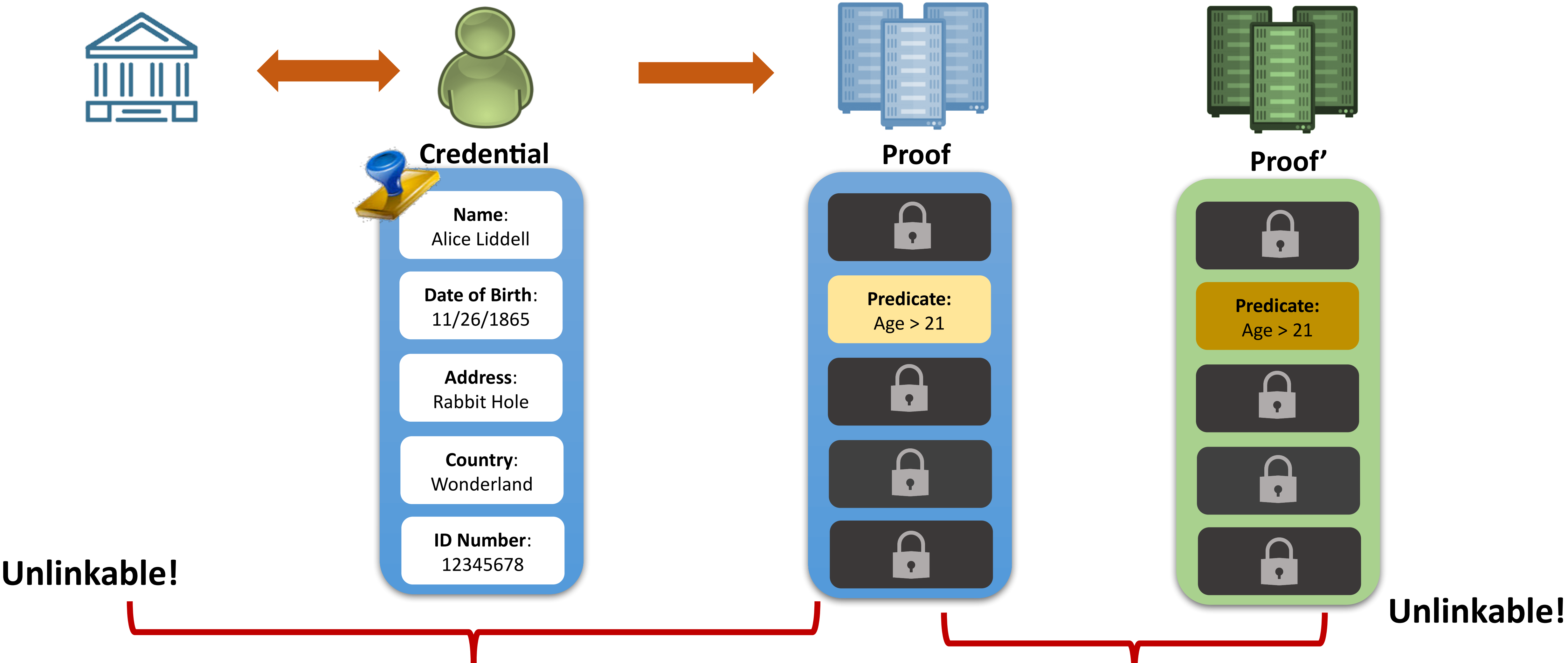
Challenges and new Features for Anonymous Credentials: Revocation and Decentralization

Foteini Baldimtsi

GEORGE MASON UNIVERSITY

Anonymous Authentication



Credential

Name: Alice Liddell

Date of Birth: 11/26/1865

Address: Rabbit Hole

Country: Wonderland

ID Number: 12345678

Proof

Predicate: Age > 21

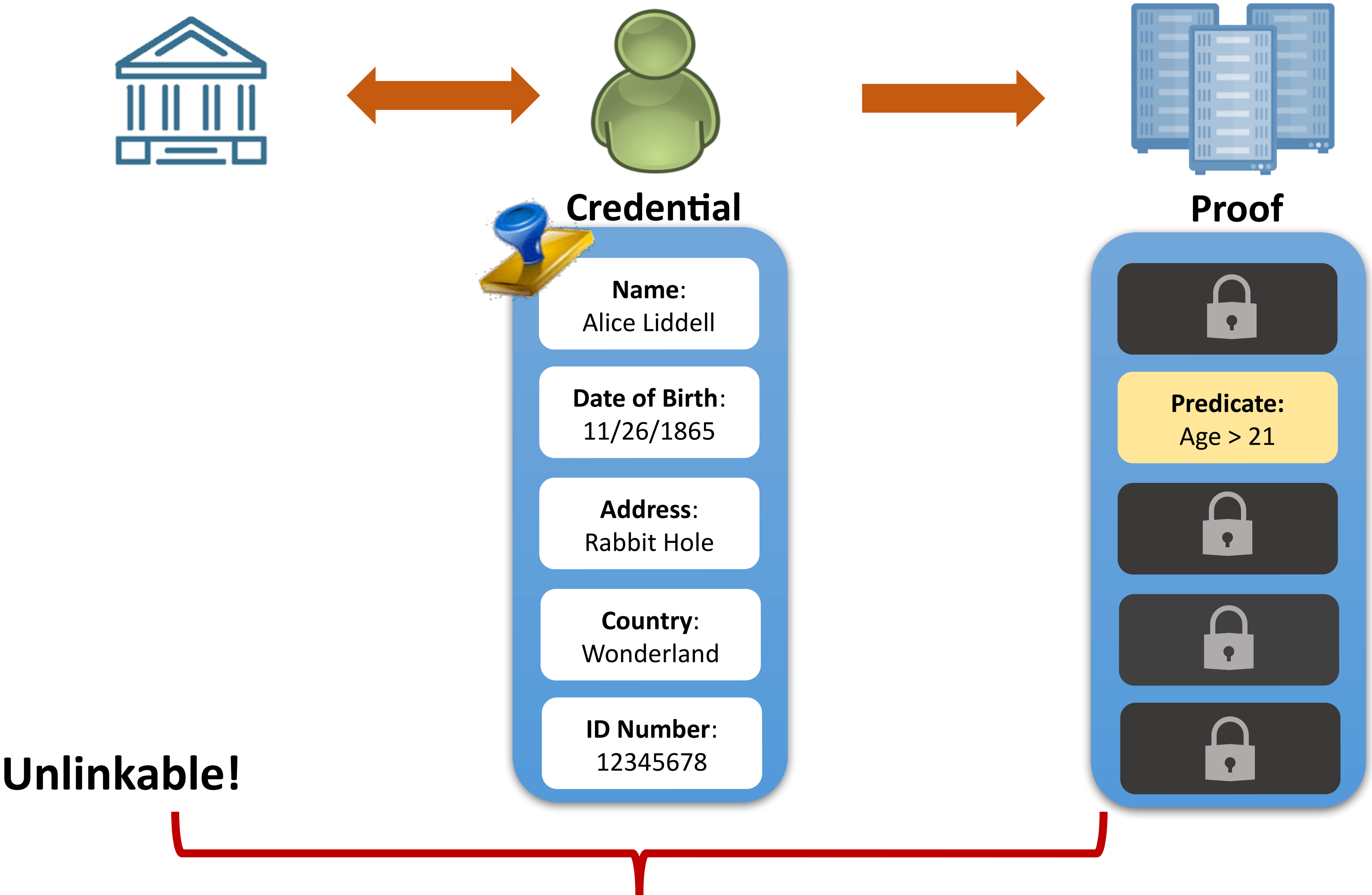
Proof'

