Auditable Credential Anonymity Revocation Based on Privacy-Preserving Smart Contracts

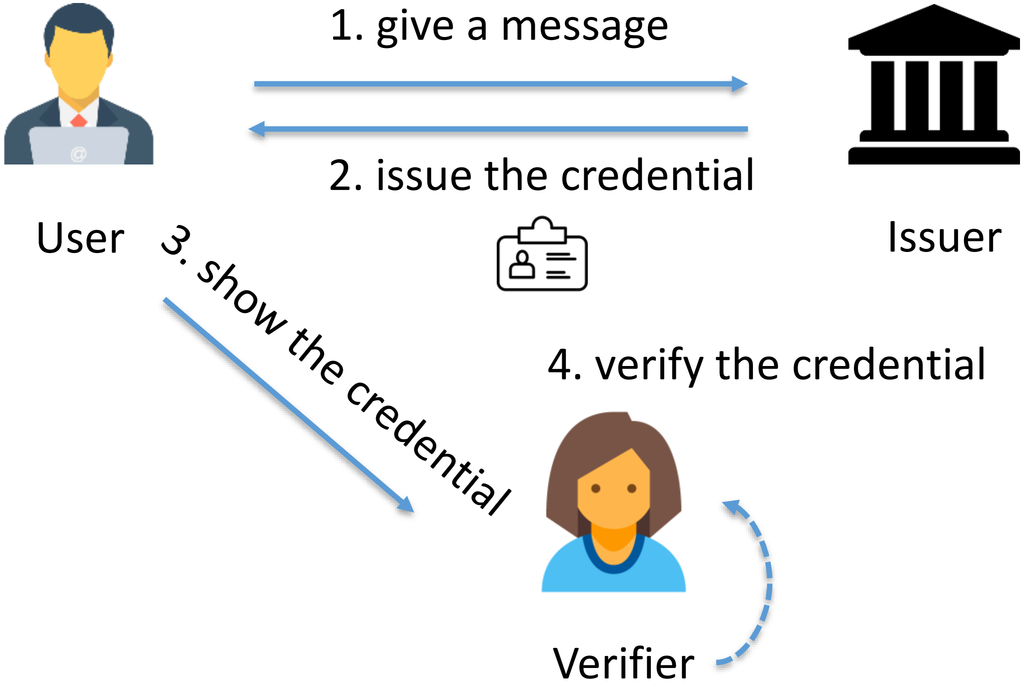
Rujia Li[1,2], David Galindo[2,3], Qi Wang[1]

1 Southern University of Science and Technology, Shenzhen, China
2 University of Birmingham, Birmingham, United Kingdom
3 Fetch.AI, Cambridge, United Kingdom

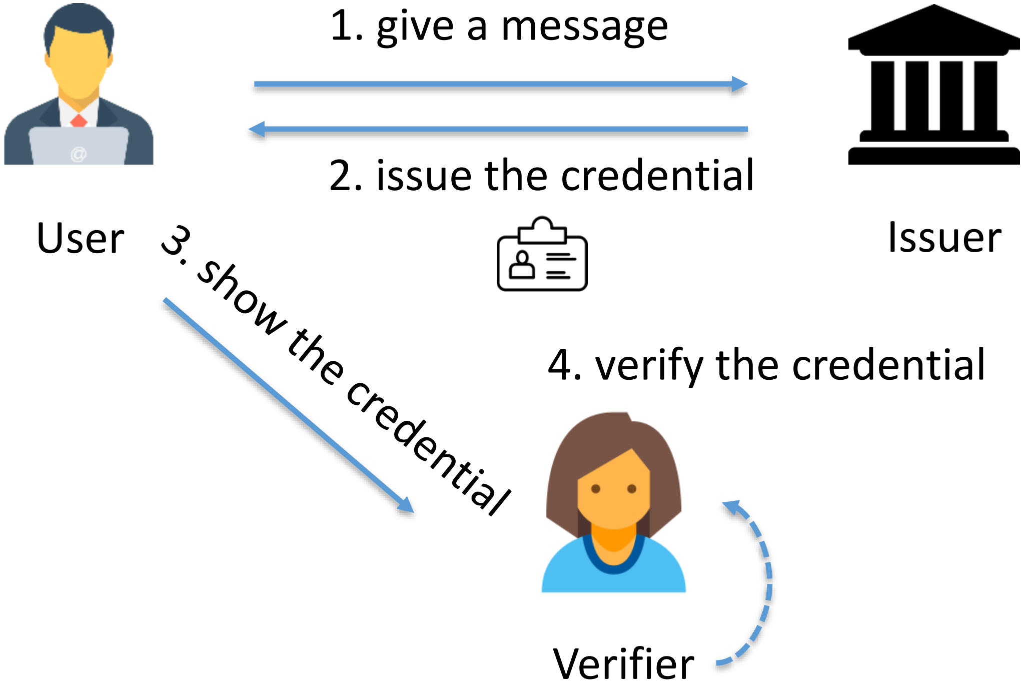
September 27, 2019.

ESORICS CBT'19

Basic credential system



Big brother issue



1. give a message

2. issue the credential

User

Issuer

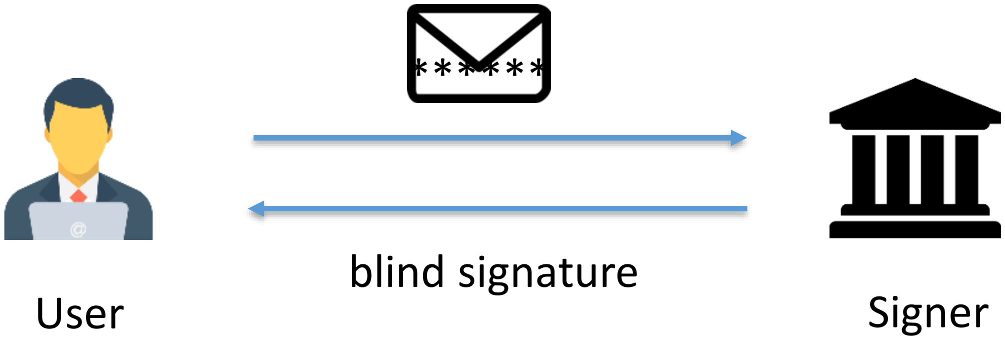
3. show the credential

4. verify the credential

Verifier

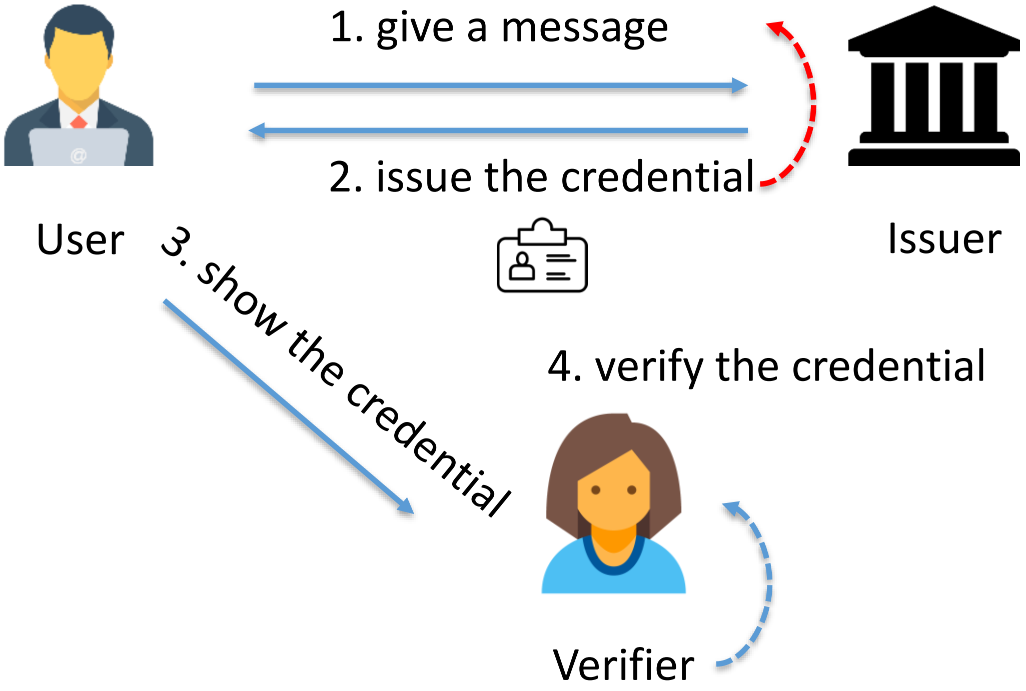
The issuer acts as the "big brother".

