guardtime

APRIL 2020

Guardtime Health
Real World Data Engine

Classification: Public

GUARDTIME.COM

Executive Summary

The Real World Data Engine by Guardtime Health has been designed to provide aggregate reports across multiple health care providers in a privacy-preserving manner.

The system primarily targets answering questions of the form “how many patients there are that satisfy condition X?”, where the data needed to evaluate the condition X reside in different data sources (such as health care providers, medical registries, etc.). Condition X above could in practice mean “how many patients use drug Z for indication Y”, or “how many patients using drug Z had a biomarker D with a value below E or above F”. The data is assumed to be sensitive, thus cannot be sent out or shared with other providers without privacy-preserving measures. The system:

a) breaks the question down into sub-questions “how many patients satisfy A?” and “how many patients satisfy B?”, where A and B are pre-agreed sub-conditions such that they can each be answered from a single data source and X can be expressed as their logical combination (“A and B”, “A or B”, “A but not B”, etc.);
b) receives the answers to sub-questions from the respective data sources;
c) uses secure multi-party computation (MPC) techniques to combine these sub-question answers and obtain the result without leaking any sensitive information.

Internally the system operates on lists of patient identifiers. The identifiers are encrypted locally at the data source and never decrypted. The MPC protocol operates on encrypted data only and no cleartext ever leaves the premises of the data source.

Externally, only the number of patients (the size of the result list) is reported as the answer. Even the encrypted lists are not communicated outside the parties of the MPC protocol.

The system supports distributed auditing: the correctness of behavior of each participant can be verified separately and independently, implying both the correctness of the results of the whole computation and the privacy of patient data.

The rest of this document gives an overview of the architecture and the technologies used to achieve these properties.

General Architecture

The overall architecture of the system is shown in Figure 1. Each data source (such as an electronic health record, a hospital pharmacy, a national out-patient prescription database, a disease/patient registry, etc.) operates an MPC service node that participates in the secure MPC protocol on their behalf. The MPC nodes of all data sources communicate with the coordinator node to jointly execute the following MPC protocol.

Each MPC node has a locally generated encryption key. For each reporting period, the MPC node receives from the respective data source the list of patient identifiers answering a pre-agreed sub-question of the form “which patients satisfy A?” or “which patients satisfy B?”. The MPC node then encrypts the list with its encryption key and only the encrypted identifiers participate in the MPC protocol.

Each MPC node is hosted and operated by the data source. As each MPC node encrypts the list of patient identifiers, no cleartext of the patient data ever leaves the premises of the data source.

