Practical Attacks on  Implementations

Juraj Somorovsky
Ruhr University Bochum, HGI
3curity

RUHR UNIVERSITÄT BOCHUM RUB

3curity

@jurajsomorovsky

Recent years revealed many crypto attacks…

- ESORICS 2004, Bard, [illegible] bility of SSL to Chosen Plainte[illegible]

2011 BEAST

- Eurocrypt 2002, Va[illegible] Flaws Induced by CBC Padding—App[illegible] EC, WTLS

2013/14 POODLE, Lucky13

- Crypto 1998, Bleichenbacher: Chosen Ciphertext Attacks Ag[illegible] RSA Encryption Standard PKCS #1

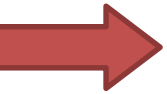
2012 XML Encryption

Standards updated

- Countermeasures defined

- What could go wrong in RWC implementations?

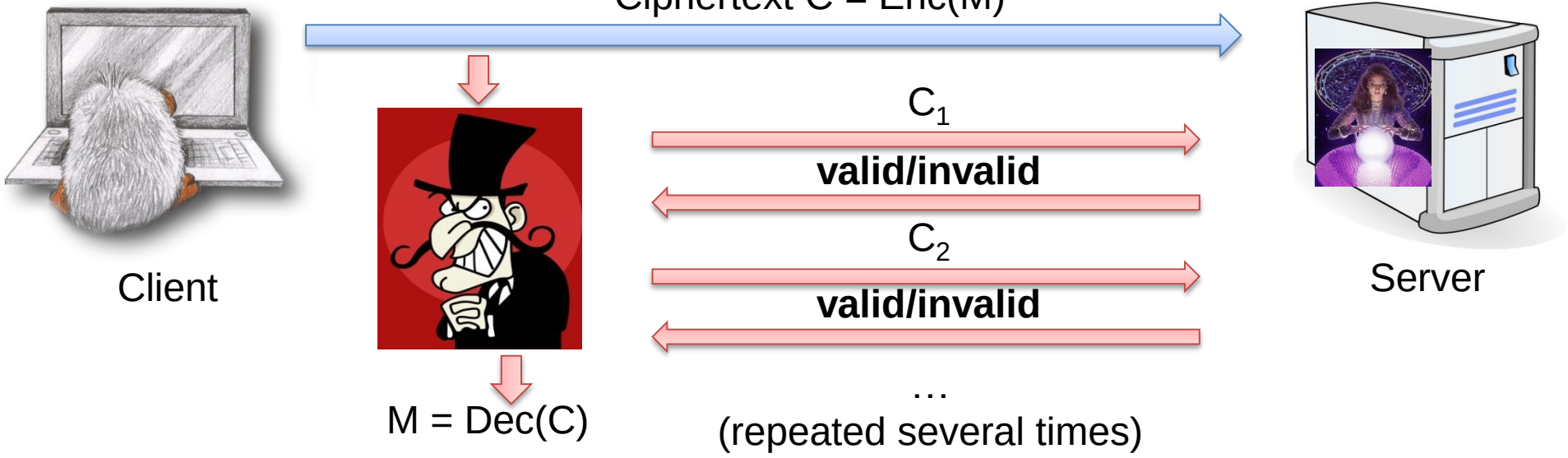
Overview



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



RSA-PKCS#1 v1.5: Countermeasures

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

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

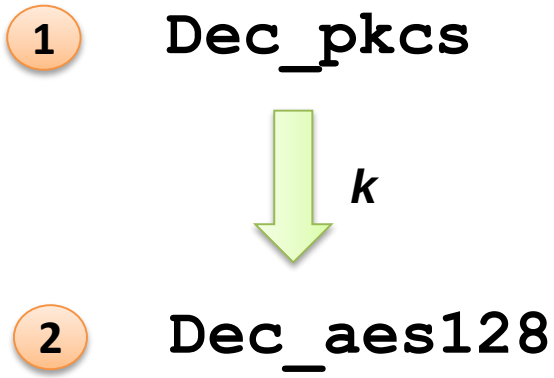
Overview

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:

