Enhancing Cryptocurrency Blocklisting: A Secure, Trustless, and Effective Realization

Yuefeng Du, Anxin Zhou, Cong Wang

City University of Hong Kong

Prevalent Cryptocurrency Crimes

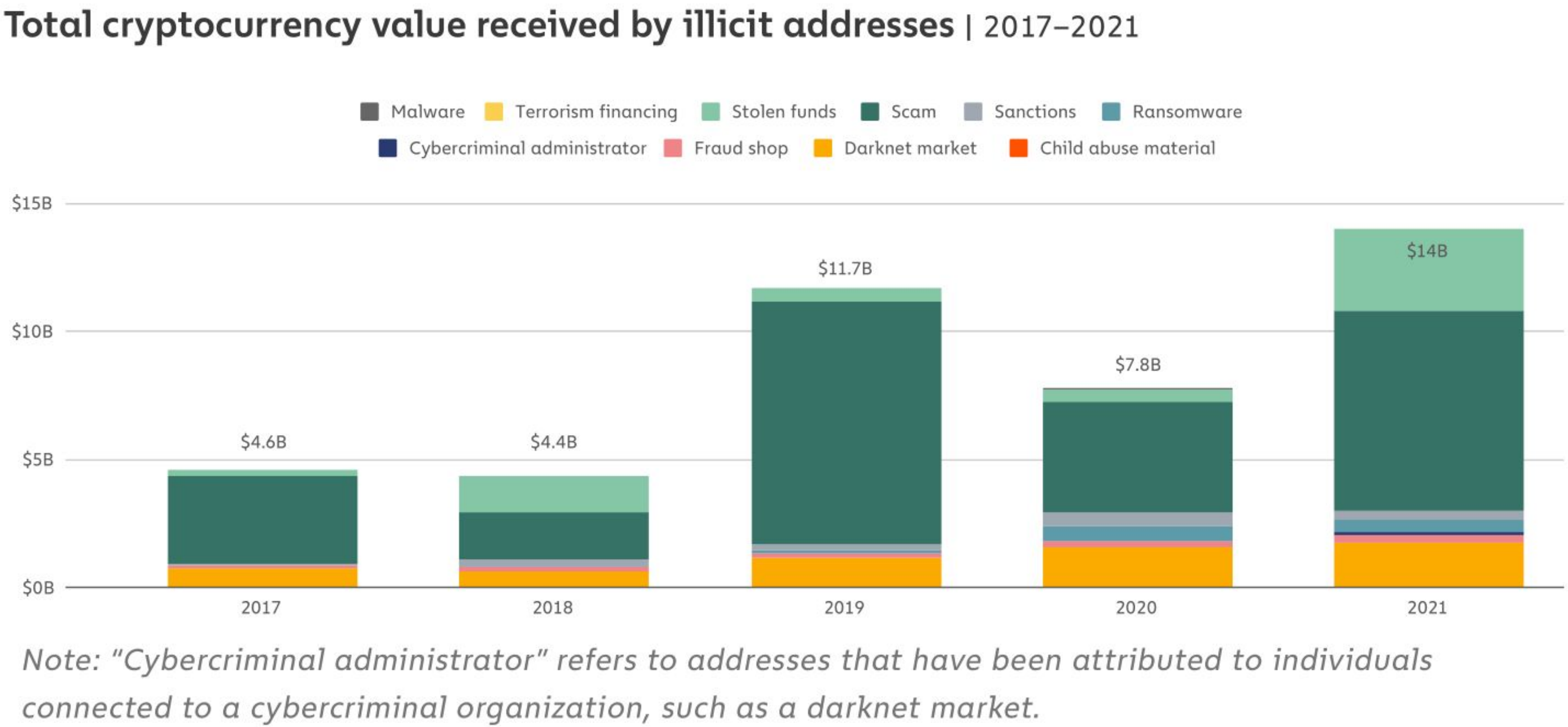
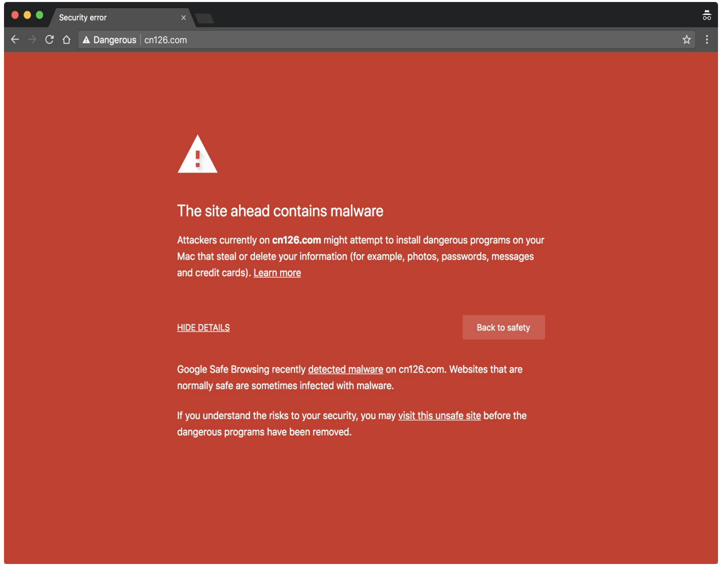
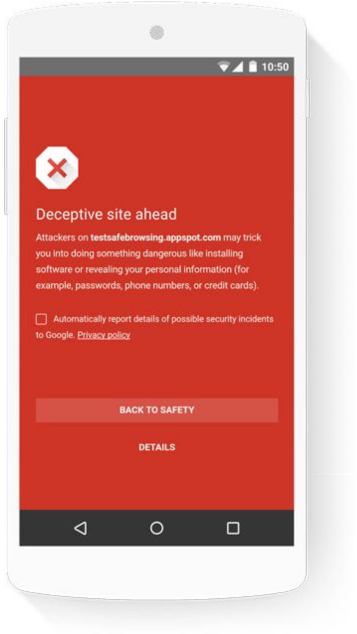


