RUHR-UNIVERSITÄT BOCHUM

RUB

Revisiting SSL/TLS Implementations

New Bleichenbacher Side Channels and Attacks

***Christopher Meyer*, Juraj Somorovsky, Eugen Weiss, Jörg Schwenk, Sebastian Schinzel, Erik Tews**

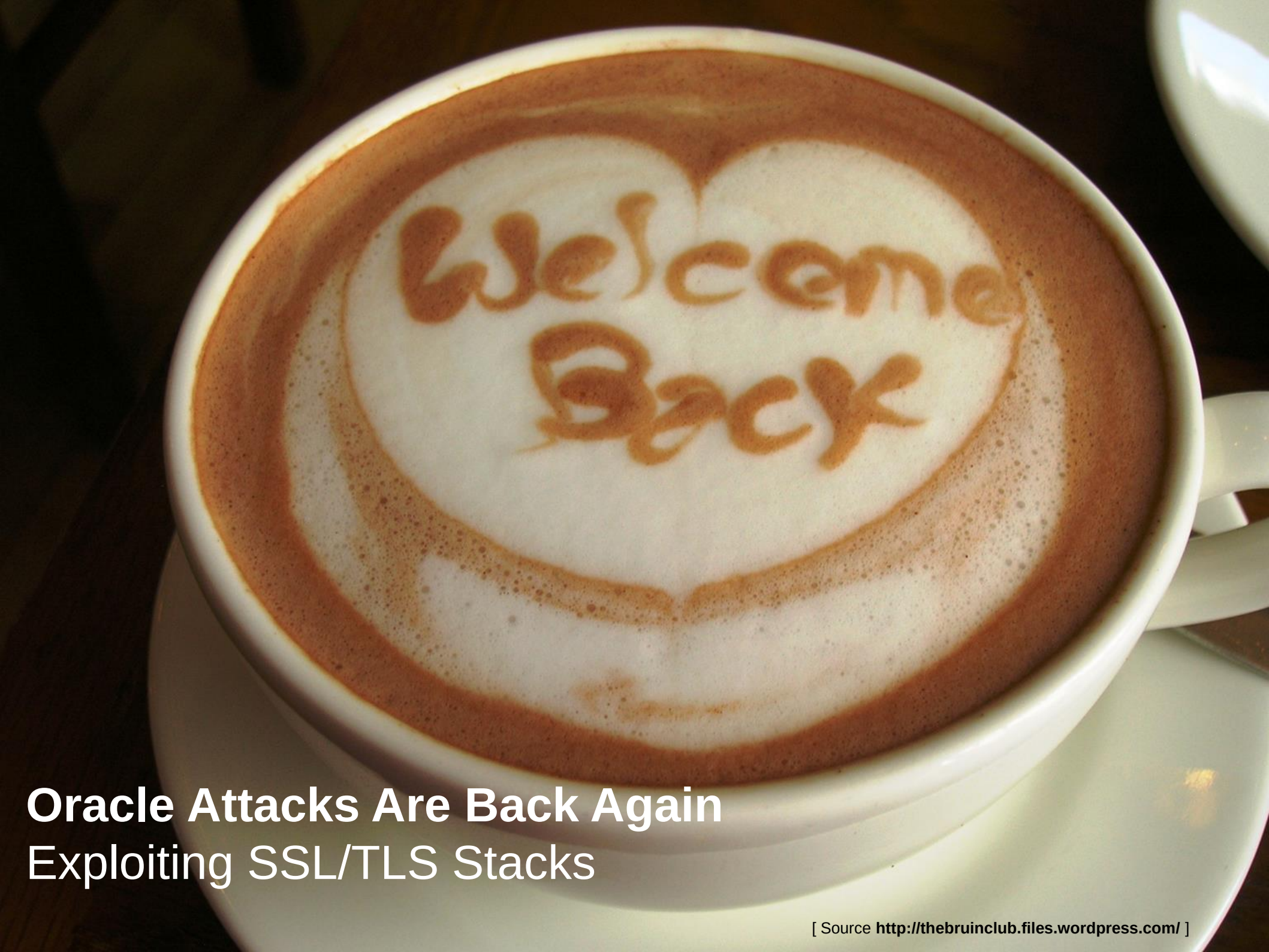


[Source **http://www.cafe-reichard.de**]


TLS
… is Like Intelligent Glas
[Source **http://www.cafe-reichard.de**]



[Source **http://www.cafe-reichard.de**]



Oracle Attacks Are Back Again

Exploiting SSL/TLS Stacks

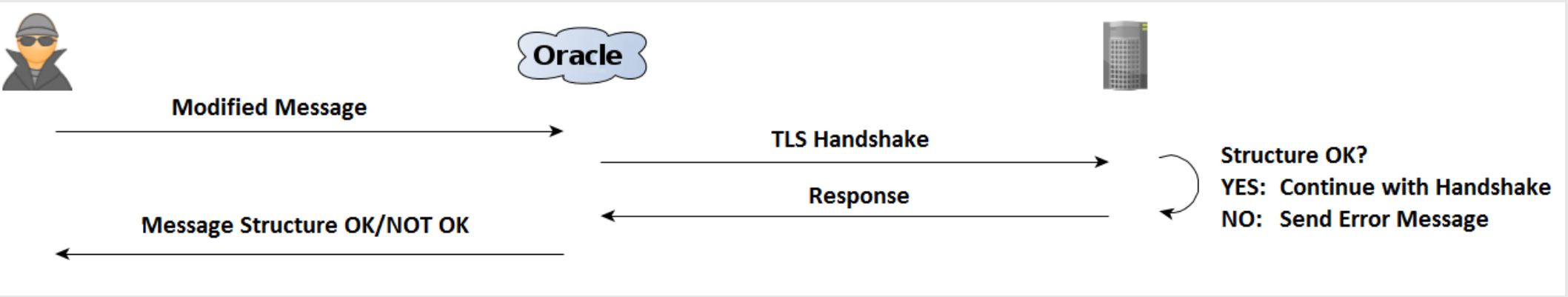
[Source **http://thebruinclub.files.wordpress.com/**]

