Discussion exercise sheet 3

Marc-Philippe Bartholomä
Student Assistant for Network Security 2020
08 October 2020, HG F1

Administrative

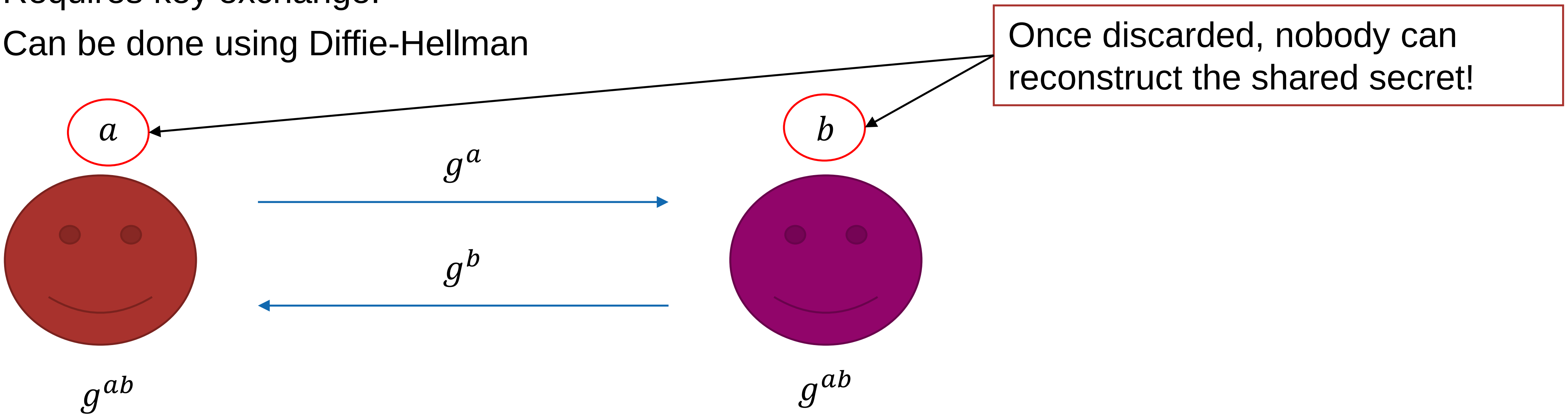
- Hand-in off this exercise sheet was extended until tomorrow, October 9th, 23:59!
- You can leave before we discuss solutions.

- Deadline for Project 1 is November 6th
- There is only <u>one</u> question hour beforehand: October 22nd
- Keep in mind that questions must be posted one day ahead!

No solution spoilers for now!

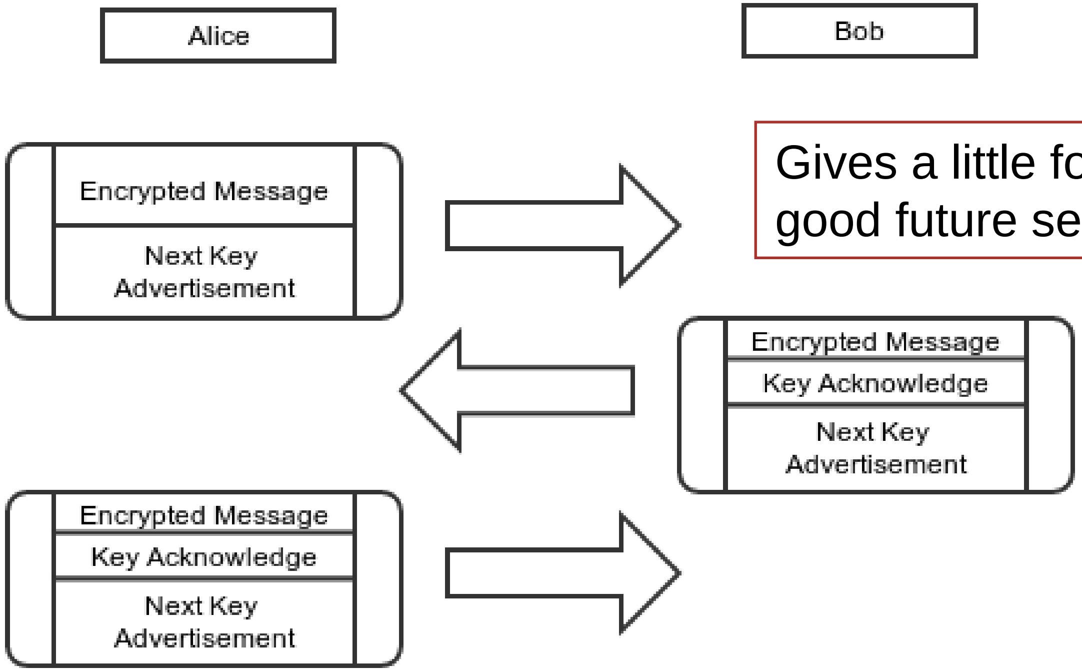
Question 1

- Related Material: 03-04-TLS: slide 46, 01b-crypto-refresher: slide 29
- Question: Perfect Forward Secrecy in Messaging
- Background:
  - If a scheme provides forward secrecy, secrecy of data is guaranteed even if the key is compromised in the future.
  - Requires key exchange!
  - Can be done using Diffie-Hellman


Once discarded, nobody can reconstruct the shared secret!

