Countering GATTACA*: Efficient and Secure Testing of Fully-Sequenced Human Genomes (Full Version)[†]

Pierre Baldi[†,‡] Roberta Baronio[†] Emiliano De Cristofaro[‡] Paolo Gasti[‡] Gene Tsudik[‡]

[†] Institute for Genomics and Bioinformatics [‡] Department of Computer Science
{pfbaldi,rbaronio,edecrist,pgasti,gts}@uci.edu
University of California, Irvine

ABSTRACT

Recent advances in DNA sequencing technologies have put ubiquitous availability of fully sequenced human genomes within reach. It is no longer hard to imagine the day when everyone will have the means to obtain and store one's own DNA sequence. Widespread and affordable availability of fully sequenced genomes immediately opens up important opportunities in a number of health-related fields. In particular, common genomic applications and tests performed *in vitro* today will soon be conducted computationally, using digitized genomes. New applications will be developed as genome-enabled medicine becomes increasingly preventive and personalized. However, this progress also prompts significant privacy challenges associated with potential loss, theft, or misuse of genomic data. In this paper, we begin to address genomic privacy by focusing on three important applications: *Paternity Tests*, *Personalized Medicine*, and *Genetic Compatibility Tests*. After carefully analyzing these applications and their privacy requirements, we propose a set of efficient techniques based on *private set operations*. This allows us to implement in *in silico* some operations that are currently performed via *in vitro* methods, in a secure fashion. Experimental results demonstrate that proposed techniques are both feasible and practical today.

1. INTRODUCTION

Over the past four decades, DNA sequencing has been one of the major driving forces in life-sciences, producing full genome sequences of thousands of viruses and bacteria, and dozens of eukaryotic organisms, from yeast to man (e.g., [32, 2, 81, 43]). This trend is only being accentuated by modern High-Throughput Sequencing (HTS) technologies: the first diploid human genome sequences were recently produced [55, 83, 78] and a project to sequence 1,000 human genomes has been essentially completed [46, 70, 21]. Different HTS technologies are competing to sequence an individual human genome — composed of about 3 billion DNA nucleotides (or bases) — for less than $1,000 by 2012 [69], and even less than $100 five years later, reaching the point where human genome sequencing will be a commodity costing less than an X-ray or an MRI scan. Ubiquity of human and other genomes creates enormous opportunities and challenges. In particular, it promises to address one of the greatest societal challenges of our time: the unsustainable rise of health care costs, by ushering a new era of genome-enabled predictive, preventive, participatory, and personalized medicine ("P4" medicine). In time, genomes could become part of the *Electronic Medical Record* of every individual [40].

However, widespread availability of HTS technologies and genomic data exacerbates ethical, security, and privacy concerns [12]. A full genome sequence not only uniquely identifies each one of us; it also contains information about, for instance, our ethnic heritage, disease predispositions, and many other phenotypic traits [25, 68]. Traditional approaches to privacy, such as de-identification or aggregation [57, 41], become completely moot in the genomic era, since the genome itself is the ultimate identifier. To further compound the privacy problem, health information is increasingly shared electronically among insurance companies, health care providers and employers. This, coupled with the possibility of creating large centralized genome repositories, raises the specter of possible abuses.

Some federal laws have been passed to begin addressing privacy issues. The 2003 Health Insurance Portability and Accountability Act (HIPAA) provides a general framework for protecting and sharing Protected Health Information (PHI) [22, 52, 58]. In 2008, the Genetic Information Nondiscrimination Act (GINA) was adopted to prohibit discrimination on the basis of genetic information. with respect to health insurance and employment [77]. While providing general guidelines and a basic safety net, current legislation does not offer detailed technical information about safe and privacy-preserving ways for storing and querying genomes. In short, technical issues of security and privacy for HTS and genomic data remain both important and relatively poorly understood.

While privacy issues are not yet hampering progress in basic genomic research, it is not too early to start investigating them, particularly, in light of their complexity, potential impact on society, and current efforts to reform the health care system. It remains unclear where personal genomic information will be stored, who will have access to it, and how it will be queried and shared. To remain flexible, we can imagine a general framework comprised of two kinds of basic entities: (1) Data Centers where genomic data is stored, and (2) Agents/Agencies interested in querying this data. Granularity of Data Centers could vary. At one end of the spectrum, every individual could be her own Data Center and store the genome on a personal computer, cell phone, or some other device. At the other extreme, we could envision national or even international Data Centers storing

*See http://www.imdb.com/title/tt0119177/.