Bleichenbacher's Attack
Restoring the Key Material

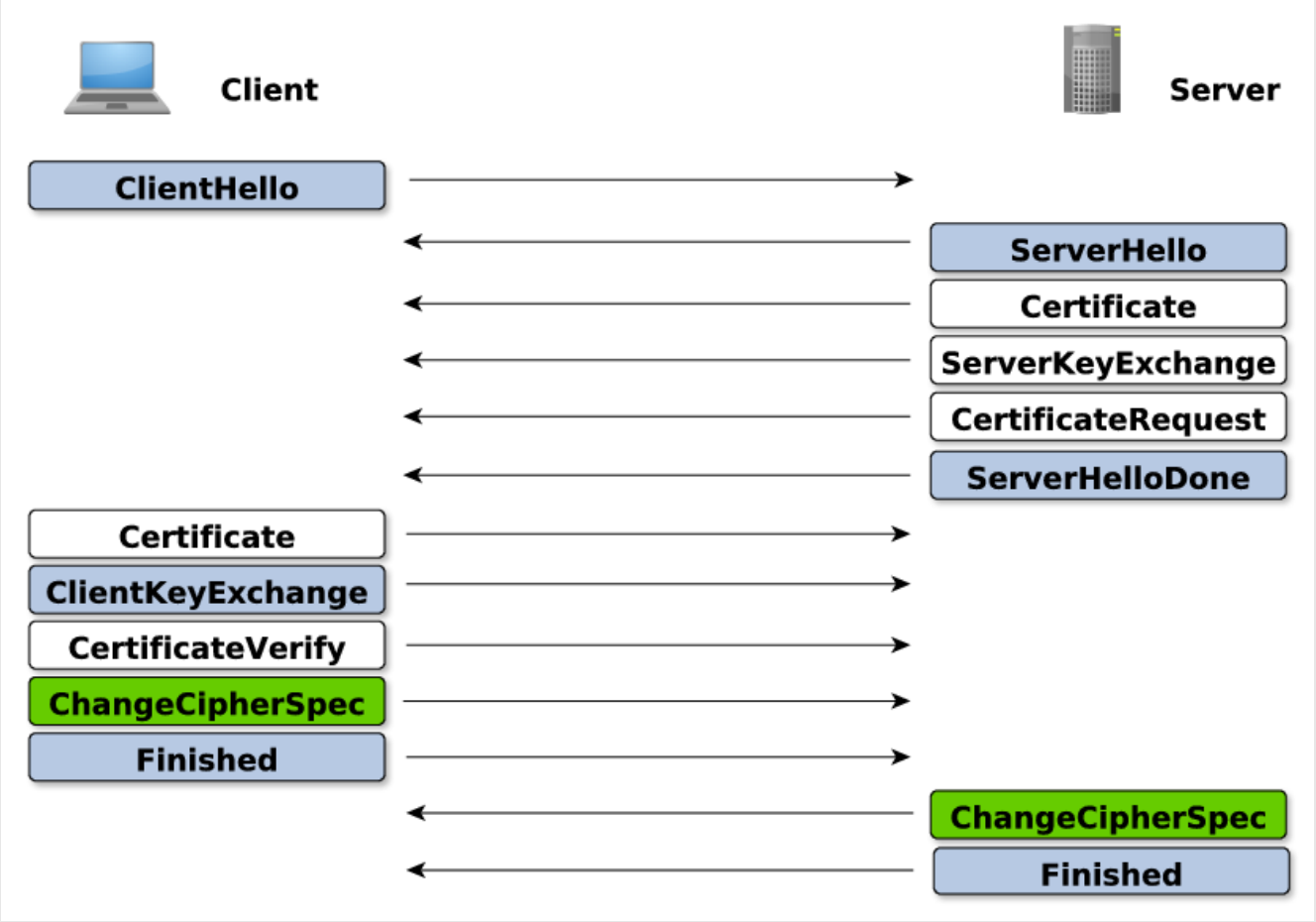
Bleichenbacher's Attack
Restoring the Key Material

- Preconditions
  - RSA-Based Key Exchange
  - Previously Recorded SSL/TLS Session
  - Connection to Target Server Possible

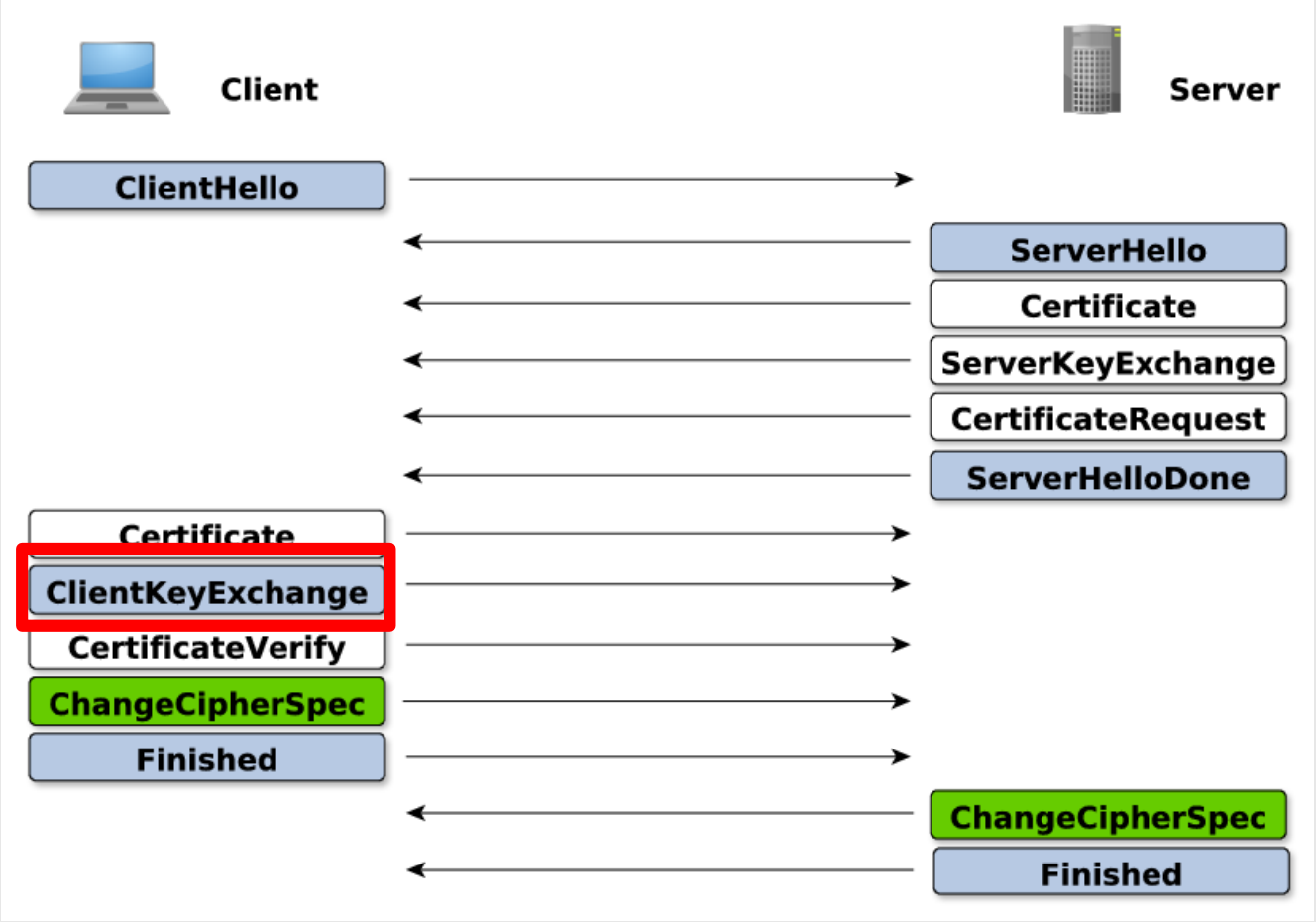
Bleichenbacher's Attack
Restoring the Key Material



