IoTrace

A contact tracing app built using IOTA

Luca Giuliani, Alessandro Lombardi, Diego Mazzieri

Alma Mater Studiorum · Università di Bologna

January 27, 2021

Introduction

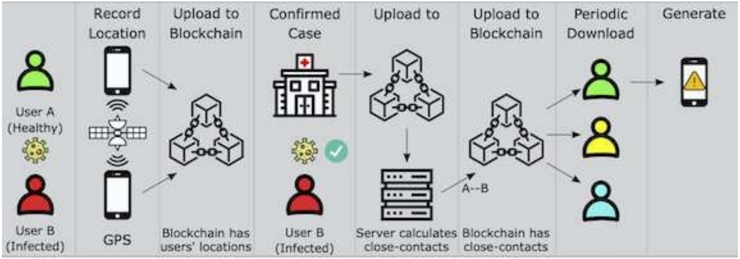
BeepTrace proposes an alternative framework for contact tracing applications based on blockchain technology. The initiative is open and is still under development and testing.

Objectives

- Transparency
- Immutability
- Security
- Privacy

Reference

Xu, H., Zhang, L., Onireti, O., Fang, Y., Buchanan, W.B., & Imran, M. (2020). BeepTrace: Blockchain-enabled Privacy-preserving Contact Tracing for COVID-19 Pandemic and Beyond. ArXiv, abs/2005.10103.



Actors

Users	Each person using actively the application.
Diagnosticians	GPs that verify whether the user is the rightful holder of pseudonyms and sign positive cases.
GeoSolvers	Servers which compute risks and update notifications for the users.
Authorities	Governments or other authorities which provide authorizations and keys.
Pos. Providers	GNSS, Bluetooth, Cellular Towers, and WiFi, used to certificate new users as well

Privacy

Privacy is guaranteed by design enabling only authorized entities to see the information strictly related to their duty.

Blockchain

Solution based on two DLTs, the **notification** one and the **tracing** one. Consensus mechanism represents a crucial decision, they opted for a DAG-based solution. Contact tracing can be seen as a special case of Internet of Things.

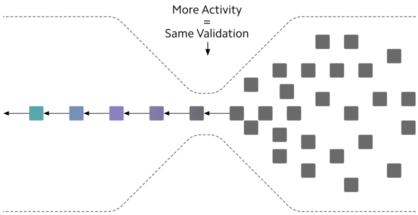
What Is IOTA?

- **Distributed Ledger Technology** for **Internet-Of-Things** scenarios
- Leverages a **Directed Acyclic Graph** data structure (the **Tangle**)

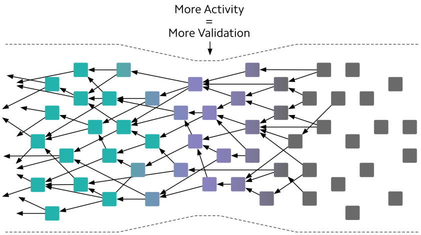
Main Advantages

- Cheap **Proof-Of-Work**, suitable for big data from sensors
- **Scales** well with intense activity
- **Feeless** transactions
- **Public** devnet opened for developers and testing purposes
- Well-documented **APIs** and frameworks

THE BLOCKCHAIN BOTTLENECK



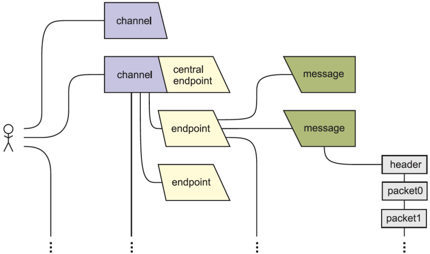
THE IOTA TANGLE SCALES!



