Practical Invalid Curve Attacks on TLS-ECDH

Tibor Jager, Jörg Schwenk, Juraj Somorovsky

Horst Görtz Institute for IT Security
Ruhr University Bochum

@jurajsomorovsky

About Me and Our Institute

- Security Researcher at:
  - Chair for Network and Data Security
    - Prof. Dr. Jörg Schwenk
    - Web Services, Single Sign-On, (Applied) Crypto, SSL, crypto currencies
    - Provable security, attacks and defenses
  - Horst Görtz Institute for IT-Security
    - Further topics: embedded security, malware, crypto…
  - Ruhr University Bochum
- Penetration tests, security analyses, workshops…

Recent years revealed many attacks on TLS…

- ESORICS 2004, Bard: Th bility of SSL to Chosen Plainte

2011 BEAST

- Eurocrypt 2002, Va laws Induced by CBC Padding—App EC, WTLS

2013/14 POODLE, Lucky13

- Crypto 1998, Bleichenbacher: Chosen Ciphertext Attacks Aga RSA Encryption Standard PKCS #1

2014 at USENIX Sec

Another “forgotten” attack

- Invalid curve attack
- Crypto **2000**, Biehl et al.: Differential fault attacks on elliptic curve cryptosystems
- Targets elliptic curves
  - Allows one to extract private keys

- Are current libraries vulnerable?

Overview

1. **Elliptic Curves**
2. **Invalid Curve Attacks**
3. **Application to TLS ECDH**
4. **Evaluation**
5. **Bonus Content**

Elliptic Curve (EC) Crypto

- Key exchange, signatures, PRNGs
- Many sites switching to EC
- Fast, secure

Algorithm	Signatures
256 bit ECDSA	9516 per sec
RSA 2048 bits	1000 per sec

- https://blog.cloudflare.com/ecdsa-the-digital-signature-algorithm-of-a-better-internet/

Elliptic Curve

- Set of points over a finite field

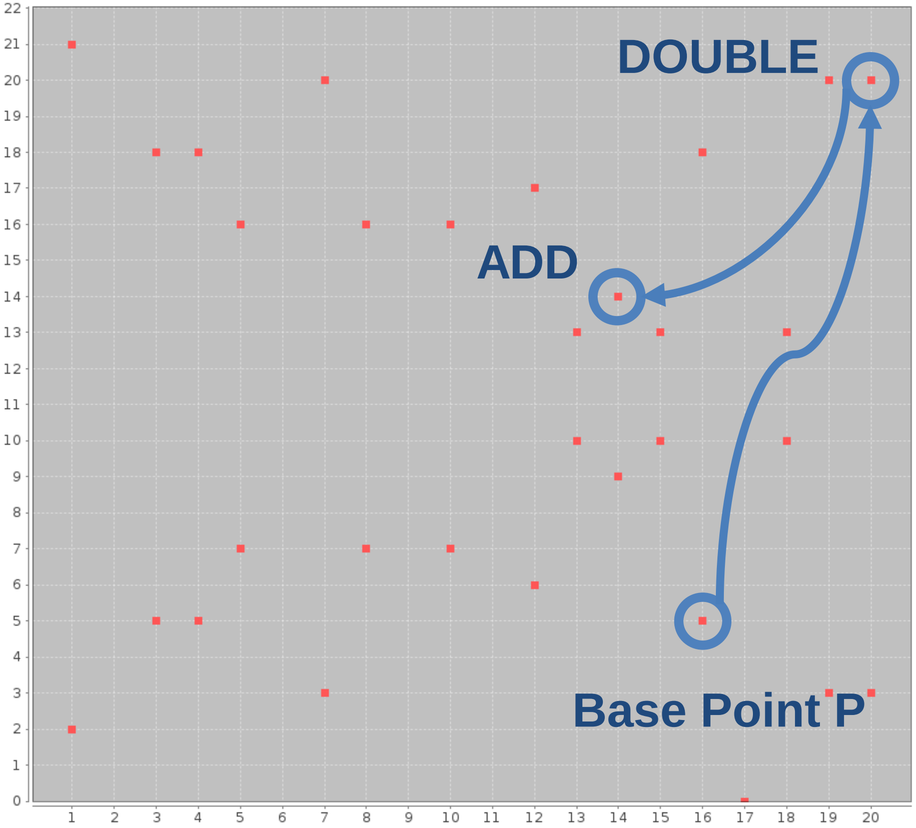
$$E:\ \ y^2 = x^3 + ax + b\ mod\ p$$

- Operations: ADD and DOUBLE
- Example:

