

Coconut: Threshold Issuance Selective Disclosure Credentials with Applications to Distributed Ledgers

Authors

Alberto Sonnino*

Mustafa Al-Bassam*
Shehar Bano*
Sarah Meiklejohn*
George Danezis*

* University College London

February 2019



The Authors



Alberto Sonnino



Mustafa Al-Bassam



Bano Shehar



Sarah Meiklejohn



George Danezis



Challenges in blockchains



Strong integrity



Poor privacy



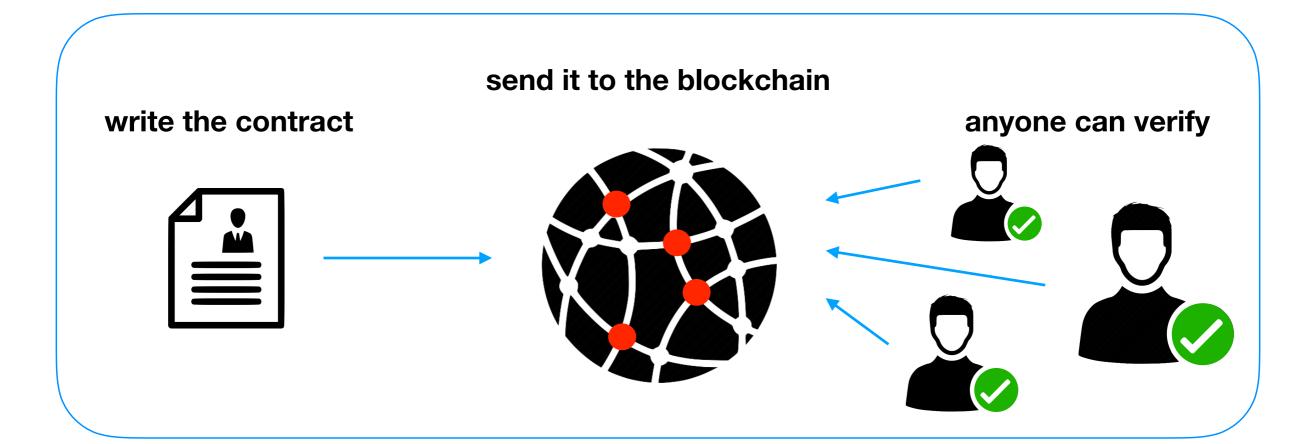
Challenges in blockchains



Strong integrity



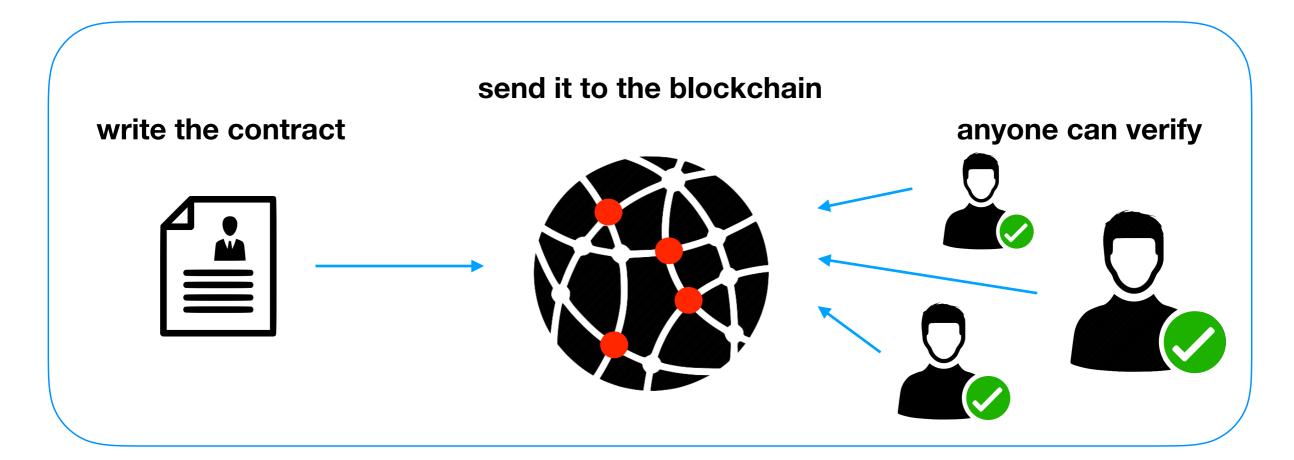
Poor privacy





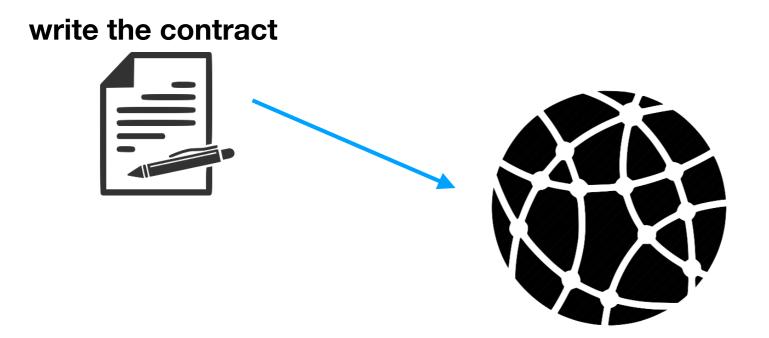
Challenges in blockchains

Can we issue credentials in this setting?



