Encryption-based 2FA for Server-side Qualified Signature Creation

Christof Rath, christof.rath@iaik.tugraz.at

Institute for Applied Information Processing and Communications
Graz University of Technology, Austria

September 1, 2015
The Beginning: Smartcard-based Solutions

Rollout costs: \( \approx 10\€ \) per card + overheads

Smartcard-based solutions require:

- Smartcard reader
- Driver/middleware software installation

Usability issue! \( \rightarrow \) since 2003: 150k activations
The Beginning: Smartcard-based Solutions

Rollout costs: \( \approx 10\€ \text{ per card + overheads} \)

Smartcard-based solutions require:

- Smartcard reader
- Driver/middleware software installation

Usability issue! \( \rightarrow \) since 2003: 150k activations

Mitigation strategies: Online middleware (Java Applet)

- Modern browsers no longer support Java Applets
Alternative: Mobile Signatures

Signature creation is triggered by mobile phone:

- No additional hard-/software
- Reduced rollout costs (existing user devices)
- Usable with mobile computing
- Increased usability and acceptance
Mobile Signature Solutions

Mobile Signatures

SIM-based
- Private key on SIM
- Special SIM card
- Vendor lock-in
- Roaming
- Privacy issues

Server-side
- Only SMS required
- Ordinary SIM card
- Roaming
- Private key at server
- Privacy issues

Christof Rath, IAIK
September 1, 2015
Mobile Signature Solutions

- SIM-based
  - Private key on SIM
  - Special SIM card
  - Vendor lock-in
  - Roaming
  - Privacy issues

- Server-side
  - Only SMS required
  - Ordinary SIM card
  - Roaming
  - Private key at server
  - Privacy issues
Server-side Qualified Signature Creation

EU Signature Directive: Advanced electronic signature:
- Created using means that the signatory can maintain under their sole control
Server-side Qualified Signature Creation

EU Signature Directive: Advanced electronic signature:
- Created using means that the signatory can maintain under their sole control

Signing server stores private signing key in DB:
- Encrypted by HSM wrapping key
- Encrypted by user password
Server-side Qualified Signature Creation

EU Signature Directive: Advanced electronic signature:
- Created using means that the signatory can maintain under their sole control

Signing server stores private signing key in DB:
- Encrypted by HSM wrapping key
- Encrypted by user password

Private signing key can only be used:
- With user consent (password)
- Inside a specific HSM (wrapping key)
Server-side Qualified Signature Creation

To issue a signature the following steps are necessary:

1. Authenticate using phone number and password
2. Prepare decryption of private signing key
3. Send one-time password (OTP) via SMS
4. Verify OTP via web form
5. Decrypt and load private signing key into HSM
6. Sign document
7. Discard private signing key
Drawbacks of the SMS-based Authorization

Cost-factor:

- In Austria, about 10,000 signatures/day at 5¢/SMS ⇒ ≈ 180,000 €/year
- Could be payed by users → acceptance, feasibility
Drawbacks of the SMS-based Authorization

**Cost-factor:**
- In Austria, about 10,000 signatures/day
  at 5¢/SMS \(\Rightarrow\) \(\approx\) 180,000 €/year
- Could be payed by users \(\rightarrow\) acceptance, feasibility

**Security issues:**
- Malicious software can access SMS (on some platforms)
  e.g., Euro Grabber
- Transmission of SMS may not be encrypted (via GSM)
Server-side Qualified Signature Creation

To issue a signature the following steps are necessary:

1. Authenticate using phone number and password
2. Prepare decryption of private signing key
3. Send one-time password (OTP) via SMS
4. Verify OTP via web form
5. Decrypt and load private signing key into HSM
6. Sign document
7. Discard private signing key
Requirements of a New Solution

✓ Usability
✓ Platform independence
✓ Transaction binding
✓ Security
✓ Feasibility
✓ Cost efficiency
Encryption-based Two-factor Authentication

Requirements of a New Solution

- Usability
- Platform independence
- Transaction binding
- Security
- Feasibility
- Cost efficiency

- Billing
- Feasibility
- Reverse SMS
  - Usability
- Standard OTP
  - Transaction binding

