Cryptographic Toolbox for Privacy Preserving with Applications to Self Sovereignty

Renaud Dubois

®Ledger
6 rue Gretry, 75002 Paris – France
firstname.lastname@ledger.fr

Abstract. Self-sovereign identity (SSI) is an approach to digital identity that gives individuals control of their digital identities. In the Web3 framework, this **D**ecentralized **Id**entity is referred as DId [W3C12]. In the litterature, the dedicated Cryptographic solution is referred as Anonymous Credentials (AC) [CL02], [CL04]. It is a complex protocol that requires the use of several Privacy-Preserving protocols. The aim of this memo is to provide an overview of existing DIDs frameworks and the underlying cryptographic mechanisms in order to give an input to reflexion about how SSI could benefit to Web3 and be enforced by Ledger products and solution.
keywords: Decentralized Idenfiers, Credential, Zero knowledge, Cryptographic Commitments

Introduction

Decentralized Idenfiers. Self Sovereign Identity (SSI) : is a paradigm shift from centralized and trusted Credential issuing to a 'user-centric' management of the identity. According to [W3C12], Decentralized identifiers (DIDs) are a new type of identifier that enables verifiable, decentralized digital identity. A DID refers to any subject (e.g., a person, organization, thing, data model, abstract entity, etc.) as determined by the controller of the DID. In contrast to typical, federated identifiers, DIDs have been designed so that they may be decoupled from centralized registries, identity providers, and certificate authorities. Specifically, while other parties might be used to help enable the discovery of information related to a DID, the design enables the controller of a DID to prove control over it without requiring permission from any other party. DIDs are URIs that associate a DID subject with a DID document allowing trustable interactions associated with that subject.

Anonymous Credentials. In traditional electronical authentication, user identities are readily available to service providers. These latter can exchange collected data about any particular user among themselves. Anonymous credentials aims at mitigating such privacy breaches and giving the user control over his data. Informally, as explained in [BBC$^+$18], a user acts under an arbitrary number of unlinkable pseudonyms rather than under his identity. In anonymous credentials, these access rights are described by attributes. A service provider can issue a credential to a user, which is parameterized with attributes. These attributes can, for example, encode access rights to a service or some user data. The user can then prove possession of a credential to the same or to other service providers in a privacy-preserving way. This process is called showing a credential.

The first efficient realization of anonymous credentials was first issued by Carmenish in [CL02] and [CL04] which demonstrated how to build such protocol using the following primitives:

– a commitment scheme (the analogue of digital envelop),
– a signature scheme,
– an efficient signature scheme (a protocol to obtain a signature on a commited value without revealing the value to the signer),
– a zero knowledge proof of knowledge (ZKPoK, NIZKP).

The three laters are the real algorithmic stake because they are the major bottleneck.

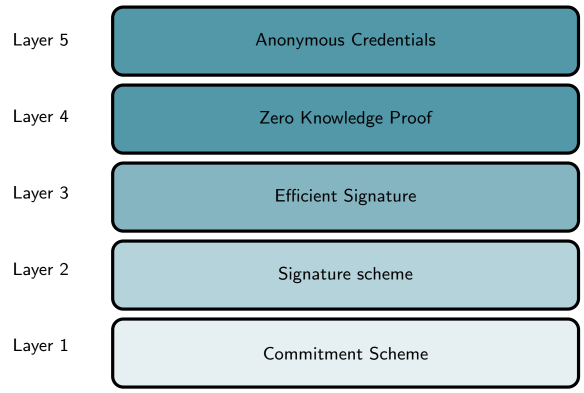

Fig. 1. The cryptographic stack for AC construction

Contributions The efficiency of the design of ZKP and efficient schemes relies on the cunning imbrication of the commitment and signatures with specific properties (structure preserving, use of homomorphic additive properties, etc.). This note tries to give an overview of those imbrications and the evolution of the state of the art to its current shape. We acknowledge the previous work of [WSL$^+$19] and section D.5.2.1 of Prometheus project [Pro21] as a starting point of the redaction of this note and our comprehension. The note is structured as follows:

– the first section gives an overview of the cryptographic primitives used for AC constructions (figure 1),
– the second section provides a state of the art of existing Anonymous Credentials in the wild with fair Comparizon,
– the last section provides a survey of existing project and applications in ®Ledger that would benefit of such privacy preserving mechanisms.