Predicate: Age > 21

Unlinkable!

Unlinkable!

Anonymous Authentication



Credential

Name: Alice Liddell

Date of Birth: 11/26/1865

Address: Rabbit Hole

Country: Wonderland

ID Number: 12345678

Proof

Predicate: Age > 21

Unlinkable!

Generic construction

- Issuance: digital signatures
- Credential showing: ZK proofs

Efficiency?*

"Anonymous Credentials”

* zk-creds: a recent work suggesting anonymous credentials from SNARKs [RWGM’22]

Anonymous Credentials

Multi Use
Anonymous Credentials
based on signatures
and ZK proofs

Single Use
Anonymous Credentials
based on blind signatures
(+ optional ZK proofs)

Chaum'81, Chaum'82, Brands'93, CFY'98, CL'01, CL'02, CL'04, CHL'05, CG'08, BCKL'09, FDV'09, IB'12, CNR’12, …
BL’13b, RHBP’13, BCHKLN’13, RBHP’15, BCDLRSY’17 …]

U-Prove ABC4TRUST HYPERLEDGER INDY Signal

Anonymous Credentials

Multi Use
Anonymous Credentials
based on signatures
and ZK proofs

Single Use
Anonymous Credentials
based on blind signatures
(+ optional ZK proofs)

- Subscription Services
- Digital IDs

?

- Single-use coupons
- E-cash
- E-voting

☑ More efficient*

*If only to be used a small number of times single use credentials are preferable

Multi-use Anonymous Credentials

Introduced by [CL01,CL04,…]

ABC4TRUST

HYPERLEDGER INDY

Core Primitive: CL signatures (Signatures on committed values that allow for efficient proofs of signature ownership)

CL signatures are based on RSA

[BBS’04, ASM’06]: BBS+ signatures based on bilinear pairings

Standardization?

Anonymous Credentials

Multi Use
Anonymous Credentials
based on signatures
and ZK proofs

Single Use
Anonymous Credentials
based on blind signatures
(+ optional ZK proofs)

- Subscription Services
- Digital IDs

?

- Single-use coupons
- E-cash
- E-voting

☑ More efficient*

*If only to be used a small number of times single use credentials are preferable

Single-Use Credential + Attributes

[BL’13b] “Anonymous Credentials Light”

Efficiency: EC based, 3-rounds

Issuance: 13 exponentiations for User + 7 exponentiations of Signer

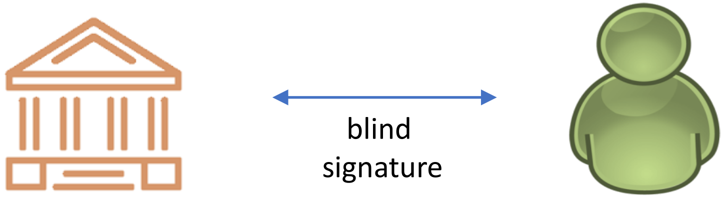
Verification: 8 exponentiations

Security: Unlikability/Blindness under DDH in RO

Unforgeability under DL in RO in the sequential setting

[BLLOR’21]: most efficient DL blind signatures (including [BL’13b]) are not secure under parallel issuance

Single-Use Credential + ~~Attributes~~

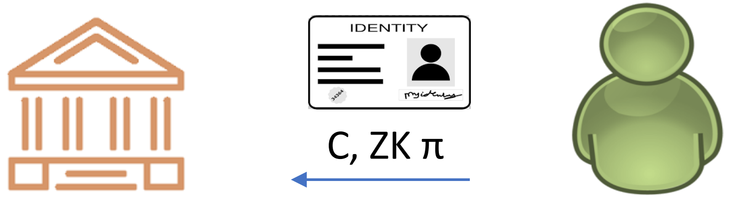


A valid blind signature is a credential/token!

Avoid double-use?

Single-Use Credential + Attributes

Registration Phase

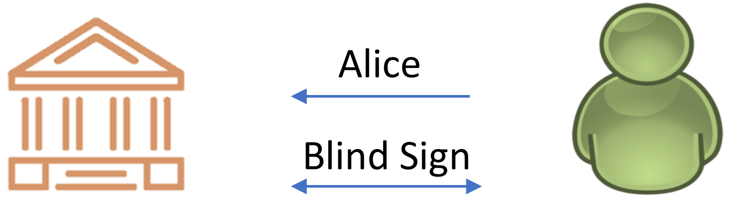


C, ZK π

Open account for (Alice, C)

Alice
C= commit (user attributes)

Issuance Phase



Alice

Blind Sign

Accounts: (Alice, C)

Alice, C

A valid blind signature is a credential/token!

If you double show your credential, (parts of) C will be disclosed!