Chaum’s blind signature [Cha83]



Blindness: the user hides the message to be signed from the signer.

Anonymous credential system



blindly issuing

The relationship of the credential and its holder.

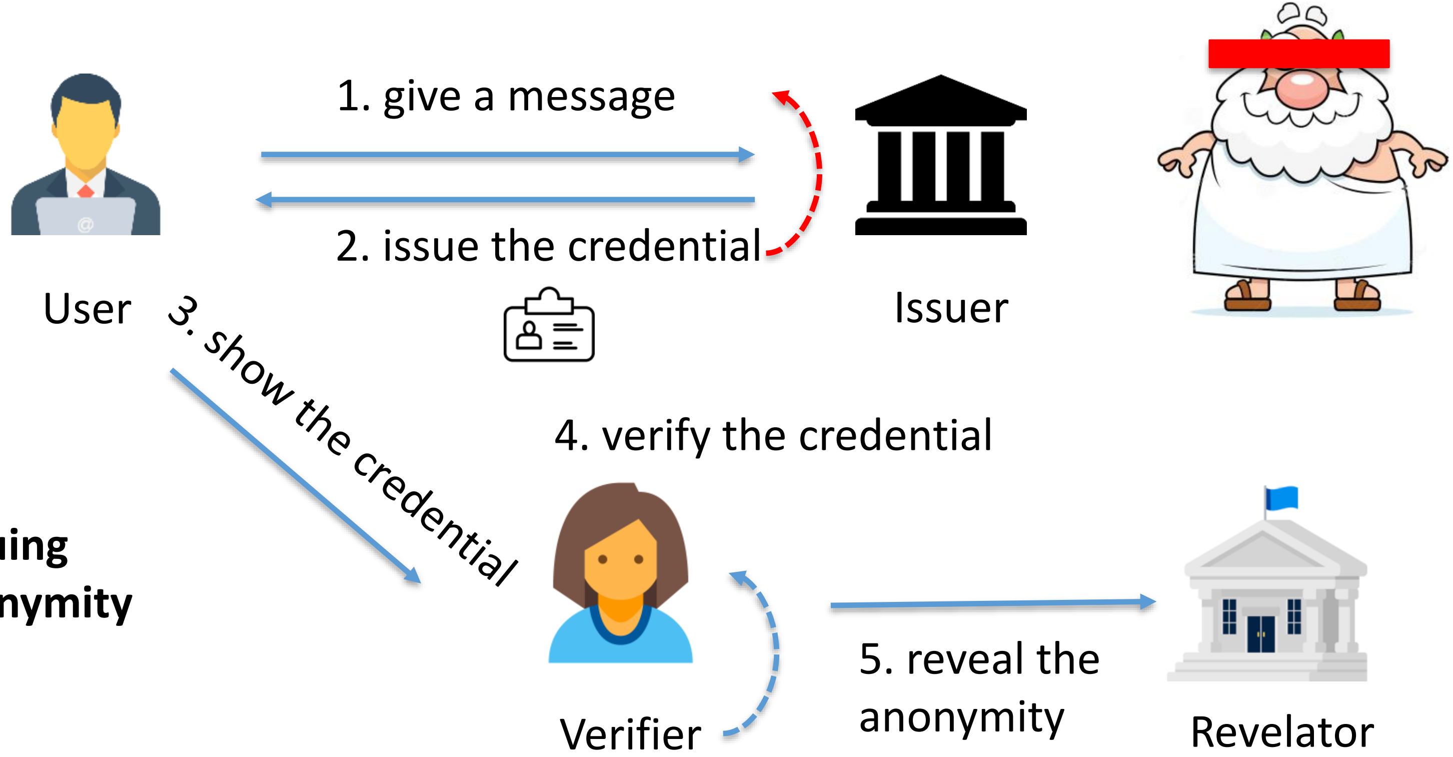
Blind signature does not save the world

“ Blind signatures can protect individuals from the “big brother is watching” situation, it may on the other hand create the situation where these same individuals may be deprived of some other tyPe of protection.

Blind signatures can therefore provide potential problems for law enforcement of some types of crimes.” [VN92]

-- Sebastiaan von Solms and David Naccache

Tradeoff: anonymity revocation



1. give a message

2. issue the credential

User

Issuer

3. show the credential

4. verify the credential

Verifier

5. reveal the anonymity

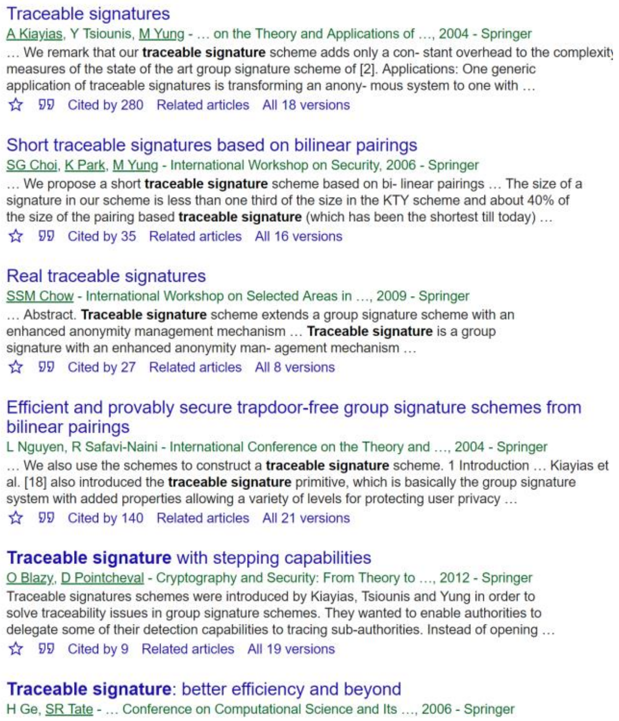
Revelator

1. **blindly issuing**
2. **reveal anonymity**

Who can be the revelator ?

- **(1) The User (Credential Holder)**
  Microsoft’s U-Prove [PZ11].

