On the Security of TLS 1.3 and QUIC Against Weaknesses in PKCS#1 v1.5 Encryption

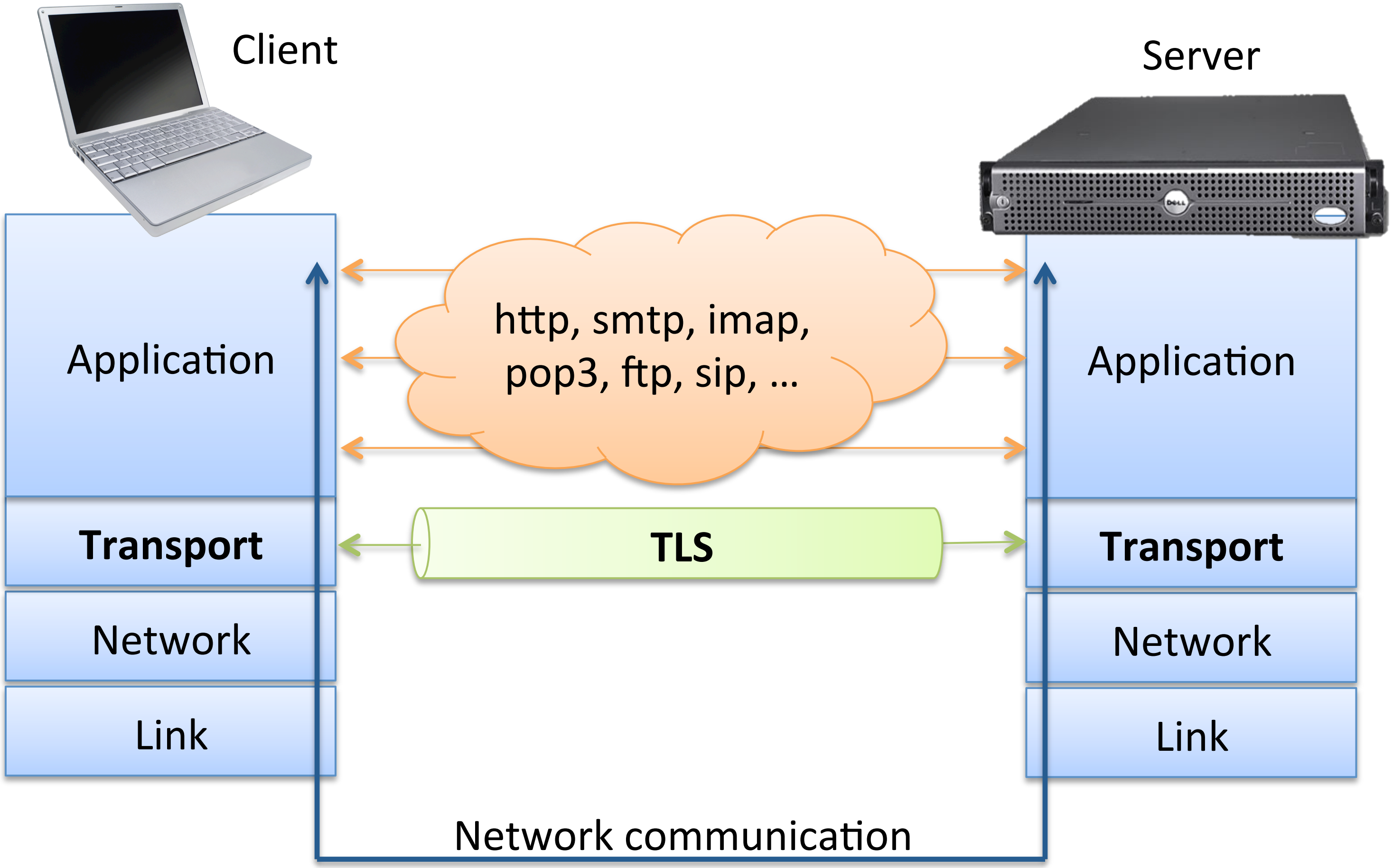
Tibor Jager, Jörg Schwenk, Juraj Somorovsky

Horst Görtz Institute for IT Security

Ruhr-University Bochum

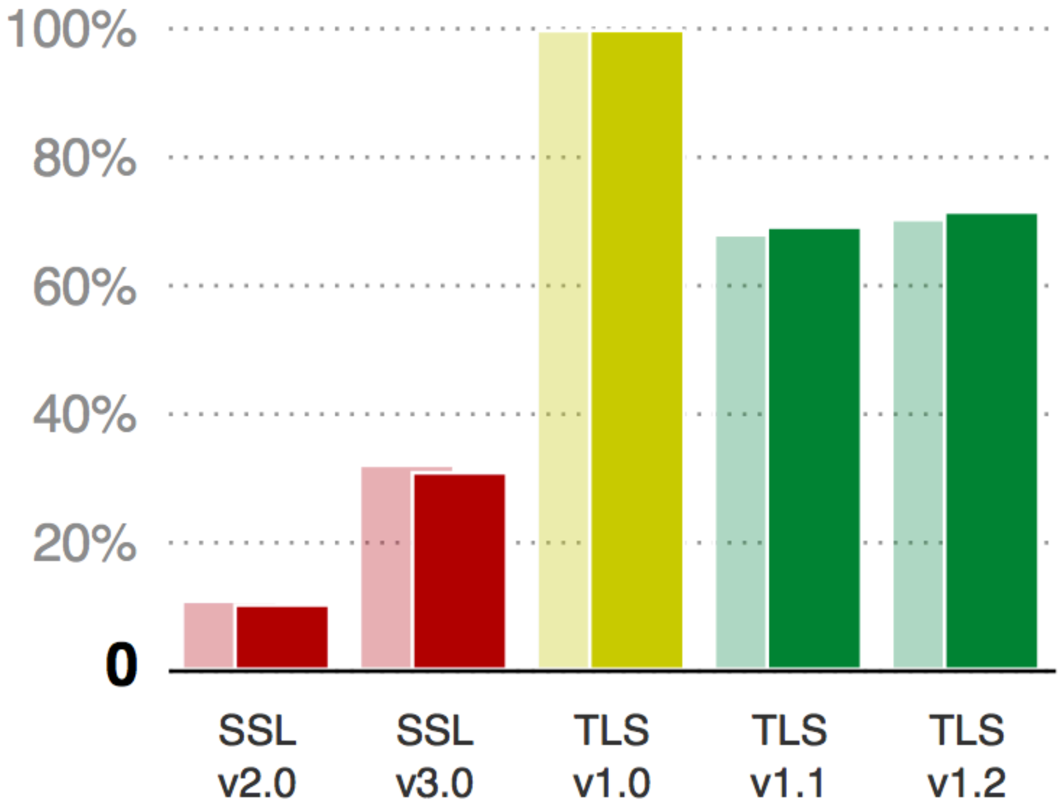
Real World Cryptography Conference 2016
6 January 2016
Stanford, CA, USA

Transport Layer Security (TLS)



Goal: provide **confidential**, **authenticated**, **integrity-protected** channel

Support of TLS versions in practice

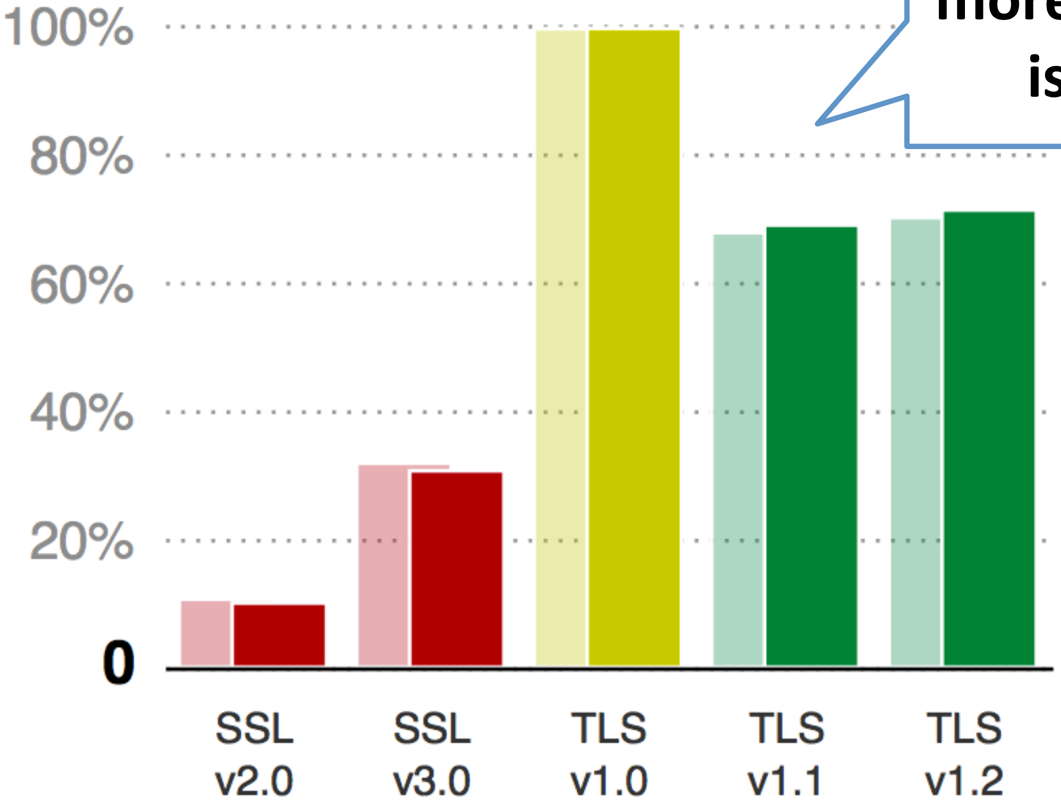


TLS v1.3

(1994) (1995) (1999) (2006) (2008) (2016?)