Challenge: embed C in the credential while maintain unlinkability

no concurrently secure efficient schemes ⚠

(Anonymous) Digital Credentials in the Age of Decentralization

Decentralized Credentials

Decentralizing Issuance Of Anonymous Credentials

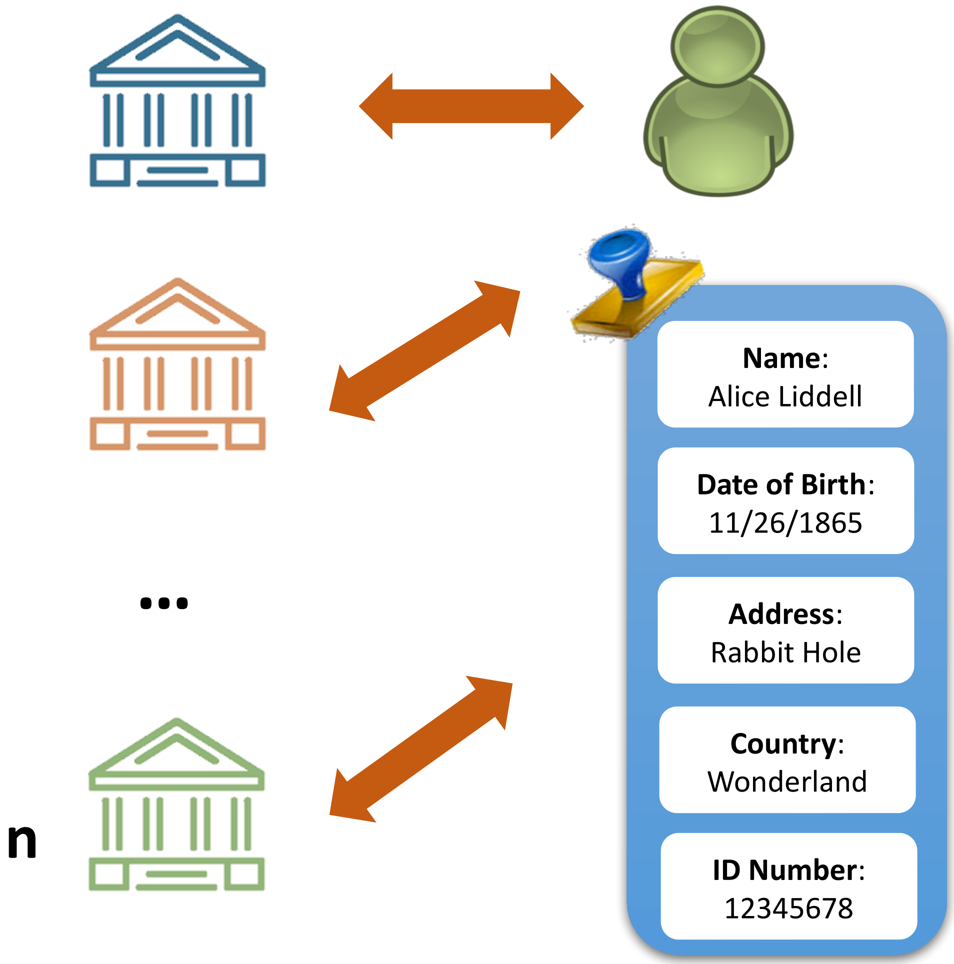
[Coconut’17]

NYM

DID: Decentralized Identities (anonymity)

[GGM’13,CanID’21]

Decentralizing Issuance- Threshold Blind Signs



Name: Alice Liddell

Date of Birth: 11/26/1865

Address: Rabbit Hole

Country: Wonderland

ID Number: 12345678

Use threshold (blind) signatures for the credential issuance [Coconut’17, NYM]

Smart contract

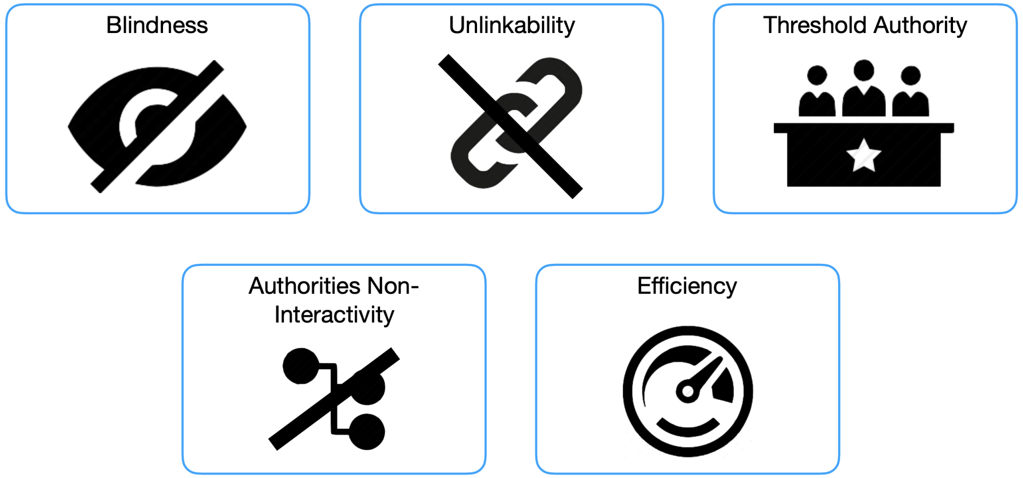
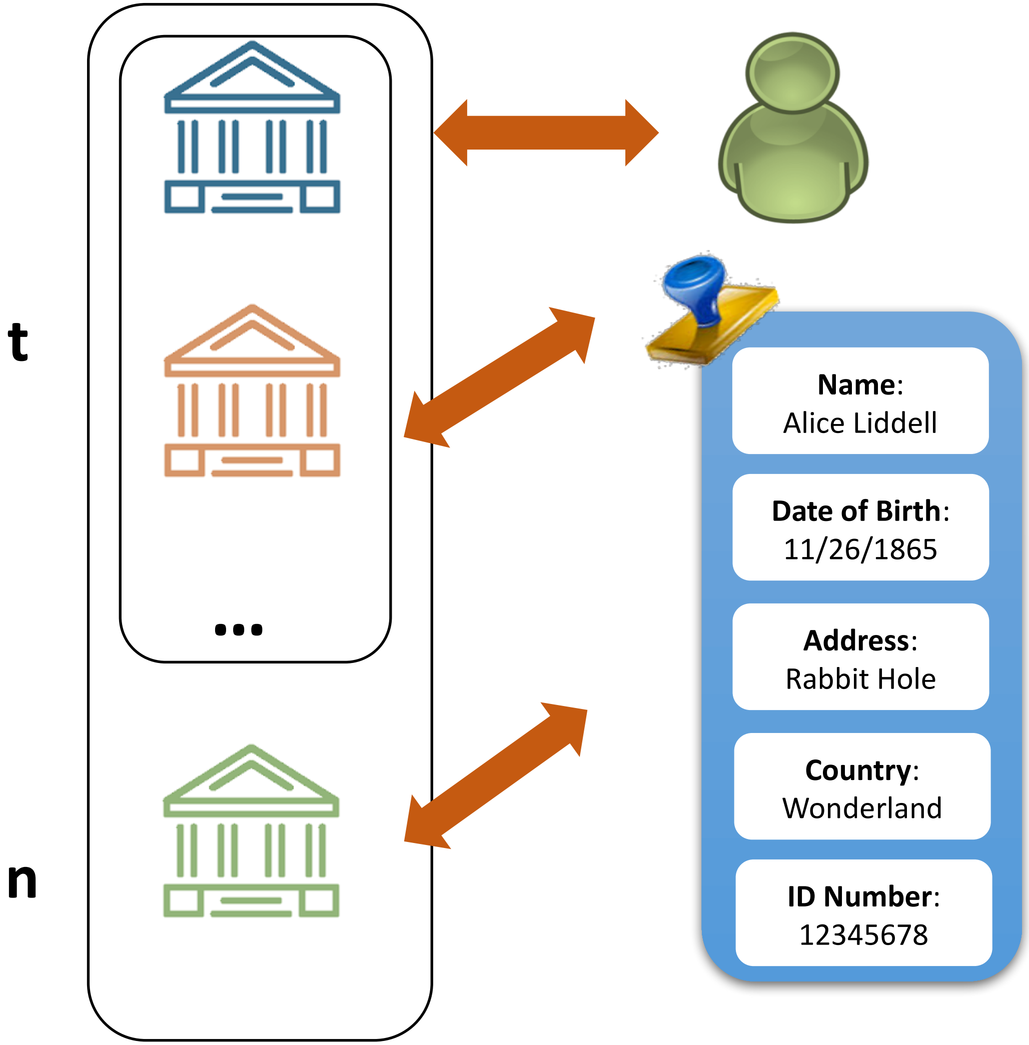