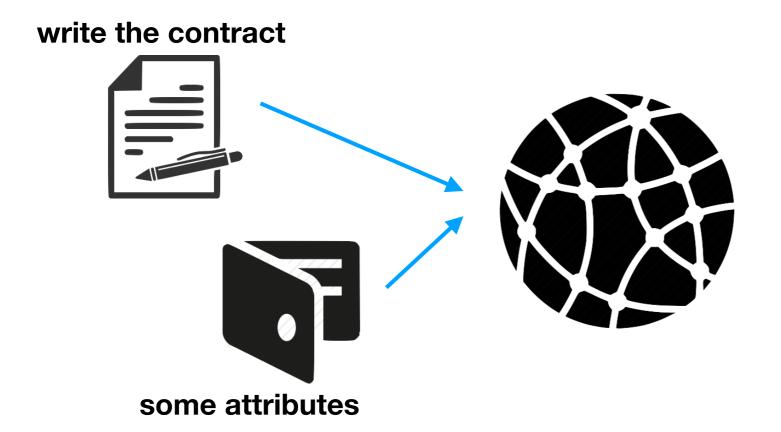


Issuing credentials through smart contracts



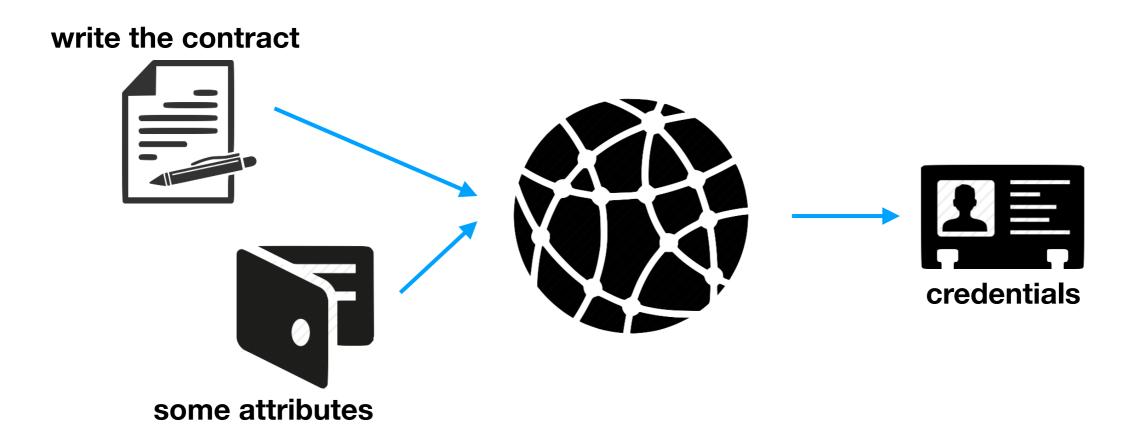


Issuing credentials through smart contracts



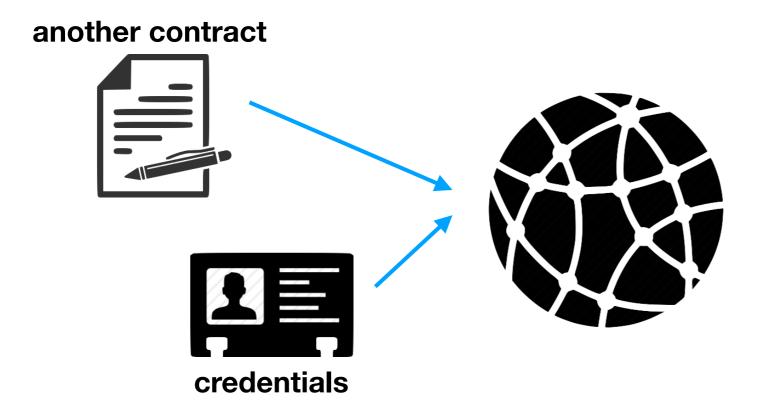


Issuing credentials through smart contracts





Issuing credentials through smart contracts





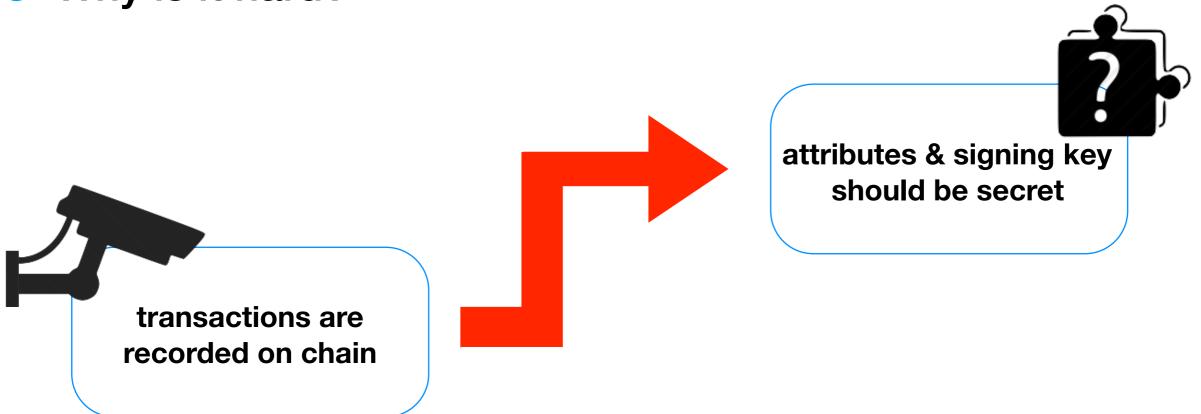
Why is it hard?



In a decentralised setting



Why is it hard?