- **(2) The Judge (Trusted Third Party).**
  IBM’s Identity Mixer [CMS10]
  ABC4Trust [RCS15]
  Traceable Anonymous Certificate [Par+09]
  Fair blind signature scheme [SPC95]
  Traceable signature [KTY04]
  Fair Partially Blind Signatures [RS10]



Traceable signatures
A Kiayias, Y Tsiounis, M Yung - … on the Theory and Applications of …, 2004 - Springer
… We remark that our **traceable signature** scheme adds only a con- stant overhead to the complexit
measures of the state of the art group signature scheme of [2]. Applications: One generic application of traceable signatures is transforming an anony- mous system to one with …
☆ ” Cited by 280 Related articles All 18 versions

Short traceable signatures based on bilinear pairings
SG Choi, K Park, M Yung - International Workshop on Security, 2006 - Springer
… We propose a short **traceable signature** scheme based on bi- linear pairings … The size of a signature in our scheme is less than one third of the size in the KTY scheme and about 40% of the size of the pairing based **traceable signature** (which has been the shortest till today) …
☆ ” Cited by 35 Related articles All 16 versions

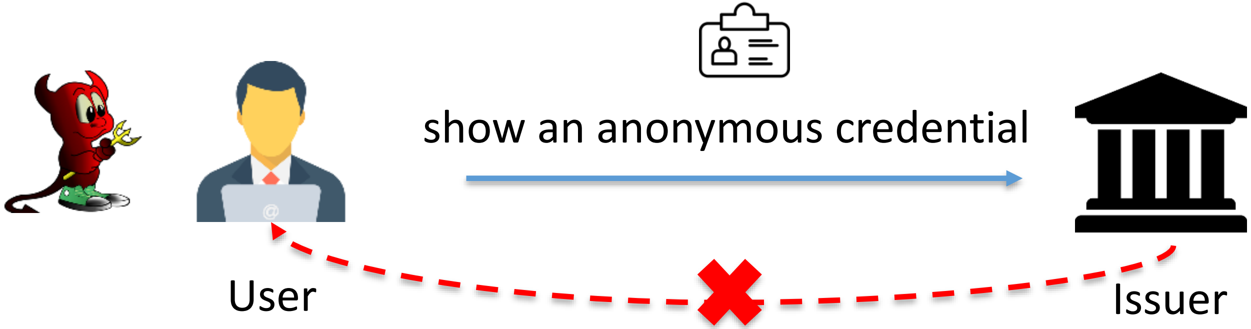
Real traceable signatures
SSM Chow - International Workshop on Selected Areas in …, 2009 - Springer
… Abstract. **Traceable signature** scheme extends a group signature scheme with an enhanced anonymity management mechanism … **Traceable signature** is a group signature with an enhanced anonymity man- agement mechanism …
☆ ” Cited by 27 Related articles All 8 versions

Efficient and provably secure trapdoor-free group signature schemes from bilinear pairings
L Nguyen, R Safavi-Naini - International Conference on the Theory and …, 2004 - Springer
… We also use the schemes to construct a **traceable signature** scheme. 1 Introduction … Kiayias et al. [18] also introduced the **traceable signature** primitive, which is basically the group signature system with added properties allowing a variety of levels for protecting user privacy …
☆ ” Cited by 140 Related articles All 21 versions

Traceable signature with stepping capabilities
O Blazy, D Pointcheval - Cryptography and Security: From Theory to …, 2012 - Springer
Traceable signatures schemes were introduced by Kiayias, Tsiounis and Yung in order to solve traceability issues in group signature schemes. They wanted to enable authorities to delegate some of their detection capabilities to tracing sub-authorities. Instead of opening …
☆ ” Cited by 9 Related articles All 19 versions

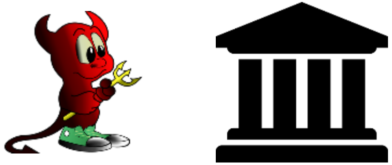
Traceable signature: better efficiency and beyond
H Ge, SR Tate - … Conference on Computational Science and Its …, 2006 - Springer

The user acts as the revelator



Denial problem: the revelator behaves maliciously and rejects to cooperate with the issuer.

The TTP acts as the revelator



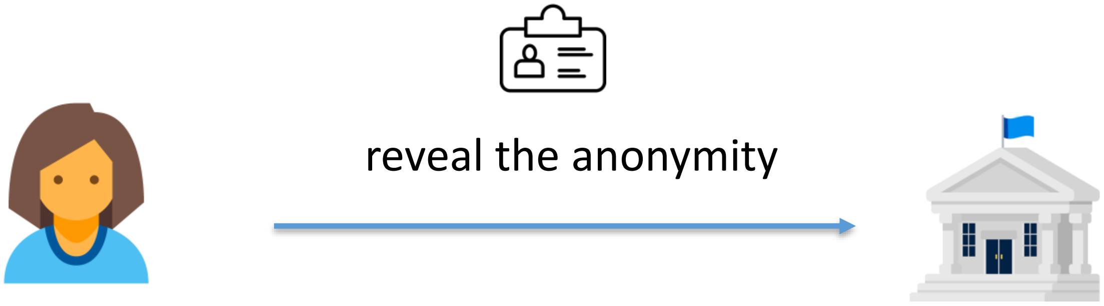
Issuer

- **Lack of transparency**: the revelator and the issuer may conspire to map the credential to the real identity of that user.

Trusted Third Party

- **Non-availability problem**: the revelator may not be always online, which is a single point of failure.

Research problem



reveal the anonymity

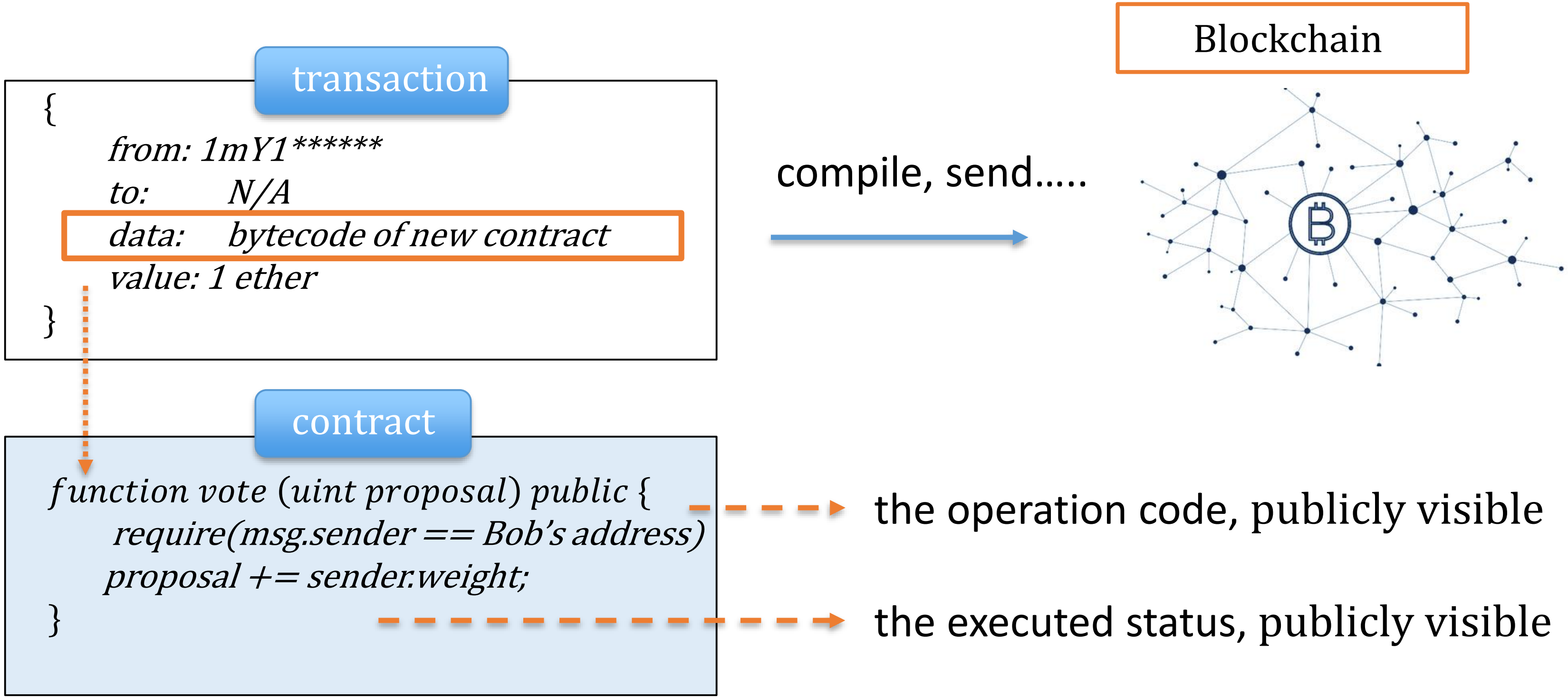
Issuer

Privacy-Preserving Smart Contract

Find a revelator that satisfies the requirements:

(1) Neutral & Keep honest
(2) Auditable and Accountable for her action
(3) High-availability of the service

Smart contract



transaction

```
{
    from: 1mY1******
    to:       N/A
    data:     bytecode of new contract
    value: 1 ether
}
```

compile, send.....