Image: from [Coconut] authors slides

Decentralizing Issuance- Threshold Blind Signs



Name: Alice Liddell

Date of Birth: 11/26/1865

Address: Rabbit Hole

Country: Wonderland

ID Number: 12345678

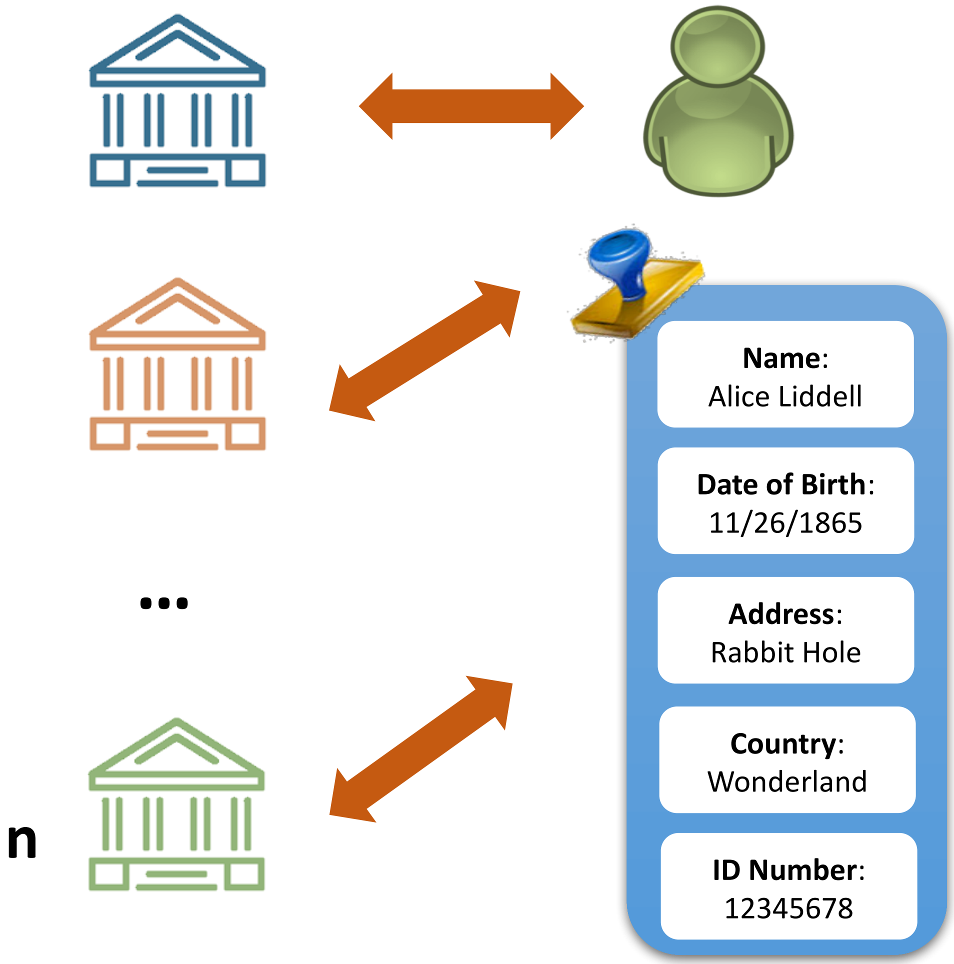
Use threshold (blind) signatures for the credential issuance [Coconut,NYM]

- User Requests for a credential
- Authorities Issue (at least t honest authorities)
- User collects, aggregates and randomizes
- User presents credential

Building blocks: PS'16 randomizable signatures, El Gamal Encryption, DL based ZK proofs