Question 1

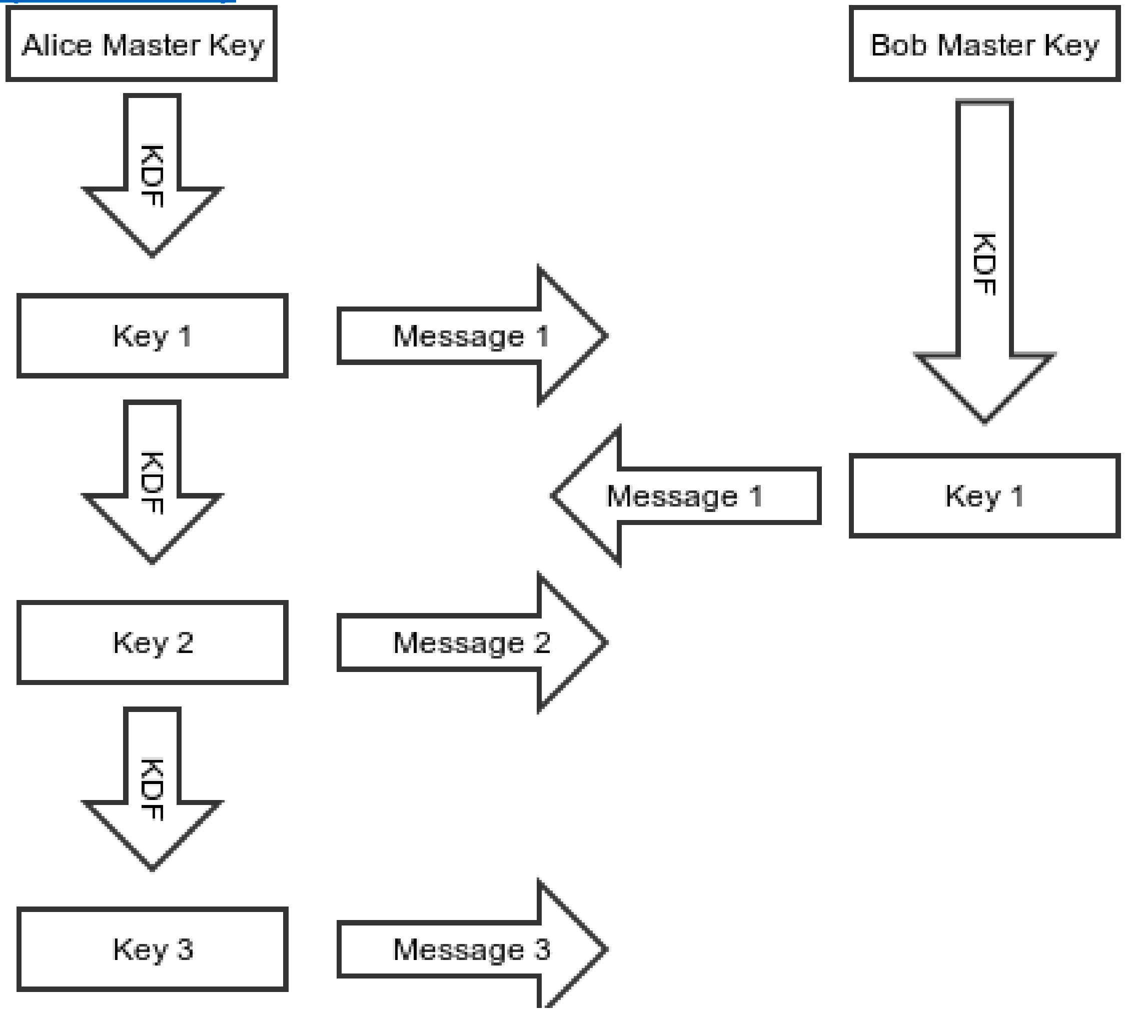
- Source: Blog Post Additional Material: Specification (technical!)
- Question: Perfect Forward Secrecy in Messaging
- Signal Protocol: “Double Ratchet”
  - Two ratchets that inspired it: OTR Ratchet



Gives a little forward secrecy &
good future secrecy (heals after compromise)

Question 1

- Source: Blog Post Additional Material: Specification (technical!)
- Question: Perfect Forward Secrecy in Messaging
- Signal Protocol: "Double Ratchet"
  - Two ratchets that inspired it: SCIMP
  - (by an earlier competitor called Silent Circle)

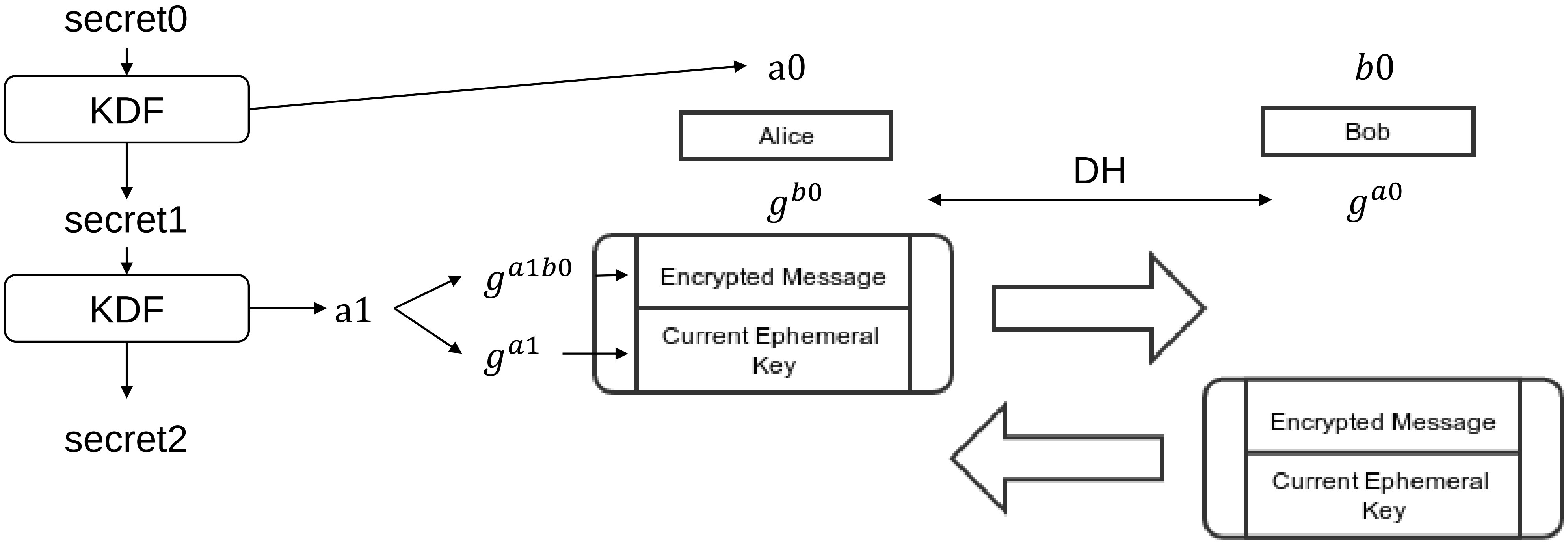


Gives perfect forward secrecy

Question 1

- Source: Blog Post Additional Material: Specification (technical!)
- Question: Perfect Forward Secrecy in Messaging
- Signal Protocol: “Double Ratchet”, massively simplified

