Using DCAF with Less Constrained Devices

draft-gerdes-ace-dcaf-examples

Stefanie Gerdes, Olaf Bergmann, Carsten Bormann
{gerdes | bergmann | cabo}@tzi.org

2015-07-22
Initial Trust Relationships

DTLS/TLS
CAM
SAM
DTLS
(PSK)
C
we have these
DTLS
(PSK)
S
Trust: The Complete Picture

![Diagram showing DTLS/TLS, CAM, SAM, DTLS (PSK), C, S, and relationships between them. The diagram highlights that we have these and we want this.](image-url)
Contact S’s Less Constrained Device for Authorization

Diagram:
- **CAM**
  - Access Request
  - Access Ticket
- **SAM**
- **C**
- **S**
Access Ticket

Face:
[server authorization info]
timestamp
[lifetime]

Client Information:
verifier (session key)
Access Ticket: Obtaining an Access Ticket

Access Request

CAM

SAM

Access Ticket

Face:
[server authorization info]
timestamp
[lifetime]

Client Information:
verifier (session key)
[client authorization info]
timestamp
[lifetime]
Access Ticket Parts

- Access Request
  - CAM
  - Access Ticket
  - SAM

Face:
- [server authorization info]
- timestamp
- [lifetime]

Client Information:
- verifier (session key)
- [client authorization info]
- timestamp
- [lifetime]
PSK Derivation

DTLS channel

psk_identity = Ticket Face

PSK = Verifier

derive PSK from Ticket Face and $K_{S,SAM}$
Authorized Requests Over DTLS

DTLS channel

use Client Info for authorization

CoAP traffic

use Ticket Face for authorization
Flexibility

- DCAF can be used as a simple protocol for secure transmission of DTLS pre-shared keys (implicit authorization).
- DCAF can additionally securely transmit authorization information to the server and/or the client.
- DCAF defines how combinations of actors work together.
- DCAF can be used as needed.
Authenticated Authorization on the Less Constrained Level

DTLS/TLS

CAM

DTLS
(PSK)

How do we get this?

S

DTLS
(PSK)

SAM
Example: Posting Temperature Values on a Blog

- Sarah has a blog where she posts details about her daily work.
- During a heat wave, she wants her temperature sensor to autonomously post the current room temperature of her office in the blog.
- She is registered with an OpenID Provider.
Use OpenID Connect with OAuth and DCAF

- Use OpenID Connect for authentication on the less constrained level.
- Use OAuth for authorization on the less constrained level.
- Use an OAuth Authorization Endpoint that can speak OAuth and DCAF and issues DCAF Tickets.
Flow Part 1: Authentication with OpenID Connect

Flow Diagram:

1. **UA** (User Agent) redirects to **CAM** (Resource Owner) with a `config` request.
2. CAM requests a `password` grant from **OP / SAM / Authorization Endpoint**.
3. **OP / SAM / Authorization Endpoint** responds with **Request credentials**.
4. **OP / SAM / Authorization Endpoint** verifies credentials.
5. **OP / SAM / Authorization Endpoint** asks for authz.
7. **OP / SAM / Token Endpoint** is used to generate a token request.

Path:
- Redirect from **UA**
- Password grant
- Request credentials
- Verify credentials
- Ask for authz
- Authz response
- Use code to generate token request
Flow Part 2: Authorization with OAuth/DCAF

UA

CAM (RP)

Token Request

Validate ID Token etc.

Refresh Request with Authz scope

Provide ticket to C and instruct it to access the blog

OP / SAM / Authorization Endpoint

TokenResponse: ID Token, Access Token, Refresh Token

OP / SAM / Token Endpoint

Validate Refresh Token and grant DCAF Ticket
Summary

- Use DCAF to provide simplified authentication and authorization to constrained devices.
- Enable constrained devices to enact the principals’ security objectives.
- Use less constrained nodes for the more difficult tasks.
- Use common protocols on the less constrained level that interoperate with DCAF.
Next Steps

- Describe more interaction examples (please feel free to contribute).
- Standardize DCAF as one of the ACE building blocks.