In a decentralised setting

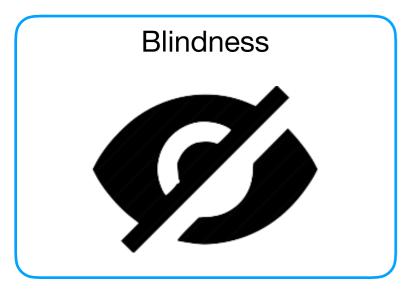


Why is it hard? attributes & signing key should be secret transactions are recorded on chain credentials showing should be unlinkable

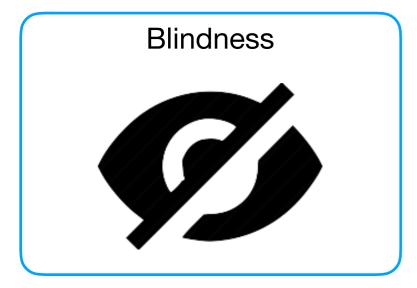
In a decentralised setting

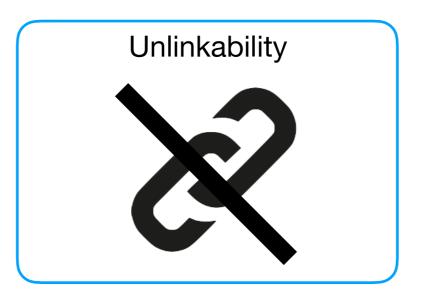




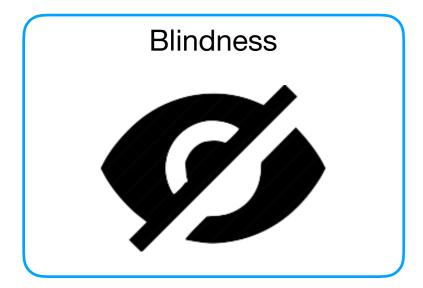


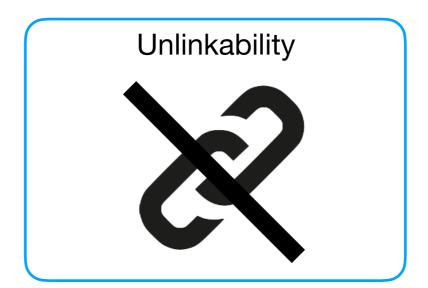






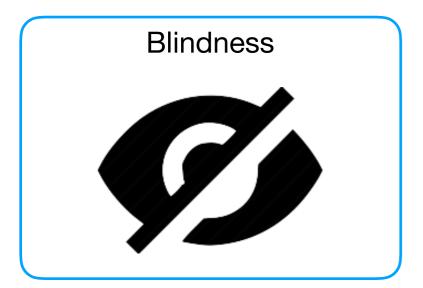


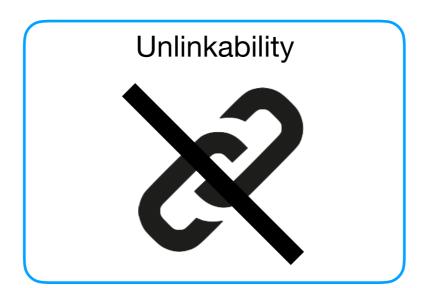




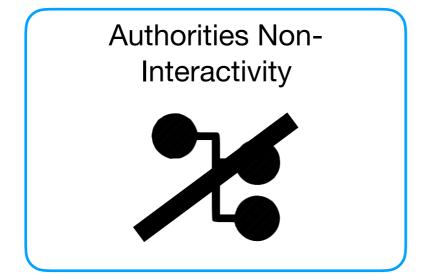




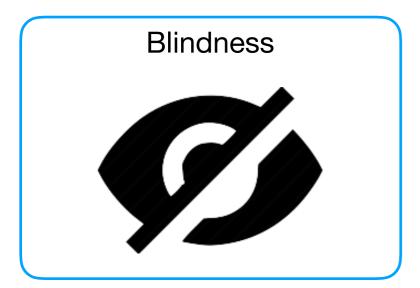


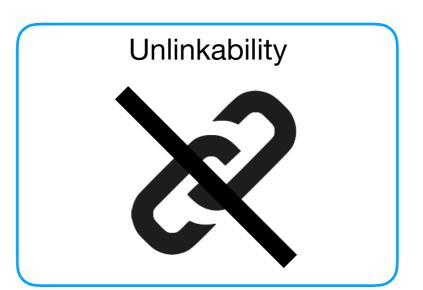




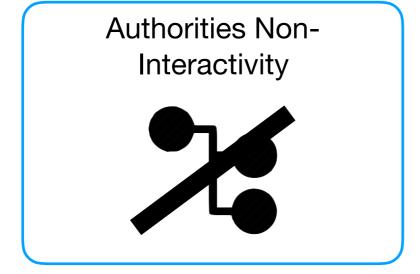
















So we built Coconut





What is Coconut?



What is Coconut?

Contribution I

Coconut credentials scheme





What is Coconut?

