-ZKProof 7 Sofia-

Towards Adoption of distributed ledgers in traditional financial institutes

Presenters: Shaltiel Eloul and Yash Satsangi

Introducing PADL: “a framework for exploring a Private, Auditable and Distributed Ledger”

Authors: Shaltiel Eloul, Yash Satsangi, Yeoh Zho, Omar Amer, George Papadopoulos, and Marco Pistoia

Global Tech. Applied Research, J.P. Morgan Chase

JPMorganChase

Controlling every Step

Helping, teaching, observing…

Please… give me some privacy!

Now I can deal with a distributed system ☺

Controlling every Step
Helping, teaching, observing...
Please... give me some privacy!
Now I can deal with a distributed system ☺

ZK proof for financial institutes enabling Distribution of ledger:

- *Remote/privacy preserving Auditing*
- *Reduce settlements overhead*

Typically not a decentralized system problem, but other requirements appears

Considerations for encrypted distributed ledgers in banking systems

- **Multi-assets/multi-actors in transaction**
- **No-Trusted setup** - initial stage.
- **Auditing and Selective disclosure** - and maybe not full anonymity.
- **Flexibility and customization** to the financial market:

  Open/Usable/Established cryptography, independent of platform, evm compatibility.

Attractive starting points for constructing a transaction schemes: ***ZKLedger, and Zether***

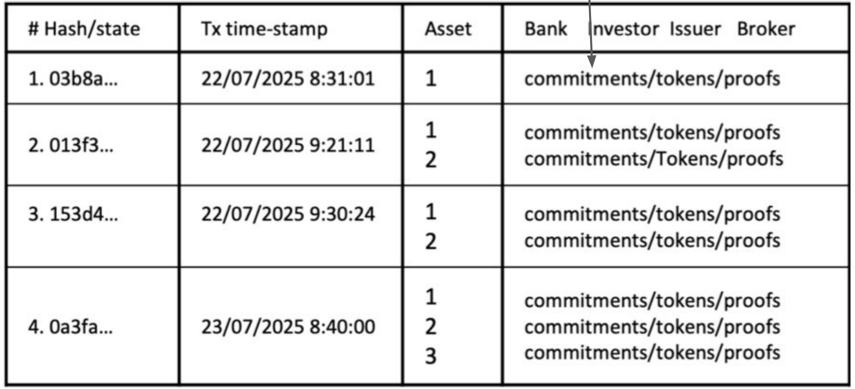
Institute Considerations	Multi-asset	no-trusted setup	Auditing	Smart contract
PADL	🟩		🟩	🟩
Zether	🟥	🟩	🟥	🟩
zkLedger	🟥	🟩	🟩	🟥

PADL - A framework to explore Private Auditable and Distributed Multi-asset Ledger

A Private Auditable and Distributed Multi-asset Ledger

$$\mathsf{Cell}_{t,p,a} = \{\mathsf{cm}_{t,p,a}, \mathsf{tk}_{t,p,a}, \mathsf{cm}'_{t,p,a}, \mathsf{tk}'_{t,p,a}, \pi^{\mathsf{A}}_{t,p,a}, \pi^{\mathsf{C}}_{t,p,a}, \pi^{\mathsf{C}'}_{t,p,a}, \pi^{\mathsf{EQ}}_{t,p,a}\}$$

t-transaction, p-participant, a-asset



# Hash/state	Tx time-stamp	Asset	Bank Investor Issuer Broker
1. 03b8a...	22/07/2025 8:31:01	1	commitments/tokens/proofs
2. 013f3...	22/07/2025 9:21:11	1 2	commitments/tokens/proofs commitments/Tokens/proofs
3. 153d4...	22/07/2025 9:30:24	1 2	commitments/tokens/proofs commitments/tokens/proofs
4. 0a3fa...	23/07/2025 8:40:00	1 2 3	commitments/tokens/proofs commitments/tokens/proofs commitments/tokens/proofs

3D table: here assets are also confidential

Each row represents a multi-assets transaction, and each column represents a participant.

