IN, LCA1: Decentralized Verifiable Computation on Distributed Ledger

Due on Summer 2018

D.Froelicher, J.Troncoso-Pastoriza

Max Premi

April 29, 2018

Abstract

Data sharing systems are becoming more popular these past years, and are used in several domains, such as economics [REF], software validation [REF], and even medical field [REF]. They eliminate singles point of failures, enforces transparency, and provides an efficient way to ensure security, authentication, and privacy. The latter is required to respect privacy laws but introduces overhead such as encryption, verification of computation (correctness), and tracking of error (robustness).

UnLynx is such a system that uses Elliptic curve ElGamal encryption, zero-knowledge proof, and noise addition, as well as several other protocols to maintains all properties stated above. However, it only supports a small subset of operation (sum, count, average), and it has a strong threat model.

This project is a part of a design of a new decentralized system Lemal, supporting a large set of operation (mean, variance, logistic regression,...), while keeping all properties in an efficient way, in addition of universal verifiability of computations and results. In this project, we introduce a way to make all the proofs public and verifiable by anyone, through distributed ledger. The goal of this project is to:

- First, propose a theoretical system implementation of the Skipchain, to create a Collective Authority of Verifying nodes that guarantees correctness and robustness of computation.
- Then, implement protocols that handle the Skipchain operations, based on the previous implementation done by DeDis.
- Implement the interface between the Collective authority doing the computation and the Collective authority of verifying nodes.
- Finally, measure the performance of this system.

Contents

1 Introduction

Blockchain [REF] technology has emerged in 2008, with the Bitcoin, that use distributed and decentralized ledger to create token and exchange them in a trusted way with immutability. However, its application can be extented to other topics, such as support for correctness and robustness of computation, as well as completely public zero-knowledge proofs.

Indeed, in a data sharing system, one may want to prove to any party that what he has done is correct. Under certain threat model, it can be assumed that all parties are malicious, and it is needed to make tremendous effort to prove that what has been done is correct. A distributed ledger can easily make a decentralized authority that can be consulted at anytime to get data that were inserted and cannot be modified.

In this paper, we present the implementation of Skipchain [REF] into Lemal's framework, to deal with robustness and correctness of computation done by Lemal system. The chain will store information to verify the computation via zero-knowledge proof, without leaking anything more than what the proofs actually leak. Then a performance evaluation is done to look at the effiency of such an implementation

2 Contribution

3 Background

This section introduces some fundamental concepts used throughout the rest of the report. *Collective Authority* that is the base of both system functionality. *ElGamal encryption* is used in Lemal to ensure privacy, while *Skiphain* is used in the verifying node as distributed Ledger. This section also introduce some fundamental background about Blockchain, as Skipchain is a structure derived from Blockchain.

3.1 Collective Authority

Nowadays, applications and systems rely on third-party authorities to provide security services. For example the creation of certificates to prove ownership of a public key. A collective authority is a set of m servers that are deployed in a decentralized and distributed way, to support a given number of protocols.

Each of them possesses a private-public key pair (k_i, K_i), where $K_i = k_iB$ with k_i a scalar and K_i a point in a given Elliptic Curve. This authority constructs a public key $K = \sum_{i=1}^{m} K_i$ which is the sum of all the server's public keys. To decrypt a message, each server i partially decrypts a message encrypted using k_i. Thus the collective authority key provides strongest link security, as no intermediate can decrypt the data without the contribution of all the servers.

3.2 ElGamal Encryption

All the involved scalars belong to a field $\mathbb{Z}_p$.

For Unlynx, data are encrypted using Elliptic Curve ElGamal, more precisely, P is a public key, x is a message mapped to a point and B is a base point on the curve γ. The encryption is the following, with r a random nonce:

$E_P(x) = (rB, x+rP)$. The additive homomorphic property states that $\alpha E_P(x_1)+\beta E_P(x_2) = E_P(\alpha x_1+\beta x_2)$

To decrypt, the owner of the private key p satisfying $P = pB$ multiplies rB and p to get rP and substracts it from $x + rP$ to recover x.

3.3 Skipchain and Blockchain

4 Lemal System

This section presents Lemal [REF] system in general. It goes throught the system design, the assumptions made about the parties taking part in the different protocols, the properties that hold, and a example of a query that the system can handle.

