#### **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



## Privacy-preserving credentials

#### ...without a single issuer



#### **Blockchains**











### Byzantine

### Hard to build



### No failure

### Conventional

Expensive

Cheap









#### **The Authors**



Alberto Sonnino



Mustafa Al-Bassam



**Bano Shehar** 



Sarah Meiklejohn



**George Danezis** 

#### **Challenges in blockchains**



#### **Challenges in blockchains**

#### Can we issue credentials in this setting?



### What are we trying to do?

Issuing credentials through smart contracts



### What are we trying to do?

Issuing credentials through smart contracts



### What are we trying to do?

Issuing credentials through smart contracts



### What are we trying to do?

Issuing credentials through smart contracts



### What are we trying to do?

The more traditional setting



#### ... but without any central authority

# Distributed settings



### What are we trying to do?

#### Why is it hard?



#### In a decentralised setting

### What are we trying to do?



#### In a decentralised setting

### What are we trying to do?



#### In a decentralized setting

#### Introduction

#### Introduction



#### Introduction



#### Introduction



#### Introduction





#### Introduction





#### So we built Coconut



#### Introduction

#### Related works

Scheme	Blindness	Unlinkable	Aggregable	Threshold	Signature Size
[39] Waters Signature	×	×	0	×	2 Elements
[26] LOSSW Signature	×	×	$\widehat{}$	×	2 Elements
[8] BGLS Signature	×	×	$\bullet$	$\checkmark$	1 Element
[15] CL Signature	$\checkmark$	$\checkmark$	0	×	O(q) Elements
[ <b>31</b> ] Pointcheval <i>et al</i> .	$\checkmark$	$\checkmark$	$\widehat{}$	×	2 Elements
Coconut	1	1	•	$\checkmark$	2 Elements



not aggregable
sequentially aggregable
user-side aggregable
number of attributes

#### Introduction

• What is Coconut?

#### Introduction

- What is Coconut?
- **Contribution I**

**Coconut credentials scheme** 



#### Introduction

#### What is Coconut?

#### **Contribution I**

**Coconut credentials scheme** 



#### **Contribution II**

Coconut smart contract library & example of applications





A. Primitives evalue  $f_{i}\overline{\sigma n} (g_{1}^{\kappa})^{-1}$ 

Then, Tibeysightatuneir solutions thas dottion simplomented to in python listu Lingart debtailed arypoolkhravites dgetlipbo[bf] standy ibgglith [2]. The buildilifteen phiring neorkaluever and the itoe Neak-Iniga [st.7] curve, using

$$\pi = \mathrm{PK}\{\left(k
ight): 
u = \left(g_{1}^{k}
ight)^{UUID} \ \land \ \kappa = lpha eta^{k}\}$$

Adding  $\nu$  to L prevent a citizen to vote twice during the same campaign (prevent double spending), while the proof  $\pi$  ensures that  $\nu$  has been

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

A. Pri

The using bilinear pairing works over a Barreto-Naehrio II 71 curve, using OpenSSL as the arithmetic backend. We have released the code as an open-source project on GitHub. 2 (CUz Intel Year This table shows)

### TABLE II: Co

<sup>3</sup>https://github.c and 3 time faste aggregation of 1 Table II sho size of each ex presented in fig. of signing author message on wh that in practice therefore set to 2 132 bytes. The

ask a-signature

hiorsaulion labor

Jeta-coi

table ohow

and stahoigr & lot you and the state of the

described the section of the section

uthori tian

poter

Then, They sight their solutions that determined to be appreciated to be the solution of the s

$$\pi = \mathrm{PK}\{\left(k
ight): 
u = \left(g_{1}^{k}
ight)^{UUID} \ \land \ \kappa = lpha eta^{k}\}$$

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

C. Mapping authorities to blockchain nodes

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

A. Pri

The using bilinear pairing works over a Barreto-Naehrio II 71 curve, using OpenSSL as the arithmetic backend. We have released the code as an open-source project on GitHub. 2 (CUz Intel Year This table shows)