SSL Labs, https://www.trustworthyinternet.org/ssl-pulse/, Jan 5, 2016

Support of TLS versions in practice



(1994) (1995) (1999) (2006) (2008)

TLS v1.3 (2016?)

Support of **more than one version is very common**

SSL Labs, https://www.trustworthyinternet.org/ssl-pulse/, Jan 5, 2016

RSA-PKCS#1 v1.5 Encryption

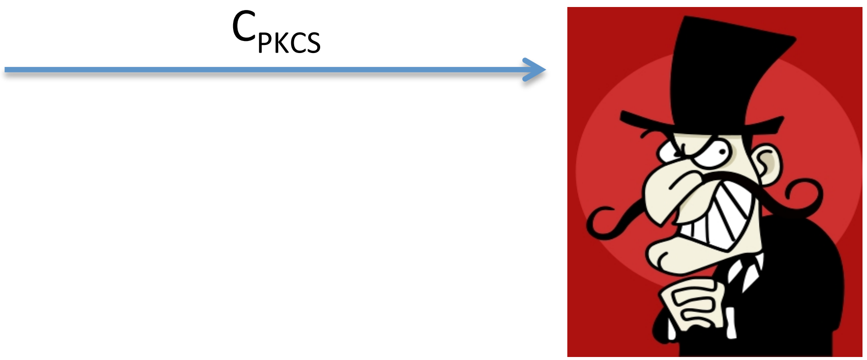
- **Most frequently used** key transport mechanism in TLS **before v1.3**
  - “Textbook-RSA encryption” with additional **randomized padding**
  - A ciphertext is “valid”, if it contains a **correctly padded** message

RSA-PKCS#1 v1.5 Encryption

- **Most frequently used** key transport mechanism in TLS **before v1.3**
  - "Textbook-RSA encryption" with additional **randomized padding**
  - A ciphertext is "valid", if it contains a **correctly padded** message
- **Deprecated** in TLS 1.3
  - Vulnerable: **Bleichenbacher's attack** (CRYPTO `98)
  - **Sufficient to protect against its weaknesses?**

Bleichenbacher's Attack

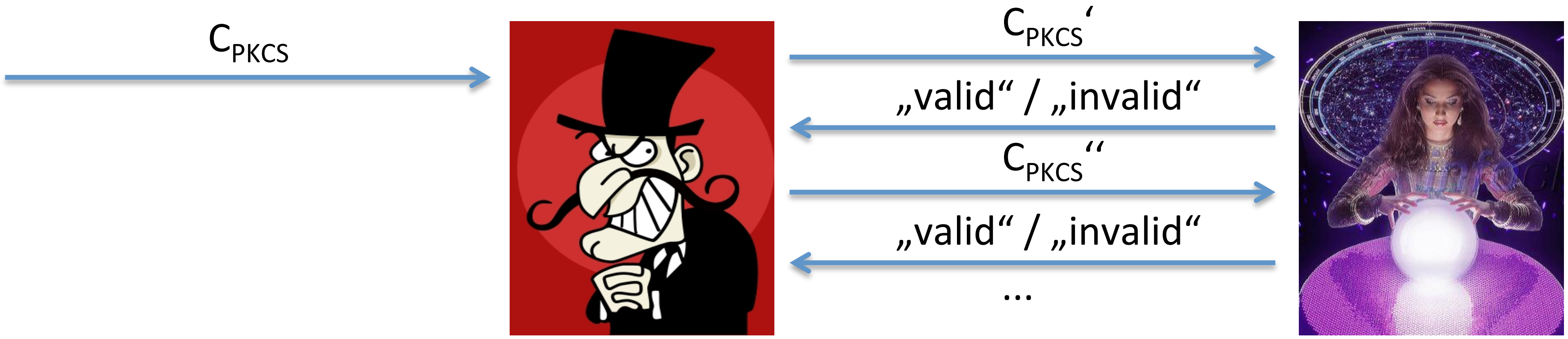
(CRYPTO 1998)



C_{PKCS}

Bleichenbacher’s Attack

(CRYPTO 1998)



C_{PKCS}

C_{PKCS}'

„valid“ / „invalid“

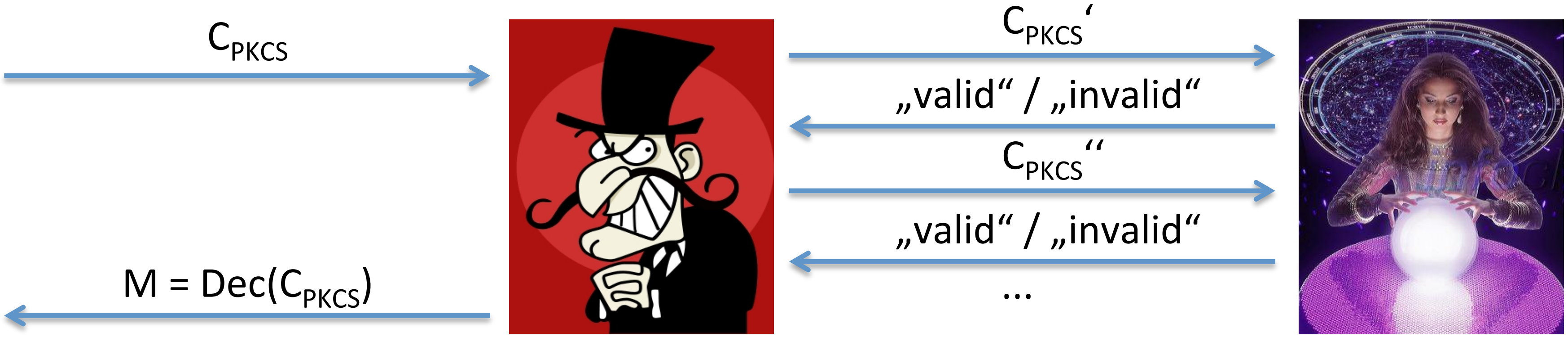
C_{PKCS}''

„valid“ / „invalid“

…

Bleichenbacher’s Attack

(CRYPTO 1998)



C_{PKCS}

C_{PKCS}‘

„valid“ / „invalid“

C_{PKCS}‘‘

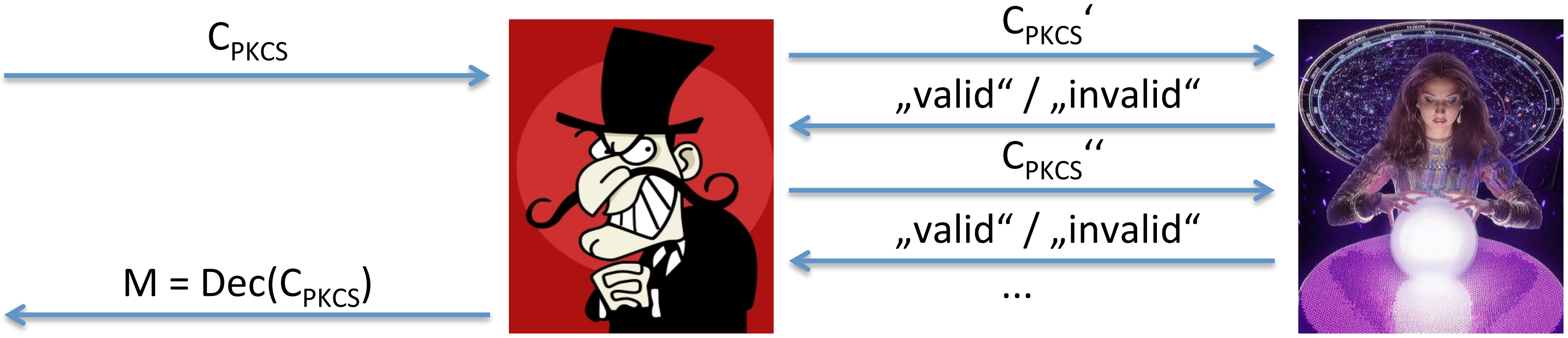
„valid“ / „invalid“

…

$M = Dec(C_{PKCS})$

Bleichenbacher's Attack

(CRYPTO 1998)