Multi Use
Anonymous Credentials based on signatures and ZK proofs

Decentralizing Issuance- Threshold Blind Signs



Name: Alice Liddell

Date of Birth: 11/26/1865

Address: Rabbit Hole

Country: Wonderland

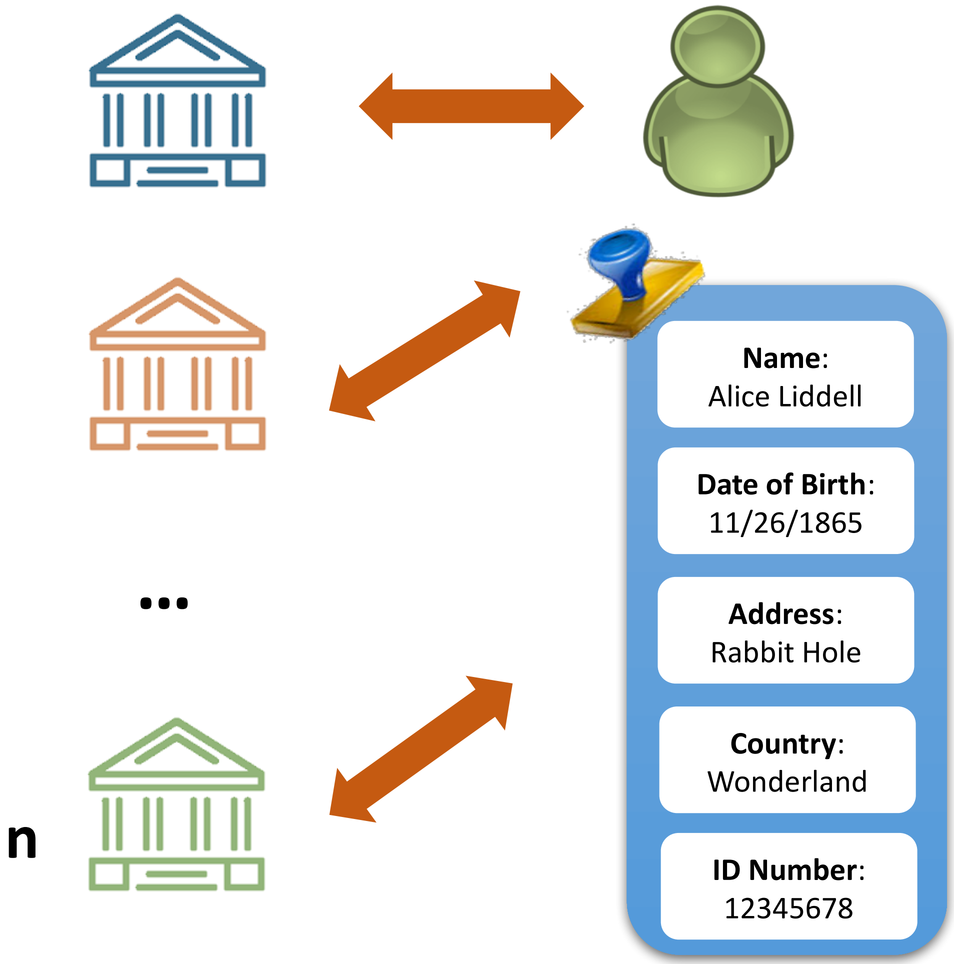
ID Number: 12345678

Use threshold (blind) signatures for the credential issuance [Coconut,NYM]

- Signer privacy, blindness/unlinkability ✓
- Interactive, non-flexible threshold key generation phase
- Smart contract relatively expensive (on-chain verification 4.2M ETH gas - $90 USD Aug’22) ⚠
- No support for revocation + auditability

[Zebra’22]: reduces gas costs x10 (uses SNARKs), adds revocation + auditability features

Decentralizing Issuance – Multi-Blind Signs [BKKL’23]



Name:
Alice Liddell

Date of Birth:
11/26/1865

Address:
Rabbit Hole

Country:
Wonderland

ID Number:
12345678

Use multi (blind) signatures [new approach]

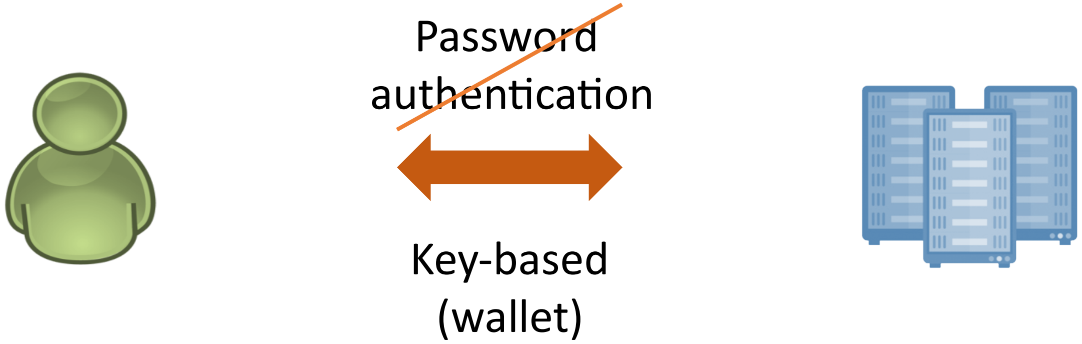
- Signer accountability
- More flexible signing process
- Controlled set anonymity

Constructions for both multi-use and single-user credentials (so much cheaper to verify! no zk proofs!)

Based on BLS signatures

Decentralized Identities – Self created

[GGM]: decentralized non general credentials, “self assertation of credential”



~~Password authentication~~

Key-based (wallet)

- Addresses bootstrapping (using oracles like DECO)
- Accountability (sanction lists)
- Weak privacy (if collusions happen)
- Not efficient revocation (not privately)

CANDID

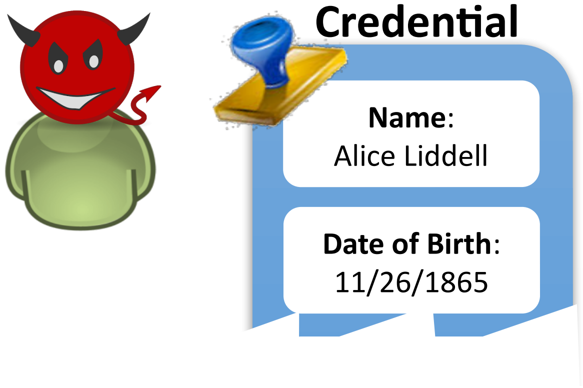
Anonymous Credentials

Multi Use
Anonymous Credentials
based on signatures
and ZK proofs

Single Use
Anonymous Credentials
based on blind signatures
(+ optional ZK proofs)

What about revocation? ?

Credential Revocation

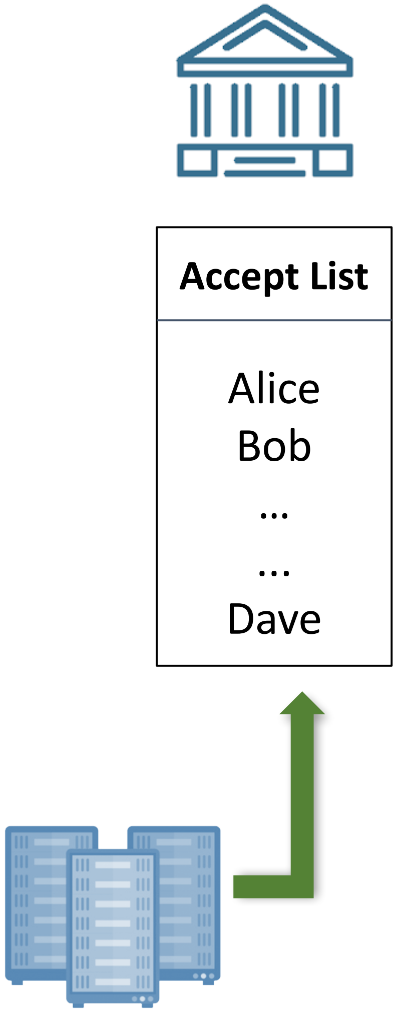


Revocation of credentials is a hard problem.

- In TLS millions of certificates are revoked per year
- Simultaneous/fast (i.e. due to critical security issues like Heartbleed) revocation is very hard
- More challenges in restricted settings, i.e. web browsers, IoT devices etc

Revocation of *anonymous* credentials is an even harder problem.

