zkLedger

Privacy-preserving auditing for distributed ledgers

Neha Narula

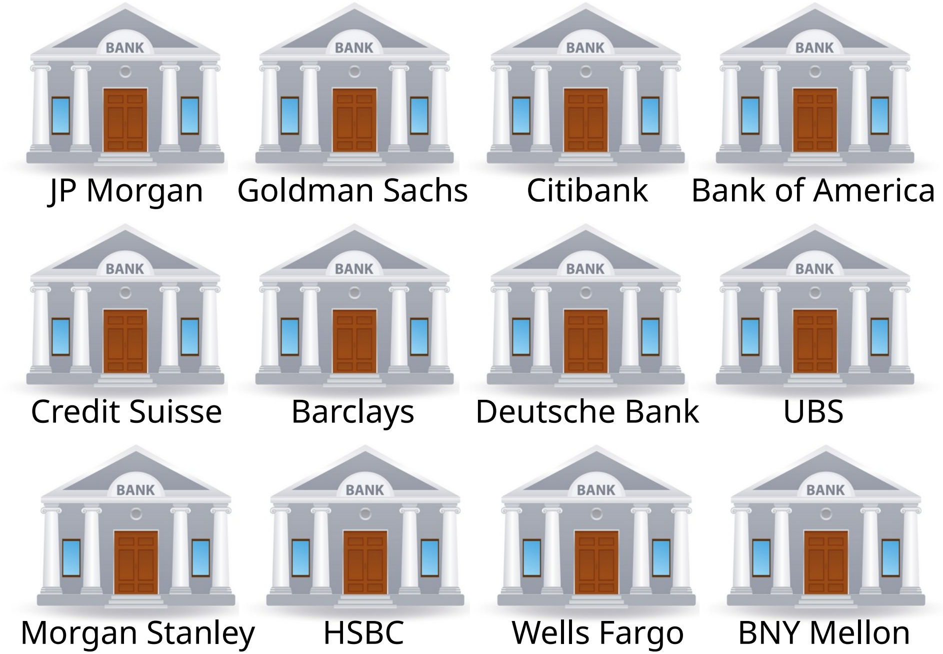
Willy Vasquez

Madars Virza

mit media lab

digital currency initiative

Structure of the financial system



- Dozens of large investment banks
- Trading:
  - Securities
  - Currencies
  - Commodities
  - Derivatives
- 40% unregulated
- Trillions of dollars
- Tens of trades/minute

Financial Investments Regulatory Authority on OTC markets

A ledger records financial transactions

ID	Asset	From	To	Amount	
90	$	Citibank	Goldman Sachs	1,000,000	sig
91	€	JP Morgan	UBS	200,000	sig
92	€	JP Morgan	Barclays	3,000,000	sig

Citibank

JP Morgan

Barclays

Can verify important financial invariants

ID	Asset	From	To	Amount	
90	$	Citibank	Goldman Sachs	1,000,000	sig
91	€	JP Morgan	UBS	200,000	sig
92	€	JP Morgan	Barclays	3,000,000	sig

Verify

✓ Consent to transfer

✓ Has assets to transfer

✓ Assets neither created nor destroyed

Examining ledger

Banks care about privacy

Trades reveal sensitive strategy information

Verifying invariants are maintained with privacy

ID	Asset	From	To	Amount	
90	$	Citibank	Goldman Sachs	1,000,000	sig
91	€	JP Morgan	UBS	200,000	sig
92	€	JP Morgan	Barclays	3,000,000	sig

Verify

Consent to transfer

Has assets to transfer

Assets neither created nor destroyed

Verifying invariants are maintained with privacy

ID	Asset	From, To, Amount
90	$	▨
91	€	▨
92	€	▨

Zerocash (zk-SNARKs) [S&P 2014]

Solidus (PVORM) [CCS 2017]