Blockchain

contract

```
function vote (uint proposal) public {
    require(msg.sender == Bob's address)
    proposal += sender.weight;
}
```

the operation code, publicly visible

the executed status, publicly visible

Privacy-preserving smart contract

transaction

```
{
      from: 1mY1******
      to:        N/A
      data:     bytecode of new contract
      value: 1 ether
}
```

compile, send.....

Blockchain

contract

```
function vote (uint proposal) public {
      require(msg.sender == Bob's address)
      proposal += sender.weight;
}
```

the operation code , publicly visible

the executed status

~~publicly visible~~

Privacy-preserving smart contract (PPSC)

Project	Technology
Zether project, [Bunz +19] (eprint 2019)	Zero-knowledge proof
Ekiden project , [Che+19] (EuroS&P)	Trusted execution environment
On/Off-chain SC project, [LPX19] (arXiv, 2019)	On/off-chain contract split
Hawk project, [Kos+16] (IEEE S&P 2016)	Zero-knowledge proof
Enigma project, [ZNP15] (arXiv, 2015)	Multi-party computation

PPSC example: Ekiden (EuroS&P, 2019)

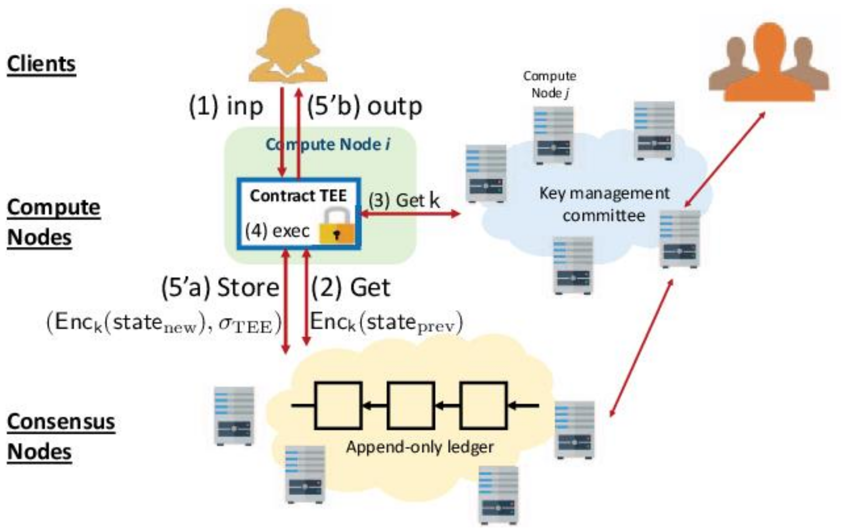


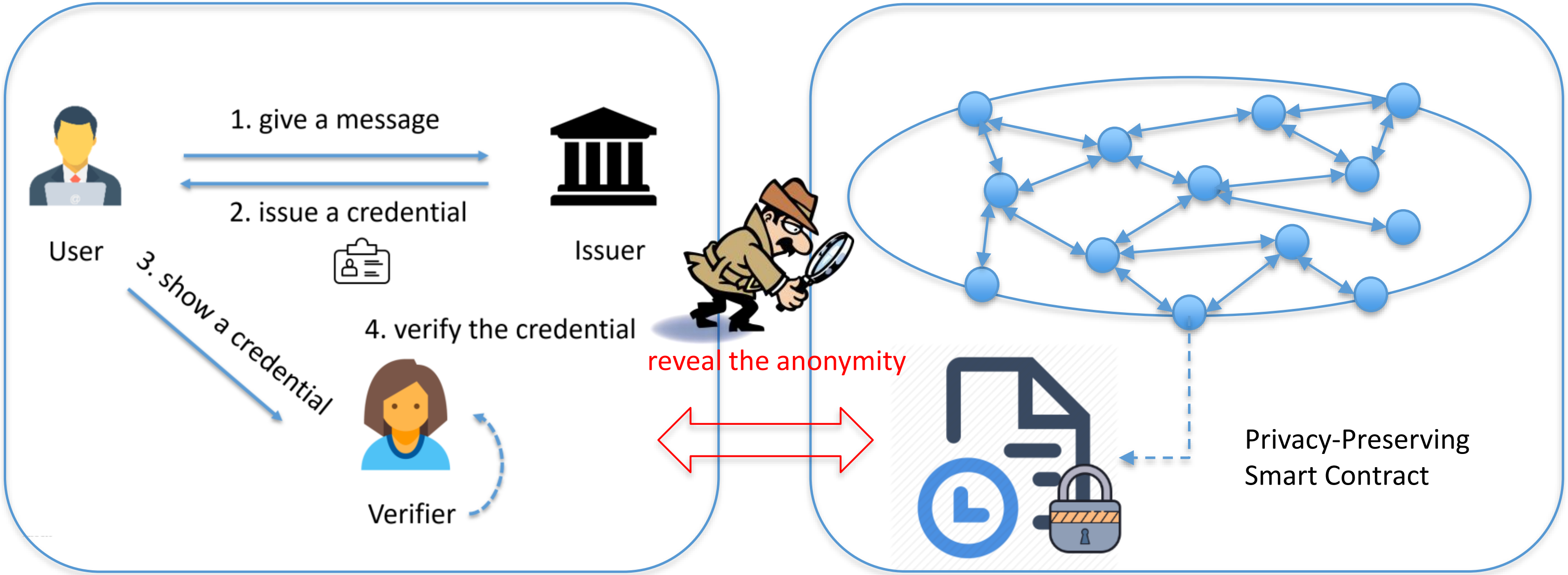
Image source [Che+19]

Clients can create contracts or execute existing ones with secret input.

Compute nodes process requests from clients by running the contract in a contract TEE and generating attestations proving the correctness of state updates.

Consensus nodes maintain a distributed append-only ledger, i.e. a blockchain, by running a consensus protocol.