[†] A preliminary version of this paper appeared in ACM CCS 2011. The present version supercedes it. New material includes new techniques for privacy-preserving paternity testing in Section 4.1.3.

millions (or even billions) of genomic sequences. Data Centers could also be envisioned at the granularity of family, school, pharmacy, laboratory, hospital, city, county or state. Likewise, many different types of Agents/Agencies are conceivable, ranging from individuals and personal physicians, to family members, pharmacies, hospitals, insurance companies, employers and government agencies (e.g., the FBI), or international organizations. Various Agents/Agencies might be allowed to query different aspects of genomic data and might be required to satisfy different query privacy requirements. In addition, one could imagine cases (e.g., criminal search or proprietary diagnostic technology) where both the genomic data and queries against it must remain private.

The main security and privacy challenge is how to support such queries with low storage costs and reasonably short query times, while satisfying privacy and security requirements associated with a given type of transaction. Unfortunately, current methods for privacy-preserving data querying do not scale to genomic data sizes. Several cryptographic techniques have been proposed that — though not addressing the case of fully-sequenced genomes — focus on private computation over genomic fragments. Specifically, they allow two or more parties to engage in protocols that reveal only the end-result of a given computation on their respective genomic data, without leaking any additional information. The main thrust of this paper is to adapt and deploy efficient cryptographic techniques to address specific genomic queries and applications, described below.

1.1 Applications

As mentioned above, availability of affordable full genome sequencing makes it increasingly possible to query and test genomic information not only *in vitro*, but also *in silico* using computational techniques. We consider three concrete examples of such tests and corresponding privacy-relevant scenarios.

Paternity Tests establish whether a male individual is the biological father of another individual, using genetic fingerprinting. Advances in biotechnology facilitated DNA paternity tests and stimulated the creation of hundreds of online companies offering testing via self-administered cheek swabs for as little as $79 (e.g., `http://www.gtldna.net`). However, this practice raises several security and privacy concerns: the testing company must be trusted with privacy and accuracy of test results, as well as with swabs that might yield full genome sequencing. We believe that, ideally, any two individuals, in possession of their genomes should be able to conduct a privacy-preserving paternity test with no involvement of any third parties. Only the outcome of the test ought to be learned by one or both parties and no other sensitive genomic information should be disclosed.

Personalized Medicine is recognized as a significant paradigm shift and a major trend in health care, moving us closer to a more precise, powerful, and holistic type of medicine [82]. With personalize medicine, treatment and medication type/dosage would be tailored to the precise genetic makeup of individual patient. For example, measurements of *erbB2* protein in breast, lung, or colorectal cancer patients are taken before selecting proper treatment. It has been showed that the trastuzumab monoclonal antibody is effective only in patients whose genetic receptor is over-expressed [67]. Furthermore, the FDA has recently recommended testing for the thiopurine S-methyltransferase (*tpmt*) gene, prior to prescribing for 6-mercaptopurine and azathioprine — two drugs used for treating childhood leukemia and autoimmune diseases. The *tpmt* gene codes for the TPMT enzyme that metabolizes thiopurine drugs: genetic polymorphisms affecting enzymatic activity are correlated with variations in sensitivity and toxicity response to such drugs. Patients suffering from this genetic disease (1 in 300) only need 6-10% of the standard dose of thiopurine drugs; if treated with the full dose, they risk severe bone marrow suppression and subsequent death [1]. Not surprisingly, experts predict that availability of full genome sequencing will further stimulate development of personalized medicine [31].

Genetic Tests are routinely used for several purposes, such as newborn screening, confirmational diagnostics, as well as pre-symptomatic testing, e.g., predicting Huntington's disease [36] and estimating risks of various types of cancer. We focus on genetic *compatibility* tests, whereby potential or existing partners wish to assess the possibility of transmitting to their children a genetic disease with Mendelian inheritance [59]. Modern genetic testing can accurately predict whether a couple is at risk of conceiving a child with an autosomal recessive disease. Consider, for instance, *Beta-Thalassemia minor*, that causes red cells to be smaller than average, due to a mutation in the *hbb* gene. It is called *minor* when the mutation occurs only in one allele. This *minor* form has no severe impact on a subject's quality of life. However, the *major* variant — that occurs when both alleles carry the mutation — is likely to result in premature death, usually, before age twenty. Therefore, if both partners silently carry the *minor* form, there is a 25% chance that their child could carry the major variety. Another example is the *Lynch Syndrome* (also known as Hereditary Nonpolyposis Colon Cancer), a genetic condition — most commonly inherited from a parent — associated with the high risk of colon cancer [48]. Parents with this syndrome have a 50% chance of passing it on to their children. Since the possibility of inheritance is maximized if both parents carry the mutations, testing for Lynch Syndrome is crucial.

