Lagutin, Dmitrij; Kortesniemi, Yki; Fotiou, Nikos; Siris, Vasilios A.

Enabling Decentralised Identifiers and Verifiable Credentials for Constrained IoT Devices using OAuth-based Delegation

Published in:
Proceedings of DISS 2019 – Workshop on Decentralized IoT Systems and Security

DOI:
10.14722/diss.2019.23005

Published: 01/01/2019

Document Version
Publisher's PDF, also known as Version of record

Please cite the original version:
Lagutin, D., Kortesniemi, Y., Fotiou, N., & Siris, V. A. (2019). Enabling Decentralised Identifiers and Verifiable Credentials for Constrained IoT Devices using OAuth-based Delegation. In Proceedings of DISS 2019 – Workshop on Decentralized IoT Systems and Security Internet Society.
https://doi.org/10.14722/diss.2019.23005

Enabling Decentralised Identifiers and Verifiable Credentials for Constrained IoT Devices using OAuth-based Delegation

Dmitrij Lagutin
Department of Communications and Networking
School of Electrical Engineering
Aalto University
dmitrij.lagutin@aalto.fi

Yki Kortesniemi
Department of Computer Science
School of Science
Aalto University
yki.kortesniemi@aalto.fi

Nikos Fotiou
and Vasilios A. Siris
Mobile Multimedia Laboratory
Department of Informatics
Athens University of Economics and Business
fotiou@aueb.gr, vsiris@aueb.gr

***Abstract*—Decentralised identifiers (DIDs) and verifiable credentials (VCs) are upcoming standards for self-sovereign privacy-preserving identifiers and authorisation, respectively. This focus on privacy can help improve many services and open up new business models, but using DIDs and VCs directly on constrained IoT devices can be problematic due to the management and resource overhead. This paper presents an OAuth-based method to delegate the processing and access policy management to the Authorisation Server thus allowing also systems with constrained IoT devices to benefit from DIDs and VCs.**

Keywords: Internet of Things, authorisation, decentralised identifiers, verifiable credentials, privacy, OAuth, ACE

I. Introduction

The problem of accessing IoT devices in a scalable, secure, and privacy-preserving manner can be a challenging one: there may be multiple stakeholders involved, and in most cases both the users and the IoT devices must be authenticated. At the same time, the privacy of users and their IoT devices is increasingly important, making solutions that emphasise the privacy and self-sovereignty of both highly relevant. These include the emerging authentication and authorisation solutions decentralised identifiers (DID) [1] and verifiable credentials (VC) [2] that can be used as privacy-enhancing identifiers and proofs of attributes for individuals and IoT devices [3], respectively. Their challenges, however, include implementing support for constrained (legacy) IoT devices that cannot process DIDs or VCs directly on the IoT device itself.

This is illustrated by the scenario where multiple printing services have installed their devices in publicly accessible places, e.g. in libraries (while this use case is about printing, the same principles can be applied to almost any kind of IoT device authorisation problem where there are multiple parties involved). One of the printing services has a contract with the local University, allowing people affiliated with the University

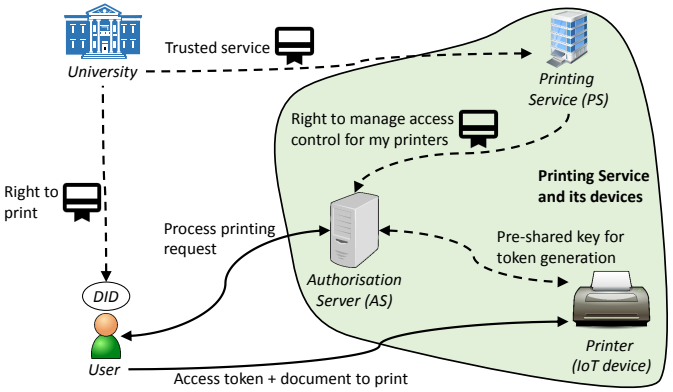

Fig. 1. An overview of the printing use case

to use all of its printers regardless of the location. A visiting Lecturer wants to use a printer before the lecture and the University has authorised Lecturer to print material as part of the lecturing agreement. Lecturer can use any of the printers of the trusted Printing Service (PS), but wants to guarantee that it is indeed one of their printers so as not to leak confidential information. At the same time, the printer wants to guarantee the material comes from someone authorised to print. However, the identity of Lecturer is not relevant to the printer or the PS as long as Lecturer has the right to print - and Lecturer would prefer a solution that maintains this privacy.