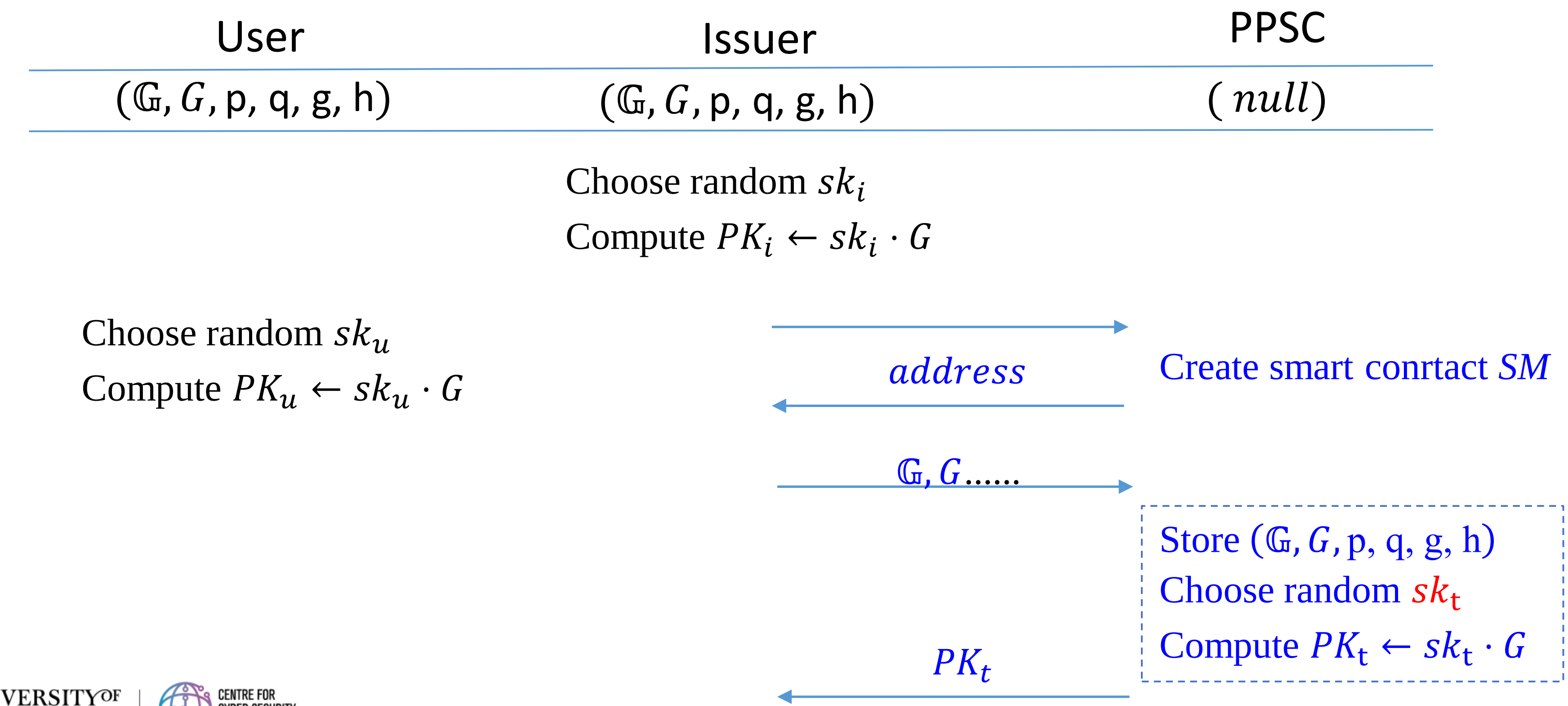
Our anonymity revocation framework



Credential issuing / verifying

Credential tracing / Identity tracing

Parameter generation



User	Issuer	PPSC
$(\mathbb{G}, G, \mathrm{p}, \mathrm{q}, \mathrm{g}, \mathrm{h})$	$(\mathbb{G}, G, \mathrm{p}, \mathrm{q}, \mathrm{g}, \mathrm{h})$	$(\,null)$

Choose random sk_i

Compute $PK_i \leftarrow sk_i \cdot G$

Choose random sk_u

Compute $PK_u \leftarrow sk_u \cdot G$

Create smart conrtact SM

$address$

$\mathbb{G}, G$......

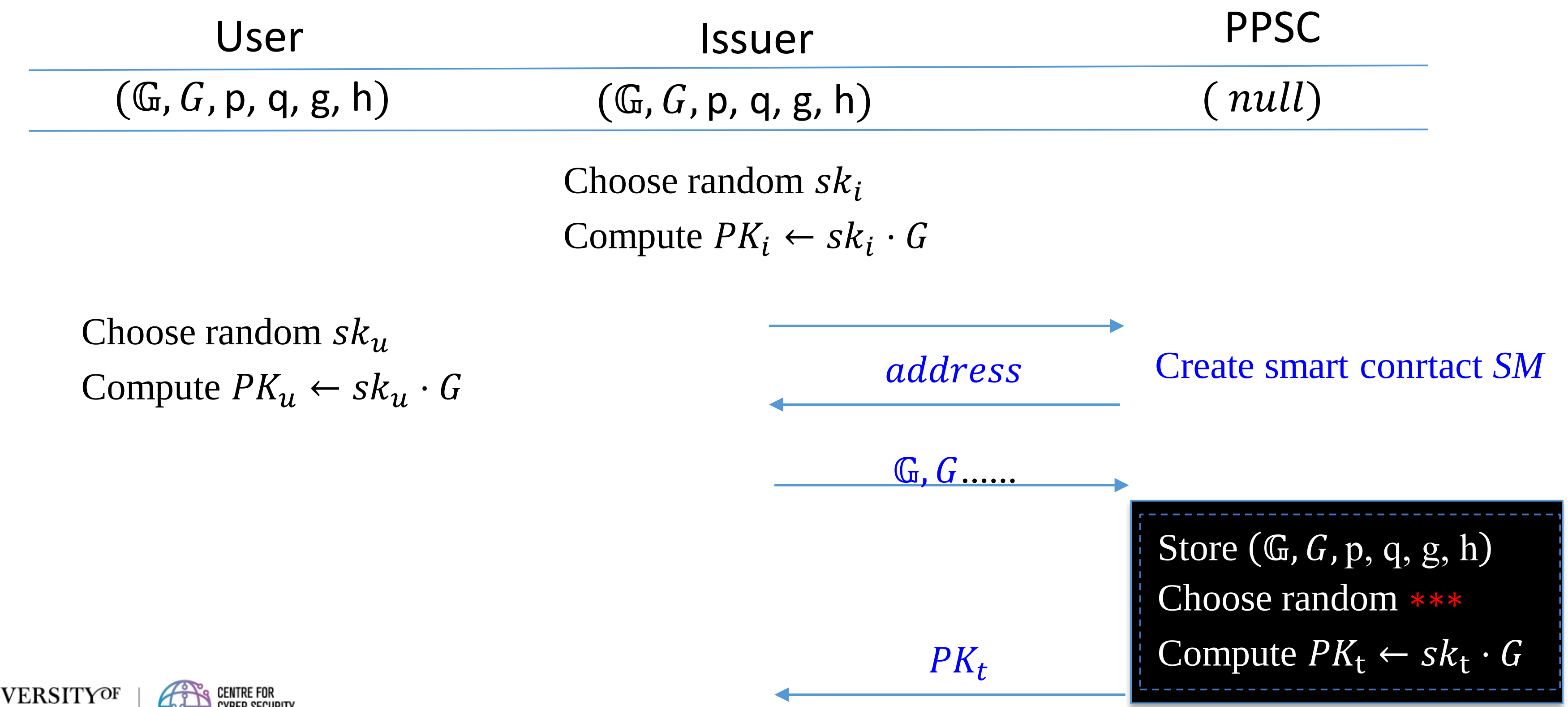
Store $(\mathbb{G}, G, \mathrm{p}, \mathrm{q}, \mathrm{g}, \mathrm{h})$

Choose random sk_{t}

Compute $PK_{\mathrm{t}} \leftarrow sk_{\mathrm{t}} \cdot G$

PK_t

Parameter generation



User	Issuer	PPSC
$(\mathbb{G}, G, \text{p}, \text{q}, \text{g}, \text{h})$	$(\mathbb{G}, G, \text{p}, \text{q}, \text{g}, \text{h})$	$(\,null)$