A Private Auditable and Distributed Multi-asset Ledger

Homomorphic Ledger with Privacy via encryption
Auditability via zk-proofs (Σ-protocols, Bulletproofs)

Each cell in the 3D table,
can be verified or audited and it has the structure:

$$\mathsf{Cell}_{t,p,a} = \{\mathsf{cm}_{t,p,a}, \mathsf{tk}_{t,p,a}, \mathsf{cm}'_{t,p,a}, \mathsf{tk}'_{t,p,a}, \pi^{\mathsf{A}}_{t,p,a}, \pi^{\mathsf{C}}_{t,p,a}, \pi^{\mathsf{C}'}_{t,p,a}, \pi^{\mathsf{EQ}}_{t,p,a}\}$$

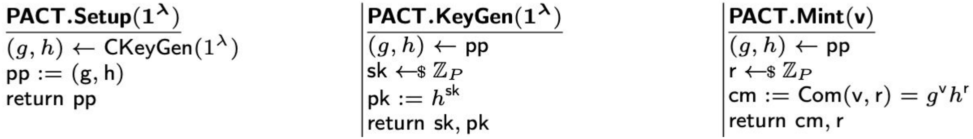
t-transaction, p-participant, a-asset

- cm(v,r) - pedersen commit, tk(pk,r) - token
- cm’, tk’ - Complementary commit and token
- Proof-of-Equivalence: commit to the same value/and signing
- Proof-of-Asset: RangeProof
- Proof-of-Consistency: for randomness between tk/cm
- Extraction (bruteforce): cm/tk

Actors interaction example

https://github.com/jpmorganchase/PADL

Main crypto functions:

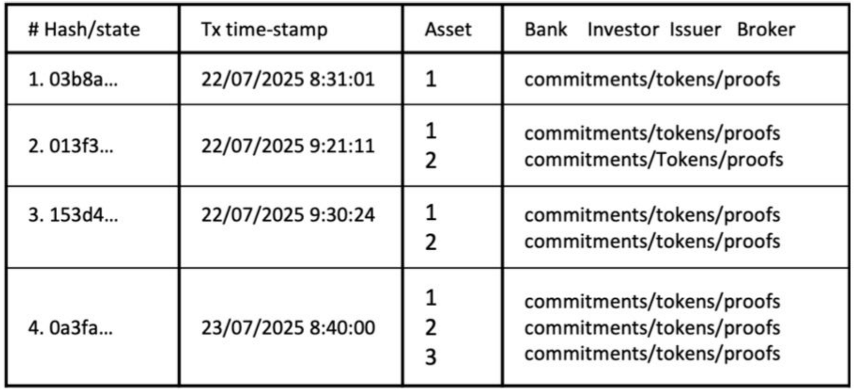


```
PACT.Setup($1^\lambda$)
$(g, h) \leftarrow$ CKeyGen($1^\lambda$)
pp := (g, h)
return pp
```

```
PACT.KeyGen($1^\lambda$)
$(g, h) \leftarrow$ pp
sk $\leftarrow\$ \mathbb{Z}_P$
pk := $h^{\mathsf{sk}}$
return sk, pk
```

```
PACT.Mint(v)
$(g, h) \leftarrow$ pp
r $\leftarrow\$ \mathbb{Z}_P$
cm := Com(v, r) = $g^{\mathsf{v}} h^{\mathsf{r}}$
return cm, r
```