Contribution I

Coconut credentials scheme



Contribution II

Coconut smart contract library & example of applications



A. Primitives evaludti \overline{on} (g_1^{κ})

Then, Three ysight tulneir subtemos thes obtains implemented to in systems listuzingantdebtxild aryzoolkhravitsdpetlibboolf] shoodvibpolitha[2] ~ The buildinean pairing morked were a Bahreitoe Neathning [st. 7] curve, using

$$\pi = \mathrm{PK}\{(k) : \nu = \left(g_1^k\right)^{UUID} \ \land \ \kappa = \alpha\beta^k\}$$

Adding ν to L prevent a citizen to vote twice during the same campaign (prevent double spending), while the proof π ensures that ν has been

C. Mapping ai

Alberto: @C can also be C. of CoCoNut w actually built a

A. Pri

The using

destidention (axumos) of bout bilinear pairing works election election de la company de la compan OpenSSL as the arithmetic backend tesust of 10,000 measured on an Octa-core as an open-source project on GitHub. 2. 6CHz Intal Vacan

TABLE II: Coi

³https://github.c and 3 time faste aggregation of l

Table II sho size of each ex presented in fig. of signing author message on wh that in practice therefore set to ? 132 bytes. The

ask a-signature eand San Holard 10 Avention Using L described the secure I

uthoritian

his table of on hiersaulten likon

an exclame sage

attrorities able shows Bo

Then, Threey signature ir subtemo the obtains implemented to in sytemon listuzingantdebtxild aryzoolkhravitsdpetlibboolf] shoodvibpolitha[2] ~ The buildinean pairing morked were a Bahreitoe Neathning [st. 7] curve, using

$$\pi = \mathrm{PK}\{(k) : \nu = \left(g_1^k\right)^{UUID} \ \land \ \kappa = \alpha\beta^k\}$$

Adding ν to L prevent a citizen to vote twice during the same campaign (prevent double spending), while the proof π ensures that ν has been built from a signed private key k.



Alberto: @C can also be C. of CoCoNut w actually built a

A. Pri

The using

bilinear pairing works irio III curve using enionisactione lagame OpenSSL as the arithmetic backend. We have released the code an Octa-core as an open-source project on GitHub. 2. 6CHz Intal Vacan

TABLE II: Coi

³https://github.c and 3 time faste aggregation of l

Table II sho size of each ex presented in fig. of signing author message on wh that in practice therefore set to ? 132 bytes. The

ask a-signature eand San Holard 10 Avention Using L described the best gheseoder I his table of on

uthoritias

tikiersætttén labon an exclare sage

attrocrities able shows Bo

doed densition (axuaes), of bout - a

Then, Threey signature ir subtemo the obtains implemented to in sytemon listuzingantdebtxild aryzoolkhravitsdpetlibboolf] shoodvibpolitha[2] ~ The buildinean pairing morked were a Bahreitoe Neathning [st. 7] curve, using

$$\pi = \mathrm{PK}\{(k) : \nu = \left(g_1^k\right)^{UUID} \ \land \ \kappa = \alpha\beta^k\}$$

Adding ν to L prevent a citizen to vote twice during the same campaign (prevent double spending), while the proof π ensures that ν has been built from a signed private key k.

C. Mapping authorities to blockchain nodes

Alberto: @George, Describe how the CoCoNut authorities can also be Chainspace nodes (to make clear the poter) of CoCoNut when doordy_rolated to blockchaines actually built a

A. Pri

The using

destidention (axumes), of out bilinear pairing works rehachdenerhoodkingestionedageme OpenSSL as the arithmetic backens esust of 10,000 measured on an Octa-core as an open-source project on GitHub. 2. 6CHz Intal Vacan

TABLE II: Coi

³https://github.c and 3 time faste aggregation of l

Table II sho size of each ex presented in fig. of signing author message on wh that in practice therefore set to ? 132 bytes. The

describe and the second I

tikipusautės Ihbor

an exclame sage

atthor Itlesable shows Bo

ask arsignature Waehaigrald la vanture ususer de hlis vizabite) oli (pw Then, Threey signature ir subtemo the obtains implemented to in sytemon listuzingantdebtxild aryzoolkhravitsdpetlibboolf] shoodvibpolitha[2] ~ The buildinean phiring now orkaluse er a Bahreitoe Neathning [st. 7] curve, using

$$\pi = \mathrm{PK}\{(k) : \nu = \left(g_1^k\right)^{UUID} \ \land \ \kappa = lpha eta^k\}$$

Adding ν to L prevent a citizen to vote twice during the same campaign (prevent double spending), while the proof π ensures that ν has been built from a signed private key k.

C. Mapping authorities to blockchain nodes

Alberto: @George, Describe how the CoCoNut authorities can also be Chainspace nodes (to make clear the poter)

of CoCoNut when deeply related to blockchains) ince weard lower and the relation proof actually built all of Character at the packeture described the section of the section of

A. F

usin

bilin hill checking ention legetime dageme Ope red on an Octa-core as an open-source project on GitHub.

TABLE II: Coi

³https://github.c and 3 time faste aggregation of l

Table II sho size of each ex presented in fig. of signing author message on wh that in practice therefore set to ? 132 bytes. The ask arsignature

author Tiesable show Bo destanded designation (attended) (about - (

³https://github.c

and 3 time faste

aggregation of l

size of each ex

presented in fig.

of signing author

message on wh

that in practice

Table II sho

Then, Threey signature ir subtemo the obtains implemented to in sytemon listuzingantdebtxild aryzoolkhravitsdpetlibboolf] shoodvibpolitha[2] ~ The buildinean pairing morked were a Bahreitoe Neathning [st. 7] curve, using

$$\pi = \mathrm{PK}\{(k) : \nu = \left(g_1^k\right)^{UUID} \ \land \ \kappa = \alpha \beta^k\}$$

Adding ν to L prevent a citizen to vote twice during the same campaign (prevent double spending), while the proof π ensures that ν has been built from a signed private key k.



nociteties.