The note has a minor contribution in the generalization of functional commitment to linear function described in [LRY16] to ideal Universal commitment. Unfortunately the practical (in term of complexities) commited functions are however quite limited.

1 State of the Art

1.1 Commitments

First examples A commitment scheme emulates a publicly observed sealed envelope; it allows a party to commit to a message m so that this message is not revealed

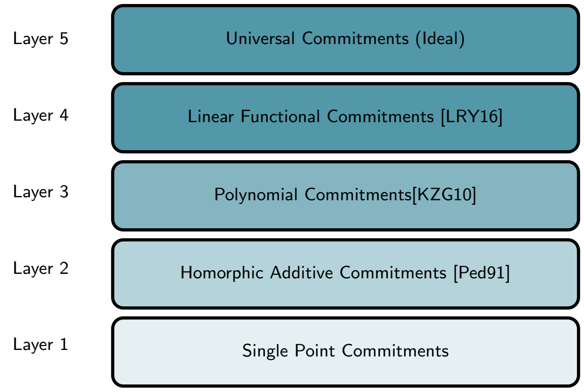

Fig. 2. Evolution of Commitment schemes

until a later moment when the commitment is opened and the receiver gets convinced that the message was indeed m. Two important security properties of commitment schemes are called **hiding** and **binding**. The first property requires that no information about the committed message is revealed to an observer. The second property means that the committing party cannot alter the message after committing to it. The first well known way to commit is to commit to the hash of a value. In this trivial scheme, the value of the envelope m is commited though its hash $h(m)$. It is possible for the Prover to proove the knwoledge of m by providing its commitment. This solution is classically used in online Password authentication, which can be considered as a zero knowledge proof of the value m. In [Ped91], the author describes a commitment scheme with **homorphic additive property**. Pedersen Vector Commitment is a direct corollary of this additive feature. He also provides the first description of a Verifiable Secret Sharing Scheme, which use the proposed commitment as a tool to provide a verification aspect to the well known Shamir Secret Sharing Scheme [Sha79] (which is in fact a diverted way to use a Reed Solomon Code). This scheme introduce the idea of an "**Opening function**", formalized later on. Indeed while commiting to x_1 and x_2, it is possible to reveal later the value $x_1 + x_2$, without revealing separately the x_i. This property is used in Monero to hide transaction amount with blinding factors. It has also been used in BulletProof for a **range proof system**. *TODO : hunt for more Perdersen Commitment uses in the litterature.*

Pedersen Commitment

- $Setup(1^\kappa, t) : < G, Q > \xleftarrow{\$} E(F_p)$
- $Commit(x, r) : \mathcal{C}(x, r) = xG + rQ$
- $Addition : \sum \mathcal{C}(x_i, r_i) : \mathcal{C}(\sum x_i, \sum r_i)$

Pedersen Vector Commitment

- $Setup(1^\kappa, t) : < G_1, \ldots, G_n, Q > \xleftarrow{\$} E(F_p)$
- $Commit(\vec{x}, r) : \mathcal{C}(\vec{x}, r) = \sum x_i G_i + rQ$

Fig. 3. Pedersen Commitments [Ped91]

Definitions. Now that some insight with basic constructions have been provided, the definition of a Universal Commitment is given. It appears as a natural extension of the Functional Linear Commitment of [LRY16] to any function.

