Improving HIV treatment choice with multi-party computation

Mark Abspoel

13 September 2021

CWI, Cryptology Group

Based on TKI project with TNO, CWI, UvA/IAS, Philips.

Joint work with: Thomas Attema (CWI/TNO), Ronald Cramer (CWI), Serge Fehr (CWI), Jan de Gier (TNO), Maran van Heesch (TNO), Pia Kempker (TNO), Emiliano Mancini (UvA/IAS), Peter Sloot (UvA/IAS), Gabriele Spini (CWI), Thijs Veugen (CWI/TNO), Daniël Worm (TNO).
Affiliations at the time of the project (2017–2018).

Context: TKI project

- Private/public collaboration between:
  - TNO
  - CWI, Cryptology group
  - University of Amsterdam, Institute of Advanced Studies
  - Philips Research
- Project duration: 1 year, start mid 2017
- Goal: innovative application of MPC techniques to practical use cases

Results

- Identified two use cases in the medical domain
- Developed solution using MPC
- Proof of concept implementations

Choosing HIV treatment

- Treating HIV is not straightforward: multiple possible treatments, many different viruses
- Virus mutates as it replicates. Bad treatment leads to more replication, which means:
    - Treatment failure
    - Accumulation of drug resistances
    - Faster progression to AIDS
- Even with optimal treatment, virus will eventually mutate

Doctors' decisions

Doctors have ≈ 5 minutes per patient to take decisions based on

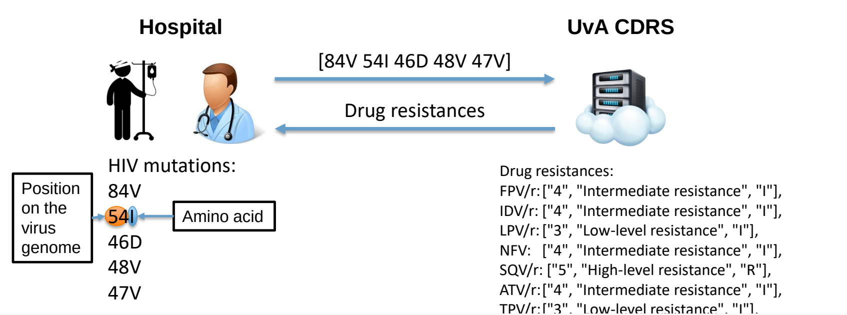
- Guidelines based on medical research
- Knowledge
- Experience

Doctors' decisions

Doctors have ≈ 5 minutes per patient to take decisions based on

- Guidelines based on medical research
- Knowledge
- Experience

UvA developed Comparative Drug Ranking System (CDRS) to assist doctors.



Using treatment data

CDRS is based on current research / clinical trials.

Every time a patient needs new treatment $\rightarrow$ feedback on prior treatment.
Can we use this data?

Using treatment data

CDRS is based on current research / clinical trials.

Every time a patient needs new treatment → feedback on prior treatment.
Can we use this data?

Two problems:

1. Doctors do not want to publish decisions for liability concerns
2. Patient's HIV genotype is privacy-sensitive

Using treatment data

CDRS is based on current research / clinical trials.

Every time a patient needs new treatment $\rightarrow$ feedback on prior treatment.
Can we use this data?

Two problems:

1. Doctors do not want to publish decisions for liability concerns
2. Patient's HIV genotype is privacy-sensitive

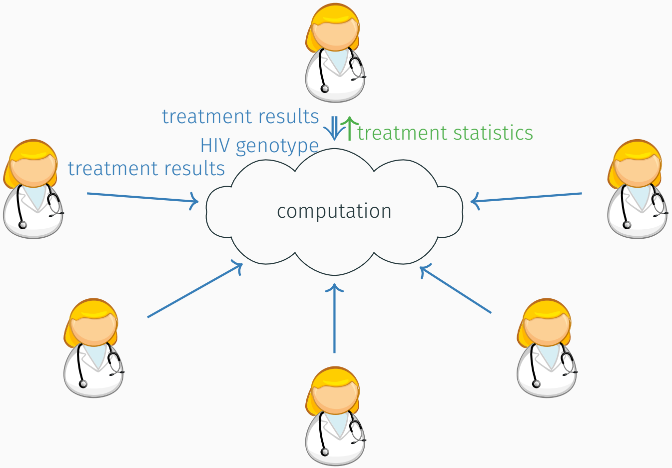
Solution: multi-party computation!

Protocol to run each time

Given a patient's HIV genotype, for each treatment compute average time to failure for patients with similar HIV virus

Protocol to run each time

Given a patient's HIV genotype, for each treatment compute average time to failure for patients with similar HIV virus



Secret-shared database

Long running computation $\rightarrow$ computation parties maintain an encrypted (secret-shared) database:

$$(\texttt{HIV genotype}, \texttt{treatment}, \texttt{time to failure})$$

Genotype is encoded as vector of relevant mutations (length = ℓ).

Secret-shared database

Long running computation $\rightarrow$ computation parties maintain an encrypted (secret-shared) database:

$$(\texttt{HIV genotype}, \texttt{treatment}, \texttt{time to failure})$$

Genotype is encoded as vector of relevant mutations (length = ℓ).

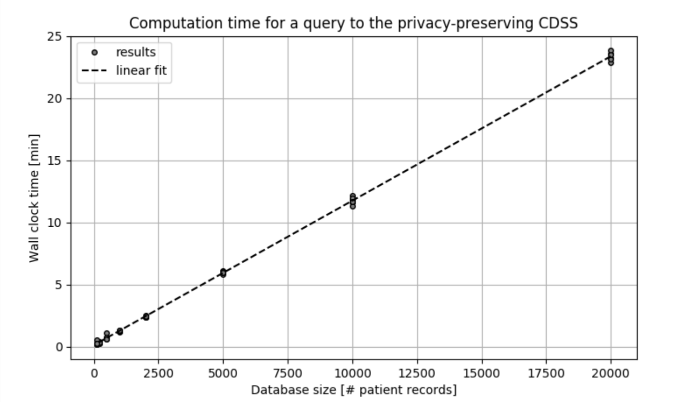
To query database, we need to check against each row!

$\implies$ Computation scales linearly in:

- The number of treatments Q
- The length ℓ
- The number of rows N

Implementation

- Encode genotype as binary vector of relevant mutations, $Q = 100, \ell = 200, N \leq 20\,000$.
- Implementation using Bristol-SPDZ framework (predecessor of MP-SPDZ / SCALE-MAMBA), 2 machines connected through LAN, "Low Gear"



Lessons learned

- Identifying a good use case can be hard.
  Initial use case: search CDRS without leaking patient genotype.

Lessons learned

- Identifying a good use case can be hard.
  Initial use case: search CDRS without leaking patient genotype.
  Computation could be done locally without MPC! But we run into other organizational challenges (frequent updates, fast machines).
  When to use MPC: *mutual privacy requirement*

Lessons learned

- Identifying a good use case can be hard.
  Initial use case: search CDRS without leaking patient genotype.
  Computation could be done locally without MPC! But we run into other organizational challenges (frequent updates, fast machines).
  When to use MPC: *mutual privacy requirement*
- MPC enables new solutions

Lessons learned

- Identifying a good use case can be hard.
  Initial use case: search CDRS without leaking patient genotype.
  Computation could be done locally without MPC! But we run into other organizational challenges (frequent updates, fast machines).
  When to use MPC: *mutual privacy requirement*
- MPC enables new solutions
- Performance of MPC can be good enough for practical applications