Re(AC)$^t$

Reputation and Anonymous Credentials for Access Control (t=2)

Project Group
AG Codes & Cryptography
Prof. Blömer
Authentication
Authentication (OpenID/OAuth)

Sign up in seconds

Sign up with Google Account

OR

Your name

Email

Password

Create My Account
Authentication (OpenID/OAuth)

Sign up in seconds

Sign up with Google Account

OR

Your name

Email

Password

Create My Account
Authentication (OpenID/OAuth)

Sign up in seconds

- Sign up with Google Account

OR

- Your name
- Email
- Password

Create My Account
Authentication (OpenID/OAuth)

Sign up in seconds

- **Sign up with Google Account**

Your name

Email

Password

Create My Account

Sign in - Google Accounts - Google Chrome

One account. All of Google.

myTestAccount@google.com

Sign in

Stay signed in

Forgot password?

One Google Account for everything Google

Sign in with a different account
Authentication (OpenID/OAuth)

Sign up in seconds

- Sign up with Google Account
- OR

Your name

Email

Password

Create My Account

Google

myTestAccount@gmail.com

- The web service would like to:

  - View your email address

By clicking Allow, you allow this app and Google to use your information in accordance with their respective terms of service and privacy policies. You can change this and other Account Permissions at any time.

[Deny] [Allow]
Hi, myTestAccount!
You have been authenticated
Let's go ➔

The web service would like to:

- View your email address

By clicking Allow, you allow this app and Google to use your information in accordance with their respective terms of service and privacy policies. You can change this and other Account Permissions at any time.

[Allow] [Deny]
Authentication (OpenID/OAuth)

• Google vouches for our identity
  • Service knows we have *authorization*

• Google shares specified data with the service (e.g., email address)
  • User has *control* over what is shared
  • Service is ensured *validity* of data

⇒ Authentication via trusted third party.
Authentication (OpenID/OAuth)

• Google vouches for our identity
  • Service knows we have *authorization*

• Google shares specified data with the service (e.g., email address)
  • User has *control* over what is shared
  • Service is ensured *validity* of data

⇒ Authentication via trusted third party.

... but has to be involved each time we authenticate

... as long as we trust Google about this
Authentication (OpenID/OAuth)

- Google vouches for our identity
  - Service knows we have *authorization*

- Google shares specified data with the service (e.g., email address)
  - User has *control* over what is shared
  - Service is ensured *validity* of data

⇒ Authentication via trusted third party.

... but has to be involved each time we authenticate

... as long as we trust Google about this

Goals:
- *Decentralize* authentication.
- *Eliminate* need for trust.
- Bonus: *Enable* user anonymity.
Authentication via Credentials
Authentication via Credentials
Authentication via **Credentials**
Authentication via Credentials

Sign up in seconds

Sign up via Credential

OR

Your name

Email

Password

Create My Account
Authentication via Credentials

Sign up in seconds

Sign up via Credential

OR

Your name

Email

Password

Create My Account
Authentication via Credentials

Sign up in seconds

Sign up via Credential

OR

Your name

Email

Password

Create My Account

Show (partial) Credential?

Reveal the following information?

- Real Name
- Email address
- Age

CANCEL OK
Authentication via Credentials

Sign up in seconds

Sign up via Credential

OR

Your name

Email

Password

Create My Account
Authentication via Credentials

Hi, myTestAccount!
You have been authenticated

Let's go →
Authentication via Credentials

Properties:

• **No third party** involved in authentication.
• **No trust** required in credential issuer.
• **No trust** required in authenticator.
• User has option to stay **anonymous**.
  • User chooses what information is revealed.
  • Even if issuer and authenticator collaborate.
The Project Group
Our goals

Anonymous Credential System

• Theory
  • Efficient construction based on recent papers
  • Want to add important features like revocation and identity escrow
  • Techniques: Digital signatures, encryption, zero-knowledge proofs

• Implementation
  • Java (preferably)
  • We have Java implementation of elliptic curve groups etc.
  • Need software support for interactive protocols
  • Need API, GUI, and scheme implementation

Reputation System (optional)
What we offer

• Meaningful project:
  • Will be used (SFB 901 testbed implementation)
  • Will be published (open source project)

• Challenges in both theory *and* practice
  • Set your own focus

• Insights into modern crypto
Prerequisites

• For theory: basic probability theory, basic algebra
• For practice: Java

• Nice to have: Introduction to Cryptography

• All „advanced basics“ will be covered in the seminar phase.
Thanks!