Note on Non-human Genomes: Although this paper focuses on human genomes, some aforementioned scenarios apply to other organisms, e.g., crops and animals [3]. For instance, a paternity test may certify a purebred dog's bloodline or genetic tests may determine the quality of a racing horse. In fact, DNA "barcodes" identifiers are already embedded in genomes of genetically modified species. Conceivably, future veterinary treatments may also involve elements of personalized medicine for animals.

1.2 Roadmap

Motivated by the emerging affordability of full genome sequencing, we combine domain knowledge in biology, genomics, bioinformatics, security, privacy and applied cryptography in order to better understand the corresponding security and privacy challenges. In particular, we analyze specific requirements of three types of applications discussed above: Paternity Tests, Personalized Medicine and Genetic Tests. In the process, we carefully consider today's *in vitro* procedure for each application and analyze its security and privacy requirements in the digital domain. This type of approach allows us to gradually craft specialized protocols that incur appreciably lower overhead than state-of-the-art. However, as is well known, "lower overhead" does not necessarily imply practicality. Therefore, we demonstrate — via experiments on commodity hardware — that proposed protocols are indeed viable and practical *today.* Source code of our implementations is publicly available. We hope that it can help in developing privacy-aware operations on full genomes and allows individuals (in possession of their sequenced genomes) to run genetic tests with privacy.

Organization. We overview related work in the next section. Then, Sec. 3 introduces biological and cryptographic background used throughout the rest of the paper. The core of the paper is in Sec. 4 that includes step-by-step design of protocols for each aforementioned application. It also presents experimental results. Next, Sec. 5 provides security arguments for proposed protocols, followed by the summary and the discussion of future work in Sec. 6.

2. RELATED WORK

As discussed in Section 1, traditional approaches to privacy, such as de-identification, are often ineffective on genomic data, since the genome itself is the ultimate identifier. We refer to [57, 41, 79, 86] for details on privacy risks associated to releasing genomic information, even when aggregated. Motivated by the sensitivity of genomic information, the security research community has begun to develop mechanisms to enable secure computation on genomic data. A number of cryptographic protocols have been proposed for private searching, matching and evaluating similarity of strings, including DNA sequences. Also, prior work has considered specific (privacy-preserving) genomic operations. This section overviews relevant prior results and highlights their potential limitation.

Searching and Matching DNA

Troncoso-Pastoriza, et al. [75] proposed an error-resilient privacy-preserving protocol for string searching. In it, one party (e.g., Alice), with her own DNA snippet, can verify the existence of a short template (e.g., a genetic test held by a service provider – Bob) within her snippet. This technique handles errors and maintains privacy of both the template and the snippet. Each query is represented as an automaton executed using a finite state machine (FSM) in an oblivious manner. Communication complexity is $O(n \cdot (|\Sigma| + |Q|))$, where n is snippet length, $|\Sigma|$ – alphabet size (i.e., 4 for DNA), and $|Q|$ – number of states. Computational complexity is $O(n \cdot |\Sigma| \cdot |Q|)$ and $O(n \cdot |Q|)$ cryptographic operations for Alice and Bob, respectively. However, the number of FSM states is always revealed to all parties. To obtain error-resilient and approximate DNA matching, [75] also shows how to construct an automaton that, given Alice's string x, accepts all strings with Levenshtein distance [54] at most d from x.

Blanton and Aliasgari [4] improve on [75], reducing Alice's work by a factor of $|\Sigma|$ and Bob's — by a factor of $\log(|Q|)$, incurring, however, a potentially increased communication complexity (if the security parameter is smaller than $\log(|Q|)$). This work also introduces a protocol for secure outsourcing of computation to an external service provider and a modified *multi-party* protocol.

A set of cryptographic protocols for secure pattern matching are presented in [29] and [38]. Given a binary string T of length n, held by Alice, and a binary pattern p of length m, held by Bob, pattern matching lets Bob learn all locations in T where p appears. Secure computation guarantees that nothing except m is learned by Alice, and nothing about T is revealed to Bob (besides n and locations where p appears). [29] proposes one such protocol, secure in the semi-honest setting, based on homomorphic encryption, with $O(m+n)$ communication and computation complexities. It includes another protocol, secure in the malicious setting, based on secure oblivious automata evaluation, with quadratic complexity and m rounds. Subsequently, [38] presented an improved protocol, with malicious security, using homomorphic encryption and incurring $O(m+n)$ complexity.