Figure taken from The 2022 Crypto Crime Report, Chainalysis

Safe Browsing: URL Blocklisting





Block **malware** or **phishing**

- **Chrome, Firefox, Safari …**
- **4 billions** devices

Safe Transaction: Cryptocurrency Blocklisting

Client

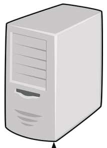
Blocklist Service

Blockchain account/address

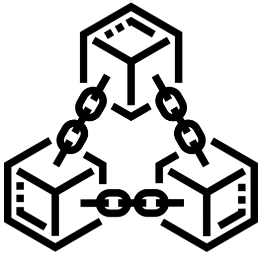
Suspicious? →

← *Yes/No/Info*

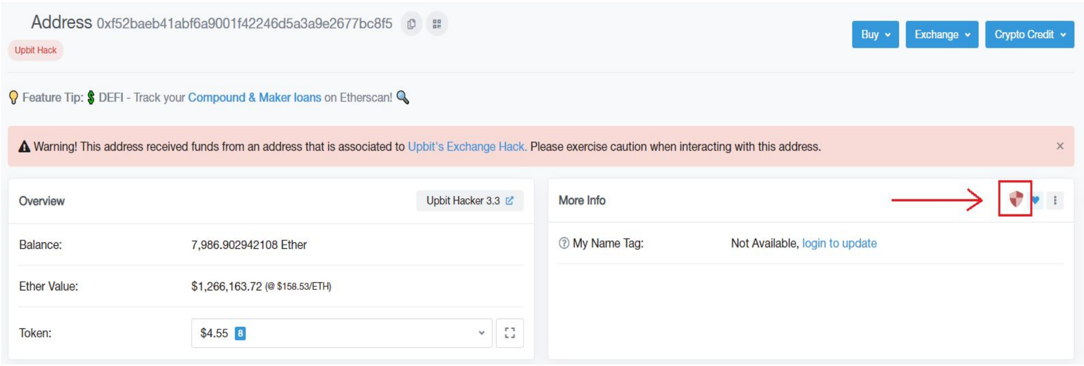
List of unsafe addresses



Data mining

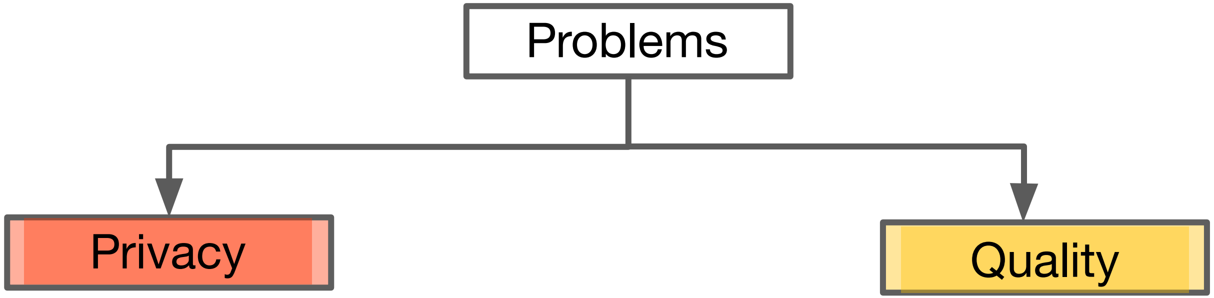


Blockchain Records



ETHProtect *warns* ***Etherscan*** *users of phishing, scams, and hacks.*

Problems with Cryptocurrency Blocklisting



Problem #1: Privacy

- Blocklist service providers see sensitive user queries in the clear
  - Facilitate data collection & user profiling
  - Leak user intention (e.g., frontrunning attacks, forcing up tx fee, DoS)