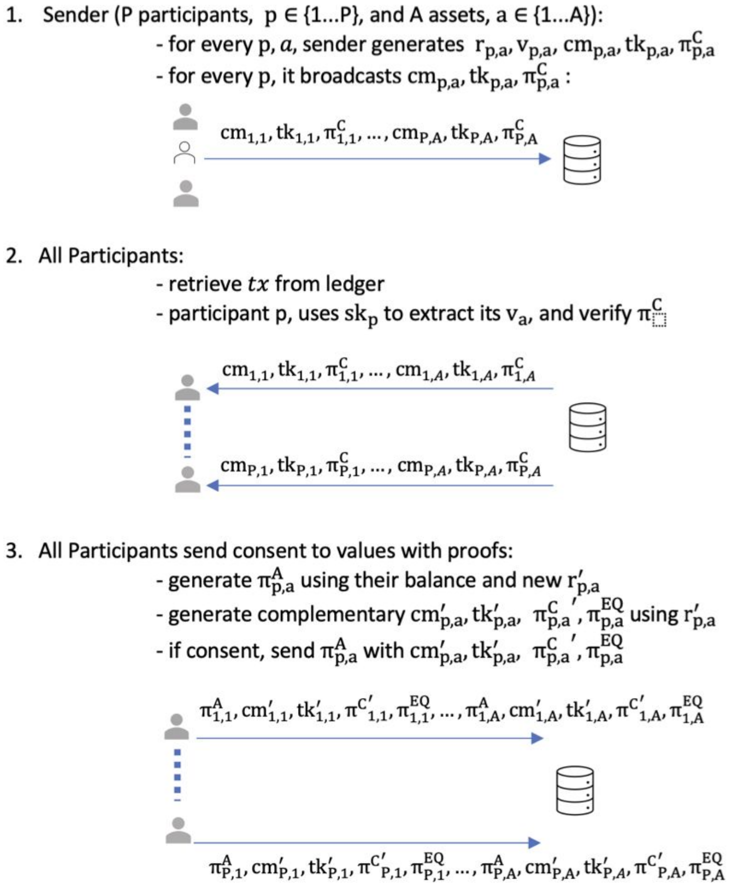
3D table: here assets are also confidential



# Hash/state	Tx time-stamp	Asset	Bank Investor Issuer Broker
1. 03b8a...	22/07/2025 8:31:01	1	commitments/tokens/proofs
2. 013f3...	22/07/2025 9:21:11	1 2	commitments/tokens/proofs commitments/Tokens/proofs
3. 153d4...	22/07/2025 9:30:24	1 2	commitments/tokens/proofs commitments/tokens/proofs
4. 0a3fa...	23/07/2025 8:40:00	1 2 3	commitments/tokens/proofs commitments/tokens/proofs commitments/tokens/proofs

Cryptography in BN254, Secp256 developed in Rust, Interfaced with python, and deployable as smart-contracts with verifications (including bulletproof verification onchain).

Multi Asset Tx Scheme



1. Sender (P participants, p ∈ {1…P}, and A assets, a ∈ {1…A}):
   - for every p, *a*, sender generates $r_{p,a}, v_{p,a}, cm_{p,a}, tk_{p,a}, \pi^C_{p,a}$
   - for every p, it broadcasts $cm_{p,a}, tk_{p,a}, \pi^C_{p,a}$:

$cm_{1,1}, tk_{1,1}, \pi^C_{1,1}, \ldots, cm_{P,A}, tk_{P,A}, \pi^C_{P,A}$

2. All Participants:
   - retrieve *tx* from ledger
   - participant p, uses sk_p to extract its v_a, and verify $\pi^C_{\square}$

$cm_{1,1}, tk_{1,1}, \pi^C_{1,1}, \ldots, cm_{1,A}, tk_{1,A}, \pi^C_{1,A}$

$cm_{P,1}, tk_{P,1}, \pi^C_{P,1}, \ldots, cm_{P,A}, tk_{P,A}, \pi^C_{P,A}$

3. All Participants send consent to values with proofs:
   - generate $\pi^A_{p,a}$ using their balance and new $r'_{p,a}$
   - generate complementary $cm'_{p,a}, tk'_{p,a}, \pi^{C}_{p,a}{}', \pi^{EQ}_{p,a}$ using $r'_{p,a}$
   - if consent, send $\pi^A_{p,a}$ with $cm'_{p,a}, tk'_{p,a}, \pi^{C}_{p,a}{}', \pi^{EQ}_{p,a}$

