Secure Evaluation Protocol for Personalized Medicine

Mentari Djatmiko, Arik Friedman, Roksana Boreli, Felix Lawrence, Brian Thorne, and Stephen Hardy

NICTA, Sydney, Australia
{firstname.surname}@nicta.com.au

Abstract. The increasing availability and use of genome data for applications like personalized medicine have created significant opportunities for the improved diagnosis and treatment of a range of serious medical conditions. The use of such highly personal and identifiable data also has a negative side, with a potential to be used for discrimination (e.g., based on a predisposition to specific illnesses or conditions), or other targeting of individuals, thereby presenting a set of serious challenges in privacy and security. In this paper, we propose a secure evaluation algorithm to compute genomic tests that are based on a linear combination of genome data values (we use the Warfarin dosing algorithm as a representative example). Our proposal relies on a combination of partially homomorphic Paillier encryption and private information retrieval. We implement a prototype system that includes the Paillier encryption part of our protocol in an Android app and back-end server architecture. We perform an initial evaluation of the implemented system and demonstrate a good potential for real time use in a physician-patient scenario, with a response time of around 200ms in a Wi-Fi communications environment.

1 Introduction

The recent advances in genomics have led to cost effective and efficient genome sequencing, with state-of-the-art machines that can sequence a whole human genome at a cost below $US1,000[1]. In parallel, there is a huge increase in the use of genome information for medical research, clinical diagnosis and treatment. Clinical genomics focuses on the use of genomic sequencing information in patient diagnosis and treatment. Personalized medicine utlizes genomics to assist the diagnosis and to tailor the treatment of diseases in regards to both choice of drugs and the level of dose to suit individuals[2]. However, the use of genomic data introduces serious privacy and security concerns, as this data contains highly personal information. Genomic data can be related to a person's predisposition to a number of diseases including mental illness, it can also determine paternity and ancestry, ethnic origin, etc. Prior research work has identified privacy issues and research challenges in genomics [2] and demonstrated how anonymization techniques are ineffective for this highly identifying data. For example, researchers have shown that the genome owners can be de-identified [20]. In personalized medicine, there is also a requirement to protect the details of specific medical tests, which can have significant commercial value.

In this paper, we investigate a secure computation technique for personalized medicine. In particular, we propose a secure evaluation algorithm to compute genomic tests that are based on a linear combination of test-specific genome components (Single Nucleotide Polymorphisms, i.e., SNPs) and coefficients defined by the test. We use the Warfarin Pharmacogenetic dosing [13] algorithm as an example of such tests. In the considered scenario, a clinician requires the test result, and the test resides on a dedicated medical repository server. Our aim is to protect both the confidentiality of the genome data and the details of the medical test. We note that in this scenario, the person's genome data is stored on a mobile device or a personal cloud service. This is in line with other research work advocating personal data storage (e.g., personal vaults [26], and [14] for genome data), as means of enabling direct control of user's personal information.

Prior works in privacy preserving personalized medicine focused on a different type of tests i.e., determining the compatibility of a patient with a specific drug (e.g., using genome sequence comparison [14]) or have considered a different system architecture, where (encrypted) genome data is stored on a dedicated (shared) server. Compared to prior work, we make the following contributions.

[1] "Australia's Garvan Institute of Medical Research acquires machines that can sequence a whole human genome at a base cost below $US1,000", at http://www.garvan.org.au/news-events/news/the-future-of-genomic-medicine-has-arrived-in-australia

[2] "Personalized medicine", at http://lifescientist.com.au/content/biotechnology/article/personalised-medicine-1203527424

We propose a secure computation scheme for privacy preserving genomic personalized medicine tests. Our proposal relies on a combination of partially homomorphic encryption (Paillier encryption [30]) and private information retrieval (PIR) [11].

We study the feasibility of the scheme for the Warfarin dosing test example, considering the communications and computation complexity. We show that the scheme using a constant rate PIR mechanism based on [19] would enable private computing based on genome data, while requiring a relatively small communications overhead of under 100 kB.

We implement a prototype application of the Warfarin dosing test. This comprises an app and the back-end server implementing Paillier encryption. We characterize the performance of our application and show that, compared to a scheme that offers no privacy, the delays introduced by the partially homomorphic based computation are in the order of 200 ms (for a 1,024 bit public key encryption). This is in line with delays deemed acceptable for online browsing [22] and could therefore be considered acceptable for general public (i.e., patient) use. Future work includes evaluation of the app via a user acceptability study.

The organization of the paper is as follows. In Section 2 we present the common use cases for genome data use and formulate the problem. The background on cryptographic protocols is discussed in Section 3 and we present the proposed secure computation protocol in Section 4. In Section 5 we present a preliminary implementation and characterization, followed by the discussion in Section 6. Section 7 discusses the related work in genome privacy. We conclude in Section 8.

2 Use Cases for Computing Based on Genome Data

In this section, we describe the use cases considered more broadly in our work and a specific personalized medicine case that we address in the remainder of the paper.

We envisage a scenario where the patients have direct control of their genome data, that is stored either in the patient's mobile device or in a personal cloud. The genome data is encrypted (and permuted for the latter case), to enable privacy protection from the cloud operator and to ensure that any security breach in either the mobile device or the cloud storage will not expose the valuable data. We note that in the cloud storage case the patient would still have direct control of access to data, via the encryption key stored on their mobile device. Therefore, the patient's explicit agreement is required for any use of their data. In the remainder of the paper we consider the use of a mobile device, but the approach we describe can be easily extended to support personal cloud storage.

There are a number of possible use cases for genome data, we summarize the common examples below.

Personalized medicine: During a clinical consultation, a doctor needs to prescribe a drug to treat a patient for a diagnozed illness; for this treatment, a personalized dose based on the patient's genetic information and general traits (age, height, weight, etc.) needs to be calculated. The dose is an outcome of a specific test (selected by the doctor), that resides, e.g., on a medical repository (server). The dose calculation may be the confidential intellectual property of a pharmaceutical company, and therefore cannot be revealed to the doctor or transferred (in the clear) to the patient's device.

