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

Shehar Bano
University College London
Chainspace

s.bano@ucl.ac.uk
@thatBano
What is a selective disclosure credential scheme?

- A cryptographic scheme allowing to issue and verify credentials
  - Credentials can embed multiple attributes (e.g., 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?

1. request
2. issue
3. aggregate & randomize
4. show

authorities
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

honest authorities

authorities

$t$

$n$
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

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Blindness</th>
<th>Unlinkable</th>
<th>Aggregable</th>
<th>Threshold</th>
<th>Signature Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>[39] Waters Signature</td>
<td>✗</td>
<td>✗</td>
<td></td>
<td>✗</td>
<td>2 Elements</td>
</tr>
<tr>
<td>[26] LOSSW Signature</td>
<td>✗</td>
<td>✗</td>
<td></td>
<td>✗</td>
<td>2 Elements</td>
</tr>
<tr>
<td>[8] BGLS Signature</td>
<td>✗</td>
<td>✗</td>
<td></td>
<td>✗</td>
<td>1 Element</td>
</tr>
<tr>
<td>[15] CL Signature</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>O(q) Elements</td>
</tr>
<tr>
<td>[31] Pointcheval et al.</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>2 Elements</td>
</tr>
<tr>
<td>Coconut</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>2 Elements</td>
</tr>
</tbody>
</table>

○ not aggregable
● sequentially aggregable
● user-side aggregable

$q$ number of attributes
Coconut Credentials Scheme

Where do coconuts come from?


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!
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

1. proof of identity
2. credentials

happens only once

1. sign petition
2. create petition

happens every campaign
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
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 verified credentials
  - Proxies cannot be linked to volunteers, enabling coercion-resistance
Performance

- How fast is Coconut?

<table>
<thead>
<tr>
<th>Operation</th>
<th>$\mu$ [ms]</th>
<th>$\sqrt{\sigma^2}$ [ms]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Keygen</td>
<td>2.392</td>
<td>± 0.006</td>
</tr>
<tr>
<td>Sign</td>
<td>0.445</td>
<td>± 0.001</td>
</tr>
<tr>
<td>AggregateSign</td>
<td>0.004</td>
<td>± 0.000</td>
</tr>
<tr>
<td>AggregateKeys</td>
<td>0.017</td>
<td>± 0.000</td>
</tr>
<tr>
<td>Randomize</td>
<td>0.545</td>
<td>± 0.002</td>
</tr>
<tr>
<td>Verify</td>
<td>6.714</td>
<td>± 0.005</td>
</tr>
<tr>
<td>PrepareBlindSign</td>
<td>2.633</td>
<td>± 0.003</td>
</tr>
<tr>
<td>BlindSign</td>
<td>3.356</td>
<td>± 0.002</td>
</tr>
<tr>
<td>ShowBlindSign</td>
<td>1.388</td>
<td>± 0.001</td>
</tr>
<tr>
<td>BlindVerify</td>
<td>10.497</td>
<td>± 0.002</td>
</tr>
<tr>
<td>AggregateThSign</td>
<td>0.454</td>
<td>± 0.000</td>
</tr>
</tbody>
</table>

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?

<table>
<thead>
<tr>
<th>Transaction</th>
<th>complexity</th>
<th>size [B]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Signature on public attribute:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 request credential</td>
<td>$O(n)$</td>
<td>32</td>
</tr>
<tr>
<td>2 issue credential</td>
<td>$O(n)$</td>
<td>132</td>
</tr>
<tr>
<td>3 verify credential</td>
<td>$O(1)$</td>
<td>162</td>
</tr>
</tbody>
</table>

| Signature on private attribute: |            |          |
| 1 request credential           | $O(n)$     | 516      |
| 2 issue credential             | $O(n)$     | 132      |
| 3 verify credential            | $O(1)$     | 355      |

Signatures scale 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?

![Graph showing client latency versus threshold parameter]

client latency VS number of authorities
Performance

What is the client-perceived latency?

- Latency mostly due to remote geo-locations.

Latency mostly due to remote geo-locations.
Coconut: Threshold Issuance Selective Disclosure Credentials
with Applications to Distributed Ledgers

Alberto Sommio
University College London
Mustafa Al-Bassam
University College London
George Danezis
University College London
Shehar Bano
University College London
The Alan Turing Institute

Abstract
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 [46], Hyperledger [14] and Chainspace [5]. They all share the limitation that verifiable 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 [3]). 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. These credentials cannot be forged by some 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. Coconut addresses this challenge, and allows a subset of decentralized mutually distrustful authorities to jointly issue credentials, on public or private attributes. The Coconut scheme is based on a threshold issuance signature scheme, that allows partial claims to be aggregated into a single credential. Coconut uses short and computationally efficient credentials, and efficient revelation of selected attributes and verification protocols. Each partial credential and the
Conclusion

Main take-aways

- Threshold issuance ➔ Suitable for distributed ledgers
- Randomizable ➔ Multi-use & privacy
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.