$\pi^A_{1,1}, cm'_{1,1}, tk'_{1,1}, \pi^{C'}{}_{1,1}, \pi^{EQ}_{1,1}, \ldots, \pi^A_{1,A}, cm'_{1,A}, tk'_{1,A}, \pi^{C'}{}_{1,A}, \pi^{EQ}_{1,A}$

$\pi^A_{P,1}, cm'_{P,1}, tk'_{P,1}, \pi^{C'}{}_{P,1}, \pi^{EQ}_{P,1}, \ldots, \pi^A_{P,A}, cm'_{P,A}, tk'_{P,A}, \pi^{C'}{}_{P,A}, \pi^{EQ}_{P,A}$

“Injective” Tx (zkLedger/Zether ‘like’), single asset

1. ‘Sender’ sends all proofs of positivity (but only works for credit).

   Note: ZKLedger suggests Additional disjunctive proof, which would require also additional two complementary commits, and consistency proof.

2. No consent is needed, but can be added.

Extra communication is only relevant when consent is anyway required, i.e. Asset Exchange, but leads to:

- Anonymity, as all provides proof of asset (remove ‘or’ proof)
- Concurrent range-proofs generation.
- Providing proof of asset, also signs the Tx (with extra sigma protocol).

Auditing

Selective disclosure of a cell:

$$\mathsf{Cell}_{t,p,a} = \{\mathsf{cm}_{t,p,a}, \mathsf{tk}_{t,p,a}, \mathsf{cm}'_{t,p,a}, \mathsf{tk}'_{t,p,a}, \pi^{\mathsf{A}}_{t,p,a}, \pi^{\mathsf{C}}_{t,p,a}, \pi^{\mathsf{C}'}_{t,p,a}, \pi^{\mathsf{EQ}}_{t,p,a}\underline{, \mathsf{tk}^{I}_{t,p,a}, \pi^{\mathsf{C}^{I}}_{t,p,a}}\}$$

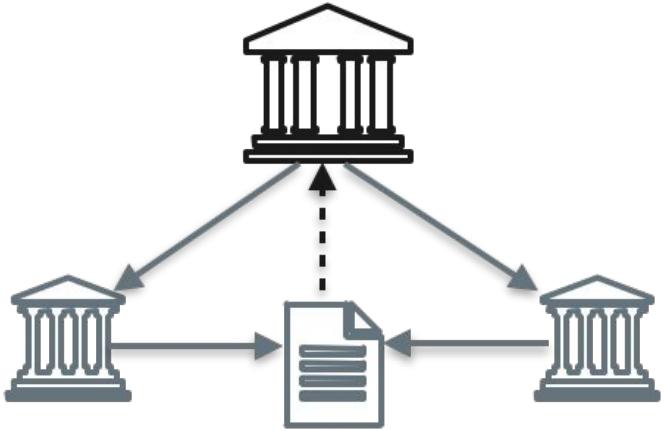
Each cell, can contain additional audit tk to auditing pk's, or committing data KYC.

In a Settlement Bank scenario, the cell can shrink for example to:

$$\mathsf{Cell}_{t,p,a} = \{\mathsf{cm}_{t,p,a}, \mathsf{tk}_{t,p,a}, \pi^{\mathsf{C}}_{t,p,a}, \mathsf{tk}^{I}_{t,p,a}, \pi^{\mathsf{C}^{I}}_{t,p,a}\}$$

Privacy Preserving Auditing

1. Asset Balance, Average, etc.
2. Inter-Asset Auditing



Inter Confidential Asset Auditing

$$\frac{\sum_{t\in Txs}\mathsf{v}_{t,p,a^*}}{\sum_{a\in A}\sum_{t\in Txs}\mathsf{v}_{t,p,a}} < f$$ = D/N

Liquidity

Prove in ZK that Ratio is under value.

