Practical Attacks on Real World Crypto Implementations

Juraj Somorovsky
Ruhr University Bochum, HGI 3curity

@jurajsomorovsky
Recent years revealed many crypto attacks...

- ESORICS 2004, Bard: The vulnerability of SSL to Chosen Plaintext

- Eurocrypt 2002, Vaudenay: Flaws Induced by CBC Padding—Applications to SSL, IPSEC, WTLS

- Crypto 1998, Bleichenbacher: Chosen Ciphertext Attacks Against Protocols Based on the RSA Encryption Standard PKCS #1

- 2011 BEAST

- 2013/14 POODLE, Lucky13

- 2012 XML Encryption
Standards updated

• Countermeasures defined

• What could go wrong in RWC implementations?
Overview

1. Bleichenbacher’s Attack
   • XML Encryption
   • TLS
2. Invalid Curve Attack
   • TLS
   • Hardware Security Modules
RSA-PKCS#1 v1.5

- Used to encrypt symmetric keys
- Vulnerable to an adaptive chosen-ciphertext attack

Client

\[ C = \text{Enc}(M) \]

Server

\[ C_1 \]
valid/invalid

\[ C_2 \]
valid/invalid

\[ \ldots \]
(repeated several times)
RSA-PKCS#1 v1.5: Countermeasures

1. Use RSA-OAEP (PKCS#1 v2)
2. Apply specific countermeasure

```python
generate random
decrypt ciphertext: m = dec(c)
if ( padding correct )
    proceed with m
else
    proceed with random
```
Overview

1. Bleichenbacher’s Attack
   - XML Encryption
   - TLS

2. Invalid Curve Attack
   - TLS
   - Hardware Security Modules
RSA PKCS#1 v1.5 in XML Encryption

• Hybrid encryption:

\[
\begin{align*}
    k &= \text{Dec}_{\text{pkcs}}(\text{priv}, C_1) \\
    m &= \text{Dec}_{\text{aes128}}(k, C_2)
\end{align*}
\]
Attack Countermeasure

• Hybrid encryption:

\[ k = \text{Dec}_{\text{pkcs}}(\text{priv}, C1) \]
\[ m = \text{Dec}_{\text{aes128}}(k, C2) \]
Case Apache WSS4J

- Hybrid encryption:

\[
\begin{align*}
k &= \text{Dec}_{\text{pkcs}}(\text{priv}, C1) \\
m &= \text{Dec}_{\text{aes128}}(k, C2)
\end{align*}
\]
Case Apache WSS4J

• Hybrid encryption:

\[ k = \text{Dec}_{\text{pkcs}}(\text{priv}, C1) \]
\[ m = \text{Dec}_{\text{aes128}}(k, C2) \]
Case Apache WSS4J

• Original bug much more complicated

• CVE-2015-0226

• Dennis Kupser, Christian Mainka, Jörg Schwenk, Juraj Somorovsky: How to Break XML Encryption – Automatically (WOOT‘15)

• Found automatically using WS-Attacker

• https://github.com/RUB-NDS/WS-Attacker
Overview

1. Bleichenbacher’s Attack
   - XML Encryption
   - TLS

2. Invalid Curve Attack
   - TLS
   - Hardware Security Modules
How About TLS?


• Practical attacks on JSSE, Bouncy Castle, Cavium Accelerator

• Bug in OpenSSL
Case JSSE

- No direct TLS error messages
- Uses PKCS#1 unpadding function:

```java
private byte [] unpadV15 (byte[] padded) {
    if (PKCS valid) {
        return unpadded text;
    } else {
        throw new BadPaddingException();
    }
}
```

- Caught, random generated...what’s wrong?
Case JSSE (CVE-2014-411)

- Exception consumes about 20 microseconds!

Bleichenbacher’s Attack over LAN!
Overview

1. Bleichenbacher’s Attack
   - XML Encryption
   - TLS

2. Invalid Curve Attack
   - TLS
   - Hardware Security Modules
Elliptic Curve

- Set of points over a finite field
- Used e.g. for key exchange

Key: $sP$
Invalid Curve Attack

- Crypto 2000: Biehl, Meyer, Müller
- Attacker sends an invalid point of small order (e.g. 5)

- Attacker computes:
  \[ s_1 = s \mod 5 \]
Invalid Curve Attack

• Choose points of small co-prime order \((5, 7, 11, \ldots)\)
• Send to the server
• Compute:
  \[
  s_1 = s \mod 5 \\
  s_2 = s \mod 7 \\
  s_3 = s \mod 11 \\
  s_4 = s \mod 13
  \]
• Compute \(s\) with CRT
Overview

1. Bleichenbacher’s Attack
   • XML Encryption
   • TLS
2. Invalid Curve Attack
   • TLS
   • Hardware Security Modules
Practical Attacks?

- Analyzed 8 libraries
- 2 vulnerable
  - Bouncy Castle: 3300 TLS queries
  - Oracle JSSE: 17000 TLS queries
Impact

- Attacks extract server private keys
- Java servers using EC certificates vulnerable
  - For example Apache Tomcat

Demo
Overview

1. Bleichenbacher’s Attack
   - XML Encryption
   - TLS

2. Invalid Curve Attack
   - TLS
   - Hardware Security Modules
Attacker Model in HSM Scenarios

- Storage of crypto keys
- Keys never leave HSMs
Attacker Model in HSM Scenarios

• Storage of crypto keys
• Keys never leave HSMs
How about Invalid Curve Attacks?

- CVE-2015-6924 (with Dennis Felsch)
- Utimaco HSMs vulnerable
- < 100 queries to get a key...Heartbleed effect

- Thanks to cooperation of Utimaco
  - Provided sample code, fast fix
- Utimaco HSM is FIPS certified

"Catastrophic" is the right word. On the scale of 1 to 10, this is an 11.
Conclusions

• Old attacks relevant for RWC implementations
• Old algorithms in the newest standards
  – RSA PKCS#1 v1.5 (attack: 1998)
    2008: TLS 1.2
    2013: XML Encryption 1.1
    2015: JSON Web Encryption

11.4. Adaptive Chosen-Ciphertext Attacks

When decrypting, particular care must be taken not to allow the JWE recipient to be used as an oracle for decrypting messages. RFC 3218 [RFC3218] should be consulted for specific countermeasures to attacks on RSAES-PKCS1-v1_5. An attacker might modify the contents of the

– Positive example: **TLS 1.3**
Conclusions

• For standard designers:
  – Remove old crypto

• For developers:
  – Analyze possible side-channels, best practices
    • Check point is on curve

• For pentesters:
  – More tools / analyses of crypto applications needed