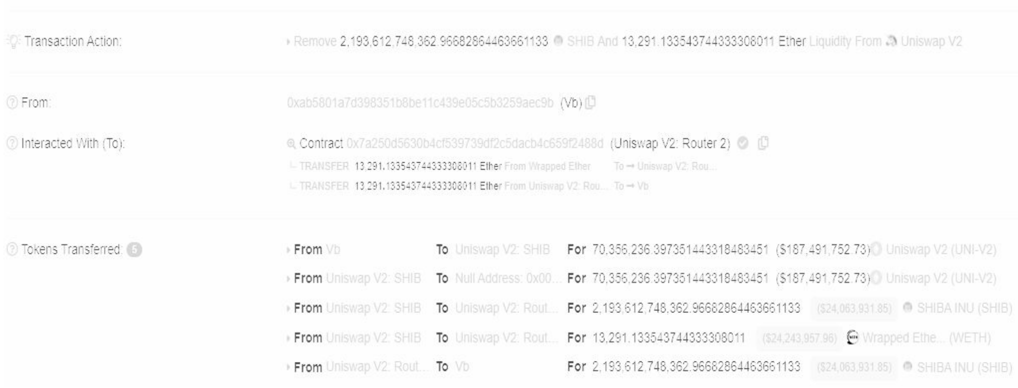
Problem #1: Privacy

- Blocklist service providers see sensitive user queries in the clear
  - Facilitate data collection & user profiling
  - Leak user intention (e.g., frontrunning attacks, forcing up tx fee, DoS)
- Blocklists are proprietary assets by the service providers
  - Should avoid disclosure to unauthorized parties

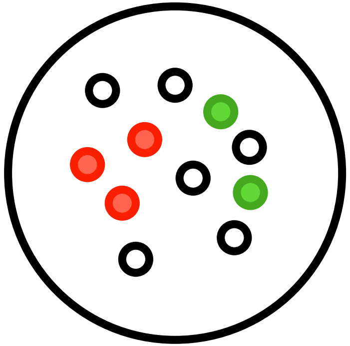
Problem #1: Privacy

- Blocklist servers see sensitive user queries in the clear
  - Facilitate data collection user profiling
  - Lea

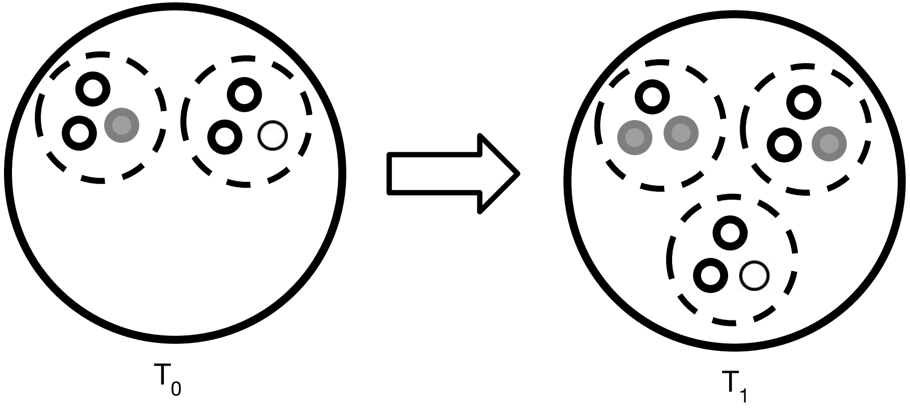
Goal: Enable privacy-preserving blocklist queries for cryptocurrency addresses



Problem #2: Quality



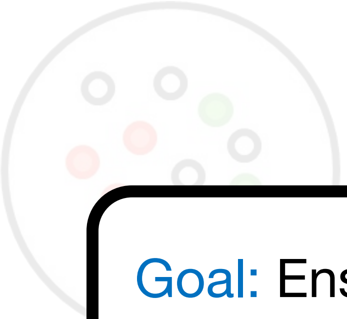
- Real threats unrecognized unintendedly /deliberately
- Safe addresses mis-identified as dangerous ones.



