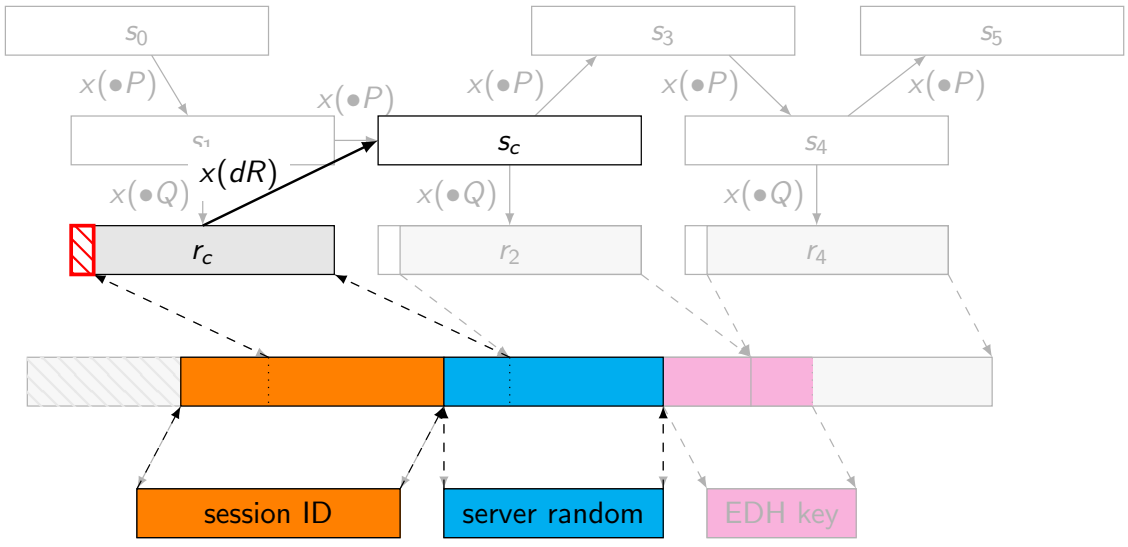
Graphic thanks to Ruben Niederhagen.

Attack – Example: BSAFE-C



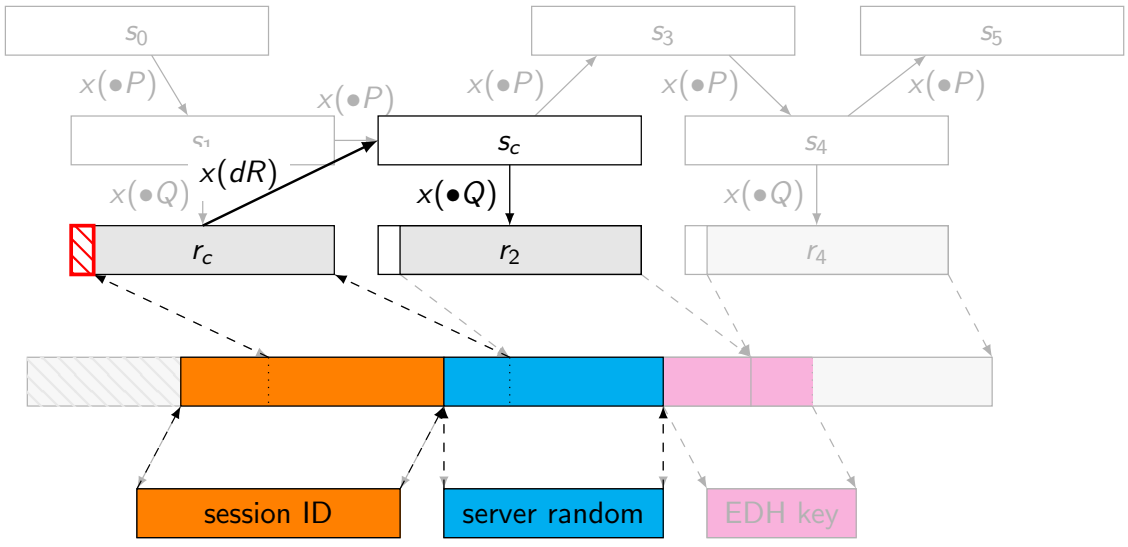
Graphic thanks to Ruben Niederhagen.

