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.

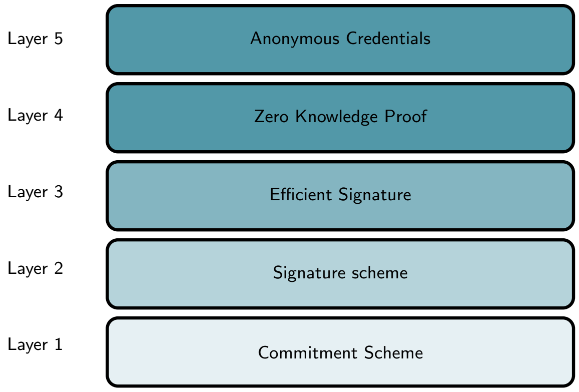


Fig. 1. The cryptographic stack for AC construction

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

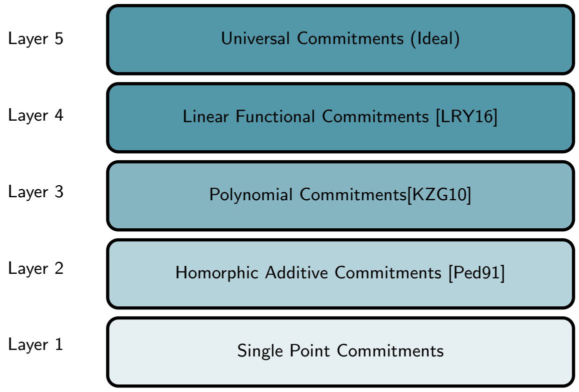

Layer 5
Universal Commitments (Ideal)
Layer 4
Linear Functional Commitments [LRY16]
Layer 3
Polynomial Commitments[KZG10]
Layer 2
Homorphic Additive Commitments [Ped91]
Layer 1
Single Point Commitments

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

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

Pedersen Vector Commitment

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

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 Commitments 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]

Functional Linear Commitments

Applications

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

1.2 Efficient Signatures

Schnorr Signature Schnorr signatures [Sch89] are almost as old as ECDSA, which was selected mainly because Schnorr used to be covered by patents (now expired), limiting its adoption. Due to some different favorable algebraic properties, it is feasible to convey additive properties into the computation. Doing so, combining SS [Sha79] with Schnorr provide threshold signature, combining with Groth signatures, it enable Sibling Signatures leading to AC [CDD17]. It leads to Multisignature by enabling key aggregation ([NRS21]). The diversity of such ZK-friendly constructions is very high. This richness is such that the Bitcoin community (prudent among blockchain ones) is waiting for its adoption through BIP340 [WNR].

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

Multi, Aggregate and Threshold signatures In a multi-signature scheme [NRS21], public keys of a group of users are aggregated such that verifiers are given a signature and an aggregate public key and do not learn whether the signature was created by a single signer or by a group of signers. This leads to possible uses that may benefit to the privacy of users.

Ring Signatures A ring signature is a scheme allowing to sign messages on behalf of a group (including thyself), without revealing its identity.

Linkable Signatures Linkability is an additional property to ring signature allowing to detect if two messages have been issued by the same entity, learning nothing more about its identity.

Ring CT

1.3 Zero knowledge Proofs

A zero-knowledge protocol or proof (ZKP) is a method by which one party (the prover) can prove to another party (the verifier) that a given statement is true while no information is learned except the fact that the statement is indeed true. As for commitment this property covers a wild area, from proving knowledge of a Password, a private key (a signature is a ZKP), to general purpose circuit. The complexity of the proof depends on the richness of the proofs that can be done with the protocol.

Groth NIZK argument Groth's constant size NIZK argument is based on constructing a set of polynomial equations and using pairings to efficiently verify these equations.

Commit-and-prove ZKsnarks

LegoSnark: gadget for Commitments [?] provides . It is interesting to note that an implementation is also provided in github. (TODO: source available ZKP primitives implementations).

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

Table 3. protocols build on top of ZKP schemes

2 Anonymous Credentials/DIds

This section describes some existing DIds frameworks with pointers to the underlying cryptographic mechanisms. The last subsection focus on [?] which proposes a construction providing a decentralized approach for multi-issuers DIds.