Medical research setting: A medical researcher is investigating the association between a given disease and the presence of certain variations in an individual's genome. The scientist designs an algorithm that can detect these associations, taking into account confounding factors such as environment. In order to be able to access a large pool of genome data, volunteers are invited to take part in the test. The test algorithm is uploaded into the individual's mobile devices, executed and the result (statistically aggregated genome data) is returned to the researcher. In addition, the user may be asked to answer questions relating to the confounding factors, and other data relating to these factors may be collected from their devices.

The main difference between the two use cases is the type of algorithm being evaluated: in the first case, the calculation comprises a combination, e.g., string matching or a linear combination of genome data of a single user and the test data; in the second case, a (large) number of genomes are utilized to calculate more complex statistical functions. In the remainder of this paper, we focus on the personalized medicine use case.

2.1 Personalized Medicine in a Clinical Setting

Dose calculation for personalized medicine[3] uses genetic variation from the patient's genome, such as Single-Nucleotide Polymorphisms (SNPs). SNP refers to DNA sequence variation which occurs when a single nucleotide in the genome differs between members of a biological species or a paired chromosome within the same species. DNA consists of nucleotides (i.e., organic molecule that serves as a subunit of DNA), which has four nuclelobases (alphabets): guanine (G), adenine (A), thymine (T) and cytosine (C). As a consequence, the SNPs also consist of the four alphabets. Based on dbSNP Build 138[4], the total number of SNPs in human DNA as of April 2013 is $62,676,337$ ($44,278,189$ SNPs have been validated). Note that the number of SNPs is calculated based on the available human DNA samples, and as a result the number may increase in the future as more DNA samples become available.

We use the Warfarin dose calculation as an example personalized medicine test. The pharmacogenetic weekly Warfarin dose algorithm, summarized in Table 1, uses a combination of the patient's genomic and non-genomic data. The non-genomic (general traits) data include physical traits (i.e., height and weight), demographic information (i.e., age and race) and whether the patient takes certain medications (i.e., enzyme inducer and amiodarone status). The genomic data focus on the genetic variation of two genes, CYP2C9[5] (a gene which produces enzyme that metabolises Warfarin) and VKORC1[6] (a gene for which mutation can be associated with Warfarin resistance).

Table 1. Pharmacogenetic Warfarin Dose Algorithm [13, Supplementary Appendix S1e.]

	5.6044
−	0.2614 × Age (decades)
+	0.0087 × Height (cm)
+	0.0128 × Weight (kg)
−	0.8677 × VKORC1 A/G
−	1.6974 × VKORC1 A/A
−	0.4854 × VKORC1 genotype unknown
−	0.5211 × CYP2C9 *1/*2
−	0.9357 × CYP2C9 *1/*3
−	1.0616 × CYP2C9 *2/*2
−	1.9206 × CYP2C9 *2/*3
−	2.3312 × CYP2C9 *3/*3
−	0.2188 × CYP2C9 genotype unknown
−	0.1092 × Asian race
−	0.2760 × Black/African American race
−	0.1032 × Missing/mixed race
+	1.1816 × Enzyme inducer status
−	0.5503 × Amiodarone status
=	Square root of weekly Warfarin dose

2.2 Problem Formulation

The actors in the personalized medicine use case and the interactions between them are shown in Figure 1. The medical practitioner (physician) initially authorizes the patient (client device) to access a specific

[3] "SNPs – a shortcut to personalized medicine", at http://www.genengnews.com/gen-articles/snps-a-shortcut-to-personalized-medicine/2507/

[4] dbSNP is a free public archive of genetic variation hosted by NCBI, Build 138 is available at http://www.ncbi.nlm.nih.gov/projects/SNP/snp_summary.cgi?view+summary=view+summary&build_id=138

[5] Cytochrome P450, family 2, subfamily C, polypeptide 9 (http://www.ncbi.nlm.nih.gov/gene/1559)

[6] Vitamin K epoxide reductase complex, subunit 1 (http://www.ncbi.nlm.nih.gov/gene/79001)

server-based medical test from the test library; this could be done e.g., via a bespoke app that would scan a QR code on physician's screen, thus obtaining an access token. The device subsequently downloads this test. We assume that test calculations are performed on the client's mobile device and that the medical server can also perform some related computing functions. The device derives the test result, which can subsequently be used for prescribing the medication dose. In the following, we focus on the interactions between the client device and the medical server, as authentication is a standard component of online systems (we will present an example app designed for this scenario in Section 6).

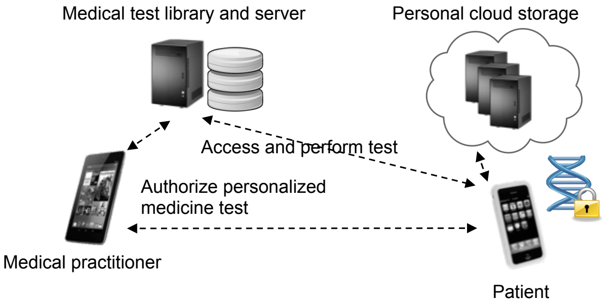


Fig. 1. Personalized medicine scenario and players

Threat Model The computations in this scenario need to consider the confidentiality of (1) the patient's data (i.e., the genome and other personal data should not be revealed to the test provider); and (2) the personalized dose formula which may be the pharmaceutical company's trade secret (i.e., the dose formula should not be revealed to the client). We consider the honest-but-curious adversarial model, where one of the participants attempts to infer the input data of another participant (i.e., the dose calculation for the patient and the genome data for the test provider) while conforming to the protocol.

Formal Description We formally describe the problem for the Warfarin dosing algorithm, presented in Table 1. The algorithm is a linear combination that uses input data from the patient (i.e., the client) and coefficients from the test provider (the server). Note that the Warfarin dosing formula coefficients consist of fixed-point values with a precision of 4 digits. For the simplicity of securely computing the formula, we assume that all input data are integers. To enable integer based computation of floating point coefficient values (for the Warfarin example), each coefficient should be up-scaled by 10^4. After the result of the computation is obtained, scaling down by the appropriate factor results in the correct computation outcome.

Let us assume that the client stores the patient's data (both genomic and non-genomic) in a vector. Furthermore, we assume that both the server and the client know the type of data in the vector, given a specific location (e.g., the data in location index i is patient's age).