- Blocklists can be
  - Diverse
  - Inaccurate [1]
  - Evolving [2]

[1] BLAG: Improving the Accuracy of Blacklists, Ramanathan et al., In Proc. of NDSS, 2020.
[2] Blocklist babel: On the transparency and dynamics of open source blocklisting, Feal et al., IEEE Trans. Netw. Serv. Manag. 18(2), 2021

Problem #2: Quality



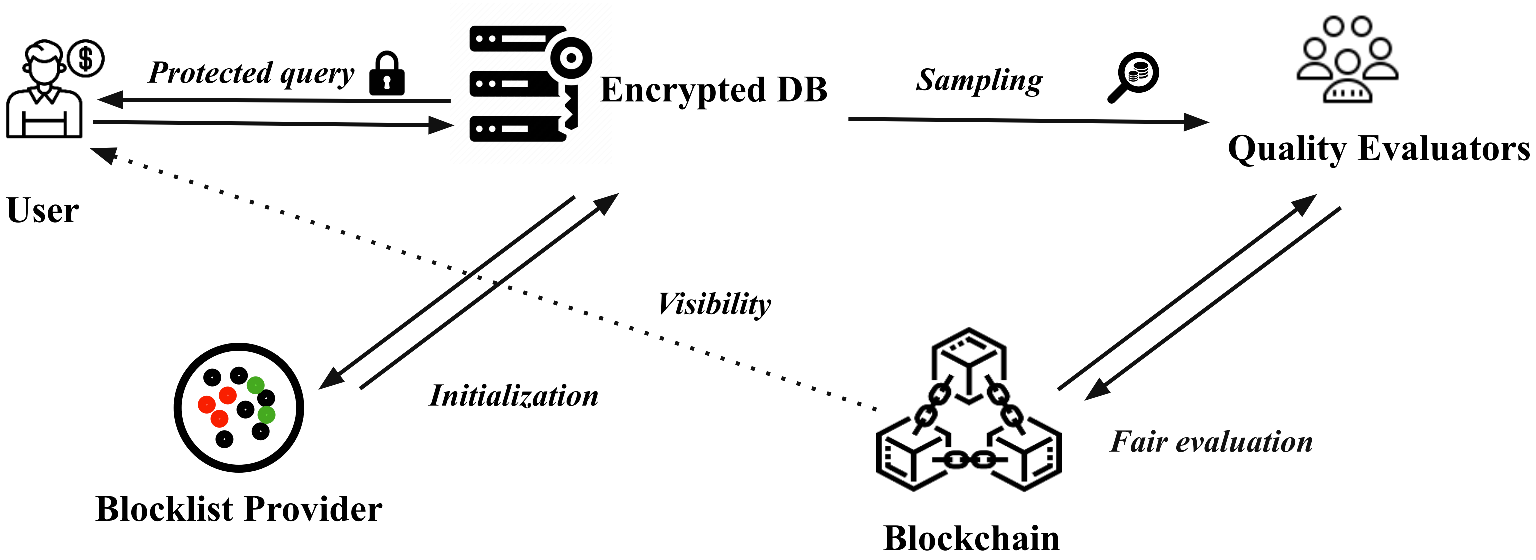
Real threats unrecognized unintendedly /deliberately

- Blocklists can be
  - Inconsistent

Goal: Ensure high-quality blocklist services with a proper quality evaluation mechanism

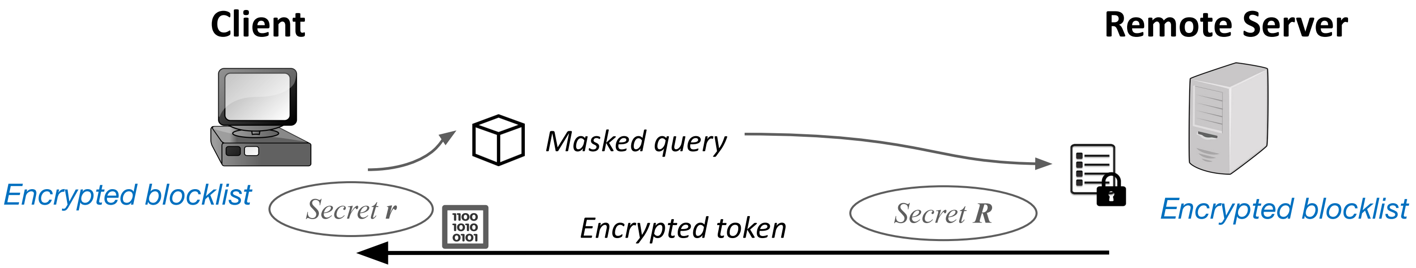
1. BLAG: Improving the Accuracy of Blacklists, Ramanathan et al., In Proc. of NDSS, 2020.
2. Blocklist babel: On the transparency and dynamics of open source blocklisting, Feal et al., IEEE Trans. Netw. Serv. Manag. 18(2), 2021

