Towards a standardized model for privacy-preserving Verifiable Credentials

JESÚS GARCÍA-RODRÍGUEZ*, RAFAEL TORRES MORENO, JORGE BERNAL BERNABÉ, and ANTONIO SKARMETA, University of Murcia, Spain

Lack of standardization and the subsequent difficulty of integration has been one of the main reasons for the scarce adoption of privacy-preserving Attribute-Based Credentials (p-ABC). Integration with the W3C's Verifiable Credentials (VC) specification would help by encouraging homogenization between different p-ABC schemes and bringing them all closer to other digital credentials. What is more, p-ABCs can help to solve privacy issues that have been identified in applications of VCs to use cases like vaccination passports. However, there has not been much work focusing on the collaboration between p-ABCs and VCs. We address this topic by establishing initial steps for extra standardization of elements that will help with the integration of p-ABCs into the standard. Namely, we propose a data model for predicates, which are a staple of p-ABC systems, and tools and guidelines to ease the adaptation process like a validation meta-schema. These ideas have been applied in a proof-of-concept implementation of the OLYMPUS distributed p-ABC scheme paired with serialization following the VC data model.

CCS Concepts: • **Security and privacy → Pseudonymity, anonymity and untraceability**.

Additional Key Words and Phrases: Verifiable Credentials, p-ABC, privacy

ACM Reference Format:
Jesús García-Rodríguez, Rafael Torres Moreno, Jorge Bernal Bernabé, and Antonio Skarmeta. 2021. Towards a standardized model for privacy-preserving Verifiable Credentials. In *The 16th International Conference on Availability, Reliability and Security (ARES 2021), August 17–20, 2021, Vienna, Austria.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3465481.3469204

1 INTRODUCTION

Privacy-preserving Attribute-Based Credentials (p-ABC) have been proposed as a possible solution to increase users privacy and enable their sovereignty over their identity. In a nutshell, the user receives a credential that contains a set of certified attributes (e.g. date of birth, nationality, etc). When she wants to access a service, which specifies an access policy P, she can generate a presentation token using her credential. The presentation token contains only the minimal attributes requested by the policy and is even capable of including predicate proofs which allow avoiding data disclosure to a further extent (i.e. the user is over 18 instead of being exactly 20 years old). Adoption of p-ABCs has been scarce because of multiple reasons like lack of efficiency of existing solutions. One of the biggest issues has been the difficulty of integration with existing technologies because of the complexity and particularity of the processes and structures involved.

The World Wide Web Consortium (W3C) Verifiable Credential (VC) specification [9] establishes a model for representing digital credentials in an interoperable and machine-verifiable way. The specification focuses on credentials and presentations in a general sense but also opens a window for privacy scenarios and, in particular, zero-knowledge proofs and p-ABCs. However, the integration of p-ABCs with the structures proposed in the specification is not trivial. There

Manuscript submitted to ACM

is room for improvement on actually ensuring privacy is attained and on extra devices and definitions to encourage adoption of the data model by p-ABC systems.

In this work, we aim to bridge that gap by providing tools that will be useful for all p-ABC systems that want to follow the specification. Chiefly, we propose a data model for representing predicates over attributes in Verifiable Credentials and Presentations, as well as materials and guidelines facilitating integration like a validation meta-schema.This is accompanied by a description of how we have applied the results to the distributed p-ABC system we have implemented in the H2020 OLYMPUS project.

The (work-in-progress) VC standard has been applied in some scenarios, with debatable success in achieving goals like privacy [3, 4]. However, no work in the literature considers the application of p-ABCs along with the VC specification. With these contributions, we partially try to tackle this issue. The discussion and tools described will be helpful for any p-ABC system, even with the many differences and particularities among them.

The rest of this paper is structured as follows. Section 2 gives quick overview on related works. Section 3 introduces key concepts of the W3C VC specification and an analysis of their impact on privacy scenarios, and more specifically on p-ABCs. Section 4 describes the proposed data models for p-ABCs within the context of the standard, while Section 5 gives an overview of their application to the OLYMPUS project[1]. Section 6 concludes this work and outlines future work spanning from these results.

