A closer look at SQRL

Agenda

• SQRL introduction
• Related work
• SQRL design details
• Research questions
• Research method
• Research findings
• Conclusion
A closer look at SQRL

**SQRL introduction:** trigger

**Secure Quick Reliable Login**

![Worst Passwords of 2013](image)

![Single Sign-On](image)

UvA-SNE-RP1 presentation
A closer look at SQRL

SQRL introduction: how it works

QR-scanning

QR-tapping

QR-clicking
A closer look at SQRL

SQRL introduction: design goals

- SSO
- 2FA
- out-of-band (OOB) authentication
- no secret(s) exchange
- anonymity
- no (additional) TTP
- low friction deployment
A closer look at SQRL

Related work: sSO

- Open standards
  - OpenID
  - TiQR
A closer look at SQRL

SQRL design details: crypto
Compromised ID?
- ID revocation support
- proves ID ownership
- uses additional keys
  - Lock (disable)
  - Unlock (enable/change)
A closer look at SQRL

SQRL design details: messages

client (laptop) → enrolment request → server

QR (URL enrolment + challenge) → (SQRl-app) smartphone
  - extract site’s domain name
  - generate site-specific private key
  - sign extracted QR data

response → authentication handler
  - signed extracted QR data
  - public key

OK → (laptop)

UvA-SNE-RP1 presentation
A closer look at SQRL

Research questions

• How does SQRL improve authentication security compared to related solutions?
  • What does SQRL offer to both parties?
  • What constraints must be met to guaranty this behaviour?

• What additional features are relevant to extend deployability?

• What attacks remain feasible and what countermeasures are to be considered?
A closer look at SQRL

Research method: attacks

- Attacks exploit vulnerabilities
- Causes of vulnerabilities
  - design errors
  - implementation errors
  - user mistakes
A closer look at SQRL

Research method: attacks

- Attacks exploit vulnerabilities
- Causes of vulnerabilities
  - design:
    - uses TLS
    - covers MiTM
    - covers eavesdropping
    - uses HMAC
    - no reverse operation
    - uses scrypt
    - covers brute-force
A closer look at SQRL

Research method: attacks

- Attacks exploit vulnerabilities
- Causes of vulnerabilities
  - design errors
  - implementation errors
    - no current (mature) app/server
A closer look at SQRL

Research method: attacks

- Attacks exploit vulnerabilities
- Causes of vulnerabilities
  - design errors
  - implementation errors
  - user mistakes
A closer look at SQRL

Research method: attacks

SQRL user interaction
- SQRL-app installation
- SQRL Identity password generation & use
- SQRL Master Key backup & restore
- SQRL (Un)lock Key backup & restore

SQRL design dependencies
- Responsible users
  - No malware installed
  - No shoulder surfing
  - Master Key safely stored (QR on paper)
  - (Un)lock Key safely stored (QR on paper)
A closer look at SQRL

Research findings: attacks

Malware needs to be addressed

Crypto in crypto-chip
A closer look at SQRL

Research findings: attacks

Malware needs to be addressed

Crypto in nfc-chip
A closer look at SQRL

Research findings: research question 2

- What additional features are relevant to extend deployability?

- Site-specific key-pairs

- E-mail
- Membership
- Registration
A closer look at SQRL

Research findings: research question 1

How does SQRL improve authentication security compared to related solutions?

- What does SQRL offer to both parties?
- What constraints must be met to guaranty this behaviour?

SSO

2FA

out-of-band (OOB) authentication

no secret(s) exchange

anonymity

no (additional) TTP

ID revocation facility
Related work: SSO-Open standards

- SURFnet
- OCRA (OATH Challenge Response Algorithm) RFC6287
A closer look at SQRL

Related work: SSO-Open standards

- OpenID Authentication 2.0
- Support of algorithms (not prescribed)
### A closer look at SQRL

**Related work:** SSO-Open standards

<table>
<thead>
<tr>
<th>Feature</th>
<th>TiQR</th>
<th>OpenID</th>
<th>SQRL</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSO</td>
<td>✓ (?)</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>2FA</td>
<td>✓</td>
<td>?</td>
<td>✓</td>
</tr>
<tr>
<td>OOB</td>
<td>✓</td>
<td>?</td>
<td>✓</td>
</tr>
<tr>
<td>No secret(s) exchange</td>
<td>X</td>
<td>?</td>
<td>✓</td>
</tr>
<tr>
<td>Anonymity</td>
<td>✓ (?)</td>
<td>?</td>
<td>✓</td>
</tr>
<tr>
<td>No (additional) TTP</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
</tr>
<tr>
<td>Low Friction Deploy</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>ID revocation</td>
<td>X</td>
<td>?</td>
<td>✓</td>
</tr>
</tbody>
</table>

UvA-SNE-RP1 presentation
How does SQRL improve authentication security compared to related solutions?

• What does SQRL offer to both parties?
• What constraints must be met to guaranty this behaviour?

User:
• SSO
• 2FA security
• anonymity
• no cross-site coupling of ID’s
• ID revocation support

Website:
• authenticated identity
• alongside alternative solutions
How does SQRL improve authentication security compared to related solutions?

• What does SQRL offer to both parties?
• What constraints must be met to guaranty this behaviour?

• HTTP over TLS
• user responsibility/awareness
A closer look at SQRL

Conclusion

SQRL is

• open
• no new technology
• a combination of Best Practices
• unique in its offered properties
• not operational yet

SQRL depends on

• responsible users

SQRL needs

• additional secret protection
A closer look at SQRL

Questions