### TABLE II: Co

<sup>3</sup>https://github.c and 3 time faste aggregation of 1 Table II sho size of each ex presented in fig. of signing author message on wh that in practice therefore set to 3 132 bytes. The

ask a-signature

signatures labor

Jeta-coi

table ohow

and standing of 107 yearing Tryber Ar

described the sector of the se

uthoritian

poter

Then, They sight their solutions that determined the determined to be the solution of the solution of the listual of the solution of the solut

$$\pi = \mathrm{PK}\{\left(k
ight): 
u = \left(g_{1}^{k}
ight)^{UUID} \ \land \ \kappa = lphaeta^{k}\}$$

Adding  $\nu$  to L prevent a citizen to vote twice during the same campaign (prevent double spending), while the proof  $\pi$  ensures that  $\nu$  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  $C_0 C_0 N_{L} t$  when  $d_0 e^{-1} V_{-rolated}$  to blockchains. Since we

actually built a

A. Pri

The using bilinear pairing works over a Barreto-Naehrio 1171 curve, using OpenSSL as the arithmetic backend. We have released the code as an open-source project on GitHub. 2 (CUr Intel Year This table shows)



<sup>3</sup>https://github.c and 3 time faste aggregation of 1 Table II sho size of each ex presented in fig. of signing author message on wh that in practice therefore set to 2 132 bytes. The

ask a-signature

tionsaulton labor

Jeta-coi

table ohow

stehoigra lokavanta us use h

avective the section of the section

Then, They sight their solution that determined the determined to be the solution of the solution of the listual of the solution of the soluti

$$\pi = \mathrm{PK}\{\left(k
ight): 
u = \left(g_{1}^{k}
ight)^{UUID} \ \land \ \kappa = lpha eta^{k}\}$$

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

C. Mapping authorities to blockchain nodes



#### TABLE II: Co

<sup>3</sup>https://github.c and 3 time faste aggregation of 1 Table II sho size of each ex presented in fig. of signing author message on wh that in practice therefore set to 2 132 bytes. The

Alb	perto: @George, Describe how the CoCoNut authoritis:	therefore set to .
can a	ulso be Chainspace nodes (to make clear the poter	132 bytes. The
of Co	Const when deeply related to blockchains ince we	-ask a-signature
actua	ly built all of this to baye credentials in smart contracts	that the proof at he section I
	The signature scheme has been imported as all and the province the result of the province	11 Bythops, the 1 ed on 7an Octa-cot
	using the state of the praries pethol in the phi	at onisvizore ohow
A. I	Barreto-Nehrig	urve, úsing stigigisatures (hbø
<b>T</b> 1	ickend, we have release the schemency of the schemency of the scheme is the scheme of	ona lear an exal
usin	5.60HZ I a LUCOTIU	<b>EStable show</b> Bo
bilin	iring works over a Barreto-Nachrig 11/1 curve usingtion	hatters y about - (
Ope	as the grithmetic backend. We have released the code	Octa-core
as an	open-source project on GitHub. 2 (CII Intel Vern This to	able about

Then, They sight ture is solven to the option simplemented to in python listu Lingart debtailed arypoolk mavited get libro [bf] shoot inglith [2]. The build lifter phiring neorkaluever and the itoe Neak-miga [st.7] curve, using

$$\pi = \mathrm{PK}\{\left(k
ight): 
u = \left(g_{1}^{k}
ight)^{UUID} \ \land \ \kappa = lpha eta^{k}\}$$

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

C. Mapping authorities to blockchain nodes



#### TABLE II: Co

<sup>3</sup>https://github.c and 3 time faste aggregation of 1 Table II sho size of each ex presented in fig. of signing author message on wh that in practice therefore set to 2 132 bytes. The