Another related result is the recent work in [50]. It realizes secure computation of the CODIS test [73] (run by the FBI for DNA identity testing), that could not be implemented using pattern matching or FSM. It achieves efficient secure computation of function $M(T, p, e, l) = 1$ iff $|l_{max}(T, p) - l| \leq \varepsilon$, where T is a DNA fragment, p a pattern, (ε, l) some additional information, and $l_{max}(T, p) \geq 0$ is the largest integer l' for which $p^{l'}$ appears as a substring in T. A general technique for secure text processing is introduced, combining garbled circuits and secure pattern matching. (The latter is reduced to private keyword search and solved using Oblivious Pseudorandom Functions (OPRFs) [26, 37].) The resulting protocol can compute several functions (including CODIS) on sample T and pattern p, using the number of circuits linear in the number of occurrences of p. Complexity incurred by the underlying keyword search protocol is linear in $|T|$. However, common knowledge of some threshold on the number of occurrences needs to be assumed.

Similarity of DNA Sequences

Another set of cryptographic results focus on privately computing the *edit distance* of two strings α, β of size m and n, respectively.[1] Privacy-preserving computation of Smith-Waterman scores [71] has also been investigated and used for sequence alignment.

Jha, et al. [45] proposed techniques for secure edit distance using garbled circuits [84], and showed that the overhead is acceptable only for small strings (e.g., a 200-character strings require 2GB circuits). For longer strings, two optimized techniques were proposed; they exploit the structure of the dynamic programming problem (intrinsic to the specific circuit) and split the computation into smaller component circuits. However, a quadratic number of oblivious transfers is needed to evaluate garbled circuits, thus limiting scalability of this approach. For example, 500-character string instances take almost one hour to complete [45]. Optimized protocols also extend to privacy-preserving Smith-Waterman scores [71], a more sophisticated string comparison algorithm, where costs of delete/insert/replace operations, instead of being equal, are determined by special functions. Again, scalability is limited: experiments in [45] show that evaluation of Smith-Waterman for a 60-character string takes about 1,000 seconds.

Somewhat less related techniques include [47] that proposed a cryptographic framework for executing queries on genomic databases where privacy is attained by relying on

[1] Edit distance is the minimum number of operations (delete, insert, or replace) needed to transform α into β.

two anonymizing and non-colluding parties. Danezis, et al. [16] used negative databases to test a single profile against a database of suspects, such that database contents cannot be efficiently enumerated.

Specialized Protocols

Wang, et al. [80] proposed techniques for computation on genomic data stored at a data provider, including: edit distance, Smith-Waterman and search for homologous genes. Program specialization is used to partition genomic data into "public" (most of the genome) and "sensitive" (a very small subset of the genome). Sensitive regions are replaced with symbols by data providers (DPs) before data consumers (DCs) have access to genomic information. DCs perform concrete execution on public data and symbolic execution on sensitive data, and may perform queries to DPs on sensitive nucleotides. However, only queries that do not let DCs reconstruct sensitive regions are allowed by DPs and generic two-party computation techniques are used during query execution. Portions of sensitive data are public information. We note that, due to the current limited knowledge of human genome, parts that are considered non-sensitive today may actually become sensitive later.

Also, Bruekers, et al. [7] presented privacy-preserving techniques for a few DNA operations, such as: identity test, common ancestor and paternity test, based on STR (Short Tandem Repeat; see Sec. 3.1). Homomorphic encryption is used on alleles (fragments of DNA) to compute comparisons. Testing protocols tolerate a small number of errors, however, their complexity increases with the number of tolerated errors [4]. Also, [7] leaves as an open problem the scenario where an attacker (honestly) runs the protocol but executes it on arbitrarily chosen inputs. In this setting, attackers, given STR's limited entropy, can "lie" about their STR profiles and run multiple dependent protocols thus reconstructing the other party's profile.

Using Current Techniques?

