Separ: Towards Regulating Future of Work Multi-Platform Crowdworking Environments with Privacy Guarantees

Mohammad Javad Amiri\textsuperscript{1}, Joris Duguépéroux\textsuperscript{2}, Tristan Allard\textsuperscript{2}, Divyakant Agrawal\textsuperscript{3}, Amr El Abbadi\textsuperscript{3}

\textsuperscript{1}University of Pennsylvania, \textsuperscript{2}Univ Rennes, CNRS, IRISA, \textsuperscript{3}UC Santa Barbara

April 16, 2021
Crowdworking platforms are online intermediaries between requesters and workers.

- Envisioned as key technological components of the future of work.
Guaranteeing the compliance of crowdworking platforms with regulations

“Whereas universal and lasting peace can be established only if it is based upon social justice; . . . for example, by the regulation of the hours of work . . .”

Preamble of the constitution of the International Labor Organization [Commission on International Labor Legislation, 1919]

Figure: Members of the Commission on International Labor Legislation to the Paris Peace Conference (1919).
FLSA: Total work hours of a worker per week may not exceed 40 hours.

In California, Assembly Bill 5 (AB5) entitles workers to greater labor protections, such as minimum wage laws, sick leave, and unemployment and workers' compensation benefits.

CA Proposition 22 imposes its set of regulations, e.g., requires a worker to work at least 25 hours per week to qualify for healthcare subsidies.
There is more than one platform …

• Workers often work on several platforms
• Requesters submit tasks on multiple platforms
Privacy Rights of Participants

- No participant obtains or infers any information beyond what is strictly needed
  - A driver who works for both Uber and Lyft, does not want either of them know that she works for the other.

- How to enforce regulations?
  - We need to reconcile transparency with privacy
Problems!

- Guarantee the compliance of crowdworking platforms with regulations
  - Local (per platform) regulations exist: maximum driving time per day
- Transparent and Privacy-preserving regulation enforcement
- Collaboration among independent competing platforms
  - Enforcement of global regulations
  - Complex tasks that may need multiple contributions
Our Vision for Future Regulation Systems

- **Goal:** Enforce regulations on multi-platform crowdworking environments while preserving privacy

- Three main design dimensions
  - **D1:** Type of supported regulations
    - e.g., aggregate or not
  - **D2:** Privacy guarantees given to participants
  - **D3:** Architecture of the system
    - e.g., fully decentralized, partially centralized

![Diagram of workers, platforms, requesters, and legal and social institutions connected in a network]
Design Space: (D1) Regulation Type

• Express regulations as **SQL** constraints over a universal table **U-TABLE**
• Categorize them according to their **SQL** expression.
• Characterized by:
  • **Complexity**: _simple if no JOIN operation, complex otherwise_.
  • **Aggregate** (presence of Aggregate function(s), with **GROUP BY** and **HAVING**):
    • row-only, aggregate-only and mixed.
• **Enforceable**: must always hold
  • e.g., maximum work hours
• **Verifiable**: must hold periodically
  • e.g., minimum work hours

<table>
<thead>
<tr>
<th>WORKER</th>
<th>PLATFORM</th>
<th>REQUESTER</th>
<th>TIMECOST</th>
<th>CONTRIBUT</th>
</tr>
</thead>
<tbody>
<tr>
<td>w1</td>
<td>p2</td>
<td>r1</td>
<td>3H</td>
<td>...</td>
</tr>
<tr>
<td>w1</td>
<td>p3</td>
<td>r2</td>
<td>2H</td>
<td>...</td>
</tr>
<tr>
<td>w2</td>
<td>p1</td>
<td>r1</td>
<td>6H</td>
<td>...</td>
</tr>
</tbody>
</table>
Regulation Example

- r1: the wage proposed by each task must be at least a given amount $\theta$
- A simple, row-only type of regulation.

```
ALTER TABLE U-TABLE ADD CONSTRAINT r1 CHECK (
    NOT EXISTS (SELECT * FROM U
    WHERE TIMECOST <= $\theta$
    )
);
```
Regulation Example

• r2: each worker works at most a given amount of time units $\theta$ per time period $\rho$
• A simple, mixed with SUM-aggregate regulation.

```
ALTER TABLE U-TABLE ADD CONSTRAINT r2 CHECK ( NOT EXISTS ( SELECT * FROM U WHERE WORKER=w AND current_time()-TS_BEGIN $\leq$ $\rho$ GROUP BY WORKER HAVING SUM(TIMEOUT) $\geq$ $\theta$ ) );
```
Design Space: (D2) Privacy Guarantees