C. Mapping authorities to blockchain nodes

Alberto: @George, Describe how the CoCoNut authorities can also be Chainspace nodes (to make clear the poter) of CoCoNut when deeply related to blockchains incentional like a signature actually built all o

therefore set to ? 132 bytes. The ask arsignature Chairmann a characteristic described the secure described the secure is

GOTTO TALLENOT. THE SALLO Show BO using the two crypanily spins the most cand believed the tipe tipe the two crypanily spins the matter than the transfer of the court of bilinear pairing works techtothere checker werking en onlagetime stageme OpenSSL as the arit as an open-source project on GitHub. 2. 6CHz Intal Vacan



System Overview

Threshold authorities







System Overview

Threshold authorities

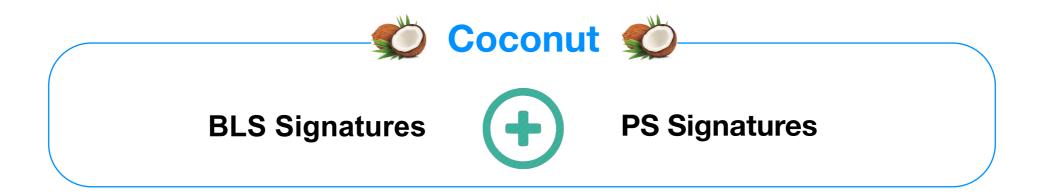


only t shares



Coconut Credentials Scheme

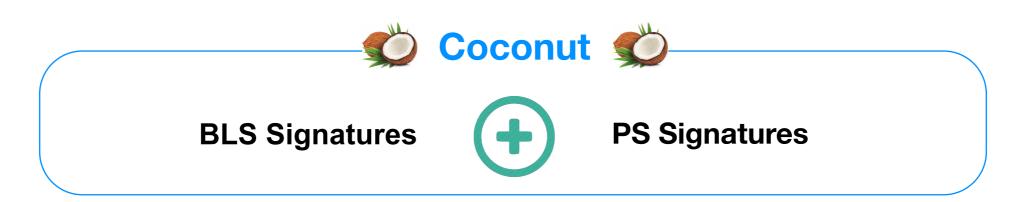
From where do coconuts come from?





Coconut Credentials Scheme

From where do coconuts come from?



What do they look like?

take an attribute: m compute: $h \leftarrow H(c_m)$ signature: $\sigma \leftarrow (h, h^{x+my})$ & secret key: (x, y)

The listu built Cocc Cor repeat t times ;

The listu

C. Mapping authorities to blockchain nodes

Alberto: @George, Describe how the CoCoNut authorities can also be Chainspace nodes (to make clear the potential

Adding vfto LCprevent havitizen plo votate vico v

V. EVALUATION

messa

that i

C. MappingPauthavitiesvtduklaskchain nodes

Alberto The signature section to hash been Cimplementedities python can also using Chaintwoccrypolelibraries abethibe (1) than prophibia [2]. The of Cochilinear pairing works over the advertock with the course, using a country of the second inverted ackiend. Works cropers that the adjusting structure of the code of a singular concherge of the code of t

A. PORTESTE ANTIVERS THE METAL DESCRIPTION OF THE ANTI-LEGAL ANTI-LEGAL PROJECT OF STATES OF THE CONSERVE STRUCTURE OF STATES OF THE CONSERVE STRUCTURE OF STATES OF THE S

Alberto: @George, Describe how the CoCoNut authorities SNIMI The UUII parameter uniquely identifies the petit Contract the orceres oparamenter holds, the laitizen's byotesh linitialization and NO: and the felds owner and verifier respectively hold the public key of the third party exeating the petition and of 1) authorities issuing the credentials. In order to sign the petition, the users compute a value ν as follows. V. EVALUATION

A. Primitives evaluati \overline{on} $(g_1^k)^{ar{U}ar{U}ar{I}ar{D}}$

Then, Threy signature ir solven to the option implemented to in python listuLinganthebtuild aryppoolkhravidedgetlibrooff] shoodvibglithat] v The buildliftean plairing meorladuever as Baltreitoe Nealthrijg [st.7] curve, using

$$\pi = \mathrm{PK}\{(k) :
u \in \mathfrak{O} \text{ for act info} \ \kappa = lpha eta^k\}$$

Adding ν to L prevent a citizen to vote twice during the same campaign (prevent double spending), while the proof π ensures that ν has been built from a signed private key k.

attributes

C. Mapping authorities to blockchain nodes

Alberto: @George, Describe how the CoCoNut authorities can also be Chainspace nodes (to make clear the potential of CoCoNut when diagnosis of company of the office of the control actually built approprietalist telles telles to the proprietable of the proprietable o

the examination of sack and specific policy of the signature speci

as an open-source project on Sithub. 10,000 measured on an Octa-core is Athach: faster are veltifyinged ignatures with alternatives Table I shows the mean (µ) and standard deviation (v.p. described in section section II.

that signing is much faster than verifying signal about the execution of each procedure described in section section II.

Each entry is the result of 10,000 measured on an Octa-core 33 Alberto: compare results (speed and size) with alternatives

Adding ν to L prevent a citizen to vote twice during the campaign (prevent double spending), while the proof π that ν has been built from a signed private key k.