Verify

✓ Consent to transfer

✓ Has assets to transfer

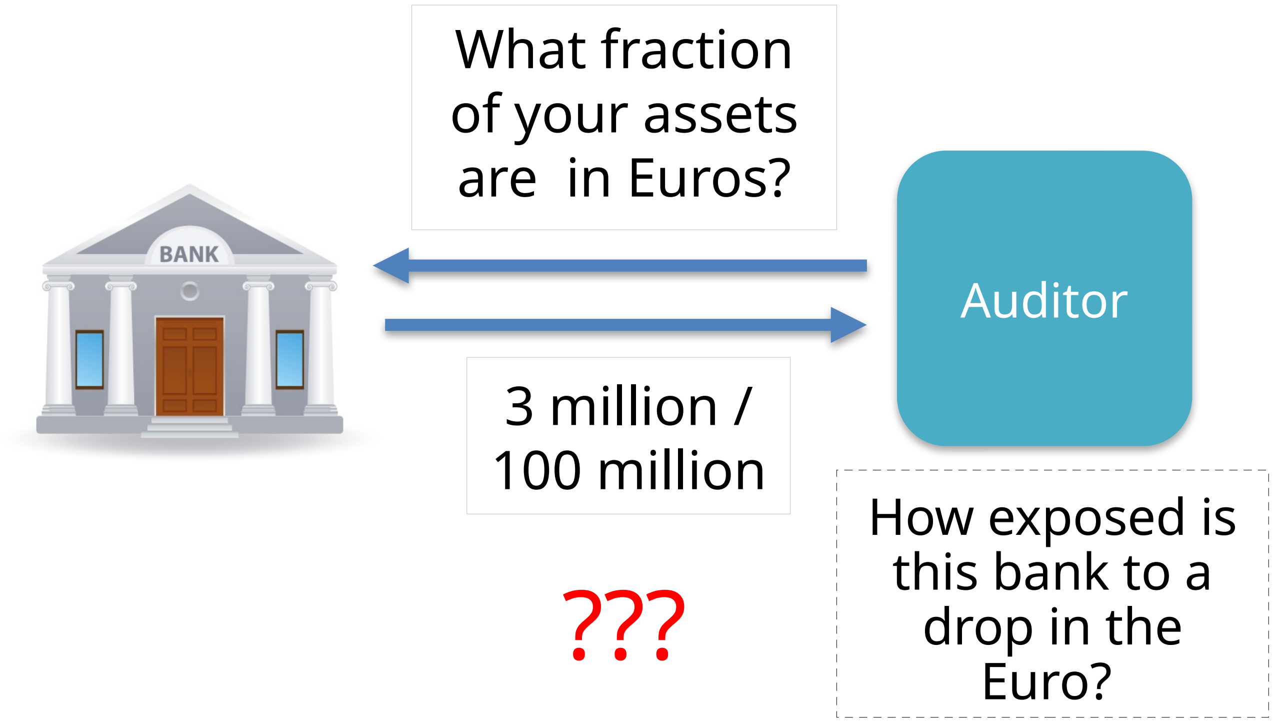
✓ Assets neither created nor destroyed

Problem

Regulators need insight into markets to maintain financial stability and protect investors

- Leverage
- Exposure
- Overall market concentration

How to confidently audit banks to determine risk?



What fraction of your assets are in Euros?

Auditor

3 million / 100 million

???

How exposed is this bank to a drop in the Euro?

zkLedger

A private, auditable transaction ledger

- **Privacy:** Hides transacting banks and amounts
- **Integrity with public verification:** *Everyone* can verify transactions are well-formed
- **Auditing:** Compute provably-correct linear functions over transactions

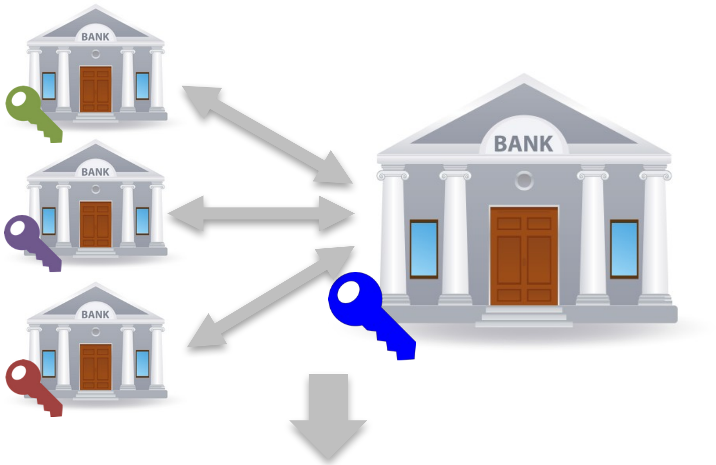
Outline

- System model
- zkLedger design
  - Hiding commitments
  - Ledger table format
  - Zero-knowledge proofs
- Evaluation