Handshake Protocol
Recap

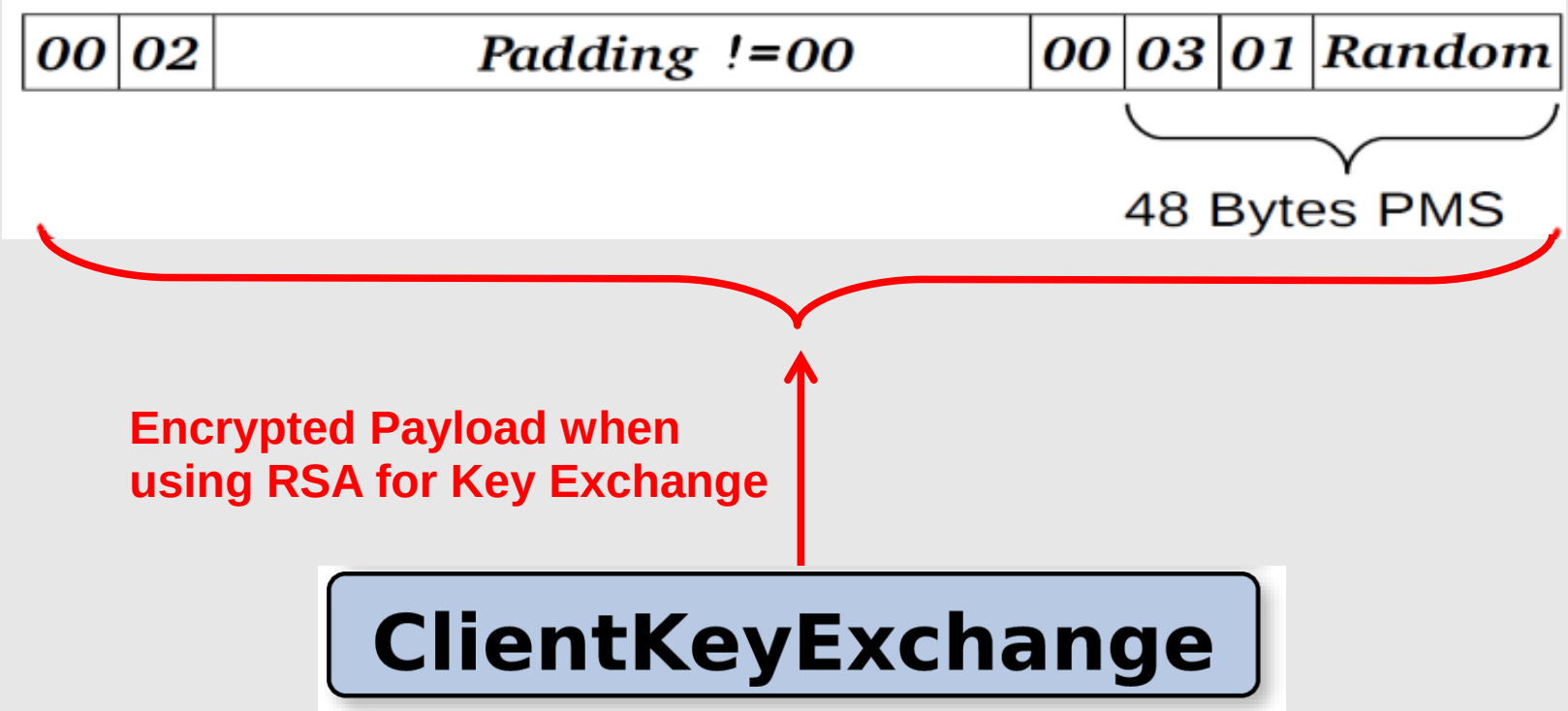


Handshake Protocol
Recap



Handshake Protocol

Content of a *ClientKeyExchange* Message (RSA Case)