C. Mapping authorities to blockchain nodes

Alberto: @George, Describe how the CoCoNut aut can also be Chainspace nodes (to make clear the p of CoCoNut when deeply related to blockchains), si

TABLE II: Communication complexity and crearsaction is ize art co

V. EVALUATION ³https://github.com/asonnino/coconut

and 3 time faster rimitines schemetion hidden messages). Also, aggregation of the signature at the attraction of the signature of the sig

Table II shows the two monumication is opposity and thous size of each probangaining of the signature Nachange, 178 curv presented in Great Theasemen of signing authorities (n) an message on white shows that in practice meistibe be therefore set to 32 bytes (forh 132 bytes. The highest transf

pr<u>esen</u>

of Sign

messat thatain

therefo

132Stay

ask **9**

that the

318 by

isend.awthranumbleased t sent the size of the and standard de Viation (v aldenesisse in section sec 1000 frae signaturenisan Oc igaratwheathe ther table names from the fact pures TO SECURITION OF THE PARTY OF T

he ashathathadhawes laand

AlAl

to se

Dell deskton computer 3 6GHz Intel Xeon. This table shows to good what is and stuff, not many solvers a hour getting hear

Application

Privacy-pre

that real oreen built to the sound of the key k. options, scores)

C. Mapping authorities to blockehim nodes $\stackrel{\cdot}{\nu} = \stackrel{\cdot}{g_1}$

Alberto: @George, Describe how the CoCoNut auth The UUID parameter uniquely identifies the petition of significant beauthorities (see, and ||m|| represents the size the osceres oparameter holds the leitizen's byotesh limitialized zeraetulalytheticase of thas petition, the entire are and NO; and the felds owner and verifier respectively hold the public key of the third party creating the petition and of authorities issuing the credentials. In order to sign the petition, the users compute a value ν as follows. V. EVALUATION

A. Primitives evalud $ti\overline{\sigma n} \left(g_1^k\right)^{\widetilde{U} \widetilde{U} \widetilde{I} \widetilde{D}}$

Then, Three y significant representation of the continuous and the continuous states of the cont listusingarthebtsild arypoolkhravitsdgetlibobf shootvibplithat \ \ \ \ The buildliftean phicing morked over a Bahreitoe Neathnia [st 7] curve, using

$$\pi = \mathrm{PK}\{(k) : \nu = \left(g_1^k\right)^{UUID} \ \land \ \kappa = \alpha \beta^k\}$$

Adding ν to L prevent a citizen to vote twice during the same campaign (prevent double spending), while the proof π ensures that ν has been built from a signed private key k.

·C: Mapping authorities to blockchain nodes

Alberto: @George, Describe how the CoCoNut authorities can also be Chainspace nodes (to make clear the potential

using the styrn cryparte brarian ever the med work in the lient lient essages bilinear pairing works over a Barreto-Naehrig [17] curve, using OpenSSL as the arithmetic backend. We have released the code check he time it takes to hear back from t authorities.

as an openimitive epropertion GitHub.3.

petition rability and create petition as the dard lepisition (2) python tor exceptione of cooling possible and september 1 period of the exception creator exampling of wach proceduraries grided in seation best joing !!. The that asigning pen-muche fortgeethan configured signatures (about

15 times fastern for the medame working condition and several of the execution of each procedure described in section section II. Each entry is the result of 10,000 measured on an Octa-core Dell desktop computer, 3.6GHz Intel X Legite table shows that signing is much faster than verifying signatures (about

15 times faster for the scheme working on clear messages,

and 3 time flatibility for the story and 3 time flatibility for the story and 3 time flatibility for the story for the story and 3 time flatibility for the story for the story for the story flatibility for the story flatibility for the story flatibility for the story flatibility fl aggregation of keys and signatures are extremely efficient

Table II shows the communication complexity ar sizeNutbeach authoriangenin vour rivethe 3:18 hature scher resented in fig. 1. The complexity is expressed as the h mes age signature that give his signature is the hash of the actual message at the hash of the actual message at the hash of the actual message at the hash of the therefore set to 32 bytes (for SHA-2). The size of a signal 132 Staytes: The hidden test agas action size appears when the ask a signification a hidden message (This comes from t that the proof π_v associated with the message is approximately entry signature 318 bytes; the proof π_v is only 157 bytes.

TABLE II: Communication complexity and transaction

³https://github.com/asonnino/coconut

smart

and 3 time faster for the scheme on hidden messages). ag rhaippensys and signatures are extremely efficient Table II shows the communication complexity are size of each exchange involved in the signature scheme presented in fig. 1. The complexity is expressed as the n signing authorities (n), and ||m|| represents the size message on which the user wish to obtain a signature that in practice m is the hash of the actual message, therefore set to 32 bytes (for SHA-2). The size of a signal **N2** bytes. The highest transaction size appears when the of Cock to the control of the contr trto: Bano, tost system only 153 bytes: morny). ene tropic bytes: Helphy company for any blind signatures) ar

Shappens everyith Related Works

3https://github/cbm/acconindacedscon WITH RELATED WORKS

Alberto: discuss crypto related works Alberto: compare results (speed and size) with altern see why it is cool stuff; not many scheme have actuall

³https://github.com/asonnino/coconut



Performance

What is out there?





Performance

What is out there?

The Coconut cryptographic library

Python & Timing benchmark



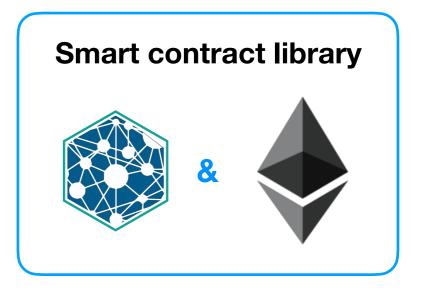


What is out there?

The Coconut cryptographic library

Python & Timing benchmark





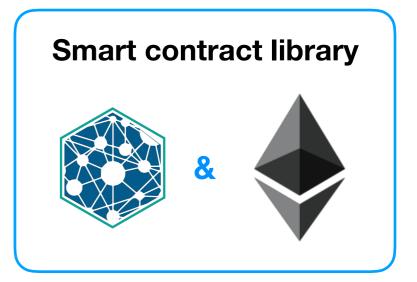


What is out there?