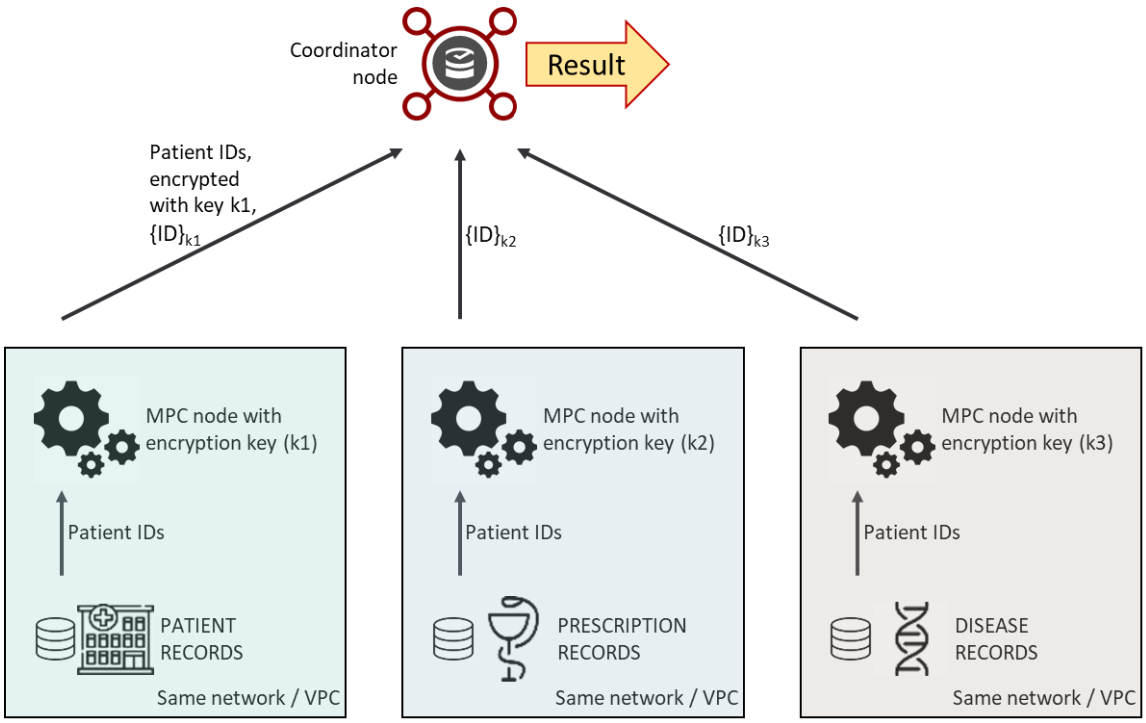

Figure 1: General Architecture.

Secure Multi-Party Computation

To obtain the answer to the primary question “how many patients there are that satisfy condition X?” based on the encrypted answers to the sub-questions “which patients satisfy A?” or “which patients satisfy B?”, the MPC nodes with the help of the coordinator node execute an MPC protocol based on a commutative encryption scheme.

In general, when a value is encrypted repeatedly, the result depends on the order in which the keys are applied. When a value V is doubly-encrypted with two keys k1 and k2, the result of applying k1 first and k2 second (which we can denote as $V_{k1,k2}$) is generally different from the result of applying k2 first and k1 second ($V_{k2,k1}$). An encryption scheme where it is guaranteed that the result is the same in both cases (that is, $V_{k1,k2}=V_{k2,k1}$) is called commutative.

The MPC protocol based on a commutative encryption scheme works as follows:

- First, each MPC node encrypts the list of identifiers from its data source with its encryption key and sends the result to the coordinator. For example, if {ID1} denotes the list from the first data source, then its MPC node sends $\{ID1\}_{k1}$ to the coordinator.
- The coordinator sends each list through other nodes in sequence. The other nodes each apply their own encryption to the already encrypted list. For example, $\{ID1\}_{k1}$ from the first data source becomes $\{ID1\}_{k1,k2}$ and then $\{ID1\}_{k1,k2,k3}$.
- Finally, all lists will have an encryption layer from each MPC node, as shown in Figure 2. The commutative encryption scheme ensures that an identifier that appears in multiple lists will have the same encrypted form in all the lists it appears in.
- This enables the coordinator to find the size of the intersection or union of the lists (the number of values that appear in all lists, or the total number of unique values in all the lists combined), and report this as the result of the computation. However, neither the coordinator nor any of the MPC nodes will be able to recover the cleartext identifiers making up any of the lists.