The client input consists of variables $x_1, x_2, \ldots, x_n$, which correspond to the patient's genomic (i.e., genome variation such as SNPs) and non-genomic data. Each x_i is either an integer value (e.g., age in decades or height in cm) or it could be a binary indicator (e.g., of whether the patient exhibits a certain genetic variation, where 1 means the presence of the variation). The server inputs consist of a vector of coefficients $c_1, c_2, \ldots, c_m$. We first define a data subset of size l that will be combined (securely) with the m coefficients, taking into account the values of genetic variations. We assume, without loss of generality, that the first set of j coefficients is related to non-genomic data (this would be a simple reorganization of the information in Table 1). Therefore: $x_{i_1}, \ldots, x_{i_j}$ represent non-genomic data and $x_{i_{j+1}} \ldots x_{i_l}$ represents genome related client data. Finally, genetic variations used in the proposed protocol described in Section 4 necessitate the definition of processed values that will be used for computing: $x'_{i_1}, \ldots, x'_{i_m}$. We note that the secure computation used in this paper is defined as a linear combination of the elements of two vectors: $c_1, c_2, \ldots, c_m$ and $x'_{i_1}, \ldots, x'_{i_j}, x'_{i_{j+1}}, \ldots, x'_{i_m}$. The final goal of the protocol is for the client to learn $\sum c_i \cdot x'_i$

(without learning anything else about the inputs from the server), and for the server to learn nothing. I.e., we securely evaluate the function $f(\{x'_i\}, \{c_i\}) \rightarrow (\sum c_i \cdot x'_i, \perp)$.

2.3 Communication and Computation Constraints

To design a practical solution (as presented in Section 6) we need to consider the communication and computation constraints of the client mobile device. As we assume that in the clinical setting, the secure evaluation to compute the Warfarin dose is performed during a patient's visit to the doctor, we relate the computation constraints and data transmission delays to the total delay experienced by the patient. Overall, the secure evaluation should not be longer than the delay of a standard online service, e.g., web page access during browsing – we assume that this would be the most familiar delay for most patients and therefore acceptable.

3 Background

Based on the problem description outlined in Section 2, the target secure computation could be divided into two components: a method to enable secure combination of a linear combination of values in the client device (so that both personal data and test coefficients remain confidential) and a mechanism that enables secure access of information in a way that the accessed party does not learn what is retrieved (the mechanism to keep the locations to which the coefficients relate to confidential). To address these, we consider two types of cryptographic techniques, described below. We note that an alternative approach considering a mechanism that can deliver both components with one type of solution is a subject for future study.

Partially homomorphic encryption techniques enable computation of a linear combination of encrypted values, as long as the addition and multiplication are not performed in a single step. As a good compromise between the provided security and complexity of the scheme, we consider the use of Paillier cryptosystem [30].

The secure selection of the patient's input data can be achieved in a number of ways. Using a naive approach, the server could send a vector of (probabilistically encrypted) coefficients to the client, with a length equivalent to the total number of the input data and the non-used elements represented as 0s. This simplistic solution is not practical due to the high communication overhead, since the patient's input data has around 62 million elements (note the Warfarin evaluation only uses less than 18 input data). A more complex solution could be achieved by k-out-of-n Oblivious Transfer (OT) [12, 27]. However, the communication complexity of k-out-of-n OT protocol is similar to the complexity of the naive solution. In this work, we focus on Private Information Retrieval (PIR) techniques [1, 9, 11, 19, 25], as here the communication complexity is bounded (i.e., it is expected to be sublinear to the size of the patient's input data) [29], resulting in a (theoretically) feasible solution.

3.1 Paillier Cryptosystem

Paillier cryptosystem [30] is an encryption scheme with additive homomorphic properties. Encryption and decryption using Paillier scheme require, respectively, a public key (n, g) and a private key (λ, μ). The public key is calculated as: $n = p \cdot q$ (where p and q are large prime numbers) and g is a random integer, where $g \in \mathbb{Z}^*_{n^2}$. The private key is computed as: $\lambda = \text{lcm}(p-1, q-1)$ and $\mu = (L(g^\lambda \mod n^2))^{-1} \mod n$ (where $L(u) = \frac{u-1}{n}$). Given the public key, a plaintext s can be encrypted by calculating $g^s \cdot r^n \mod n^2$, where $r < n$ is a random number. In addition, decryption of ciphertext c requires both the public and private key, and is performed by evaluating $L(c^\lambda \mod n^2) \cdot \mu \mod n$.

The homomorphic properties of the Paillier cryptosystem are as follows:

1. The sum of two plaintexts $s_1 + s_2 \mod n$ can be calculated by decrypting the product of two ciphertexts $(D(E(s_1) \cdot E(s_2) \mod n^2))$, or the product of a ciphertext with g to the power of a plaintext $(D(E(s_1) \cdot g^{s_2} \mod n^2))$.
2. The multiplication of a plaintext and a constant $s_1 \cdot K \mod n$ can be calculated by decrypting ciphertext to the power of the constant $(D(E(s_1)^K \mod n^2))$.

As discussed in Section 2.2 the dose calculation represents a linear combination and hence partial (additive) homomorphic cryptosystem is sufficient for this purpose.

3.2 Private Information Retrieval

Private Information Retrieval (PIR) [9,11,19,25] is a two-party cryptographic protocol that allows a user to retrieve an item from a database, without revealing any information about the retrieved item to the database (however, with no guarantee for the confidentiality of the database). Note that in our case, the database represents the patient's input data stored in the client device and the PIR user is the server. [11] proves that the simplest PIR method is to send the whole database to the user. While this method provides information theoretic security, it is inefficient due to high communication overhead.

Chor et al. propose PIR with information theoretic security [11], which assumes multiple copies of the database are stored in different servers, and that the database is a binary string of length n. The proposed mechanism uses two servers and has communication complexity of $O(n^{\frac{1}{3}})$. In [25], Kushilevitz and Ostrovsky propose PIR scheme using a single database based on the intractability of quadratic residuosity problem. Hence, unlike the previous PIR scheme, this scheme only provides computational security. The scheme's communication complexity is $2^{O(\sqrt{log(n) \cdot log\ log(n)})}$, which is less than $O(n^{\epsilon})$ for any $\epsilon > 0$. Cachin et al [9] propose an improvement to Kushilevitz and Ostrovsky's scheme for a single-database PIR which is based on ϕ-hiding assumption. Given ϕ is Euler's totient function, ϕ-hiding assumption refers to the intractability to decide whether a given small prime divides $\phi(m)$, where m is a composite integer of unknown factorization. Note that computing $\phi(m)$ is as hard as factoring m. The communication complexity for the scheme is poly-logarithmic ($O(log^8(n))$, given the security assumption).