Forward secrecy & future secrecy



Question 2

- Related Material: 03-04-TLS: slides 14, 51, https://ciphersuite.info/cs/
- Question: Cipher suites (TLS 1.2)
- Example: TLS_DHE_RSA_WITH_CHACHA20_POLY1305_SHA256
- DHE: Ephemeral Diffie-Hellmann
  - Key Exchange, ephemeral means that new DH-values must be used for each connection
- RSA: Rivest Shamir Adleman algorithm (well, usually RSA)
  - Authentication
- CHACHA20_POLY1305: AEAD cipher
  - Encryption & Integrity
- SHA256: SHA2 with 256bits
  - Hash (MAC if necessary)

Question 3

- Related Material: 03-04-TLS: slides 39 - 56
- Question: TLS 1.3 handshake

- Downgrading allows exploiting vulnerabilities in older versions of TLS/SSL
- What mechanisms are there?
  - Version negotiation
  - ClientFinished & ServerFinished

Question 4

- Related Material: 03-04-TLS: slides 39 - 56
- Question: Weak Randomness on nonces

- Nonce: Number only used ONCE!
  - What can happen if it is used multiple times?

Spoilers ahead!

Question 1

- Related Material: 03-04-TLS: slides 46
- Question: Perfect Forward Secrecy in Messaging

Signal	✓	Double Ratchet
WhatsApp	✓	Signal Protocol
Google Duo	✓	Signal Protocol
Skype	?	Reported to use Signal Protocol

Question 1

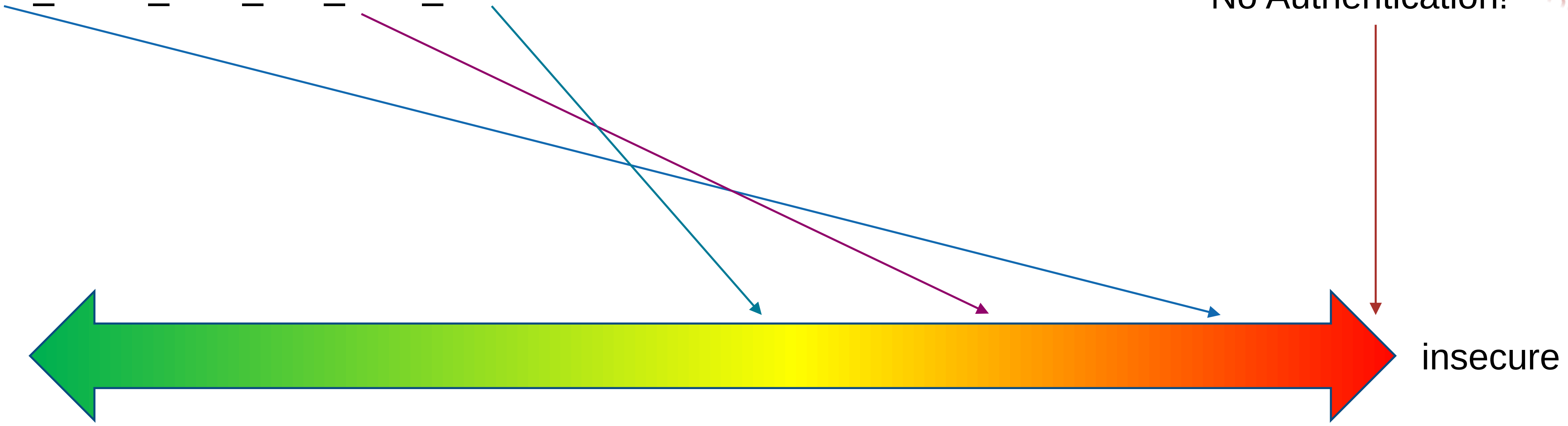
- Related Material: 03-04-TLS: slides 46
- Question: Perfect Forward Secrecy in Messaging

Telegram	🎲	Change keys every 100 messages
PGP – Pretty Good Privacy	✗	Basically RSA
WeChat	✗	Not even End-to-End-Encryption

Question 2

Qualitiative overview, check the exercise solution!

- Related Material: 03-04-TLS: slides 14, 51, https://ciphersuite.info/cs/
- Question: Cipher suites (TLS 1.2)
- TLS_DH_WITH_AES_256_CBC_SHA

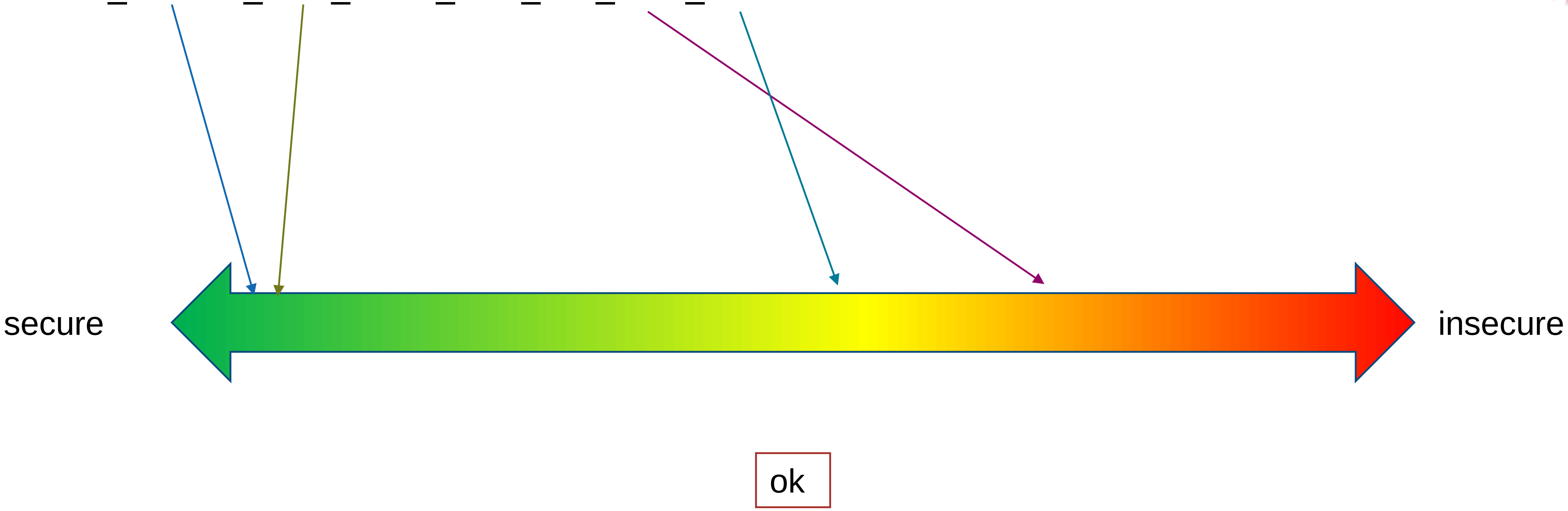


Don’t use!