2 RELATED WORK

Although the W3C Verifiable Credential (VC) data model specification [9] is relatively recent and still evolving, it has already received a spike of attention. Works like [3, 7] propose the usage of Verifiable Credentials empowered with other technologies like biometric authentication with FIDO. These solutions, however, are not optimal for privacy heavy scenarios, as they have problems achieving minimal disclosure and unlinkability. Specifically, they rely on credentials with a single attribute for selective disclosure and are not fit for proving predicates over attributes.

The VC specification is usually considered alongside Decentralized Identifiers (DIDs) [8], like in [1], which proposes a Distributed Ledger Technology (DLT) based registry to fulfil the role of verifiable data registry in those standards. DLTs are fit to act as distributed databases, as they are a consensus of information replicated and shared among multiples nodes spread in different locations. Because of their decentralized nature, and their offer of data immutability, DLTs are indeed a natural candidate for the verifiable data registry role, especially in the case of DIDs. This work, however, does not really treat the usage of VCs or the potential privacy issues coming from it.

The standard and its application to specific use cases have not been free from criticism. In [4], privacy flaws in applications of DIDs+VCs to propose vaccination passports (e.g., failure to demonstrate the security of the proposals, or how data is actually protected as zero-knowledge are not applied). Brunner et al. [2] even suggest that DIDs have glaring issues for privacy-focused scenarios (particularly if they are used to identify users or credentials) that hinder the combined usage of DIDs and VCs.

Nevertheless, there are no publications that specifically consider the usage of p-ABC schemes in conjunction with the VC standard. This issue is especially jarring because proper usage of p-ABCs while avoiding identifiers (which, admittedly, are optional as per the specification) would solve most or all of the privacy concerns that have been pointed in the existing literature. As the integration of these technologies into the standard is not a trivial matter, we hope that our work in creating data models and supporting constructs will help with further adoption.

[1] https://olympus-project.eu/

3 W3C VERIFIABLE CREDENTIALS SPECIFICATION

The W3C Verifiable Credential standard establishes a model for representing digital credentials in an interoperable machine-readable way, while also being a human-readable format. The standard mainly focuses on Verifiable Credentials, which are cryptographically protected, but also considers the concept of (Verifiable) Presentations, which share data coming from credentials to a relying party. With these two constructs, it aims to empower user's control over their identities, drawing near to the goal of Self-Sovereign Identity (SSI). SSI proposes principles for identity management related to security of identity data (like protection in storage or minimization during exchanges), controllability (who controls data, owner's consent…) and portability (e.g., interoperability, very relevant in this topic) [10].

The scenario outlined in the specification considers four main roles. **Issuers** generate verifiable credentials to **holders**, which use them to present information to **verifiers**. This process is supported by a **verifiable data registry**, which keeps identifiers, public schemas or verification parameters (e.g. public keys). While multiple technologies can play the role of a data registry (e.g. a simple trusted database), Distributed Ledger Technologies are posed as a leading tool for the task, further embracing the SSI principles. This system is not a novel idea, but a formalization of the natural entities and processes for digital (arguably even physical) credentials.

The specification establishes definitions and constraints for various properties. Here, we briefly introduce the ones that will be most relevant for following this paper. The **context** property is used to map short and human-readable property names to the URIs that define those elements. While not mandatory, the context is intended to be parsed using JSON-LD [6]. Conversely, schemas (**credentialSchema**) are used for imposing the structure and content of the credential, verifying that they conform to a predefined format or even transforming it into a different encoding. The **credentialSubject** property defines a set of objects containing properties related to a subject of the verifiable credential, forming her partial identity. Finally, the **proof** property introduces the necessary details to evaluate a credential or a presentation and is mandatory to consider the structure as verifiable. **Embedded proofs** are particularly relevant because they are necessary to embody p-ABC specific proof formats. The specification takes into account the variability for proofs and leaves the set of name-value pairs that are expected inside a proof open (except for *type*, which is mandatory).