In [19], Gentry and Ramzan propose an efficient single database Private Block Retrieval (PBR), an extension of PIR where a user can retrieve a block of data of size d bits, using smooth subgroups (i.e., subgroups that have many small primes dividing their order). The proposed scheme divides the database to blocks of data of size l and associates each block of data with a small prime number. A query generated by the user is essentially a description of a cyclic group G where the order is divisible by the small primes. Then the database is encoded using Chinese Remainder Theorem, that is, given C_i is a block of data and π_i is the corresponding small prime number, the database can be encoded by $e \equiv C_i mod(\pi_i)$ for all i. After receiving $g^e \in G$ (where g is a random element of G), the user can recover the block that it wants by computing a discrete logarithmic computation using the Pohlig-Hellman method [31]. For retrieving multiple blocks, the database is permuted every time the user retrieves a new block of data. The communication complexity of this scheme is $O(k + d)$, where k refers to the security parameter ($k \geq \log n$) and d refers to the size of the retrieved block. Despite the low communication complexity, the scheme's computation complexity is $\mathit{\Theta}(n)$.

The scheme proposed by Aguilar-Melchor and Gaborit in [1] aims to lower the computational complexity of single database PIR. They propose a scheme based on linear algebra techniques using lattices, where the security is based on the hardness of the differential hidden lattice problem. Compared to Gentry and Ramzan scheme, the scheme by Aguilar-Melchor and Gaborit has lower computational complexity, since it does not rely on modular multiplication operations, yet its communication complexity is slightly higher [1]. However, as stated in [28], the scheme is still relatively new and its security is not as well understood as the PIR scheme, which is based on number theory.

4 Secure Evaluation of Warfarin Dose

The proposed secure evaluation protocol comprises three main steps: (a) the server securely selects the patient's data, used for the secure evaluation; (b) the server processes the (encrypted) patient's genomic variation data; and (c) the client performs secure computation. There are also some additional preliminary steps for both client and server, as described below. Figure 2 shows an overview of the protocol, while the full details of the protocol are presented in Algorithm 1. We discuss the steps in detail below.

In the protocol description, we denote by $E_s(.)$ and $D_s(.)$, respectively, the encryption and decryption operations using the server's keys and by $E_c(.)$ and $D_c(.)$ the corresponding operations using the client's encryption keys. We use the notations defined in Section 2; we remind the reader that the client has a total of n data records (including both genomic and generic personal data) and that the server has m test-specific coefficients, that need to be combined with client's data to calculate the personalized medicine algorithm, e.g., the Warfarin dosing formula.

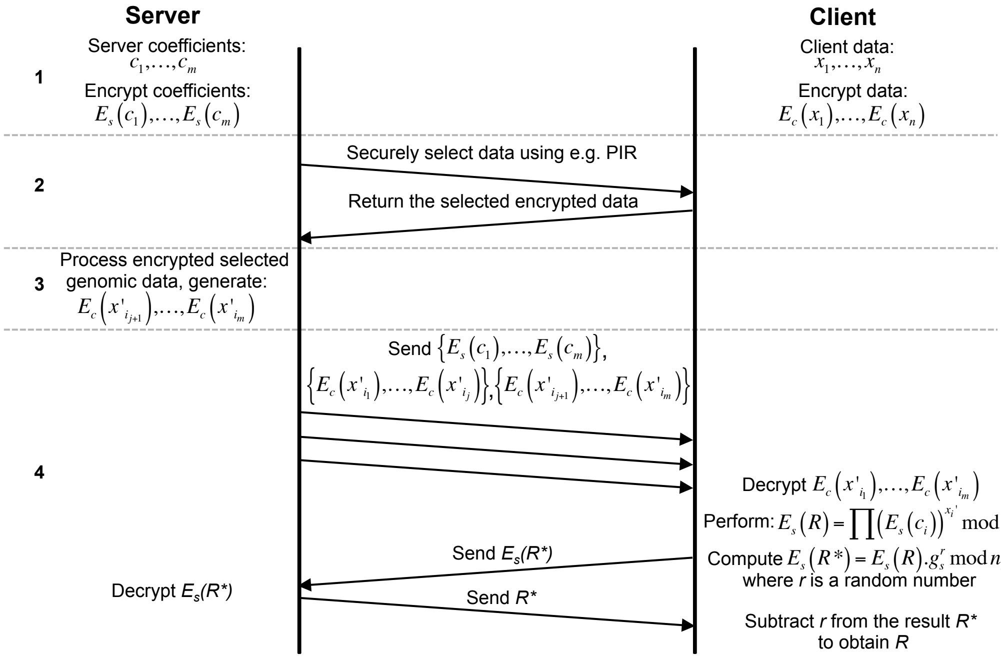

Server
Client
Server coefficients:
$c_1,\dots,c_m$
Encrypt coefficients:
$E_s(c_1),\dots,E_s(c_m)$
Client data:
$x_1,\dots,x_n$
Encrypt data:
$E_c(x_1),\dots,E_c(x_n)$
1
2
Securely select data using e.g. PIR
Return the selected encrypted data
3
Process encrypted selected genomic data, generate:
$E_c(x'_{i_{j+1}}),\dots,E_c(x'_{i_m})$
4
Send $\{E_s(c_1),\dots,E_s(c_m)\}$,
$\{E_c(x'_{i_1}),\dots,E_c(x'_{i_j})\},\{E_c(x'_{i_{j+1}}),\dots,E_c(x'_{i_m})\}$
Decrypt $E_c(x'_{i_1}),\dots,E_c(x'_{i_m})$
Perform: $E_s(R)=\prod(E_s(c_i))^{x_i'} \bmod n_s^2$
Compute $E_s(R^*)=E_s(R).g_s^r \bmod n_s^2$,
where r is a random number
Send $E_s(R^*)$
Decrypt $E_s(R^*)$
Send R^*
Subtract r from the result R^* to obtain R

Fig. 2. Secure evaluation protocol for Warfarin dose calculation

4.1 Preliminary Operations

We assume that before the protocol is invoked, the client and the server have generated their own set of Paillier encryption keys and have exchanged their public keys. The client also needs to convert the genome data format (in the VCF file), to include numerical values in place of characters that denote specific genetic variations (A, G, C, T, N/A). We map each character to a 4-bit value, that is further converted to an integer, in order to facilitate secure computing (we omit details of this mapping due to space constraints).

