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

Shehar Bano
University College London
Chainspace

s.bano@ucl.ac.uk
@thatBano

Co-authors

Alberto Sonnino

Mustafa Al-Bassam

George Danezis

CHAINSPACE

What is a selective disclosure credential scheme?

- A cryptographic scheme allowing to issue and verify credentials
  - Credentials can embed multiple attributes (eg. age, name, etc)
  - Selective disclosure means that you can choose which attributes to reveal (and which ones to keep secret) when you show your credentials

What is Coconut?

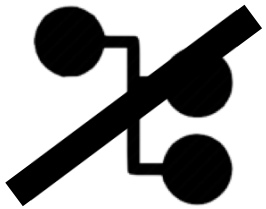
- A scheme for selective disclosure credentials that supports:

Blindness

Unlinkability

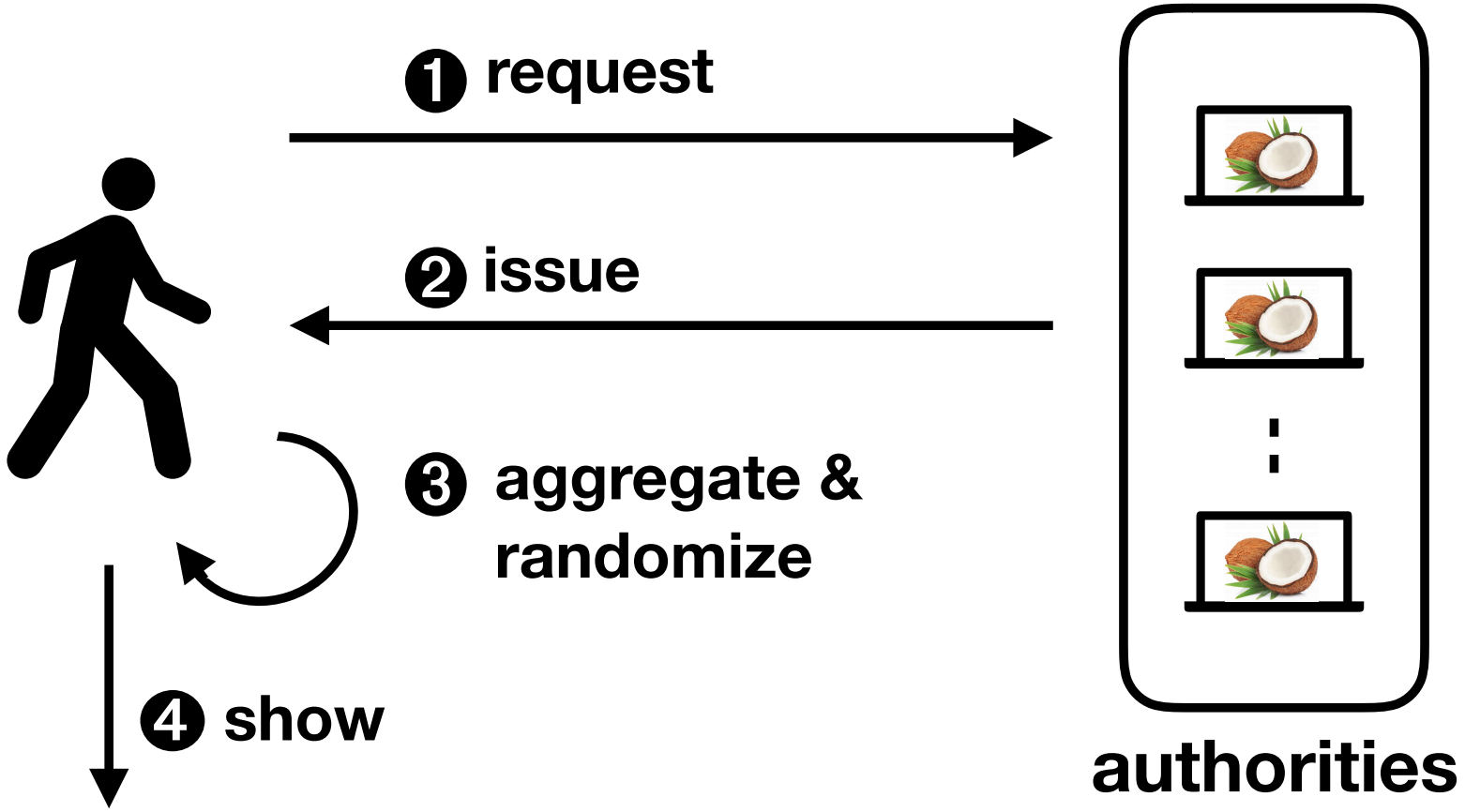
Threshold Authorities

Authorities non-interactivity



System Overview

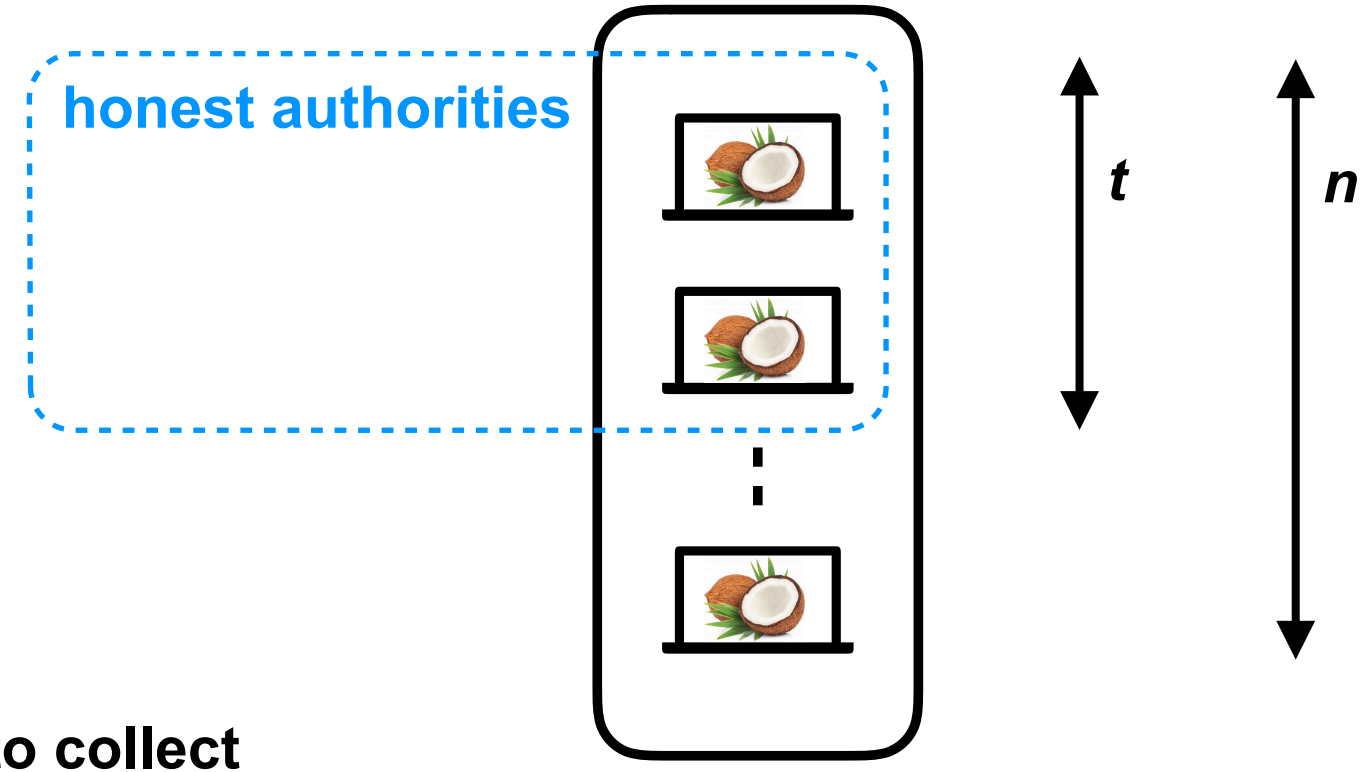
- How Coconut works?



What is Coconut?

- A selective credential scheme that supports:
  - **Threshold authorities:** Only a subset of the authorities is required to issue partial credentials in order to allow the users to generate a consolidated credential
  - Non-interactivity
  - Blindness
  - Unlinkability

Threshold Authorities



Users need to collect only *t* shares

What is Coconut?

- A selective credential scheme that supports:
  - Threshold authorities
  - **Non-interactivity:** The authorities may operate independently of each other, i.e., following a simple key distribution and setup phase, they do not need to synchronize or further coordinate their activities
  - Blindness
  - Unlinkability

What is Coconut?

- A selective credential scheme that supports:
  - Threshold authorities
  - Non-interactivity
  - **Blindness:** The authorities issue the credential without learning any additional information about the private attributes included in the credential
  - Unlinkability

What is Coconut?

- A selective credential scheme that supports:
  - Threshold authorities
  - Non-interactivity
  - Blindness
  - **Unlinkability:** It is impossible to link multiple showings of the credentials with each other, or the issuing transcript, even if all the authorities collude