IOTA

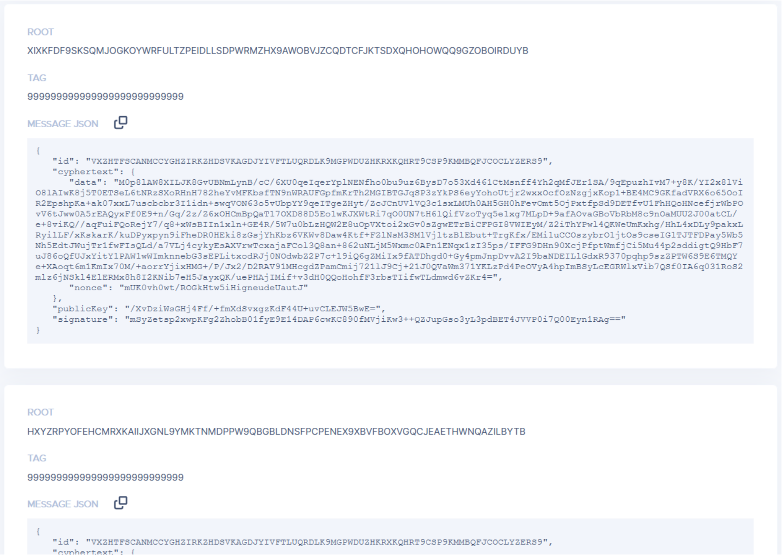
IOTA Limitations

- The Tangle provides **no high-level organisation** for the data
- Transaction can be retrieved by tag but not by date of publications
  - ⇒ this can lead to a waste of time
  - ⇒ e.g., in our case the geosolver would have to download the same transactions from the tangle every time it is run



Mam Channels

- **Mam Channels** – or ***Streams***, in the new release of IOTA – represent an "**abstract layer**" upon the Tangle
- Each channel has a root transaction, and each transaction maintains the link to the next root, thus it is possible to append a new message to the channel
- Given a root, it is possible to fetch all the subsequent messages as if they were part of a linked list
- There is already an API to use Mam Channels in IOTA, as well as an "inspector" at the link: `explorer.iota.org/devnet/streams/0/`



Example of the Mam Channel of an agent inspected with the IOTA explorer.

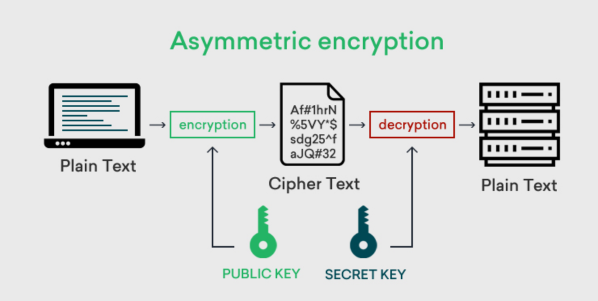
Privacy

Motivation

One of the benefits of using blockchain technology is its transparency and accessibility. How do we protect users' private information? We followed the strategy proposed in the paper by encrypting the information in different portions.

Technology

We used the **Asymmetric Encryption** strategy to encrypt each portion.



Privacy

Signatures

Data is encrypted both to guarantee user' **privacy** and to **avoid intrusions**.

- Knowing the seed of a MAM channel is enough for a malicious node to intrude
- Still, the asymmetric encryption allows to sign the **ciphertexts** with the private key
  - ⇒ It is important to notice that **everybody** can **verify** a signature, as no key is needed
  - ⇒ e.g., in our case the diagnostician is able to verify the correctness of the agents' transactions without the need of decrypting them