Ideal Universal Commitment

- $Setup(1^\kappa, t)$: takes security parameter κ and additional parameters t then output an algebraic structure, optionally some $< P_k, S_k >$ elements.
- $Commit(S_K, r)$: outputs commitment $\mathcal{C}$ to element r and (optionally) auxiliary information `aux`
- $Open(S_K, r)$ output element r
- $Witness(S_K, r, i)$ output $< i, f(i,r), w_i >$ where w_i is a witness of the evaluation of f in i relatively to the commited element r.
- $VerifyOpen(S_K, \mathcal{C}, i, r, f(i,r))$
- $VerifyEval(P_K, i, f(i,r), w_i)$

Fig. 4. Ideal Universal Commitments Formalization (this note)

Scheme	Commit	Open	Witness	f

Table 1. Instanciation of known commitment schemes in the Universal Commitment framework (TBD)

Polynomial Commitment Polynomial commitments are used to provide exact range proof. The idea is to commit a Polynomial that is evaluated to 0 in all elements of the interval. It is done by commiting a polynomial P=($\prod x_i - a_i$)Q where Q is used for security.

Polynomial Commitment

- (Trusted) Setup : $(g, g^\alpha, g^{\alpha^2}, \ldots, g^{\alpha^d}), (h, h^\alpha, h^{\alpha^2}, \ldots, h^{\alpha^d})$
- Com(p,r) : $(\prod_{j=0}^d (h^{\alpha^j})^{p_j}).(\prod_{j=0}^d (g^{\alpha^j})^{r_j})$, $p(x) = \sum p_j x^j, r(x) = \sum r_j x^j$
- Open: output (p, r)
- VerifyPoly : output $\mathcal{C} \stackrel{?}{=} g^{p(\alpha)}.h^{r(\alpha)}$
- Witness :
- VerifyEval :

Fig. 5. Polynomial Commitment [KZG10]

Applications

1.2 Efficient Signatures

Groth Signature Groth signature [Gro15] is a structure-preserving signature that sign vectors of group elements.

Scheme	Required for	Description
Pedersen	VSS Monero BulletProof Credentials	Verifiable Secret Sharing Transaction amount hiding Range Proof [LCKO21], [ECA21]
Polynomial C.	eVSS	Strongly Verifiable Secret Sharing Range Proof Credentials Zk-Snark
Functional Linear		

Table 2. protocols build on top of described commitments schemes

1.3 Zero knowledge Proofs

Scheme	Required for	Description
Groth16	Tornado Nova Zcash v1	Transaction hiding (Mixer) Coin ownership
Stark		

Table 3. protocols build on top of ZKP schemes

LegoSnark: gadget for Commitments

2 Anonymous Credentials/DIds

This section describes some existing frameworks with pointers to the underlying cryptographic mechanisms. The last subsection highlight a construction providing a decentralized approach.

2.1 Existing SSI

	[ECA21]	[LCKO21]	Sovrin [Sov21]	IPv8	YL20	Uport
Control	✓	✓	✓	✓	✓	✓
Data Minimization	✓	✓	✓	✓	✓	×
Commitment Type						
Signature Type	[Gro15]					
ZKP type	[EG14]	[Gro16]	[CL02]	[CL02]+[PB10]		-
Issuer Anonymity Optimization	✓	×	×	×	×	×

Table 4. Comparizon of existing DIds (completed from[LCKO21])

2.2 A multi-issuer scheme

3 Applications for Ledgers

In this section we provide a list of Ledgers services which could benefit to an improvment of the privacy using some element of the "toolbox".

	RuleMaking			Operation		Security*			
	Network	Registry	Specification	Network	Registry	Pro	Per	Int	Conf
did:btcr	● □	● □	○	● □	● □	+	+	+	±
did:v1	◑†	◑†	○	◑†	◑†	−	±	±	+†
did:ethr	● □	$N/A^{\dagger}$	◑	● □	● ■	+	+	+	−
did:sov	◑ ■	◑ ■	◑	◑ ■	◑ ■	−	±	+	±
did:web	◑	◑	○	◑ □	◑ □	−	−	−	±
did:peer	●	●	○	◑†	◑†	±	−	−	+