Attack – Example: BSAFE-C



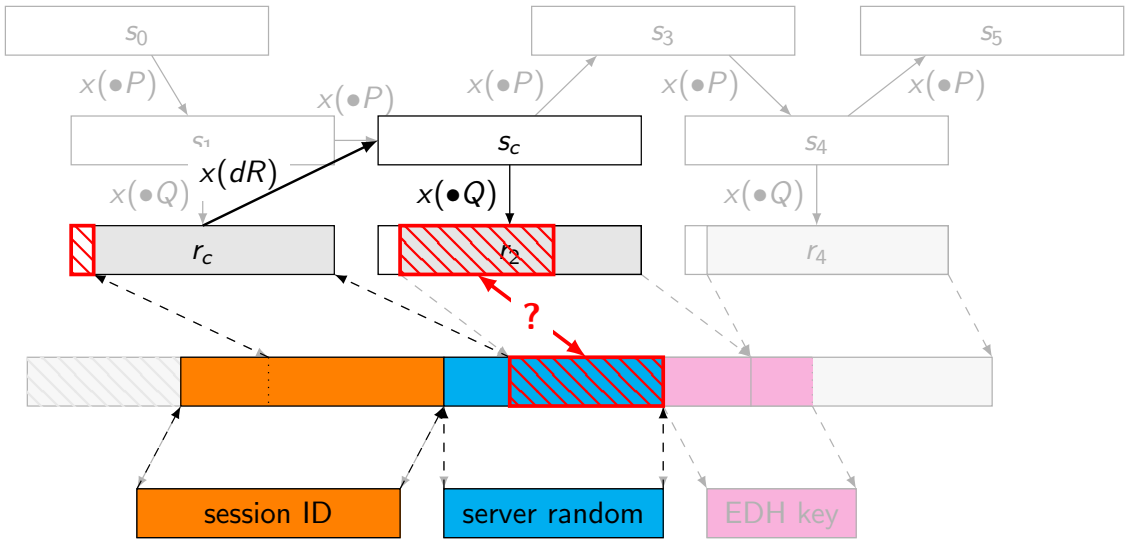
Graphic thanks to Ruben Niederhagen.

Attack – Example: BSAFE-C



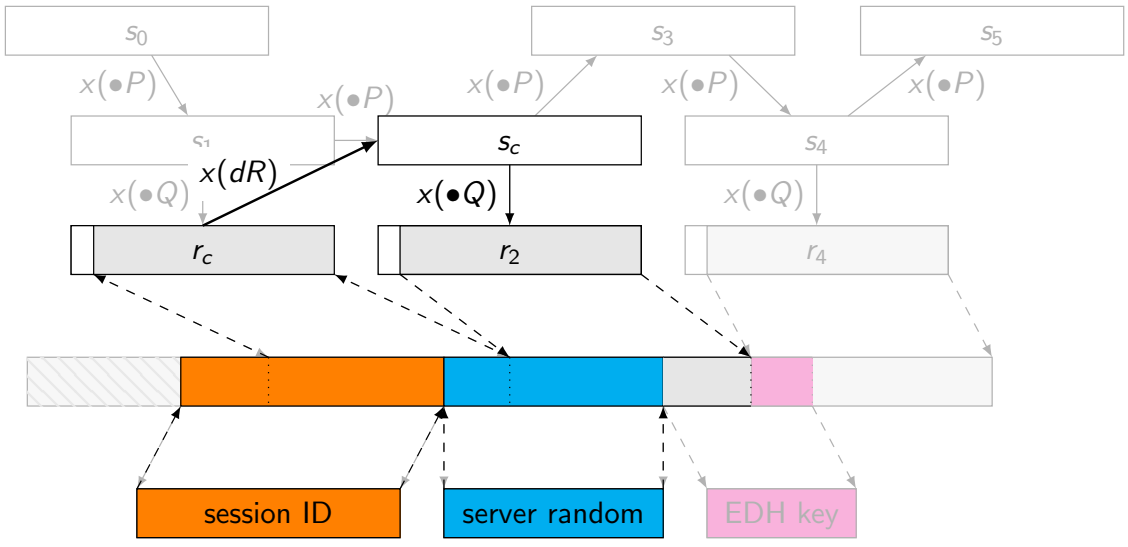
Graphic thanks to Ruben Niederhagen.