2.1 Existing SSI

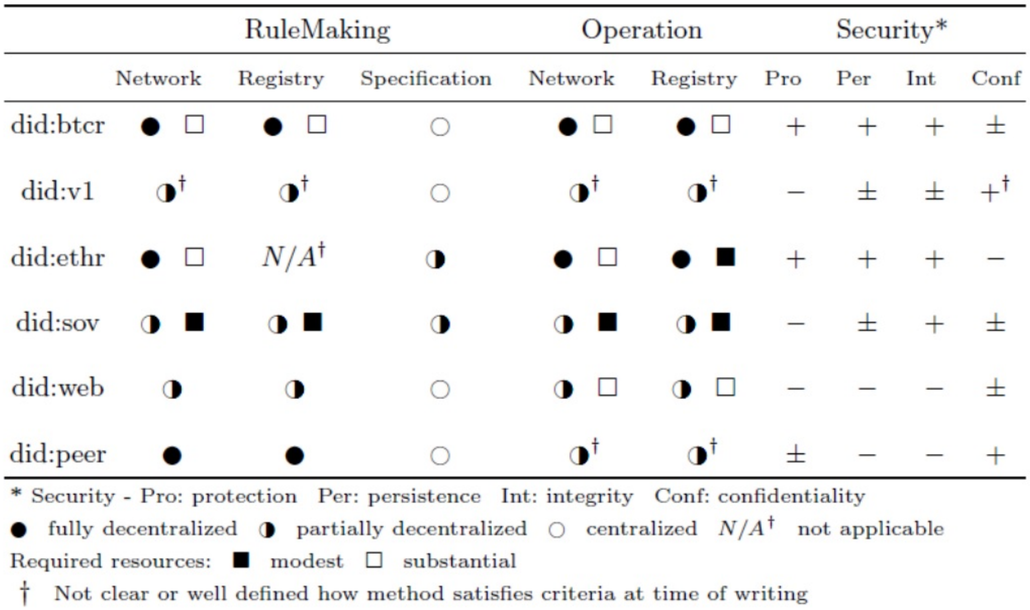
	[ECA21]	[LCKO21]	Sovrin [Sov21]	IPv8	YL20	Uport
Control	✓	✓	✓	✓	✓	✓
Data Minimization	✓	✓	✓	✓	✓	×
Commitment Type						
Signature Type	[Gro15]					
ZKP type	[EG14]	[Gro16]	[CL02]	[CL02]+[PB10]		-
Issuer Anonymity Optimization	✓	×	×	×	×	×

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" (sorry Data Team).



	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])

3.1 Endorsement.

During Ledger genuine check of a Nano device on the Ledger Live, 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 learning the device public key.

3.2 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) described in [Ped91]. While the scheme is secure under the assumption that the gathering of shares is, it could be a concern if not. The scheme is as secure as the strongest back up provider access control. In the Pedersen VSS, if an attacker success in obtaining one of the share, which is protected by KYC in Protect, the share is known, unless performing a resharing. In a ZKP-VSS scheme [KZG10], the attacker learns nothing if the reconstruction fails.

3.3 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. As for the genuine check, the sender learns the device public key. Using a linkable signature [SALY17] would provide the desired service, without having the sender learning nothing except the eligibility to the service.

3.4 Supporting Bitcoin evolutions: Musig2

As the “king of all blockchains”, Ledger should be prepared to its evolution and dispose of “on the shelf” implementation when the multisig BIP is deployed.

3.5 DIds

The latest DIds described table 4 use advanced ZKP mechanisms (compared to the nearly 20 years old [CL02]). The IBM proposal described in [ECA21] appears very promising in term of features. As opposed to blockchain mechanisms where the complex parts of the protocol (namely the BFT and POW) are left to the backbone (miners and nodes), in SSI an important part of the protocol is computed at the endpoint (the user). What appears to be a challenged will be to preserve the privacy of the user, using techniques as ’clear signing’ (use the secure display of a device to avoid luring the user to sign diverted messages), without “overloading” them with too much information. Two objectives may be distinguished for **®Ledger** :

1. **Technologic readiness** : having the necessary tools “on the shelf” to be an actor of SSI.
2. **Domain contribution** : be part of the specification of the SSI Architecture of the future.

For the moment this note only address the first objective in the perspective of a Ledger integration in advanced frameworks.

DIds Nano and Host Side As for the previous mechanism, the first building block to provide is a bolos sdk3 providing pairings computation to build ZKP protocols on top of it. The protocol choice will then depends on the DIds implementation retained. Clearly advanced ZKP features in the current incredibly innovative blokchain context seem a good bet. As such providing higher building block such as Groth signatures, Groth-Sahai proofs and LegoSnark Tools would prove the technologic advance of Ledger over its competitors. There is an architecture investigation to conduct about the host/Nano repartition of the tasks. What appears as a certainty is that the signature mechanisms have to be computed by the Secure Element. To be prepared, we recommand to put in a roadmap with a TBD prioritization the efficient signatures mechanisms described in 1.2. Note that all previous initiatives would contribute to this bigger one.