Alberto: @George, Describe how the CoCoNut authorities 132 bytes. The can also be Chainspace nodes (to make clear the poter ask a-signature of CocoNut when deeply related to blockchains) incentional lange when a particle in the prover and an end of the prover an end of th actually built all Equinactionse de la place du ce des de la place de la Gitatoso table of on icic charender an be sectivity section labor A. Primitives eva nd We have release of the corten evalue The signature BOTTE BLOCK I TIES Able show Bo using the two cryponia The mouth (faster of that doer den at 1921) (fortit - ( bilinear pairing works brockenscherkonstingentionlagetime Hageime OpenSSL as the arit ured on an Octa-core as an open-source project on GitHub.". 2 (CUT Intel Veen This table above



EVALUATION 318 bytes; the proof  $\pi_v$  is or rim Over view of the Malet petition and then the above explored the option implemented to prove the option of the main provide the providence of the proviBuild a system of the state of an phicing workdower a Baneitoe Nathriga [st7] curve, using  $U = PK\{(k) : \nu = (g_1^k)^{UUID} \land \kappa = \alpha \beta^k\}$ <sup>3</sup>https://github.com/asonninp/cocor and 3 time faster for the selfe aggregation of keys and signat Table II shows the comm  $\nu$  to L prevent a citizen to vote twice during the same size of each exchange introduce on (prevent double spending), while the proof  $\pi$  ensures selfenand fuse these token to steal all the money. The comple of signing authorities (n) and honest authorities ne buffer. The threshold proper of of the proper of the in practice m is the the ser we have blockchain nodes that the nady ersary needs to corrupte ato cast 32 bytes (for \$1 orities for this attack fo happen. If 132 bytes. The highest transac prevention TAPESI A CONTRACTOR OF A CONTRACTOR O Ging the ev al ohowancy vo terve ver a Barreto Ave hrig A chinicit cluster testan beer itives eva HE HERE BER CANGE TRANSFERRE We have released the case is the presence of the second of the second of the second of the case of the second of t as the apithmetic

### **System Overview**

Threshold authorities



Users need to collect only (2f+1) shares

### **Coconut Credentials Scheme**

Cryptographic primitives



#### **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: mcompute:  $h \leftarrow H(c_m)$ signature:  $\sigma \leftarrow (h, h^{x+my})$  & secret key: (x, y)

	The						
	listu						
	buit	C. Mapping auth	orities to	o blockchain	nodes		
Cocc		Alberto: @Geo	orge, Des	scribe how t	he CoCoNi	ıt authori	ities
		can also be Cha	inspace	nodes (to n	nake clear	the poten	itial
	Adding	oft@ DCprevent ha	nitideeplo	votatedice	dulrialg hbien	sameince	we
	campaig	n (prely entrolbuble	specialing	ghawhileethe	iprdofimsem	streontra	CIS.
	that $ u$ ł	as been built fron	n a signe	ed private ke	ey <b>k</b> .		of sic
			V.	EVALUATIO	DN		messa
	С. Мар	AngPauthavitsesvti	ti leheokci	hain nodes			that i
	Albe.	. The signature:	scheme	mashebeen C	methe	ditims pyt	there
repeat	can als	usingCtheinspacer	ypolelibri	tariesa betläbe	(11]thanpoor	hiba[2]. '	the t
<b>t</b> times ;	of CoC	bilinean pairing w	orksløtær	t Barbetock	aehrig finc	curve, us	ask a
	actually	openss Dashhera	Athmetic	edaokiend.iM	lenharte crehe	set the c	
		as angonaeursouche	emeojhat	me Enit Hup!	emented in	python	
•	usir	g that tenasharing at	n <b>etwaria</b> s	Appendix Stah	and applibt	$bal(\sqrt{be^2})$	) of $l_{l_0}$
	bili A Drive	the pairing on orke	aeteproB	edrato Asochi	bed[in7secu	sa, secitigi	1  II.
	A.	ESSE analyse arrithm	ncessalleac	kandowe ha	veuredaser	theOodec	Bre Sy
	The	i statutskuppenpin	phats 1, be	<b>ECHop</b> letele	extedn.inThp	that the sho	ows
	us ng f	th two is burying a line bu	ewiks (fp)st	thist stand chood	ficiplicated and	vides) (aft	out
	bilindar	preiringen offkstærhe	pro Beetue	the blace horest	ingecuivels	actinge stag	ges.
	Ope <b>ES</b> S	L antilye iarithmetsi	lbækto	d)Wenhaasuu	eleasochthe	coalecore	
	as a <b>De</b>	edeskupec properte	nøb.60H	Indintel Xeo	n. This tab	le shows	to