Our architecture



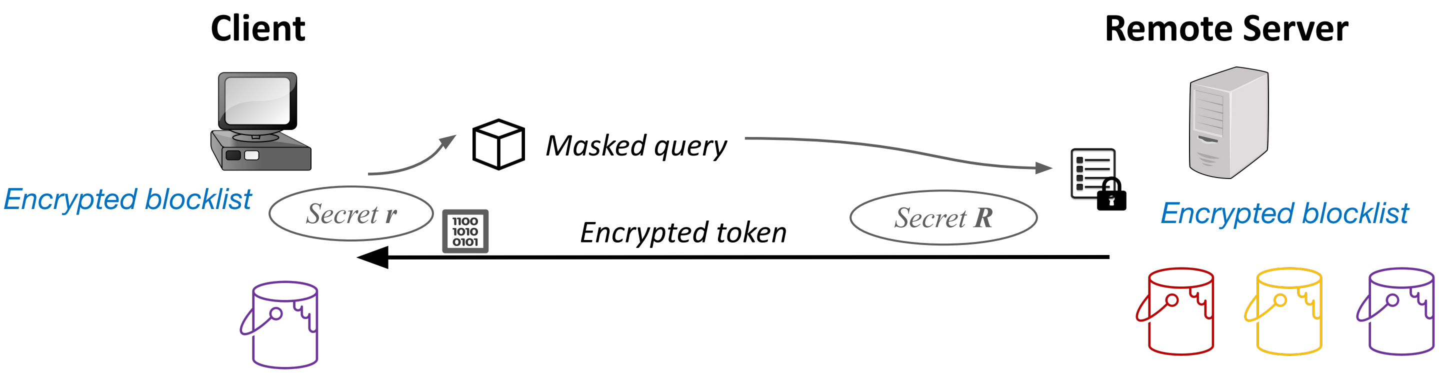
- Decoupling the curation and serving of blocklists
- Decentralized evaluation of blocklist quality

Addressing Problem #1: Private Query



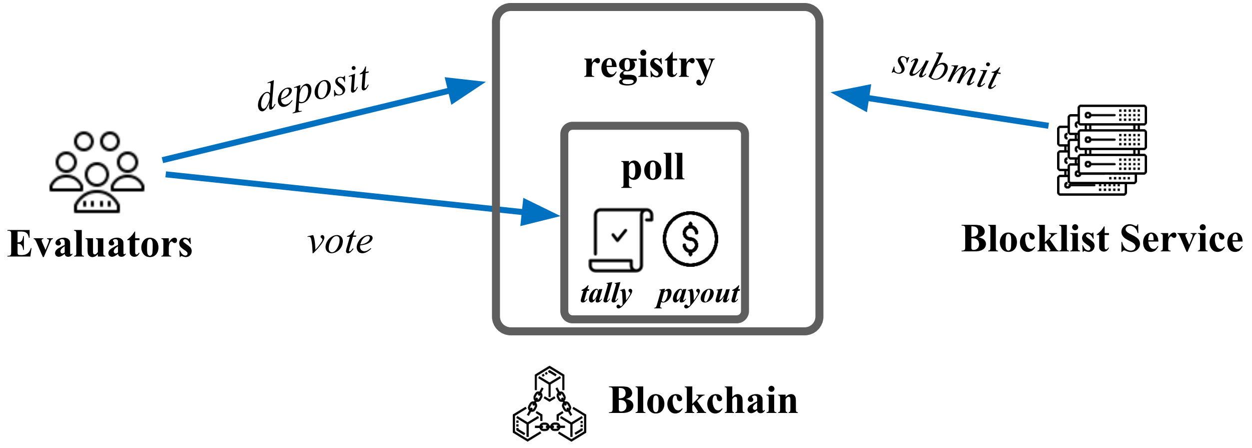
- Goal: same query complexity as the existing blocklist services
  - One round-trip per query, precluding the hefty crypto primitives like PIR
- We propose to store an encrypted (and searchable) blocklist at the client side
  - Client asks server for authorised search tokens

Addressing Problem #1: Private Query



- Goal: same query complexity as the existing blocklist services
  - One round-trip per query, precluding the hefty crypto primitives like PIR
- We propose to store an encrypted (and searchable) blocklist at the client side
  - Client asks server for authorised search tokens
- Further enhancement:
  - Use bucketization for large list; more friendly for fresh update