To implement the service, the Printing Service cannot assume that all users have identifiers issued by their host organisation, e.g., the visiting Lecturer does not have a University identifier or user account. Therefore the service instead relies on decentralised identifiers (DIDs) (which anyone can create for themselves) to authenticate the users and printers, and on verifiable credentials (VCs) issued by the host organisation (in this case: University) to prove the right to use the printing service, and by the Printing Service to prove the printer is part of the Printing Service, respectively. The Printing Service has a large number of printers and customers (universities, companies, individuals, etc.) to support, so to simplify the management of printers and rights to print and to enable even constrained printers that cannot process VCs or even DIDs

Workshop on Decentralized IoT Systems and Security (DISS) 2019
24 February 2019, San Diego, CA, USA
ISBN 1-891562-56-8
https://dx.doi.org/10.14722/diss.2019.23005
www.ndss-symposium.org

directly, the Printing Service builds on the OAuth framework [4] and its ACE extension (Authentication and Authorisation for Constrained Environments extension for OAuth) [5], and relies on the Authorisation Server (AS) to authenticate the printer for Lecturer and to process the authorisation request from Lecturer. The actors of the use case are shown in Fig. 1, where dashed lines denote the trust relationships between actors, and solid lines show the communication during the use case.

This paper presents a method for delegating the processing of DIDs and VCs from constrained IoT devices to the OAuth Authorisation Server by extending the ACE-OAuth flow, which significantly extends the number of devices able to utilise DIDs and VCs (an open-source source implementation is available at https://github.com/SOFIE-project). The paper also proposes a solution for local authentication, an area not addressed by ACE-OAuth. Finally, the paper analyses the resultant improvements and discusses opportunities for further optimising the protocol. The rest of the paper is organised as follows: Section II describes current authentication and authorisation solutions used with IoT devices and Section III describes Decentralised Identifiers and Verifiable Credentials. Section IV then proposes a method of integrating DIDs and VCs with ACE-OAuth, while the solution is analysed in Section V. Finally, Section VI suggests areas for future work, and Section VII concludes the paper.

II. Authentication and Authorisation with IoT

IoT devices are used for multiple use cases [6] that range from the transportation of fruits, which requires the continuous monitoring of climate sensor values used to control climate actuators, to seamlessly configuring new lighting devices and their authorisation policies, and to providing guests with temporary permissions in smart homes. This also means that the devices can and often do have very different capabilities. This section describes the key solutions currently used with IoT for authentication and authorisation, namely OAuth 2.0 and related technologies such as Authentication and Authorisation for Constrained Environments (ACE), OpenID Connect, and User-Managed Access (UMA) 2.0. All these protocols can be used with different authentication methods, and integrating them with DIDs and VCs provides more flexibility by adding support for those privacy-enhancing technologies.

A. OAuth 2.0

OAuth 2.0 [4] is an authorisation framework that enables a third-party application or *client* to obtain (limited) access to a protected resource hosted on a *resource server (RS)*. The client submits a resource access request to the *authorisation server (AS)* managing access to that resource. The AS, after obtaining explicit consent from the resource owner (either in advance or during the authorisation process), generates an access token that can be used by the client to prove to the RS that it is allowed to access the protected resource within the scope of the access token. Optionally, the AS can also authenticate the client on behalf of the resource server. This means that the RS only has to understand the access token as all other authentication and authorisation issues are handled by the AS. OAuth 2.0 also requires that the AS must authenticate the resource owner, but the way it is done (the *local authentication* method) is not defined by OAuth 2.0, therefore introducing DIDs provides a method for authenticating the involved parties.