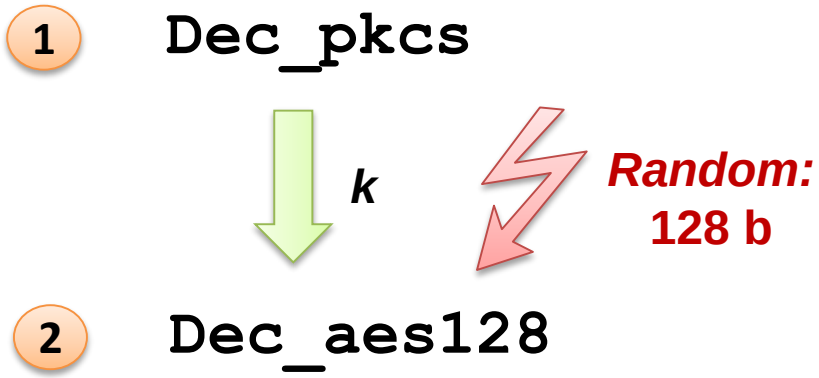
```
k = Dec_pkcs(priv,C1)
m = Dec_aes128(k,C2)
```



Attack Countermeasure

- Hybrid encryption:

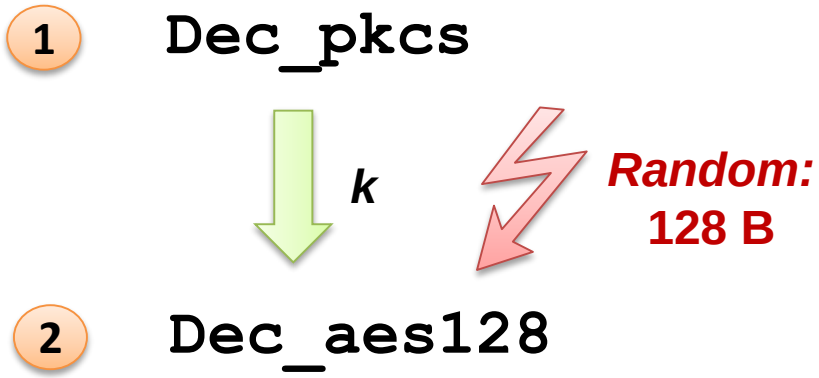
```
k = Dec_pkcs(priv,C1)
m = Dec_aes128(k,C2)
```



Case Apache WSS4J

- Hybrid encryption:

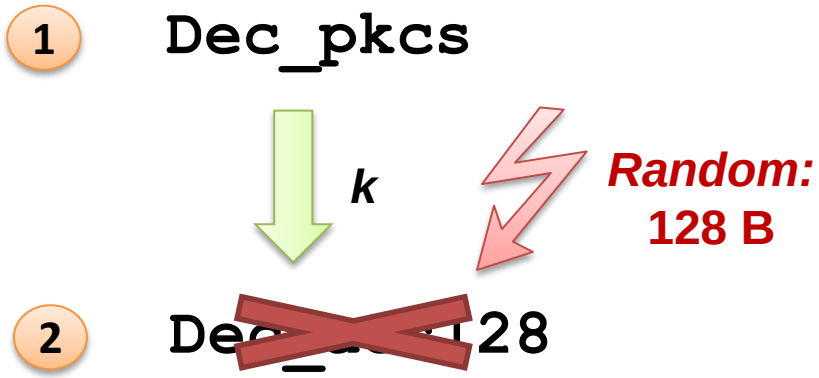
```
k = Dec_pkcs(priv,C1)
m = Dec_aes128(k,C2)
```