$a = 9$

$b = 17$

$p = 23$



Elliptic Curve Diffie Hellman (ECDH)

Client

Secret q

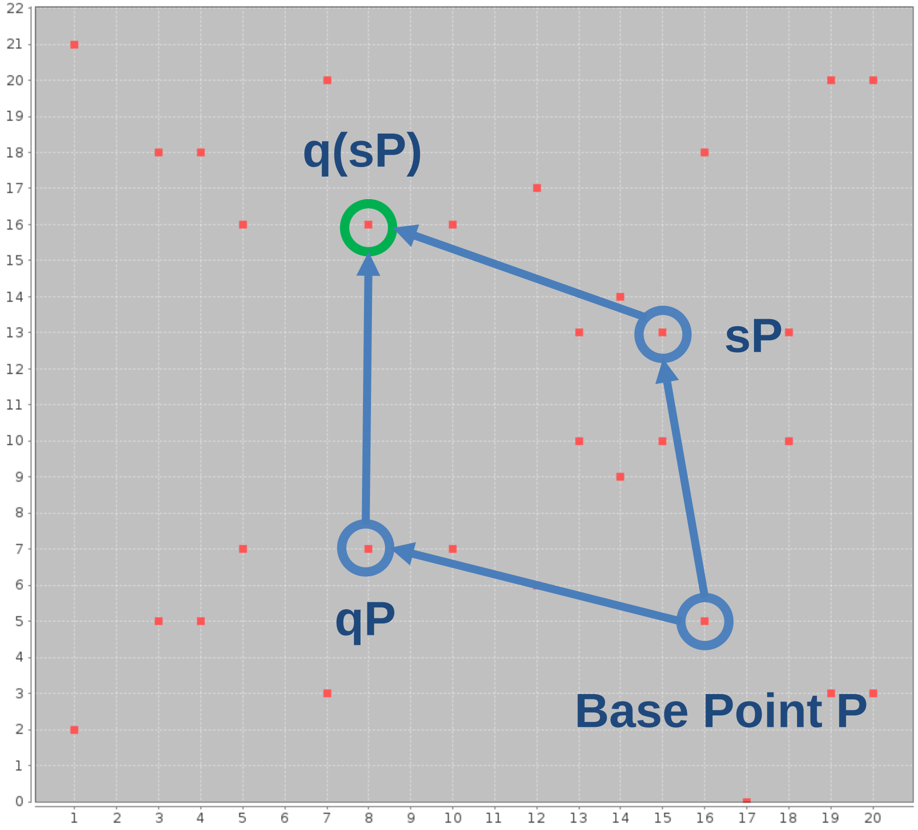
Server

Secret s

qP →

← sP

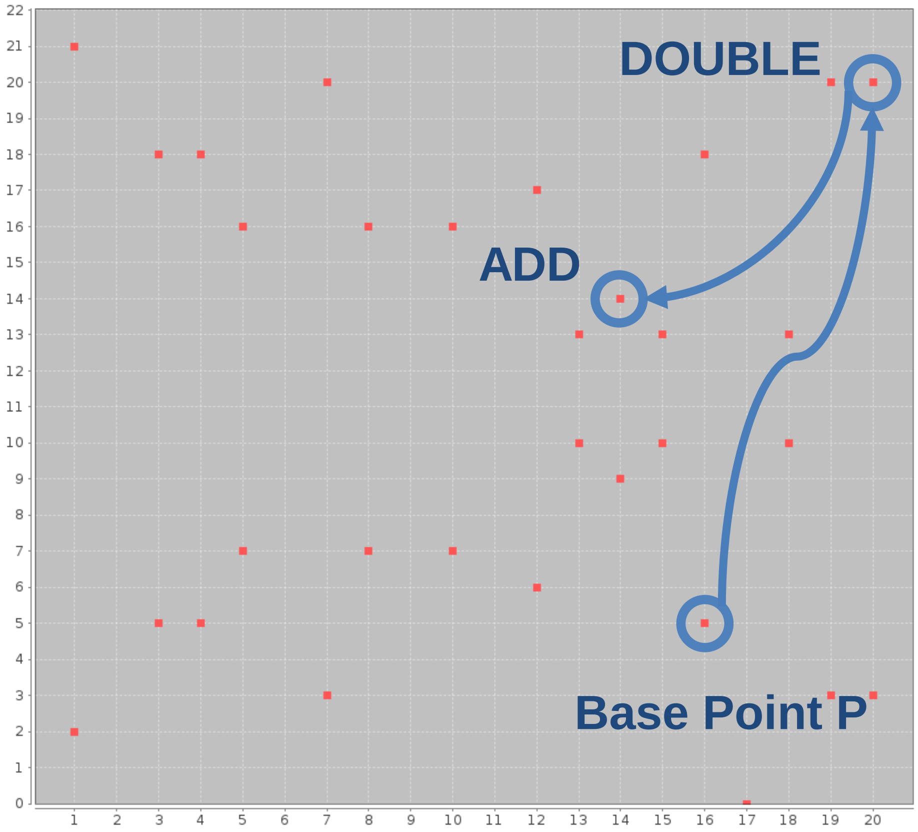
Shared secret: s(qP) = q(sP)