Question 2

Qualitiative overview, check the exercise solution!

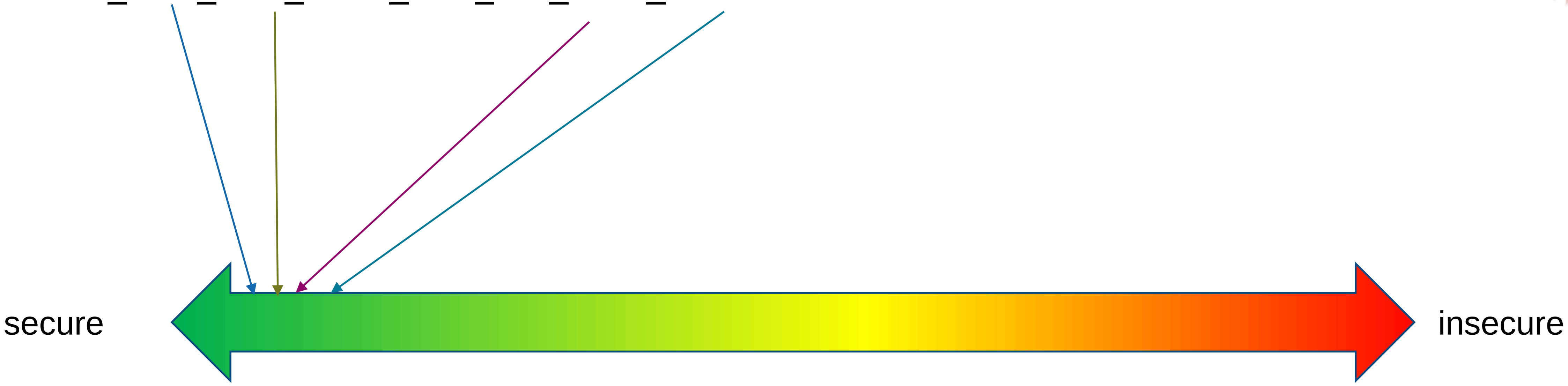
- Related Material: 03-04-TLS: slides 14, 51, https://ciphersuite.info/cs/
- Question: Cipher suites (TLS 1.2)
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA



Question 2

Qualitiative overview, check the exercise solution!

- Related Material: 03-04-TLS: slides 14, 51, https://ciphersuite.info/cs/
- Question: Cipher suites (TLS 1.2)
- TLS_DHE_RSA_WITH_AES_128_GCM_SHA256

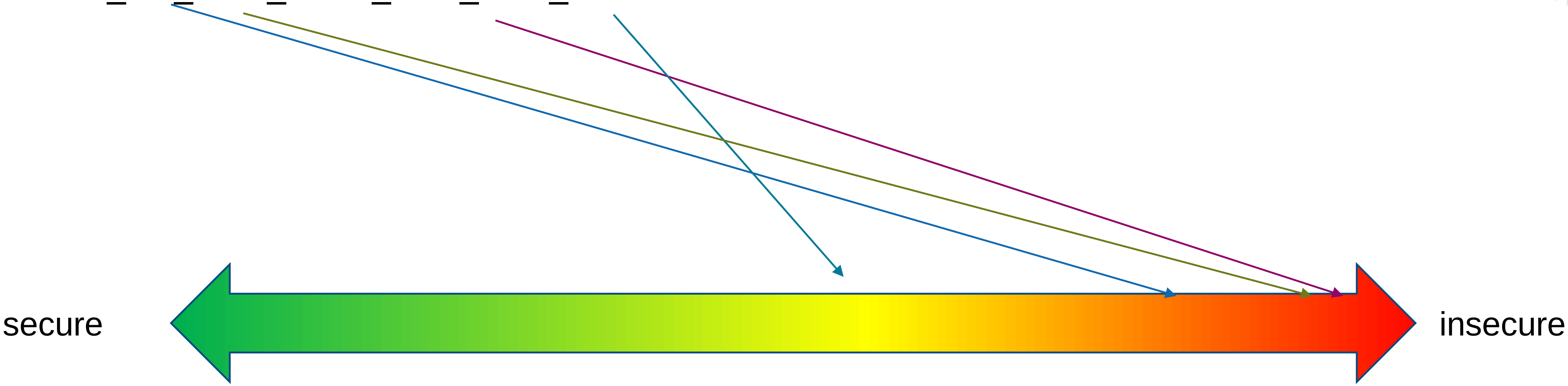


Safe

Question 2

Qualitiative overview, check the exercise solution!

- Related Material: 03-04-TLS: slides 14, 51, https://ciphersuite.info/cs/
- Question: Cipher suites (TLS 1.2)
- TLS_DH_anon_WITH_DES_CBC_SHA

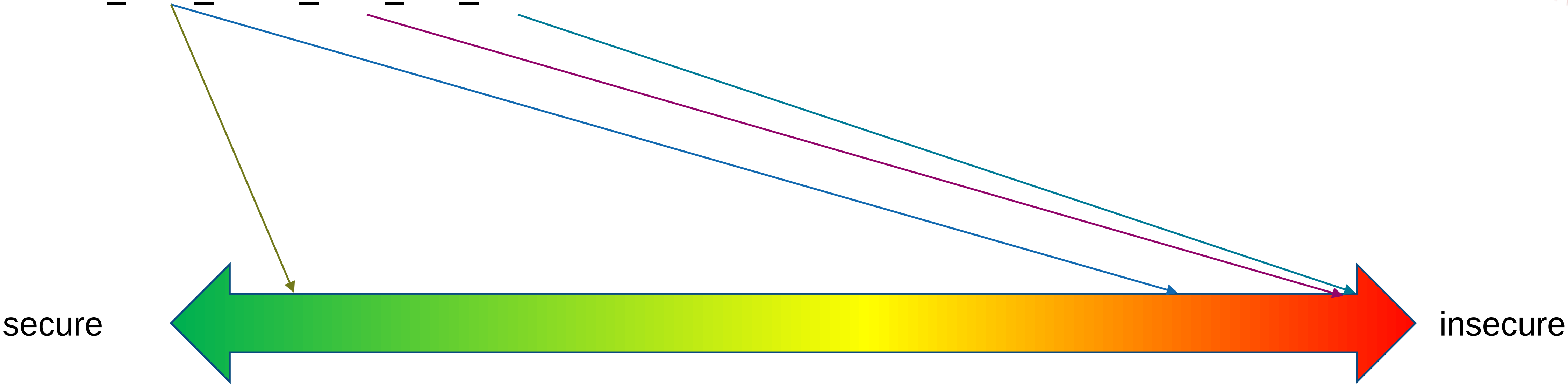


Don’t use!