We aim to obtain secure and private computation on fully sequenced genomes, in scenarios where individuals possess their own genomic data. As discussed in Sec. 1, we focus on paternity testing, personalized medicine and genetic compatibility testing. Prior work has yielded a number of elegant (if not always efficient) cryptographic protocols for secure computation on DNA sequences. However, we identify some notable open problems:

1. ***Efficiency:*** Most current protocols are designed for DNA snippets (e.g., hundreds of thousands nucleotides) and it is unclear how to scale them to full genomes (i.e., three billion nucleotides).
2. ***Error Resilience:*** Most prior work attempts to achieve resilience to sequencing errors *in computation* (e.g., using approximate matching or distance with errors). Not surprisingly, this results in: (i) significant computation and communication overhead, and (ii) ruling out more efficient and simpler cryptographic tools, i.e., those geared for exact matching. (Whereas, our goal is error-resilience by design.) Also, as the cost of full genome sequencing drops, so do error rates. By increasing the number of sequencing runs, the probability of sequencing errors can be rapidly reduced.
3. ***Inter-String Distance:*** Analyzing the distance between sequenced strings works for the creation of phylogenetic trees, parental analysis, and homology studies. However, it does not suit applications, such as genetic diseases testing, that require much more complex comparisons.
4. ***Paternity Testing:*** To the best of our knowledge, the only available technique for privacy-preserving genetic paternity testing is [7]. However, it does not prevent a participant from manipulating its input to reconstruct the counterpart's profile. Also, as shown in Sec. 4.1, overhead can be significantly reduced using techniques that obtain error resilience by design.
5. ***Genetic Testing via Pattern Matching:*** The use of pattern matching over full genomes to test for genetic compatibility and/or personalized medicine is not straightforward. Suppose that a party wants to privately search for certain gene mutation, e.g., Beta-Thalassemia. The pattern representing this mutation might be very short — a few nucleotides — but needs to be searched in the full genome, as restricting the search to the specific gene would trivially expose the nature of the test. Therefore, naïve application of pattern matching would return all locations (presumably millions) where the pattern appears. This would be detrimental to both privacy and efficiency of the resulting solution. We could modify the pattern to include nucleotides expected to appear immediately before/after the mutation, such that, with high probability, this pattern would appear at most once. However, this needs to be done carefully, since: (i) nucleotides added to the pattern must appear in *all* human genomes, and (ii) the choice of pattern length should not expose the mutation being searched. Plus, extending the pattern would also increase computation and communication overhead.

3. PRELIMINARIES

This section provides some relevant biology and cryptography background information.

3.1 Biology Background

Genomes represent the entirety of an organism's hereditary information. They are encoded either in DNA or, for many types of viruses, in RNA. The genome includes both the genes and the non-coding sequences of the DNA/RNA. For humans and many other organisms, the genome is encoded in double stranded deoxyribonucleic acid (DNA) molecules, consisting of two long and complementary polymer chains of four simple units called nucleotides, represented by the letters A, C, G, and T. The human genome consists of approximately 3 billion letters.

Restriction Fragment Length Polymorphisms (RFLPs) refers to a difference between samples of homologous DNA molecules that come from differing locations of restriction enzyme sites, and to a related laboratory technique by which these segments can be illustrated. In RFLP analysis, a DNA sample is broken into pieces (digested) by restriction enzymes and the resulting restriction fragments are separated according to their lengths by gel electrophoresis. Thus, RFLP provides information about the length (but not the composition) of DNA subsequences occurring between known subsequences recognized by particular enzymes. Although it is being progressively superseded by inexpensive DNA sequencing technologies, RFLP analysis was the first DNA profiling technique inexpensive enough for widespread application. It is still widely used at present. RFLP probes are frequently used in genome mapping and in variation analysis, such genotyping, forensics, paternity tests and hereditary disease diagnostics. (For more details, see [64].)

Single Nucleotide Polymorphisms (SNPs) are the most common form of DNA variation occurring when a single nucleotide (A, C, G, or T) differs between members of the same species or paired chromosomes of an individual [72]. The average SNP frequency in the human genome is approximately 1 per 1,000 nucleotide pairs.[2] SNP variations are often associated with how individuals develop diseases and respond to pathogens, chemicals, drugs, vaccines, and other agents. Thus SNPs are key enablers in realizing *personalized medicine* [10]. Moreover, they are used in genetic disease and disorder testing, as well as to compare genome regions between cohorts in genome-wide association studies.

Short Tandem Repeats (STRs) occur when a pattern of two or more nucleotides are repeated and repeated sequences are directly adjacent to each other. The pattern can range in length from 2 to 50 nucleotides or so. Unrelated people likely have different numbers of repeat units in highly polymorphic regions, hence, STRs are often used to differentiate between individuals. STR *loci* (i.e., locations on a chromosome) are targeted with sequence-specific primers. Resulting DNA fragments are then separated and detected using electrophoresis. By identifying repeats of a specific sequence at specific locations in the genome, it is possible to create a genetic profile of an individual. There are currently over 10,000 published STR sequences in the human genome.