Christof Rath, IAIK
September 1, 2015
Basic Concept

Proof of possession:
- Of a device (a pre-shared key)
- By decrypting an authorization code (OTP)
Basic Concept

Proof of possession:
- Of a device (a pre-shared key)
- By decrypting an authorization code (OTP)

Security features:
- Forward secrecy by changing IV
- Data transfer via HTTPS connections
- Hardware keystore/ARM TrustZone support possible
- Cryptographic transaction binding
Basic Concept – Pairing

To bind a mobile device to a user account:

1. S: Start Activation
   - Generate code
   - Send SMS

2. M: Continue Activation
   - Open OTP-App
   - Send activation code

3. Both
   - DH key exchange
   - Store shared secret
Basic Concept – Pairing

To bind a mobile device to a user account:

1. S: Start Activation
   - Generate code
   - Send SMS

2. M: Continue Activation
   - Open OTP-App
   - Send activation code

3. Both
   - DH key exchange
   - Store shared secret
Basic Concept – Pairing

To bind a mobile device to a user account:

1. S: Start Activation
   - Generate code
   - Send SMS

2. M: Continue Activation
   - Open OTP-App
   - Send activation code

3. Both
   - DH key exchange
   - Store shared secret
Encryption-based Two-factor Authentication

Basic Concept – Usage

To authenticate a user:
1. S: Verify password
2. M: Request OTP
3. S: Send encrypted OTP
4. M: Return decrypted OTP
5. S: Verify OTP
TanApp

The TanApp was a first prototype:

- HTML5/Javascript-based
  ✓ Platform independence
- Web clip icons to provide app style UI
  ✓ Usability, Acceptance
- Keys stored in HTML5 local storage
  ✗ Usability, Security

Christof Rath, IAIK
September 1, 2015
Encryption-based Two-factor Authentication

TanApp – Usage

To issue a signature:

1. S: Verify password
2. M: Request OTP
3. M: Receive encrypted OTP
4. M: Display decrypted OTP
5. S: Verify OTP
6. S: Issue signature
TanApp – Usage

To issue a signature:

1. S: Verify password
2. M: Request OTP
3. M: Receive encrypted OTP
4. M: Display decrypted OTP
5. S: Verify OTP
6. S: Issue signature
Encryption-based Two-factor Authentication

TanApp – Usage

To issue a signature:

1. S: Verify password
2. M: Request OTP
3. M: Receive encrypted OTP
4. M: Display decrypted OTP
5. S: Verify OTP
6. S: Issue signature
TanApp – User Test Findings

- HTML5 well suited for platform-independent development
- HTML5 local storage **unsuited** for key material
  - No hardware security features
  - Binding lost when clearing browser cache
- Usability issue: **Too similar** to SMS-based solution
The QR-TanApp was our second approach:

- Native Android app
  - ✓ Usability, Acceptance
  - X Platform independence requires multiple apps
- Support for hardware security features
  - ✓ Security
- Support for offline operation
  - ✓ Acceptance
Encryption-based Two-factor Authentication

QR-TanApp – Usage

To issue a signature:

1. S: Verify password
2. M: Take picture of QR-tag
3. M: Encrypt OTP
4. M: Send encrypted OTP
5. S: Verify OTP
6. S: Issue signature
Encryption-based Two-factor Authentication

QR-TanApp – Usage

To issue a signature:

1. S: Verify password
2. M: Take picture of QR-tag
3. M: Encrypt OTP
4. M: Send encrypted OTP
5. S: Verify OTP
6. S: Issue signature
QR-TanApp – Usage

To issue a signature:

1. S: Verify password
2. M: Take picture of QR-tag
3. M: Encrypt OTP
4. M: Send encrypted OTP
5. S: Verify OTP
6. S: Issue signature
QR-TanApp – Usage

To issue a signature:

1. S: Verify password
2. M: Take picture of QR-tag
3. M: Encrypt OTP
4. M: Send encrypted OTP
5. S: Verify OTP
6. S: Issue signature
Conclusion

- SMS-delivered OTPs must be regarded outdated
- We have proposed a novel authentication scheme
  - Higher level of security
    → strong cryptographic algorithms
    → hardware security support
    → completely avoid SMS
  - Enhanced usability
    → QR-codes prevent users from manually copying unintelligible OTPs