C_{PKCS}

$M = Dec(C_{PKCS})$

C_{PKCS}‘

„valid“ / „invalid“

C_{PKCS}‘‘

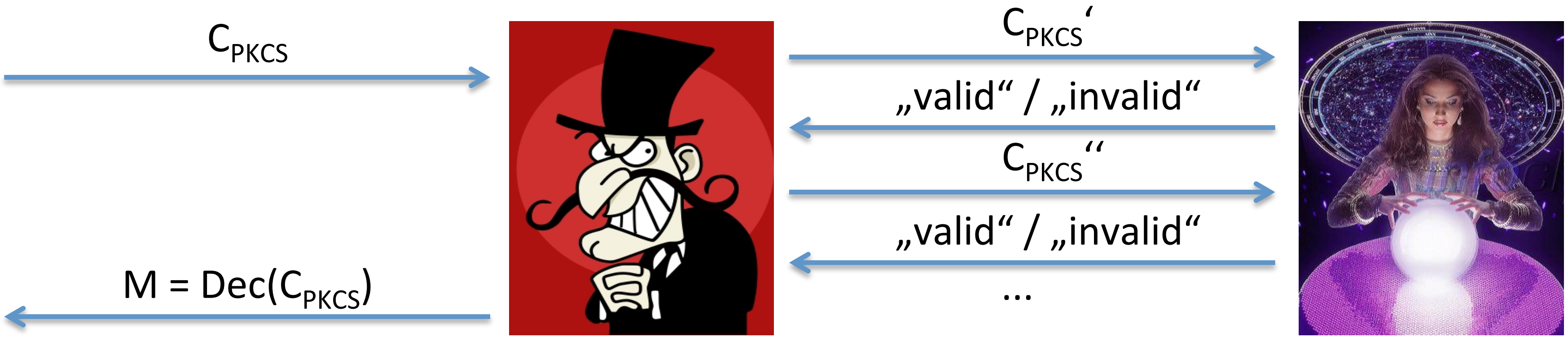
„valid“ / „invalid“

…

- Oracle usually provided by a server:
  - **Error message** if ciphertext is invalid
  - Other **side channels, like timing** (see Juraj's talk on Fri)
  - Other **side channels**

Bleichenbacher’s Attack

(CRYPTO 1998)



C_{PKCS}

$M = Dec(C_{PKCS})$

C_{PKCS}‘

„valid“ / „invalid“

C_{PKCS}‘‘

„valid“ / „invalid“

...

- Oracle usually provided by a server:
  - **Error message** if ciphertext is invalid
  - Other **side channels, like timing** (see Juraj’s talk on Fri)
  - Other **side channels**
- Allows to perform **RSA secret key operation**
  - Decrypt RSA-PKCS#1 v1.5 ciphertexts
  - Compute digital RSA signatures

Bleichenbacher attacks over and over

- Bleichenbacher (CRYPTO 1998)
- Klima et al. (CHES 2003)
- Jager et al. (ESORICS 2012)
- Degabriele et al. (CT-RSA 2012)
- Bardou et al. (CRYPTO 2012)
- Zhang et al. (ACM CCS 2014)
- Meyer et al. (USENIX Security 2014)
- …

Many different techniques to construct the required oracle

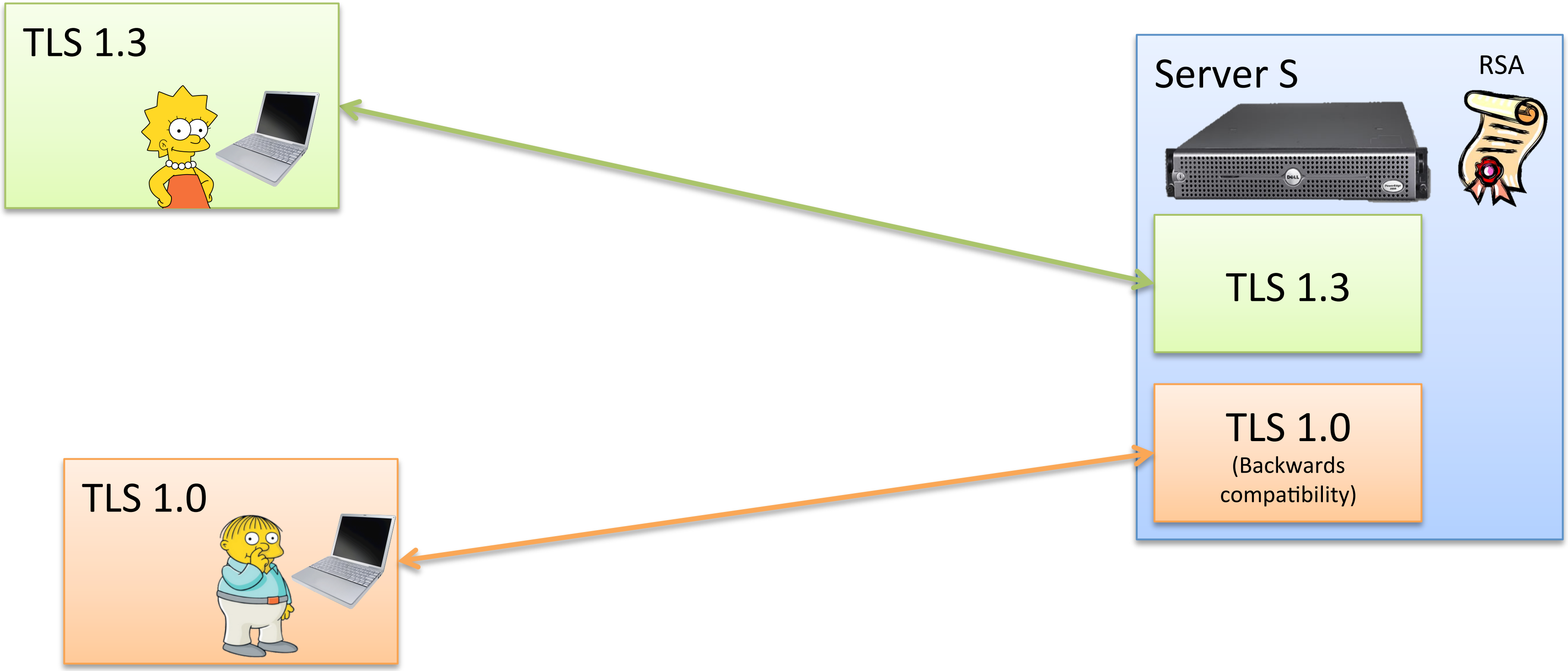
Bleichenbacher attacks over and over

- Bleichenbacher (CRYPTO 1998)
- Klima et al. (CHES 2003)
- Jager et al. (ESORICS 2012)
- Degabriele et al. (CT-RSA 2012)
- Bardou et al. (CRYPTO 2012)
- Zhang et al. (ACM CCS 2014)
- Meyer et al. (USENIX Security 2014)
- …