* Security - Pro: protection Per: persistence Int: integrity Conf: confidentiality

● fully decentralized ◑ partially decentralized ○ centralized $N/A^{\dagger}$ not applicable

Required resources: ■ modest □ substantial

† Not clear or well defined how method satisfies criteria at time of writing

Table 5. Comparizon of existing DIds (source : [AM18])

Endorsement. During Ledger endorsement of a Nano device, the public key is send to the BackEnd, then a "challenge-response" between Ledger services and Nano is performed. Using a Zero-knowledge proof instead of sending the public key would reduce the information that Ledger could store about links between a device and a host (smartphone or labtop). This could be achieved using a Group Signatures as described in this document instead. The backend only learns that it is indeed a Nano that is connecting to the Ledger Live application, without further information.

Protect. In Protect, the seed element (input to the BIP32 protocol, the core of all user security) is stored via a threshold mechanism which is a VSS (verifiable Secret Sharing scheme) which was designed by Pedersen. While

Device ID: Airdrop One of the application of the Device Identifier (which is a very risky initiative in term of Privacy) is to perform Airdrops to the owner of nanos.

Issue Name	Description	# (Link)
Computational Delegation of pairings	Hybrid Architecture for efficient Nano PBC	
Strongly verifiable secret sharing	eVSS for Protect	
Musig2	Schnorr Multisignature	
Nano with Group Signature endorsement	Group signature for Ledger endorsement	31
Linkable signature for Airdrop	Linkable signature for Anonymous Airdrop	30

Table 6. Innovation issues related to Privacy Preserving

DIds resolution

4 Conclusion

In this note we provided an overview of some privacy preserving tools and principles. We also provide a non exhaustive list of applications that could immediately benefit from the described mechanisms. Most of those building blocks rely on the use of computation of pairing over elliptic curves.

References

AM18. Tatiana Gayvoronskaya Christoph Meinel Alexander Mühle, Andreas Grüner. A survey on essential components of a self-sovereign identity. https://arxiv.org/pdf/1807.06346.pdf, 2018.

BBC+18. Carsten Baum, Jonathan Bootle, Andrea Cerulli, Rafaël del Pino, Jens Groth, and Vadim Lyubashevsky. Sub-linear lattice-based zero-knowledge arguments for arithmetic circuits. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part II*, volume 10992 of *Lecture Notes in Computer Science*, pages 669–699. Springer, 2018.

CL02. Jan Camenisch and Anna Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In *CRYPTO - 2004 - The 22nd Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, Proceedings*, volume 2442 of *Lecture Notes in Computer Science*, pages 61–76. Springer, 2002.

CL04. Jan Camenisch and Anna Lysyanskaya. Signature schemes and anonymous credentials from bilinear maps. In Matthew K. Franklin, editor, *Advances in Cryptology - CRYPTO 2004, 24th Annual International CryptologyConference, Santa Barbara, California, USA, August 15-19, 2004, Proceedings*, volume 3152 of *Lecture Notes in Computer Science*, pages 56–72. Springer, 2004.

ECA21. Kaoutar Elkhiyaoui, Angelo De Caro, and Elli Androulaki. Multi-issuer anonymous credentials without a root authority. *IACR Cryptol. ePrint Arch.*, page 1669, 2021.

EG14. Alex Escala and Jens Groth. Fine-tuning groth-sahai proofs. In Hugo Krawczyk, editor, *Public-Key Cryptography - PKC 2014 - 17th International Conference on Practice and Theory in Public-Key Cryptography, Buenos Aires, Argentina, March 26-28, 2014. Proceedings*, volume 8383 of *Lecture Notes in Computer Science*, pages 630–649. Springer, 2014.

Gro15. Jens Groth. Efficient fully structure-preserving signatures for large messages. In Tetsu Iwata and Jung Hee Cheon, editors, *Advances in Cryptology - ASIACRYPT 2015 - 21st International Conference on the Theory and Application of Cryptology and Information Security, Auckland, New Zealand, November 29 - December 3, 2015, Proceedings, Part I*, volume 9452 of *Lecture Notes in Computer Science*, pages 239–259. Springer, 2015.