Elliptic Curves in Crypto

- Have to be chosen very carefully: **high** order
  - P -> ADD -> ADD -> … -> ADD -> P

order



- Predefined curves

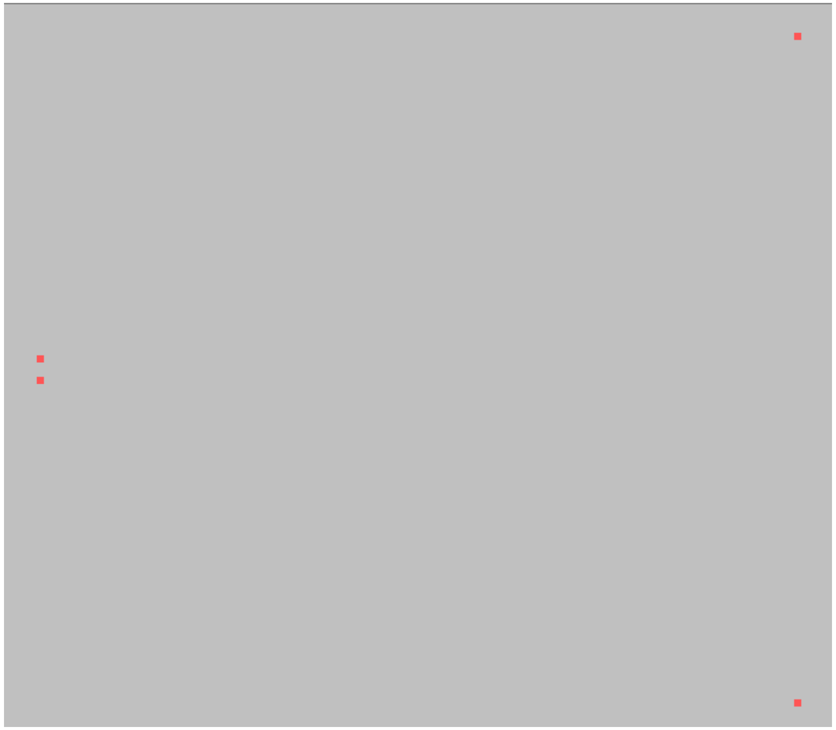
  > 256 bits

Overview

1. Elliptic Curves
2. Invalid Curve Attacks
3. Application to TLS ECDH
4. Evaluation
5. Bonus Content

Invalid Curve Attack

- What if we compute with a point P’ outside of curve E?
- P’ belongs to curve E’
- E’ can have a small order