In the initial step (steps 1 (a) and (b) in Algorithm 1), the client encrypts all of its data and the server encrypts its coefficients with their respective public keys.

4.2 Secure Selection

The server first securely selects a subset of client data (step 2 in Algorithm 1), that will be used for computing. This step needs to comply with the confidentiality constraint that the client should not learn which data is required for the test (we note that the requirement that the server should not learn client's data is ensured by having all client data encrypted). The secure selection step can be performed using e.g., PIR. The outcome of this step is the server having a subset of encrypted client data that includes both genomic and other information (see Figure 2).

4.3 Processing of Encrypted Genetic Variation Information

The patient's selected genomic data need to be securely processed before the secure computation. This processing step is required to determine whether the patient has certain genetic variations, as per the Warfarin dose formula (see Table 1). For example, "VKORC1 A/G" in the Warfarin dosing algorithm is a binary indicator of whether the patient's VKORC1 SNP (rs9923231) has the A/G variant. To perform this step, the server tests the selected SNPs for a match with a list of variants (e.g., for VKORC1 SNP, the variants are A/A, A/G, G/G and unknown). As the Paillier scheme is used to encrypt the input data, we can use subtraction to accomplish this task. I.e., the SNP variant x represented as an integer (e.g., A/A = 1, A/G = 2, G/G = 3, unknown = 0) can be tested against a value t (i.e., a variant of the SNP) to derive

```
Algorithm 1: Secure Evaluation of Warfarin Dose
  Input:
    $(n_c, g_c)$ – Client's public Paillier key, known to all parties
    $(\lambda_c, \mu_c)$ – Client's private Paillier key, known only to the client
    $(n_s, g_s)$ – Server's public Paillier key, known to all parties
    $(\lambda_s, \mu_s)$ – Server's private Paillier key, known only to the server
    $x_1, \ldots, x_n$ – Client's input, which consist of genomic and non-genomic data
    $c_1, \ldots, c_m$ – Server's input, the coefficients
  Output:
    Client learns the linear combination result, server learns nothing
1 (a) Client: encrypts all the genomic and non-genomic data using its public key $E_c(x_1), \ldots, E_c(x_n)$.
  (b) Server: encrypts the coefficients using its public key $E_s(c_1), \ldots, E_s(c_m)$.
2 Server: securely selects $l$ client's input data using e.g., PIR. (The server receives the selected encrypted data
  $E_c(x_{i_1}), \ldots, E_c(x_{i_l})$ from the client.)
3 Server: processes encrypted selected genetic variation data. This generates $E_c(x'_{i_{j+1}}), \ldots, E_c(x'_{i_m})$.
4 (a) Server: sends the encrypted coefficients $E_s(c_1), \ldots, E_s(c_m)$, the encrypted non-genomic data
  $E_c(x'_{i_1}), \ldots, E_c(x'_{i_j})$ (where $E_c(x'_{i_1}) = E_c(x_{i_1})$) and the encrypted genetic variation data
  $E_c(x'_{i_{j+1}}), \ldots, E_c(x'_{i_m})$.
  (b) Client: decrypts $E_c(x'_{i_1}), \ldots, E_c(x'_{i_m})$. For the genetic variation data (i.e., $E_c(x'_{i_{j+1}}), \ldots, E_c(x'_{i_m})$),
  convert zero values to 1s and non-zero values to 0s.
  (c) Client: calculates $E_s(R) = \prod(E_s(c_i))^{x'_i} \mod n_s^2$, which is effectively the encryption of $R = \sum c_i \cdot x'_i$.
  (d) Client: calculates $E_s(R^*) = E_s(R) \cdot g_s^r \mod n_s^2$ ($r \in \mathbb{Z}_{n_s}$ is a random number) and sends it to the server.
  This effectively changes the encrypted value to $R + r$.
  (e) Server: decrypts the ciphertext to obtain the plaintext $R^* = r + \sum c_i \cdot x'_i$, which it then sends to the
  client.
  (f) Client: subtracts $r(\mod n_s)$ from $R^*$ to obtain the desired outcome $R$.
```

the comparison result x'. Given that $x' = x - t$, then $x' = 0$ when $x = t$ and $x' \neq 0$ if $x \neq t$. Hence, the SNP has a variant t if the result is 0, other values indicate the absence of this variant.

Some elements in the dosing algorithm need to test the genetic variations of multiple SNPs. For example in Table 1, the elements which have CYP2C9 need to test the variants of two SNPs, i.e., rs1057910 (for CYP2C9 *1 and CYP2C9 *3) and rs1799853 (for CYP2C9 *2). For this, we need to combine the results of the variant testing, which can be performed by adding the comparison results. If all the SNPs tested have the variants, then the combined result will also be 0.