Question 2

Qualitiative overview, check the exercise solution!

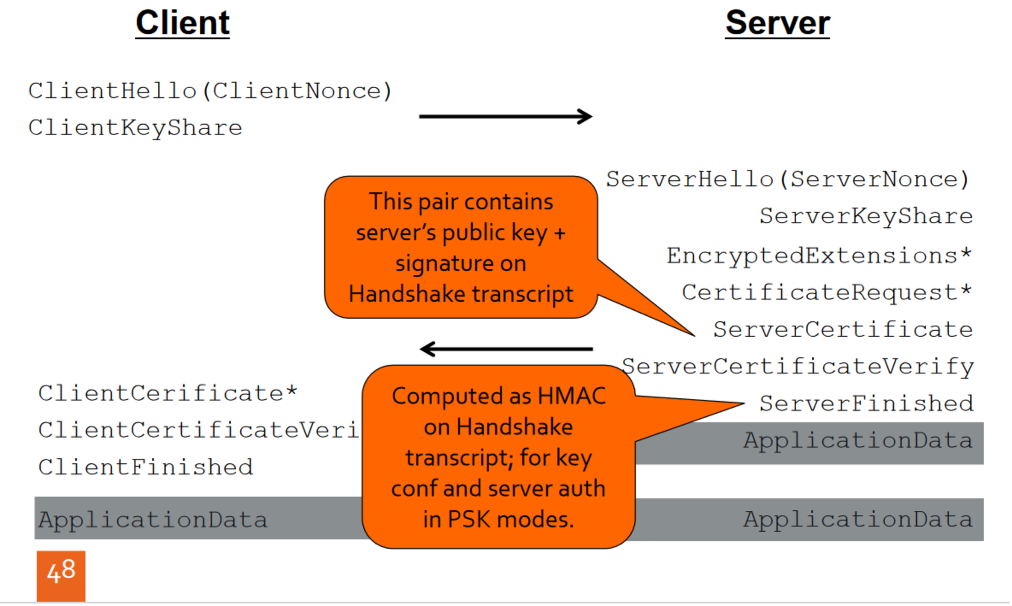
- Related Material: 03-04-TLS: slides 14, 51, https://ciphersuite.info/cs/
- Question: Cipher suites (TLS 1.2)
- TLS_RSA_WITH_RC4_128_MD5



Don’t use!

Question 3

- Related Material: 03-04-TLS: slides 39 - 56
- Question: TLS 1.3 handshake
- Can Drop or Modify Packets. Downgrade possible?
  - No, ServerFinished and ClientFinished guarantee integrity of handshake.
- Are both needed?
  - No, one is enough!
- Fallback mechanism: Restart with lower Version
  - Now downgrade is possible.
- Fundamentally why?
  - Server doesn’t know about previous attempts



This pair contains server’s public key + signature on Handshake transcript

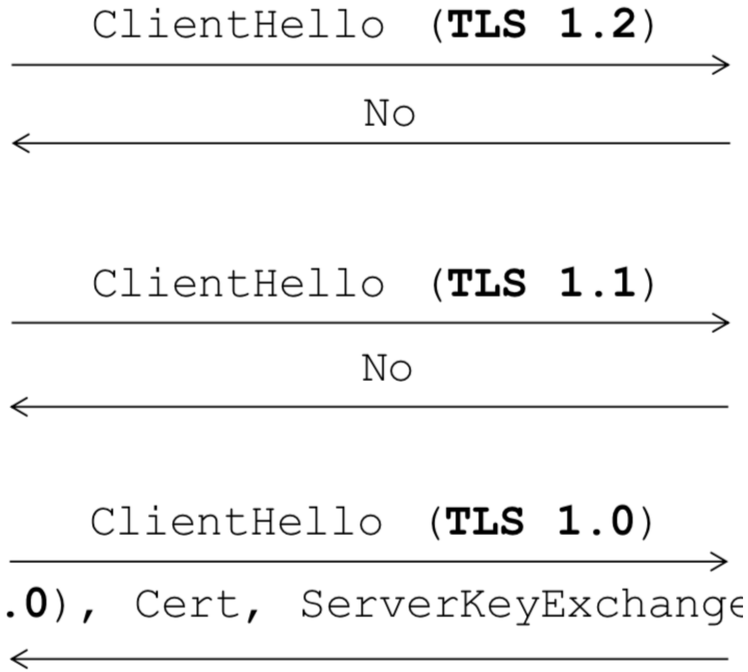
Computed as HMAC on Handshake transcript; for key conf and server auth in PSK modes.

Question 3

- Slides adapted from Tommaso Ciussani
- Question: TLS 1.3 handshake
- Better solution to have no downgrade attacks and support legacy servers?
- Attempt #1

- Include a **reason** in the cipher refusal messages of modern servers
- Attacker could just fake these refusal replies

- **Server is still not authenticated!**



ClientHello (**TLS 1.2**)

No

ClientHello (**TLS 1.1**)