DIds Back End Side The Back End will have to deal the following problematics:

- credential issuance/revocation specification and capabilities (backbone),
- KYC (Know your clien authentication systems),
- reputation system and “Soulbond” concepts implementation.

For back end, the separation between technologic readiness and domain contribution is less clear than for endpoint. Obviously the Back end should implement the counterpart of each primitive implemented in Nano (ZKP signature verification, ZKP verification). KYC is out of scope of cryptography and is addressed by organizational measures and AI problematics, thus out of scope of this note. Concerning the last point, while Nano can focus on low-level primitives, the back end will have to imagine the use cases. Some recent concepts like “Soulbond” [**?**] are still to be precised in their implementation.

4 Conclusion

This notes provides 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 to **®Ledger**. Most of those building blocks rely on the use of computation of pairings over elliptic curves. Hence it appears has a prioritary block to implement on Nano in the context of SSI and Web3.0.

Issue Name	Description	# (Link)
Computational Delegation of pairings	Hybrid Architecture for efficient Nano PBC	48
Strongly verifiable secret sharing	eVSS for Protect	41
Musig2	Schnorr Multisignature	40
Nano with Ring Signature endorsement	Ring signature for Ledger endorsement	31
Linkable signature for Airdrop	Linkable signature for Anonymous Airdrop	30
SDK3 for Web3	Hack the Web3.0 proposal	

Table 6. Innovation Team issues related to Privacy-Preserving mechanisms

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.

CDD17. Jan Camenisch, Manu Drijvers, and Maria Dubovitskaya. Practical uc-secure delegatable credentials with attributes and their application to blockchain. In Bhavani Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, pages 683–699. ACM, 2017.

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.

NRS21. Jonas Nick, Tim Ruffing, and Yannick Seurin. Musig2: Simple two-round schnorr multi-signatures. In Tal Malkin and Chris Peikert, editors, *Advances in Cryptology - CRYPTO 2021 - 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16-20, 2021, Proceedings, Part I*, volume 12825 of *Lecture Notes in Computer Science*, pages 189–221. Springer, 2021.

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.

SALY17. Shifeng Sun, Man Ho Au, Joseph K. Liu, and Tsz Hon Yuen. Ringct 2.0: A compact accumulator-based (linkable ring signature) protocol for blockchain cryptocurrency monero. In Simon N. Foley, Dieter Gollmann, and Einar Snekkenes, editors, *Computer Security - ESORICS 2017 - 22nd European Symposium on Research in Computer Security, Oslo, Norway, September 11-15, 2017, Proceedings, Part II*, volume 10493 of *Lecture Notes in Computer Science*, pages 456–474. Springer, 2017.

Sch89. Claus-Peter Schnorr. Efficient identification and signatures for smart cards. In Gilles Brassard, editor, *Advances in Cryptology - CRYPTO '89, 9th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 1989, Proceedings*, volume 435 of *Lecture Notes in Computer Science*, pages 239–252. Springer, 1989.

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.

WNR. P. Wuille, J. Nick, and T. Ruffing. Schnorr signatures for secp256k1. bitcoin improvement proposal 340.

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.

Annexes

A Issues and Roadmap

B Implementation of Pairings

B.1 Pairings used in blockchain framework

Scheme	Curves	Ref.
Zexe Penumbra (Cosmos)	BLS12-377	eprint 2018/962
Zcash	BLS12-381	Zcash white paper

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

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

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

F Schemes and Assumptions

Scheme	Reference	Assumptions	Pairings (Y/N)

Table 7. Cryptographic schemes and underlying Assumptions and Tools

G Available ZKP primitives implementations

Libsec256k1 is an absolute reference for its concisness and efficiency. Arkwork is widely used and should be considered as a target for ledger rust bindings. blst is clear and concise C pairing based library, retained as target for the hackaton. Other libraries are given for completeness.

Library	Language	Feature	Link
Libsec256k1-zkp	C	Musig2, RangeProof	ElementsProject git
blst	C	BLS12-381 pairings	
Arkwork	rust		
Snarkjs java library for zkSNARKs	js, Web Assembly	Groth16, PLONK	iden3
LegoSnark	C++	[?]	imdea
libsnark	C++	zkSNARK	scipr-lab
Legogroth16	rust	Groth16	kobigurk
Z0Krates		Zksnark over ethereum	zokrates
libStark	C++	Zk-Stark	libSTARK