3.2 Cryptography Background

We now overview a set of cryptographic concepts and tools used in the rest of the paper. For ease of exposition, we omit basic notions and refer to [34, 49, 60] for details on various cryptographic primitives, such as hash functions, number-theoretic assumptions, as well as encryption and signature schemes.

Private Set Intersection (PSI) [27]: a protocol between Server with input $\mathcal{S} = \{s_1, \ldots, s_w\}$, and Client with input $\mathcal{C} = \{c_1, \ldots, c_v\}$. At the end, Client learns $\mathcal{S} \cap \mathcal{C}$. PSI securely implements: $\mathcal{F}_{\mathsf{PSI}} : (\mathcal{S}, \mathcal{C}) \mapsto (\bot, \mathcal{S} \cap \mathcal{C})$.

Private Set Intersection Cardinality (PSI-CA) [27]: a protocol between Server with input $\mathcal{S} = \{s_1, \ldots, s_w\}$, and Client with input $\mathcal{C} = \{c_1, \ldots, c_v\}$. At the end, Client learns $|\mathcal{S} \cap \mathcal{C}|$. PSI-CA securely implements: $\mathcal{F}_{\mathsf{PSI\text{-}CA}} : (\mathcal{S}, \mathcal{C}) \mapsto (\bot, |\mathcal{S} \cap \mathcal{C}|)$.

Authorized Private Set Intersection (APSI) [18]: a protocol between Server with input $\mathcal{S} = \{s_1, \ldots, s_w\}$, and Client with input $\mathcal{C} = \{c_1, \ldots, c_v\}$ and $\mathcal{C}_\sigma = \{\sigma_1, \ldots, \sigma_v\}$. At the end, Client learns:

$$\mathsf{ASI} \stackrel{\text{def}}{=} \mathcal{S} \cap \{c_i \mid c_i \in \mathcal{C} \ \wedge \ \sigma_i \textit{ valid auth. on } c_i\}.$$

APSI securely implements: $\mathcal{F}_{\mathsf{APSI}} : (\mathcal{S}, (\mathcal{C}, \mathcal{C}_\sigma)) \mapsto (\bot, \mathsf{ASI})$.

Additively Homomorphic Encryption. Let $(K; Enc; Dec)$ be a homomorphic encryption scheme, where K is the key generator algorithm selecting public/secret key-pair (pk, sk). Assume that the message space for a public key pk is $\mathbb{Z}_p$ for some integer p, then $Enc(m)$ denotes encryption under key pk, and $Dec(c)$ denotes decryption under key sk. The following additive homomorphic properties hold: (1) the product of two ciphertexts is a ciphertext for the sum of the plaintexts, i.e., for any $a, b \in \mathbb{Z}_p$, we have $Dec(Enc(a) \cdot Enc(b)) = a + b$, and (2) raising a ciphertext for a message a to power r gives a ciphertext $r \cdot a$, i.e., for any $r \in \mathbb{Z}_p$, we have $Dec(Enc(a)^r) = r \cdot a$.

Adversarial Model. We use standard security models for secure two-party computation. One distinguishing factor is the adversarial model that is either semi-honest or malicious. (In the rest of this paper, the term *adversary* refers to insiders, i.e., protocol participants. Outside adversaries are not considered, since their actions can be mitigated via standard network security techniques.)

Following definitions in [34], protocols secure in the presence of *semi-honest adversaries* assume that parties faithfully follow all protocol specifications and do not misrepresent any information related to their inputs, e.g., size and content. However, during or after protocol execution, any party might (passively) attempt to infer additional information about the other party's input. This model is formalized by considering an ideal implementation where a trusted third party (TTP) receives the inputs of both parties and outputs the result of the defined function. Security in the presence of semi-honest adversaries requires that, in the real implementation of the protocol (without a TTP), each party does not learn more information than in the ideal implementation.

Security in the presence of *malicious parties* allows arbitrary deviations from the protocol. However, it does not prevent parties from refusing to participate in the protocol, modifying their inputs, or prematurely aborting the protocol. Security in the malicious model is achieved if the adversary (interacting in the real protocol, without the TTP) can learn no more information than it could in the ideal scenario. In other words, a secure protocol emulates (in its real execution) the ideal execution that includes a TTP. This notion is formulated by requiring the existence of adversaries in the ideal execution model that can simulate adversarial behavior in the real execution model.