No

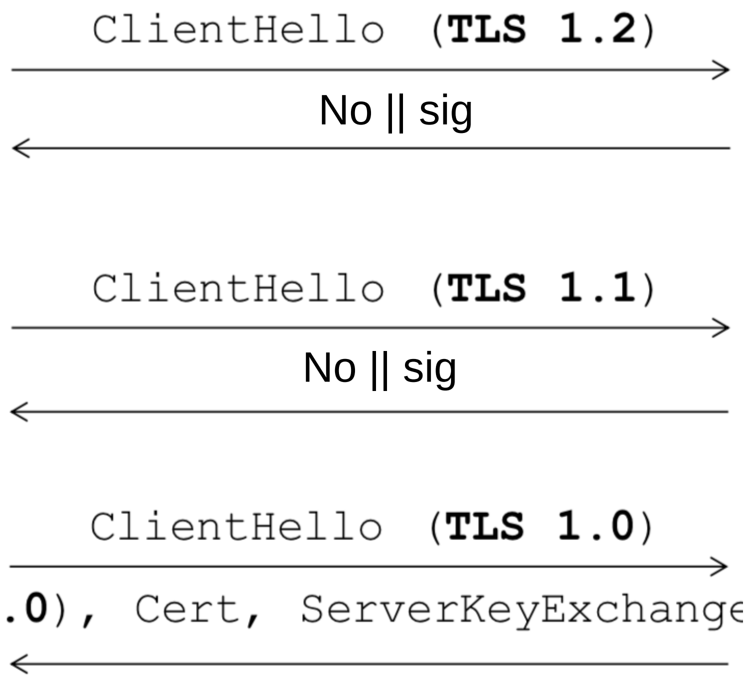
ClientHello (**TLS 1.0**)

.0), Cert, ServerKeyExchange

Question 3

- Slides adapted from Tommaso Ciussani
- Question: TLS 1.3 handshake
- Better solution to have no downgrade attacks and support legacy servers?
- Attempt #2

- Server **signs negative messages**
- Works partially: the **client needs to support this.**
- Old clients will not work with modern servers
- Fix by **sending a flag**
- **Why not just sending the flag?**



ClientHello (**TLS 1.2**)

No || sig

ClientHello (**TLS 1.1**)

No || sig

ClientHello (**TLS 1.0**)

.0), Cert, ServerKeyExchange

Question 3

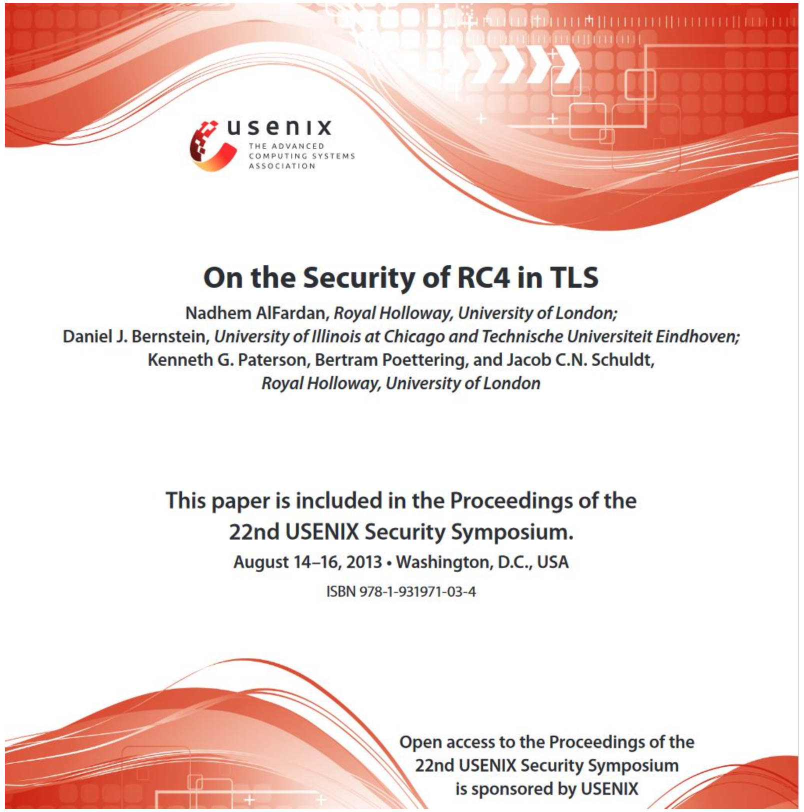
- Slides adapted from Tommaso Ciussani
- Question: TLS 1.3 handshake
- Better solution to have no downgrade attacks and support legacy servers?
- The IETF solution: RFC 7507

TLS_FALLBACK_SCSV

- New TLS cipher suite pseudo-value
- Signaling Cipher Suite Value
- **Not an actual suite**
- Sent by the client to notify the server of previous connection attempts

Question 3

- Related Material: 03-04-TLS: slide 7
- Question: TLS 1.3 handshake
- Why is downgrade to SSLv3 especially bad?
  - Considered broken.
  - Block ciphers: Padding Oracle Attack
  - Stream cipher (RC4): Statistical attack



usenix
THE ADVANCED COMPUTING SYSTEMS ASSOCIATION

On the Security of RC4 in TLS

Nadhem AlFardan, *Royal Holloway, University of London;*
Daniel J. Bernstein, *University of Illinois at Chicago and Technische Universiteit Eindhoven;*
Kenneth G. Paterson, Bertram Poettering, and Jacob C.N. Schuldt,
Royal Holloway, University of London