B. Authentication and Authorization for Constrained Environments (ACE)

Constrained (IoT) devices are nodes with limited processing power, storage space, and transmission capacities, often battery-powered, and in many cases do not have a user interface. Due to the above constraints, deploying common security protocols, such as TLS and public/private key cryptography, to constrained devices may be difficult [6] (e.g. OAuth 2.0 requires that the communication between the client and the AS uses TLS). IETFs Authentication and Authorisation for Constrained Environments (ACE) extends OAuth 2.0 to constrained IoT environments by providing the necessary building blocks for adjusting OAuth 2.0 to IoT's requirements and a description of how these building blocks relate to the various IoT constraints (ACE-OAuth) [5]. A key contribution of ACE is the proof-of-possession access tokens (or PoP access tokens), where the access token may be bound to a cryptographic key (PoP key) that is used by the resource server to authenticate requests from a client; this allows authorisation over insecure links between the client and the protected resource. Another change is that with ACE the consent of the resource owner (giving a client access to a protected resource) can be provided either in a synchronous manner as in legacy OAuth 2.0, or it can be pre-configured as authorisation policies on the AS.

C. OpenID Connect

OpenID Connect 1.0 [7] is a simple identity layer on top of the OAuth 2.0 protocol, which allows a *relying party (RP)* to retrieve profile information about an end-user. This information is often the name of the user, but could also be used to carry other profile information e.g. the right to print with a specific printing service. The information is retrieved using OAuth 2.0 and it is expressed as a new type of token, known as the *ID token*. OpenID Connect has been studied also in the context of the IoT (e.g., [8]). In the printer use case, however, the same functionality is provided with DIDs and VCs.

D. UMA 2.0

One of the OAuth 2.0's weaknesses is that it is mainly used by resource owners for authorising their own applications to access their own resources, which also means that OAuth is not helpful when a resource wishes to share resources with a 3^{rd} party, referred to as the *Requesting party (RqP)*. The User-Managed Access (UMA) 2.0 specification [9] defines an extension to OAuth 2.0 that solves this problems by defining a centralised authorisation server where resource owners can register resources and define access control policies. Similarly, RqPs can use this AS for requesting access to a resource, even *if the resource owner is offline*. However, UMA 2.0 does not specify any particular access control definition mechanism. Furthermore, UMA 2.0 does not define any RqP authentication mechanism, but OpenID Connect is often used for this purpose (e.g., in WS02 identity server). For the printing use case UMA 2.0 is not suitable since it requires the RS (i.e., the device) to always be online and capable of securely exchanging access tickets with the AS, something a constrained device cannot guarantee.

E. Other related solutions

Capabilities-based access control (CBAC) is another related technology. A *CBAC token* defines the operations that a user is authorised to perform with a device. *CBAC tokens* are issued and digitally signed by a trusted party. CBAC has been studied in the context of the IoT by many research efforts, e.g. [10], [11]). The main drawback of these systems is that IoT devices are required to understand the business logic encoded in the tokens.

Eclipse Keti [12] on the other hand, is a token-based access control system which hides business logic from IoT devices. Using Keti, an IoT device may query an *access control service* if a user is allowed to perform a particular operation. The main drawback of Keti is that it requires IoT devices to be able to (securely) communicate with the access control services whenever the user wants to perform an operation.

III. Decentralised Identifiers and Verifiable Credentials

For decades, individuals on the Internet have successfully carried out transactions requiring identifiers, but as there have been no standard interoperable solutions for these identifiers, each service has been forced to create their own. More recently, some large companies such as Google, Facebook and Twitter, have introduced solutions known as *social logins* (implemented with OAuth and similar protocols), where the identifiers for that companys services can also be used to login to many other services. For the individual this reduces the number of identifiers they have to manage, but also means that individuals are dependent on the service providing the identifiers, and it also puts the service in a position to monitor the individuals' use of other services, which is detrimental to privacy. Clearly, there is a need for an identity solution that is *controlled by the individual* and provides *sufficient* privacy. In addition, to provide privacy preserving pseudonymity for humans, all addressable entities in a system including individuals, organisations and devices will need to support architectural level pseudonymity, otherwise the identity of e.g. an IoT device may give away the identity of the individual via means of attacks, which are not directly linked to the identifiers [13].