Case Apache WSS4J

- Hybrid encryption:

```
k = Dec_pkcs(priv,C1)
m = Dec_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?

- Christopher Meyer, Juraj Somorovsky, Jörg Schwenk, Eugen Weiss, Sebastian Schinzel, Erik Tews: **Revisiting SSL/TLS Implementations: New Bleichenbacher Side Channels and Attacks**. USENIX Security 2014
- Practical attacks on JSSE, Bouncy Castle, Cavium Accelerator
- Bug in OpenSSL

Case JSSE

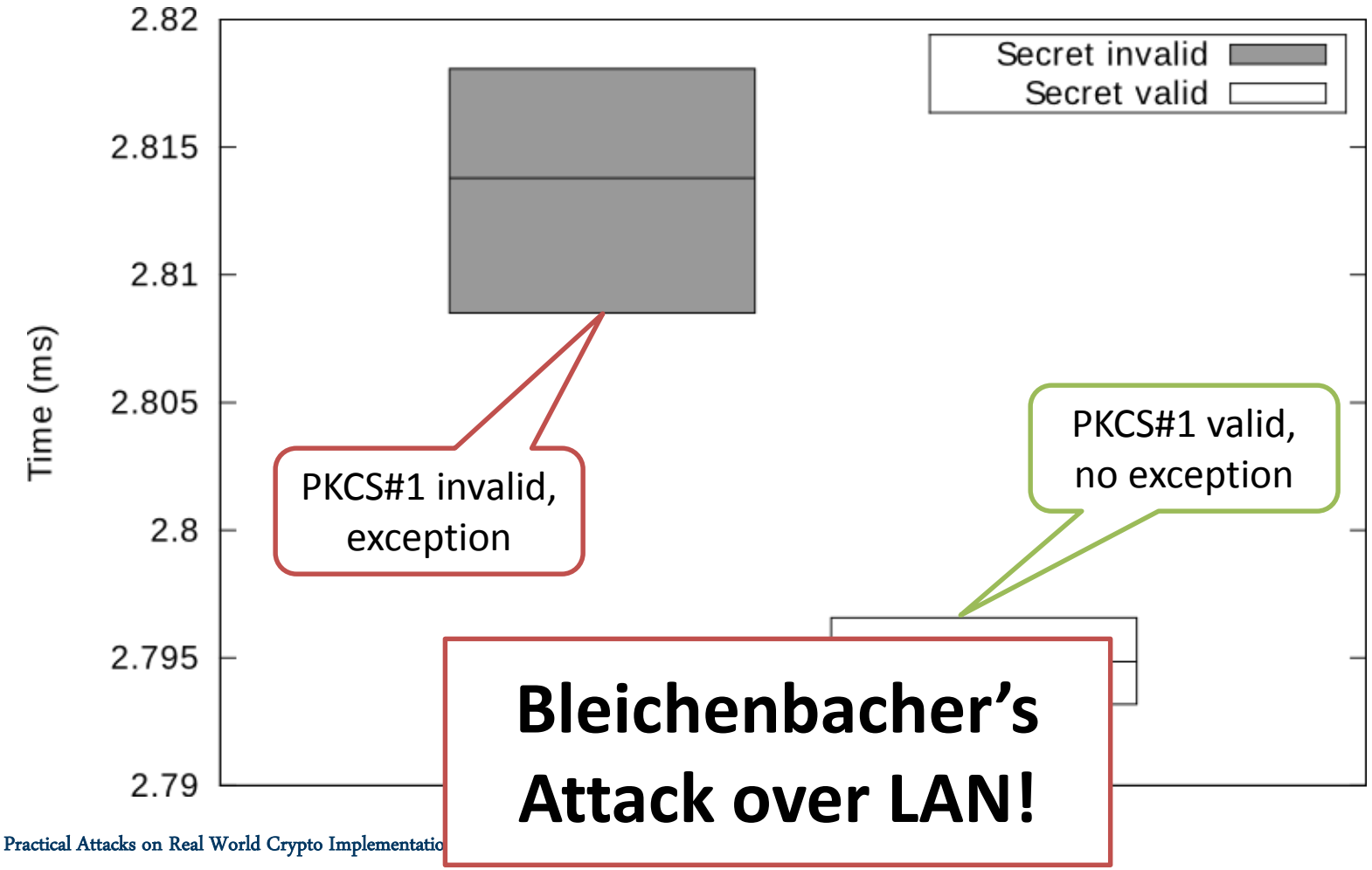
- No direct TLS error messages
- Uses PKCS#1 unpadding function:

```
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!