Entities' Keys

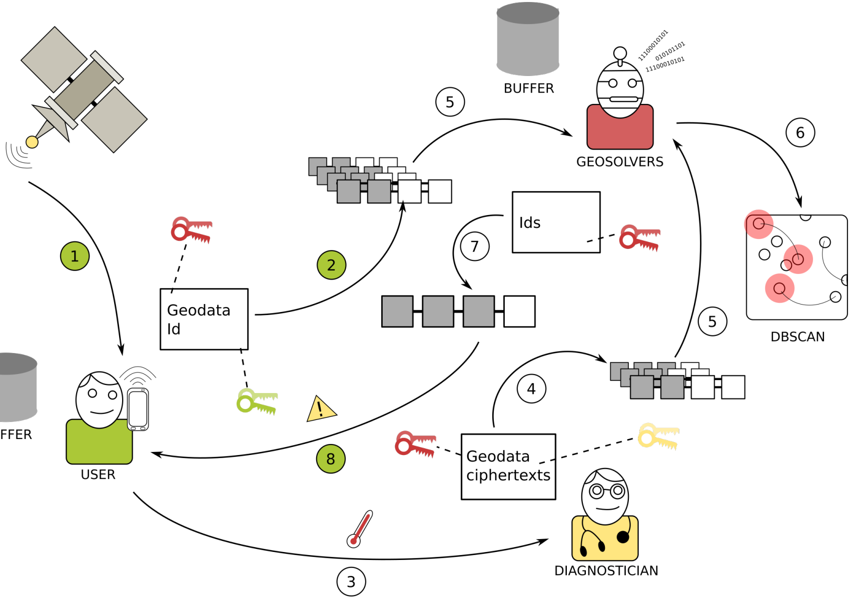
Each entity has its own pair of private and public keys to have a one-to-one encryption.

- **Agents** use their keys to encrypt their personal information for the geosolver
- **Diagnosticians** use their keys to encrypt and sign the messages containing the positive agents' data, which can be decrypted by the geosolver only as well
- The **GeoSolver** uses its keys to decrypt both diagnosticians' and agents' messages

Additionally, as one-to-many encryption is not directly supported, the **GeoSolver** encrypts its data in order to be able to guarantee their authenticity, but these messages can be decrypted by anyone who possesses a shared private key.

Architecture

Agents


BUFFER
GEOSOLVERS
5
6
Ids
7
DBSCAN
1
2
Geodata
Id
5
4
BUFFER
USER
8
Geodata
ciphertexts
3
DIAGNOSTICIAN

Agents

Characteristics

- **Generic individuals** moving in the space
- Have a **unique ID**, which is randomly generated
- Own their personal **MAM channel**
- Own a pair of **keys** to encrypt data for the geosolver
- Own a **shared key** to read data from the geosolver

Tracking Procedure

Agents keep their geospatial data in a cache, which is updated every slot of time, then, whenever it is full:

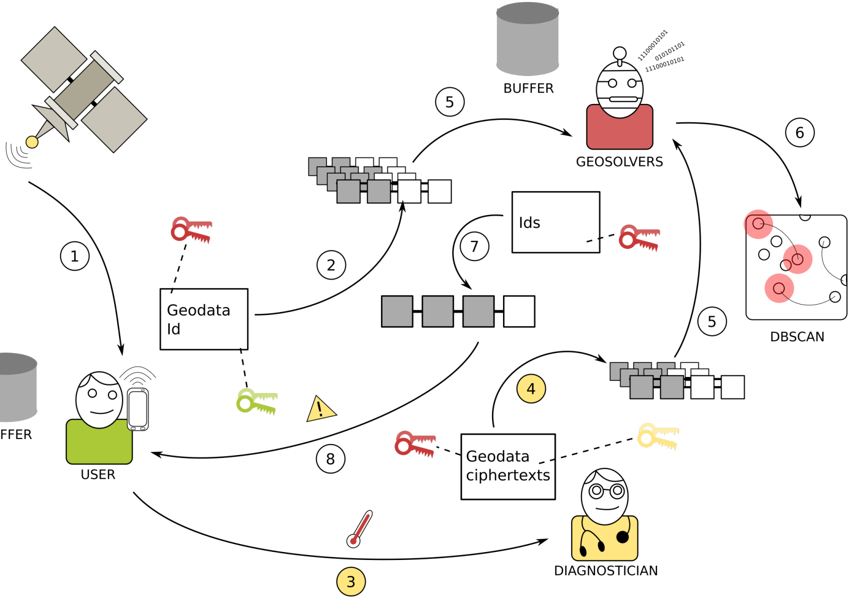
1. Its content is encrypted using the agents' private key and the geosolver's public key
2. The obtained ciphertext is signed using the agents' private key
3. Then, agents append this ciphertext to the MAM channel along with the respective signature and both their public key and their ID