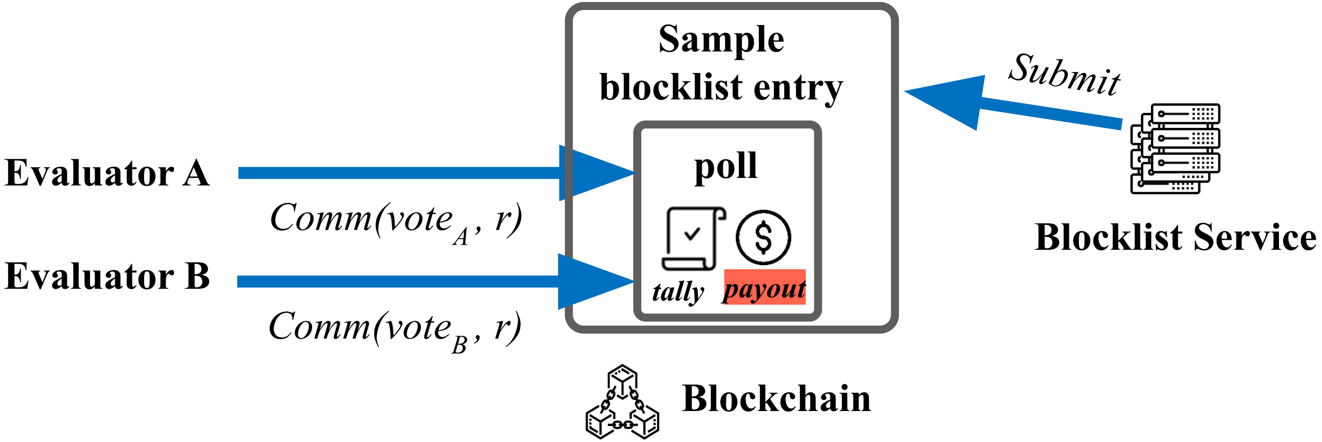
Addressing Problem #2: Decentralized Fair Blocklist Evaluation



- Inspired by Token Curated Registry (TCR) [1]
  - “Stake, and then vote for what you will use”
    - Vote weight proportional to stake
  - Assumption: economically rational participants

[1] Token curated registries - a game theoretic approach, Asgaonkar et. al., arXiv, 2018.

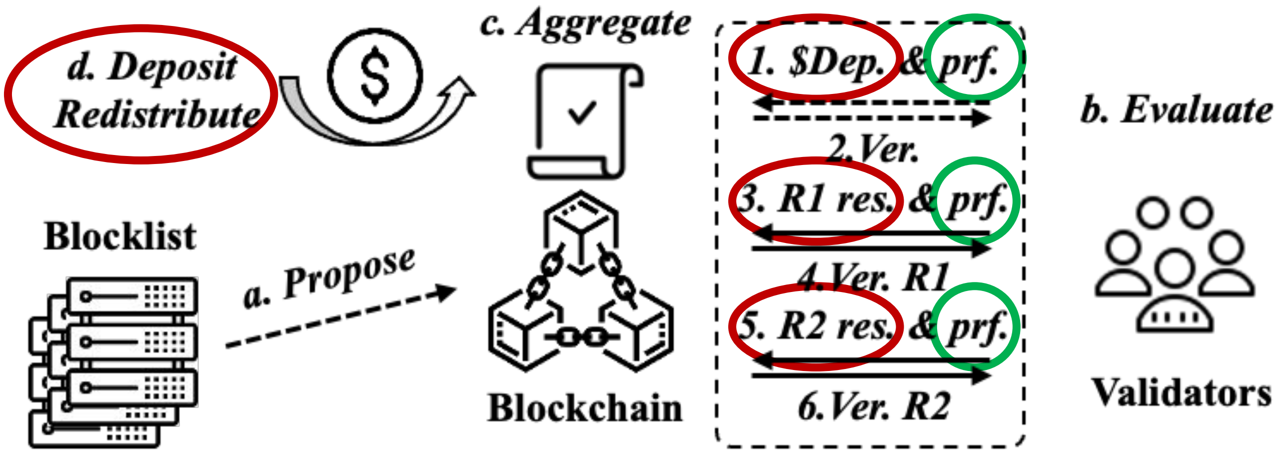
Challenge: Fair Evaluation



- The existing TCR practice is known to produce unfair results:
  - ***Biased outcome*** due to revealing order [1]
  - **Coercion** out of economic incentives [2]

[1] SHARVOT: secret SHARe-based Voting on the blockchain, Bartolucci et. al., Proc. of ICSE, 2018.
[2] Quadratic Voting in Blockchain Governance., Nicola Dimitri, *Information* 2022.