Attack – Example: BSAFE-C



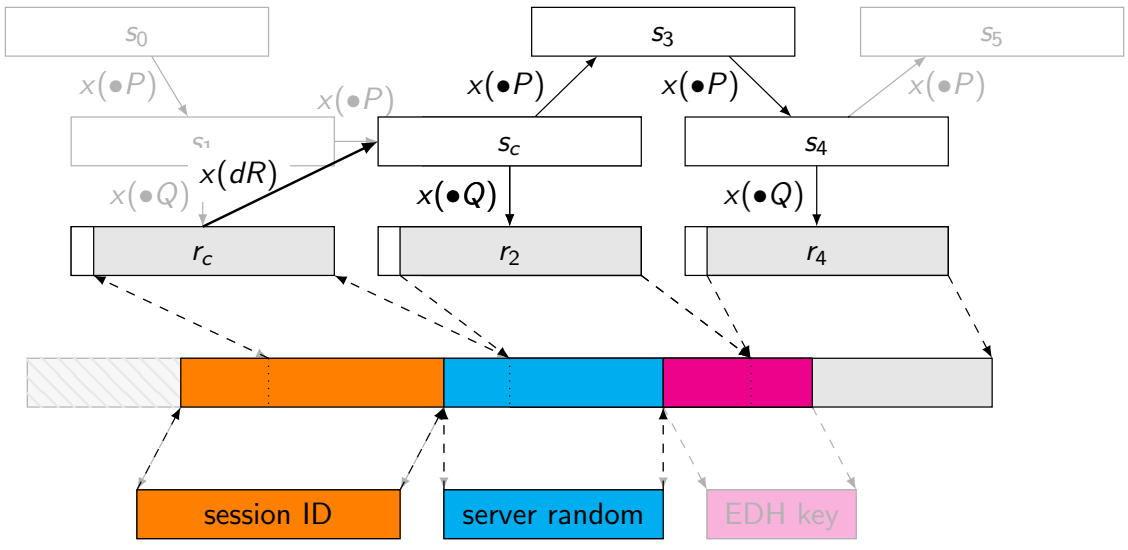
Graphic thanks to Ruben Niederhagen.

Attack – Example: BSAFE-C



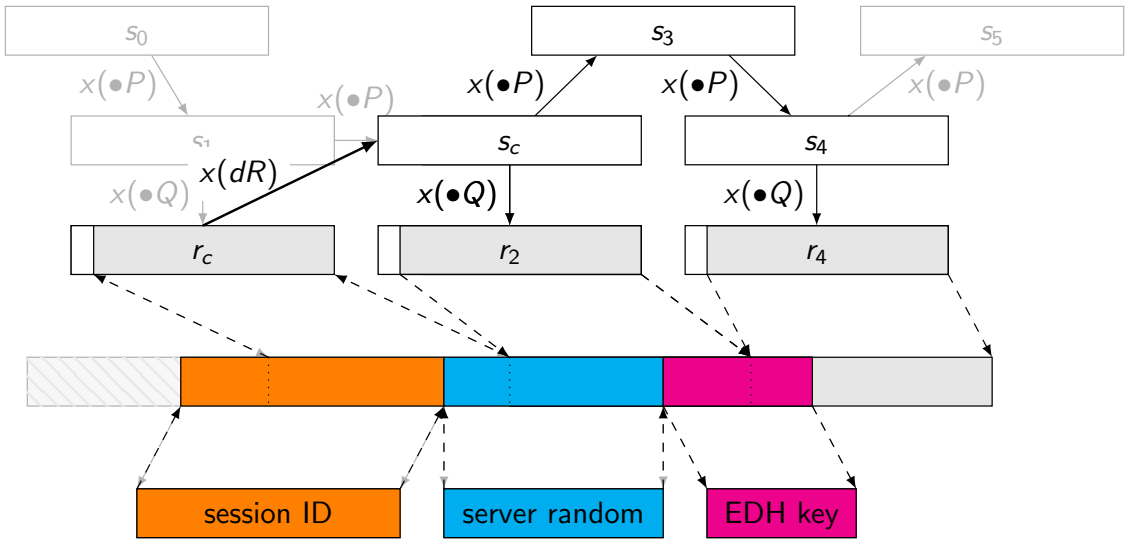
Graphic thanks to Ruben Niederhagen.

