PriCLeSS

Privacy-Conscious Legally-Sound blockchain Storage

WIDE, Inria Rennes, IRISA, UR1
CIDRE, Inria Rennes, IRISA, UR1
GDD, LS2N, University of Nantes
IODE, University of Rennes 1

Overall Objectives

1-Leverage blockchains to provide legal and technical tools to automate and audit operations that access or exploit personal data.

2-provide providing legal and technical tools to addresses the challenges posed by distribution and cross-border exchanges

3-design an ecosystem of legal and technical tools that can support blockchain-based distributed storage applications, while satisfying privacy and legal requirements

Workpackages

WP 1 - Harnessing Blockchain Assets for Privacy Protection

- **Task 1.1:** Privacy Opportunity Analysis.
- **Task 1.2:** From Legal Requirements to Specification.
- **Task 1.3:** Smart Contracts for Legal Compliance.

WP 2 - Legal Compliance and Scalability through Distribution

- **Task 2.1:** Challenges of Distribution.
- **Task 2.2:** Combining legal specifications and distribution requirements.
- **Task 2.3:** Improving Blockchain storage.

WP 3 - An Ecosystem to address the Blockchain's shortcomings

- **Task 3.1:** Privacy versus technical characteristics of the Blockchain.
- **Task 3.2:** Enforcing privacy policies.
- **Task 3.3:** Composing data structures into a consistent ancillary ecosystem.

Tasks 1.1-1.3

Blockchain as a privacy risk

The blockchain itself

- Immutability
  - Violation of the GDPR (Article 5...)
  - Data disclosure, a privacy risk.
- Absence of rights management
  - how to determine the data controller?
  - how to enforce legal actions?

Applications

- Issues around the Internet of Things
  - Generalized and undifferentiated collection of personal data
  - Extraterritoriality makes it difficult to implement rights
- Issues arouns self-sovreign identity
  - New identity management
  - Risk of generalized surveillance

Blockchain as a privacy guarantee

Privacy-friendly storage on blockchain

- A variety of storage mechanisms
  - Geo-Controlled replication as a potential solution
  - Blockchain as hash storage only
- Establish reliable traceability by encryption
  - An asset for accurate data proofing
  - A new form of electronic archival system?

Decentralised trust: The ultimate goal for Privacy?

- Decentralised trust for privacy
  - Self sovereign identity
  - Towards generalized automation (Smart contract...)
- Evolution of services and trusted third parties
  - Joint use of signatures, stamps and electronic time stamps
  - Trusted services and third parties. The guarantors of a privacy-friendly blockchain

Requirements for GDPR-compliant data replication

GDPR's core requirements

- Right of access, rectification and deletion of data
- Regulation of data portability
- Right to object to fully automated data processing
- Material and territorial scope of the GDPR (Articles 2 and 3)
- Lawful, fair and transparent data processing (Article 5)

Blockchain properties

- **Transparency:** Participants can access all registered data
- **Replication and Decentralization:** Several copies of the blockchain exist simultaneously on different machines
- **Irreversibility:** Once data is entered, it cannot be changed or deleted.
- **Disintermediation:** Decisions recached through consensus without a centralized arbitrator

Shared Memory with Byzantine Actors

Advantages of a memory abstraction

Ease of use resulting from intuitive properties like **Linearizability**: i.e. an operation knows all updates applied by operations that ended before it started.

Challenges

- Memory with Byzantine actors has received little attention.
- We do not know exactly what it allows us to implement.

First Contribution

- We studied three abstractions and how to pass from one to the other.

read/write ⇄ (?) read/write-increment ⇄ (?) read/append

Read/Write register

- Read() will return the last value write in this register.
- Write(v) will write the value 'v' in this register.

Read/Write-Increment register

- Read() will return the last value written and the number of write calls on this register.
- Write(v) will write the value 'v' and increment the write counter of this register by 1.

Read/Append register

- Read() will return the history of all value written in this register.
- Append(v) will add the value 'v' at the end of the history of this register.

Our previous work

The comparaison of theses registers was already discussed in "Atomic Read/Write Memory in Signature-Free Byzantine Asynchronous Message-Passing Systems", were an implementation of Read/Write-Increment from Send/Receive is proposed with a resilience of $t < \frac{n}{3}$. This implies the existence of an implementation of Read/Write-Increment from Read/Write with a resilience of $t < \frac{n}{3}$.

read/write → ($t < \frac{n}{3}$) send/receive → ($t < \frac{n}{3}$) read/write-increment

Our contributions

We observe that

- the definition of Read/Write register is included in that of definition of Read/Write-increment.
- the definition of the Read/Write-increment register is included in the that of the Read/Append register.

So, we have wait-free algorithms for both transformations.

read/write ← ($t \leq n$) read/write-increment ← ($t \leq n$) read/append

From read/write to read/write increment

We proved that $t < \frac{n}{3}$ is necessary and sufficient to implement a read/write increment from read/write.