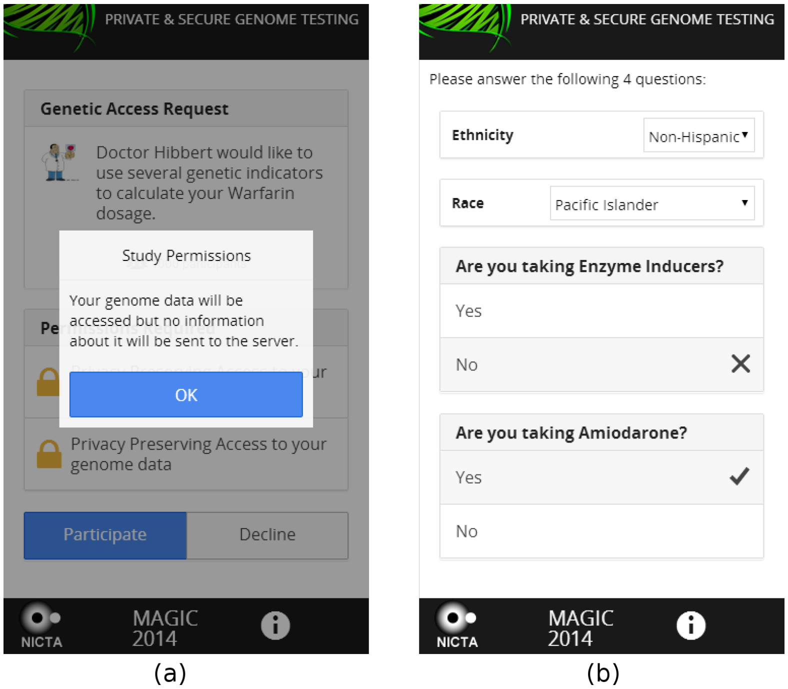
4.4 Secure Computation

Once the processing step is finalized the server sends to the client its encrypted coefficients, the pre-processed encrypted genetic variation data (encrypted using the client's public key) and the remainder of the client's data (i.e., non-genomic data) that did not require processing and is sent with no modifications (step 4(a) in Algorithm 1). Since Paillier encryption does not support multiplication of two ciphertexts, the client needs to decrypt the genetic variation data and the non-genomic data. Furthermore, for the genetic variation data, the client converts zero values to 1s and non-zero values to 0s for use in secure computation (please note the genetic variation result is 0 when a SNP matches a certain variant).

The client then performs the secure computation of the linear combination of the encrypted server's coefficients and the decrypted genetic variation and non-genomic data. Because the server's coefficients are encrypted with the server's public key, the client cannot directly decrypt the linear combination result. Hence, the client selects a random number and adds it to the encrypted linear combination result, and subsequently forwards this (encrypted) sum to the server, as per Figure 2. After receiving the modified linear combination result, the server decrypts the modified result and sends it to the client. (Note that the server cannot determine the secure computation result, as it does not know the value of the random number.) Finally the client subtracts the random number from the modified result to obtain the linear combination result.

5 Prototype Implementation

We have implemented a prototype system for Warfarin dosing and other personalized medicine tests, where the client side takes the form of an Android app (Figure 3). We note that our implementation only includes steps 4 (c) - (f) of Algorithm 1 (i.e., it does not yet implement PIR, or server genome data processing), as a first step towards a complete solution, and to enable us to characterise the performance impact of using Paillier encryption. The client is written mostly in HTML 5 and Javascript with Apache Cordova (for easy cross-platform compatibility). The server side of this part of the protocol is straightforward, and is written in Python, using a Paillier encryption library that we are planning to open source.


PRIVATE & SECURE GENOME TESTING
NICTA
MAGIC 2014
OK
Participate
Decline
PRIVATE & SECURE GENOME TESTING
NICTA
MAGIC 2014

Genetic Access Request

Doctor Hibbert would like to use several genetic indicators to calculate your Warfarin dosage.

Study Permissions

Your genome data will be accessed but no information about it will be sent to the server.

Pe

ur

Privacy Preserving Access to your genome data

(a)

Please answer the following 4 questions:

Ethnicity: Non-Hispanic

Race: Pacific Islander

Are you taking Enzyme Inducers?
Yes
No ✗

Are you taking Amiodarone?
Yes ✓
No

(b)

Fig. 3. Screenshots from our prototype Android app: (a) explaining the privacy implications; (b) requesting test-specific personal details.

We have performed an initial evaluation of selected performance characteristics on the Nexus 5 phone, that has a 2.26GHz CPU and 2GB RAM, with the server running on a dual core 2.36GHz Xeon processor machine with 32GB RAM. We use a 1,024 bit public key for Paillier encryption and the tests were done using our company's Wi-Fi network. For the calculation in Table 1, step 4 (c) involves modular exponentiations of 17 2,048 bit numbers, followed by the modular product of 18 2,048 bit numbers. In our Javascript implementation, step 4 (c) takes around 50 ms on a Nexus 5. Steps 4 (d) - 4 (f) take around 150 ms over an internal Wi-Fi network. We expect that the network usage is latency-limited, since the transmitted data is small (at core, a 2,048 bit number in each direction). Even adding 100 ms expected delay from mobile networks [32], the total duration of 0.3 s for steps 4 (c) - (f) is significantly less than benchmark values for acceptable web browsing [22], where 1.8 s is considered fast, 4.3 s is considered moderate, and 6.8 s is considered slow.

6 Discussion

We first discuss the practicality of the proposed secure evaluation protocol, for the client's secure computation and the server's secure selection mechanisms.

The communication overhead for the secure computation protocol described in Section 4 is relatively small. As shown in Section 5, for a 1,024 bit public key, each encrypted value is 2,048 bits long. For the 18 coefficients in the Warfarin test, this means a message size of 36,864 bits (around 4 kB). This algorithm is also well within the capabilities (CPU, memory) of the Android based Nexus 5 phone used for the prototype.

The PIR based solution was chosen based on the potential for a lower communication overhead, compared to the naive and k-out-of-n schemes discussed in Section 3. Considering that the current number of known SNPs is 62.676 million and that the size of ciphertext for each coefficient is 2,048 bits, both the naive and the k-out-of-n based solution that have a communication complexity of $O(n)$, would require transfer of around 15GB of data between the client and server, for the secure selection. The PIR scheme proposed in [19] has a communication overhead of $O(k + d)$ for one block of data, where k is a security parameter and d is the number of retrieved bits. If we adopt k=1,024 (as per [1]) and d as $11 \cdot 2,048 = 22,528$ (note there are 11 client's data records used in the Warfarin dose calculation), and assuming that we retrieve a large block of data, the communication overhead estimate is a very low 2.875kB. The computation complexity of this scheme however, has not been verified in an implementation up to now, although there is an implementation of the scheme proposed in [1].