Outline

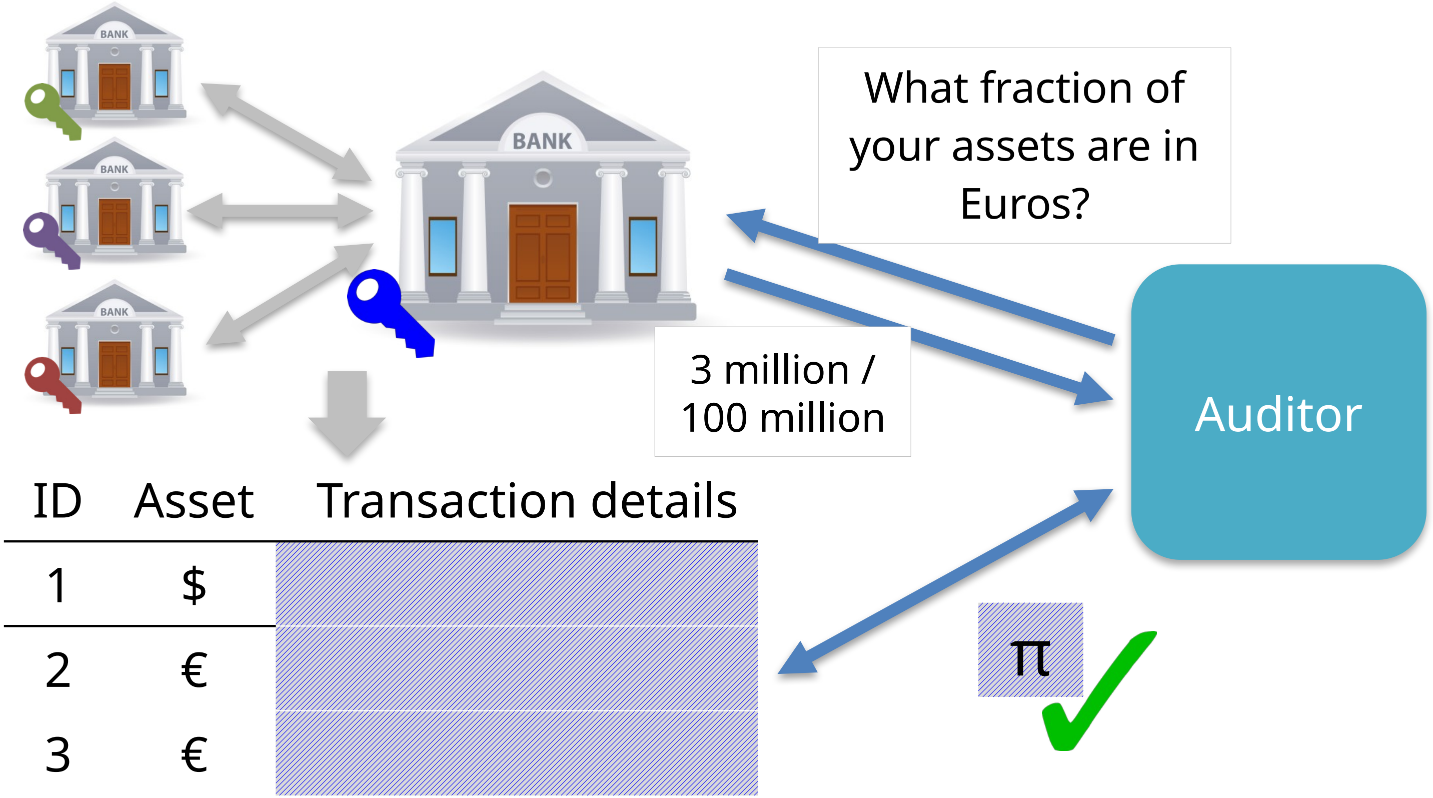
- **System model**
- zkLedger design
  - Hiding commitments
  - Ledger table format
  - Zero-knowledge proofs
- Evaluation

zkLedger system model



ID	Asset	Transaction details
1	$	▨
2	€	▨
3	€	▨

An auditor can obtain correct answers on ledger contents



What fraction of your assets are in Euros?

3 million / 100 million

Auditor

π ✓

ID	Asset	Transaction details
1	$	▨
2	€	▨
3	€	▨

Measurements zkLedger supports

- Ratios and percentages of holdings
- Sums, averages, variance, skew
- Outliers
- Approximations and orders of magnitude
- Changes over time
- Well-known financial risk measurements (Herfindahl-Hirschmann index)

Small amounts of well-defined leakage

Security goals

Privacy

- The auditor and non-involved parties **cannot see** transaction participants or amounts

Completeness

- Banks **cannot lie** to the auditor or **omit** transactions

Integrity

- Banks **cannot violate** financial invariants
  - Honest banks can always **convince** the auditor of a correct answer

Progress

- A malicious bank **cannot block** other banks from transacting

Threat model

Banks might attempt to steal or hide assets, manipulate balances, or lie to the auditor

Banks can arbitrarily collude

Banks or the auditor might try to learn transaction contents

Out of scope:

- A ledger that omits transactions or is unavailable
- An adversary watching network traffic
- Banks leaking their own transactions

Outline

- System model
- **zkLedger design**
    - Hiding commitments
    - Ledger table format
    - Zero-knowledge proofs
- Evaluation

Example public transaction ledger

ID	Asset	From	To	Amount
1	€	Depositor	Goldman Sachs	30,000,000
2	€	Goldman Sachs	JP Morgan	10,000,000
3	€	JP Morgan	Barclays	1,000,000
4	€	JP Morgan	Barclays	2,000,000

Depositor injects assets to the ledger

ID	Asset	From	To	Amount
1	€	Depositor	Goldman Sachs	30,000,000
2	€	Goldman Sachs	JP Morgan	10,000,000
3	€	JP Morgan	Barclays	1,000,000
4	€	JP Morgan	Barclays	2,000,000

Goals: auditing + privacy

ID	Asset	From	To	Amount
1	€	Depositor	Goldman Sachs	30,000,000
2	€	Goldman Sachs	JP Morgan	10,000,000
3	€	JP Morgan	Barclays	1,000,000
4	€	JP Morgan	Barclays	2,000,000

Goals:

- Provably audit Barclays to find Euro holdings
- Hide participants, amounts, and transaction graph