$$\mathsf{v}_r = D\sum_{a\in A}\sum_{t\in Txs}\mathsf{v}_{t,p,a} - N\sum_{t\in Txs}\mathsf{v}_{t,p,a^*}.$$

$$\prod_{t\in Txs}\mathsf{cm}'_{t,p,a^*} := c_1,\ \prod_{a\in A}\prod_{t\in Txs}\mathsf{cm}'_{t,p,a} := c_2,\ \text{and},\ c_r = c_2^D/c_1^N$$

$$\frac{\sum_{t\in txs_1\subset T}\mathsf{v}_{t,p,a} = \Sigma v_1}{\sum_{t\in txs_2\subset T}\mathsf{v}_{t,p,a} = \Sigma v_2} = Rate,$$ = D/N

$$\prod_{t\in txs_1}\mathsf{cm}_{t,p,a} := c_1,\ \prod_{t\in txs_2}\mathsf{tk}_{t,p,a} := \tau_1,\quad \prod_{t\in txs_2}\mathsf{cm}_{t,p,a} := c_2,\ \prod_{t\in txs_2}\mathsf{tk}_{t,p,a} := \tau_2$$

Rate/Traceability

$$c = c_1^N \cdot c_2^{-D},\ \text{and}\ \tau = \tau_1^N \cdot \tau_2^{-D} \qquad dlog_c\tau \equiv dlog_c c^{\mathsf{sk}}$$

Example: Bond Exchange and Coupon Rate Example in PADL

Bond Exchange Scenario

- 2 investors buying bonds
- 2-year maturity, 10% yearly coupon rate
- Par value: $10 per bond unit

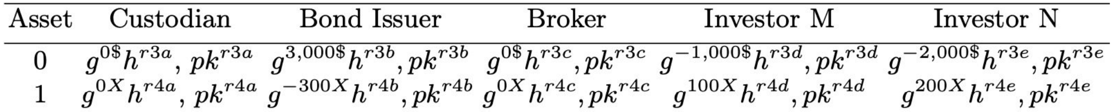
Actors involved

- **Custodian:** Safekeeper, mints USD tokens
- **Bond Issuer:** Issues bonds, borrows USD
- **Broker:** Manages the exchange
- **Investors:** invest in bonds.

Privacy Requirements

- Broker knows bond allocation but not other investors' data
- Custodian only knows issued amounts, not ownership details
- Bond issuer knows the total USD received, not individual contributions
- Investors don't know other investors' transactions
- Transactions are encrypted, but timestamps are public

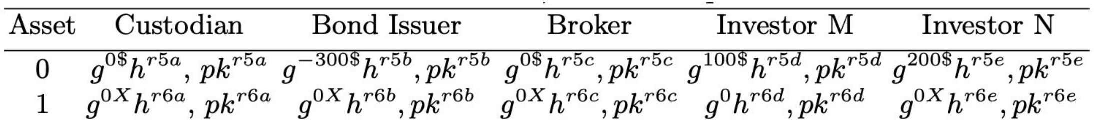
Exchange bond



Asset	Custodian	Bond Issuer	Broker	Investor M	Investor N
0	$g^{0\$}h^{r3a}, pk^{r3a}$	$g^{3{,}000\$}h^{r3b}, pk^{r3b}$	$g^{0\$}h^{r3c}, pk^{r3c}$	$g^{-1{,}000\$}h^{r3d}, pk^{r3d}$	$g^{-2{,}000\$}h^{r3e}, pk^{r3e}$
1	$g^{0X}h^{r4a}, pk^{r4a}$	$g^{-300X}h^{r4b}, pk^{r4b}$	$g^{0X}h^{r4c}, pk^{r4c}$	$g^{100X}h^{r4d}, pk^{r4d}$	$g^{200X}h^{r4e}, pk^{r4e}$