The frameworks posed in the standard are a seamless fit for privacy-focused scenarios, namely ones based on privacy-preserving Attribute-Based Credentials (p-ABCs, sometimes also known as anonymous credentials). While the specification does mention them, it mainly focuses on definitions for credentials in general. Thus, the discussion on potential privacy concerns affecting the specification and possible solutions is somewhat lacking for particular topics (though others, like device fingerprinting, are perfectly covered).

For instance, to limit the Personal Identifiable Information (PII) revealed in the usage of Verifiable Credentials, the specification proposes using abstract claims like "ageOver". This mechanism has been considered in other solutions and even (under development) standards like [5]. However, except for particular cases, it could be labelled as a "band-aid fix" instead of completely taking care of the issue. Using this kind of *attributes* poses some problems:

- If we want to cover different situations, (i.e., *over18*, *over19* …), credential size increases a lot, including many "uninformative" attributes.
- Flexibility on what can be proven is lost, so even if we approach data minimization, in some cases it cannot be achieved. For example, the previously mentioned approach of adding *overX* attributes for age would leave out non-integer ages like 17 years and six months, which can be useful for some use-cases (driving classes).

- Using this kind of derived attributes instead of the original (e.g., date of birth) leads to heavily increased complexity for ensuring that the asserted values are valid in the instant an interaction occurs.

With p-ABCs, predicates over the original attributes can simply be proven during the presentation phase, avoiding these issues. Another example in the specification where a proposed solution pursuing privacy could be evaded with the use of p-ABCs is the endorsing of single-use credentials (the underlying issue can be partly attributed to over-emphasis of the specification on identifiers). This case would lead to a heavy increase of credential issuances (which can be costly) and hamper the possibility for "offline" cases, where holders only need communication with verifiers (e.g. through NFC). The unlinkability property of p-ABCs could be leveraged to solve the underlying issues tackled by this solution, so they could be mentioned to achieve a more exhaustive discussion.

In regards to usage of the standard for p-ABCs, we find that, despite disparities between different solutions, some extra effort on standardization beyond the specification tailored to those systems would benefit potential adopters. In particular, establishing a data model for representing predicates over the holder's attributes in presentations and, what is more, the standardization of policies used by services to determine the requested information would have a positive impact on adoption and interoperability, as it is a common need among p-ABC systems. Another point of interest comes from the fact that the specification points out the need and usefulness of validation and encoding schemas when integrating p-ABCs. While full standardization of these elements is not really practical because of the varying needs of different p-ABC schemes, we find that extra facilities can be provided, like a meta-schema with helpful definitions or design guidelines.

4 TOWARDS A DATA MODEL FOR P-ABCS IN VERIFIABLE CREDENTIALS

As outlined in previous sections, adoption of the Verifiable Credential standard by p-ABC systems and, consequently, reception of the systems themselves would benefit from taking the data models beyond the current specification. The main element we have identified is establishing a model for representing the predicates over attributes that characterize p-ABCs, supported by extra work on facilitating integration into the standard.

The key idea is representing the predicates as simple JSON objects so there are two possibilities for attributes (JSON properties) when doing a presentation: a valid value or a predicate. Predicates will be formed by an *operation* tag representing the relationship between what is being proved and the *value*, whose internal structure will depend on the specific operation (Figure 1 shows valid and invalid example predicates). For this document, we consider operations for range predicates and proofs of membership, which among the most common relationships considered in p-ABC schemes, but we find that extension to other cases will be simple. In particular, we contemplate:

- The tags *"ge"*, *"le"* and *"inRange"* correspond to *greater-or-equal*, *lesser-or-equal* and *between-values* relationships. The *value* property must be comprised of a *"lowerBound"*, *"upperBound"*, and both elements, respectively.
- The tags *"memberOf"* and *"nonMemberOf"* represent proofs of *membership* and *non-membership*, respectively. In both cases, the *value* must contain a *"set"*. In our data model, the *"set"* property can be an explicit collection of the values in the set, but we expect that the second allowed form, a URL pointing to a definition of the set, will be more useful in many cases. The reason is that in many proof-of-membership systems, public parameters and/or setup is needed for each set.