Currently, an identity technology receiving much attention are the decentralised identifiers (DIDs). A key aspect of DIDs is that they are designed not to be dependent on a central issuing party (Identity Provider or IdP) that creates and controls the identity. Instead, DIDs are managed by the identity owner (or a guardian on the owners behalf), an approach known as *self-sovereign identity* [14]. There are several different DID technologies in development [15], some of the most prominent being Sovrin [16], uPort [17] and Veres One [18]. These technologies started with similar but individual goals in mind, but lately many of them have adopted the approach and format of the W3C DID specification [1] being developed by the Decentralised Identity Foundation [19], thus rendering them more and more interoperable. The specification defines a DID as a random string (prefixed by "did" and a string indicating the particular DID technology), often derived from the public key used with the identity. The fact that a DID is a random string makes it privacy-promoting. Furthermore, if a new DID is allocated for every party one operates/communicates with, correlating one's activities with different parties would be significantly harder to achieve. This property can be further enhanced by replacing existing DIDs with new ones at suitable intervals, even after just a single use, if privacy is of paramount importance.

Yet DIDs alone do not suffice, as some means of distributing the related public keys, any later changes to the keys, or other identity-related information is required. To this end, many of the DID solutions rely on a distributed ledger (DLT) or a blockchain for public DIDs (e.g., used by organisations that want to be known), whereas for private DIDs (e.g. used by individuals) an application specific channels is used to distribute the information. Some DID technologies, e.g. Sovrin and Veres One, are launching their own DLTs based on the Byzantine fault tolerance (BFT) consensus [20], while others rely on existing blockchains (e.g. uPort is built on top of Ethereum [21]). All three example technologies originally intended to use DLTs/blockchains for distributing information about DIDs belonging to individuals and IoT devices in adition to organisations, but the emergence of the General Data Protection Regulation (GDPR) [22] in the EU and other similar changes have made storing personally identifiable information on a non-mutable platform such as a DLT/blockchain problematic. For this reason, Sovrin and Veres One have already excluded individuals DIDs from the ledger - and similar treatment may face the DIDs of IoT devices if they reveal personal information about their owner.

In many cases, there is a also need to associate machine verifiable properties to the identifier of an entity, e.g. in the printer use case the right to print for Lecturer. This is accomplished with Verifiable Credentials [2] which are analogous to traditional authorisation certificates. In a VC, the party issuing the credential (i.e. the *issuer*) claims that according the them, the party about which the credential is made, know as the *prover*, has the stated properties. These could be e.g. the person's name, date of birth etc. in the case of driver's license issued by the police, or it could be the prover's right to print using the University's quota. To rely on a credential to prove something, the prover also has to demonstrate that the credential was issued to them. This can be done e.g. by proving the possession of the private key corresponding to the public key used in the credential (if the credential format supports such information), or with a separate proof built on the credential. With a suitably created credential, a proof can also be used to only reveal *some of the attributes* of the credential (know as *selective disclosure*) or even prove that e.g. one is over a certain age without revealing the actual age attribute (a property known as *zero-knowledge proof*). However, the types of credentials supported vary depending on the DID technology, so the exact method of proving to whom the credential was issued is technology dependent.

IV. ACE-OAuth-based authentication and authorisation with DIDs and VCs

This section presents a method for delegating the processing of decentralised identifiers and verifiable credentials to the OAuth Authorisation Server. This allows also constrained IoT devices to benefit from better privacy and flexibility during the authorisation process.