PMS = PreMasterSecret

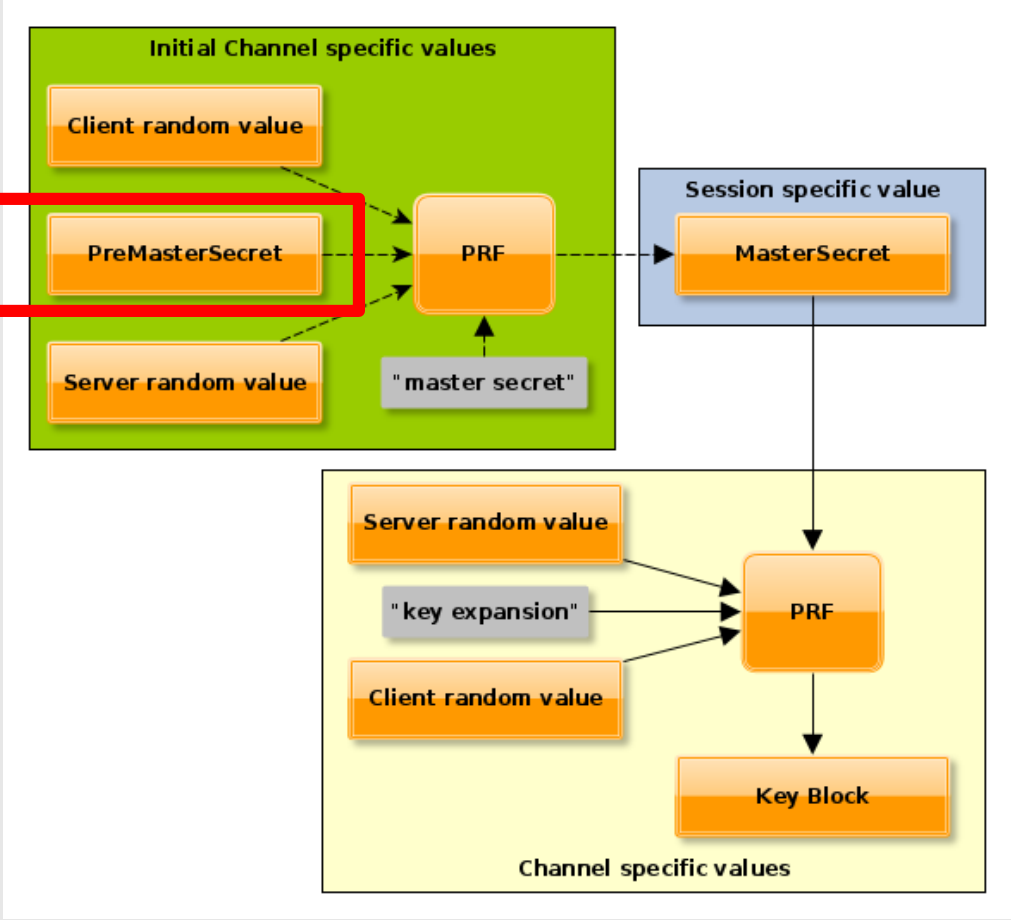
Key Derivation

Why is the PMS so Sensitive?

Known

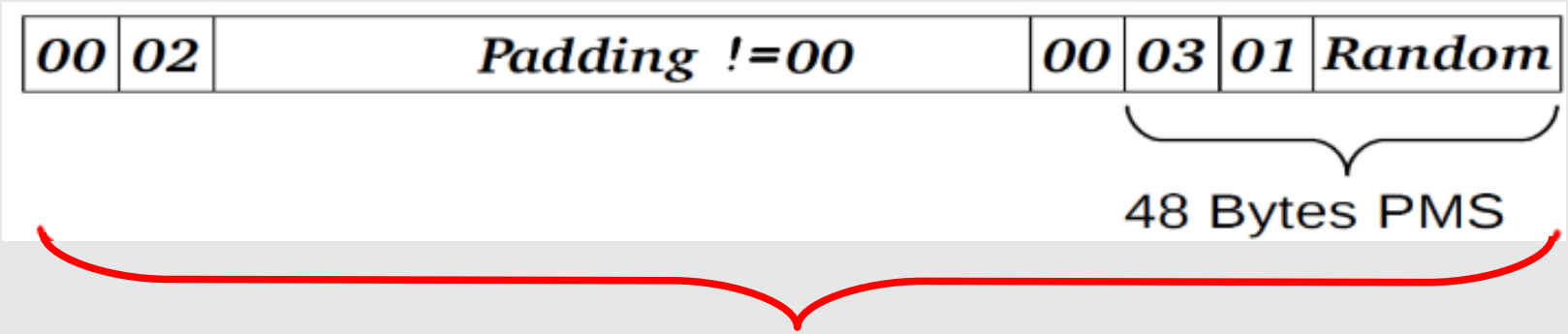
Unknown ⟶

Known



Handshake Protocol

Content of a *ClientKeyExchange* Message (RSA Case)

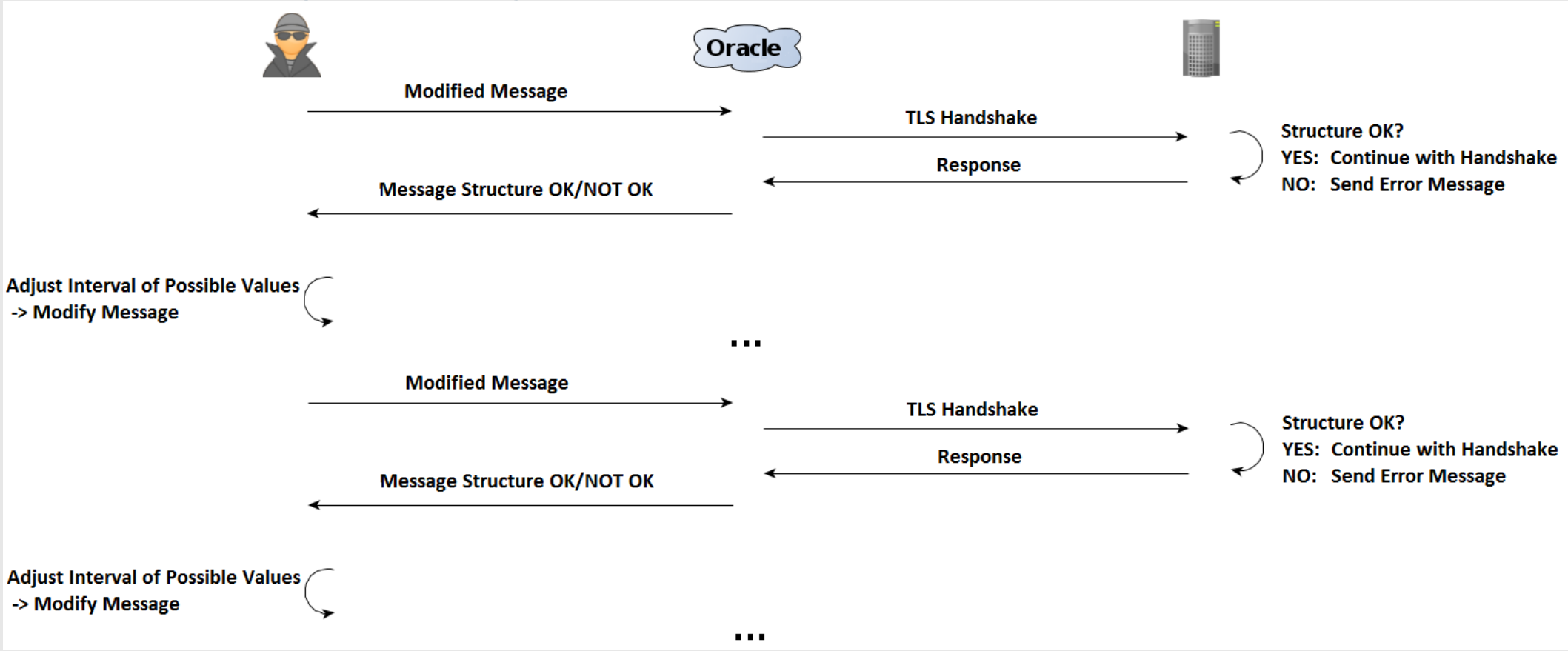


Encrypted Payload when using RSA for Key Exchange

ClientKeyExchange

Structure Violations Cause Different Response Messages!