• **Threat model:**
  - e.g., honest-but-curious, covert, malicious
  - System-dependent (we do not specify it further)

• **Privacy model:** pluggable disclosures (to be personalized). We consider:
  - Disclosures to the participants that are not involved in the crowdworking process $\pi$ and that have not received task $t$ from requester $r$: $\delta^\pi_{\neg R \neg I}$
  - Disclosures to the platforms and workers that have received the task $t$ from $r$ but that are not involved in $\pi$: $\delta^\pi_{R \neg I}$
  - Disclosures to the participants that are directly involved in $\pi$ (and have thus received task $t$): $\delta^\pi_{RI}$
Design Space: (D3) Architectural Choices

- Any regulation system is made of two critical components:
  - Regulation management: models and enforces the regulations
  - Global state management: stores the global state of the system

- Each component can be implemented either
  - Centralized
    - easier to rapid prototype
    - difficult to ensure fault-tolerance, privacy, and trustworthiness
  - Decentralized
    - more compatible with the multi-platform settings
    - resulting in more overhead and complex communication protocols among entities
SEPAR: a Point in the Design Space

<table>
<thead>
<tr>
<th>Regulation supported:</th>
</tr>
</thead>
<tbody>
<tr>
<td>• U-TABLE focuses on the interactions and consists in: WORKER, PLATFORM, REQUESTER, TIMECOST.</td>
</tr>
<tr>
<td>• (simple, mixed with SUM-aggregate) regulation, with lower-than (enforceable) or higher-than (verifiable) thresholds.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Privacy guarantees:</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Covert non-colluding adversaries.</td>
</tr>
<tr>
<td>• Aims at inferring anything that can be inferred from the execution sequence</td>
</tr>
<tr>
<td>• Is able to deviate from the protocol if no other participant detects it</td>
</tr>
<tr>
<td>• Disclosures sets: (given crowdfunding process $\pi$: (BEGIN, END, w, p, r, t))</td>
</tr>
<tr>
<td>• $\delta^{\pi}_{R-I} = (\text{BEGIN, END, p})$</td>
</tr>
<tr>
<td>• $\delta^{\pi}_{R-I} = (\text{BEGIN, END, p, r, t})$</td>
</tr>
<tr>
<td>• $\delta^{\pi}_{RI} = (\text{BEGIN, END, w, p, r, t})$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Hybrid architecture:</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Registration Authority (RA): Centralized. Registers participants, models regulations, distributes crypto material.</td>
</tr>
<tr>
<td>• Multi-Platform Infrastructure: Decentralized. Maintains the global state within a blockchain</td>
</tr>
<tr>
<td>• Consensus protocols:</td>
</tr>
<tr>
<td>• Local (nodes of the same platform)</td>
</tr>
<tr>
<td>• Cross-platform (platforms having received the same task)</td>
</tr>
<tr>
<td>• Global (all platforms)</td>
</tr>
</tbody>
</table>
A Simple Token-Based System

• Inspired by e-cash systems, SEPAR implements enforceable and verifiable regulations by managing two budgets per participant
• Lightweight, single-use, and anonymous tokens

The registration authority refreshes participants tokens periodically

• **GENERATE**: initializing the budgets and refilling them
  • Enforceable and Verifiable tokens
• **SPEND**: spending portions of the budgets
• **PROVE**: providing proof for verifiable regulations to a third party
• **CHECK**: checking whether a given spending is allowed or not
• **ALERT**: reporting dubious spending
SEPAR Architecture

Workers: W₁, W₂, W₃, W₄

Platforms: P₁, P₂, P₃, P₄

Requesters: R₁, R₂, R₃, R₄

Registration Authority

At Most f Crash Failures
- quorum A
- quorum B
- Platform size: 2f+1

At Most f Malicious Failures
- quorum B
- quorum A
- Platform size: 3f+1

Separ: Future of Work Multi-Platform Crowdworking
Execution Sequence

Tasks:
- Internal
- Cross-Platform

Transactions:
- Submission
- Claim
- Verification

Initialization
Publication
Assertion
Verification
Execution

Registration Authority
Worker
Requester
Platform
Blockchain (involved platforms)
Blockchain (all platforms)

Tokens
Task
Submission Transaction
Claim Transaction
Exchange of Contribution and Reward

Demand for Tokens
Demand for Signature of Tokens
(Grand) Signature
Verify the Tokens and signatures

Verify the Tokens and signatures
Submission Transaction
Claim Transaction
Exchange of Contribution and Reward