Hide amounts with commitments

ID	Asset	From	To	Amount
1	€	Depositor	Goldman Sachs	30M
2	€	Goldman Sachs	JP Morgan	comm(10M) ×
3	€	JP Morgan	Barclays	comm(1M) ×
4	€	JP Morgan	Barclays	comm(2M)

= comm(13M)

Pedersen commitments

Bank creates comm(v) = $g^v h^r$

<u>Important properties</u>

- Binding
- Homomorphically combined
- Fast

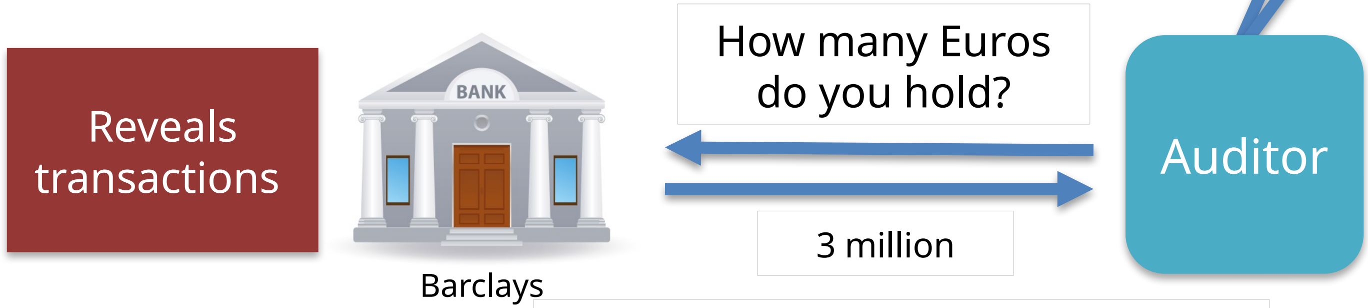
Can achieve all auditing functions with Pedersen Commitments! (see paper)

Hide participants with other techniques

ID	Asset	From	To	Amount
1	€	Depositor	Goldman Sachs	30M
2	€	Goldman Sachs	JP Morgan	`comm(10M)`
3	€	JP Morgan	Barclays	`comm(1M)`
4	€	JP Morgan	Barclays	`comm(2M)`

Strawman: audit by opening up combined commitments

ID	Asset	From	To	Amount
1	€	Depositor	Goldman Sachs	30M
2	€	Goldman Sachs	JP Morgan	comm(10M)
3	€	JP Morgan	Barclays	comm(1M)
4	€	JP Morgan	Barclays	comm(2M)



Reveals transactions

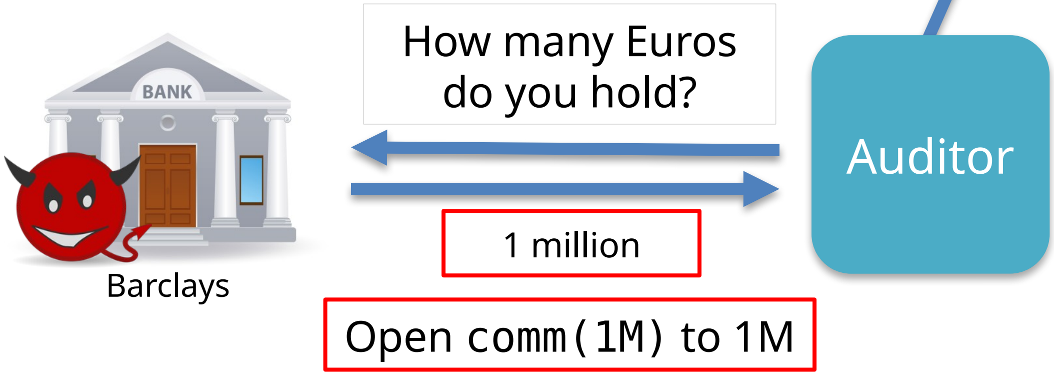
How many Euros do you hold?

3 million

Open comm(1M) × comm(2M) to 3M

A malicious bank could omit transactions

ID	Asset	From	To	Amount
1	€	Depositor	Goldman Sachs	30M
2	€	Goldman Sachs	JP Morgan	`comm(10M)`
3	€	JP Morgan	Barclays	`comm(1M)`
4	€	JP Morgan	Barclays	`comm(2M)`



How many Euros do you hold?

1 million

Open `comm(1M)` to 1M

A malicious bank could omit transactions

ID	Asset	From	To	Amount
1	€	Depositor	Goldman Sachs	30M
2	€	Goldman Sachs	JP Morgan	`comm(10M)`
3	€	JP Morgan	Barclays	`comm(1M)`
4	€	JP Morgan	Barclays	`comm(2M)`

zkLedger design: an entry for every bank in every transaction

ID	Asset	**Goldman Sachs**	**JP Morgan**	**Barclays**
1	€	Depositor, Goldman Sachs, 30M		
2	€	`comm(-10M)`	`comm(10M)`	**`comm(0)`**
3	€	**`comm(0)`**	`comm(-1M)`	`comm(1M)`
4	€	**`comm(0)`**	`comm(-2M)`	`comm(2M)`

Depositor transactions are public

Spender’s column commits to negative value, receiver’s positive value

For non-involved banks, entries commit to 0

Indistinguishable from commitments to non-zero values

Key insight: auditor audits *every* transaction

ID	Asset	Goldman Sachs	JP Morgan	Barclays
1	€	Depositor, Goldman Sachs, 30M		
2	€	comm(-10M)	comm(10M)	comm(0)
3	€	comm(0)	comm(-1M)	comm(1M)
4	€	comm(0)	comm(-2M)	comm(2M)

Barclays

How many Euros do you hold?