Attack – Example: BSAFE-C



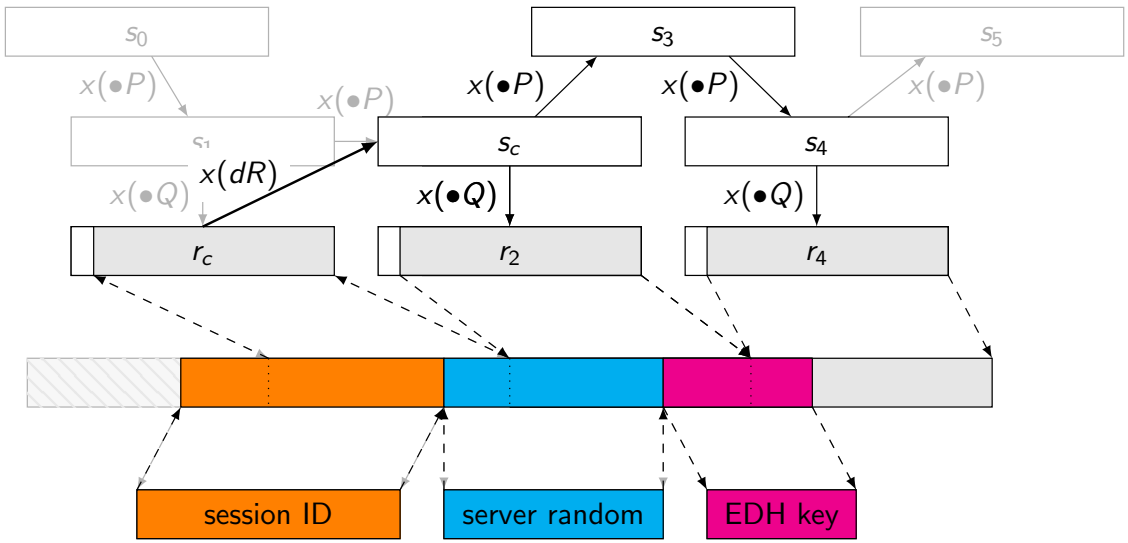
Graphic thanks to Ruben Niederhagen.

Attack – Example: BSAFE-C



Graphic thanks to Ruben Niederhagen.

Attack – Example: BSAFE-C



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

RELATED VIDEO

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.
- 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

How did we get here . . .

Full TODO list

- Design: Construct a PRNG that secretly contains a back door.
- Evaluation: Publish statements that the PRNG is secure.
- Standardization: Edit standards to include the PRNG.
- Standardization maintenance: Monitor changes to the PRNG standard, and counteract changes that make the PRNG more difficult to exploit.
- Auxiliary standardization: If necessary modify other standards to make the PRNG easier to exploit.
- Selection, implementation, and deployment: Provide incentives to cryptographic libraries to implement this PRNG.
- Attack optimization: Reduce the cost of computation required to exploit the back door, through algorithmic improvements and through influencing the way the PRNG is used in practice.

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.

Miles Smid's 2004 slides say
"ANSI X9.82 Concepts submitted as input to ISO/IEC CD 18031. (See Debby Wallner)".

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.