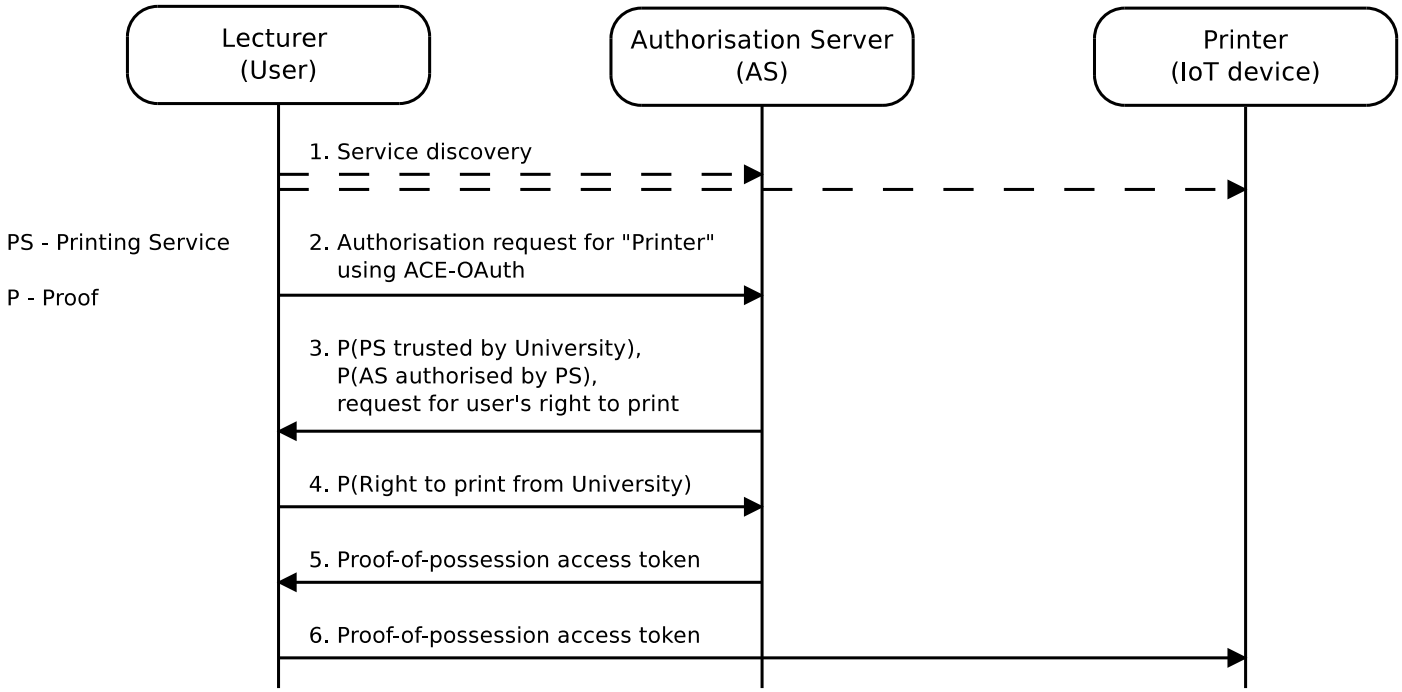

Fig. 2. IoT device authorisation flow using verifiable credentials and ACE-OAuth

In the printing use case, the actors include Lecturer that wants to use the printer (an IoT device), the Printing Service (PS) that owns the printer, the University that has issued Lecturer a verifiable credential (VC) for printing, the printer, and the authorisation server (AS) that handles authentication and authorisation on behalf of the printer as was shown in Figure 1. The AS authorises users based on access control policies defined by the PS. They can come in many forms, e.g. as a combination of information previously stored on the AS by the PS (e.g. the University is allowed to issue credentials for printing) and Lecturer's credential that was issued by the University. Together, they form a proof that Lecturer is allowed to print with the PS's printers[1]. Furthermore, Lecturer trusts any printing service accredited by the University and only wants to access printers owned by that service, so mutual authentication between Lecturer and the printer is necessary. In order for the AS to prove the printer's trustworthiness, two sets of proofs (based on the VCs) will be presented to Lecturer. First, the proof stating that PS is trusted by the University to offer printing services, and second, the proof stating that the AS is authorised to authenticate, handle authorisation requests, and issue tokens to users on behalf of that particular printer.

The communication flow between Lecturer, AS, and the printer is presented in Fig. 2 and described below.

1) Lecturer (user) discovers the printer and the related authorisation server (AS) using some means, such as local service discovery using Bluetooth or Wi-Fi Direct.
2) Lecturer indicates the desire to use the printer, and requests from the AS a proof that a) PS is trusted by the university, and b) that AS is authorised by the PS to handle printer access requests.
3) The AS uses the corresponding credentials to generate the requested proof. The proof along with the AS's request for Lecturer to provide a proof of the right to print, is sent to Lecturer.
4) Lecturer verifies AS's proof and if everything is in order (PS is trusted by the University and AS is trusted by PS), sends to AS proof of the right to print created using the credential issued by the University.
5) The AS verifies Lecturer's proof and sends back a proof-of-possession (PoP) access token.
6) Lecturer proceeds to communicate with the printer using the access token.

With this arrangement, the printer can utilise DIDs and VCs even if the printer itself does not understand them, and Lecturer only has to reveal the minimum amount of information (that they have a right to print) while being guaranteed only to print to a trusted printer. Furthermore, the printer does not need to communicate with AS during the authorisation process.