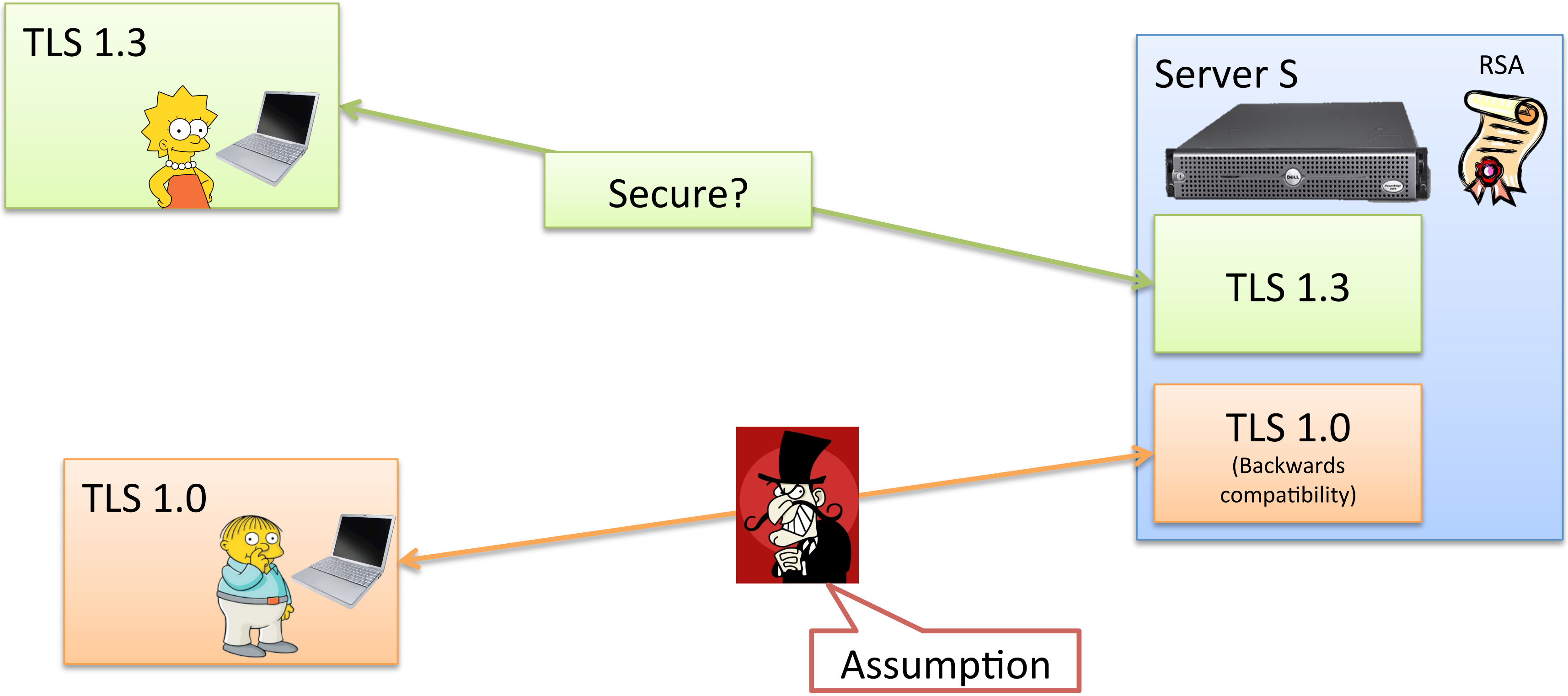
Many different techniques to construct the required oracle