Motivation

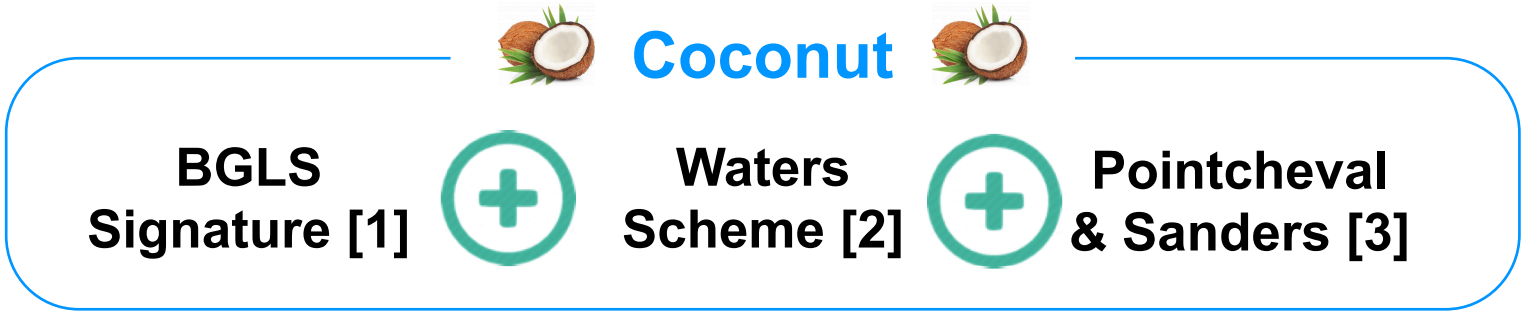
- Related works

Scheme	Blindness	Unlinkable	Aggregable	Threshold	Signature Size
[39] Waters Signature	✗	✗	○	✗	2 Elements
[26] LOSSW Signature	✗	✗	◒	✗	2 Elements
[8] BGLS Signature	✗	✗	●	✓	1 Element
[15] CL Signature	✓	✓	○	✗	$O(q)$ Elements
[31] Pointcheval *et al.*	✓	✓	◒	✗	2 Elements
 Coconut	✓	✓	●	✓	2 Elements

○ not aggregable
◒ sequentially aggregable
● user-side aggregable
q number of attributes

Coconut Credentials Scheme

- Where do coconuts come from?



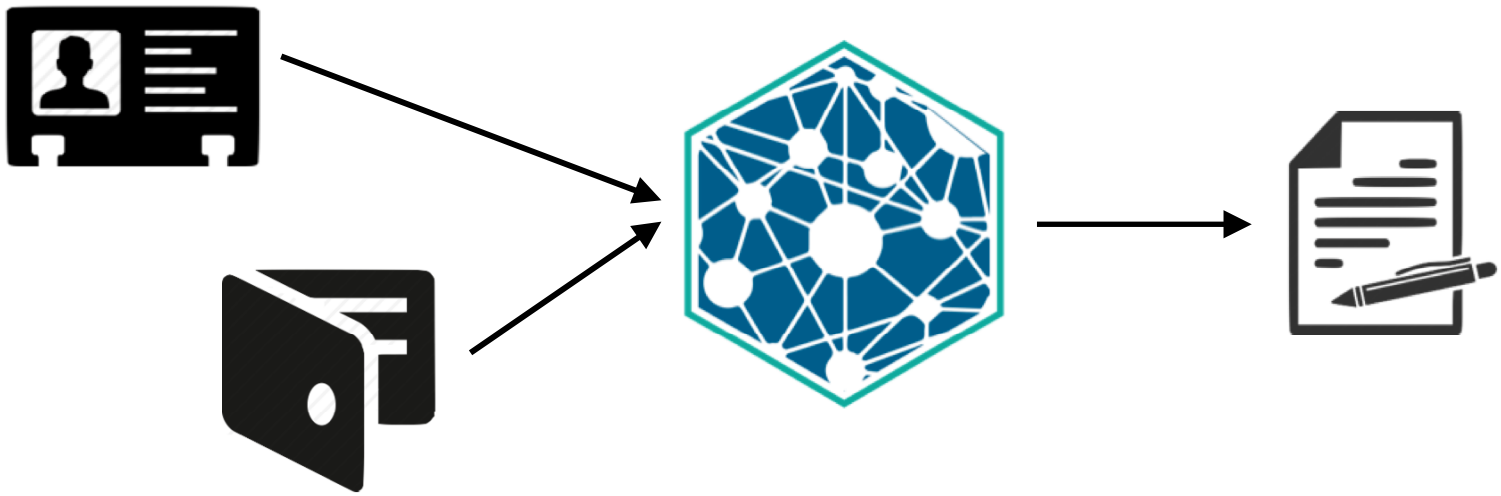
[1] Dan Boneh, Craig Gentry, Ben Lynn, and Hovav Shacham. 2003. Aggregate and Verifiably Encrypted Signatures from Bilinear Maps. In Eurocrypt, Vol. 2656. Springer, 416–432.

[2] Brent Waters. 2005. Efficient Identity-Based Encryption Without Random Oracles. *Eurocrypt*, Vol. 3494. Springer, 114–127.

[3] David Pointcheval and Olivier Sanders. 2016. Short Randomizable Signatures. In Cryptographers' Track at the RSA Conference. Springer, 111–126.