We also discuss the security of our proposal. A completely secure protocol would reveal nothing but the personalized medicine dose. This would have been the case if we computed the Warfarin formula with all the genome data and without the secure selection process. In our case, the client learns that certain generic information (age, weight etc. values may be easily recognizable as such) is used in the test. However, most genomic tests use those values (we note that the values also need to be provided to the client app), therefore this is not considered as a threat. The client also learns the values of genetic variations, however these will still not allow the patient to learn the coefficient (all decrypted values for variations in step 4 (a) of Algorithm 1 are binary), but the client may discern, e.g., how many of the coefficients play an active (non-zero) role in the computation. We also note that client can learn coefficients by running the protocol with different inputs, but that would be possible even with an ideal (trusted third party) solution, and can be mitigated by limiting the number of executions of the protocol (e.g., the protocol can run only with doctor's authorization).

7 Related Work

Privacy in Genome Wide-Association Study (GWAS) is one of the important topics in genome privacy research. GWAS is an approach to find genetic variations associated with particular diseases.[7] It is useful in finding genetic variations that contribute to common, complex diseases. Homer et al. [21] and Wang et al. [34] demonstrated the privacy risk of releasing GWAS results, even in aggregated form. Consequently, several works [17, 23] proposed methods to protect the privacy of individuals in GWAS results using differential privacy. In particular, Feinberg et al. [17] presented methods for releasing differentially private minor allele frequencies, χ-square statistics and p-value. Johnson and Shmatikov [23] proposed a set of practical privacy preserving algorithms for GWAS datasets, which support exploratory data analysis (i.e., when SNPs are not known a priori). In contrast, Kamm et al. [24] proposed a mechanism for GWAS computation using secure multi-party computation (MPC). To enable MPC, the work assumes that the GWAS datasets are stored in distributed storage using secret sharing technique.

Privacy preserving pattern matching is a commonly used method for private genomic testing [8,15,18,33]. In this case, an entity (e.g., a patient) has a digitized genome and another entity (a medical entity like a physician, a hospital or a pharmaceutical company) has DNA markers (i.e., substrings) that may or may not be present in the digitized genome. The goal is for the patient to learn whether the DNA markers are present in the genome, while ensuring that the DNA markers are not revealed to the patient and that the medical entity learns nothing about the patient's genome. Several privacy-preserving pattern matching schemes based on oblivious automata have been proposed [8, 18, 33]. Furthermore, De Cristofaro et al. [15] proposed a pattern matching scheme that hides the size and the position of the DNA markers in the genome.

Another common method for genomic testing is sequence comparison [7,14,16,35]. These works considered the scenario where two entities want to determine whether two genomes are closely related. Wang et al. [35] proposed a distributed framework for privacy preserving sequence comparison using program specialization, which partitions a genomic computation according to the sensitivity level of the genome data. Eppstein et al. [16] studied a method to improve the efficiency of privacy preserving compressed DNA sequence comparison, and proposed a privacy-enhanced invertible bloom filter to compute set difference. Baldi et

[7] "What is a genome-wide association study?", at `https://www.genome.gov/20019523#gwas-1`

al. [7] presented privacy preserving techniques for paternity tests, personalized medicine and genetic tests based on private set operations. The techniques were implemented as Genodroid for Android phone [14].

Ayday et al. [3, 4] stated that pattern matching and sequence comparison are not sufficient for genomic testing. Their work computed a disease susceptibility test using weighted average. The privacy preserving techniques relied on a modified Paillier cryptosystem and on proxy re-encryption. Furthermore, they proposed a Storage and Processing Unit (SPU), which stores the encrypted patients' genomes and perform the computation. The scheme in [5] performed the same test, while including clinical and environmental data of the patients besides their genome data. Note that these works involve secure computation of weighted average, which is similar to our proposal. The main difference is in the architecture, where in our case the majority of the secure computation is performed on the patient's device, while in these works the computation is performed on the SPU.

Finally, Ayday et al. [6] proposed a privacy preserving system for the storage, retrieval and processing of raw genomic data. The system allows a medical entity to privately retrieve a subset of the genome (short reads) without revealing the test to the genome data centre. In addition, Chen et al. [10] presented a privacy preserving mapping technique for DNA sequence analysis, where the computation is outsourced to hybrid clouds (i.e., a combination of private and public clouds).

8 Conclusion

This paper presents a protocol for secure computations in personalized medicine. The scheme is based on a combination of partially homomorphic Paillier cryptosystem, PIR and additional steps that enable a client mobile device with personal (including genome) data to compute tests (we use the Warfarin dosing example) in a way that preserves the privacy of personal data and the details of the test. We present details of a partial implementation of this protocol. Future work includes full protocol implementation and evaluation of user acceptability of the system.

References

1. Aguilar-Melchor, C., Gaborit, P.: A lattice-based computationally-efficient private information retrieval protocol. In: Western European Workshop on Research in Cryptology. (2007)
2. Ayday, E., Cristofaro, E.D., Hubaux, J.P., Tsudik, G.: The chills and thrills of whole genome sequencing. CoRR **abs/1306.1264** (2013)
3. Ayday, E., Raisaro, J.L., Hubaux, J.P.: Personal Use of the Genomic Data: Privacy vs. storage Cost. In: IEEE Global Communications Conference, Exhibition and Industry Forum GLOBECOM. (2013)
4. Ayday, E., Raisaro, J.L., Hubaux, J.P., Rougemont, J.: Protecting and evaluating genomic privacy in medical tests and personalized medicine. In: Proceedings of the 12th ACM Workshop on Workshop on Privacy in the Electronic Society, New York, NY, USA, ACM (2013) 95–106
5. Ayday, E., Raisaro, J.L., McLaren, P.J., Fellay, J., Hubaux, J.P.: Privacy-preserving computation of disease risk by using genomic, clinical, and environmental data. In: Presented as part of the 2013 USENIX Workshop on Health Information Technologies, Washington, D.C., USENIX (2013)
6. Ayday, E., Raisaro, J., Hengartner, U., Molyneaux, A., Hubaux, J.P.: Privacy-preserving processing of raw genomic data. In: Data Privacy Management and Autonomous Spontaneous Security. Springer Berlin Heidelberg (2014) 133–147
7. Baldi, P., Baronio, R., De Cristofaro, E., Gasti, P., Tsudik, G.: Countering gattaca: Efficient and secure testing of fully-sequenced human genomes. In: Proceedings of the 18th ACM Conference on Computer and Communications Security, New York, NY, USA, ACM (2011) 691–702
8. Blanton, M., Aliasgari, M.: Secure outsourcing of dna searching via finite automata. In: Proceedings of the 24th Annual IFIP WG 11.3 Working Conference on Data and Applications Security and Privacy, Berlin, Heidelberg, Springer-Verlag (2010) 49–64
9. Cachin, C., Micali, S., Stadler, M.: Computationally private information retrieval with polylogarithmic communication. In: Advances in Cryptology EUROCRYPT 99. Volume 1592. Springer Berlin Heidelberg (1999) 402–414
10. Chen, Y., Peng, B., Wang, X., Tang, H.: Large-scale privacy-preserving mapping of human genomic sequences on hybrid clouds. In: Proceeding of the 19th network & distributed system security symposium. (2012)
11. Chor, B., Kushilevitz, E., Goldreich, O., Sudan, M.: Private information retrieval. J. ACM **45**(6) (November 1998) 965–981