Assumption: Bleichenbacher-like attacks remain a realistic threat

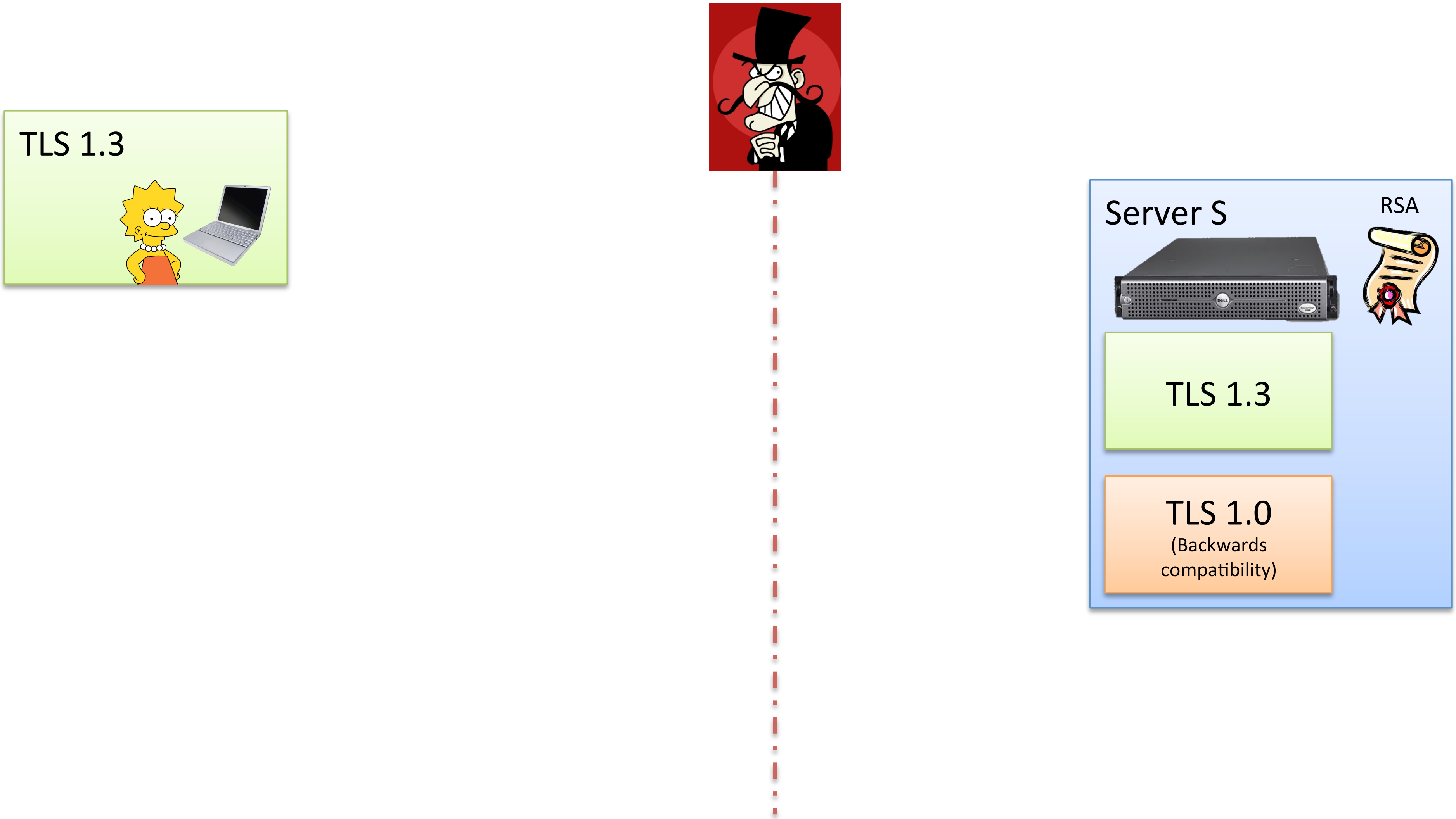
Typical use of TLS 1.3 in practice



Typical use of TLS 1.3 in practice



High-level Attack Description



TLS 1.3

Server S

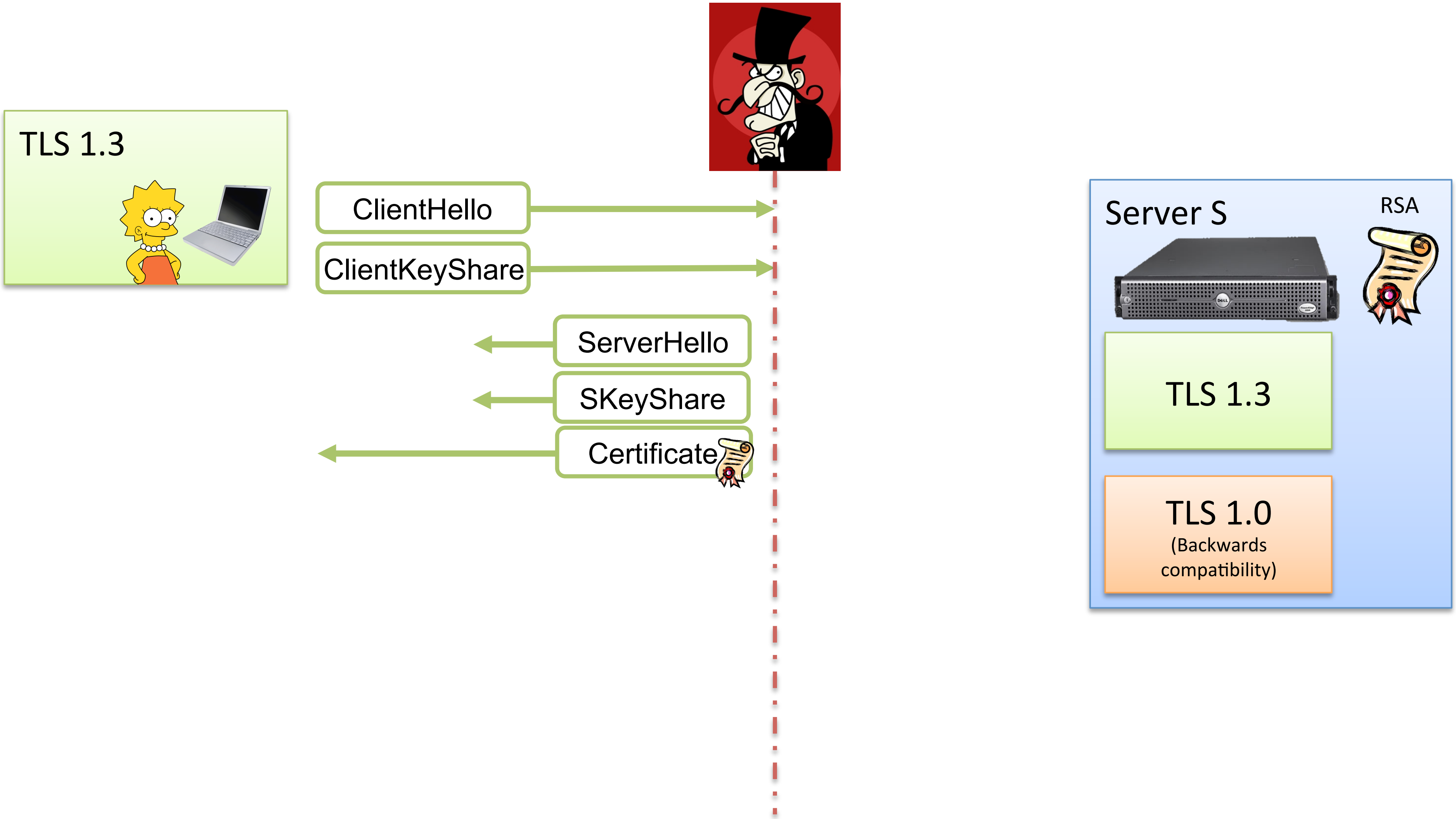
RSA

TLS 1.3

TLS 1.0

(Backwards compatibility)

High-level Attack Description



TLS 1.3

ClientHello

ClientKeyShare

ServerHello

SKeyShare

Certificate

Server S

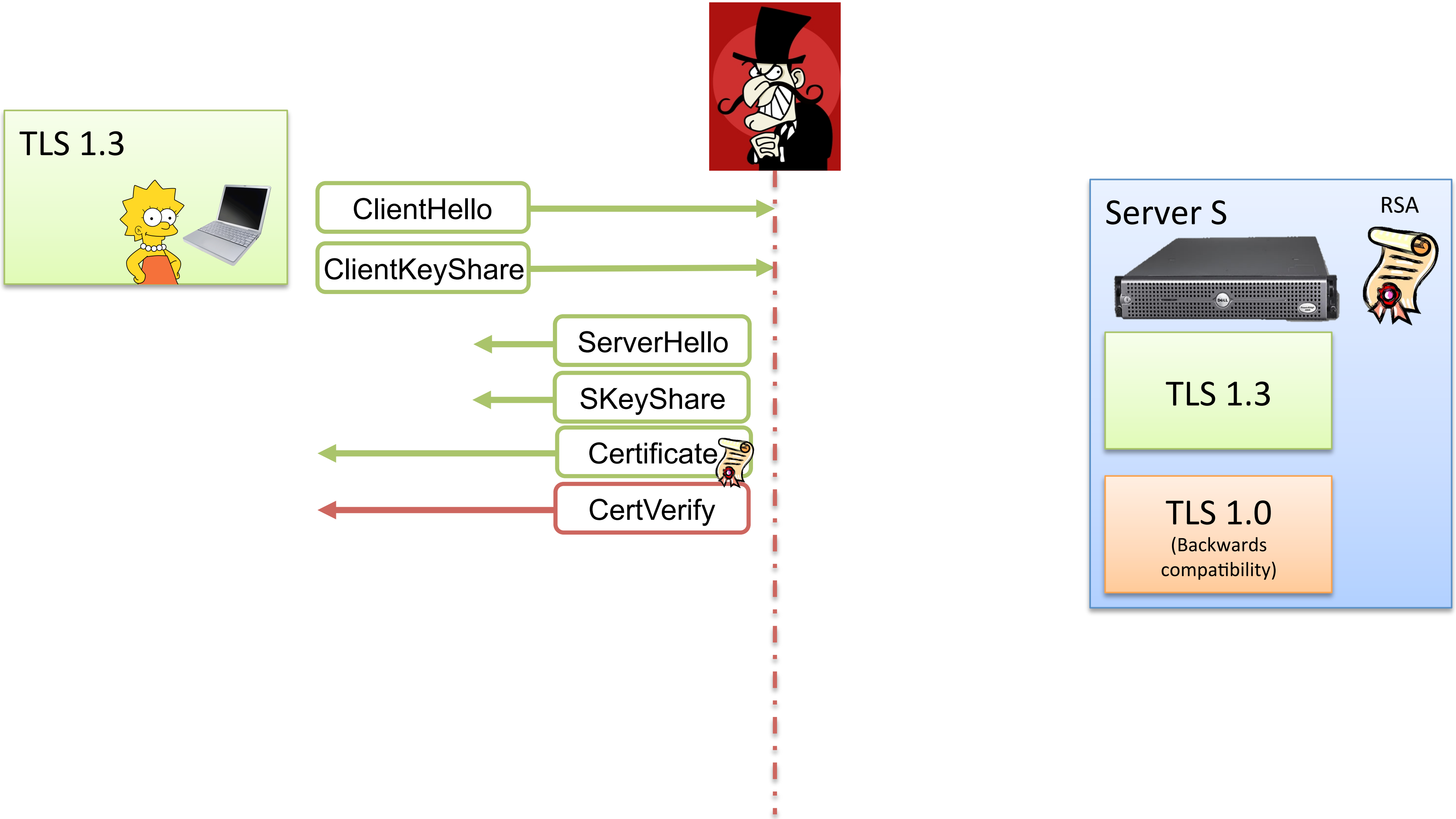
RSA

TLS 1.3

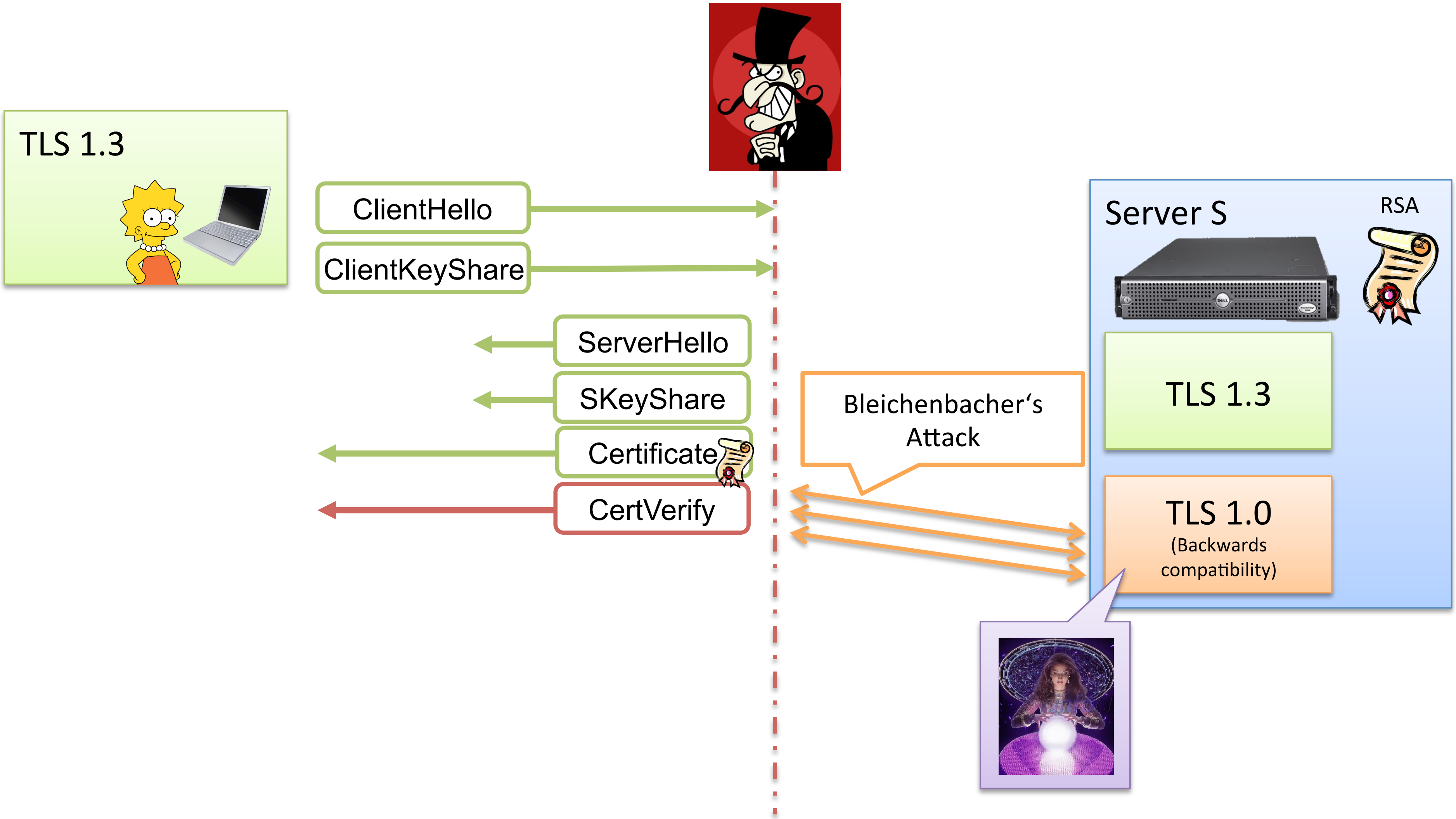
TLS 1.0

(Backwards compatibility)