Choose random sk_i

Compute $PK_i \leftarrow sk_i \cdot G$

Choose random sk_u

Compute $PK_u \leftarrow sk_u \cdot G$

Create smart conrtact SM

$address$

$\mathbb{G}, G$……

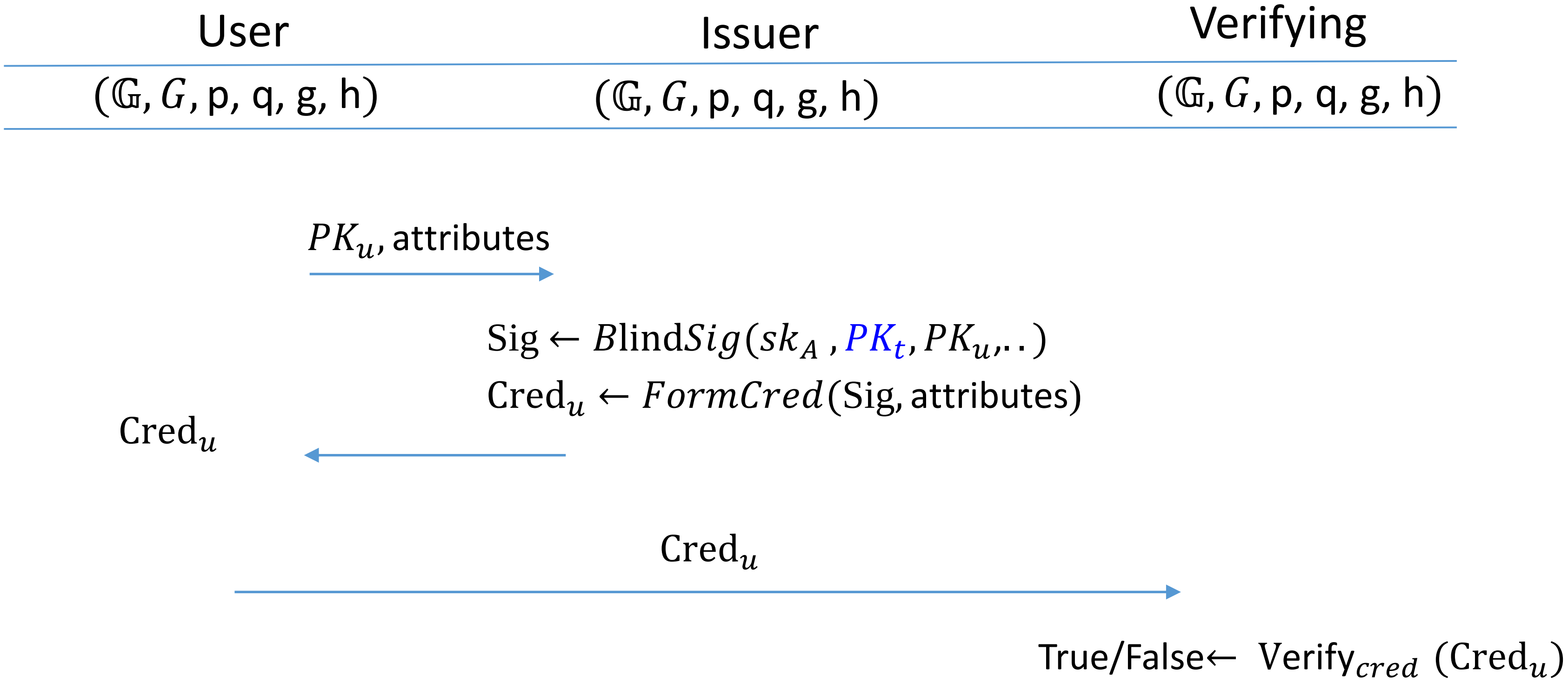
Store $(\mathbb{G}, G, \text{p}, \text{q}, \text{g}, \text{h})$

Choose random ***

Compute $PK_{\mathrm{t}} \leftarrow sk_{\mathrm{t}} \cdot G$

PK_t

Credential issuing / verifying



User

Issuer

Verifying

$(\mathbb{G}, G, \mathrm{p}, \mathrm{q}, \mathrm{g}, \mathrm{h})$

$(\mathbb{G}, G, \mathrm{p}, \mathrm{q}, \mathrm{g}, \mathrm{h})$

$(\mathbb{G}, G, \mathrm{p}, \mathrm{q}, \mathrm{g}, \mathrm{h})$

PK_u, attributes

$\mathrm{Sig} \leftarrow BlindSig(sk_A\ , PK_t, PK_u, ..)$

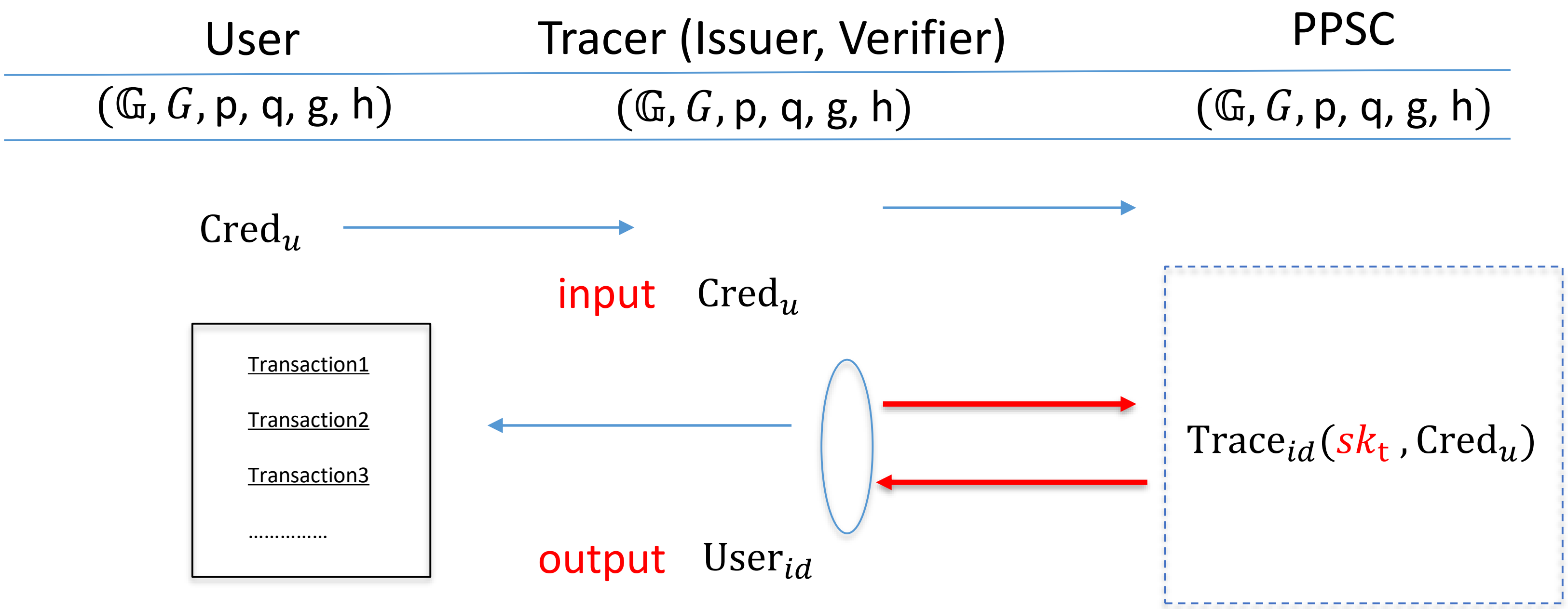
$\mathrm{Cred}_u \leftarrow FormCred(\mathrm{Sig}, \mathrm{attributes})$

Cred_u

Cred_u

True/False← $\mathrm{Verify}_{cred}\ (\mathrm{Cred}_u)$

Identity tracing



User

Tracer (Issuer, Verifier)