Alberto: @George, Describe how the CoCoNut authorities  
The UISDID parameter uniquely (identifies the petit of Penific of Sign  
the occors oparameter helds y the leitizen's brotesh limit, a street to be stated  
the occors oparameter helds y the leitizen's brotesh limit, a street to be stated  
zerad tuln y the it asset of thas petition, the leantions are and bott there is a stated of the state of the public key of the third party creating the petition and of 1  
authorities issuing the credentials. In order to sign the petition, that is the there is a value 
$$\nu$$
 as follows.

EVALUATION A. Primitives evaluation  $(g_1^k)^{\overline{UUID}}$ 

Then, They signature is solven to the option simplement of the python listu singantebtuild arypoolkbravies dgetlibo [1] shook ibglith [2] v The buildifican phiring morked wever as Balanci to eNeatennia [st7] curve, using

$$\pi = \mathrm{PK}\{(k): 
u = \mathbf{copt}^{UUID}_{k}$$
  $\kappa = lpha eta^k \}$ 

Adding  $\nu$  to L prevent a citizen to vote twice during the same campaign (prevent double spending), while the proof  $\pi$  ensures that  $\nu$  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 the signification of the significat actually built all of this terrange last the new fond the actually built all of the state last of the second working don't the second s

The second and the experimental second and balls by the second of the experimental second and the experimental second and balls been implemented in py too been were second and balls been implemented in py too been were second and balls been implemented in py too been were second and balls been implemented in py too been were second and balls been implemented in py too been were second and balls been implemented in py too b

A. Primitive lipeaturation works over a Barreto-Naehright verifying as provide signatures in the signature source of the source of the source of the signature source of the source Al Al OpenSSL as the arithmetic backena. We have released in section section desktop computer, 3.60 Hz intel Xeon. This table shows as an open-source project on Cittud 10,000 measured on an Octa-core is Albach faster offare vetifying dignatures (about ernatives)

Dell deskton computer 3 6GHz Intel Xeon This table shows to account it is acclustiff, not many scheme have actually hear

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

#### C. Mapping authorities to blockchain nodes

k¶ § that the 318 by 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 mansattibnisize art co

V. EVALUATION <sup>3</sup>https://github.com/asonnino/coconut

and 3 time\_faster fonitinessevenuetion hidden messages). Also,

aggregation of the signal the attension of the strength affinishented in Table II aboves the two menupoic this are compared to be it and the lib [

size of each man gain my wed in the signature schenge, 17\$ curv presented in OpenSTheasoner of signing authoritiesemsour message on what the shows that in practice meisthe be therefore set to 32 bytes (for 132 bytes. The highest transf

sign



send.awtheavertheased t entsuthe size of the and signature deviation ( $\mathbf{v}$ aldeserises in section sec 600 measureturenisan Oc garaten here table amesefremthesignatures STANDING REAL WAR ppliction II. d ourse Wingcore le ashie the to de aves a a a a



A. Primitives evalue 
$$(g_1^k)^{UUII}$$

Then, They signature is solven the determinant per anter drain system listusingantiebtid arypoolknaviesdgetliboolf] shodvibglithat ]. The buildliftean phiring works were a Batter toe Natel mig [st7] curve, using