The Coconut cryptographic library

Python & Timing benchmark





Applications

Coin tumbler E-Petition (CRD proxy distribution)

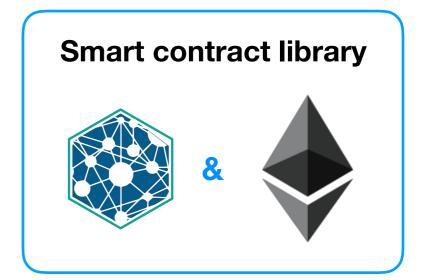


What is out there?

The Coconut cryptographic library

Python & Timing benchmark





Applications

Coin tumbler E-Petition (CRD proxy distribution)

Everything is released as open source software

https://github.com/asonnino/coconut





How fast is Coconut?

	Operation	μ [ms]	$\sqrt{\sigma^2}$ [ms]
	PrepareBlindSign	2.633	$\pm \ 0.003$
sign	BlindSign	3.356	± 0.002
	Unblind	0.445	± 0.002
	AggCred	0.454	$\pm~0.000$
	ProveCred	1.544	$\pm \ 0.001$
verify	VerifyCred	10.497	± 0.002

signing is fast, verifying takes 10ms



What is the size of the credentials?

2 Group Elements

No matter how many attributes...

No matter how many authorities...



How does Coconut scale?

	Number of authorities: <i>n</i> , Signature size: 132 bytes Transaction complexity size [B]		
	Signature on public attribute:		
	• request credential	O(n)	32
	2 issue credential	O(n)	132
	3 verify credential	O(1)	162
	Signature on private attribute:		
issue	• request credential	O(n)	516
'	2 issue credential	O(n)	132
verify	3 verify credential	O(1)	355

Signing scales linearly, verifying is constant time



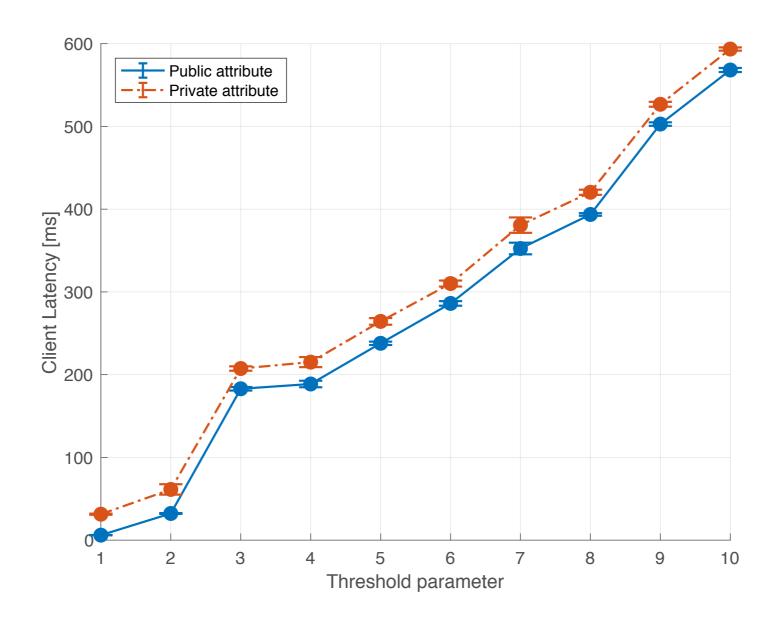
Did you evaluate it in the real world?



pick 10 locations across the world



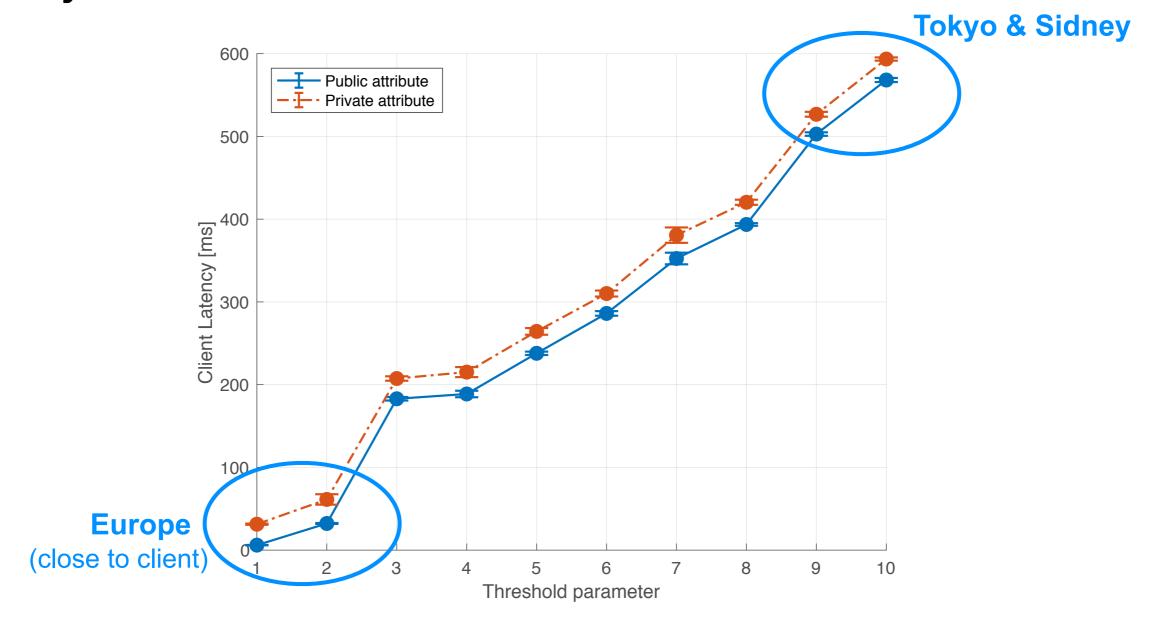
Did you evaluate it in the real world?



client latency VS number of authorities



Did you evaluate it in the real world?



client latency VS number of authorities



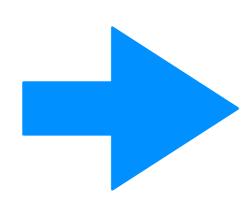
What else is in the paper?