High-level Attack Description



High-level Attack Description



TLS 1.3

ClientHello

ClientKeyShare

ServerHello

SKeyShare

Certificate

CertVerify

Bleichenbacher's Attack

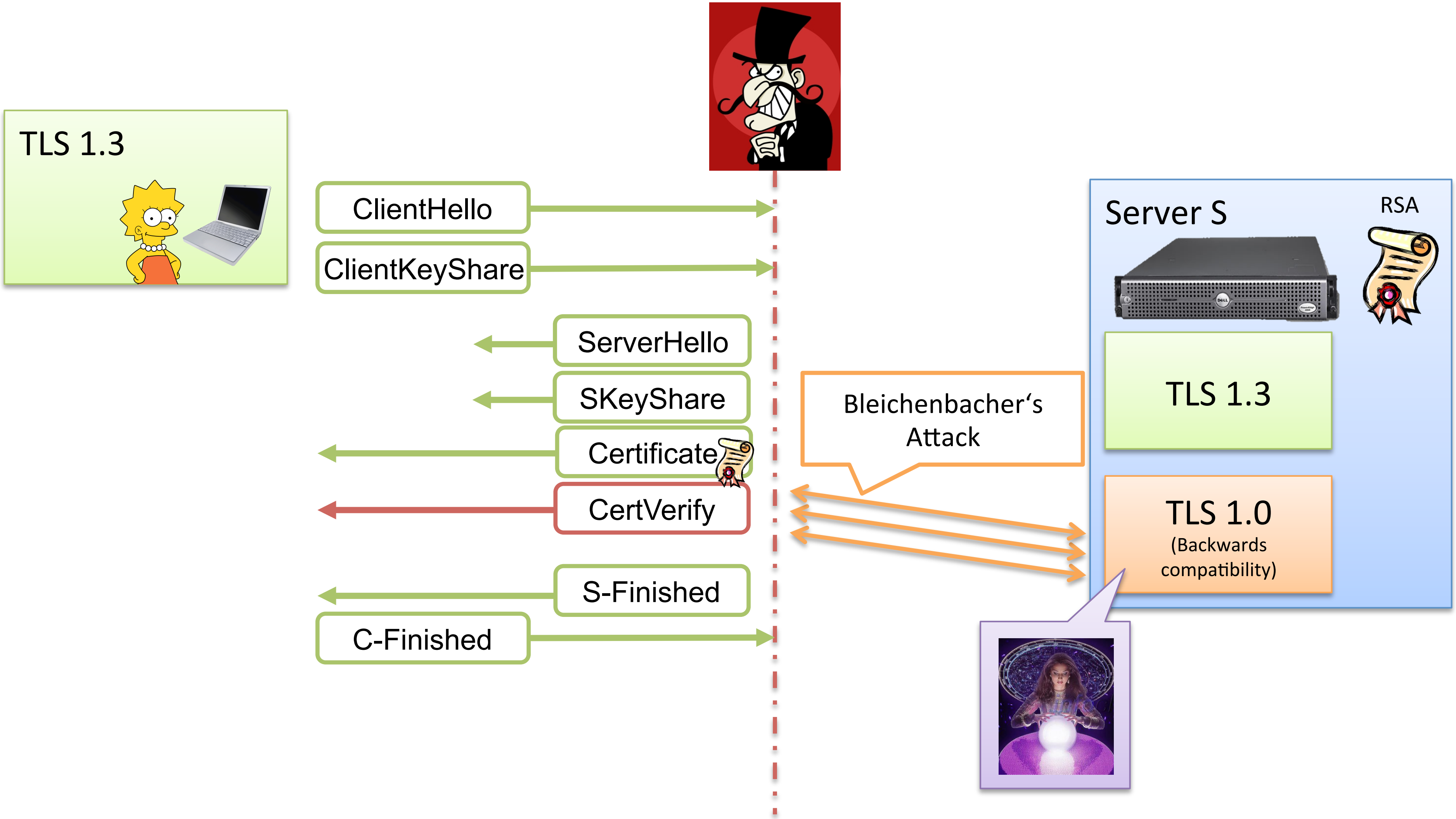
Server S

RSA

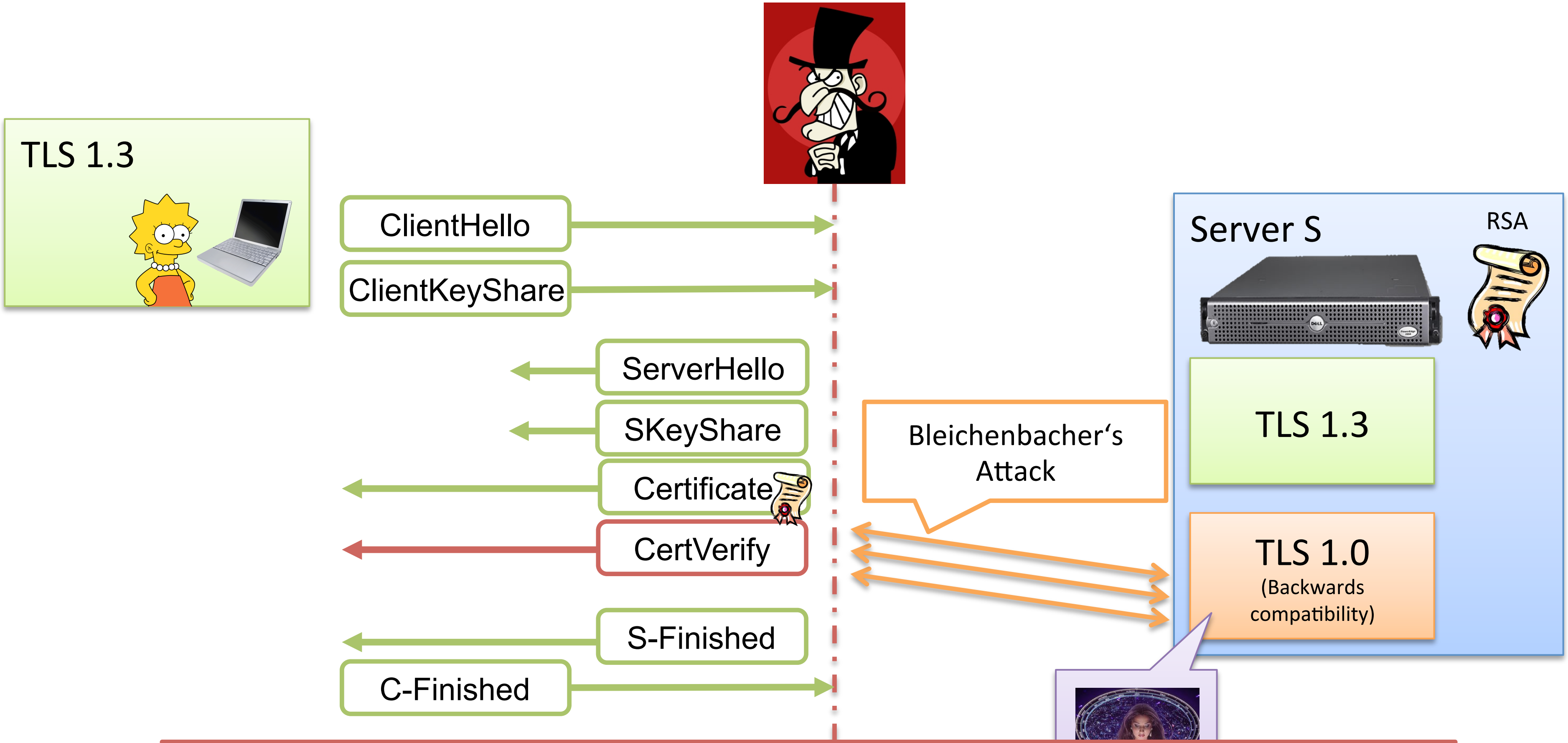
TLS 1.3

TLS 1.0
(Backwards compatibility)

High-level Attack Description



High-level Attack Description



TLS 1.3 may be vulnerable to Bleichenbacher's attack, **even though PKCS#1 v1.5 encryption is not used**!