Gro16. Jens Groth. On the size of pairing-based non-interactive arguments. *IACR Cryptol. ePrint Arch.*, page 260, 2016.

KZG10. Aniket Kate, Gregory M. Zaverucha, and Ian Goldberg. Constant-size commitments to polynomials and their applications. In Masayuki Abe, editor, *Advances in Cryptology - ASIACRYPT 2010 - 16th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 5-9, 2010. Proceedings*, volume 6477 of *Lecture Notes in Computer Science*, pages 177–194. Springer, 2010.

LCKO21. Jeonghyuk Lee, Jaekyung Choi, Jihye Kim, and Hyunok Oh. Privacy-preserving identity management system. *IACR Cryptol. ePrint Arch.*, page 1015, 2021.

LRY16. Benoît Libert, Somindu C. Ramanna, and Moti Yung. Functional commitment schemes: From polynomial commitments to pairing-based accumulators from simple assumptions. In Ioannis Chatzigiannakis, Michael Mitzenmacher, Yuval Rabani, and Davide Sangiorgi, editors, *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, volume 55 of *LIPIcs*, pages 30:1–30:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016.

PB10. Kun Peng and Feng Bao. An efficient range proof scheme. In Ahmed K. Elmagarmid and Divyakant Agrawal, editors, *Proceedings of the 2010 IEEE Second International Conference on Social Computing, SocialCom / IEEE International Conference on Privacy, Security, Risk and Trust, PASSAT 2010, Minneapolis, Minnesota, USA, August 20-22, 2010*, pages 826–833. IEEE Computer Society, 2010.

Ped91. Torben P. Pedersen. Non-interactive and information-theoretic secure verifiable secret sharing. In Joan Feigenbaum, editor, *Advances in Cryptology - CRYPTO '91, 11th Annual International Cryptology Conference, Santa Barbara, California, USA, August 11-15, 1991, Proceedings*, volume 576 of *Lecture Notes in Computer Science*, pages 129–140. Springer, 1991.

Pro21. Project prometheus privacy preserving post-quantum systems from advanced cryptographic mechanisms using lattices. 2021.

Sha79. Adi Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.

Sov21. Sovrin. How sovrin works. https://eprint.iacr.org/2021/1459.pdf, 2021.

W3C12. Decentralized Identifiers (DIDs) v1.0,Core architecture, data model, and representations. https://www.w3.org/TR/did-core/, 2012.

WSL+19. Licheng Wang, Xiaoying Shen, Jing Lib, Jun Shao, and Yixian Yanga. Cryptographic primitives in blockchains. volume 127, pages 43–58. Journal of Network and Computer Applications, 2019.

A Issues and Roadmap

B Implementation of Pairings

B.1 Pairings used in blockchain framework

B.2 Hybrid Architecture for Nano

As a constrained device, the Implementation of pairings only on the device may The pairings are a bilinear map

$$e : (G_1 \times G_2) \to \mathbb{F}_{p^k},$$

where G_1 and G_2 are elliptic curves define over $\mathbb{F}_p$ and $\mathbb{F}_{p^d}$ and $\mathbb{F}_{p^k}$ some prime field extension of degree k. The pairing e is homomorphic such that $e(aP, bQ) = e(P, Q)^{ab}$. Using this property it is possible to blind the computation using the following algorithm, inspired by the well known Coron countermeasure:

- randomly select a and b in Fq (q being the order of the curves)
- send aP and bQ to the host
- host compute $e(aP, nQ)$ send back results
- delegator (Nano) computes $e(aP, bQ)^{-ab}$

Note that if element of G_2 are public elements, it is possible to avoid the Implementation of elliptic curve over extension fields.

C Group Signature for Endorsement

D Linkable Signature for Anonymous Airdrop

E Strongly Verifiable Secret Sharing for Seed Protection