Coconut + Blockchains

- User verification through smart contracts (analogy with classic login username/password authentication)



Coconut + Blockchains: Motivation

- Every blockchain node that processes the transaction will see the password

- Problem is solved if we get a randomized credential from a single trusted third-party
  - but that re-centralises the system!

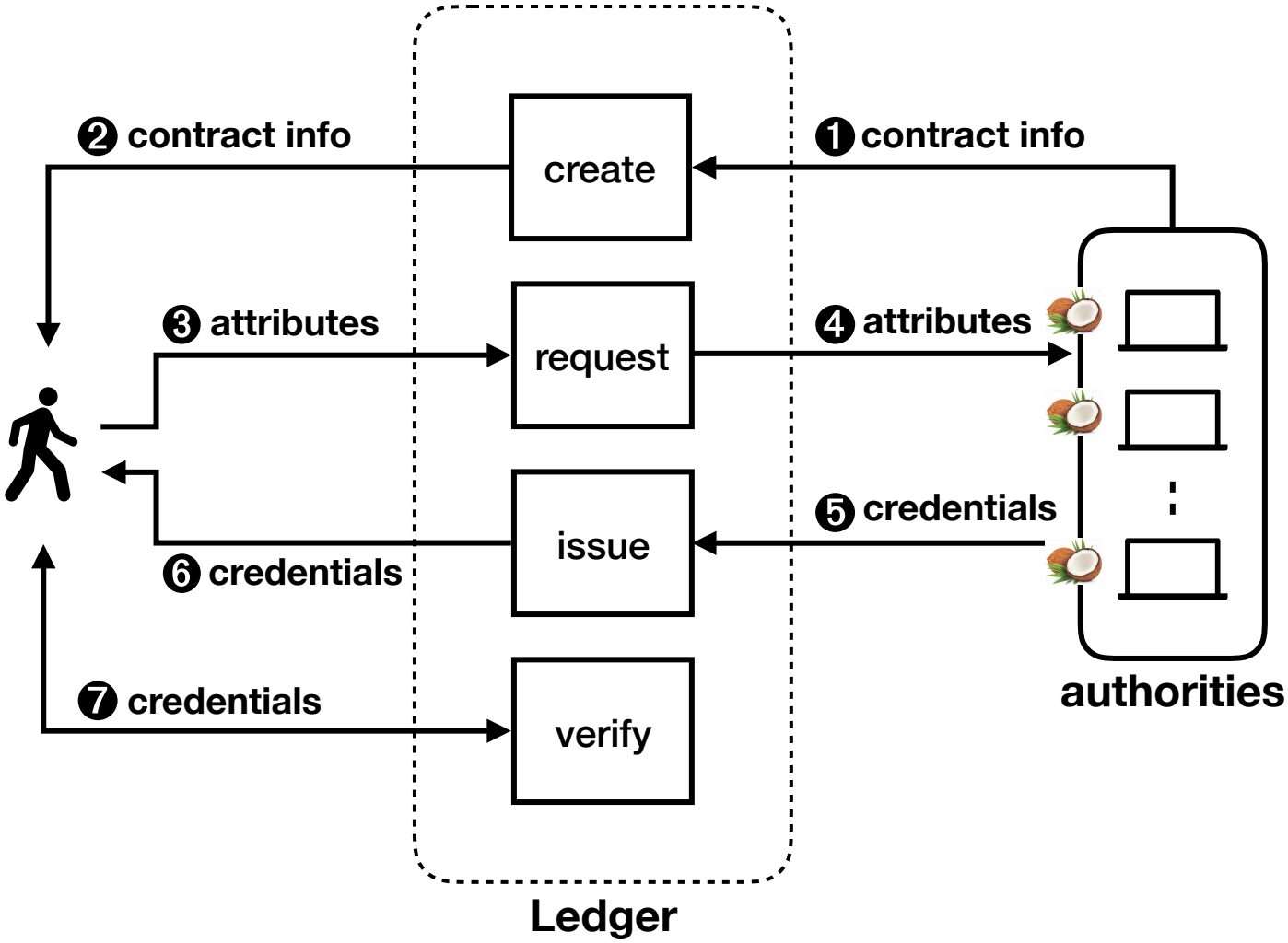
Coconut + Blockchains

- Scenario 1: Authorities outside the blockchain
  - (default model assumed through the rest of the slides)

- Scenario 2: Deeper integration -> Blockchain nodes also serve as authorities
  - (applies to permissioned blockchains only)

Smart Contract Library

- Coconut library for Chainspace (also for Ethereum)