3 million

Auditor

Open comm(0) × comm(1M) × comm(2M) to 3M

A malicious bank can’t produce a proof for a different answer

ID	Asset	Goldman Sachs	JP Morgan	Barclays
1	€	Depositor, Goldman Sachs, 30M		
2	€	`comm(-10M)`	`comm(10M)`	`comm(0)`
3	€	`comm(0)`	`comm(-1M)`	`comm(1M)`
4	€	`comm(0)`	`comm(-2M)`	`comm(2M)`

Barclays

How many Euros do you hold?

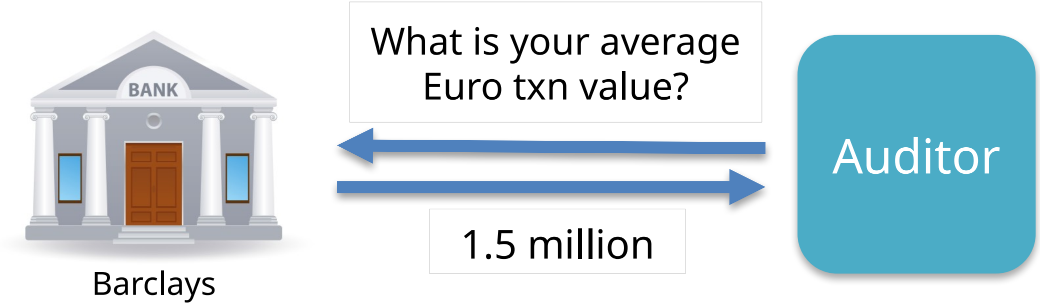
~~milli~~

~~Open co M) to 1M~~

Auditor

Computing averages

ID	Asset	Goldman Sachs	JP Morgan	Barclays
1	€	Depositor, Goldman Sachs, 30M		
2	€	`comm(-10M)`	`comm(10M)`	`comm(0)`
3	€	`comm(0)`	`comm(-1M)`	`comm(1M)`
4	€	`comm(0)`	`comm(-2M)`	`comm(2M)`