An implementation of the solution is available as open source at: https://github.com/SOFIE-project. There, an OAuth2 php server [23] has been modified to accept DID-based user authentication using Hyperledger Indy [24] and its SDK [25] to emulate our printing use case. Hyperledger Indy, on which Sovrin is based, is a decentralised ledger-based identity system that provides tools for creating, managing, and using digital identities. The use case involves the following phases:

Network setup. During this phase, which is executed only once, a "pool" of Indy nodes is created. The configuration file of this pool includes the DID of a "Steward" node that is responsible for writing information to the Indy ledger. This DID is considered to be well-known.

[1]All actors (Lecturer, University, Printing Service, Authorisation Server and Printer) also utilise decentralised identifiers to which the verifiable credentials are tied.

University setup. The University first creates a DID known as the "Trust Anchor", which is used for signing publishing requests sent to the Steward. Then it creates the credential definition scheme and publishes it to the Hyperledger Indy ledger (through the Steward). These actions are executed once.

Afterwards, each Lecturer creates a DID and sends a "credential request" to the University, which responds with the corresponding credential containing "right to print".

Access request. The Lecturer requests authorisation to access the printing service using the following steps:

1) The Lecturer makes an HTTP request to the OAuth2 server specifying a "grant_type=DIDs".
2) The OAuth2 server generates a proof request, asking the Lecturer to prove that he holds the "right to print", issued by the university. This request includes among other fields a nonce.
3) The lecturer generates a proof based on credential and repeats the request by including now the proof in the payload.
4) The OAuth protocol proceeds as usual.

V. Discussion

Utilising DIDs and VCs for IoT authorisation and authentication offers several advantages over how traditional X.509 certificates [26] and Public Key Infrastructures (PKIs) are currently used (theoretically all the improvements could also be implemented with them as well, but due to e.g. the significantly higher cost of X.509 certification and the number of certificates required that would be highly impractical if not impossible). Traditional certificates are designed to be semi-permanent and human-readable: the user receives their certificate once and uses it in several situations. The certificate usually also contains much (unnecessary) information about the user including their real identity, and the user must reveal all attributes of the certificate when using it. This leads to a high cost of issuing the certificates (the user's real identity must be verified, usually by manual means) and serious privacy issues since user's activities can be easily tracked from service to service by multiple parties when using certificates.

The DIDs and VCs are designed to allow more fine-grained, machine-readable, and short-lived credentials thus improving privacy and reducing the costs of issuing the credentials. Traditionally, for Lecturer to access the printer service at the University requires an IT account at the university, and university policies often restricts or even prohibit issuance of such accounts to outsiders. With the proposed solution, Lecturer would receive a credential that is only valid for printing during the visit. If Lecturer uses the printing service on another day (e.g. in relation to another lecture), it would be with a different DID and associated VC, therefore the AS, the printing service, or the printer will not be able to track Lecturer.

While there are proposals to allow X.509 certificates to support zero knowledge properties [27], certain DID solutions such as Sovrin contain built-in support for zero knowledge proofs [28], which in turn further improve privacy by allowing the users to prove properties about themselves without disclosing their credentials.

Currently there exist multiple federated identity management solutions such as single sign-on systems and eduroam [29]. However, they usually rely on their own non-standard identity management solutions and allow only certain members to participate in the first place (it is not easy to interconnect public organisations located in different countries or for non-education institutions to join eduroam). DIDs and VCs are open standards allowing easy deployment and adoption, thereby allowing any organisations to co-operate with each other with a low barrier of entry.

The proposed solution is compatible with and solves issues not covered by OAuth and its extensions. While OAuth defines the format of message exchange in the authorisation flow, it leaves the detail of how local authentication is performed to be extension or application specific. This includes the authentication of the resource owner to the AS, and the mutual authentication of the client and the AS, the latter of which is required e.g. in the ACE framework. Furthermore, with the proposed solution an OAuth resource server and AS can be decoupled, as opposed to what OAuth implicitly requires. Therefore the existing OAuth infrastructure can be adapted in a straightforward manner to utilise the proposed solution based on DIDs and VCs.

VI. Future work

An interesting direction for the future work is to identify how the unique features of DIDs and VCs can be utilised to benefit IoT use cases, such as those discussed in [6], while interoperating with existing authentication and authorisation frameworks in a scalable manner. The decentralised features of DIDs and the fine-grained user controllable information disclosure of VCs can be important in this direction. Similarly, W3C is working on describing verifiable credentials in JSON Web Token (JWT) [30] format, used by OpenID Connect among others. This would allow expressing VCs as OpenID *ID tokens*, thereby facilitating easy updating of the existing OpenID infrastructure to utilise DIDs and VCs.

