UnLYNX: A Decentralized System for Privacy-Conscious Data Sharing

PETS 2017

More medical data are digitized

Percentage of office-based physicians with electronic medical records in U.S.A, 2001-2013

National Ambulatory Medical Care Survey (NAMCS)
MORE HEALTH DATA COLLECTED


More medical data = better treatments?

Cancer death rates* among men, USA, 1930-2014

*Per 100,000, age adjusted to the 2000 US standard population. †Mortality rates for pancreatic and liver cancers are increasing.

Note: Due to changes in ICD coding, numerator information has changed over time. Rates for cancers of the liver, lung and bronchus, uterus, and colon and rectum are affected by these coding changes.

**More medical data = better treatments?**

Cancer death rates* among men, USA, 1930-2014

*Per 100,000, age adjusted to the 2000 US standard population. †Mortality rates for pancreatic and liver cancers are increasing.

Note: Due to changes in ICD coding, numerator information has changed over time. Rates for cancers of the liver, lung and bronchus, uterus, and colon and rectum are affected by these coding changes.

**Source:** US Mortality Volumes 1930 to 1959 and US Mortality Data 1960 to 2014, National Center for Health Statistics, Centers for Disease Control and Prevention.
More medical data = better treatments?

Cancer death rates* among men, USA, 1930-2014

*Per 100,000, age adjusted to the 2000 US standard population. †Mortality rates for pancreatic and liver cancers are increasing.

Note: Due to changes in ICD coding, numerator information has changed over time. Rates for cancers of the liver, lung and bronchus, uterus, and colon and rectum are affected by these coding changes.

Sensitive-data sharing is difficult
Sensitive-data sharing is difficult

http://www.gmmill.net/proje/Grain-Storage-Silos
Allow statistical queries on multiple independent databases while ensuring privacy and confidentiality for data providers.
Existing data sharing solutions

**Centralized Solutions**

- Single point of failure

**Decentralized Solutions**

- Limited number of data providers/computation entities in an adversarial model
Requirements

- Confidentiality
- Accountability
- Unlinkability
- Correctness
- Differential Privacy
- Decentralized Trust
Building Blocks

Collective Authority
Building Blocks

Additively-homomorphic ElGamal crypto scheme

Collective Authority
Building Blocks

Zero-Knowledge Proofs of Correctness

Additively-homomorphic ElGamal crypto scheme

Collective Authority
Building Blocks

Verifiable Shuffle

Zero-Knowledge Proofs of Correctness

Additively-homomorphic ElGamal crypto scheme

Collective Authority

https://simple.wikipedia.org/wiki/Zero-knowledge_proof
• Collective authority of $m$ servers $S$
• $n$ Data Providers $DPs$
• Clients $Q$ querying the system
• m-1 servers out of m are malicious (Anytrust Model)
Threat Model

- $m-1$ servers out of $m$ are malicious (Anytrust Model)
- Data Providers are honest-but-curious
Threat Model

- m-1 servers out of m are malicious (Anytrust Model)
- Data Providers are honest-but-curious
- Queriers are malicious

**DP = Data Provider**
**S = Server**
Query Processing Workflow

1. Initialisation (Step 0)
2. Query (Step 1)
3. Verifiable Shuffle (Step 3)
4. Distributed Deterministic Tag (Step 4)
5. Collective Aggregation (Step 5)
6. Distributed Results Obfuscation (Step 6)
7. Key Switch (Step 7)
8. Decryption Using the Querier’s Private Key (Step 8)

Undertaken by:
- Querier
- Data Provider
- Collective Authority
Each server constructs his public-private ElGamal Key pair.
Workflow - Initialisation (Step 0)

Each server constructs his public-private ElGamal Key pair.

Collective Key: \( \mathbf{K} = \mathbf{K}_1 + \mathbf{K}_2 + \mathbf{K}_3 \)
Workflow - Initialisation (Step 0)

Each server constructs his public-private ElGamal Key pair.

Collective Key: $\mathcal{K} = \mathcal{K}_1 + \mathcal{K}_2 + \mathcal{K}_3$

Data Providers use the Collective Key to encrypt their data.
Workflow - Query (Step 1)

Query (Step 1)

```
SELECT SUM(cholesterol_rate), COUNT(*)
FROM dp1, ..., dp20
WHERE age IN [40:50] AND ethnicity = caucasian
GROUP BY gender
```
Workflow - Query (Step 1)

Initialisation (Step 0)

Query (Step 1)

SELECT SUM(cholesterol_rate), COUNT(*)
FROM DP1, ..., DP20
WHERE age in [40:50] AND ethnicity = caucasian
GROUP BY gender

Query broadcasted to Data Providers
Workflow - Response (Step 2)

<table>
<thead>
<tr>
<th>ID</th>
<th>Gender</th>
<th>Age</th>
<th>Ethnicity</th>
<th>flu</th>
<th>Cholesterol_rate</th>
<th>cancer</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>E(1)</td>
<td>E(40)</td>
<td>E(1)</td>
<td>E(1)</td>
<td>E(23)</td>
<td>E(0)</td>
</tr>
<tr>
<td>2</td>
<td>E(2)</td>
<td>E(40)</td>
<td>E(2)</td>
<td>E(0)</td>
<td>E(34)</td>
<td>E(0)</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

[[group. attr.], [where. attr.], [aggr. Attr.]]
[[E(1)], [E(40), E(1)], [E(23), E(0)]]
...