From Read/Write-increment to Read/Append

We proposed an implementation of a Read-append register from a Read/Write-increment register with a resilience of $t < \frac{n}{2}$. We also proved that this is optimal.

read/write ⇄ ($t < \frac{n}{3}$ / $t \leq n$) read/write-increment ⇄ ($t < \frac{n}{2}$ / $t \leq n$) read/append

Good-Case Latency of Early-Stopping Byzantine Reliable Broadcast

Good case latency

Number of rounds needed for the correct processes to brb-deliver a message brb-broadcast by a correct process

Early stopping

Number of rounds depends on the effective actual number f of Byzantine processes $f = n - c \leq t$ (e.g., min$(t+1, f+2)$) [1]

Strongly adaptive adversary

Is there a **deterministic** BRB algorithm whose good case latency is smaller that $t+1$?

The algorithm in a nutshell

- During a round: each process adds its signature to the message + signatures chains it receives, and sends them to each process
- Identification of a pattern in a set of messages and a predicate that allow the correct processes to brb-deliver a message m in at most max$(2, t+3-c)$ rounds in good cases (i.e., when the sender of m is correct)
- At round R, a process considers only valid message $+$ signatures chains (those have exactly R different signatures)

definitions and principles

Given a message m,

- certificate: set of signatures chains associated with m
- weight of a certificate: nb of processes whose signatures appear in the first two positions of the chains in the certificate, the corresponding processes are said to be backing m in the certificate
- Counting and propagating round-2 signatures is not enough as Byzantine process can hide part of a certificate from correct processes until round $t+1$

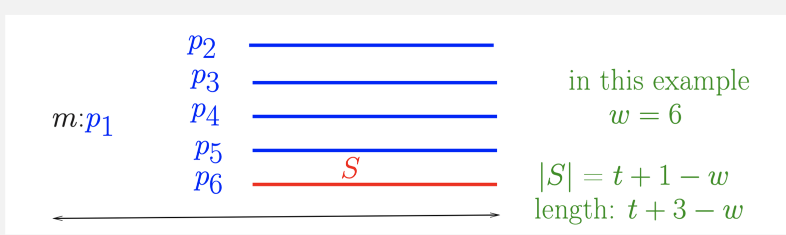
key concept: w-revealing chain

When present in a certificate, such a chain "differs sufficiently" from the w backing processes present in the certificate to allow for a safe brb-delivery

Example

Let $R = t+3-w$ be a round in which a correct process obtains a certificate whose weight w is such that there is a signature chain S starting at position 3 such that

$$\{\text{backing processes}\} \cap S = \emptyset$$



The signatures from position 3 to $t+3-w$ (S) correspond to $t+3-w-2 = t+1-w$ different processes. Added to the w backing processes p_1, ..., p_6, we obtain $(t+1-w)+w = t+1$ processes, hence we have a set including a correct process!

The case w=t+1

- When a message m has a certificate whose weight is $w = t+1$, all the correct processes received a chain containing m by round 2
- Conversely, if a process has not received a chain containing a message m' by round 2, it knows that a certificate of weight $t+1$ cannot exist for m'
- It follows that, if p_i observes a certificate of weight $t+1$ for m, and is not aware of another message $m' \neq m$ by round 2, it can safely brb-deliver m (even if the sender is Byzantine)
- rbr-delivery of m may occurs as early as round $R = 2$ (pattern depending)
- When $c \geq t+1$, rbr-delivery of m always occurs at round $R = 2$ (good case latency)

SplitChain: Resilient-Scalable Sharding

Scalabilty

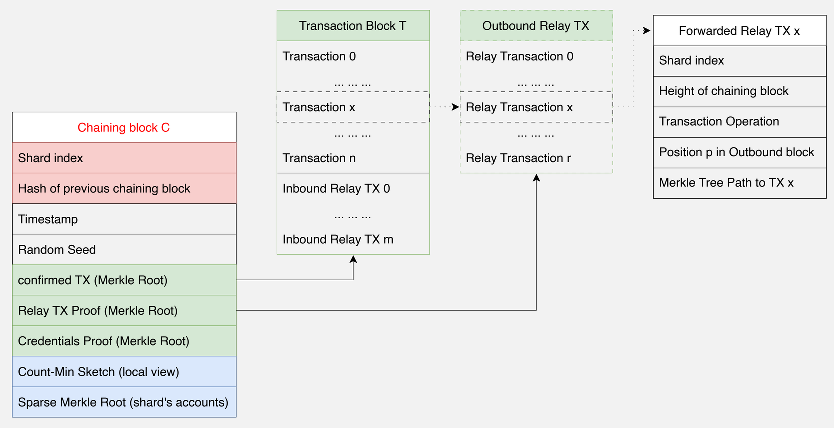
Adaptive elastic sharding, dynamically adpting to load