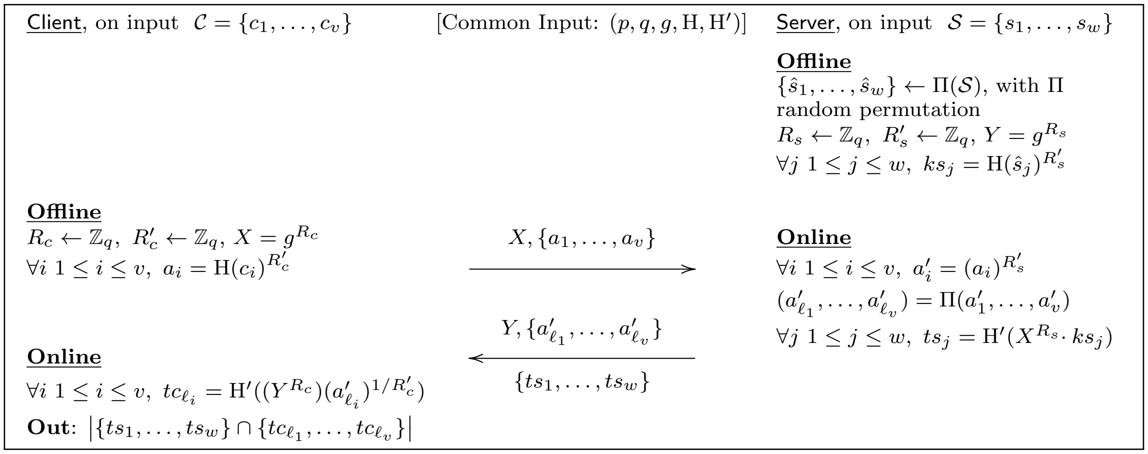
Although security arguments in this paper are made with respect to semi-honest participants, extensions to malicious participant security (with the same computation and communication complexities) have already been developed for our cryptographic building blocks: PSI, PSI-CA and APSI. We consider these extensions to be out of the scope of this paper.

4. GENOME TESTING

We now explore efficient techniques for privacy-preserving testing on fully sequenced genomes. Unlike most prior work (reviewed in Sec. 2), we do not seek generic solutions for genomic computation. Instead, we focus on a few specific real-world applications and, for each, capitalize on domain knowledge to propose an efficient privacy-preserving approach.

Notation. We assume that each participant has a digital copy of her fully sequenced genome denoted by $\mathcal{G} = \{(b_1||1), \ldots, (b_n||n)\}$, where $b_i \in \{\mathrm{A, G, C, T, -}\}$, n is the human genome length (i.e., $3 \cdot 10^9$), and "$||$" denotes concatenation. The "–" symbol is needed to handle DNA mu-

[2] NCBI maintains an interactive collection of SNPs, dbSNP, containing all known genetic variations of the human genome [62].


Client, on input $\mathcal{C} = \{c_1, \ldots, c_v\}$ [Common Input: $(p, q, g, \mathrm{H}, \mathrm{H}')$] Server, on input $\mathcal{S} = \{s_1, \ldots, s_w\}$

Offline (Server)
$\{\hat{s}_1, \ldots, \hat{s}_w\} \leftarrow \Pi(\mathcal{S})$, with Π random permutation
$R_s \leftarrow \mathbb{Z}_q,\ R'_s \leftarrow \mathbb{Z}_q,\ Y = g^{R_s}$
$\forall j\ 1 \le j \le w,\ ks_j = \mathrm{H}(\hat{s}_j)^{R'_s}$

Offline (Client)
$R_c \leftarrow \mathbb{Z}_q,\ R'_c \leftarrow \mathbb{Z}_q,\ X = g^{R_c}$
$\forall i\ 1 \le i \le v,\ a_i = \mathrm{H}(c_i)^{R'_c}$

$\xrightarrow{X, \{a_1, \ldots, a_v\}}$

Online (Server)
$\forall i\ 1 \le i \le v,\ a'_i = (a_i)^{R'_s}$
$(a'_{\ell_1}, \ldots, a'_{\ell_v}) = \Pi(a'_1, \ldots, a'_v)$
$\forall j\ 1 \le j \le w,\ ts_j = \mathrm{H}'(X^{R_s} \cdot ks_j)$

$\xleftarrow[\{ts_1, \ldots, ts_w\}]{Y, \{a'_{\ell_1}, \ldots, a'_{\ell_v}\}}$

Online (Client)
$\forall i\ 1 \le i \le v,\ tc_{\ell_i} = \mathrm{H}'((Y^{R_c})(a'_{\ell_i})^{1/R'_c})$
Out: $|\{ts_1, \ldots, ts_w\} \cap \{tc_{\ell_1}, \ldots, tc_{\ell_v}\}|$