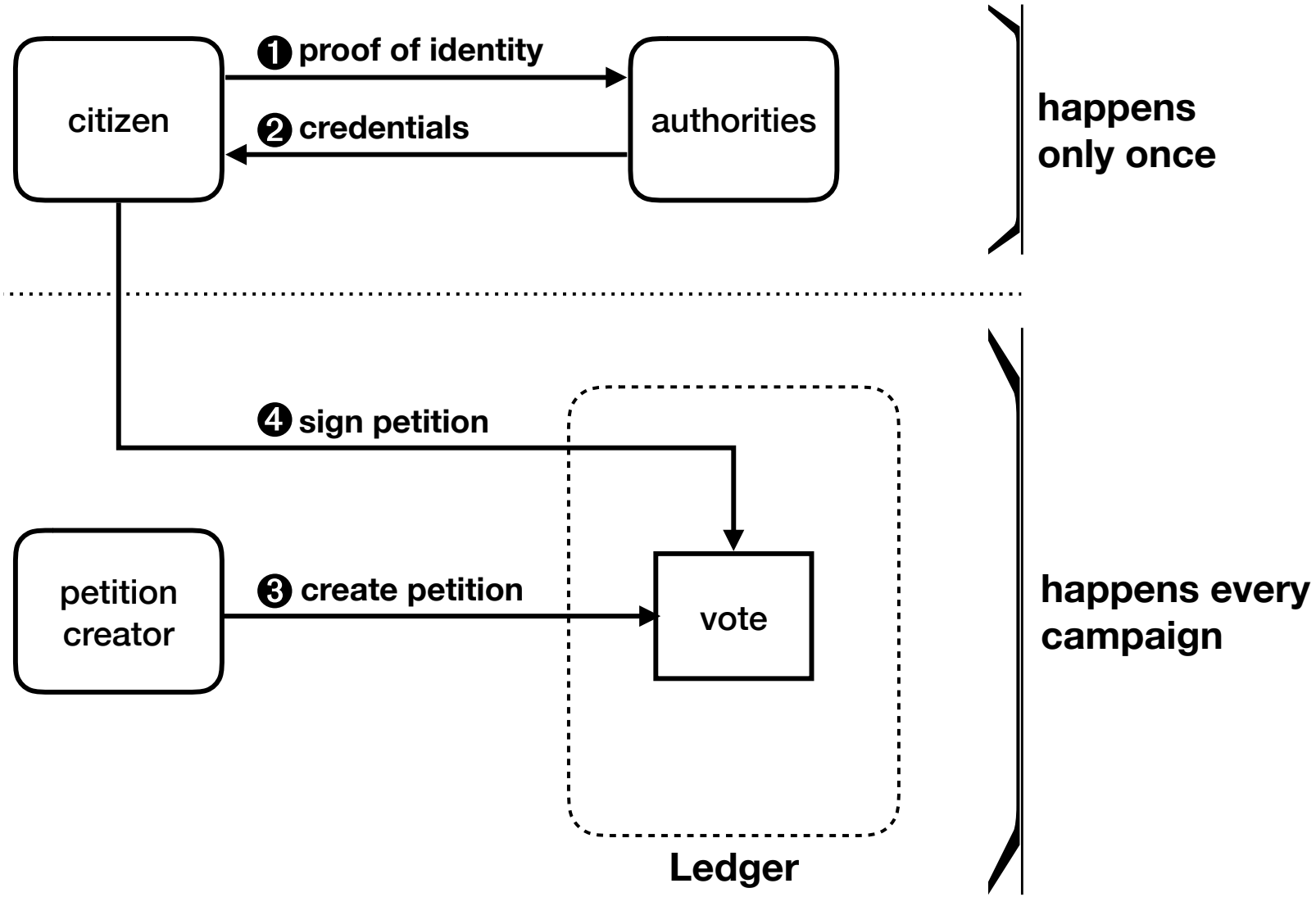


Application: Privacy-preserving Petitions

- Several authorities managing a city (e.g., Barcelona) wish to issue some long-term credentials to its citizens

- This enables any third-party to organise a privacy-preserving petition
  - i.e., users remain anonymous and unlinkable across petitions

Privacy-preserving Petitions



Privacy-preserving Petitions

- **Blindness** property prevents the authorities from learning the citizen's secret key, and misusing it to sign petitions on behalf of the citizen.

- Citizens re-use credentials multiple times while staying anonymous and unlinkable across petitions

- Allows for distributed credentials issuance, removing a central authority and preventing a single entity from creating arbitrary credentials to sign petitions multiple times.

Privacy-preserving Petitions

- Blindness property prevents the authorities from learning the citizen's secret key, and misusing it to sign petitions on behalf of the citizen.

- Citizens re-use credentials multiple times while staying **anonymous** and **unlinkable** across petitions

- Allows for distributed credentials issuance, removing a central authority and preventing a single entity from creating arbitrary credentials to sign petitions multiple times.

Privacy-preserving Petitions

- Blindness property prevents the authorities from learning the citizen's secret key, and misusing it to sign petitions on behalf of the citizen.

- Citizens re-use credentials multiple times while staying anonymous and unlinkable across petitions

- Allows for **distributed credentials issuance**, removing a central authority and preventing a single entity from creating arbitrary credentials to sign petitions multiple times.