- Example:
  - E’ with 256 bits
  - P’ generates 5 points

Invalid Curve Attack

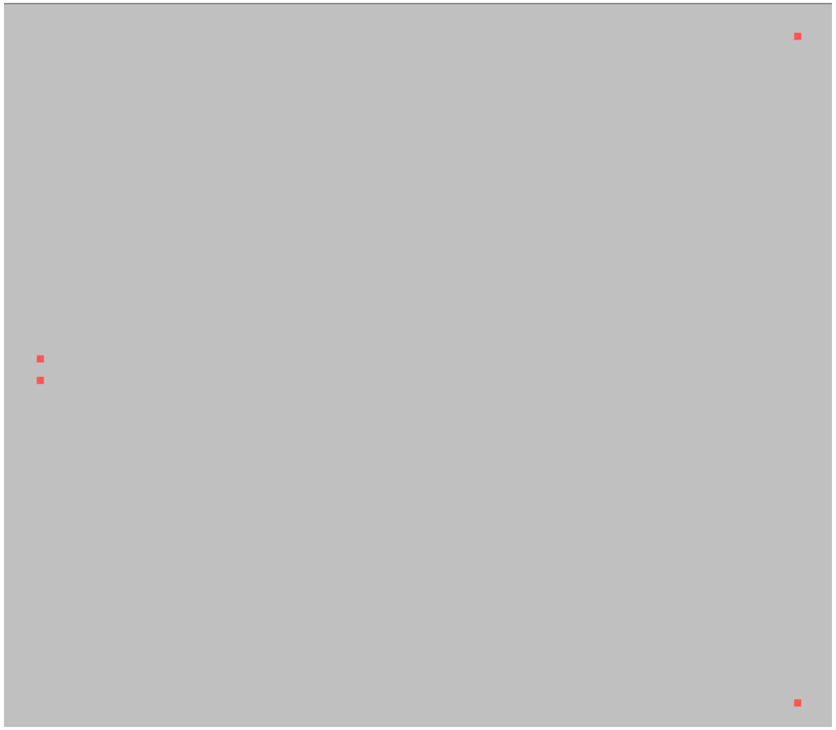
- What can we learn?
- Shared secret: sP’
  - Only 5 possible values!
- We can compute:

  $s_1 = s \; mod \; 5$

  $s_2 = s \; mod \; 7$

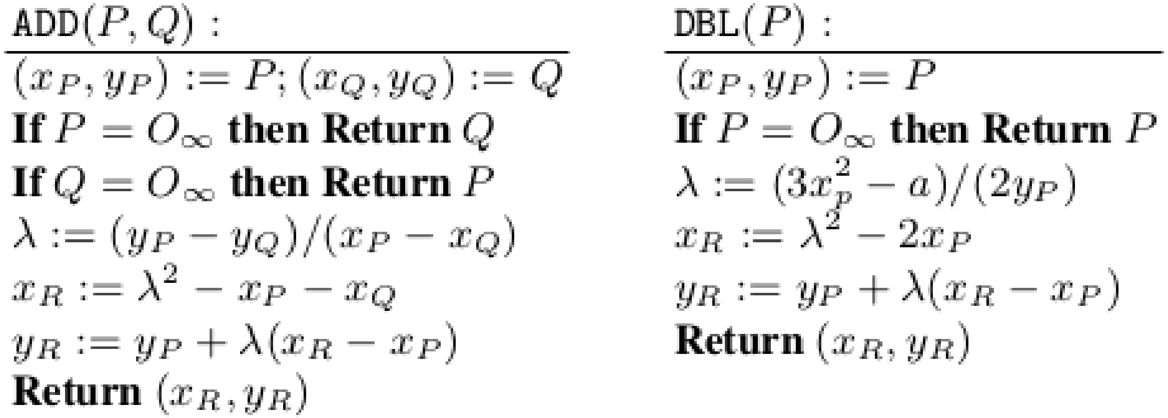
  $s_3 = s \; mod \; 11$

  $s_4 = s \; mod \; 13$
- Compute ***s*** with CRT



Invalid Curve Attack

- Possible if
  - – No point verification
  - – Test for shared secret possible
  - – *Simple* DOUBLE and ADD method
    - No sliding window etc.



```
ADD(P, Q) :
$(x_P, y_P) := P; (x_Q, y_Q) := Q$
**If** $P = O_\infty$ **then Return** $Q$
**If** $Q = O_\infty$ **then Return** $P$
$\lambda := (y_P - y_Q)/(x_P - x_Q)$
$x_R := \lambda^2 - x_P - x_Q$
$y_R := y_P + \lambda(x_R - x_P)$
**Return** $(x_R, y_R)$
```

```
DBL(P) :
$(x_P, y_P) := P$
**If** $P = O_\infty$ **then Return** $P$
$\lambda := (3x_P^2 - a)/(2y_P)$
$x_R := \lambda^2 - 2x_P$
$y_R := y_P + \lambda(x_R - x_P)$
**Return** $(x_R, y_R)$
```