PPSC

$(\mathbb{G}, G, \mathrm{p}, \mathrm{q}, \mathrm{g}, \mathrm{h})$

$(\mathbb{G}, G, \mathrm{p}, \mathrm{q}, \mathrm{g}, \mathrm{h})$

$(\mathbb{G}, G, \mathrm{p}, \mathrm{q}, \mathrm{g}, \mathrm{h})$

Cred_u

input Cred_u

Transaction1

Transaction2

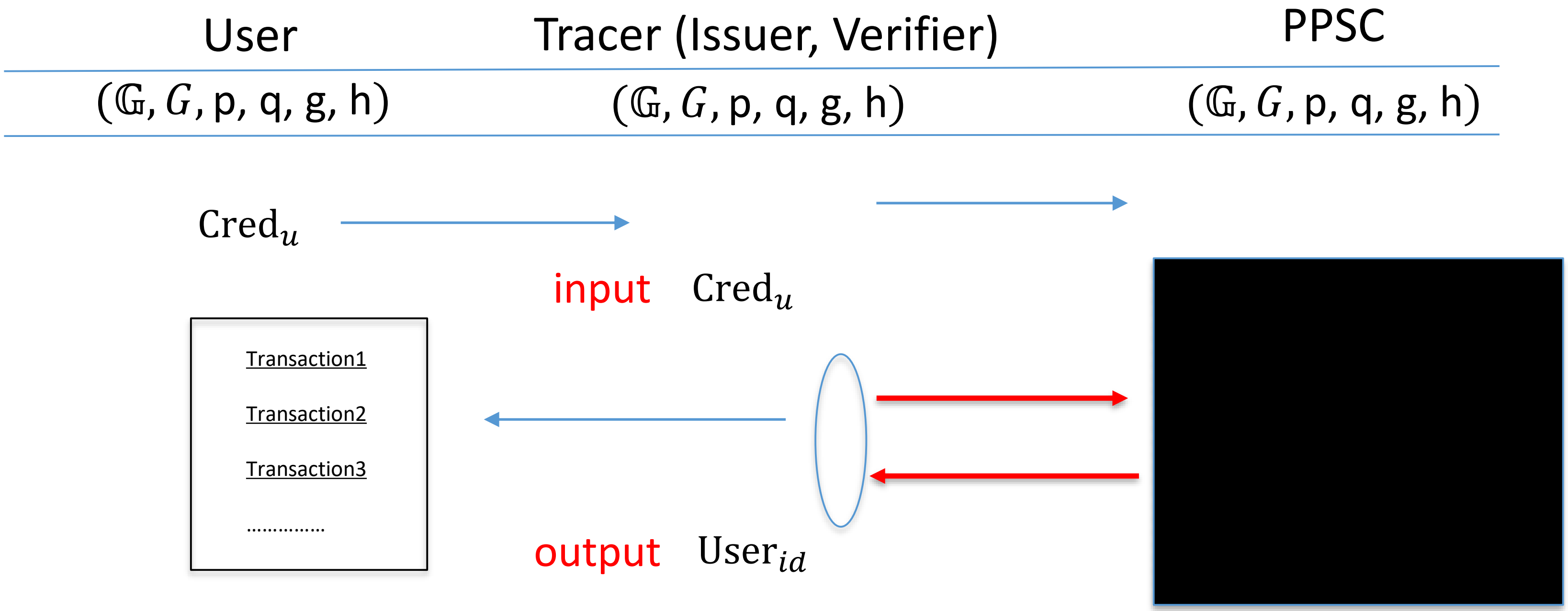
Transaction3

……………

$\mathrm{Trace}_{id}(sk_{\mathrm{t}}, \mathrm{Cred}_u)$

output User_{id}

Identity tracing



User

Tracer (Issuer, Verifier)

PPSC

$(\mathbb{G}, G, \text{p}, \text{q}, \text{g}, \text{h})$

$(\mathbb{G}, G, \text{p}, \text{q}, \text{g}, \text{h})$

$(\mathbb{G}, G, \text{p}, \text{q}, \text{g}, \text{h})$

Cred_u

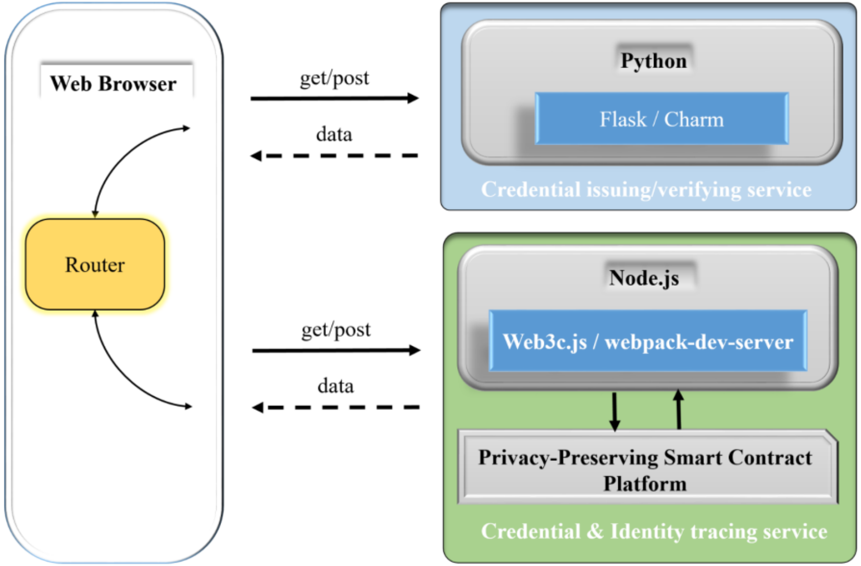
input Cred_u

output User_{id}

1. Auditable transaction records

2. End-to-end secure channel

Implementation



Web Browser
Router
get/post
data
get/post
data
Python
Flask / Charm
Credential issuing/verifying service
Node.js
Web3c.js / webpack-dev-server
Privacy-Preserving Smart Contract Platform
Credential & Identity tracing service

The issuing module (Abe's scheme [AO01]):

- Credential issuing
- Credential verifying
- Tracing inspection
- Python in 168 lines of code.

The tracing module (Oasis Devnet V 1.0):

- Credential tracing
- Identity tracing
- Solidity in 449 lines of code.

Source code: https://github.com/typex-1/auditable-credential-core.

Evaluation

Operations	Performance	Size	Gas	Latency
Parameter generation	0.84	260	20672	14.781
Credential issuing	7.40	0	0	1.601
Credential verifying	2.32	0	0	1.175
Credential tracing	3.06	132	390261	17.538
Identity tracing	4.55	132	388944	18.905
	(milliseconds)	(bytes)		(seconds)

Test result, https://github.com/typex-1/auditable-credential-core/tree/master/test/result

Framework features

Denial problem	Smart conrtact code self-execution
Lack of transparency	Auditable transaction invoking records
Non-availability problem	Blockchain distributed network

Future work

Latency issue:

The average latency of credential tracing and identity tracing is approximately eighteen seconds, which would be a primary drawback of our system.

Scalability issue:

Low throughput of on-chain transaction is a roadblock. The flexible smart contract makes our scheme easier to support batch anonymity revealing.