Localized Management

- Proof of Eligibility [4] at a local level
- Each shard managages a separate set of transactions
- No inter-shard consensus

Broadcast based intershard coordination

- Leverage recent results on money transfer [2, 3]
- Broadcast ephemaral coordiantion blocks
- Organize inter-shard trasaction in a DAG

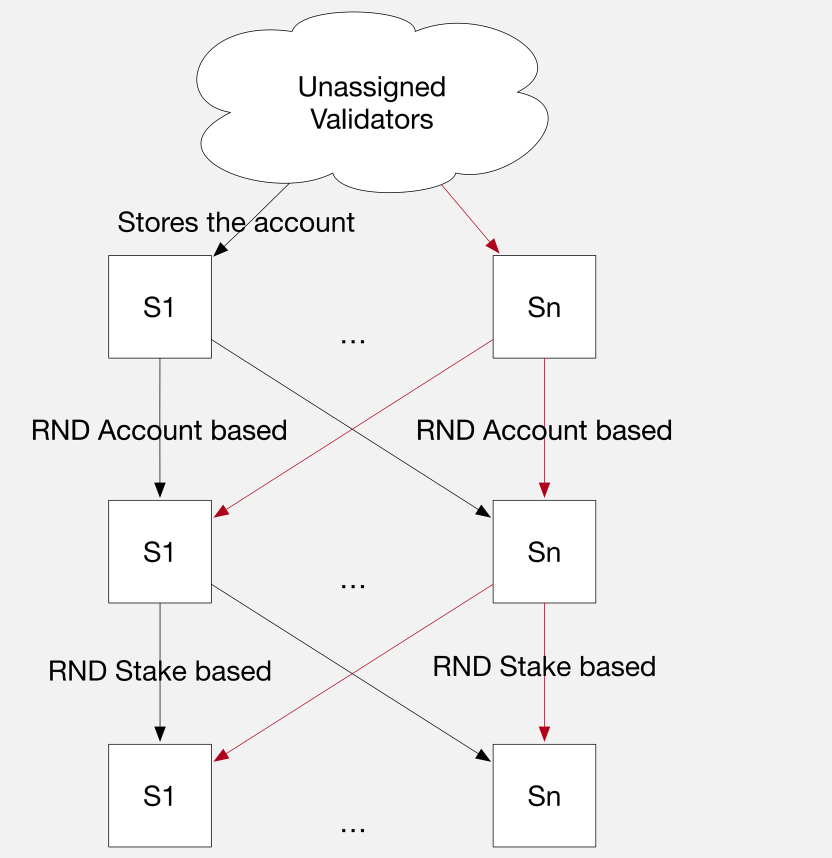


High resistance to attacks

- Resist to 1% attack typical of sharded systems
- Resist to adaptive adversary

Multi-layer eligibility control

- Nodes validate consensus in random shards
- Two steps of indirection
- First steps randomizes participation
- Second step takes into account stake



Outreach

- Brunessen Bertrand and Sandrine Turgis speakers at Colloque L'Europe et les nouvelles technologies, Nanterre, 10/06/2021.
- Blockchain & Privacy Conference (Rennes, 2022) organized by Brunessen Bertrand and Sandrine Turgis, 22 speakers from France, Belgium and Canada. To be published in 2023 with Larcier (editor).
- Timothé Albouy, Davide Frey, Michel Raynal, François Taïani. Good-case Early-Stopping Latency of Synchronous Byzantine Reliable Broadcast: The Deterministic Case. To Appear at DISC 2022, Oct 2022, Augusta, GA, United States.

References

[1] Ittai Abraham et al. "Good-Case Latency of Byzantine Broadcast: A Complete Categorization". In: Proceedings of the 2021 ACM Symposium on Principles of Distributed Computing. PODC'21. Virtual Event, Italy: Association for Computing Machinery, 2021, pp. 331–341. ISBN: 9781450385480. DOI: 10.1145/3465084.3467899. URL: https://doi.org/10.1145/3465084.3467899.

[2] Alex Auvolat et al. "Money Transfer Made Simple". In: CoRR abs/2006.12276 (2020). arXiv: 2006.12276. URL: https://arxiv.org/abs/2006.12276.

[3] Rachid Guerraoui et al. "The Consensus Number of a Cryptocurrency". In: Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing. PODC '19. Toronto ON, Canada: Association for Computing Machinery, 2019, pp. 307–316. ISBN: 9781450362177. DOI: 10.1145/3293611.3331589. URL: https://doi.org/10.1145/3293611.3331589.

[4] Geoffrey Saunois et al. "Permissionless Consensus based on Proof-of-Eligibility". In: 2020 IEEE 19th International Symposium on Network Computing and Applications (NCA). 2020, pp. 1–4. DOI: 10.1109/NCA51143.2020.9306715.