- Curve ***b*** parameter not in the computation

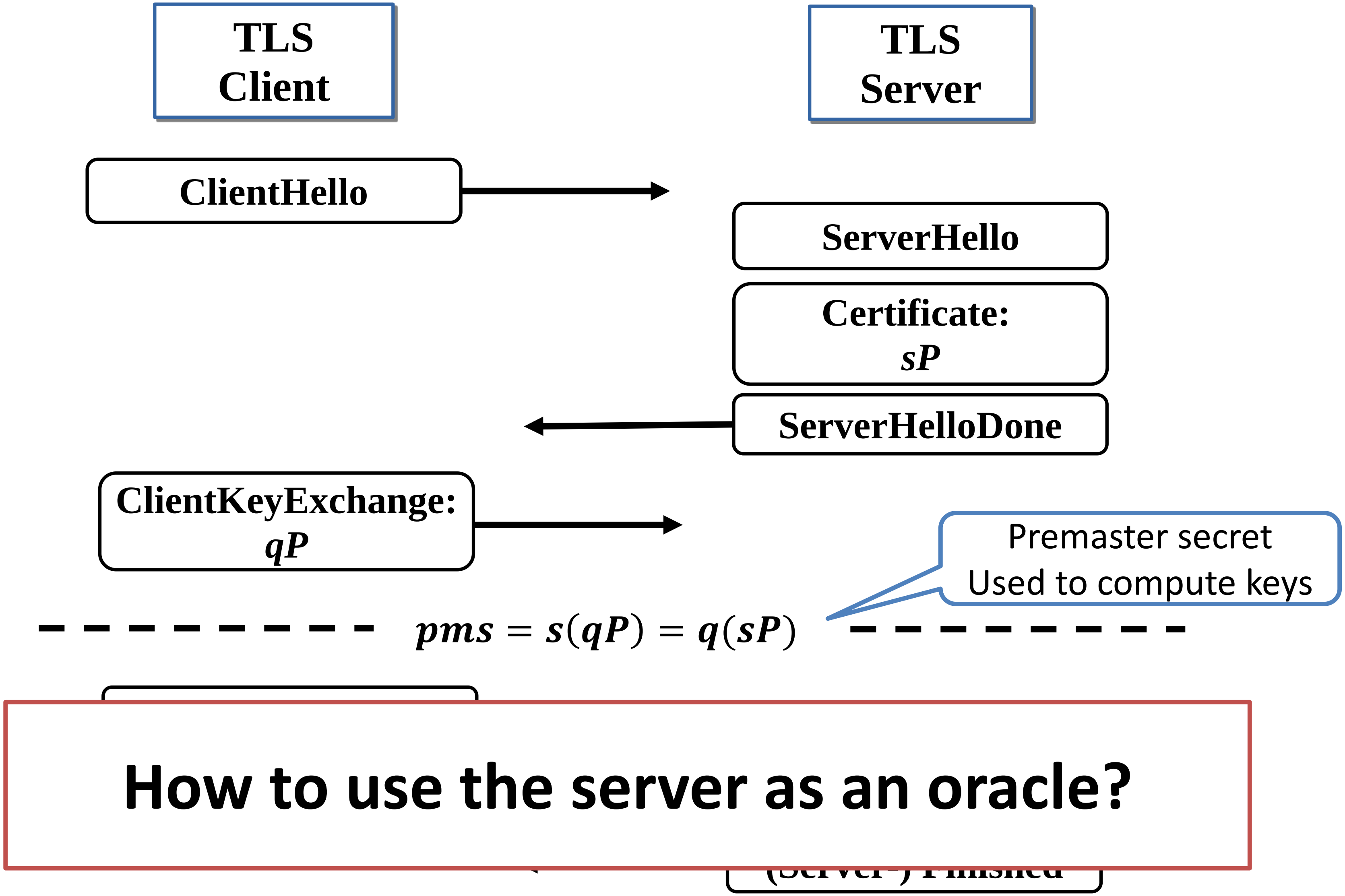
Overview

1. Elliptic Curves
2. Invalid Curve Attacks
3. Application to TLS ECDH
4. Evaluation
5. Bonus Content

Transport Layer Security (TLS)

- EC since 2006
- **Static** and ephemeral
- TLS server initialized with an EC certificate
    - Server has EC key