Credential Revocation



Accept List

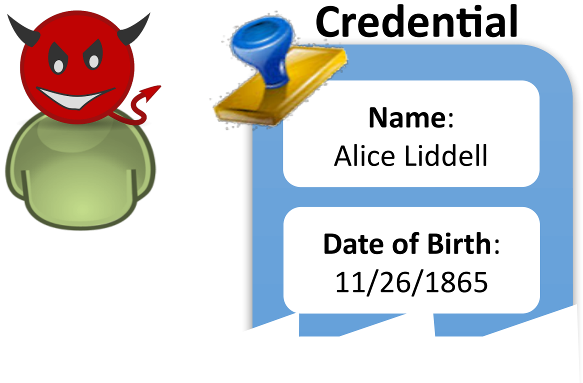
Alice
Bob
…
…
Dave

- ✓ additions/deletions

  (under central management)

- ✗ Size of Accept List O(n)

- ✗ User privacy



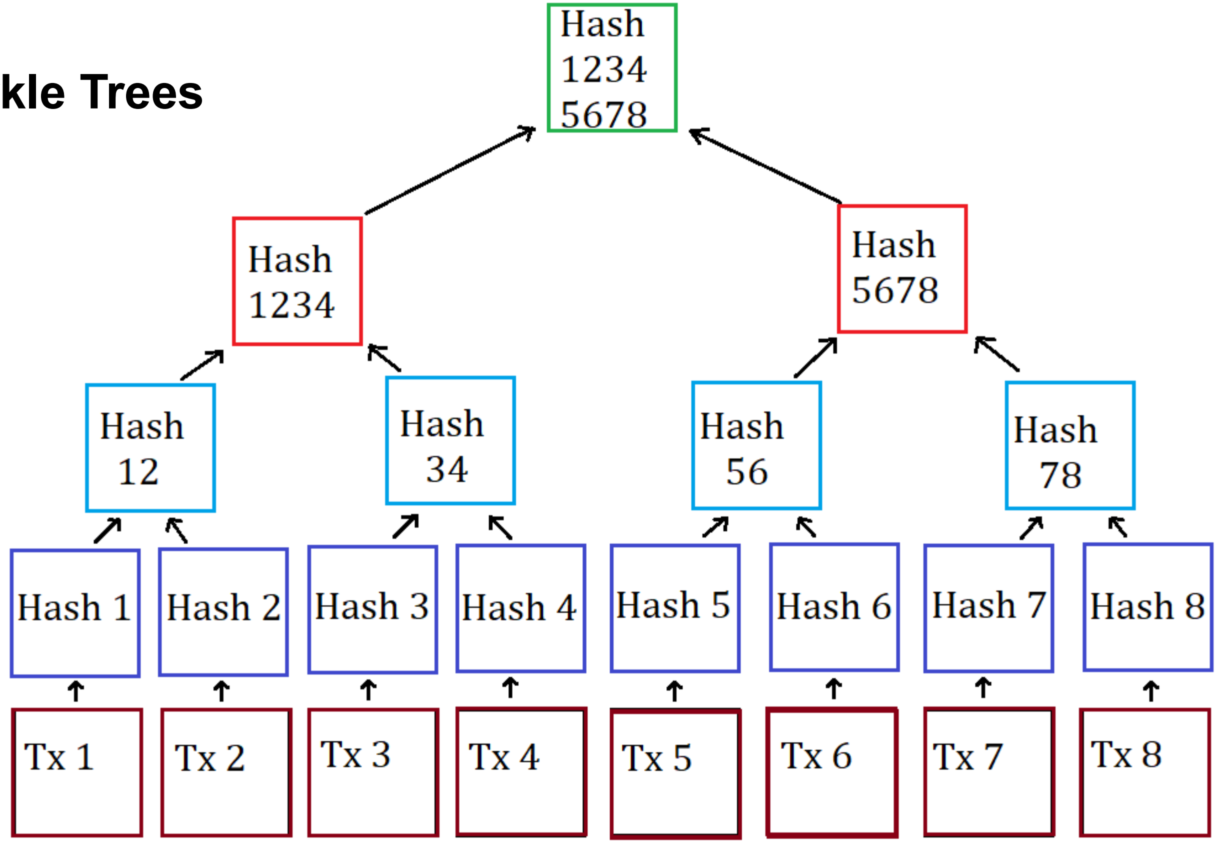
Credential

Name: Alice Liddell

Date of Birth: 11/26/1865

Credential Revocation

Merkle Trees



Digest: O(1)
Proof-of-membership: O(log n)

Cryptographic Accumulators

Digest: O(1)
Proof-of-membership: O(1)
++ more properties
(efficient non-membership proofs/updates/deletions)

Credential Revocation- Accumulators

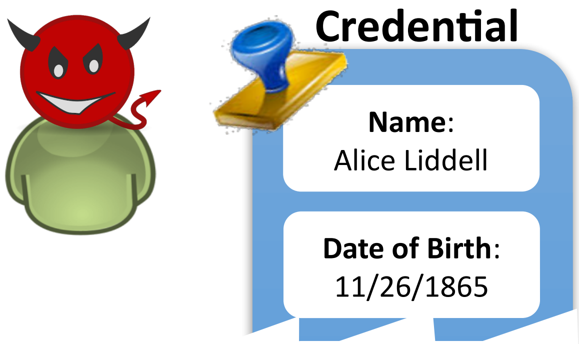
Accept List

$A_t = g^u$

- ✓ additions/deletions

  (under central management)

- ✓ Constant size
- ✗ User privacy



<u>**RSA Accumulator**</u>

Setup:

- Choose N=pq where p, q are secret primes
- *(initial state)*

Add()

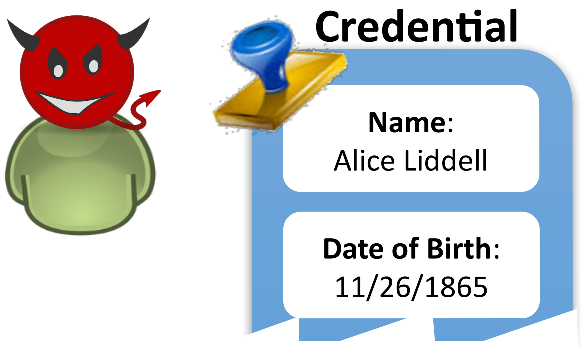
-

Del()

State after set S added:

$u=$

Credential Revocation- Accumulators



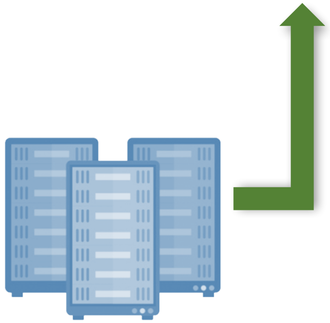
Credential

Name:
Alice Liddell

Date of Birth:
11/26/1865

Accept List

$A_t = g^u$



✓ additions/deletions

(under central management)

✓ Constant size

✗ User privacy

RSA Accumulator – How to prove membership

Add x

- $A_{t+1} = A_t^x$
- Witness $w = A_t$

Prove membership

- “x is in A_{t+1}, here is a witness w”
- Verify check: $A_{t+1} = w^x$

State after set S added:

$u=$

But what if more updates happen in between Add - Prove?