As we can see, parsing these predicates will be quite simple, as the application will know what properties to expect and simply recover them from the JSON object. However, simply using the data model as is can lead to some cumbersome

[2]Full schema in https://github.com/JesusGarciaRodriguez/W3CpABCschemas

```
{
    "operation": "inRange",
    "value": {
        "lowerBound": 12,
        "upperBound": 20
    }
}
```

(a)

```
{
    "operation": "le",
    "value": {
        "lowerBound": 12
    }
}
```

(b)

Fig. 1. Example predicates using the data model. (a) Valid predicate. (b) Invalid predicate, as it is missing an *upperBound* element for a *lesser-or-equal* operation and the *lowerBound* element is not expected.

```
"type": "object",
"required": ["operation","value"],
"properties": {
    "operation": {
        "type": "string",
        "enum": ["ge","le","inRange","memberOf","nonMemberOf"]
    },
    "value": {
        "type": "object",
        "properties": {
            "lowerBound": {
                "oneOf": [
                    {"type": "number"},
                    {"$ref": "#/definitions/dateRFC3339"}
                ]
            },
            "upperBound": {
                "oneOf": [
                    {"type": "number"},
                    {"$ref": "#/definitions/dateRFC3339"}
                ]
            },
...
```

(a)

```
"anyOf":[
    {
        "properties": {
            "operation": { "const": "ge" },
            "value":{
                "required": ["lowerBound"],
                "allOf":[
                    {"not":{"required":["upperBound"]}},
                    {"not":{"required":["set"]}}
                ]
            }
        }
    },
  ...
]
}
```

(b)

Fig. 2. Extracts from the validation schema for our predicate data model[2].

checks (Is the value property indeed there? Does it contain the expected elements?). While this would be a potential issue with any data model, we can take advantage of the context in which we are using it (W3C VC specification) to seamlessly address it. Indeed, the specification includes the concept and encourages the use of credential schemas. In particular, we can include a validation scheme to verify the credential syntactically. Through this validation, we can avoid future checks on the structure of the credential being parsed.

We define a schema for validating our predicate data model using the tools of the JSON core schemas[3]. There are multiple ways to define a schema that fulfils our objective, like trying the "brute force' approach of defining a schema for each possibility and tying them with the *"anyOf"* keyword. However, we design a specific schema looking for compactness and ease of extension. Figure 2 shows extracts from this schema. The extract in Subfigure *a* shows how the general structure of the predicate is established. Subfigure *b* illustrates how we use the *required* keyword to apply the restrictions on the *value* property depending on the operation that we described earlier.

Note that these restrictions serve a two-fold purpose. First, we ensure that the expected element/s is present. E.g., a greater than predicate defines the anticipated lower bound. This may be enough for simplifying usage in most cases, as parsers could assume that needed values are included. Second, we check that only the necessary elements are contained in the predicate (by combining *not* and *required* keywords). While extra values could be ignored when parsing the predicate (maybe leading to successful verification), conceptually those predicates would be invalid (and probably nonsensical), not following the proposed data model.

The validation schema for predicates brings us to another topic where we find extra materials will be useful for facilitating integration. While some content of validation schemas will depend on variables of specific deployments (attributes, optional fields …), others like the structure of fields defined in the standard will be common for all. We expect that defining a standard meta-schema will make the development of validation schemas easier encouraging adoption. This meta-schema would include useful definitions that can be used through the JSON schema referencing mechanisms, like validation schemas for elements included in the standard (like the one used for predicates, or the one shown in Figure 3) or supporting elements like a date in RFC3339 format as referenced in Figure 2.

```
"credSchema":{
    "type":"object",
    "required":["id","type"],
    "properties":{
        "type":{
            "type": "string",
            "format": "uri"
        },
        "id":{
            "type": "string",
            "format": "uri"
        }
    }
},
```

Fig. 3. Extract from the proposed validation meta-schema, establishing an schema for a *credentialSchema* value of the specification[2].