Bleichenbacher's Attack
Restoring the Key Material



Bleichenbacher's Attack
Adjusting the ClientKeyExchange Payload

Bleichenbacher's Attack
Adjusting the ClientKeyExchange Payload

- RSA Homomorphic-Property

$$enc_{n,e}(m_1) * enc_{n,e}(m_2)$$

$$= enc_{n,e}(m_1 * m_2)$$

Bleichenbacher's Attack
Adjusting the ClientKeyExchange Payload

- RSA Homomorphic-Property

$$enc_{n,e}(m_1) * enc_{n,e}(m_2)$$

$$= enc_{n,e}(m_1 * m_2)$$

- Explanation:

$$enc_{n,e}(m_1) * enc_{n,e}(m_2)$$

$$\equiv m_1^e * m_2^e \bmod n \equiv (m_1 * m_2)^e \bmod n$$

$$= enc_{n,e}(m_1 * m_2)$$

Bleichenbacher's Attack
Adjusting the ClientKeyExchange Payload

- s: Value Chosen by Attacker
- c: Eavesdropped, Encrypted Content of a ClientKeyExchange Msg
- m: (Unknown) Plainext of c

- $$enc_{n,e}(s) * c$$
  $$\equiv s^e * c \bmod n$$
  $$\equiv (s * m)^e \bmod n$$
  $$= enc_{n,e}(s * m)$$

Bleichenbacher's Attack
Restoring the Key Material



TLS 1.0 Defines Countermeasures

Bleichenbacher's Attack
Restoring the Key Material



TLS 1.0 Defines Countermeasures

Create a random PMS_R if there's anything wrong with the one we received and continue with PMS_R

Bleichenbacher's Attack
New Side-Channels

Bleichenbacher's Attack
New Side-Channels

- Force Different Error Messages

Bleichenbacher's Attack
New Side-Channels

- Force Different Error Messages
  - Implementation Bug



Bleichenbacher's Attack
New Side-Channels

- Force Different Error Messages
  - Implementation Bug




```
1 for (int i = 0; i < secLen; i++) {
2             pad1[i] ^= secret[i + secOff];
3             pad2[i] ^= secret[i + secOff];
4 }
```

Listing 8.1: Internal loop causing an `Exception` in special cases – *Source: com.sun.crypto.provider.TlsPrfGenerator*

Bleichenbacher's Attack
New Side-Channels

- Force Different Error Messages
  - Implementation Bug

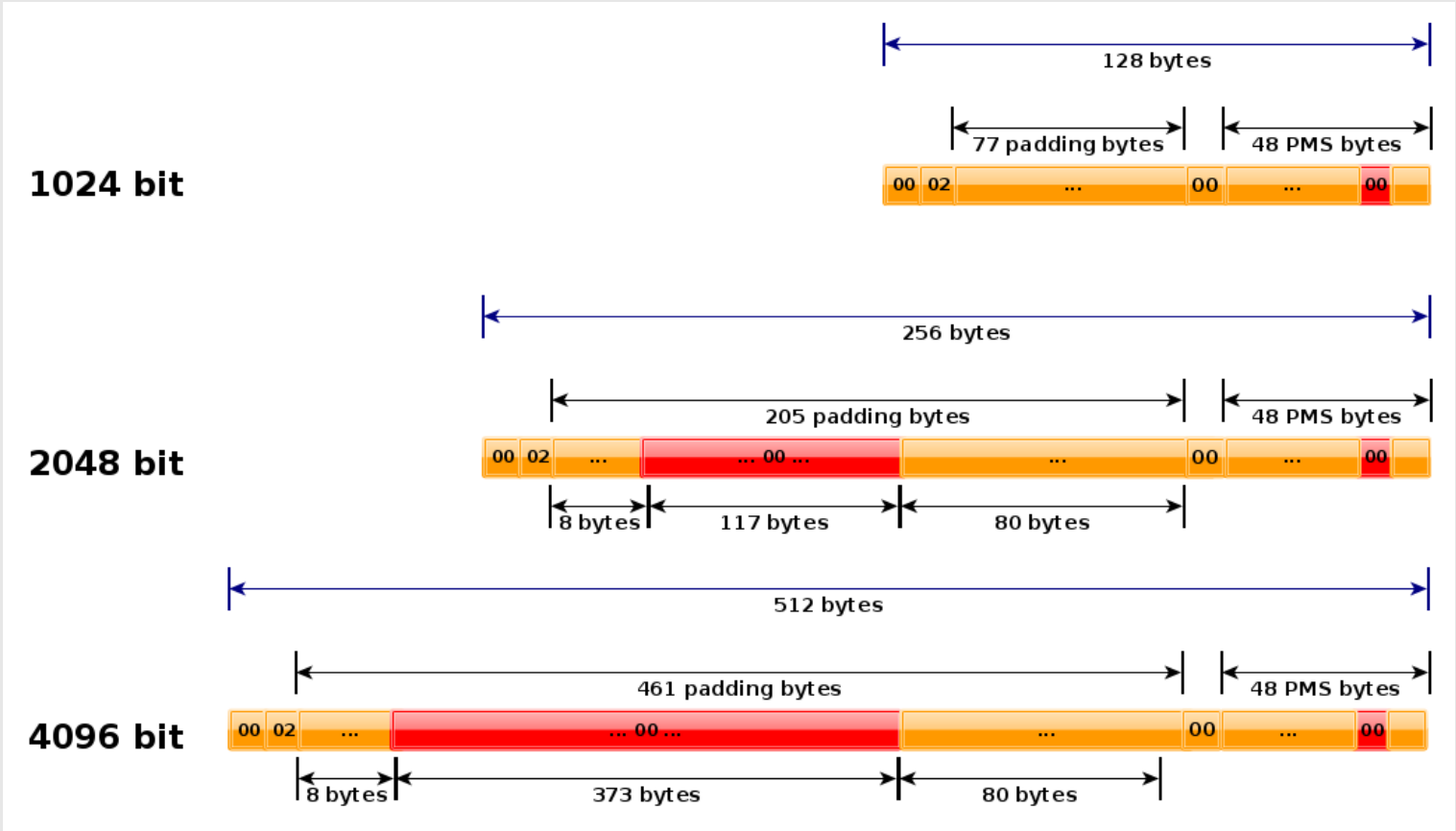



```
1 for (int i = 0; i < secLen; i++) {
2             pad1[i] ^= secret[i + secOff];
3             pad2[i] ^= secret[i + secOff];
4 }
```

Listing 8.1: Internal loop causing an `Exception` in special cases – *Source: com.sun.crypto.provider.TlsPrfGenerator*

`ArrayIndexOutOfBoundsException` causes an `INTERNAL_ERROR` alert instead of `HANDSHAKE_FAILURE`

Bleichenbacher's Attack
JSSE – Different Error Messages



Root Cause of Bug: `com.sun.crypto.provider.TlsPrfGenerator.expand(..)`

Real-World Bleichenbacher

~ 12 hours and ~177k queries later....