TLS ECDH



TLS Client

TLS Server

ClientHello

ServerHello

Certificate:
sP

ServerHelloDone

ClientKeyExchange:
qP

Premaster secret
Used to compute keys

$$pms = s(qP) = q(sP)$$

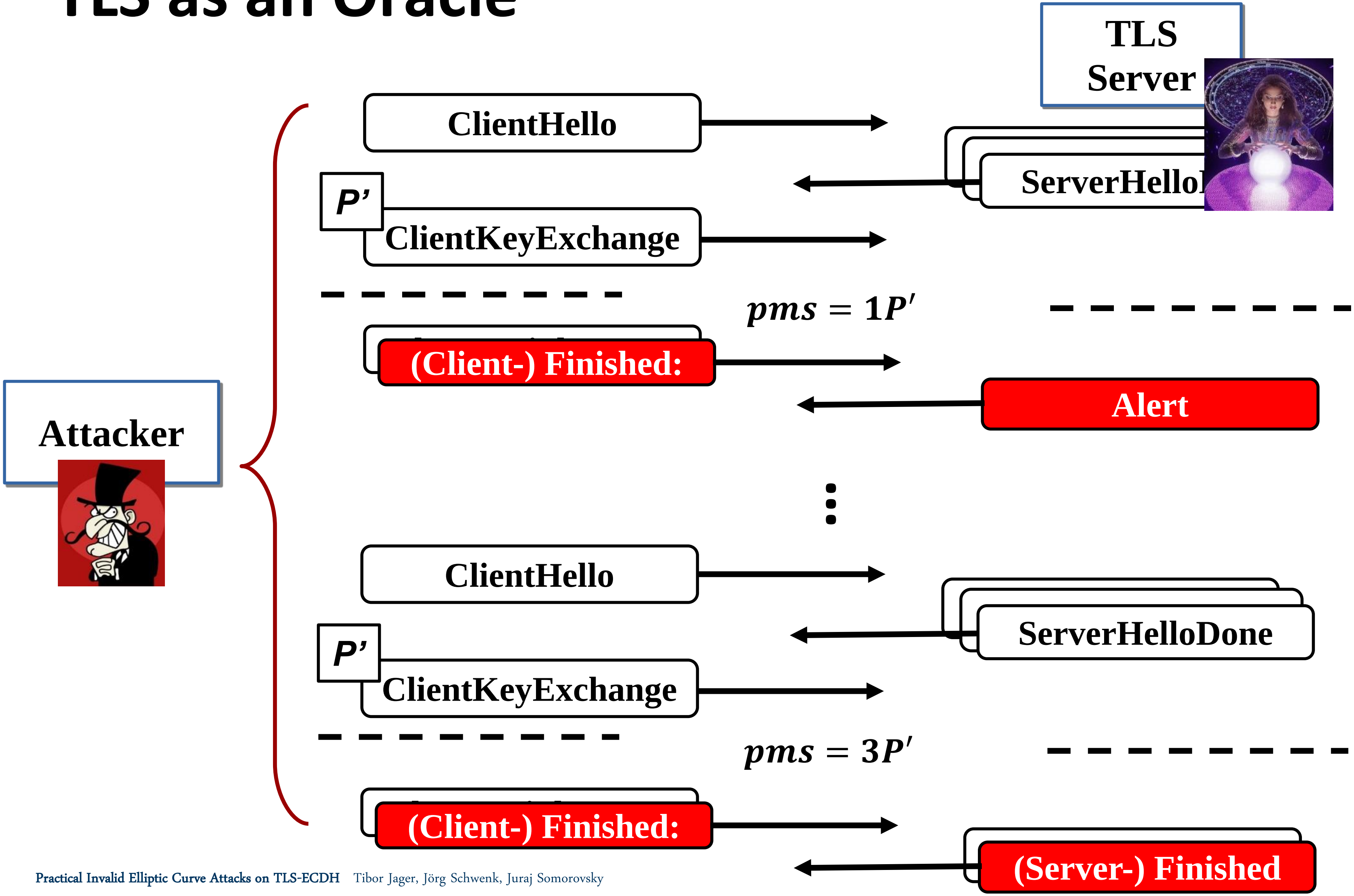
(Server-) Finished

How to use the server as an oracle?

TLS as an Oracle

- Idea:
  - Set $pms_1 = 1P', pms_2 = 2P', pms_3 = 3P', \ldots$
  - Execute TLS handshakes
  - If *pms* **correct**, ClientFinished accepted
- First described by Brumley et al.

TLS as an Oracle



TLS Server

Attacker

ClientHello

ServerHello

P’

ClientKeyExchange

$pms = 1P'$

(Client-) Finished:

Alert

⋮

ClientHello

ServerHelloDone

P’

ClientKeyExchange

$pms = 3P'$

(Client-) Finished:

(Server-) Finished

Invalid Curve Attack on TLS

1. Generate invalid points with order

   $p_i = 5, 7, 11, 13 \ldots$

2. Use oracle to get equations

   $s = s_i \; mod \; p_i$

3. Compute CRT to get secret key ***s***

Overview

1. Elliptic Curves
2. Invalid Curve Attacks
3. Application to TLS ECDH
4. Evaluation
5. Bonus Content

Evaluation

- 8 libraries
  - **Bouncy Castle v1.50**, Bouncy Castle v1.52, MatrixSSL, mbedTLS, OpenSSL, Java NSS Provider, **Oracle JSSE**, WolfSSL
- 2 vulnerable
- Practical test with NIST secp256r1
  - Most commonly used [Bos et al., 2013]