Privacy-preserving Petitions

- Being tested in pilot iDigital / BCNow Platform in partnership with Barcelona City Council and the city's digital democracy platform Decidim.Barcelona

https://decodeproject.eu/pilots

EU DECODE project, a consortium of 14 partners from across Europe

Other Coconut Applications

- Coin Tumbler
- Censorship-resistant Distribution of Proxies

Other Coconut Applications

- Coin Tumbler
  - **User can buy goods/services from a merchant without being identified**
  - Coconut prevents a single authority from creating coins to steal all the money
  - Coconut prevents a single authority from blocking the issuance of a token

- Censorship-resistant Distribution of Proxies

Other Coconut Applications

- Coin Tumbler

- Censorship-resistant Distribution of Proxies
  - **Allows volunteers to unlinkably run verified proxies for censorship evasion**
  - Users know that it is a trust-worthy proxy due to Coconut verifiied credentials
  - Proxies cannot be linked to volunteers, enabling coercion-resistance

Performance

- How fast is Coconut?

Operation	μ [ms]	$\sqrt{\sigma^2}$ [ms]
Keygen	2.392	$\pm$ 0.006
Sign	0.445	$\pm$ 0.001
AggregateSign	0.004	$\pm$ 0.000
AggregateKeys	0.017	$\pm$ 0.000
Randomize	0.545	$\pm$ 0.002
Verify	6.714	$\pm$ 0.005
PrepareBlindSign	2.633	$\pm$ 0.003
BlindSign	3.356	$\pm$ 0.002
ShowBlindSign	1.388	$\pm$ 0.001
BlindVerify	10.497	$\pm$ 0.002
AggregateThSign	0.454	$\pm$ 0.000

sign

verify

signing is fast, verifying takes 10ms

Performance

- What is the credentials size?

2 Group Elements

No matter how many attributes…

No matter how many authorities…

Performance

- How does Coconut scale?

Number of authorities: n, Signature size: 132 bytes

Transaction	**complexity**	**size [B]**
Signature on public attribute:		
❶ request credential	$O(n)$	32
❷ issue credential	$O(n)$	132
❸ verify credential	$O(1)$	162
Signature on private attribute:		
❶ request credential	$O(n)$	516
sign ❷ issue credential	$O(n)$	132
verify ❸ verify credential	$O(1)$	355