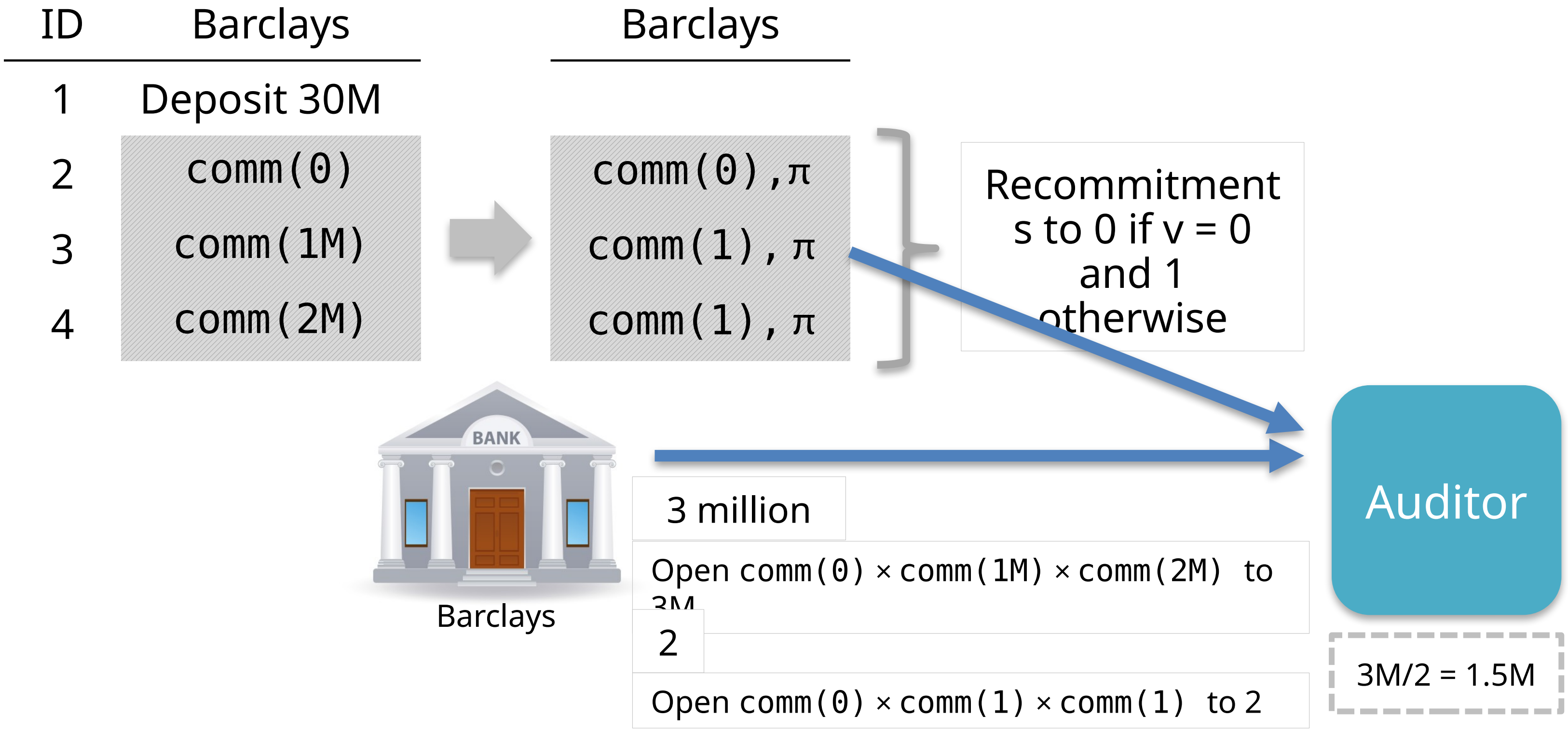
Barclays

What is your average Euro txn value?

1.5 million

Auditor

Recommitments



ID	Barclays
1	Deposit 30M
2	`comm(0)`
3	`comm(1M)`
4	`comm(2M)`

Barclays
`comm(0)`,π
`comm(1)`, π
`comm(1)`, π

Recommitments to 0 if v = 0 and 1 otherwise

3 million

Open `comm(0)` × `comm(1M)` × `comm(2M)` to 3M

2

Open `comm(0)` × `comm(1)` × `comm(1)` to 2

3M/2 = 1.5M

Security goals

Privacy	✓	The auditor and non-involved parties **cannot see** transaction participants, amounts, or transaction graph
Completeness	✓	Banks **cannot lie** to the auditor or **omit** transactions
Integrity		Banks **cannot violate** financial invariants – Honest banks can always **convince** the auditor of a correct answer
Progress		A malicious bank **cannot block** other banks from transacting

Non-interactive zero-knowledge proofs (NIZKs)

- Short, binary strings
- True statements have proofs
- False statements only have proofs with negligible probability
- Proofs don’t reveal why they are true

Achieving integrity and progress using NIZKs

- Transaction validity
  - Consent to transfer
  - Have assets to transfer
  - Assets neither create nor destroyed
- Honest banks can make progress
  - Non-interactive

Consent NIZK

Assets NIZK

Balance NIZK

Consistency NIZK

See paper for details

Proofs of transaction correctness

- **Consent** Knowledge of secret key *sk* spending
- **Assets** If spending, have assets to spend. Adding entry *i* for transaction *m*, new commitment $\mathtt{comm}_{aux}$:

  $\mathtt{comm}_{aux}$ commits to Spending: $\sum_{i=1}^{n} v_i$ OR Not spending: v_i

  and a proof that the value in $\mathtt{comm}_{aux}$ is in range

Borromean ring signatures, Confidential Assets

- **Balance** No funds created or destroyed (one per transaction):

  Choose r's such that $\sum_{i=1}^{n} r_i$ is 0

Outline

- System model
- zkLedger design
  - Hiding commitments
  - Ledger table format
  - Zero-knowledge proofs
- **Evaluation**

Implementation

- zkLedger written in Go
- Elliptic curve library: btcec, secp256k1
- Range proofs to prevent overflow: Confidential Assets [FC 2017]
- ~4000 loc

