Enabling Decentralized Identifiers and Verifiable Credentials for Constrained IoT Devices

Vasilios A. Siris
Mobile Multimedia Laboratory
Athens University of Economics and Business, Greece
vsiris@aueb.gr

EU H2020 SOFIE: Secure Open Federation for Internet Everywhere
Contents

- Why constrained IoT (including intermittent or no connectivity)?
- Authorization with constrained IoT devices
- What are Decentralized Identifiers (DIDs)?
- What are Verifiable Credentials (VCs)?
- Putting it all together: How and why use DIDs & VCs for authorization in constrained IoT environments?
Why constrained IoT environments?

- Because many IoT devices are constrained in terms of:
  - processing and storage
  - network connectivity

Reducing usage also reduces power consumption & security threats

Scalability of IoT systems can be addressed by utilizing device-to-device & wireless multihop communication

Device-to-device technologies exist and are becoming more mature

New challenge: how to achieve trusted device-to-device communication
Authorization for IoT resources

• Client seeks to access an IoT Resource which may be disconnected from the Internet

<table>
<thead>
<tr>
<th>Client</th>
<th>IoT Resource</th>
</tr>
</thead>
<tbody>
<tr>
<td>D2D</td>
<td></td>
</tr>
<tr>
<td>request</td>
<td>authorization grant</td>
</tr>
<tr>
<td>Resource Owner</td>
<td>Client</td>
</tr>
</tbody>
</table>

vsiris@aueb.gr
Authorization for IoT resources

• Client seeks to access an IoT Resource which may be disconnected from the Internet

• Authorization Server (AS) handles requests on behalf of IoT Resource
  • OAuth 2.0 authorization framework being developed by IETF’s Authentication and Authorization for Constrained Environments (ACE) working group
  • Secure binding between AS-IoT Resource
  • Requires Resource Owner consent

Client seeks to access an IoT Resource which may be disconnected from the Internet.

Authorization Server (AS) handles requests on behalf of IoT Resource.

- OAuth 2.0 authorization framework being developed by IETF’s Authentication and Authorization for Constrained Environments (ACE) working group.
- Secure binding between AS-IoT Resource.
- Requires Resource Owner consent.

Diagram:
- Client requests authorization from the Authorization Server.
- Authorization Server issues an authorization grant.
- Resource Owner grants access.
- Client accesses IoT resource with authorization token.
Authorization for IoT resources

• Client seeks to access an IoT Resource which may be disconnected from the Internet
• Authorization Server (AS) handles requests on behalf of IoT Resource
  • OAuth 2.0 authorization framework being developed by IETF’s Authentication and Authorization for Constrained Environments (ACE) working group
  • Secure binding between AS-IoT Resource
  • Requires Resource Owner consent
• Client accesses IoT Resource with authorization token

vsiris@aueb.gr
What are Decentralized Identifiers

• Self-sovereign identifiers for individuals, organizations, things

Organization in control of identity

User in control of identity
What are Decentralized Identifiers

- Self-sovereign identifiers for individuals, organizations, things
- Decentralized, persistent, resolvable, cryptographically verifiable
- Registered in a blockchain, decentralized network, or off-ledger (ledger-agnostic)
- Standardized by W3C

```
did:sov:3k9dg356wdcj5gf2k9bw8kfg7a
```

Scheme Method

Organization in control of identity

User in control of identity

vsiris@aueb.gr
What are Decentralized Identifiers

- Self-sovereign identifiers for individuals, organizations, things
- Decentralized, persistent, resolvable, cryptographically verifiable
- Registered in a blockchain, decentralized network, or off-ledger (ledger-agnostic)
- Currently being specified by W3C
- did:sov:3k9dg356wdcj5gf2k9bw8kfg7a
DID methods

- Different DID methods: did:sov, did:btcr, did:v1, did:uport, ...
- CRUD for DIDs: Create, Read (Resolve), Update, Delete (Revoke)
- Resolution: DID → DID Document
  - Set of public keys, set of service endpoints, timestamps, proofs
DID methods

- Different DID methods: did:sov, did:btcr, did:v1, did:uport, ...
- CRUD for DIDs: Create, Read (Resolve), Update, Delete (Revoke)
- Resolution: DID → DID Document
  - Set of public keys, set of service endpoints, timestamps, proofs
What are Verifiable Credentials (VCs)