Signing scales linearly, verifying is constant time

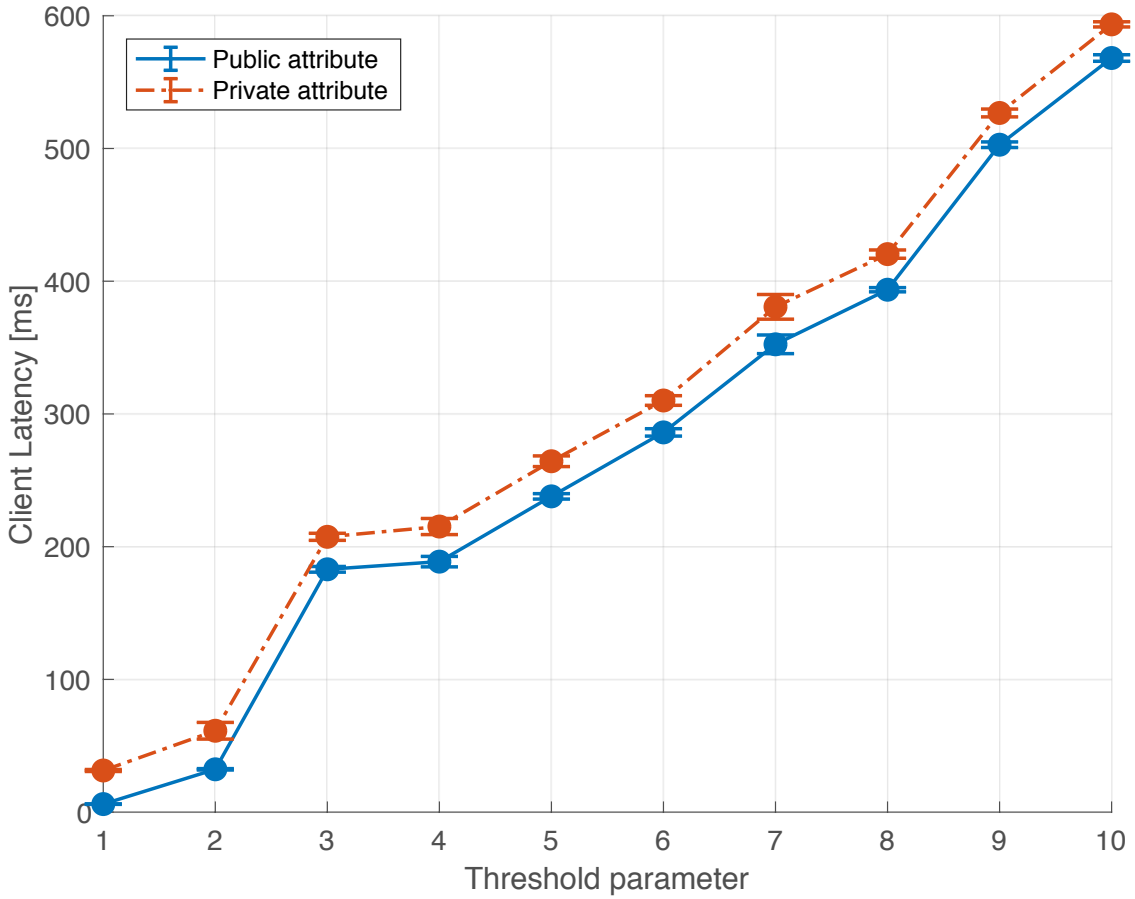
Performance

- What is the client-perceived latency?

pick 10 locations across the world

Performance

- What is the client-perceived latency?

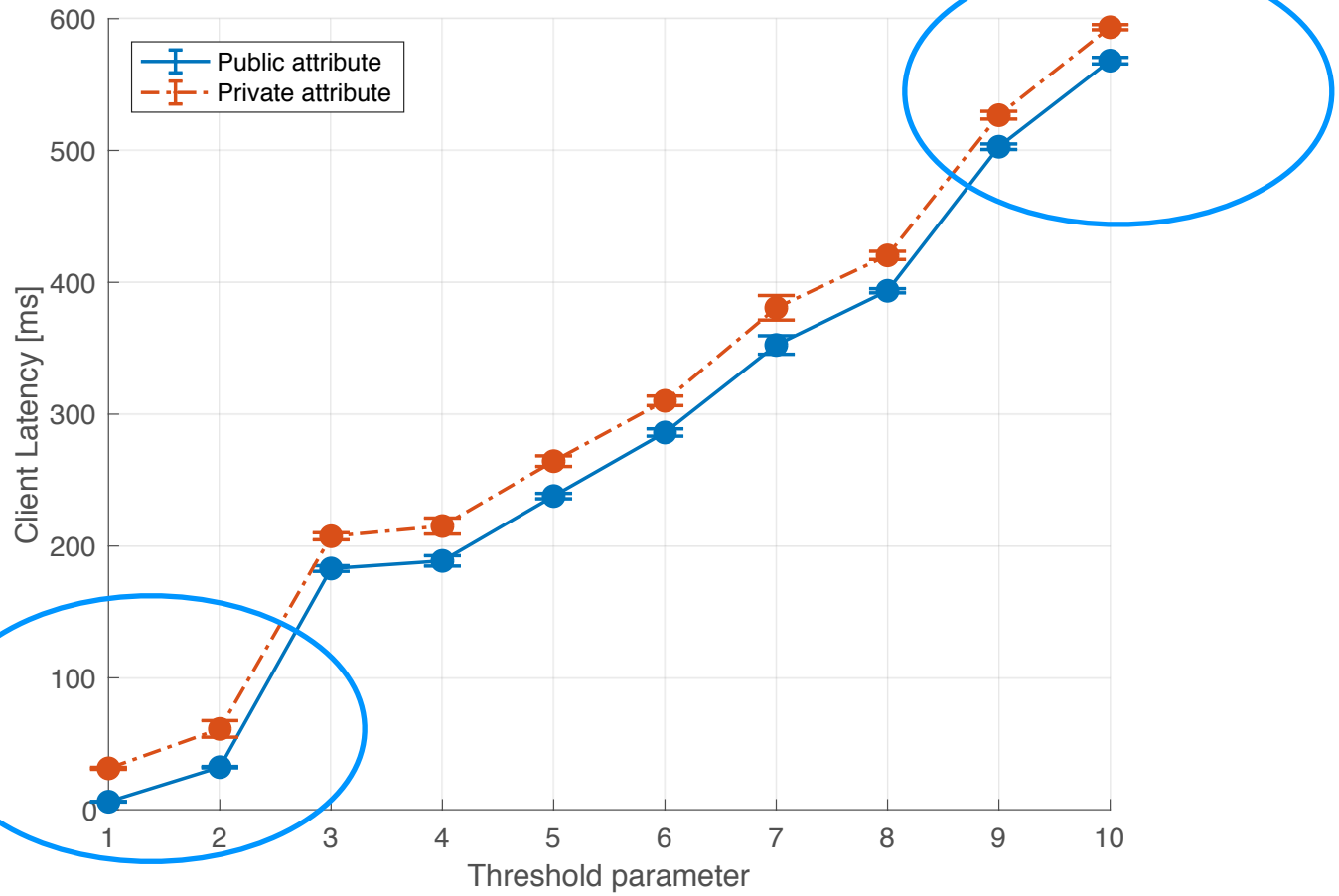


client latency VS **number of authorities**

Performance

- What is the client-perceived latency?