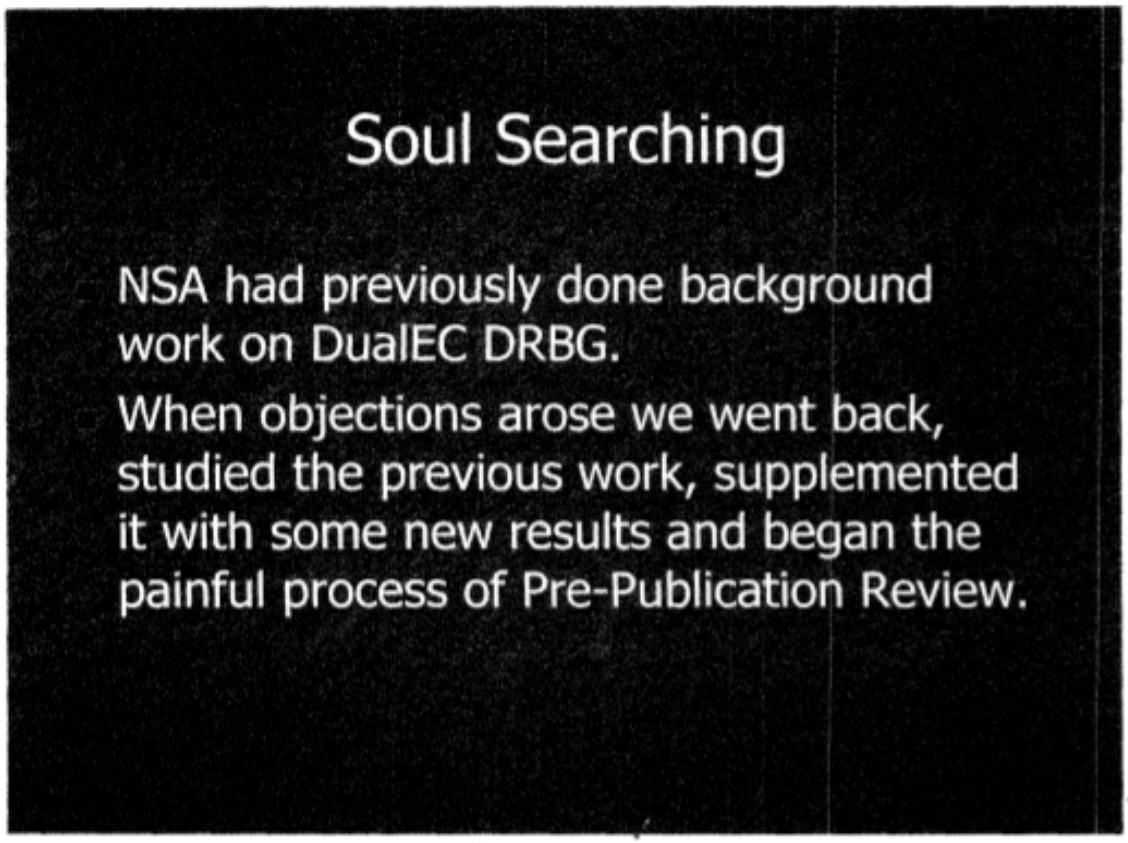
Kelsey 20 Dec 2013

Discussed in X9 Meeting

- Didn't seem like a real threat
- Obvious choice would have been to generate P and Q in a verifiably random way, make those the new system parameters.
  - At least one vendor had implemented with original P,Q.
- Instead, we allowed implementers to generate their own P and Q in a verifiably random way.
  - As far as we know, nobody actually did this..

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]]

Why does nobody use a different Q?

Appendix A: (Normative) Application-Specific Constants

A.1 Constants for the Dual_EC_DRBG

The **Dual_EC_DRBG** requires the specifications of an elliptic curve and two points on the elliptic curve. One of the following NIST approved curves with associated points **shall** be used in applications requiring certification under FIPS 140-2. More details about these curves may be found in FIPS PUB 186-3, the Digital Signature Standard.

Fake Math!

One might wonder if it would be desirable to truncate more than this amount. The obvious drawback to such an approach is that increasing the truncation amount hinders the performance. However, there is an additional reason that argues against increasing the truncation. Consider the case where the low s bits of each x-coordinate are kept. Given some subinterval I of length 2^s contained in $[0, p)$, and letting $N(I)$ denote the number of x-coordinates in I, recent results on the distribution of x-coordinates in $[0, p)$ provide the following bound:

$$\left|\frac{N(I)}{(p/2)} - \frac{2^s}{p}\right| < \frac{k * log^2 p}{\sqrt{p}},$$

where k is some constant derived from the asymptotic estimates given in [Shparlinski]. For the case of P-521, this is roughly equivalent to:

$$\left|N(I) - 2^{(s-1)}\right| < k * 2^{277},$$

where the constant k is independent of the value of s. For $s < 2^{277}$, this inequality is weak and provides very little support for the notion that these truncated x-coordinates are uniformly distributed. On the other hand, the larger the value of s, the sharper this inequality becomes, providing stronger evidence that the associated truncated x-coordinates are uniformly distributed. Therefore, by keeping truncation to an acceptable minimum, the performance is increased, and certain guarantees can be made about the uniform distribution of the resulting truncated quantities. Further discussion of the uniformity of the truncated x-coordinates is found in [Gurel], where the form of the prime defining the field is also taken into account.