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

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

#### C. Mapping authorities to blockchain nodes

OpenSSL

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

astanlypenipenic evaluation Table T shows the mean of standard deviation (v/or 2) of the standard deviation of the standar in the message is approximately in the message is a possible in the message is approximately i ton lorge the Porta Barry in Nachings With Starsing spikes Each entremestic of poly and the stand of the stand of the stand of the stand of the source of the stand with t Definitives a white the standard deviation  $(\sqrt{\sigma^2})$  of VI. VI. VI. VIdsintimae taxcicitypourbeanesnpewickingeanechonder seathor oceduite neseriped an action of the antiaean (μ) and standard deviation (γ στιστικόν vs t – askin steriounes) or Barreto-Naciana entity us veneuses ut of 10,000 measured on an Octa-core bilinear pair penSSL as the arithmetic backend. We have described in section section that the fittle in deckers do hear backetment werkfor these res Each entry is the result of 19,000 measured on an **Decker** a of Hz Intel Xeon. This table shows to see why it is cool stud an open-source project on GitHub, that signing is much faster than verifying signatures (about (speed and size) with alter Dell desktop computer, 3.0GHz the Xeon. This table shows to see why it is cool stud Table Lishows the mean (u) and standard devictions or features (about the scheme working on the states) with a signing is much the scheme working on the states of the states of the states of the second states of the shows the states of the states of the shows the second states of the shows the states of the states of the shows the states of the states of the shows the states of the states of the states of the states of the shows the states of the states of

TABLE II: Communication complexity and transaction size.

#### <sup>3</sup>https://github.com/asonnino/coconut

and 3 time faster for the scheme on hidden messages). Also, aggregation of keys and signatures are extremely efficient.

Table II shows the communication complexity and the size of each exchange involved in the signature scheme, as presented in fig. 1. The complexity is expressed as the number of signing authorities (n), and ||m|| represents the size of the message on which the user wish to obtain a signature. Note that in practice m is the hash of the actual message, and is therefore set to 32 bytes (for SHA-2). The size of a signature is 132 bytes. The highest transaction size appears when the user signature on a hidden message. This comes from the fact

### Application

#### Privacy-pr

that reatoren built of built of a sogned private key k.

options, scores)

C. Mapping authorities to block than nodes  $\nu = (g_1)^{\ell}$ smart Alberto: @George, Describe how the CoCoNut auth The UUUU parameter uniquely identifies the petit  $\mathcal{G}$  of the significant best in the size the scores oparanty for holds the sitizen's brotesh (initialized) zero tulnly the it as of thas petition, the entires are only not the and NO; and the felds owner and verifier respectively hold the public key of the third party creating the petition and  $qf_{1}$ authorities issuing the credentials. In order to sign the petition, the users compute a value  $\nu$  as follows. V. EVALUATION

#### A. Primitives evaluation $(g_1^k)^{UUID}$

Then, They signature is scheme the option implemented to a system listusingarthebtid arypoolknaviesdgetlibobf] showinglithat by The buibilifrean priving neorkal were a Bahreitoe Neathning [st7] curve, using

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

Adding  $\nu$  to L prevent a citizen to vote twice during the same campaign (prevent double spending), while the proof  $\pi$  ensures that  $\nu$  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 5tymery partition of the state of the state of the states of t

bilinear pairing works over a Barreto-Naehrig [17] curve, using OpenSSL as the arithmetic backend. We have released the code

as an openinstituce qualitation GitHub.<sup>3</sup>. petition Table I. S. Create petition destandard heriation (a) python

tor exercise the straight of t creator exercition of wach proceeder and section besite of 21. The that asigningpen-muse foresection configing signatures (about

15 times faster fors the scheme ) working on deteriances ages?) of the execution of each procedure described in section section II. Each entry is the result of 10,000-measured on an Oeta-core Dell desktop computer, 3.6GHz Intel X Lecipertable shows that signing is much faster than verifying signatures (about 15 times faster for the scheme working on clear messages,

and 3 time flat Bil foil: the scheman cos bidden timessages). aggregation of keys and signatures are extremely efficient

Table II shows the communication complexity at sizeNufbeach automaingen in Vierveue Hirehersighature scher presented in fig. 1. The complexity is expressed as the h mes age signature wish  $\binom{0}{n}$  tain a signature that it entry signature is the hash of the actual message is the signature is the hash of the actual message is t therefore set to 32 bytes (for SHA-2). The size of a signal 1328 hypers re The hild ighest straps action size appears when the ask a signature on a hidden message (This comes from t that the proof  $\pi_s$  associated with the message is approximately signature 318 bytes; the proof  $\pi_v$  is only 157 bytes.