Tokyo & Sidney



Europe
(close to client)

latency mostly due to remote geo-locations

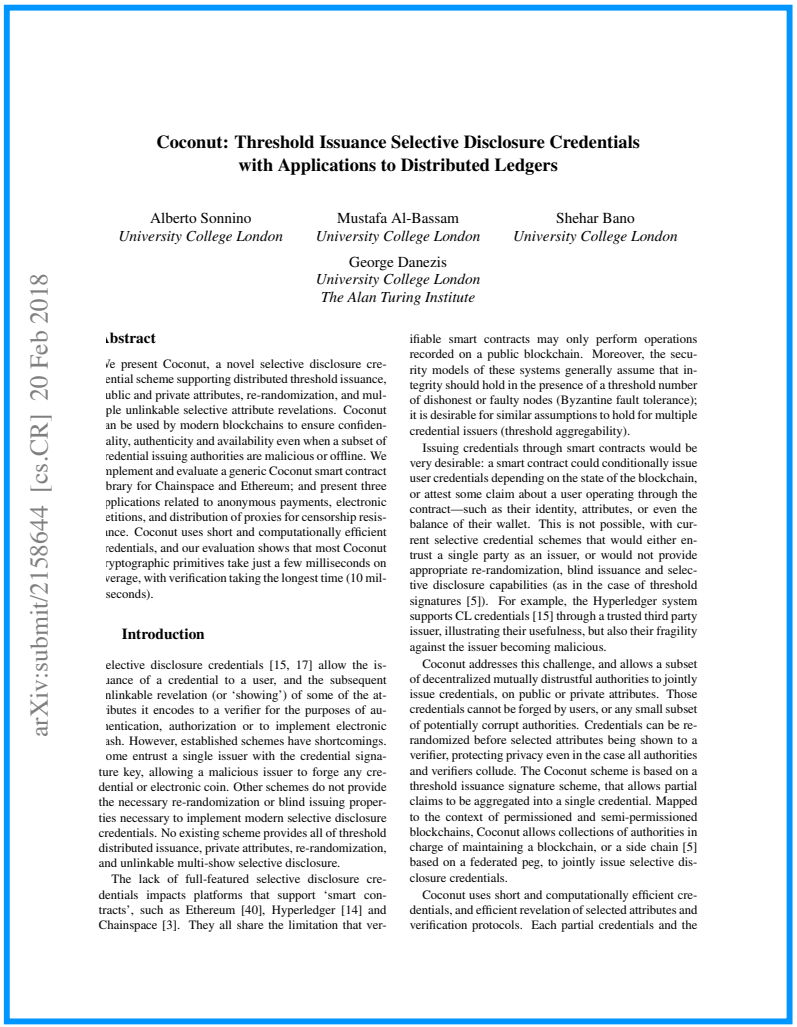
Paper(*) available online: arxiv.org/abs/1802.07344

Full cryptographic scheme

Smart contract library evaluation

Privacy-preserving petition, 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

bstract

/e present Coconut, a novel selective disclosure cre-ential scheme supporting distributed threshold issuance, ublic and private attributes, re-randomization, and mul-ple unlinkable selective attribute revelations. Coconut an be used by modern blockchains to ensure confiden-ality, authenticity and availability even when a subset of redential issuing authorities are malicious or offline. We nplement and evaluate a generic Coconut smart contract brary for Chainspace and Ethereum; and present three pplications related to anonymous payments, electronic etitions, and distribution of proxies for censorship resis-nce. Coconut uses short and computationally efficient redentials, and our evaluation shows that most Coconut ryptographic primitives take just a few milliseconds on verage, with verification taking the longest time (10 mil-seconds).

Introduction

elective disclosure credentials [15, 17] allow the is-ance of a credential to a user, and the subsequent nlinkable revelation (or 'showing') of some of the at-ibutes it encodes to a verifier for the purposes of au-entication, authorization or to implement electronic sh. However, established schemes have shortcomings. ome entrust a single issuer with the credential signa-ture key, allowing a malicious issuer to forge any cre-dential or electronic coin. Other schemes do not provide the necessary re-randomization or blind issuing proper-ties 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 cre-dentials impacts platforms that support 'smart con-tracts', 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 secu-rity models of these systems generally assume that in-tegrity 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 cur-rent selective credential schemes that would either en-trust a single party as an issuer, or would not provide appropriate re-randomization, blind issuance and selec-tive 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 re-randomized 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 dis-closure credentials.

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

(*) Also a blog post! https://www.benthamsgaze.org/2018/03/09/coconut-threshold-issuance-selective-disclosure-credentials-with-applications-to-distributed-ledgers/

Conclusion

- **Main take-aways**

Threshold issuance  **Suitable for distributed ledgers**

Randomizable  **Multi-use & privacy**

Thanks
Q/A

@thatBano
s.bano@ucl.ac.uk
https://sheharbano.com

https://github.com/asonnino/coconut

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.