Practical Impact

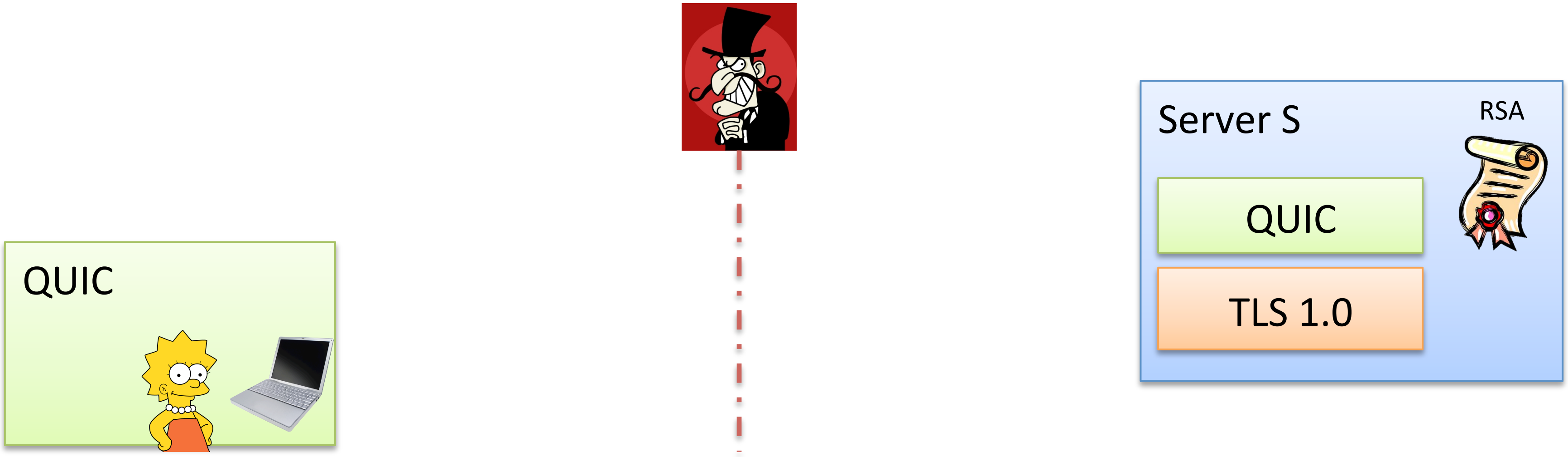
- Practical impact is **rather limited**
  - Typical Bleichenbacher-attacks take **hours or days**
  - **Would Lisa wait that long?**
  - Machine-to-machine communication?

Practical Impact

- Practical impact is **rather limited**
  - Typical Bleichenbacher-attacks take **hours or days**
  - **Would Lisa wait that long?**
  - Machine-to-machine communication?
- Nevertheless:
  - **Backwards compatibility** must be considered
  - Future **improvements of Bleichenbacher's** attack?