ISO standard

- NIST history is relatively well documented, partially because of FOIA and partially because of NIST efforts to clean up.
- But ISO standard 18031 came first (2005). NYT article says

  Classified N.S.A. memos appear to confirm that the fatal weakness, discovered by two Microsoft cryptographers in 2007, was engineered by the agency. The N.S.A. wrote the standard and aggressively pushed it on the international group, privately calling the effort "a challenge in finesse."
  "Eventually, N.S.A. became the sole editor," the memo says.

ATTACHMENT 10 TO SC27 N3685

US National Body comments on ISO/IEC 2nd CD 18031

Date: **20030822**

1	2	(3)	4	5	
NB[1]	**Clause No./ Subclause No./ Annex** (e.g. 3.1)	**Paragraph/ Figure/Table/ Note** (e.g. Table 1)	**Type of com-ment[2]**	**Comment (justification for change) by the NB**	
US	Whole document		te	The U.S. National Body has reviewed ISO/IEC 2nd CD 18031, N3578. We feel that this document is lacking sufficient depth in many areas and simply is not developed enough to be an ISO standard which encompasses both Non-deterministic and Deterministic Random Bit Generation. We do feel that ANSI X9.82 Random Bit Generation standardization work is much further developed and should be used as the basis for this ISO standard. To make ISO/IEC 18031 consistent with X9.82 would require extensive commenting and revisions. To better progress this standard, the U.S. has instead developed a contribution for ISO that is consistent with ANSI X9.82, but written in ISO format. Furthermore, we believe this contribution will also be complementary to ISO/IEC 19790. We provide this contribution as an attachment, and propose that ISO further develop this contribution as their standard. Additionally, the U.S. recognizes that ANSI X9.82 is not an approved standard and still requires further work. As ANSI X9.82 develops, the U.S. will contribute these	

US 20070189527A1

(19) **United States**

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

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

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

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

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

(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`.

Snippets from the patent application

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

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

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

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.

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

Can one keep a back door closed?

The following product families do utilize Dual_EC_DRBG, but do not use the pre-defined points cited by NIST:

1. ScreenOS*

*ScreenOS does make use of the Dual_EC_DRBG standard, but is designed to not use Dual_EC_DRBG as its primary random number generator. ScreenOS uses it in a way that should not be vulnerable to the possible issue that has been brought to light. Instead of using the NIST recommended curve points it uses self-generated basis points and then takes the output as an input to FIPS/ANSI X.9.31 PRNG, which is the random number generator used in ScreenOS cryptographic operations.

Can one keep a back door closed?

The following product families do utilize Dual_EC_DRBG, but do not use the pre-defined points cited by NIST:

1. ScreenOS*

*ScreenOS does make use of the Dual_EC_DRBG standard, but is designed to not use Dual_EC_DRBG as its primary random number generator. ScreenOS uses it in a way that should not be vulnerable to the possible issue that has been brought to light. Instead of using the NIST recommended curve points it uses self-generated basis points and then takes the output as an input to FIPS/ANSI X.9.31 PRNG, which is the random number generator used in ScreenOS cryptographic operations.

17 Dec 2015:

Important Juniper Security Announcement

Soon identified to include changes to

Can one keep a back door closed?

The following product families do utilize Dual_EC_DRBG, but do not use the pre-defined points cited by NIST:

1. ScreenOS*

*ScreenOS does make use of the Dual_EC_DRBG standard, but is designed to not use Dual_EC_DRBG as its primary random number generator. ScreenOS uses it in a way that should not be vulnerable to the possible issue that has been brought to light. Instead of using the NIST recommended curve points it uses self-generated basis points and then takes the output as an input to FIPS/ANSI X.9.31 PRNG, which is the random number generator used in ScreenOS cryptographic operations.

17 Dec 2015:

Important Juniper Security Announcement

Soon identified to include changes to their Dual EC parameters.

Can one keep a back door closed?

The following product families do utilize Dual_EC_DRBG, but do not use the pre-defined points cited by NIST:

1. ScreenOS*

*ScreenOS does make use of the Dual_EC_DRBG standard, but is designed to not use Dual_EC_DRBG as its primary random number generator. ScreenOS uses it in a way that should not be vulnerable to the possible issue that has been brought to light. Instead of using the NIST recommended curve points it uses self-generated basis points and then takes the output as an input to FIPS/ANSI X.9.31 PRNG, which is the random number generator used in ScreenOS cryptographic operations.

