A Privacy-Assuring Disclosure Filter for Genetic Information

Vinicius Vielmo Cogo
Ph.D. student from **University of Lisbon** (M5)

Working on BiobankCloud project (EU/FP7)

Advised by **Prof. Alysson Bessani**

**Current work activities:**
- **Assuring the feasibility** of our solution
- **Implementing** the first prototype
Biobanks store **biological samples** and **genetic information** for use in research.
In a near future, biobanks will not support the growth of genetic data.
Using cloud computing is a natural **economic alternative** to local IT infrastructures.

**Motivation**

- “Infinite” resources
- Low prices
- Pay-as-you-go
Motivation

However, it increases the attack surface and strengthens challenges on privacy.
Now it is about privacy
Privacy Risks and Attacks

- The main privacy risk:
  - Disclosure of private information about individuals

- Protect from two types of attacks:
  - Re-identification
    To whom an intercepted DNA belongs?
  - Differentiation
    Does this known person participates of this study?
Example

- An attacker intercepts a DNA sequence
  ...GATGCATGCATGCTAGCCTAGCCTAGCATGGGGATCG...

- Somehow he re-identified Bob

- Bob’s privacy will never more be recovered (it is not like a passwd)

- The privacy of Bob’s family is also compromised, even for several future generations
Our Approach: Disclosure Filter

- Prevent the disclosure of DNA sequences that may compromise donor’s privacy
Challenges

1) How to classify DNA sequences as private or public?

2) How to implement an efficient disclosure filter?
Challenge 1: DNA classification

- **Filter:** Black list of small DNA sequences

- Two initial classification **algorithms:**
  - **Genetic genealogy profiling**
    Based on counting small repetitions
    Used in forensic identification and paternity tests

  - **Rare variants present on individuals**
    Based on statistics about rare mutations
    Similar goals than differential privacy
Challenge 2: Efficiency

- **High throughput:**
  - Good engineering
  - High parallelism
  - Load balancing
  - Pipelined workloads

- **Space-efficient data structure:**
  - Bloom filter
    - Probabilistic
    - No false negatives
    - Configurable false positives
Thank you!

Vinicius Vielmo Cogo (vvcogo@fc.ul.pt)
Surnames are paternally inherited
As Y-chromosomes in male individuals
STR: a small repeating sequence in DNA

Ex.: DYS392

- Location: Chromosome Y
- Initial Pos.: 22633873
- Final Pos.: 22633911
- Pattern: \([TAT]^n\)
- Example of \([TAT]^5\):
cgac TAT TAT TAT TAT TAT TAT tcga
Genetic Y-STR Profile

- **Select** a list of known Y-STR
- **Count** repetitions of each of them
- **Search** in databases for similar results

Ex.: Individual 1

- DYS19: 20
- DYS389I: 18
- DYS389II: 31
- DYS390: 25
- DYS391: 14
- DYS392: 9
- DYS393: 12
Bloom Filter

- Efficient data structure (space and performance)
- Test if an element is member of a set (probabilistic)
- False positives are possible (inside set - may be wrong)
- False negatives are not (definitely not in set)

\[
\{x, y, z\}
\]

\[
0 \quad 1 \quad 0 \quad 1 \quad 1 \quad 1 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 1 \quad 0 \quad 0 \quad 1 \quad 0
\]

- \(k\) hashes algorithms (\(k=3\) in our example)
- \(w\) does not belong to the set because it maps to a position set to zero