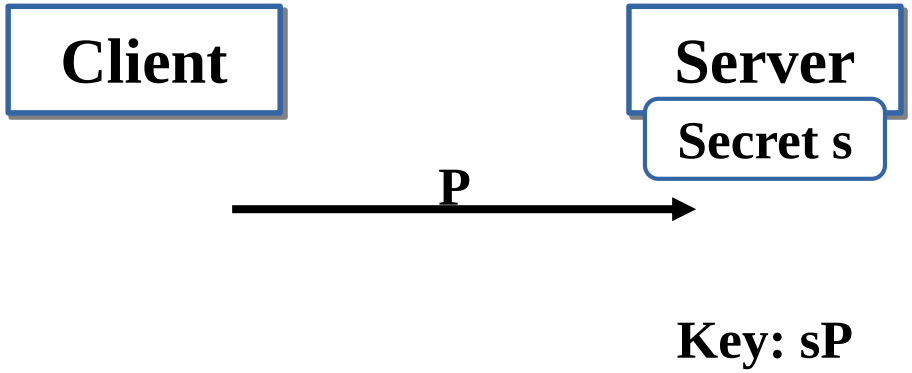


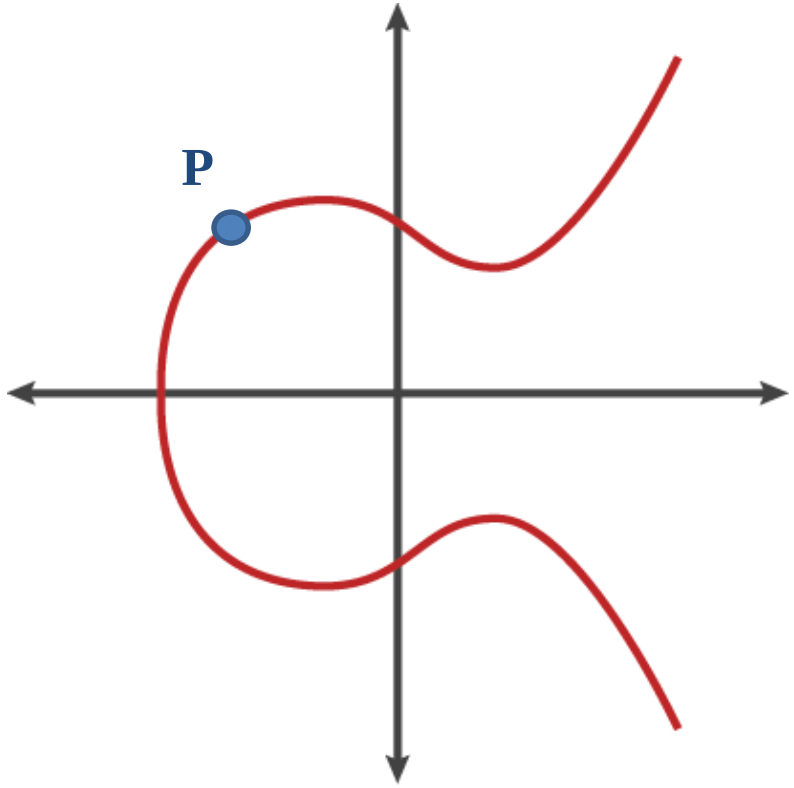
Overview

1. Bleichenbacher's Attack
   - XML Encryption
   - TLS
2. Invalid Curve Attack
   - TLS
   - Hardware Security Modules

Elliptic Curve

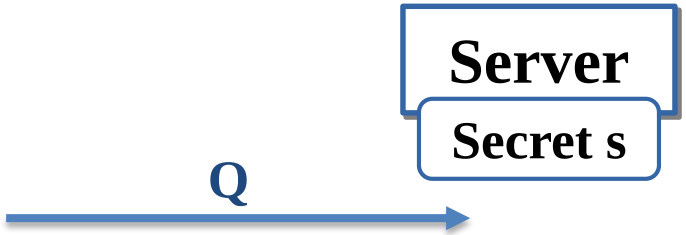
- Set of points over a finite field
- Used e.g. for key exchange

