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

TRON 1.0 Workshop 2016
21 February 2016
San Diego, CA, USA
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)

PKCS-Ciphertext C

Plaintext M

C'

“valid” / “invalid”

C''

“valid” / “invalid”

...

4
Bleichenbacher’s Attack
(CRYPTO 1998)

• Oracle usually provided by a server:
  – Error message if ciphertext is invalid
  – Other side channels, like timing
Bleichenbacher’s Attack
(CRYPTO 1998)

- Oracle usually provided by a server:
  - Error message if ciphertext is invalid
  - Other side channels, like timing
- 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)
- ...

Assumption: Bleichenbacher-like attacks remain a realistic threat
Typical use of TLS 1.3 in practice
Typical use of TLS 1.3 in practice

Server S

TLS 1.3

TLS 1.0 (Backwards compatibility)

Assumption

Secure?

TLS 1.3

TLS 1.0

RSA
High-level Attack Description

TLS 1.3

Server S

TLS 1.3

TLS 1.0
(Backwards compatibility)

RSA
High-level Attack Description

TLS 1.3

ClientHello
ClientKeyShare
ServerHello
SKeyShare
Certificate

Server S
TLS 1.3
TLS 1.0
(Backwards compatibility)

RSA
High-level Attack Description

TLS 1.3

- ClientHello
- ClientKeyShare

Server S

- TLS 1.3
- TLS 1.0 (Backwards compatibility)

- ServerHello
- SKeyShare
- Certificate
- CertVerify
High-level Attack Description

TLS 1.3

ClientHello
ClientKeyShare
ServerHello
SKeyShare
Certificate
CertVerify

Server S
TLS 1.3

TLS 1.0
(Backwards compatibility)

Bleichenbacher’s Attack

RSA
High-level Attack Description

TLS 1.3

ClientHello
ClientKeyShare
ServerHello
SKeyShare
Certificate
CertVerify
S-Finished
C-Finished

Bleichenbacher’s Attack

TLS 1.3
TLS 1.0
(Backwards compatibility)
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 on TLS 1.3 is **rather limited**
  – Typical Bleichenbacher-attacks take **hours or days**
  – **Would Lisa wait that long?**
  – Machine-to-machine communication?
Practical Impact

• Practical impact on TLS 1.3 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
    • Cf. Jager, Paterson, Somorovsky (NDSS 2013)
  – Future **improvements of Bleichenbacher’s** attack?
Attack on the QUIC protocol

Attacker A

Server S
- QUIC
- TLS 1.0

Bleichenbacher’s Attack

Full QUIC protocol
A can run Bleichenbacher’s attack before Lisa connects to S. One signature is equivalent to the secret key of S. Practical, even if attack takes weeks!
Limited Impact on TLS 1.3

- A can impersonate S only in a **single** TLS session
- Only practical with **very fast** Bleichenbacher attack

Attacker A

TLS 1.3

“Hello”

CertVerify

“Finished”

Server S

- TLS 1.3
- TLS 1.0

RSA
The difficulty of preventing such attacks (example)
The difficulty of preventing such attacks (example)
The difficulty of preventing such attacks (example)

• X.509 certificates do not contain protocol version
Further difficulties

• Key separation **not supported** by major server implementations
• Certificates **cost money** (extended validation)
• X.509 supports “sign/encrypt-only” certs
  – “Sign-only” keys for TLS >= 1.3
  – “Encrypt-only” keys for TLS <= 1.2
    • **No Forward Secrecy** for versions <= 1.2 😞
  – Do browsers really check this?
Summary and recommendations

• Removing RSA-PKCS#1 v1.5 from TLS is an excellent decision
  – Not sufficient to protect completely against weakness
• TLS 1.3 is more “robust” than QUIC
  – But not immune
  – Signing ephemeral values is a good idea
• Recommendation for future TLS versions: promote key separation
  – Talk to X.509 and software developers