The specification also points to the need for an encoding schema that manages the transformation of the content into the formats needed for zero-knowledge computations. Theoretically, it may be possible to do this using regular schemas (in fact, our meta-schema defines an annotation keyword to describe information about encoding, but only as a comment for developers for now). However, the specification gives the option of pointing to a binary that performs the transformations. We think this idea is better for making integration easier because of the particularities of different p-ABC schemes. Nonetheless, this does not mean that the validation schema cannot be useful for helping in the process. Indeed, the schema may contain most of the information needed for transformation, like restrictions on attribute values

[3]https://json-schema.org/specification.html

(type, minimum, maximum…). Because of that, we think it would be beneficial to endorse as a guideline the creation of encoding binaries for p-ABC schemes that are configurable using the information in validation schemas (or another automatic way), so there is no need for specific binaries for different deployments.

5 APPLICATION TO OLYMPUS SOLUTION

One of the approaches for privacy-preserving identity management in the OLYMPUS project relies on distributed p-ABCs[4].In an effort to ease integration into existing systems, we adapt the data model used to represent credentials and presentations to follow the W3C Verifiable Credential specification [3]. For this adaptation, we needed to establish the profile of use of the standard (i.e., which and how optional functionalities are considered) and some definitions for constructs specific to our approach.

We consider the **context** field mandatory, which must include the VC context, the project's general context and a deployment-specific context that deals with the contemplated attributes. The general context outlines the OLYMPUS credential and presentation types and, crucially, three new types of proofs needed for the cryptography involved. The first type, *OlPsSignature*, is the one used to legitimate a credential, while the other two, *OlPsDerivedProof* and *OlPsDerivedProofRange*, will be derived proofs for presentations.
In a similar vein, each deployment would have a specific **validation schema** supported by the general meta-schema. Note that, for the most part, deployers would only have to make small modifications to an example schema to adjust to the specific attributes they work with. Conversely, for the **encoding schema** we envisage a binary that works for everyone using the project's distributed p-ABCs with previous configuration (doubling down as a library for handling the proofs).

In regards to optional fields, we avoid using **"meta-properties"** like *ids*, *evidence* …in places that can identify the user. At most, they should be kept hidden in presentations, revealing them only in specific moments where the user does not mind losing privacy. In fact, we consider that meta-data identifying the issuer, lifetime and technical processing (context, schemas…) are enough for most p-ABC application cases.
Minimizing the number of fields in credentials also has a beneficial impact on efficiency, as each would have to be added as an "attribute" to the p-ABC signing scheme to ensure its validity. Note that there are some exceptions to this, as some fields can be validated implicitly (e.g., a proof will only validate against the public key associated with the correct value of the issuer field).

The implicit validation of issuer fields is also an advantage for the particular case of distributed issuance. The user receives *partial credentials*, which in our case are equivalent to credentials issued by each partial issuer. She then combines them into a credential linked to the issuer as a whole. This process involves cryptographic operations to obtain the valid signature but also manipulation of the VC structure. If fields like issuer were explicitly protected using the signing scheme, partial credentials would need to be treated as a special case. With the implicit validation approach, partial credentials are encompassed in the specification and the user can generate a final credential without issues.

The tools and constructions presented in this document are fit for any p-ABC system, although the integration would still require some work specific to the concrete schemes used. We already have definitions for the materials mentioned in this section (contexts, schemas…) and a proof-of-concept implementation that generates and validates example credentials and presentations[5]. However, there are still some limitations that will require extra work in the future. The implementation is not yet integrated with a verifiable data registry for sharing and validating public parameters like

[4]https://olympus-project.eu/
[5]Examples in https://github.com/JesusGarciaRodriguez/W3CpABCschemas

verification keys or schemas. In relation to this, the pre-configuration needed for attribute encoding or handling proofs is done somewhat "manually" using the project's own tools and structures. Also, the p-ABC system itself has not yet reached full maturity, as it only has range predicates fully implemented (i.e., set membership or other predicates are missing yet).

6 CONCLUSIONS AND FUTURE WORK

Adoption of p-ABCs would benefit from standards, like W3C's Verifiable Credentials (VC), that simplify integration with other technologies. Reciprocally, p-ABCs could be applied to avoid privacy issues in VCs. However, this synergism has been hampered by the difficulty of combining p-ABCs with the specification, and the combination has not been explored in the literature.