Evaluation: Bouncy Castle v1.50

- Vulnerable
  - 74 equations (oracle queries)
  - 3300 real server queries

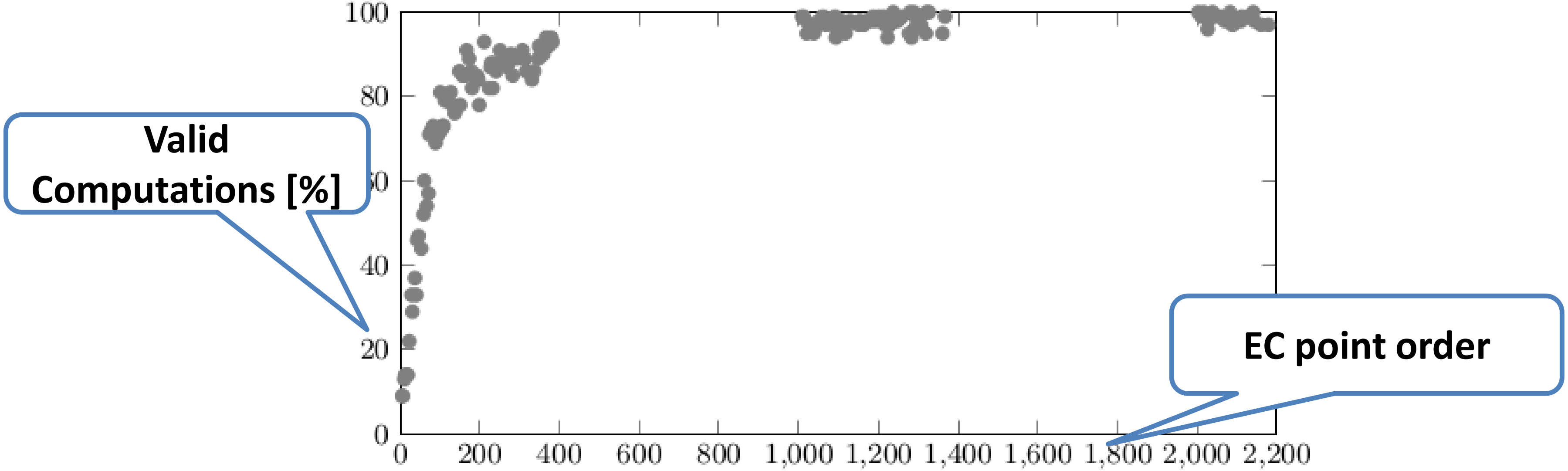
Evaluation: JSSE

- Java Secure Socket Extension (JSSE) server accepted invalid points

- However, the direct attack failed

Evaluation: JSSE

- Problem: invalid computation with some EC points



Valid Computations [%]

EC point order

- Not considered by Biehl et al.
- Attack possible:
  - 52 oracle queries, 17000 server requests

Impact

- Attacks extract server private keys
- Huge problem for Java servers using EC certificates
  - For example Apache Tomcat
  - Static ECDH enabled per default
- Key revocation

- Not only applicable to TLS
  - Also to other Java applications using EC

Overview

1. **Elliptic Curves**
2. **Invalid Curve Attacks**
3. **Application to TLS ECDH**
4. **Evaluation**
5. **Bonus Content**