Separ: Future of Work Multi-Platform Crowdfunding
Processing Tasks in Separ

Separ: Future of Work Multi-Platform Crowdworking
Consensus in SEPAR

**Local Consensus:** pluggable and depends on the failure model of nodes

**Cross-Platform Consensus:** Among the involved platforms

**Global Consensus:** Requires the participation of all platforms

<table>
<thead>
<tr>
<th>Transaction/Task</th>
<th>Internal</th>
<th>Cross-Platform</th>
</tr>
</thead>
<tbody>
<tr>
<td>Submission</td>
<td>Local</td>
<td>Cross-Platform</td>
</tr>
<tr>
<td>Claim</td>
<td>Local</td>
<td>Cross-Platform</td>
</tr>
<tr>
<td>Verification</td>
<td>Global</td>
<td>Global</td>
</tr>
</tbody>
</table>

Separ: Future of Work Multi-Platform Crowdworking
Local Consensus

- Required for *submission* and *claim* transactions of *internal* tasks
- Depending on the failure model of nodes
  - Crash failure: (Multi-)Paxos
  - Byzantine failure: PBFT
Cross-Platform Consensus

- Is required for **submission** and **claim** transactions of cross-platform tasks
- A **Byzantine** fault-tolerant protocol is used (untrustworthiness of platforms)
- **Local-majority**: required number of matching replies from nodes of a platform
- Each phase: Agreement from the **local-majority of all involved** platforms
## Global Consensus

- Is needed for **verification** transactions of **all** tasks
  - Verification transaction: group signatures and all tokens that are consumed by participants to perform a particular task
- A **Byzantine** fault-tolerant protocol is run among all nodes of every platform
- Each phase: Agreement from the local-majority of **two-thirds** of the platforms

### Diagram

![Diagram](https://via.placeholder.com/150)

Separ: Future of Work Multi-Platform Crowdfunding
Experimental Settings

- We do not focus on the description of tasks and contributions
- Platform: Amazon EC2
- Measuring performance
  - Throughput
  - Latency
SEPAR is able to generate tokens in linear time.

SEPAR generates each token in 0.07ms (1 million tokens in 76 seconds).

The class of regulations does not affect the performance.
Different Types of Constraints

Crash-Only Nodes

Byzantine Nodes

Privacy-preserving mechanism: only 11% throughput and 15% latency overhead (Crash-only)

The class of regulations does not significantly affect the performance of Separ
Scalability Over Spatial Domains

Platforms in different AWS regions: Tokyo, Hong Kong, Virginia, and Ohio

Privacy-preserving mechanism: only 10% throughput and 13% latency overhead (Crash-only)

privacy-preserving mechanisms have lower overhead in comparison to a setting with a single data center
An overall vision for future of work multi-platform regulation systems based on three dimensions: Type of Regulations, Privacy, Architecture.

SEPAR is the first to address the problem of enforcing global regulation over multi-crowdworking platforms.

A token-based system that enables official institutions to express legal regulations in simple and unambiguous terms.

Supports greater-than and lower-than (simple, SUM-aggregate) regulations.

Implemented over a permissioned blockchain that provides transparency using distributed ledgers shared across platforms.
Questions?
Required Cryptographic Material

- A pair of usual public/private asymmetric keys (e.g., RSA)
- A pair of public/private asymmetric group keys
  - Union of all workers (Requesters/platforms) forms a group (Group Signature)
  - The registration authority is the group manager
- Participants acquire them when joining SEPAR

Signatures:
1. the group signature of the token (which will be later verified by all platforms, together with the token, when it is shared with all platforms),
2. the group signature of the pair consisting of a token and a task.
Increasing the Number of Cross-Platform Tasks

SEPAR is able to process 8600 local tasks with 400 ms latency

Increasing the percentage of cross-platform tasks, reduces the overall throughput
Constraint and certificate tokens

• **Enforceable Token:**
  • is a tuple \((t_p, t_s)\) where
  • \(t_p = (\text{nonce}, (\text{nonce})\sigma_{PK(RA)})\)
  • \(t_s\) is the list of public keys of involved participants

• **Verifiable Token:**
  • is a tuple \((t_p, t_s)\) where
  • \(t_p = (\text{nonce}, (\text{nonce})\sigma_{PK(RA)})\)
  • \(t_s\) is \((\text{nonce}, o,w)\sigma_{PK(RA)}, (\text{nonce}, o,p)\sigma_{PK(RA)}, \text{ and } (\text{nonce}, o,r)\sigma_{PK(RA)}\)
  • Where \(o\) is the identity of the participant owner of the token