Full cryptographic scheme

Smart contract library evaluation

Coin tumbler, CRD proxy applications

Applications evaluation and benchmarking



Coconut: Threshold Issuance Selective Disclosure Credentials with Applications to Distributed Ledgers

Alberto Sonnino University College London Mustafa Al-Bassam University College London Shehar Bano University College London

George Danezis University College London The Alan Turing Institute

Abstract

20 Feb 2018

arXiv:submit/2158644

We present Coconut, a novel selective disclosure credential scheme supporting distributed threshold issuance, public and private attributes, re-randomization, and multiple unlinkable selective attribute revelations. Coconut can be used by modern blockchains to ensure confidentiality, authenticity and availability even when a subset of credential issuing authorities are malicious or offline. We implement and evaluate a generic Coconut smart contract library for Chainspace and Ethereum; and present three applications related to anonymous payments, electronic petitions, and distribution of proxies for censorship resistance. Coconut uses short and computationally efficient credentials, and our evaluation shows that most Coconut cryptographic primitives take just a few milliseconds overage, with verification taking the longest time (10 milliseconds).

1 Introduction

Selective disclosure credentials [15, 17] allow the issuance of a credential to a user, and the subsequent unlinkable revelation (or 'showing') of some of the attributes it encodes to a verifier for the purposes of authentication, authorization or to implement electronic cash. However, established schemes have shortcomings. Some entrust a single issuer with the credential signature key, allowing a malicious issuer to forge any credential or electronic coin. Other schemes do not provide the necessary re-randomization or blind issuing properties necessary to implement modern selective disclosure credentials. No existing scheme provides all of threshold distributed issuance, private attributes, re-randomization, and unlinkable multi-show selective disclosure.

The lack of full-featured selective disclosure credentials impacts platforms that support 'smart contracts', such as Ethereum [40], Hyperledger [14] and Chainspace [3]. They all share the limitation that ver-

ifiable smart contracts may only perform operations recorded on a public blockchain. Moreover, the security models of these systems generally assume that integrity should hold in the presence of a threshold number of dishonest or faulty nodes (Byzantine fault tolerance); it is desirable for similar assumptions to hold for multiple credential issuers (threshold aggregability).

Issuing credentials through smart contracts would be very desirable: a smart contract could conditionally issue user credentials depending on the state of the blockchain, or attest some claim about a user operating through the contract—such as their identity, attributes, or even the balance of their wallet. This is not possible, with current selective credential schemes that would either entrust a single party as an issuer, or would not provide appropriate re-randomization, blind issuance and selective disclosure capabilities (as in the case of threshold signatures [5]). For example, the Hyperledger system supports CL credentials [15] through a trusted third party issuer, illustrating their usefulness, but also their fragility against the issuer becoming malicious.

Coconut addresses this challenge, and allows a subset of decentralized mutually distrustful authorities to jointly issue credentials, on public or private attributes. Those credentials cannot be forged by users, or any small subset of potentially corrupt authorities. Credentials can be rerandomized before selected attributes being shown to a verifier, protecting privacy even in the case all authorities and verifiers collude. The Coconut scheme is based on a threshold issuance signature scheme, that allows partial claims to be aggregated into a single credential. Mapped to the context of permissioned and semi-permissioned blockchains, Coconut allows collections of authorities in charge of maintaining a blockchain, or a side chain [5] based on a federated peg, to jointly issue selective disclosure credentials.

Coconut uses short and computationally efficient credentials, and efficient revelation of selected attributes and verification protocols. Each partial credentials and the



Limitations & Future Works

Would you like to contribute?

Limitation I

Adding and removing authorities is complicated.

Can we do better than re-running the key generation algorithm?



Limitations & Future Works

Would you like to contribute?

Limitation I

Adding and removing authorities is complicated.

Can we do better than re-running the key generation algorithm?

Limitation II

Current key generation algorithms are complex to implement. Can we design a key generation algorithm for blockchains?



Limitations & Future Works

What is the next milestone?

A general framework allowing nodes to execute any kind of threshold cryptography?



Conclusion

What did we talk about?

Contribution I

Coconut credentials scheme



Contribution II

Coconut smart contract library & example of applications

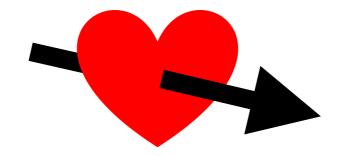




Conclusion

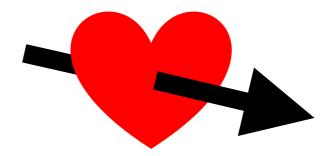
Main take-aways

Threshold issuance



Sweet for blockchains

Randomizable



Multi-use & unlinkability



Thank you for your attention

Alberto Sonnino alberto.sonnino@ucl.ac.uk https://sonnino.com

https://github.com/asonnino/coconut