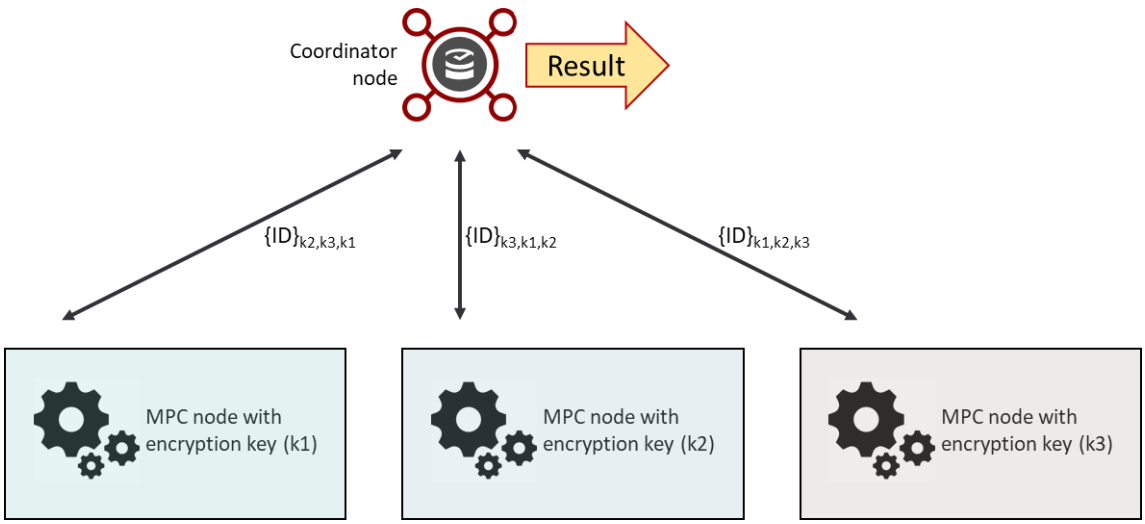

Figure 2: Multi-Party Computation over Encrypted Data.

Distributed Auditing

The system also supports a distributed auditing mechanism that allows the behavior of each party to be verified separately and the global correctness of the process to be deduced from these local audits. This removes the need to have one auditor that everyone must trust to look at their confidential data (if such a trusted party existed, all data sources could just send their inputs to that party in cleartext and have them announce the result).

The audits are supported by a bulletin board seen by all the parties (Figure 3). Before a party sends out an encrypted identifier list, it posts a commitment of the list on the board. The recipient only accepts the message if it matches the commitment. The commitments are computed using one-way hash functions, so the original data cannot be recovered from them. Yet, the commitments are binding: it is infeasible to come up with two different data sets to match the same commitment.

To remove the need to trust the bulletin board to operate correctly, two mechanisms are used. First, all updates to the board are cryptographically time-stamped, so that anything posted cannot be changed after the fact. Second, in addition to encryption key, each node also has a signing key and uses it to digitally sign the commitments before posting them to the board, so that no other parties can post commitments in the name of that node.

These features allow the board to be used as a common reference point that links the local audits of all parties into an integrated whole and allows concluding the correctness of the global process from the consistency of all local audits, and also provides hard evidence in case some party has misbehaved.

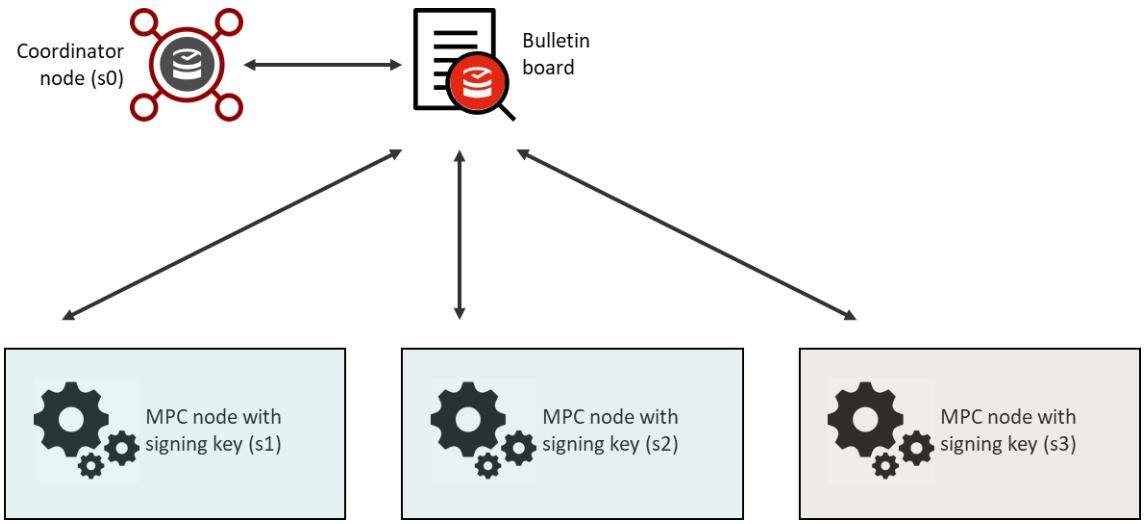

Figure 3: Bulletin Board for Distributed Auditing.