Java Total Breach of RSA-based TLS
Internal Flaws Resurrected Bleichenbacher's Attack

```
 INFO [main] 26 Sep 2012 19:35:50,388  - Step 2c: Searching with one interval left
 INFO [main] 26 Sep 2012 19:35:50,726  -  Found s2015:
2353474280691358986415452724171276939586296506243294697426729369327166239800462513683715910529286402005811714103895479929288033060730341292865411
 INFO [main] 26 Sep 2012 19:35:50,726  - Step 3: Narrowing the set of soultions.
 INFO [main] 26 Sep 2012 19:35:50,726  -  # of intervals for M2015: 1
 INFO [main] 26 Sep 2012 19:35:50,726  - Step 4: Computing the solution.
 INFO [main] 26 Sep 2012 19:35:50,726  - // Total # of queries so far: 456355
 INFO [main] 26 Sep 2012 19:35:50,727  - Step 2: Searching for PKCS conforming messages.
 INFO [main] 26 Sep 2012 19:35:50,727  - Step 2c: Searching with one interval left
 INFO [main] 26 Sep 2012 19:35:51,305  -  Found s2016:
4706948561382717972830905448342553879172593012486589394853458738654332479600925027367431821058572804011623428207790959858576066121460682585730823
 INFO [main] 26 Sep 2012 19:35:51,305  - Step 3: Narrowing the set of soultions.
 INFO [main] 26 Sep 2012 19:35:51,306  -  # of intervals for M2016: 1
 INFO [main] 26 Sep 2012 19:35:51,306  - Step 4: Computing the solution.
 INFO [main] 26 Sep 2012 19:35:51,306  - ====> Solution found!
00 02 f5 a7 9f cd b1 27 f9 39 15 21 49 71 65 97 33 99 6d 9b cd 6d 4b e3 f5 fd b5 71 d5 69 71 91 b9 39 c9 6d f5 59 f1 b9 97 b7 6b ff 33 d1 9b 85 13 d5
39 1b 00 03 01 06 26 a6 40 57 4b 50 d6 a3 d0 8a 70 16 0a 0d af 33 2a 7f 9b c8 65 a7 b5 54 e7 48 9f 57 da c9 bf 34 8b 8d d4 84 ed c9 63 2b 16 6f 2c
 INFO [main] 26 Sep 2012 19:35:51,306  - // Total # of queries so far: 456370
```

Pre Master Secret

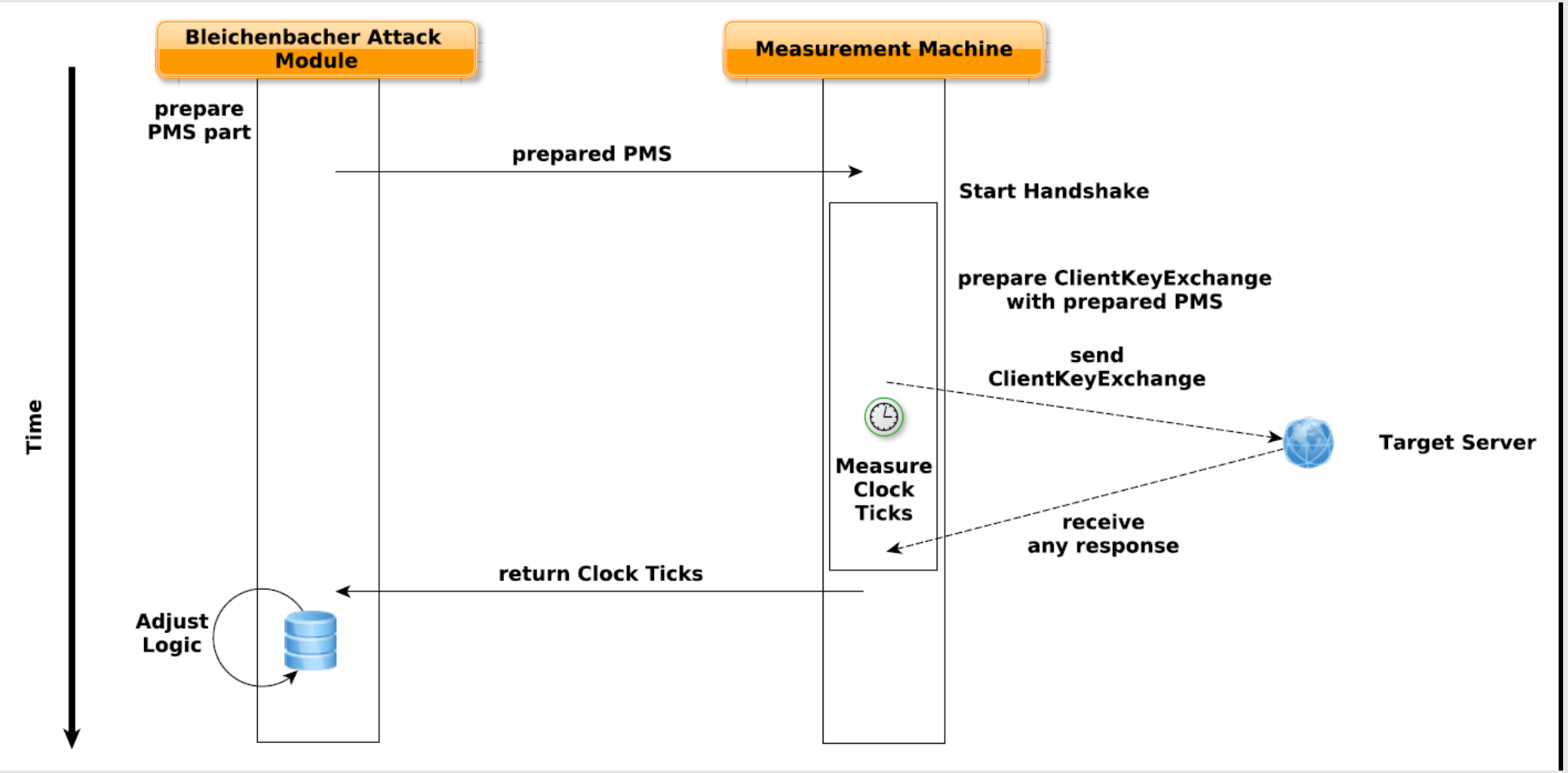
03 01 is protocol version of TLS 1.0

Bleichenbacher's Attack
New Side-Channels

- Force Different Error Messages
  - Implementation Bug
- Force Different Processing Time



Measurement Setup
How to Measure Different Processing Times



Bleichenbacher's Attack
New Side-Channels

- Force Different Error Messages
  - Implementation Bug
- Force Different Processing Time
  - Random Number Generation





Bleichenbacher's Attack
New Side-Channels

- Force Different Error Messages
  - Implementation Bug
- Force Different Processing Time
  - Random Number Generation





Only **create a random PMS_R if there's anything wrong with the received one and continue with PMS_R**

VS.