Using DIDs for IoT devices can allow trusted AS discovery, by including AS in the IoT devices DID object (if the device has a publicly accessible DID object used to describe DID-related information) or by using other means of reliably distributing DID-related information to the intended parties. This type of IoT device authentication addresses the unauthorised AS information issue (due to the channel between the client and IoT device being initially insecure) defined in the ACE framework.

Finally, since OAuth utilises the TLS protocol for security, it would be possible to simplify the proposed communication flow by submitting the necessary proofs from steps 3-4 already during the TLS handshake process. Using the Server Name Indication (SNI) [31] TLS extension, the user can specify the identity of the device they wants to access already in the TLS "client hello" message, allowing the AS to present the required proof in its TLS reply. Afterwards, the AS may request the client's proof using the "client certificate request" message. Therefore, the TLS handshake can replace steps 2-4 in the flow diagram, significantly simplifying the communication. The proofs can be embedded into an X.509-encoded certificate – currently used by TLS [32], or alternatively proofs can be

expressed using the TLS "client_ / server_certificate_type" extensions [33]. As a downside of this optimisation, an eavesdropper monitoring the network traffic between the user and AS could determine which device the user wants to access. This problem can be avoided if the user is able to discover AS's public key by some means, e.g., during the service discovery or from DNS. In that case, the Encrypted Server Name Indication extension for TLS 1.3 can be used [34].

VII. Conclusion

This paper has proposed a method of using the ACE-OAuth framework to delegate the processing of decentralised identifiers (DIDs) and verifiable credentials (VCs) to the OAuth Authorisation Server (AS). This method allows systems with devices that support ACE-OAuth but lack the resources to process DIDs or VCs directly (e.g. constrained IoT devices) to benefit from the privacy enhancing properties of DIDs and VCs. Decentralised identifiers and verifiable credentials also offer a more flexible way to authorise users and manage access control policies, which has been identified as an important direction for the future work. An implementation of the proposed solution is available as open-source at https://github.com/SOFIE-project.

Acknowledgment

This research has been undertaken in the context of projects SOFIE - Secure Open Federation for Internet Everywhere (European Union's Horizon 2020 research and innovation programme, grant agreement No. 779984), and TrustNet - Trust Network for Distributed Personal Data Management (Business Finland grant No. 3387/31/2017).

References

[1] D. Reed et al., "Decentralized Identifiers (DIDs) v0.11, Data Model and Syntaxes for Decentralized Identifiers (DIDs)," Draft Community Group Report, 29 November 2018. Available at: https://w3c-ccg.github.io/did-spec/ (Accessed December 7th, 2018).

[2] M. Sporny, D. C. Burnett, D. Longley, and G. Kellogg, "Verifiable Credentials Data Model 1.0 - Expressing verifiable information on the Web," W3C Editor's Draft, 7 December 2018. Available at: https://w3c.github.io/vc-data-model (Accessed December 7th 2018).

[3] Y. Kortesniemi, D. Lagutin, T. Elo, and N. Fotiou, "Improving the Privacy of IoT with Decentralised Identifiers (DIDs)," Journal of Computer Networks and Communications, vol. 2019, Article ID 8706760, 10 pages, 2019. https://doi.org/10.1155/2019/8706760.

[4] D. Hardt, "The OAuth 2.0 Authorization Framework," IETF RFC 6749, October 2012.

[5] L. Seitz et al., "Authentication and Authorization for Constrained Environments (ACE) using the OAuth 2.0 Framework (ACE-OAuth)," IETF Internet Draft, November 2018.

[6] L. Seitz et al., "Use Cases for Authentication and Authorization in Constrained Environments," IETF RFC 7744, January 2016.

[7] N. Sakimura et al., "OpenID Connect Core 1.0," The OpenID Foundation, November 2014.

[8] A. Blazquez, V. Tsiatsis, K. Vandikas, "Performance evaluation of OpenID Connect for an IoT information marketplace," in Proceedings of the 81st Vehicular Technology Conference (VTC Spring), 2015.