4.1 System Model

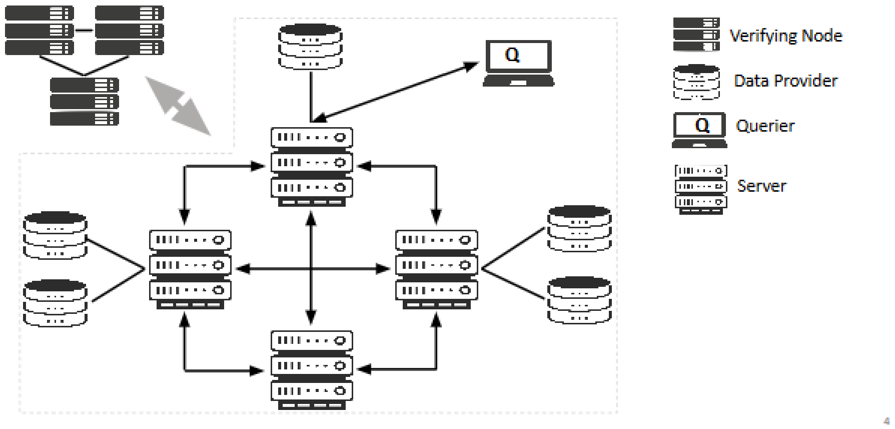

Figure 1: System model of Lemal

Lemal is a privacy-preserving data sharing system developed in collaboration of this project by LCA1. It consists of a collective authority (CA formed by m servers, $S_1, ...S_m$ and n data providers $DP_1, ...DP_n$ containing sensitive data, encrypted using EC ElGamal. Another collective authority of verifying nodes is linked to this system, which maintains the distributed ledger, is described in Section 5. This sytem is used to answer queries made by a querier Q, to produce results of some aggregate functions. Each DPs choose one server of the CA to communicate with and can change this at any given time.
Functionality: Lemal permits a large set of SQL queries, like *Where, group by, like, mean, variance, set intersection*, with some machine learning (*linear and logistic regression*) and Private recommender system functionality such as *cosine similarity, CBF-Based recommendations, ...*.
For any query, proofs are computed and stored by verifying nodes in a skipchain so that any party can verify what has been done.

4.2 Threat Model

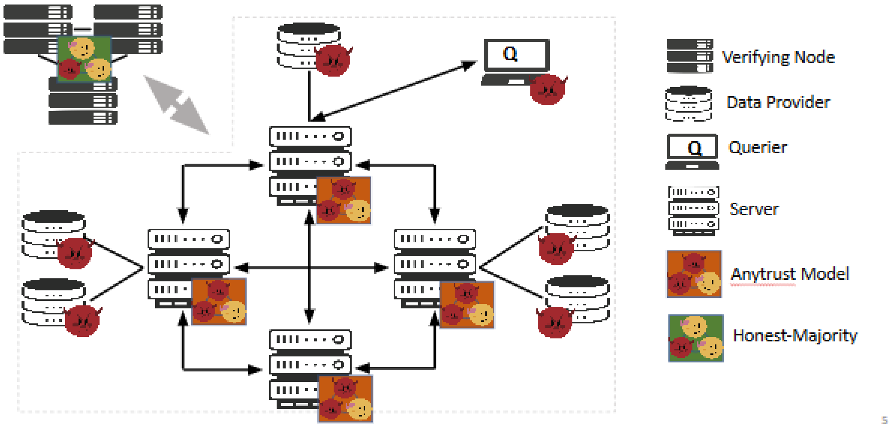

Figure 2: Threat model of Lemal

Collective authority servers It is assumed an Anytrust model [14]. It does not require a particular server to be trusted or to be honest-but-curious. Whenever one of the server is not malicious, functionnality, security and privacy are guaranteed.
Collective authority verifying nodes It is assume an Hones majority. Meaning a threshold of more than half of the nodes are honest, and the consensus is done via Byzantine fault tolerance protocol.
The Querier, and Data providers are assumed malicious. They can collude between themselves or with a subset of the CA servers.
It is also assumed that all network communication is encrypted and authenticated by using a protocol such as TLS or SSL.

4.3 Properties

Confidentiality All data are encrypted using EC ElGamal, no party sees the data in clear, except the Querier that get the aggregate result encrypted under his public key. So this property holds as long as one server is honest **Privacy Correctness** All step of the protocol **Robustness**

4.4 Query example

5 Verifying Node

5.1 System Model

5.2 Threat Model

5.3 Type of Proofs

5.4 Operation supported

6 Implementation

7 Performance Evaluation

8 Conclusion and Future Work

References

[1] Henry Corrigan-Gibbs, Prio prototype implementation
`https://github.com/henrycg/prio`