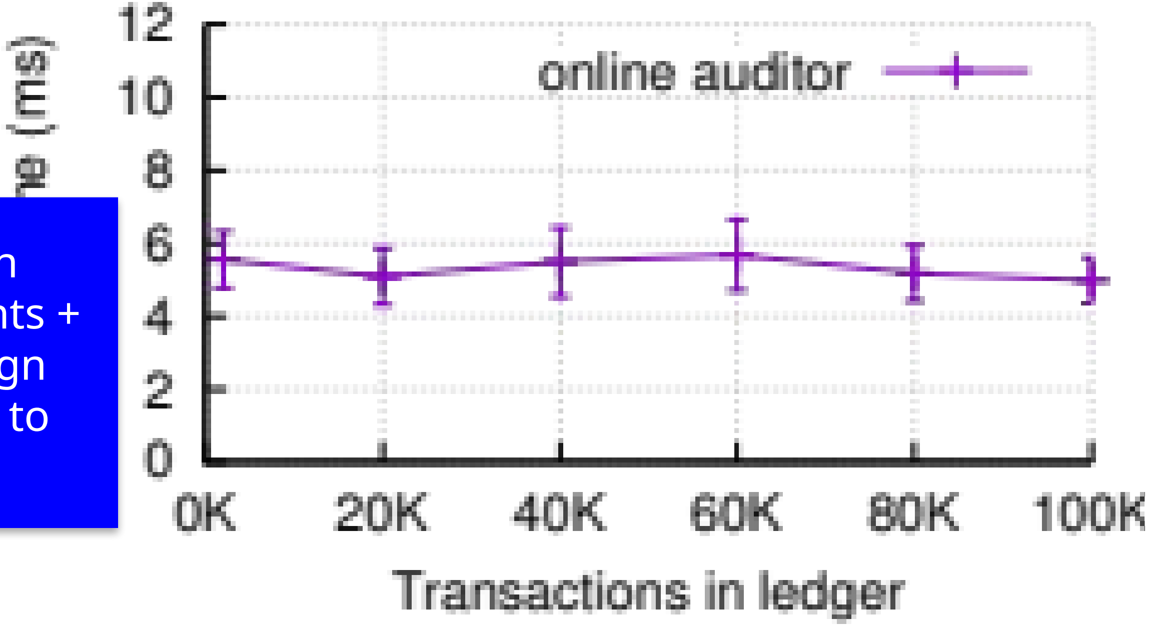
Evaluation

- How fast is auditing?
- How does zkLedger scale with the number of banks?

Experiments on 12 4 core Intel Xeon 2.5Ghz VMs, 24 GB RAM

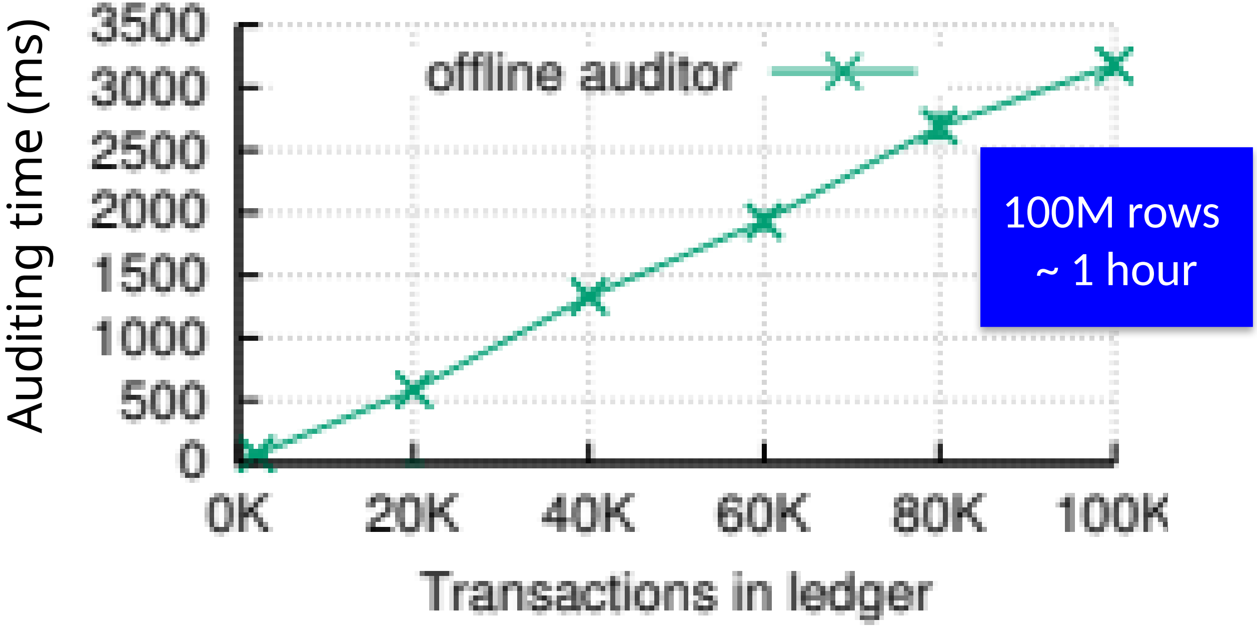
Simple auditing is fast and independent of ledger size