Initialisation (Step 0)
Query (Step 1) -> Response (Step 2)
Each server starts a verifiable shuffle protocol:

In this protocol each server sequentially:

- **Shuffle** the list of responses
- **Rerandomize** (re-encryption) all the ciphertexts

Using Neff Shuffle and the corresponding zero-knowledge proof [1]

Each server starts a distributed deterministic tagging protocol:

**Query:**
WHERE age = $E_K(40)$ AND ethnicity = $E_K(2)$

WHERE age = $DT(40)$ AND ethnicity = $DT(2)$

**Data:**
[[E_K(1)], [E_K(40), E_K(2)], [E_K(23), E_K(1)]]

[[DT(1)], [DT(40), DT(2)], [E_K(23), E_K(1)]]
Each server starts a **distributed deterministic tagging protocol**:

In this protocol each server sequentially:

- **partially decrypt** the ciphertexts
- **Blinds** the message by multiplying the ciphertexts with a random **ephemeral secret key**

→ deterministic tag depending on the value of the encrypted message

All operations are done with zero-knowledge proofs from Camenisch et al.

---

Workflow - Collective Aggr. (Step 5)

- **Initialisation (Step 0)**
- **Query (Step 1)** → **Response (Step 2)**
- **Verif. Shuffle (Step 3)** → **DDT (Step 4)** → **Collective Aggregation (Step 5)**

Servers **collectively aggregate** the responses by group.

Proofs consist in publishing the ciphertexts and the result.
**Workflow - DRO (Step 6)**

**Distributed Results Obfuscation:**

**Setup:**

Servers agree on $(\epsilon, \delta)$-differential privacy parameters and produce:

$$[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, ...]$$

= list of noise values satisfying $(\epsilon, \delta)$-differential privacy.
**Distributed Results Obfuscation:**

**Runtime:**
- A server starts a **collective shuffling** of the list of noise values
- **adds the first noise value** in the list to the query result.

→ **Oblivious noise addition** (shuffling **encrypts and shuffles** the list of noise values).
Initialisation (Step 0)

Query (Step 1) ➔ Response (Step 2)

Verif. Shuffle (Step 3)

DDT (Step 4) ➔ Collective Aggregation (Step 5) ➔ DRO (Step 6) ➔ Key Switch (Step 7)

In the key switch protocol each server:
- partially decrypt
- encrypt with a new key all the ciphertexts.

Encryption is switched from the Collective Key to the querier’s public key.

All operations are done with zero-knowledge proofs from Camenish et al.

**Workflow - Decryption (Step 8)**

Querier **decrypts** the result with his secret key.
**Servers configuration**
- Memory: 256GB RAM
- Processor: Intel Xeon E5-2680 v3 (Haswell)
- Cores: 24 (with 48 threads)
- Frequency: 2.5GHz
- Bandwidth capacity: 1Gbps

**Network and Crypto**
- Realistic virtual network emulation tool with 10ms delays btw. servers
- DeDiS’ Onet library
- DeDiS’ implementation of Ed25519 Elliptic Curve (128-bit security)

**Default parameters**
- 3 servers
- 15,000 responses in total (equally distributed in servers)
- 1 GROUP BY attribute with 10 possible values, 1 WHERE and 10 aggregating attributes
- 1000 noise values
Servers collaboration

DDT = Distrib. Deterministic Tagging
DRO = Distrib. Results Obfuscation
Runtime vs. nбр. of responses

![Runtime vs. number of responses](image)

**Communications**
- 15K: 8.4s
- 150K: 72s
- 1,500K: 672s

**Verif. Shuffle + DDT**
- 15K: 0.3s
- 150K: 0.3s
- 1,500K: 0.3s

**Other**
- 15K: 15s
- 150K: 37s
- 1,500K: 371s

**Verif. Shuffle + DDT Proof**
- 15K: 14.6s
- 150K: 2,086s
- 1,500K: 4,376s

**Verif. Shuffle + DDT Verify.**
- 15K: 5.6s
- 150K: 6.3s
- 1,500K: 153s
Performance/Security Tradeoffs

EVERYTHING IS ENCRYPTED

SELECT SUM(cholesterol_rate), COUNT(*)
FROM dp1, ..., dp20
WHERE age in [40:50] AND ethnicity = caucasian
GROUP BY gender

3 servers
400K responses with
1 GROUP BY attribute
2 WHERE attributes
2 aggregating attributes
**Conclusion**

A Decentralized System for Privacy-Conscious Data Sharing

- SQL statistical queries based on Boolean conditions
- Strongest-link security
- Data confidentiality
- Distributed differential privacy
- Distributed deterministic tagging of probabilistic ciphertexts
- Collective encryption key switching

- Runtime linear with the amount of data to process

[GitHub](https://github.com/lca1/unlynx)  
David.Froelicher@epfl.ch