17 Dec 2015:

Important Juniper Security Announcement

Soon identified to include changes to their Dual EC parameters.

Ralf Philipp Weinmann: Some Analysis of the Backdoored Backdoor

Full postmortem shows

- Juniper used Dual EC as RNG for Screen OS.
- Juniper used their own points.
- ScreenOS had a bug (?) that exposed raw Dual EC output.

Can one keep a back door closed?

The following product families do utilize Dual_EC_DRBG, but do not use the pre-defined points cited by NIST:

1. ScreenOS*

*ScreenOS does make use of the Dual_EC_DRBG standard, but is designed to not use Dual_EC_DRBG as its primary random number generator. ScreenOS uses it in a way that should not be vulnerable to the possible issue that has been brought to light. Instead of using the NIST recommended curve points it uses self-generated basis points and then takes the output as an input to FIPS/ANSI X.9.31 PRNG, which is the random number generator used in ScreenOS cryptographic operations.

17 Dec 2015:

Important Juniper Security Announcement

Soon identified to include changes to their Dual EC parameters.

Ralf Philipp Weinmann: Some Analysis of the Backdoored Backdoor

Full postmortem shows

- Juniper used Dual EC as RNG for Screen OS.
- Juniper used their own points.
- ScreenOS had a bug (?) that exposed raw Dual EC output.
- Around 2012 somebody changed these points to yet other points.

More on the standardization ecosystem

- Apr 2018 Simon and Speck do not get included in ISO lightweight standard, breaking with ISO tradition of being very permissive.
  This was a large effort, but enough cryptographers got involved and many countries were reasonable.
  But: ISO is pay to play and hard to get in for some countries.
- Nov 2018 OCB2 (standardized in ISO/IEC 19772:2009) broken in 3 different ways.
- OCB1 and OCB3 (RFC 7253) are not affected.
- 28 Aug 2018 IETF publishes TLS 1.3 in RFC 8446.

More on the standardization ecosystem

- Apr 2018 Simon and Speck do not get included in ISO lightweight standard, breaking with ISO tradition of being very permissive.
  This was a large effort, but enough cryptographers got involved and many countries were reasonable.
  But: ISO is pay to play and hard to get in for some countries.
- Nov 2018 OCB2 (standardized in ISO/IEC 19772:2009) broken in 3 different ways.
- OCB1 and OCB3 (RFC 7253) are not affected.
- 28 Aug 2018 IETF publishes TLS 1.3 in RFC 8446.
- Lots of bad ideas got rejected in the making of TLS 1.3.
- IETF / CFRG is open & welcoming, remote participation is possible. Consensus: RFC 7258 – Pervasive Monitoring Is An Attack.

Humming an Open Internet Demise in London?

By **Anthony Rutkowski**

Feb 25, 2018 12:20 PM PST | Comments: 12 | Views: 20,075 Comment | Print

In mid-March, the group dubbed by *Wired Magazine* 20 years ago as Crypto-Rebels and Anarchists — the IETF — is meeting in London. With what is likely some loud humming, the activists will likely seek to rain mayhem upon the world of network and societal security using extreme end-to-end encryption, and collaterally diminish some remaining vestiges of an "open internet." Ironically, the IETF uses what has become known as the "NRA defence": extreme encryption doesn't cause harm, criminals and terrorists do. The details and perhaps saving alternatives are described in this article.

URL

National Cyber Security Centre
a part of GCHQ

Blog post

TLS 1.3: better for individuals - harder for enterprises

URL

More on the standardization ecosystem

- Apr 2018 Simon and Speck do not get included in ISO lightweight standard, breaking with ISO tradition of being very permissive.
  This was a large effort, but enough cryptographers got involved and many countries were reasonable. But: ISO is pay to play and hard to get in for some countries.
- Nov 2018 OCB2 (standardized in ISO/IEC 19772:2009) broken in 3 different ways.
- OCB1 and OCB3 (RFC 7253) are not affected.
- Aug 2018 IETF publishes TLS 1.3 in RFC 8446.
- Lots of bad ideas got rejected in the making of TLS 1.3.
- IETF / CFRG is open & welcoming, remote participation is possible. Consensus: RFC 7258 – Pervasive Monitoring Is An Attack.