Pedersen commitments + table design amenable to caching



Auditing 4 banks measuring market concentration

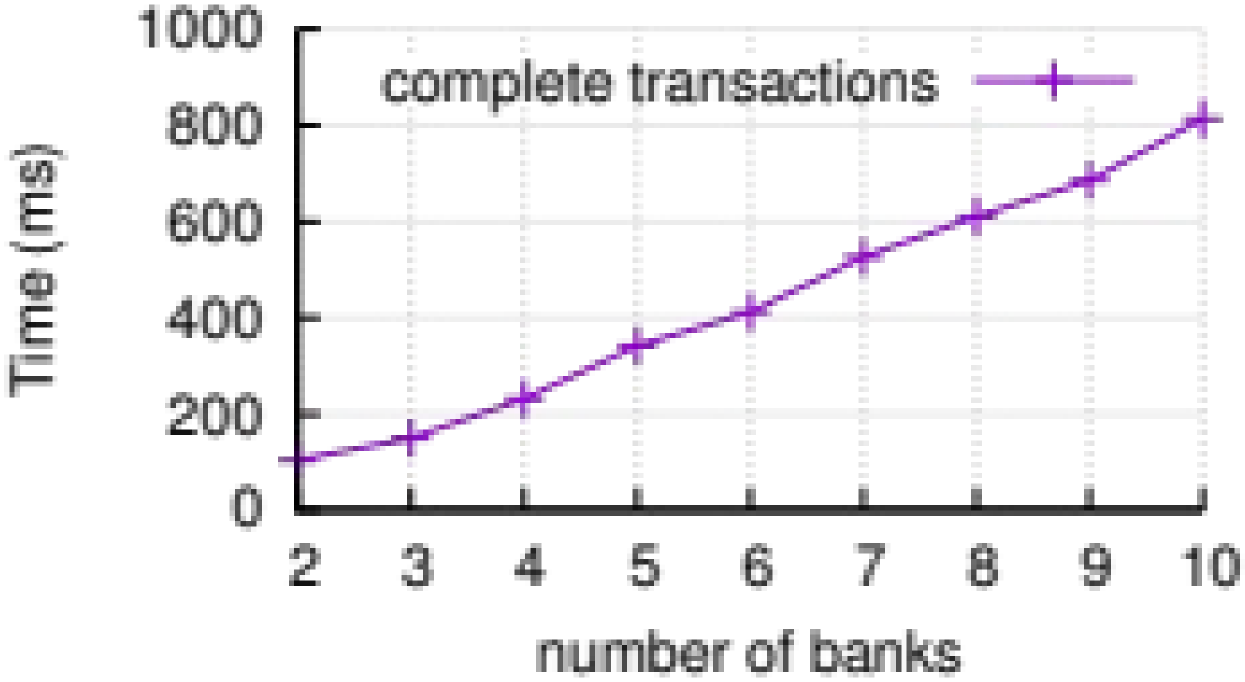
More complex forms of auditing are linear in size of ledger



100M rows
~ 1 hour

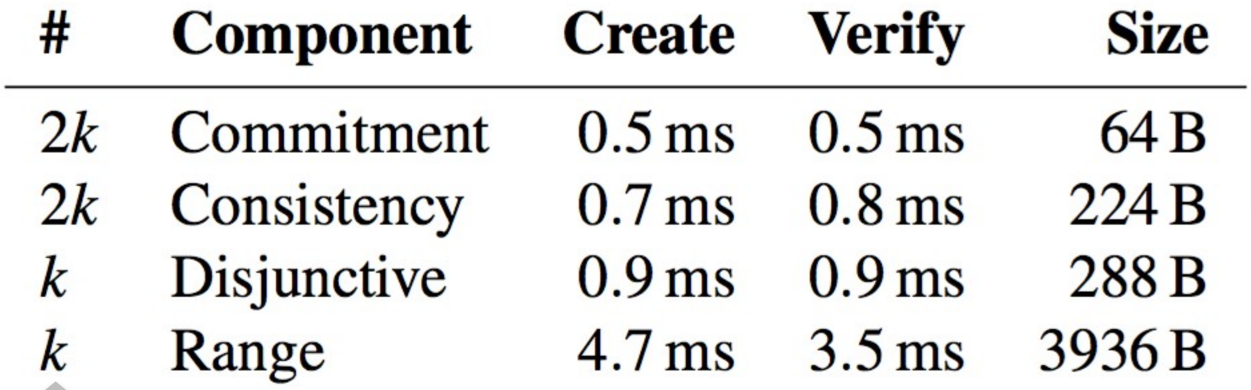
Auditing 4 banks measuring market concentration

Processing transactions scales linearly



One bank creating transactions. Includes ledger, auditor, and other banks verifying

Proof component sizes and times



#	Component	Create	Verify	Size
$2k$	Commitment	0.5 ms	0.5 ms	64 B
$2k$	Consistency	0.7 ms	0.8 ms	224 B
k	Disjunctive	0.9 ms	0.9 ms	288 B
k	Range	4.7 ms	3.5 ms	3936 B

one elliptic curve point

2X slower
4.5X larger

Number in transaction for *k* participants

Cost in a transaction per bank

- Entry size: **4.5KB**
- Creating an entry: **8ms**
- Verifying an entry: **7ms**

× # banks

Highly parallelizable

Significant opportunities for compression and speedup

Related Work

No private auditing

- Confidential Assets [FC 2017]
- Zerocash [S&P 2014]

Cannot guarantee completeness

- Privacy-preserving methods for sharing financial risk exposures [2011]
- Provisions [CCS 2015]

Our techniques might apply

Solidus [CCS 2017]

Accountable privacy for decentralized anonymous payments [FC 201

Design for policy enforcement, not auditing

Future Work

- Other applications (public bulletin board)
- Beyond Pedersen commitments
- Optimize implementation (Bulletproofs)

Conclusion

zkLedger provides practical privacy and complete auditing on transaction ledgers

zkledger.org