Authentication: something you have

Daniel Bosk

Department of Information and Communication Systems,
Mid Sweden University, Sundsvall

5th November 2020

1 Something you have/know

- Essence of authentication
- Cryptographic approaches
- Anonymous Credentials

Remark: Essence of authentication

- Authentication is a challenge–response protocol.
- The verifier gives the prover a challenge.
- The prover responds to the verifier's challenge.

Example (Passwords)

- Verifier's challenge: 'what's the password?'
- Prover's response: the password.

Remark: Predictability

- The challenge is predictable.

Remark: Essence of authentication

- Authentication is a challenge–response protocol.
- The verifier gives the prover a challenge.
- The prover responds to the verifier's challenge.

Example (Passwords)

- Verifier's challenge: 'what's the password?'
- Prover's response: the password.

Remark: Predictability

- The challenge is predictable.

Remark: Essence of authentication

- Authentication is a challenge–response protocol.
- The verifier gives the prover a challenge.
- The prover responds to the verifier's challenge.

Example (Passwords)

- Verifier's challenge: 'what's the password?'
- Prover's response: the password.

Remark: Predictability

- The challenge is predictable.

Question

- Can we make the secrets more hard to guess?
- Can we have different challenges with different responses?

Solution

- *Freshness is about challenge and response.*
- *Password-based authentication: the same challenge all the time.*
- *Improvement: random challenge, hard-to-guess response.*
- *We can do this with crypto.*

Example (Schnorr's protocol[1])

- Prover's private key x, public key g^x.
- Prover wants to prove knowledge of x for $g^x = y$.
- Prover commits to randomness r, by sending $t = g^r$.
- Verifier replies with randomly chosen challenge c.
- After receiving c, prover replies with $s = r + cx$.
- Verifier accepts if $g^s = ty^c$.

[1] **Schnorr**.

Example (Schnorr's protocol[1])

- Prover's private key x, public key g^x.
- Prover wants to prove knowledge of x for $g^x = y$.
- Prover commits to randomness r, by sending $t = g^r$.
- Verifier replies with randomly chosen challenge c.
- After receiving c, prover replies with $s = r + cx$.
- Verifier accepts if $g^s = ty^c$.

[1] **Schnorr**.

Example (Schnorr's protocol[1])

- Prover's private key x, public key g^x.
- Prover wants to prove knowledge of x for $g^x = y$.
- Prover commits to randomness r, by sending $t = g^r$.
- Verifier replies with randomly chosen challenge c.
- After receiving c, prover replies with $s = r + cx$.
- Verifier accepts if $g^s = ty^c$.

[1] **Schnorr**.

Example (Schnorr's protocol[1])

- Prover's private key x, public key g^x.
- Prover wants to prove knowledge of x for $g^x = y$.
- Prover commits to randomness r, by sending $t = g^r$.
- Verifier replies with randomly chosen challenge c.
- After receiving c, prover replies with $s = r + cx$.
- Verifier accepts if $g^s = ty^c$.

[1] **Schnorr**.

Example (Schnorr's protocol[1])

- Prover's private key x, public key g^x.
- Prover wants to prove knowledge of x for $g^x = y$.
- Prover commits to randomness r, by sending $t = g^r$.
- Verifier replies with randomly chosen challenge c.
- After receiving c, prover replies with $s = r + cx$.
- Verifier accepts if $g^s = ty^c$.

[1] **Schnorr**.

Remark

- We need password managers anyway, might just as well use x.
- It's more common to have helping devices (smartphones).
- $y = g^x$ is public, no more leaked secrets from server hacks.

Remark

- We need password managers anyway, might just as well use x.
- It's more common to have helping devices (smartphones).
- $y = g^x$ is public, no more leaked secrets from server hacks.

Remark

- We need password managers anyway, might just as well use x.
- It's more common to have helping devices (smartphones).
- $y = g^x$ is public, no more leaked secrets from server hacks.

Idea

- Schnorr protocol is identity oriented.
- Generalize to other attributes.

Example (Can do . . .)

- equalities
- inequalities
- conjunctions
- disjunctions
- knowledge of signatures

Example (Age limits)

- Bob wants to go see a film in cinema.
- Alice who works there wants to have proof of his age.
- Bob has a certificate issued by someone Alice trusts.
- Bob doesn't want to show everything in the certificate.
- He proves that certificate says > 15.

Example (Age limits)

- Bob wants to go see a film in cinema.
- Alice who works there wants to have proof of his age.
- Bob has a certificate issued by someone Alice trusts.
- Bob doesn't want to show everything in the certificate.
- He proves that certificate says > 15.

Example (Anonymous Credentials[2])

- Makes heavy use of zero-knowledge proofs of knowledge.
- Can prove equalities, inequalities, knowledge, ownership, . . .
- Implementations and approaches:

Identity Mixer	`https://www.research.ibm.com/labs/zurich/idemix/`
U-Prove	`http://research.microsoft.com/en-us/projects/u-prove/`
AnonPass	`https://eprint.iacr.org/2013/317`
IRMA	`https://www.irmacard.org/irma/`

[2] J. Camenisch, A. Lehmann and G. Neven. 'Electronic Identities Need Private Credentials'. In: *IEEE Security Privacy* 10.1 (Jan. 2012), pp. 80–83. ISSN: 1540-7993. DOI: 10.1109/MSP.2012.7. URL: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?reload=true&arnumber=6142524.

[CLN12] J. Camenisch, A. Lehmann and G. Neven. 'Electronic Identities Need Private Credentials'. In: *IEEE Security Privacy* 10.1 (Jan. 2012), pp. 80–83. ISSN: 1540-7993. DOI: 10.1109/MSP.2012.7. URL: http://ieeexplore.ieee.org/xpl/articleDetails.jsp?reload=true&arnumber=6142524.