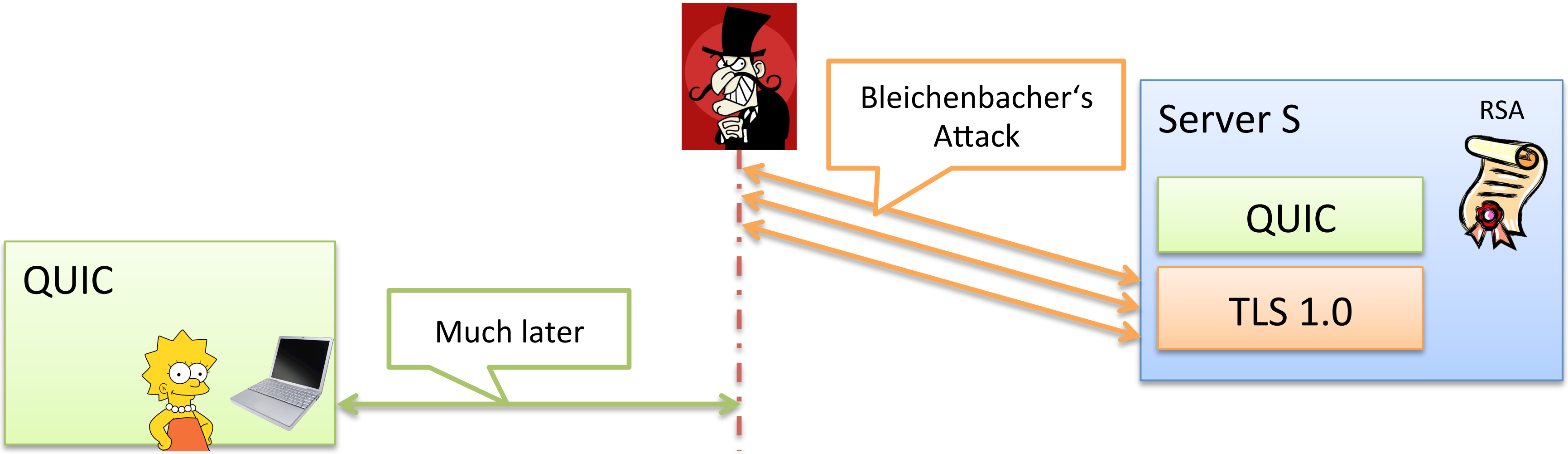
The QUIC Protocol

Google

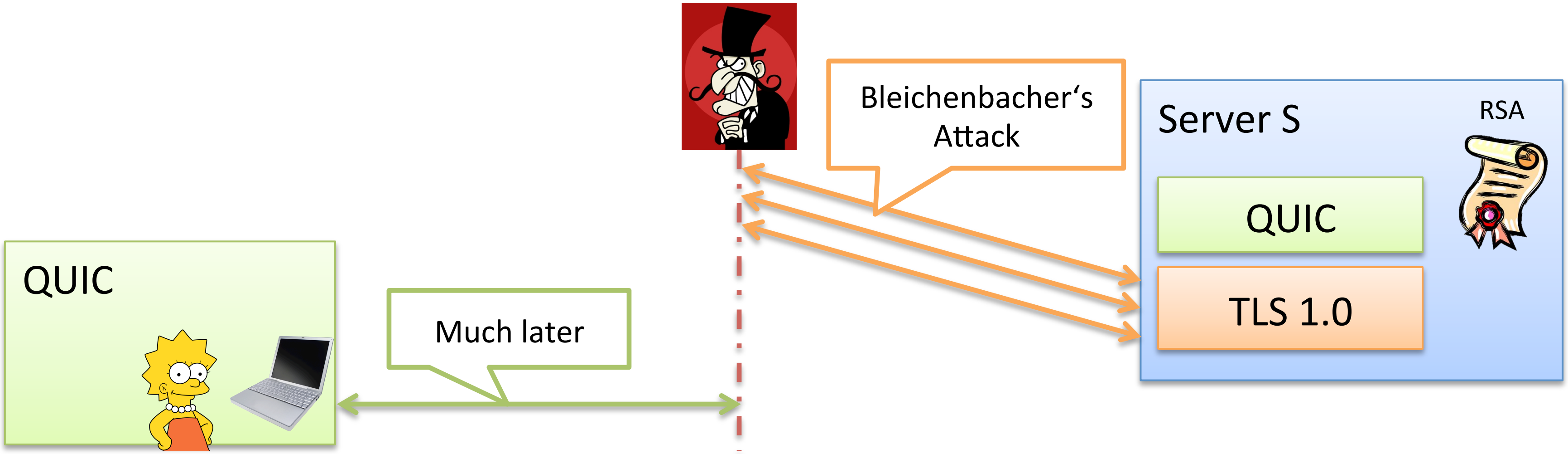


The QUIC Protocol

Google

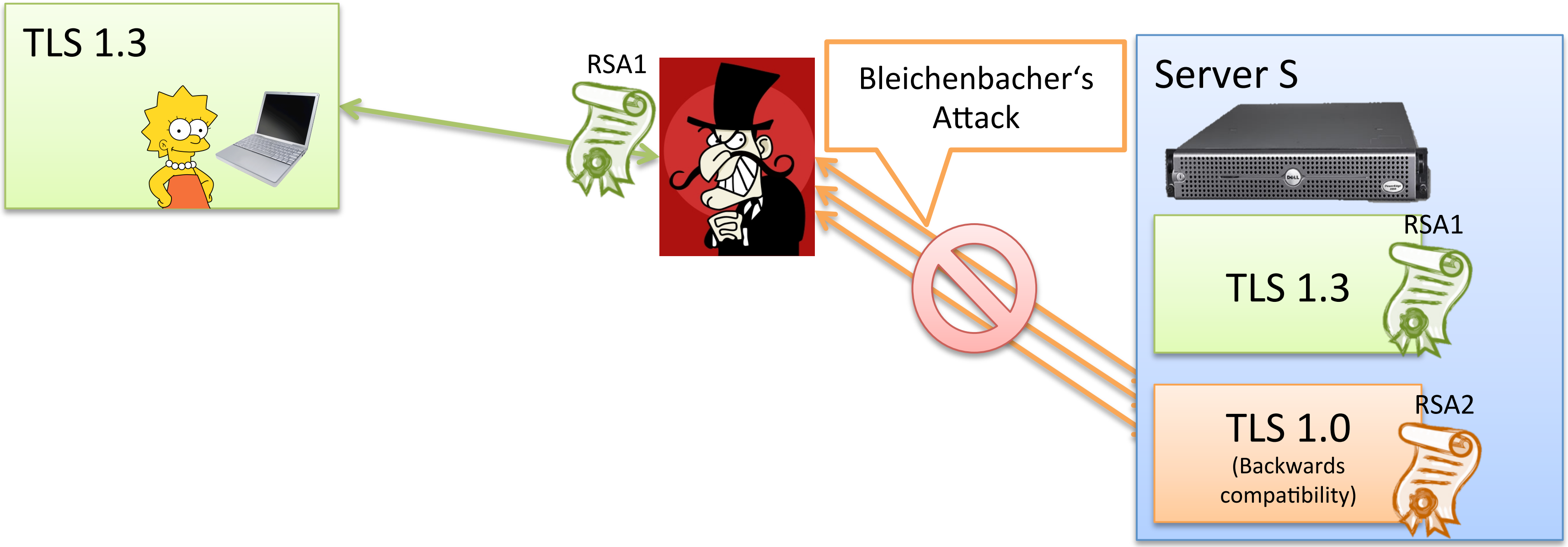


The QUIC Protocol

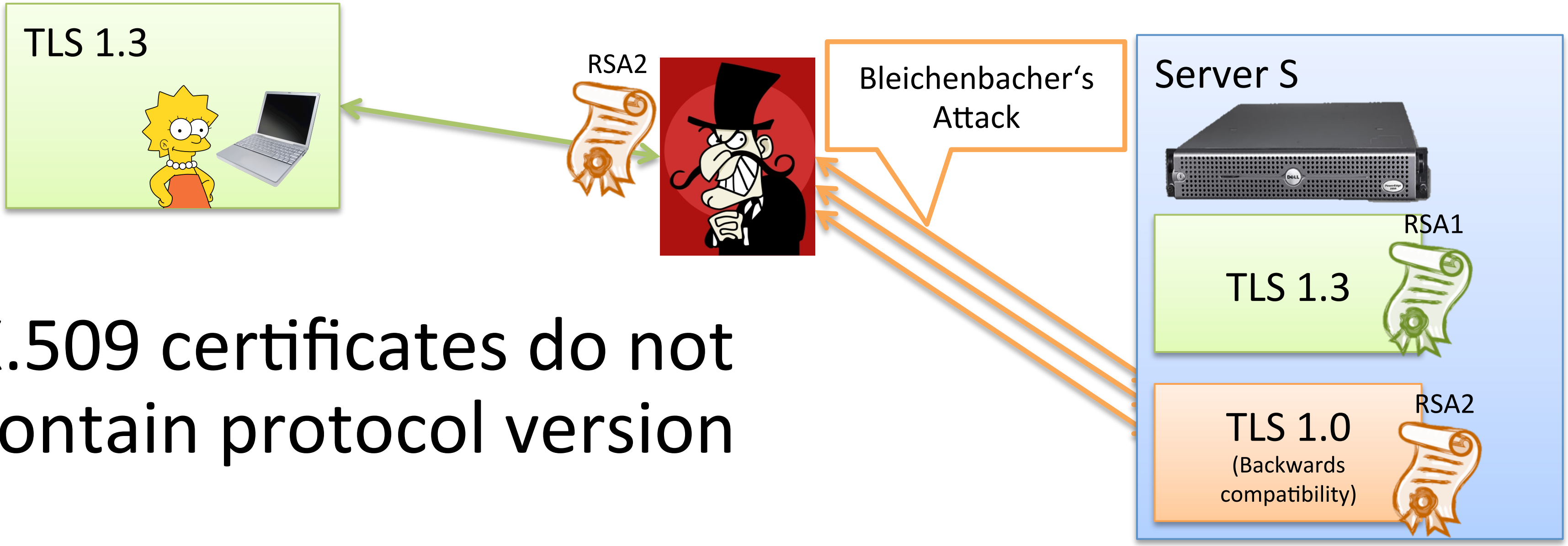


- Obtaining a digital signature is equivalent to retrieving the **server‘s secret key**!
- **Practical,** even if attack takes weeks!

The difficulty of preventing such attacks (example)

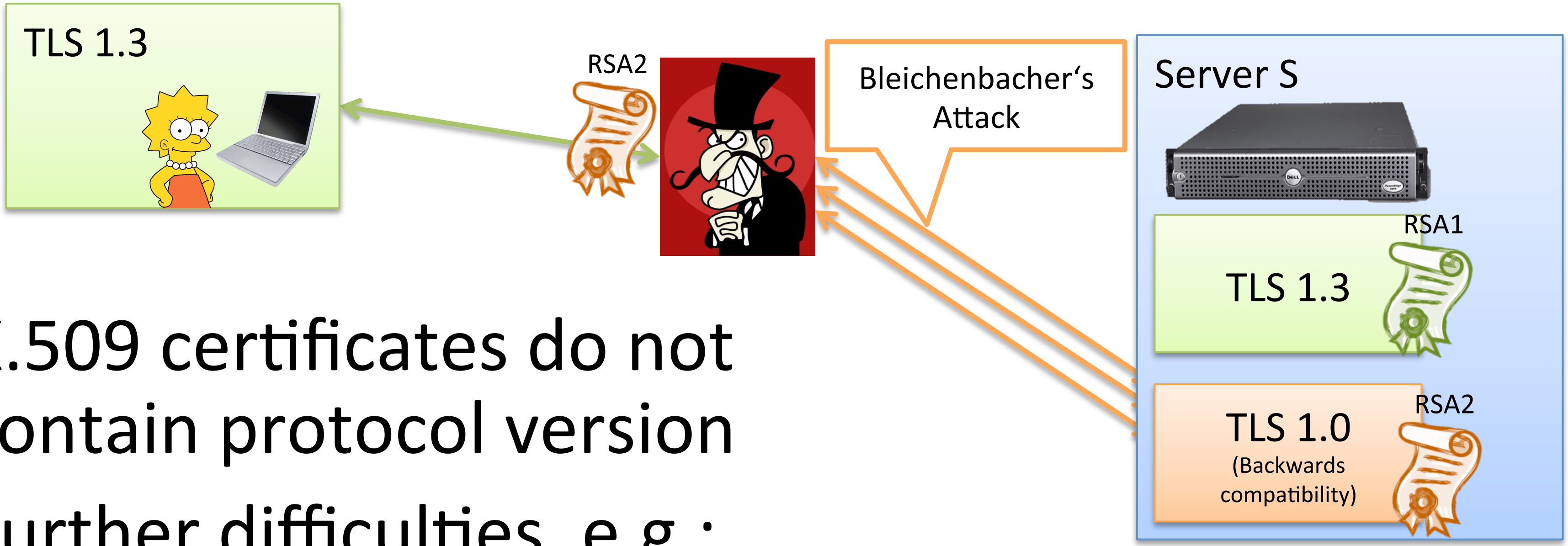


The difficulty of preventing such attacks (example)



- X.509 certificates do not contain protocol version

The difficulty of preventing such attacks (example)



- X.509 certificates do not contain protocol version
- Further difficulties, e.g.:
  - Key separation **not supported** by major server implementations
  - Certificates **cost money:** one for each version?

Summary

- Attacks on **TLS 1.3** and **QUIC**
  - Based on **backwards compatibility** and **potential Bleichenbacher** vulnerability
  - Removing an algorithm from a standard **not sufficient** to protect against its weakness
- Preventing this attack:
  - **Easy in Theory** (use key separation)
  - **Difficult in Practice** (due to practical constraints)

Summary

- Attacks on **TLS 1.3** and **QUIC**
  - Based on **backwards compatibility** and **potential Bleichenbacher** vulnerability
  - Removing an algorithm from a standard **not sufficient** to protect against its weakness
- Preventing this attack:
  - **Easy in Theory** (use key separation)
  - **Difficult in Practice** (due to practical constraints)

Thank you!