Notification Procedure

When new data is posted on the geosolver's MAM channel, the agents fetch them, then:

1. Verify the correctness of the signature
2. Decrypt the data using the shared private key
3. Check if their ID is in the list of possible infected

Diagnostician



Diagnostician

Characteristics

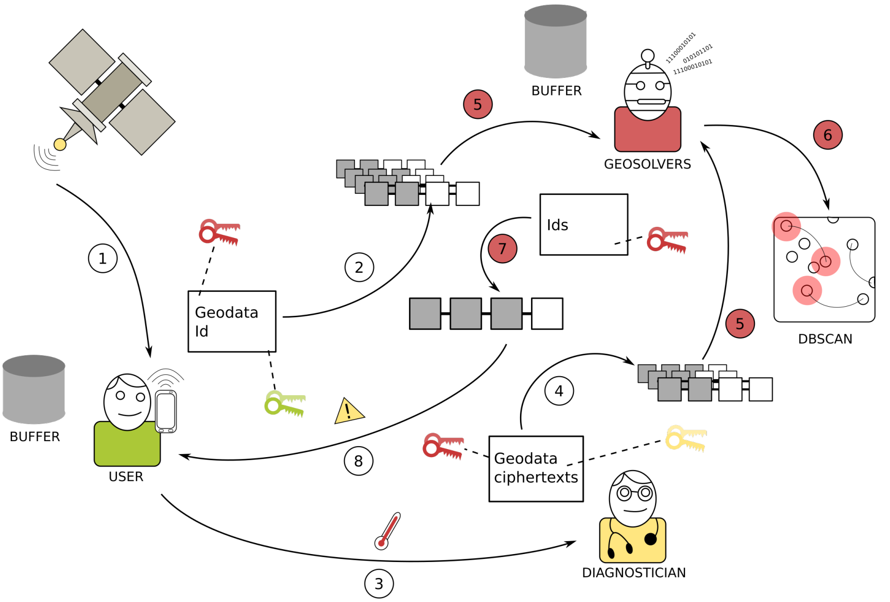
- **GPs** who are in charge to certify positive agents by publishing and signing their data
- Own their personal **MAM channel**
- Own a pair of **keys** to encrypt data for the geosolver

Certification Procedure

Diagnosticians read data from a positive agent's MAM channel, then they:

1. Verify the correctness of the agent's transactions
2. Discard the agent's ID from each transaction, so that the geosolver will not know who is the infected agent
3. Collect the ciphertext of each transaction, then put them in a list which is paired with the agent's public key so that the geosolver would be able to decrypt it
4. Encrypt this pair of data with their private key and the geosolver's public key
5. Sign the obtained ciphertext with their private key
6. Append the ciphertext to their personal MAM channel, along with the signature and their public key

GeoSolver



BUFFER
GEOSOLVERS
5
6
1
2
Geodata
Id
Ids
7
DBSCAN
5
BUFFER
USER
4
8
Geodata
ciphertexts
DIAGNOSTICIAN
3

GeoSolver

Characteristics

- **Server** that is responsible for the computation of possible infected individuals
- Owns its personal **MAM channel**, which has a publicly known seed
- Owns a pair of **keys** to decrypt data from both agents and diagnosticians
- Owns a pair of **keys** to encrypt and sign its messages for all the agents