Always **create a random PMS_R, but only continue with PMS_R if there's anything wrong with the received one**

Bleichenbacher's Attack
OpenSSL – Different Processing Times

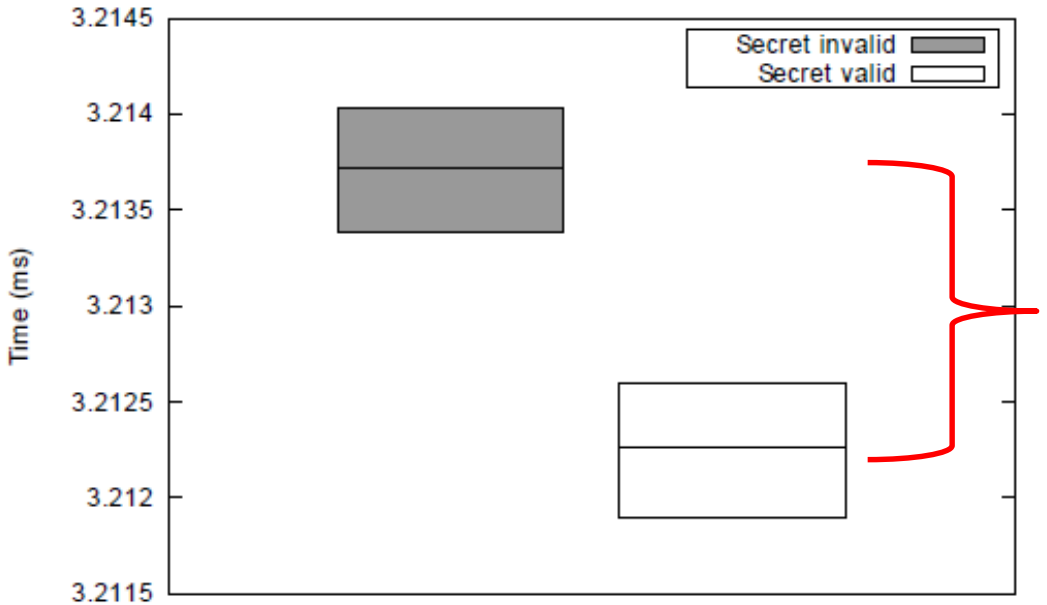


Figure 8.4.: Timing measurement results for OpenSSL 0.98. The valid secret refers to a TLS compliant ciphertext with correct *PreMasterSecret*. The invalid secret refers to a non-PKCS#1 v1.5 compliant ciphertext. In the non-PKCS#1 v1.5 compliant structure the first byte (which should be 00) was altered to 08 to provoke a random number generation on server-side.

~ 1,5 -2 µs

Bleichenbacher's Attack
New Side-Channels

- Force Different Error Messages
  - Implementation Bug
- Force Different Processing Time
  - Random Number Generation

No attack possible (so far) timing differences very small and oracle due to strict PKCS#1 checks very weak

Bleichenbacher's Attack
New Side-Channels

- Force Different Error Messages
  - Implementation Bug
- Force Different Processing Time
  - Random Number Generation
  - `Exceptions`







Bleichenbacher's Attack

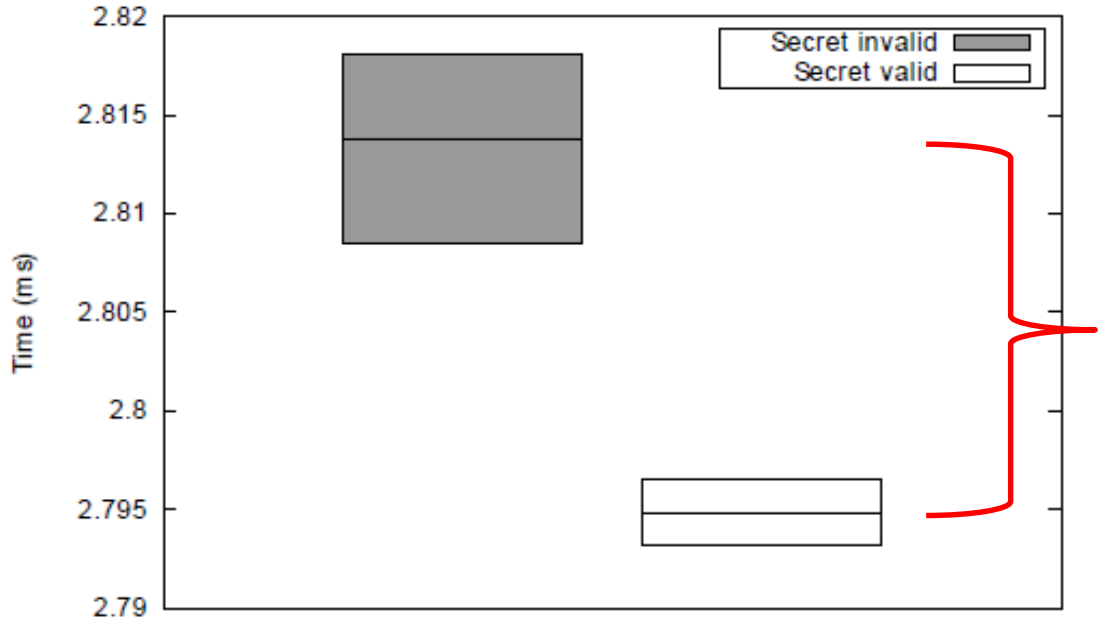
JSSE – `Exceptions` Cause Different Processing Time

- Unpadding Function Used to Remove PKCS#1 Overhead

```
private byte [] unpadV15 (byte[] padded) throws
BadPaddingException {
    if (not PKCS compliant) {
        throw new BadPaddingException();
    } else {
        return unpadded text;
    }
}
```

Bleichenbacher's Attack

JSSE – `Exceptions` Cause Different Processing Time



~ 20 µs

Figure 8.5.: Timing measurement results for Java 1.7 (JSSE). The valid secret refers to a PKCS#1 compliant ciphertext. The invalid secret refers to a non-PKCS#1 compliant ciphertext. In the non-PKCS#1 compliant structure the first byte (which should be 0x00) was altered to 0x08 to provoke an `BadPaddingException` on the server.