TABLE II: Communication complexity and transaction

#### <sup>3</sup>https://github.com/asonnino/coconut

and 3 time faster for the scheme on hidden messages). agerhappensys and signatures are extremely efficie Table II shows the communication complexity ar size of each exchange involved in the signature scher 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 Coast asking the provide the boundary of the stand of the stand of the stand of the second stand of the many scheme have active the proof  $\pi_s$  associated with the message is approximately the second stand of the second stand stand stand of the second stand st

Is by less the proof of the only 159 by test in only 159 by test i check he time it takes to hear back from t authorities.

#### Shappens every ith Related Works B

#### <sup>3</sup>https://github/cbm/accom/npAcedSign/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

<sup>3</sup>https://github.com/asonnino/coconut

#### Performance

• What is out there?



#### Performance

#### What is out there?

The Coconut cryptographic library

Python & Timing benchmark



#### Performance

#### What is out there?

The Coconut cryptographic library

Python & Timing benchmark





#### Performance

#### What is out there?

The Coconut cryptographic library

Python & Timing benchmark





#### **Applications**

Coin tumbler E-Petition (CRD proxy distribution)

#### Performance

#### 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



#### Performance

#### How fast is Coconut?

15]
,
1
1
)
1

signing is fast, verifying takes 10ms

#### Performance

What is the size of the credentials?

**2 Group Elements** 

No matter how many attributes...

No matter how many authorities...

#### Performance

#### How does Coconut scale?

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

Signing scales linearly, verifying is constant time

#### Performance

Did you evaluate it in the real world?



pick 10 locations across the world

#### Performance

#### Did you evaluate it in the real world?



#### client latency VS number of authorities

#### Performance

#### 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

Mustafa Al-Bassam

University College London

Alberto Sonnino University College London Shehar Bano University College London

George Danezis University College London The Alan Turing Institute

#### Abstract

20 Feb 2018

CR

cs.

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 on average, 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?



**Coconut credentials scheme** 

#### **Contribution II**

Coconut smart contract library & example of applications



#### Conclusion

#### Main take-aways







Multi-use & unlinkability



### Thank you for your attention Questions?

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



https://github.com/asonnino/coconut





THE ALAN

TURING INSTITUTE

This work is supported in part by EPSRC Grant EP/M013286/1, the EU H2020 DECODE project (grant agreement number 732546), and The Alan Turing Institute.



## The ugly

#### Â

#### How coconuts are made

#### **Issue credentials**

take an attribute: mcompute:  $c_m = g_1^m h_1^o$  and  $h = H(c_m)$ credential:  $\sigma_i = (h, h^{x_i+y_i\cdot m})$  and secret key  $(x_i, y_i)$ 

#### Aggregate credentials

Lagrange polynomial: 
$$l_i = \left(\prod_{i=1, j \neq i}^t (0-j)\right) \left(\prod_{i=1, j \neq i}^t (i-j)\right)^{-1} \mod p$$
  
compute:  $\prod_{i=1}^t (h^{x_i+y_i \cdot m})^{l_i} = \prod_{i=1}^t h^{(x_i l_i)} \prod_{i=1}^t h^{(y_i l_i) \cdot m} = h^{x+y \cdot m}$ 

#### How coconuts are made

#### Prove credentials

public key:  $(g_2, \alpha, \beta) = (g_2, g_2^{x_i}, g_2^{y_i})$ pick at random: r' and compute  $\sigma' = (h^{r'}, h^{(x_i+y_i\cdot m)r'})$ pick at random: r and compute  $\kappa = \alpha \beta^m g_2^r$  and  $\nu = (h^{r'})^r$ 

#### Verify credentials

parse: 
$$\sigma' = (h', s')$$
  
verify:  $e(h', \kappa) = e(s'\nu, g_2)$   
 $e(h^{r'}, g_2^{x+y\cdot m+r}) = e((h^{(x_i+y_i\cdot m)r'})(h^{r'})^r, g_2)$