Resistance to Bias: Zero-Knowledge Evaluation



- Vote & stake confidentiality is a must
  - No disclosure of (intermediate) outcome, e.g., $deposit, Round 1 & Round 2 results
- Low-cost public verification
  - Detect any behavior deviation with minimized on-chain costs

Resistance to Coercion

Coercion-resistant voting:

- Well studied in cooperative game theory, e.g., Stackelberg competition
- Goal: maximize the costs of coercion to disincentivize attacks

- Real-world incidents:
  - e.g., Dark DAO, Curve War

[1] Algorand: Scaling Byzantine Agreements for Cryptocurrencies, Gilad et.al., in Proc of SOSP, 2017

Resistance to Coercion: Cryptographic Sortition

- Real-world incidents:
  - e.g., Dark DAO, Curve War

<u>Coercion-resistant voting:</u>

- Well studied in cooperative game theory, e.g., Stackelberg competition
- Goal: maximize the costs of coercion to disincentivize attacks
- We further extend the TCR design
  - Enlarge the candidate pool for evaluators
  - Secure random evaluator selection
    - Inspired by cryptographic sortition [1]
    - We adapt it to **encrypted values**

[1] Algorand: Scaling Byzantine Agreements for Cryptocurrencies, Gilad et.al., in Proc of SOSP, 2017

Evaluation Setup

- Real-world blocklists (over 240,000 entries)
- Ethereum for decentralized blocklist evaluation
- 10-20 evaluators

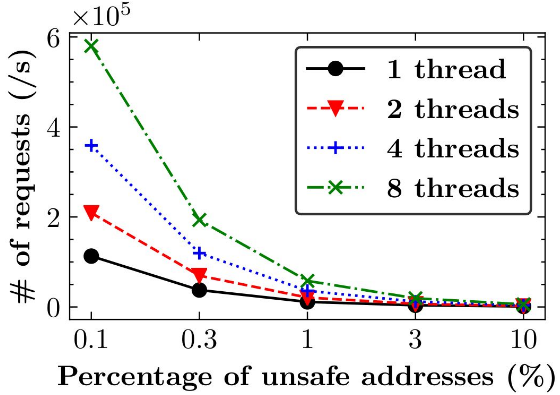
BitcoinAbuse.com
Certified Contributor

CryptoScamDB

Overhead of Private Query

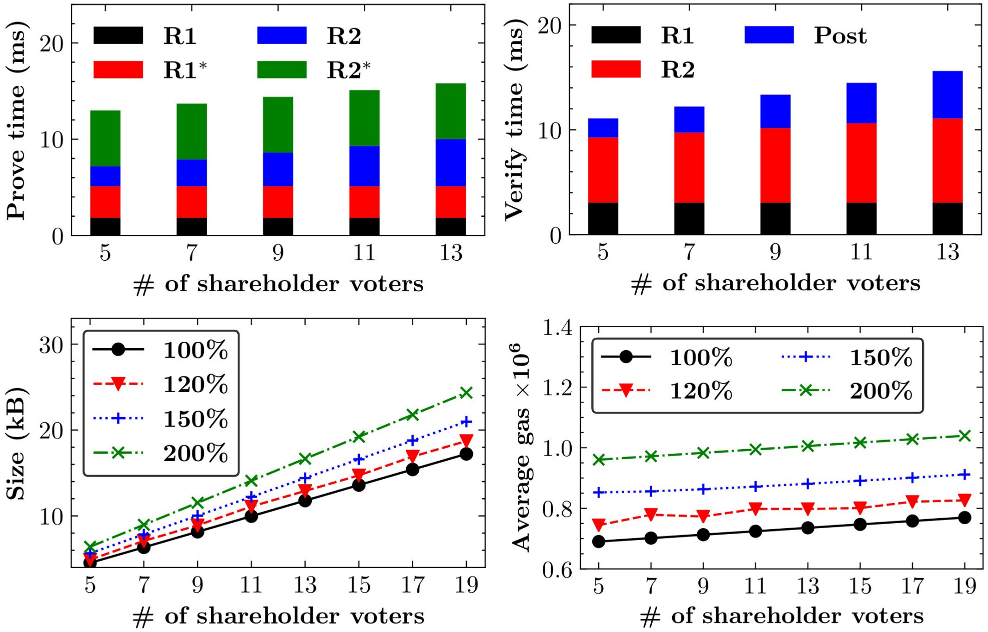
Prefix len.	Sec. wrt. k	Resp. size (kB)
16 bit	4	0.13
8 bit	977	30.53

Orac. H	Preprocess time[†]	Qry. time (ms)
Sha256	1.55 ± 0.02 *sec.*	$0.38 \pm 5\times10^{-3}$
Argon2*	1.27 ± 0.03 *hour*	147.29 ± 4.26



- Tunable security guarantees and communication overhead
- Practical initialization and query cost
- Throughput is affected by %unsafe addresses