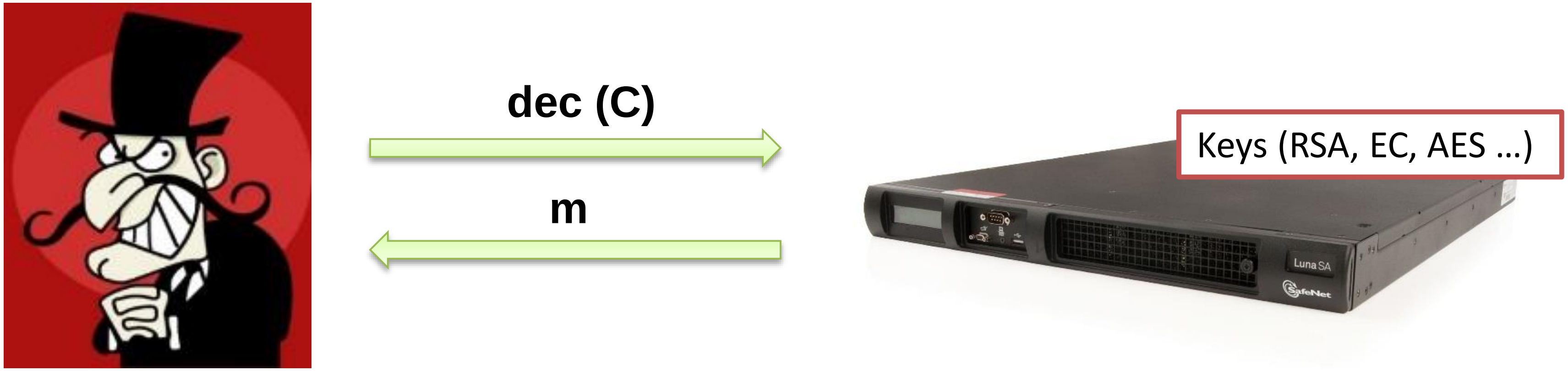
What’s next?

What’s next?

- Hardware Security Modules
- Devices for storage of crypto material

Attacker Model in HSM Scenarios

- Key never leaves HSMs



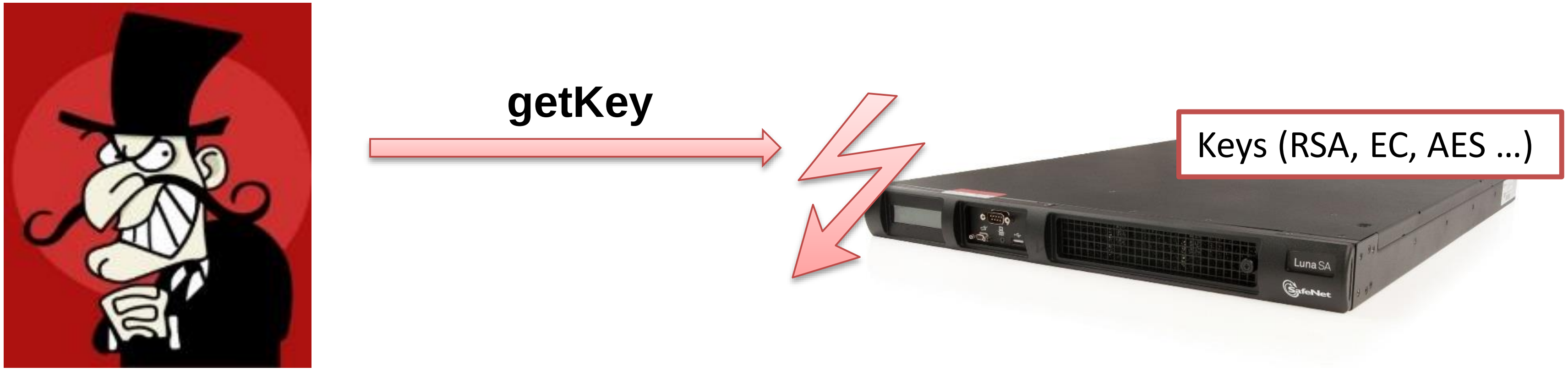
dec (C)

m

Keys (RSA, EC, AES …)

Attacker Model in HSM Scenarios

- Key never leaves HSMs

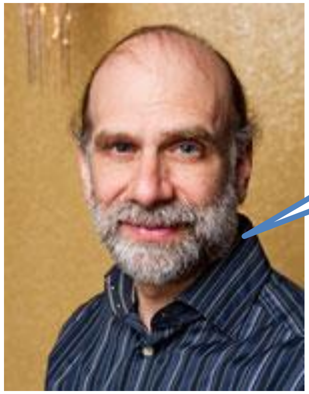


getKey

Keys (RSA, EC, AES …)

How about Invalid Curve Attacks?

- CVE-2015-6924
- Utimaco HSMs vulnerable
  - Analyzed together with Dennis Felsch
- < 100 queries to extract a 256 bit EC key



"Catastrophic is the right word. On the scale of 1 to 10, this is an 11.” [Heartbleed]

Conclusion

- Old attacks still applicable, we can learn a lot from them
- Bouncy Castle, JSSE and Utimaco broken
- More tools / analyses of crypto applications needed

- https://github.com/RUB-NDS/EccPlayground
- http://web-in-security.blogspot.de/