Infected Computation Procedure

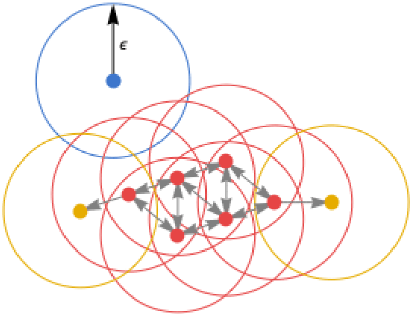
Periodically, the geosolver fetches the new data added to the MAM channels of all the agents and all the diagnosticians, then it:

1. Appends this data to its internal cache, and filters out transactions that are too old
2. Runs a **DBSCAN** algorithm on the cached data to compute the IDs of the possible infected agents based on spatial and temporal proximity
3. Encrypts this list of IDs using its private key and a public key which is paired to the shared key used by the agents
4. Signs the obtained ciphertext with its private key
5. Appends the ciphertext to its personal MAM channel, along with the signature

GeoSolver

DBSCAN

- **Clustering algorithm** like K-means
- **Density-based** rather than centroid-based
- Two hyperparameters:
  - ϵ Distance
  - n Number of neighbors within that distance (1 in this case)



Clusters

- All diagnosticians' transactions share the same fictional ID, representing an infected.
- Once clusters are computed by the algorithm, if in a cluster the aforementioned ID is present, all the members are notified.

Simulation

Motivation

Why a Simulator?

In order to **show** a possible (small scale) **real world scenario** of the application and display the macroscopic interactions between the multiple entities.

Customization

It is possible to **configure** almost **every detail** of the simulation, for instance:

- Number of agents involved
- Pace of the simulation clock
- Infection probability
- Publish frequency of the agents

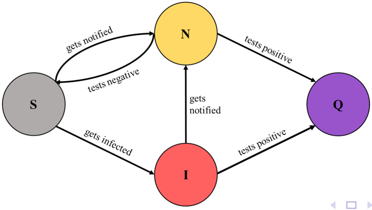
Uniqueness

- Any two simulation runs are **independent**.
- Any simulation has a **unique seed** from which the simulation is generated. All the **MAM channels** involved in the simulation are **generated using** this **seed**.
- Running a new simulation **without changing** the **seed** results in **unpredictable results** due to multiple writings at the same address.

Scenario

Agents

- The state of the agent can vary depending on the evolution of the simulation between: **susceptible**, **infected**, **notified**, **quarantined**.
- **Normally**, the target position of an agent changes randomly or can be manually imposed.
- An infected agent can **transmit** the disease to a normal agent with a given probability and according to a selected proximity range.
- If the agent is **infected**, he can exhibit symptoms with a given probability and therefore the target will become going to his diagnostician.
- If the agent is **notified**, the target immediately becomes going to his diagnostician.
- If the agent is **quarantined**, he cannot obviously move.



Key Moments

Beginning

1. Only one agent is infected at the beginning of the simulation.
2. All the agents subscribe to the geosolver communicating their MAM channels.
3. When the simulation starts, agents move from their initial position and interact each other.

Patient Zero

1. Infection continues to spread across the agents until one infected goes to his diagnostician.
2. The geosolver has now enough information to calculate some possible other threatening users and writes their IDs in its MAM channel.
3. Each agent, notified by the geosolver, compare his ID with the ones written in the geosolver's channel to understand if he needs to go to the diagnostician.

Convergence

As more infected users are discovered, more gets notified and the convergence is reached when no more infected agents circulate in the simulated environment.

Future Works

Deployment

IoT Context

- We developed a **simulator** since we did not have the possibility to run tests in a real world context
- A real-word application should be deployed for mobile devices
  - ⇒ the "**back-end**" part for this application can be easily adapted from our code
  - ⇒ the simulator should only be used for regression tests
  - ⇒ the "**front-end**" must use device sensors API
  - ⇒ Some suitable form of **pub/sub** mechanism for IoT (e.g. MQTT)

Scalability Issues

- Our simulation runs on a single computer
- We managed to simulate concurrency through multiple threads
- We could not consider possible scalability issues regarding computation, memory consumption and throughput, problems also present in the original paper