Costs of Blocklist Evaluation



# of shareholder voters	5	7	9	11
Cost (USD)	16.02	16.28	16.54	16.80

Estimated on-chain cost undertaken by each evaluator

- Off-chain computation time
- On-chain costs
  - Proof storage
  - Ethereum gas for on-chain verification
- All linear to #evaluators

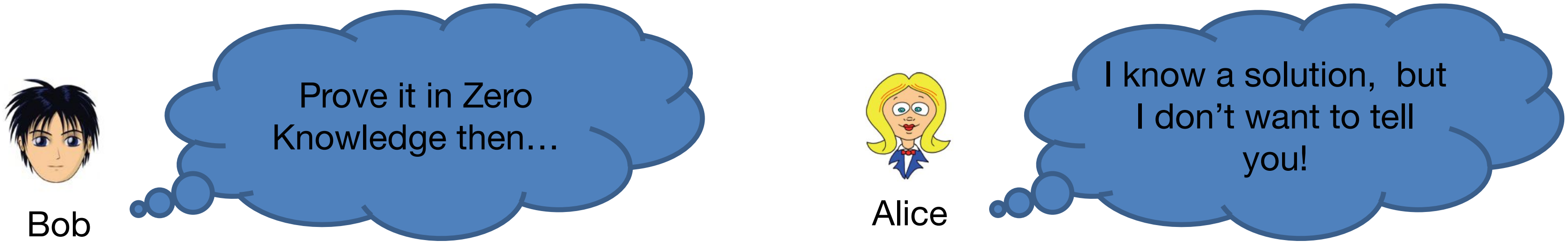
Concluding Remarks

- Two major problems in cryptocurrency blocklisting
  - No protection of sensitive queries
  - No (trustless) guarantee of blocklist quality
- Our solution raises the bar on privacy and security of this booming ecosystem

Concluding Remarks

- Two major problems in cryptocurrency blocklisting
  - No protection of sensitive queries
  - No (trustless) guarantee of blocklist quality
- Our solution raises the bar on privacy and security of this booming ecosystem

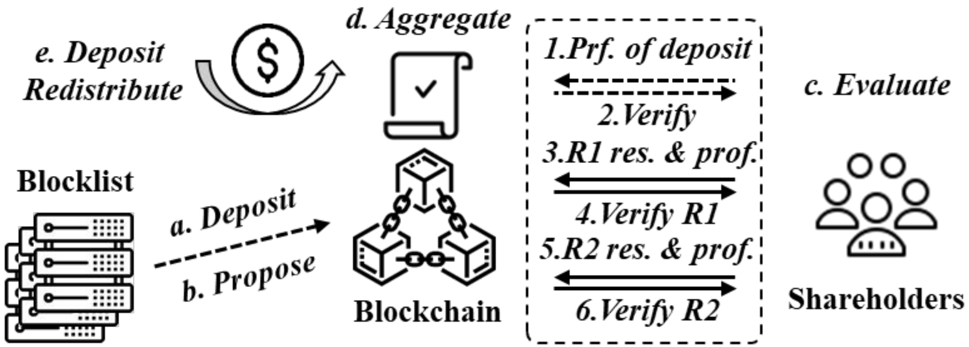
Commit-and-Prove Zero Knowledge Proof



Revealing nothing but the correctness of committed values

- *Partial vote confidentiality* ✓
- *Public verifiability* ✓

Construction Explained at a High Level



$$Q = \begin{cases} 1, & \sum_{i=0}^{n-1} \tau_i v_i > \frac{1}{2} \sum_{i=0}^{n-1} \tau_i \\ 0, & \sum_{i=0}^{n-1} \tau_i v_i \leq \frac{1}{2} \sum_{i=0}^{n-1} \tau_i \end{cases}$$

We consider a scenario where only 1-bit outcome is revealed lastly.

Q is revealed by tally and decommit Y

Deposit:

$r \leftarrow \$ F$

$C \leftarrow g^{amount} h^r)$

$\mathbf{prf_0} \leftarrow$ NIZK.Prove(R_{dep}, C, r)

R1:

$comm_0, comm_1 \leftarrow (g^r, g^{vote} h^r)$

$\mathbf{prf_1} \leftarrow$ NIZK.Prove(R_1, $comm_0$, r)

R2:

$Y \leftarrow \prod_{i=0}^{p-1} comm_{i,0} / \prod_{i=p+1}^{N-1} comm_{i,0}$

$comm_2 \leftarrow g^{vote} Y^r$

$\mathbf{prf_2} \leftarrow$ NIZK.Prove(R_2, $comm_1$, (vote, r))

Note *p* is the number of voters.