More on the standardization ecosystem

- Apr 2018 Simon and Speck do not get included in ISO lightweight standard, breaking with ISO tradition of being very permissive.
  This was a large effort, but enough cryptographers got involved and many countries were reasonable. But: ISO is pay to play and hard to get in for some countries.
- Nov 2018 OCB2 (standardized in ISO/IEC 19772:2009) broken in 3 different ways.
- OCB1 and OCB3 (RFC 7253) are not affected.
- Aug 2018 IETF publishes TLS 1.3 in RFC 8446.
- Lots of bad ideas got rejected in the making of TLS 1.3.
- IETF / CFRG is open & welcoming, remote participation is possible. Consensus: RFC 7258 – Pervasive Monitoring Is An Attack.
- Nov 2018 ETSI publishes surveillance-friendly variant.
- ETSI is fully pay to play. Interest groups make their own "standards", no oversight, no outside review.

ETSI releases standards for enterprise security and data centre management

Sophia Antipolis, 5 November 2018

eTLS was created by industry, for industry

ETSI Technical Committee CYBER has recently released a Middlebox Security Protocol specification, *Profile For Enterprise Network and Data Centre Access Control*, TS 103 523-3, known as Enterprise TLS or "eTLS". This specification was driven by, and fulfils, an industry need to perform vital data centre operations – whilst supporting a recently standardized version of TLS (1.3). Such required operations include compliance, troubleshooting, detection of attacks (such as malware activity, data exfiltration, DDoS incidents), and more, on encrypted networks – functions which are enabled in the presence of TLS 1.3 by eTLS.

eTLS allows data centre and enterprise network operators to meet their service agreements and legal mandates; eTLS protects users from being forced to revert to older, less secure protocols; and eTLS allows data centre operators and users visibility over who has access to their data.

eTLS, defined in TS 103 523-3, specifies an implementation variant of TLS 1.3. This variant is needed because TLS 1.3 removes support for certain key exchange methods, which prevents passive decryption of TLS 1.3 sessions at any scale. However, there are operational circumstances where passive decryption of sessions is necessary. Such situations generally occur where both the parties in a connection, and by inference the data being exchanged, are under the control of the same entity. eTLS uses a key exchange message that supports these use cases and provides visibility information to the end user

ETSI releases standards for enterprise security and data centre management

Sophia Antipolis, 5 November 2018

eTLS was created by industry, for industry

ETSI Technical Committee CYBER has recently released a Middlebox Security Protocol specification, *Profile For Enterprise Network and Data Centre Access Control*, TS 103 523-3, known as Enterprise TLS or "eTLS". This specification was driven by, and fulfils, an industry need to perform vital data centre operations – whilst supporting a recently standardized version of TLS (1.3). Such required operations include compliance, troubleshooting, detection of attacks (such as malware activity, data exfiltration, DDoS incidents), and more, on encrypted networks – functions which are enabled in the presence of TLS 1.3 by eTLS.

eTLS allows data centre and enterprise network operators to meet their service agreements and legal mandates; eTLS protects users from being forced to revert to older, less secure protocols; and eTLS allows data centre operators and users visibility over who has access to their data.

eTLS, defined in TS 103 523-3, specifies an implementation variant of TLS 1.3. This variant is needed because TLS 1.3 removes support for certain key exchange methods, which prevents passive decryption of TLS 1.3 sessions at any scale. However, there are operational circumstances where passive decryption of sessions is necessary. Such situations generally occur where both the parties in a connection, and by inference the data being exchanged, are under the control of the same entity. eTLS uses a key exchange message that supports these use cases and provides visibility information to the end user

References

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

On the Practical Exploitability of Dual EC DRBG in TLS Implementations with Stephen Checkoway, Matthew Fredrikson, Ruben Niederhagen, Adam Everspaugh, Matthew Green, Tanja Lange, Thomas Ristenpart, Daniel J. Bernstein, Jake Maskiewicz, Hovav Shacham. USENIX Security 2014.

Dual EC: A Standardized Back Door with Daniel J. Bernstein and Ruben Niederhagen. The New Codebreakers, LNCS 9100.

Where Did I Leave My Keys?: Lessons from the Juniper Dual EC Incident by Stephen Checkoway, Jacob Maskiewicz, Christina Garman, Joshua Fried, Shaanan Cohney, Matthew Green, Nadia Heninger, Ralf-Philipp Weinmann, Eric Rescorla, Hovav Shacham. Communications of the ACM.