- Credential: A set of one or more claims
- W3C recommendation
- Requires framework for verifying identities
- Users (Holdes) positioned between credential Issuers and Verifiers
- Users receive and store VCs from Issuers through an agent that can be untrusted
- Users provide VCs to Verifiers through an agent that can be untrusted
- VCs are associated with users and not particular services
- Users control which VCs to use and when
  - DIDs allow users to own & control their identifiers
- Users may freely choose agents to help them manage and share their VCs
Usage of DIDs

• DID for constrained IoT Resource
  • Used to bind IoT device to Resource Owner
  • Defines authentication method for Resource Owner (DID owner/controller)

• DID for Authorization Server: used for authenticating AS

• DID for Client: used for authenticating client

• Resource owner can be offline

• Multiple DIDs for IoT Resource, Client, and AS
  • pairwise unique for each transaction
  • act as pseudonyms → improved privacy
Usage of DIDs

- DID for constrained IoT Resource
  - Used to bind IoT device to Resource Owner
  - Defines authentication method for Resource Owner (DID owner/controller)

- DID for Authorization Server: used for authenticating AS

- DID for Client: used for authenticating Client

- Resource owner can be offline

- Multiple DIDs for IoT Resource, Client, and AS
  - pairwise unique for each transaction
  - act as pseudonyms → improved privacy
Usage of DIDs

- DID for constrained IoT Resource
  - Used to bind IoT device to Resource Owner
  - Defines authentication method for Resource Owner (DID owner/controller)
- DID for Authorization Server: used for authenticating AS
- DID for Client: used for authenticating client
  - DID of Client added to authorization list at AS
    - Resource Owner can be offline
Usage of DIDs

- DID for constrained IoT Resource
  - Used to bind IoT device to Resource Owner
  - Defines authentication method for Resource Owner (DID owner/controller)
- DID for Authorization Server: used for authenticating AS
- DID for Client: used for authenticating client
- DID of Client added to authorization list at AS
  - Resource Owner can be offline
- Multiple DIDs for IoT Resource, Client, and AS
  - pairwise unique for each transaction
  - act as pseudonyms → improved privacy
Usage of Verifiable Credentials

- VCs for authorization grants
  - Required by Client to verify it has authorization
  - Client discloses only necessary information to Authorization Server

- VCs for Authorization Servers
  - Used by ASes to verify they handle authorization for an IoT resource
  - Revoking VC (or expired VC) allows Resource Owner to change AS
Usage of Verifiable Credentials

• VCs for authorization grants
  • Required by Client to verify it has authorization
  • Client discloses only necessary information to Authorization Server

• VCs for Authorization Servers
  • Used by ASes to verify they handle authorization for an IoT resource
  • Revoking VC (or expired VC) allows Resource Owner to change AS
Takeaways

• Why constrained IoT (including intermittent or no connectivity)?
  • constrained CPU/storage, power efficiency, security, scalability

• Authorization with constrained IoT devices
  • IETF OAuth 2.0; both IoT Resources and Clients can be constrained devices

• What are Decentralized Identifiers (DIDs)?
  • Self-sovereign identifiers (for individuals, organizations, things) that are decentralized, persistent, resolvable, cryptographically verifiable
  • In contrast: Public Key Infrastructure (PKI) is a centralized trust infrastructure

• What are Verifiable Credentials (VCs)?
  • A set of one or more claims issued by an Issuer to a Holder that can be verified by a Verifier
Takeaways (cont)

• Putting it all together: How and why use DIDs & VCs for authorization in constrained IoT environments?
  • Bind IoT Resources to Resource Owners
  • Authenticate Authorization Servers (ASes) and Clients
  • Pairwise unique DIDs (Clients, IoT Resources, ASes) for each transaction
  • VCs for authorization grants (Resource Owner to Client) and for verifying ASes handling requests (Resource Owner to AS)

• All above in a decentralized manner with users in control of their identities, credentials, and the information disclosed
Enabling Decentralized Identifiers and Verifiable Credentials for Constrained IoT Devices

Vasilios A. Siris
Mobile Multimedia Laboratory
Athens University of Economics and Business, Greece
vsiris@aueb.gr

EU H2020 SOFIE: Secure Open Federation for Internet Everywhere

Blockchain @ AUEB’s MMlab:
https://mm.aueb.gr/blockchains/

SOFIE H2020 Project:
https://www.sofie-iot.eu/