12. Chu, C.K., Tzeng, W.G.: Efficient k-out-of-n oblivious transfer schemes. Journal of Universal Computer Science **14**(3) (feb 2008) 397–415
13. Consortium, I.W.P., et al.: Estimation of the warfarin dose with clinical and pharmacogenetic data. New England Journal of Medicine **360**(8) (2009) 753–764 PMID: 19228618.
14. De Cristofaro, E., Faber, S., Gasti, P., Tsudik, G.: Genodroid: Are privacy-preserving genomic tests ready for prime time? In: Proceedings of the 2012 ACM Workshop on Privacy in the Electronic Society, New York, NY, USA, ACM (2012) 97–108
15. De Cristofaro, E., Faber, S., Tsudik, G.: Secure genomic testing with size- and position-hiding private substring matching. In: Proceedings of the 12th ACM Workshop on Workshop on Privacy in the Electronic Society, New York, NY, USA, ACM (2013) 107–118
16. Eppstein, D., Goodrich, M.T., Baldi, P.: Privacy-enhanced methods for comparing compressed dna sequences. CoRR **abs/1107.3593** (2011)
17. Fienberg, S., Slavkovic, A., Uhler, C.: Privacy preserving GWAS data sharing. In: IEEE 11th International Conference on Data Mining Workshops. (Dec 2011) 628–635
18. Frikken, K.: Practical private dna string searching and matching through efficient oblivious automata evaluation. In: Data and Applications Security XXIII. Volume 5645. Springer Berlin Heidelberg (2009) 81–94
19. Gentry, C., Ramzan, Z.: Single-database private information retrieval with constant communication rate. In: Automata, Languages and Programming. Volume 3580. Springer Berlin Heidelberg (2005) 803–815
20. Gymrek, M., McGuire, A.L., Golan, D., Halperin, E., Erlich, Y.: Identifying personal genomes by surname inference. Science **339**(6117) (2013) 321–324
21. Homer, N., Szelinger, S., Redman, M., Duggan, D., Tembe, W., Muehling, J., Pearson, J.V., Stephan, D.A., Nelson, S.F., Craig, D.W.: Resolving individuals contributing trace amounts of dna to highly complex mixtures using high-density snp genotyping microarrays. PLoS genetics **4**(8) (2008) e1000167
22. Ibarrola, E., Liberal, F., Taboada, I., Ortega, R.: Web qoe evaluation in multi-agent networks: Validation of itu-t g.1030. In: ICAS, IEEE Computer Society (2009) 289–294
23. Johnson, A., Shmatikov, V.: Privacy-preserving data exploration in genome-wide association studies. In: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, ACM (2013) 1079–1087
24. Kamm, L., Bogdanov, D., Laur, S., Vilo, J.: A new way to protect privacy in large-scale genome-wide association studies. Bioinformatics **29**(7) (April 2013) 886–893
25. Kushilevitz, E., Ostrovsky, R.: Replication is not needed: Single database, computationally-private information retrieval. In: Proceedings of the 38th Annual Symposium on Foundations of Computer Science, Washington, DC, USA, IEEE Computer Society (1997) 364–
26. Mun, M., Hao, S., Mishra, N., Shilton, K., Burke, J., Estrin, D., Hansen, M., Govindan, R.: Personal data vaults: A locus of control for personal data streams. In: Proceedings of the 6th International Conference on emerging Networking Experiments and Technologies. (2010)
27. Naor, M., Pinkas, B.: Oblivious transfer and polynomial evaluation. In: Proceedings of the Thirty-first Annual ACM Symposium on Theory of Computing, New York, NY, USA, ACM (1999) 245–254
28. Olumofin, F., Goldberg, I.: Revisiting the computational practicality of private information retrieval. In: Financial Cryptography and Data Security. Volume 7035. Springer Berlin Heidelberg (2012) 158–172
29. Ostrovsky, R., Skeith, WilliamE., I.: A survey of single-database private information retrieval: Techniques and applications. In: Public Key Cryptography PKC 2007. Volume 4450. Springer Berlin Heidelberg (2007) 393–411
30. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Advances in Cryptology – EUROCRYPT 1999, Springer-Verlag (1999) 223–238
31. Pohlig, S., Hellman, M.: An improved algorithm for computing logarithms over gf(p) and its cryptographic significance (corresp.). Information Theory, IEEE Transactions on **24**(1) (Jan 1978) 106–110
32. Romirer-Maierhofer, P., Ricciato, F., D'Alconzo, A., Franzan, R., Karner, W.: Network-wide measurements of tcp rtt in 3g. In: Proceedings of the First International Workshop on Traffic Monitoring and Analysis. (2009)
33. Troncoso-Pastoriza, J.R., Katzenbeisser, S., Celik, M.: Privacy preserving error resilient dna searching through oblivious automata. In: Proceedings of the 14th ACM Conference on Computer and Communications Security, New York, NY, USA, ACM (2007) 519–528
34. Wang, R., Li, Y.F., Wang, X., Tang, H., Zhou, X.: Learning your identity and disease from research papers: Information leaks in genome wide association study. In: Proceedings of the 16th ACM Conference on Computer and Communications Security, New York, NY, USA, ACM (2009) 534–544
35. Wang, R., Wang, X., Li, Z., Tang, H., Reiter, M.K., Dong, Z.: Privacy-preserving genomic computation through program specialization. In: Proceedings of the 16th ACM Conference on Computer and Communications Security, New York, NY, USA, ACM (2009) 338–347