With that in mind, we have analyzed the VC specification in regards to privacy and, specifically, its interaction with p-ABCs. We proposed a data model for predicates and other tools for simplifying the adoption of the VC specification by p-ABCs, like a validation meta-schema. We have also shown that this is a viable strategy by developing a proof-of-concept implementation along with the necessary definitions. Still, as mentioned in Section 5, some more work is needed to have a complete implementation of the p-ABC system following the specification. The key element will be the introduction of DLT as a verifiable data registry for public keys, schemas…Paving the way for automating and securing trust in configuration processes and even integrating these mechanisms within DLT technologies through the use of smart contracts, bringing transparency and auditability to ecosystems. We also plan to implement set membership proofs to increase the capabilities of the p-ABC scheme.

ACKNOWLEDGMENTS

The project leading to this application has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 786725 (https://cordis.europa.eu/project/id/786725) (OLYMPUS project).

REFERENCES

[1] Bander Alzahrani. 2020. An Information-Centric Networking Based Registry for Decentralized Identifiers and Verifiable Credentials. *IEEE Access* 8 (2020), 137198–137208. https://doi.org/10.1109/ACCESS.2020.3011656

[2] Clemens Brunner, Ulrich Gallersdörfer, Fabian Knirsch, Dominik Engel, and Florian Matthes. 2020. *DID and VC:Untangling Decentralized Identifiers and Verifiable Credentials for the Web of Trust.* Association for Computing Machinery, New York, NY, USA, 61–66. https://doi.org/10.1145/3446983.3446992

[3] David W. Chadwick, Romain Laborde, Arnaud Oglaza, Remi Venant, Samer Wazan, and Manreet Nijjar. 2019. Improved Identity Management with Verifiable Credentials and FIDO. *IEEE Communications Standards Magazine* 3, 4 (2019), 14–20. https://doi.org/10.1109/MCOMSTD.001.1900020

[4] Harry Halpin. 2020. Vision: A Critique of Immunity Passports and W3C Decentralized Identifiers. In *Security Standardisation Research*, Thyla van der Merwe, Chris Mitchell, and Maryam Mehrnezhad (Eds.). Springer International Publishing, Cham, 148–168. https://doi.org/10.1007/978-3-030-64357-7_7

[5] ISO/IEC CD 18013-5:2019(E). 2019. *Personal Identification – ISO Compliant Driving Licence – Part 5: Mobile Driving Licence (mDL) application.* Committee Draft Standard. International Organization for Standardization.

[6] Gregg Kellogg, Pierre-Antoine Champin, and Dave Longley. 2020. JSON-LD 1.1: A JSON-based Serialization for Linked Data. https://www.w3.org/TR/json-ld11/

[7] Romain Laborde, Arnaud Oglaza, Samer Wazan, François Barrere, Abdelmalek Benzekri, David W. Chadwick, and Rémi Venant. 2020. A User-Centric Identity Management Framework based on the W3C Verifiable Credentials and the FIDO Universal Authentication Framework. In *2020 IEEE 17th Annual Consumer Communications Networking Conference (CCNC)*. 1–8. https://doi.org/10.1109/CCNC46108.2020.9045440

[8] Drummond Reed, Manu Sporny, Dave Longley, Christopher Allen, Ryan Grant, Markus Sabadello, and Jonathan Holt. 2020. Decentralized identifiers (dids) v1.0. *Draft Community Group Report* (2020). https://www.w3.org/TR/did-core/

[9] Manu Sporny, Dave Longley, and David Chadwick. 2019. Verifiable credentials data model v1.0. *W3C, W3C Recommendation, November* (2019). https://www.w3.org/TR/vc-data-model/

[10] A. Tobin and D. Reed. 2016. The Inevitable Rise of Self-Sovereign Identity. The Sovrin Foundation. https://sovrin.org/wp-content/uploads/2017/06/The-Inevitable-Rise-of-Self-Sovereign-Identity.pdf