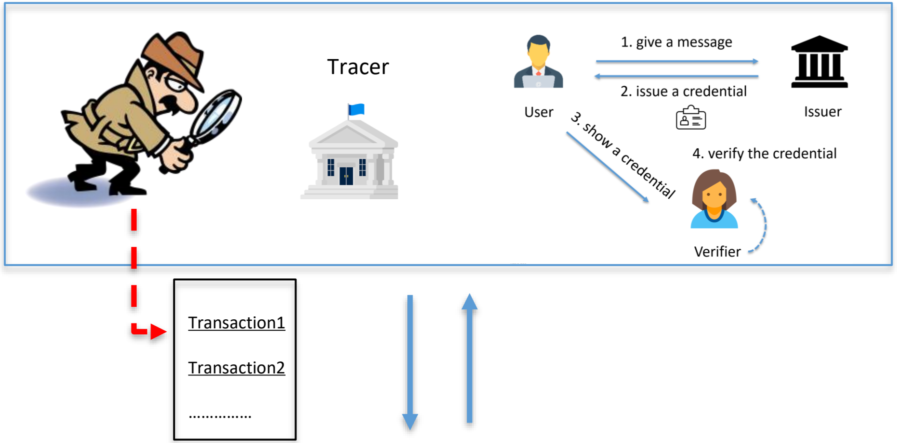
Weaken the assumptions

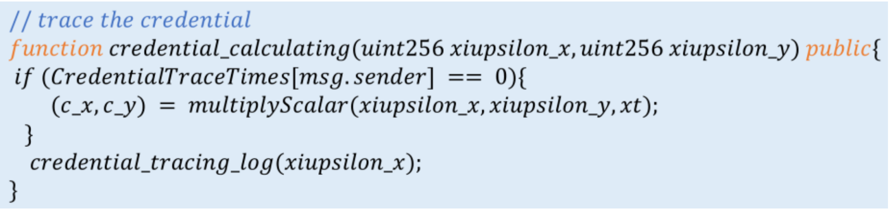
Using online/offline keys to weaken the trust assumptions on the PPSC platform.

References

- [AO01] Masayuki Abe and Miyak Ohkub. “Provably Secure air Blind Signatures with Tight Revocation”.In:International Conference on the Theory and Application of Cryptology and Information Se-curity(ASIACRYPT). Springer. 2001, pp. 583–601
- [Cha83] David Chaum. “Blind signatures for untraceable payments”. In:Advances in cryptology. Springer.1983, pp. 199–203
- [VN92] Sebastiaan Von Solms and David Naccache. “On blind signatures and perfect crimes”. In:Com-puters & Security11.6 (1992), pp. 581–583.
- [PZ11] Christian Paquin and Greg Zaverucha. “U-prove cryptographic specification v1. 1”. In:TechnicalReport, Microsoft Corporation (2011).
- [KG16] Homomorphic Encryption and Smart Contracts: Privacy and Transparency. Retrieved from https://www.newsbtc.com/2016/04/17/homomorphic-encryption-and-smart-contracts-for-privacy-and-transparency/
- [Che+19] Raymond Cheng et al. “Ekiden: A Platform for Confidentiality-Preserving, Trustworthy, andPerformant Smart Contracts”. In:2019 IEEE European Symposium on Security and Privacy(EuroS&P). IEEE. 2019, pp. 185–200.
- [PZ11] Christian Paquin and Greg Zaverucha. “U-prove cryptographic specification v1. 1”. In:TechnicalReport, Microsoft Corporation(2011).
- [RCS15] Kai Rannenberg, Jan Camenisch, and Ahmad Sabouri. “Attribute-based credentials for trust”.In:Identity in the Information Society, Springer(2015).
- [KTY04] Aggelos Kiayias, Yiannis Tsiounis, and Moti Yung. “Traceable signatures”. In:InternationalConference on the Theory and Applications of Cryptographic Techniques. Springer. 2004, pp. 571–589
- [Par+09] S Park et al. “Traceable Anonymous Certificate”. In: (2009).
- [SPC95] Markus Stadler, Jean-Marc Piveteau, and Jan Camenisch. “Fair blind signatures”. In:Interna-tional Conference on the Theory and Applications of Cryptographic Techniques. Springer. 1995,pp. 209–219
- [RS10] Markus R¨uckert and Dominique Schr¨oder. “Fair Partially Blind Signatures”. In:AFRICACRYPT.2010.
- [Kos+16] Ahmed Kosba et al. “Hawk: The blockchain model of cryptography and privacy-preserving smart contracts”. In:2016 IEEE symposium on security and privacy (SP). IEEE. 2016, pp. 839–858
- [B¨un+19] Benedikt B¨unz et al. “Zether: Towards Privacy in a Smart Contract World.” In:IACR CryptologyePrint Archive2019 (2019), p. 191
  [ZNP15] Guy Zyskind, Oz Nathan, and Alex Pentland. “Enigma: Decentralized computation platformwith guaranteed privacy”. In:arXiv preprint arXiv:1506.03471(2015)
- [LPX19] Chao Li, Balaji Palanisamy, and Runhua Xu. “Scalable and Privacy-preserving Design of On/Off-chain Smart Contracts”. In:arXiv preprint arXiv:1902.06359(2019)

Backup: framework features





```
// trace the credential
function credential_calculating(uint256 xiupsilon_x, uint256 xiupsilon_y) public{
 if (CredentialTraceTimes[msg.sender] == 0){
     (c_x, c_y) = multiplyScalar(xiupsilon_x, xiupsilon_y, xt);
  }
   credential_tracing_log(xiupsilon_x);
}
```

- **Simple:** the issuing, verification are executed independently from the Blockchain.

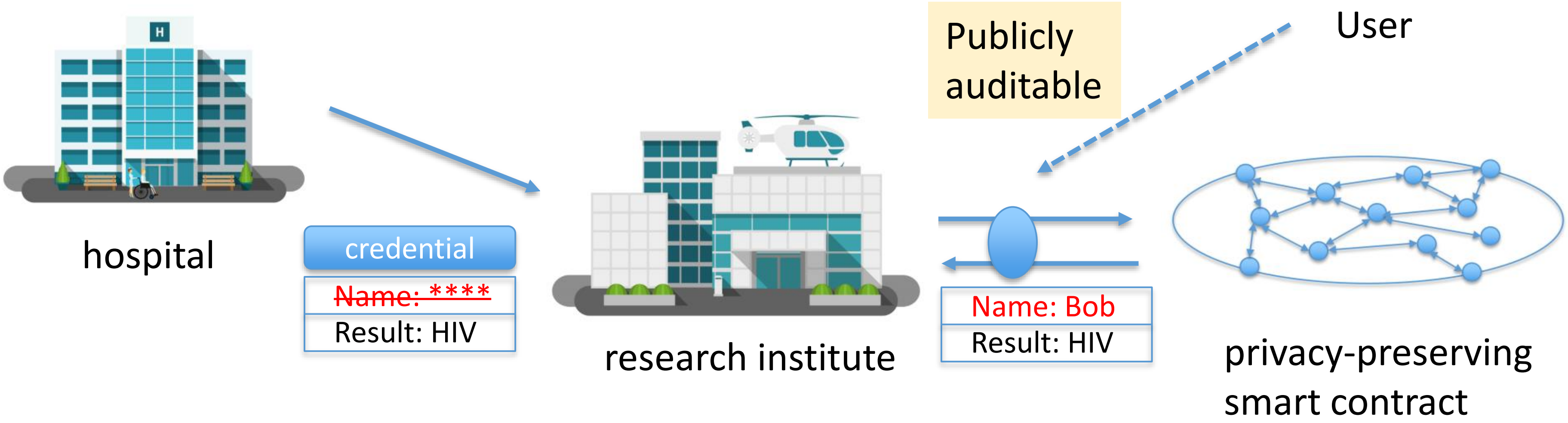
$$I_{cred} = (\xi^{\upsilon})^{x_t} = g^{\gamma\upsilon x_t} = y_t^{\gamma\upsilon} = \zeta_1.$$

- **Efficient:** One-time elliptic-curve exponentiation is adequate to conduct the complete tracing activity.

Backup: Example Application

Medical record protection system:

1. The patient records ==✖=== patients’ real identities
2. Disclose their identities with auditability by invoking PPSC.



hospital

credential

~~Name: ****~~

Result: HIV

research institute

Publicly auditable

User

Name: Bob

Result: HIV

privacy-preserving smart contract