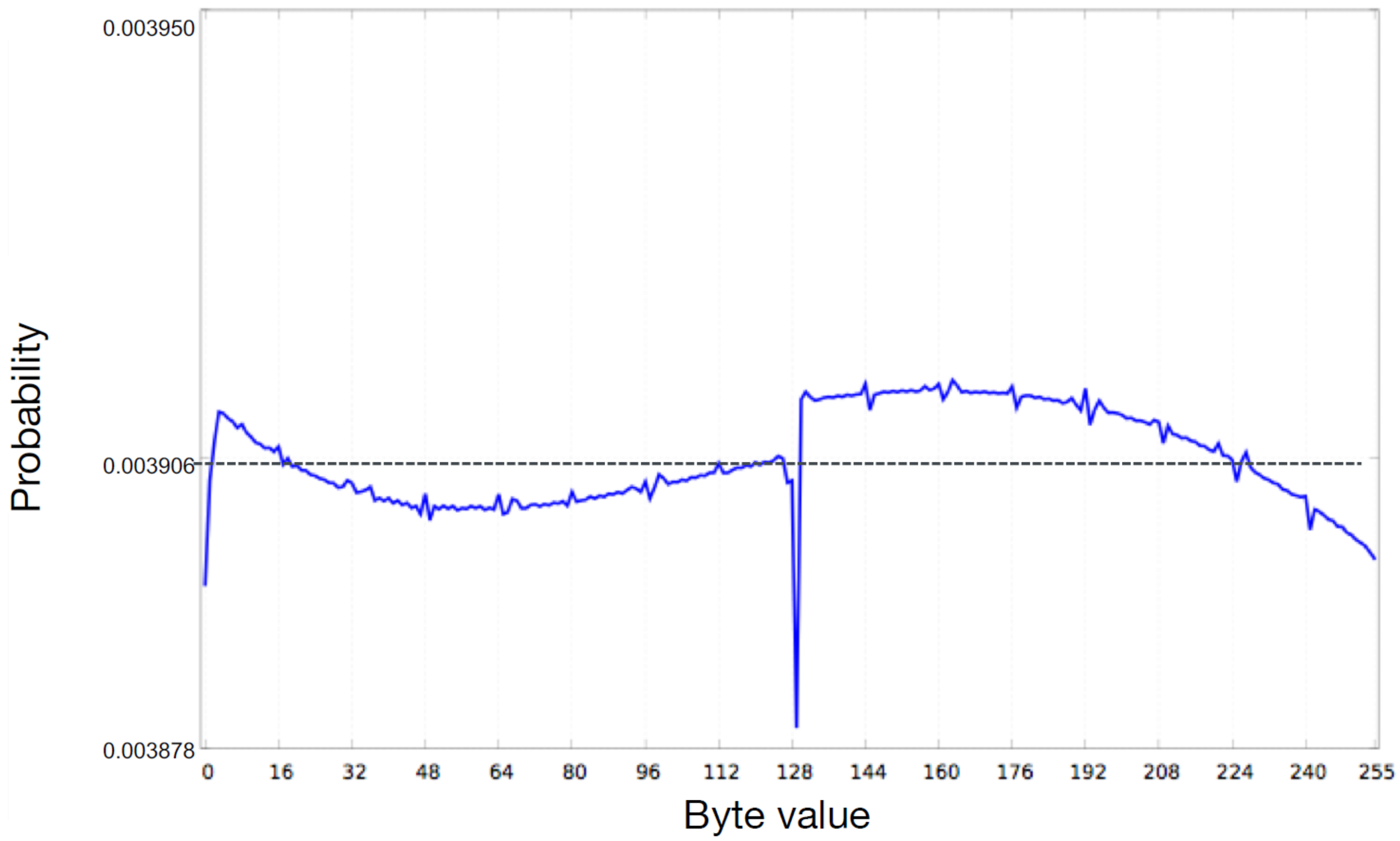
This paper is included in the Proceedings of the
22nd USENIX Security Symposium.
August 14–16, 2013 • Washington, D.C., USA
ISBN 978-1-931971-03-4

Open access to the Proceedings of the
22nd USENIX Security Symposium
is sponsored by USENIX

RC4 Paper

Keystream Distribution at Position 1

[of RC4]

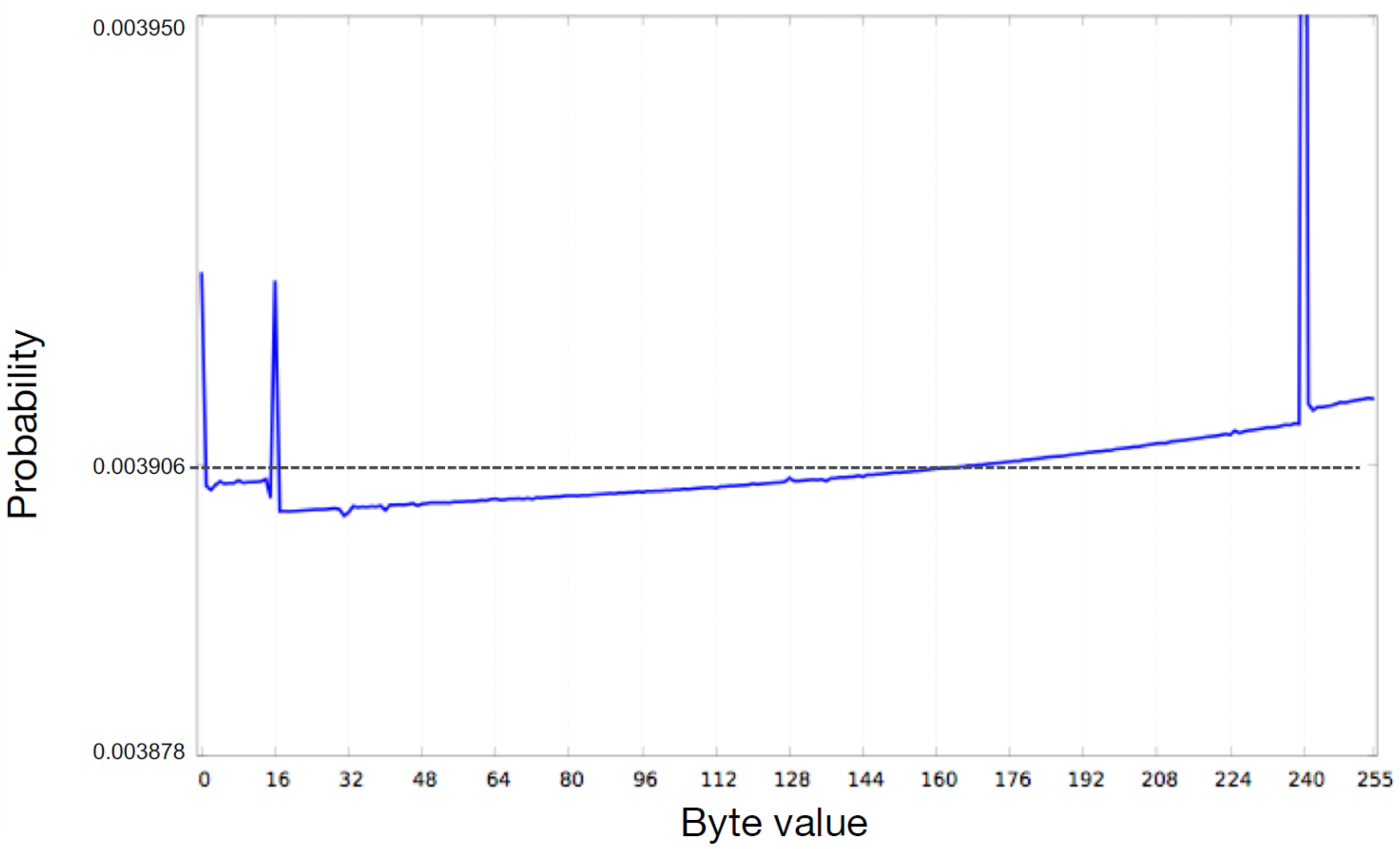


59

NetSec 2019, Prof. Paterson

Keystream Distribution at Position 16

[of RC4]



