A distributed personal medical system

Kesaobaka Moipolai, Irene Nandutu, Helen Zha

January 2021

1 Introduction

A medical system is a database of records of patients' medical histories and relevant personal information. The movement of people across countries motivates a medical system which is accessible internationally. A distributed design removes reliance a central server, improving the system's security and robustness. But it introduces problems of privacy, which the system must ensure.

We design a medical system using the abstract construction of Z-formalism, which includes states, operations and preconditions on operations. This allows for concise and clear presentation of the functionality of the system, which any implementation must satisfy.

In this work we define the state and a sample of the operations, including specifying how privacy will be ensured. The design also exploits the potential for data analytics, and we give an example of a relevant operation. Finally, we introduce an implementation based on blockchain. With more time this approach would lead to a complete design and implementation.

2 Abstract Datatype

We define the system under consideration as an abstract datatype: having state and operations.

2.1 State

In order to define state, it is convenient first to define the following.

$$
\begin{array}{|l}
\multicolumn{1}{l}{\mathit{Med}} \\
\hline
\mathit{his} : \mathrm{seq}\,\mathit{MedCon} \\
\mathit{cur} : \mathbb{P}\,\mathit{CurMed} \\
\mathit{con} : \mathbb{P}\,\mathit{All} \\
\mathit{med} : \mathbb{P}\,\mathit{MedCond} \\
\mathit{vac} : \mathbb{P}\,\mathit{Vac} \\
\hline
\end{array}
\qquad
\begin{array}{|l}
\multicolumn{1}{l}{\mathit{Per}} \\
\hline
\mathit{fin} : \mathit{FIN} \\
\mathit{pi} : \mathit{PI} \\
\mathit{sus} : \mathbb{P}\,\mathit{All} \\
\hline
\end{array}
$$

Each person's personal identifiers are associated with three functions, d, e, f, which express medical information *Med*, personal information *Per*, and living state (Boolean) respectively. The state satisfies two invariant properties. All current allergies, medical conditions, medications, and vaccines are contained within the medical history. Anyone who is dead doesn't have any current allergies, medical conditions, medications, or financial coverage.

$$
\begin{array}{|l}
\multicolumn{1}{l}{\mathit{State}} \\
\hline
d : \mathit{ID} \nrightarrow \mathit{Med} \\
e : \mathit{ID} \nrightarrow \mathit{Per} \\
f : \mathit{ID} \nrightarrow \mathbb{B} \\
\hline
\forall\, i : \mathit{ID} \bullet d(i).\mathit{cur}, d(i).\mathit{con}, d(i).\mathit{med}, d(i).\mathit{vac} \in d(i).\mathit{his} \\
\forall\, i : \mathit{ID} \bullet f(i) = 0 \;\Rightarrow\; d(i).\mathit{cur} = d(i).\mathit{con} = d(i).\mathit{med} = \{\,\} = e(i).\mathit{fin} \\
\hline
\end{array}
$$

We assume that the state has been realistically initialised.

2.2 Operations

We concentrate on the successful case of the operations. Outside their preconditions, they are assumed to give an error message.

Recall that public key cryptography has the following property:

$$\mathit{pub}_a(\mathit{priv}_a(x)) = x,$$

where the left-hand side enables anyone to confirm that the agent a signed x using their private key. To validate accessors we define the function

$$\mathit{valid}(a, \mathit{pat}, \mathit{sign}) := \left(\begin{array}{c} a \in \mathit{ApprovedList} \\ \mathit{pub}_a(\mathit{sign}) = \mathit{pat} \end{array} \right)$$

where $\mathit{sign} = \mathit{priv}_a(\mathit{pat})$.

In order to enforce privacy requirements, we introduce different access levels: shallow, medical practitioner, and patient. We will give an example of each in this document.

Shallow Access

Shallow access is used to give limited information to agents such as border-control officers or the front desk of a hospital. The database contains sensitive information, but *ShallowRead* allows access only to a patient's personal information, financial coverage, allergies, and vaccinations.

$$
\begin{array}{|l}
\multicolumn{1}{l}{\mathit{ShallowRead}} \\
\hline
\Xi \mathit{State} \\
\mathit{acc}?, i? : \mathit{ID} \\
\mathit{sign}? : \mathit{SIGN} \\
\mathit{info}! : \mathbb{P}\, \mathit{Vac} \times \mathbb{P}\, \mathit{All} \times \mathit{Per} \\
\hline
\mathit{valid}(\mathit{acc}?, i?, \mathit{sign}?) \\
\mathit{info}! = (d(i?).\mathit{vac}, d(i?).\mathit{con}, e(i?)) \\
\hline
\end{array}
$$

Medical practitioner access

Medical practitioner access is used to allow doctors and nurses to read any information, add to medical history and vaccines, and edit current medications, current medical conditions, and confirmed allergies. *MedUpdate* allows a validated medical practitioner to make additions.

$$
\begin{array}{|l}
\multicolumn{1}{l}{\textit{MedUpdate}} \\
\Delta \mathit{State} \\
\mathit{doc}?, i? : \mathit{ID} \\
\mathit{sign}? : \mathit{SIGN} \\
p? : \mathit{Med} \\
\hline
\mathit{valid}(\mathit{doc}?, i?, \mathit{sign}?) \\
d' = d \oplus \{i? \mapsto p?\} \\
e' = e \\
f' = f \\
\hline
\end{array}
$$

Patient access

Patient access allows patients to read any information, and edit their personal information and suspected allergies.

$$
\begin{array}{|l}
\multicolumn{1}{l}{\textit{PatUpdate}} \\
\Delta \mathit{State} \\
a?, i? : \mathit{ID} \\
\mathit{sign}? : \mathit{SIGN} \\
p? : \mathit{Per} \\
\hline
\mathit{valid}(a?, i?, \mathit{sign}?) \\
e' = e \oplus \{i? \mapsto p?\} \\
d' = d \\
f' = f \\
\hline
\end{array}
$$

3 Implementation

To implement our requirements, we introduce a MediChain that includes pages of the state of a patient. Our system consists of nodes, each of which contain the latest copy of MediChain. Interaction with the nodes (to add a page or update information) is via a medical wallet or booklet. The nodes may be located for example in hospitals and airports, and the medical booklet could take the form of a phone app.

Every time a page is added to the chain, the hash function SHA-256 is applied to it. A nonce is incremented until its hash matches some target, which takes about ten minutes. This length of time ensures that, provided the majority of users are non-malicious, it is highly unlikely that a malicious user will successfully edit the chain, thus ensuring security of the system.

This approach has been shown to be valid in the implementation of blockchain for bitcoin.