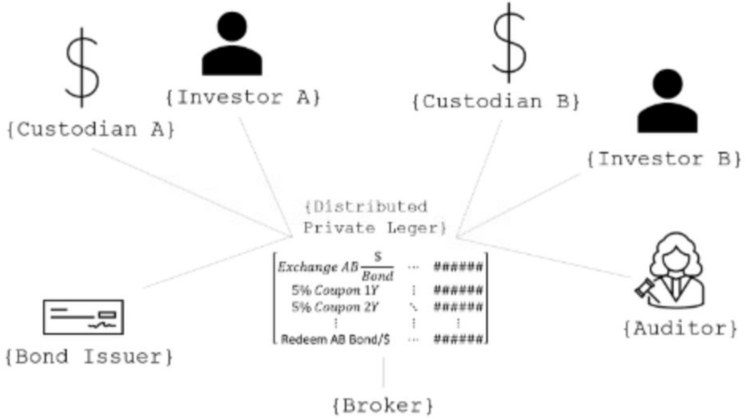
Investors also supplies Proof for ratio cash/bond doesn't exceeds capital risk.

Coupon Payment



Asset	Custodian	Bond Issuer	Broker	Investor M	Investor N
0	$g^{0\$}h^{r5a}, pk^{r5a}$	$g^{-300\$}h^{r5b}, pk^{r5b}$	$g^{0\$}h^{r5c}, pk^{r5c}$	$g^{100\$}h^{r5d}, pk^{r5d}$	$g^{200\$}h^{r5e}, pk^{r5e}$
1	$g^{0X}h^{r6a}, pk^{r6a}$	$g^{0X}h^{r6b}, pk^{r6b}$	$g^{0X}h^{r6c}, pk^{r6c}$	$g^{0}h^{r6d}, pk^{r6d}$	$g^{0X}h^{r6e}, pk^{r6e}$

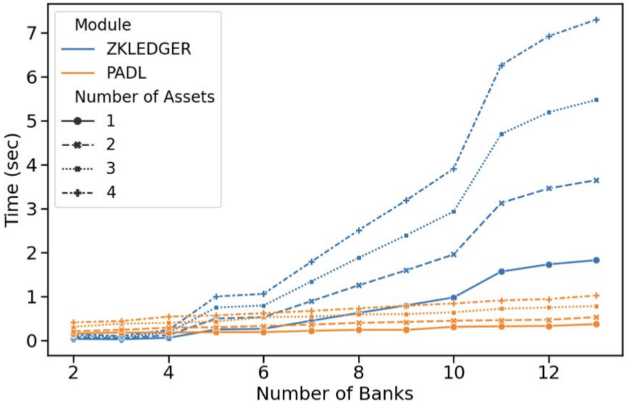
Broker also supplies rate proof between cash and bond tx to proof 10% coupon rate.

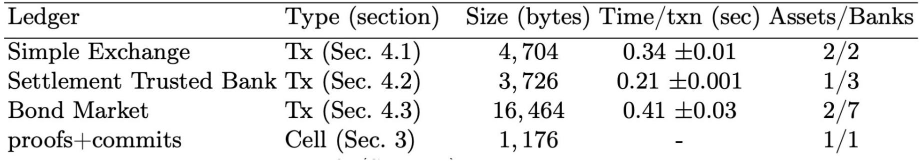


PADL Library

ZKProofs and primitives of BN254, Secp256 in Rust (bulletproofs, ZenGo-X), Interfaced with Python, and deployable as smart-contracts with on-chain verification (including bulletproof verification on-chain).

https://github.com/jpmorganchase/PADL





Ledger	Type (section)	Size (bytes)	Time/txn (sec)	Assets/Banks
Simple Exchange	Tx (Sec. 4.1)	4,704	0.34 ±0.01	2/2
Settlement Trusted Bank	Tx (Sec. 4.2)	3,726	0.21 ±0.001	1/3
Bond Market	Tx (Sec. 4.3)	16,464	0.41 ±0.03	2/7
proofs+commits	Cell (Sec. 3)	1,176	-	1/1

Thanks For your Listening!