Bleichenbacher's Attack

JSSE – `Exceptions` Cause Different Processing Time

- Optimized Algorithm (Bardou et al.)
- 2048 bit
- Strong Oracle (no TLS Version Check)
- Optimized PKCS#1 v1.5 Structure to Speed up Algorithm

~ 19,5 hours and ~18,6k queries

Bleichenbacher's Attack
New Side-Channels

- Force Different Error Messages
  - Implementation Bug
- Force Different Processing Time
  - Random Number Generation
  - `Exceptions`
  - Special Behaviour



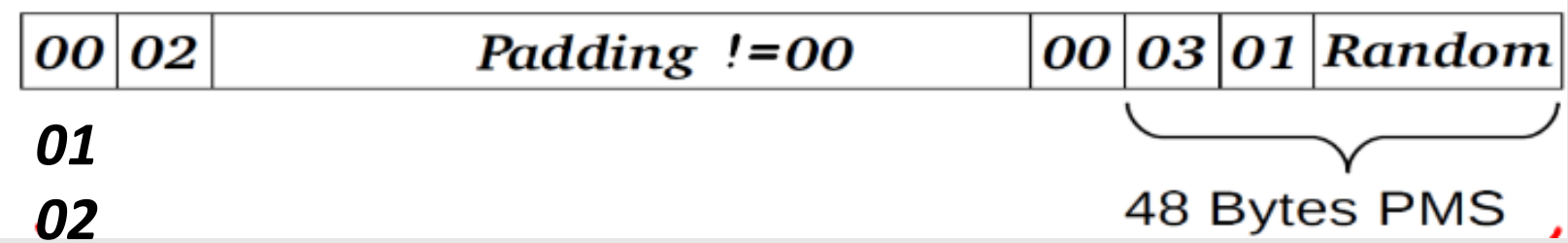






Bleichenbacher's Attack
Cavium – Misbehaving Hardware Accelerators

- Hardware did not check the first byte's value



Bleichenbacher's Attack

Cavium – Misbehaving Hardware Accelerators

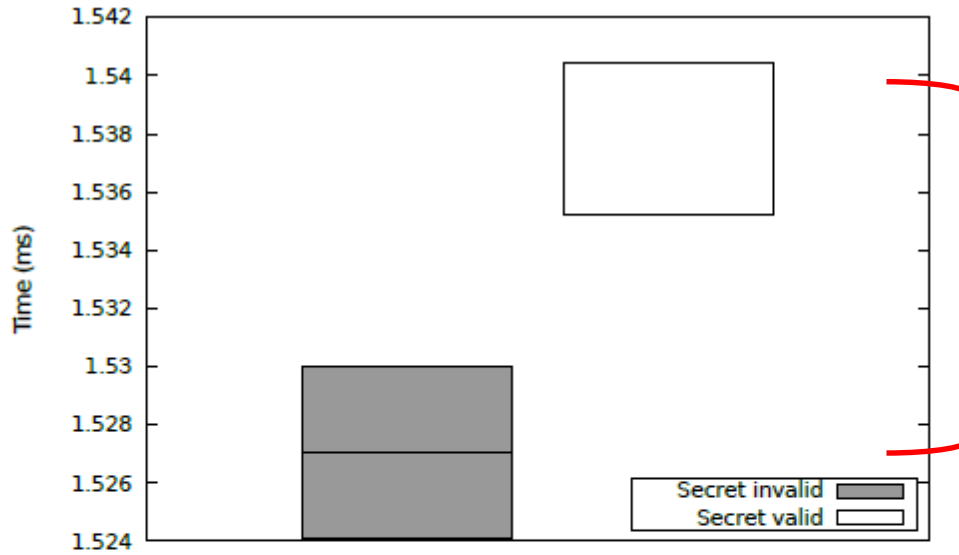


Figure 11: Timing measurement results for our *IBM Datapower*. The valid secret refers to a message, which starts with 0x??02, where 0x?? indicates an arbitrary byte. The invalid secret refers to a message starting with different bytes.

~ 15 µs

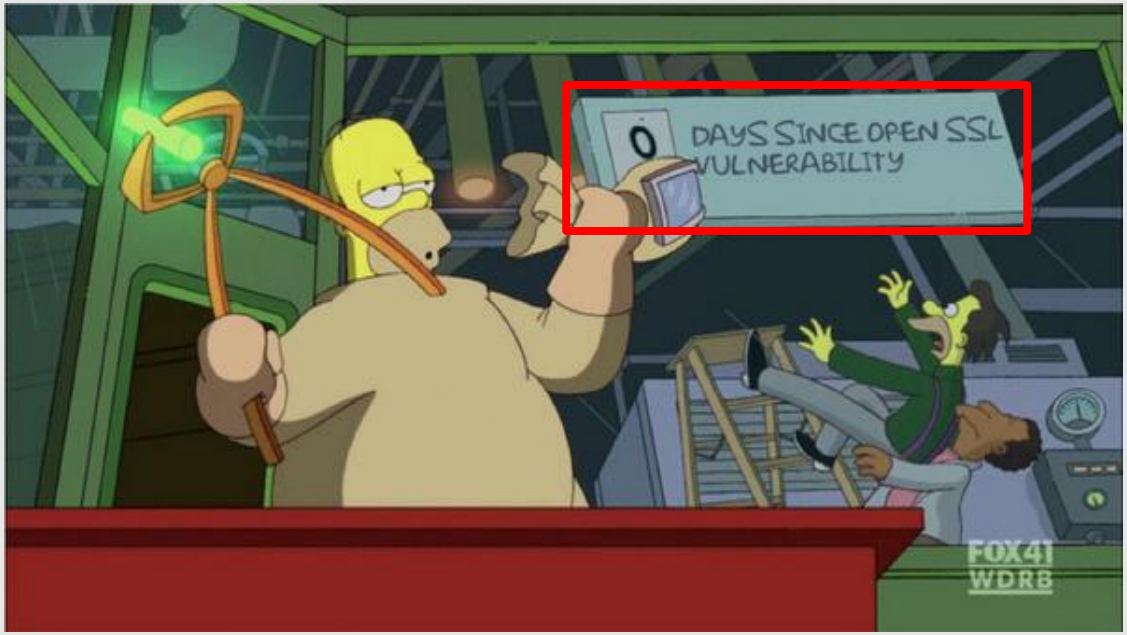
Bleichenbacher's Attack
Cavium – Misbehaving Hardware Accelerators

- Missing First Byte Check Requires Algorithm Changes (Oracle is Much Weaker)
- 2048 bit

  ~ 40 hours and ~7371 queries

Result Summary

TLS impl.	Side channel	Queries & Efficiency	
		Queries	Time
OpenSSL	timing	$O(2^{40})$	n.a.
JSSE	error message	177,000	12 h
JSSE	timing	18,600	19.5 h
Cavium	timing	7371	41 h



[Source **https://twitter.com/**]

Chris Meyer

christopher.meyer@rub.de

@armoredbarista

armoredbarista.blogspot.com