[9] E. Maler et al., "User-Managed Access (UMA) 2.0, Kantara Initiative," August 2016.

[10] L. Seitz, G. Selander, and C. Gehrmann, "Authorization framework for the internet-of-Things," in Proceedings of 14th IEEE International Symposium and Workshops on World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2013.

[11] S. Gusmeroli, S. Piccione, and D. Rotondi, "A capability-based security approach to manage access control in the internet of things," Mathematical and Computer Modelling, vol. 58, no. 5, pp. 1189 1205, 2013.

[12] Eclipse Keti. Available at: https://projects.eclipse.org/proposals/eclipseketi. (Accessed December 14th 2018).

[13] J. R. Rao et al., Can Pseudonymity Really Guarantee Privacy?, in Proceedings of the 9th USENIX Security Symposium Denver, Colorado, USA, August 2000.

[14] C. Allen, "The Path to Self-Sovereign Identity," April 2016. Available at: http://www.lifewithalacrity.com/2016/04/the-path-to-self-sovereign-identity.html (Accessed December 18th, 2018).

[15] "Blockchain and Identity: Projects/companies working on blockchain and identity," Available at: https://github.com/peacekeeper/blockchain-identity (Accessed November 7th, 2018).

[16] Sovrin Foundation, "Identity For All," Available at: https://sovrin.org/ (Accessed December 18th, 2018).

[17] Uport, "Open Identity System for the Decentralized Web," Available at: https://www.uport.me (Accessed December 18th, 2018).

[18] VeresOne, "VeresOne: A Globally Interoperable Blockchain for Identity," Available at: https://veres.one/ (Accessed December 18th, 2018).

[19] Decentralized Identity Foundation, Available at: https://identity.foundation/ (Accessed November 9th, 2018).

[20] M. Castro, et al, "Practical Byzantine fault tolerance," OSDI, Vol. 99, pp. 173186, 1999.

[21] G. Wood, "Ethereum: A Secure Decentralised Generalised Transaction Ledger," Ethereum Yellow Paper, 2014. Available at: https://github.com/ethereum/yellowpaper (Accessed December 18th, 2018).

[22] Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation).

[23] A library for implementing an OAuth2 Server in php, Available at: https://github.com/bshaffer/oauth2-server-php (Accessed December 14th 2018).

[24] Hyperledger Indy, Available at: https://www.hyperledger.org/projects/hyperledger-indy. (Accessed December 14th 2018).

[25] Hyperledger Indy SDK, Available at: https://github.com/hyperledger/indy-sdk/ (Accessed March 20th 2019).

[26] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile," IETF RFC 5280, May 2008.

[27] A. Delignat-Lavaud, C. Fournet, M. Kohlweiss, and B. Parno, "Cinderella: Turning Shabby X.509 Certificates into Elegant Anonymous Credentials with the Magic of Verifiable Computation," in Proceedings of 2016 IEEE Symposium on Security and Privacy (SP), May 2016.

[28] Sovrin, "The Sovrin Network and Zero Knowledge Proofs," Available at: https://sovrin.org/the-sovrin-network-and-zero-knowledge-proofs/ (Accessed December 14th 2018).

[29] eduroam - World Wide Education Roaming for Research & Education, Available at: https://www.eduroam.org/ (Accessed December 7th 2018).

[30] M. Jones, J. Bradley, and N. Sakimura, "JSON Web Token (JWT)," IETF RFC 7519, May 2015.

[31] S. Blake-Wilson, M. Nystrom, D. Hopwood, J. Mikkelsen, and T. Wright, "Transport Layer Security (TLS) Extensions," IETF RFC 3546, 2003.

[32] M. Sabadello et al., "Introduction to DID Auth," Available at https://github.com/WebOfTrustInfo/rebooting-the-web-of-trust-spring2018/blob/master/final-documents/did-auth.md (Accessed December 5th, 2018).

[33] P. Wouters, H. Tschofenig, J. Gilmore, S. Weiler, and T. Kivinen, "Using Raw Public Keys in Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)," IETF RFC7250, 2014.

[34] E. Rescorla, K. Oku, N. Sullivan, and C. Wood, "Encrypted Server Name Indication for TLS 1.3," IETF draft. Available at: https://tools.ietf.org/html/draft-ietf-tls-esni-02 (Accessed December 14th 2018).