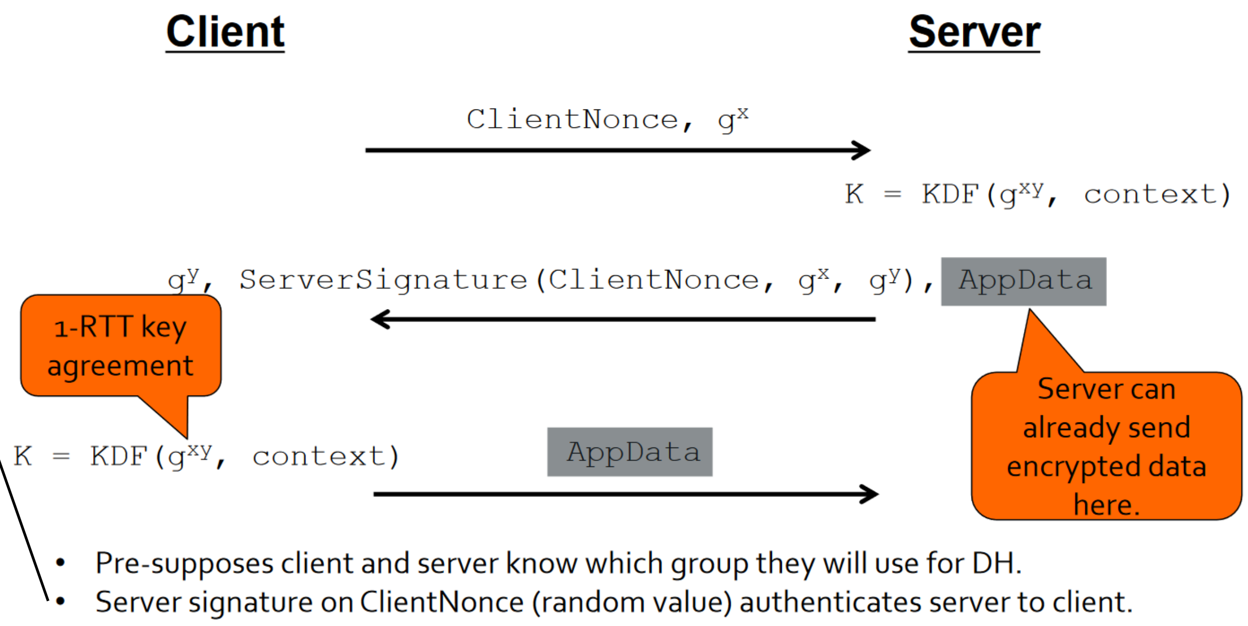
60

NetSec 2019, Prof. Paterson

Question 4

- Related Material: 03-04-TLS: slide 41 & 42, 59 – 66
- Question: Weak Randomness on nonces
- Note: The PRNG is only used for nonce generation. (not key generation)

- “Server signature on ClientNonce (random value) authenticates the server to client.”
- Replay attack?
  - No, ephemeral DH values!


Client
Server
ClientNonce, g^x
$K = KDF(g^{xy}, context)$
g^y, ServerSignature(ClientNonce, g^x, g^y), AppData
$K = KDF(g^{xy}, context)$
AppData

1-RTT key agreement

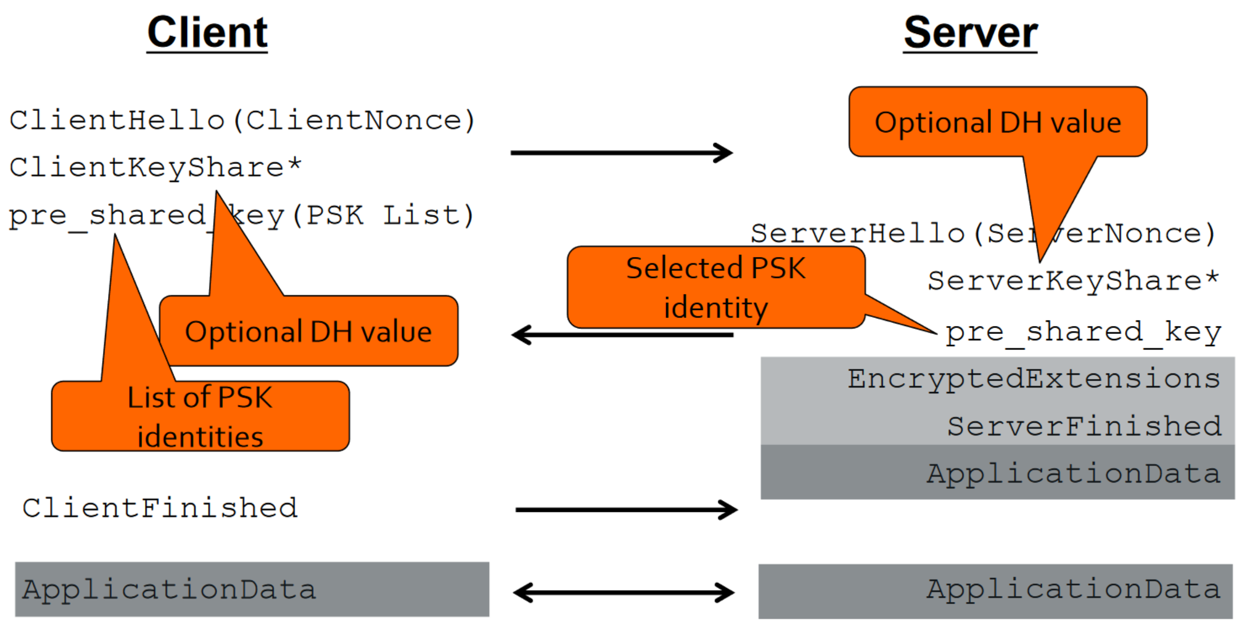
Server can already send encrypted data here.

- Pre-supposes client and server know which group they will use for DH.
- Server signature on ClientNonce (random value) authenticates server to client.

Question 4

- Related Material: 03-04-TLS: slide 41 & 42, 59 – 66
- Question: Weak Randomness on nonces
- Note: The PRNG is only used for nonce generation. (not key generation)

- “Server signature on ClientNonce (random value) authenticates the server to client.”
- Replay attack?
  - Potentially, optional DH values!



Your Questions