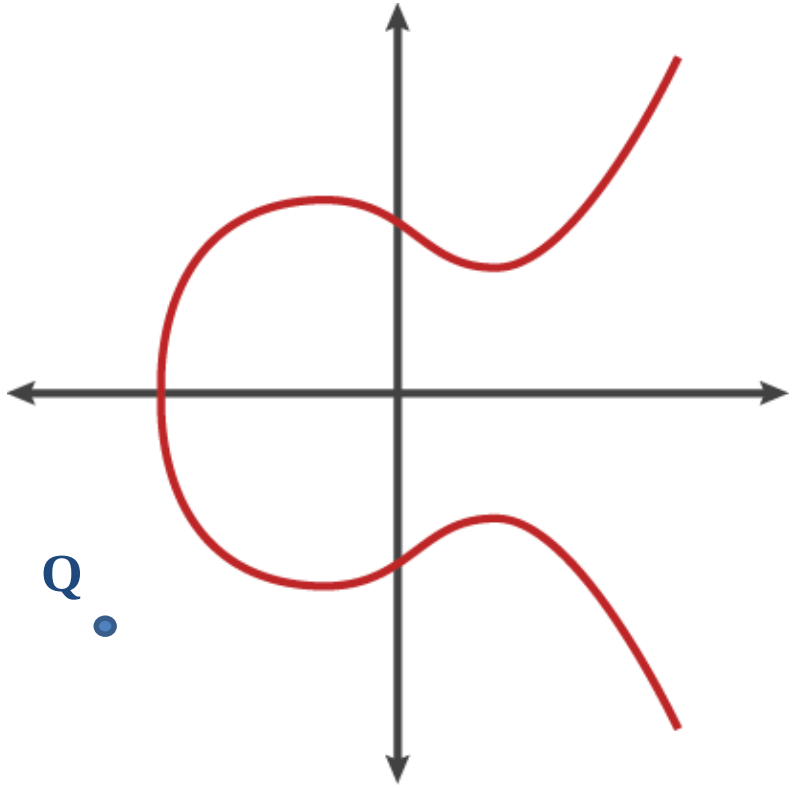




Invalid Curve Attack

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





- Attacker computes:

$$s_1 = s \bmod 5$$

Invalid Curve Attack

- Choose points of small co-prime order (5, 7, 11, …)
- Send to the server
- Compute:

$$s_1 = s \; mod \; 5$$

$$s_2 = s \; mod \; 7$$

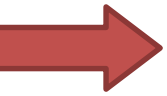
$$s_3 = s \; mod \; 11$$

$$s_4 = s \; mod \; 13$$

- Compute ***s*** with CRT

Overview

1. Bleichenbacher's Attack
   - XML Encryption
   - TLS
2. Invalid Curve Attack
   - TLS
   - Hardware Security Modules



Practical Attacks?

- Tibor Jager, Jörg Schwenk, Juraj Somorovsky: **Practical Invalid Curve Attacks on TLS-ECDH.** ESORICS 2015
- Analyzed 8 libraries
- 2 vulnerable
  - Bouncy Castle: 3300 TLS queries
  - Oracle JSSE: 17000 TLS queries

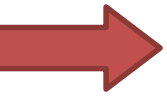
Impact

- 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

dec (C)

m

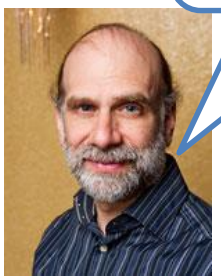
Attacker Model in HSM Scenarios

- 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



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