✗ Users need to update their witness for every single addition/deletion!

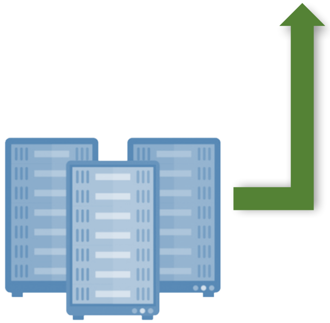
But do they really?

Credential Revocation- Modular Accumulators

[BCDLRSY’17,BK’21]

Accept List

$A_t=g^u$

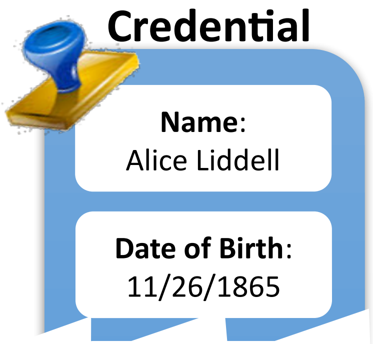


✓ additions/deletions

(under central management)

✓ Constant size

✗ User privacy



Modular Accumulators

Basic Idea

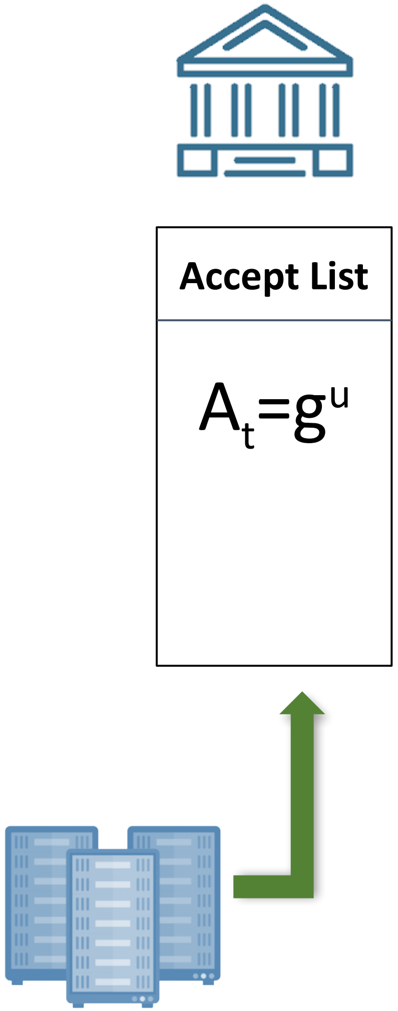
- Start with a full accumulator A_0 (includes all domain S)
- Manager (holding trapdoor) can issue membership witnesses
- No need to update witnesses after additions – A does not change
- Only changes after deletions!

O(d) updates

✗ **Reduced security (non-adaptive adversaries)**

→ Accumulator **A** for additions and **A’** for Deletions. Users need to show membership in 1st and non-membership in 2nd

Credential Revocation- Modular Accumulators

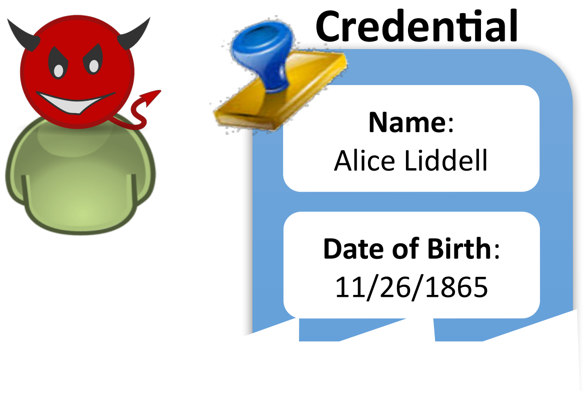


Accept List

$A_t = g^u$

- ✓ additions/deletions

  (under central management)
- ✓ Constant size
- ✗ User privacy



Credential

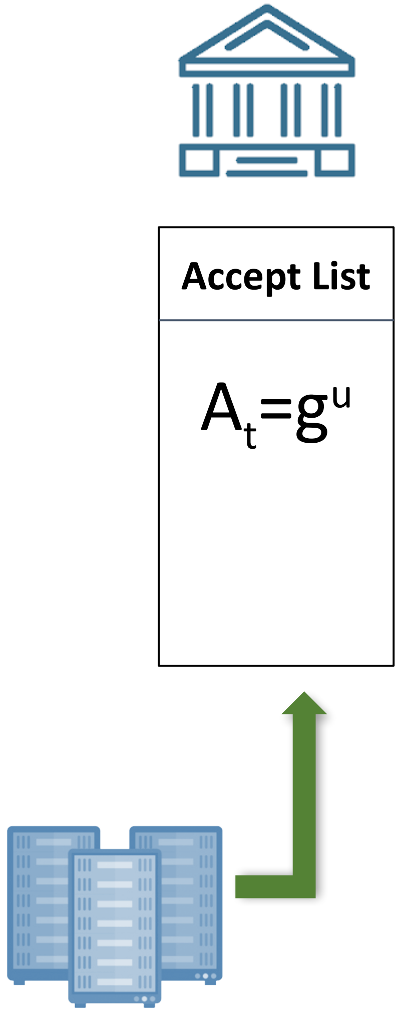
Name: Alice Liddell

Date of Birth: 11/26/1865

	Add	Del	Updates	soundness
RSA	✓	✓	O(a + d)	adaptive
Basic Idea	✓	✓	O(d)	non-adaptive
Modular	✓	✓	O(d)	adaptive

Join-Revoke unlinkability!

Credential Revocation- Modular Accumulators



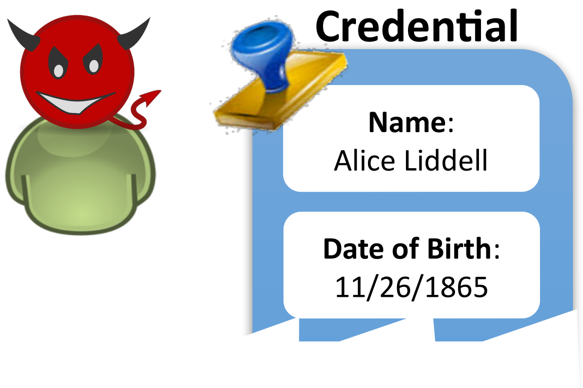
Accept List

$A_t=g^u$

- ✓ additions/deletions

  (under central management)

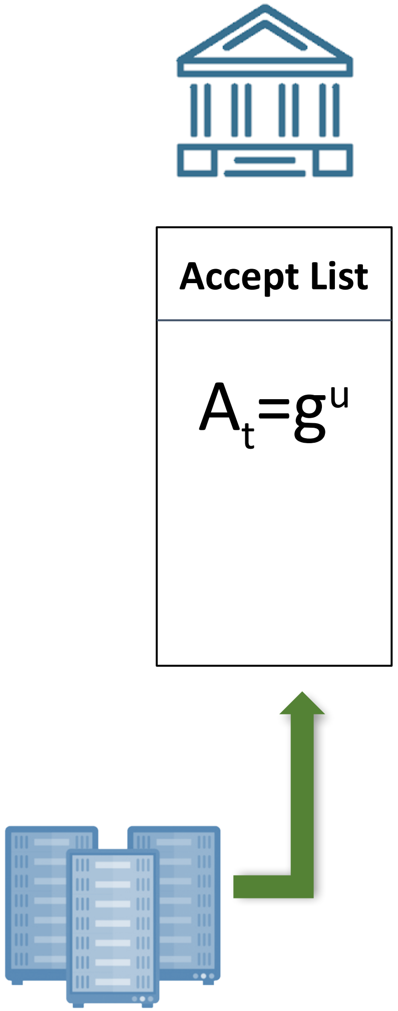
- ✓ Constant size
- ✗ User privacy



	Add	Del	Updates	soundness	Join-Revoke Unlinkability
RSA	✓	✓	O(a + d)	adaptive	✗
Basic Idea	✓	✓	O(d)	non-adaptive	✗
Modular	✓	✓	O(d)	adaptive	✓

Join-Revoke unlinkability!

Credential Revocation- Modular Accumulators



Accept List

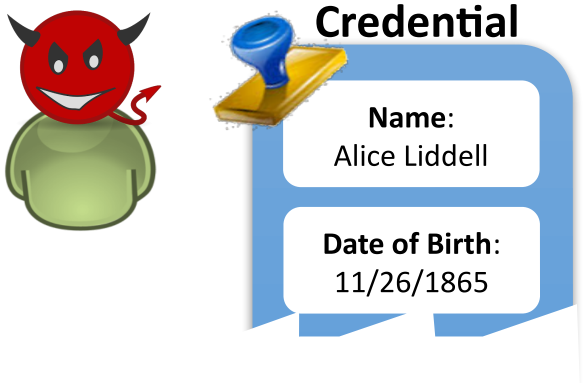
$A_t = g^u$

✓ additions/deletions

(under central management)

✓ Constant size

✓ User privacy



Credential

Name: Alice Liddell

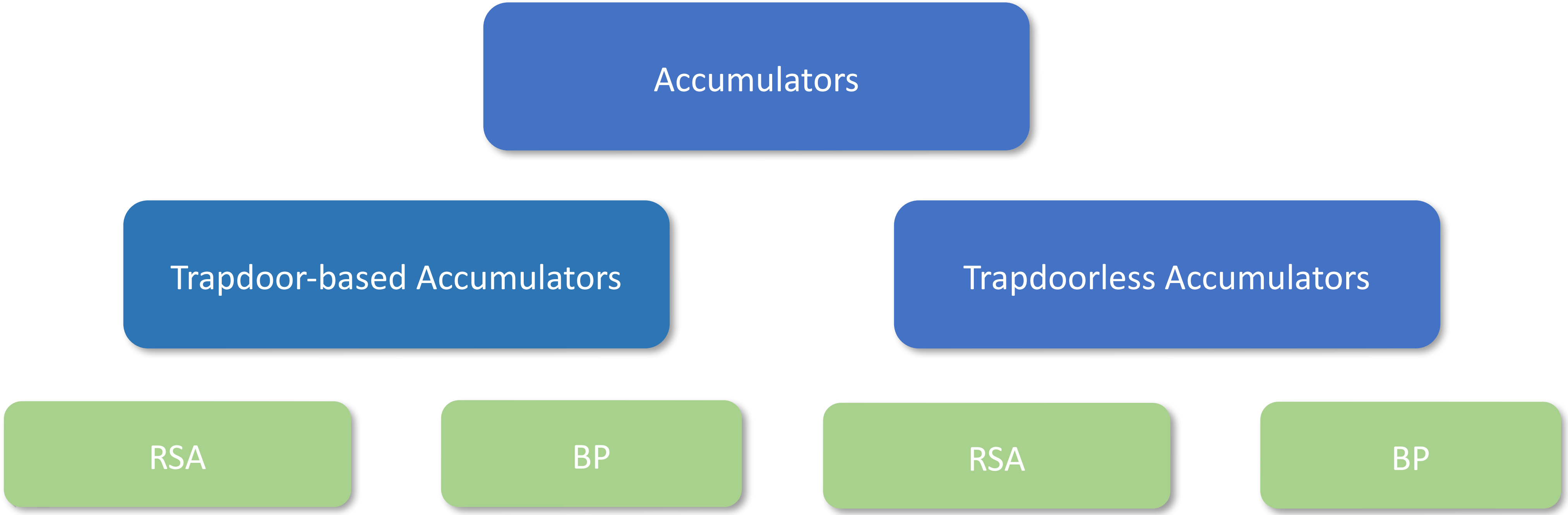
Date of Birth: 11/26/1865

Prove membership

- "x is in A_{t+1}, here is a witness w"
- Verify check: $A_{t+1} = w^x$

- To hide x you need to use ZK proofs which compose efficiently with the accumulator.

Cryptographic Accumulators



Multiple extensions and challenges: batching + aggregation, need for setup, efficiency in trapdoorless scenario….

Final Thoughts

Technology via generic solutions exists

- Efficiency challenges (time-sensitive applications, lightweight devices, gas/storage costs if on blockchain,…)
- Bootstrapping problem
- Accountability without trusted third parties – auto-executed policies
- Back-up, identity restore
- Revocation of identities/credentials
- Efficient Post-quantum secure solutions are basically non-existent