Figure 1: PSI-CA protocol from [19]. It executes on common input of two primes p and q (such that $q|p-1$), a generator g of a subgroup of size q and two hash functions, H and H′, modeled as random oracles. All computation is mod p.

tations corresponding to *deletion*, i.e., where a portion of a chromosome is missing [56]. It is also used when the sequencing process fails to determine a nucleotide. This data may be pre-processed in order to speed up execution of specific applications. For example, parties may pre-compute a cryptographic hash, $\mathrm{H}(\cdot)$, on each nucleotide, alongside its position in the genome, i.e., for each $(b_i||i) \in \mathcal{G}$, they compute $hb_i = \mathrm{H}(b_i||i)$.[3]

We use the notation $|str|$ to denote the length of string str, and $|A|$ to denote the cardinality of set A. Finally, we use $r \leftarrow R$ to indicate that r is chosen uniformly at random from set R.

Experimental Setup. The rest of this section includes some experimental results. Unless explicitly stated otherwise, all experiments were performed on a Linux Desktop, with an Intel Core i5-560M (running at 2.66 GHz). All tests were run on a single processor core and all code is written in C, using OpenSSL and GMP libraries. Cryptographic protocols use the SHA-1 hash function and 1024-bit moduli. Source code of our experiments is available at http://sprout.ics.uci.edu/projects/privacy-dna.

4.1 Genetic Paternity Test

A Genetic Paternity Test (GPT) allows two individuals with their respective genomes to determine whether there exists a biological parent-child relationship between them. A ***Privacy-Preserving Genetic Paternity Test*** (PPGPT) achieves the same result without revealing any information about the two genomes. In the following, we refer to the two participants as Client and Server. Only Client receives the outcome of the test.

4.1.1 Strawman Approach

Genomics studies have shown that about 99.5% of any two human genomes are identical. Humans carry two copies of each chromosome, inherited one from the mother and one from the father. Thus, genomes carried by two individuals tied by a parent-child relationship show an even higher degree of similarity. As a result, one immediate computational technique for GPT is to compare the candidate's genome with that of the child; the test returns a positive result if the percentage of matching nucleotides is above a given threshold τ, i.e., significantly higher than 99.5%.

First-Attempt Protocol. At first glance, protecting privacy is relatively easy: recent proposals for Private Set Intersection Cardinality (PSI-CA) protocols [27, 76, 51, 19] offer efficient and private two-party computation of the number of set elements shared by two parties. Thus, to perform PPGPT, two participants just need to run PSI-CA on input of their respective genomes.

We select the PSI-CA construction from [19] (shown in Fig. 1) since it offers the best communication and computation complexities. Also, we use PSI-CA rather than PSI since *semi-honest* participants only need to learn *how similar* their genomes are. Whereas, PSI would also reveal *where* the two genomes differ and/or where they have common features.

We emphasize that this approach provides very accurate results, and is not significantly affected by potential sequencing errors. In fact, given expected error ratio ε, one can simply modify threshold τ to accommodate errors. This is because ε is expected to be significantly smaller than the difference between τ and the percentage of nucleotides that any two individuals share.

Unfortunately, since the number of nucleotides in the human genome is extremely large (about $3 \cdot 10^9$), this technique, though optimal in terms of accuracy, is impractical using current commodity hardware, as it requires both parties to perform online computation over the entire genome. Specifically, PSI-CA entails a number of (short) modular exponentiations linear in the input size. Table 1 estimates execution times and bandwidth incurred by this naïve approach. Since Client's online computation depends on that of the Server, a single test would consume approximately 10 days.

Improved Protocol. Since about 99.5% of the human genome is the same, two parties would only need to compare the remaining 0.5%. Unfortunately, there is yet not enough

[3] In case of *insertion* mutation in the genome, e.g., an 'A' is added between positions 35 and 36, genome pre-processing computes H(A||35||1). Similarly, if insertion involves multiple nucleotides. Since insertions are rare in human genomes, we do not consider them in this paper.

	Offline	Online	
	Time	Time	Size
Client	4.5 days	4.5 days	358 GB
Server	4.5 days	4.5 days	414 GB

Table 1: Computation and communication costs of the first straw-man PPGPT protocol.

statistical knowledge to pinpoint *where* exactly this 0.5% occurs. Nonetheless, experts claim that, in practice, comparing a properly chosen 1% of the genome yields an accuracy comparable to analyzing the entire genome [30]. Running times and bandwidth overhead required by this improved method are presented in Table 2.

	Offline	Online	
	Time	Time	Size
Client	67 mins	67 mins	3.57 GB
Server	67 mins	67 mins	4.14 GB

Table 2: Computation and communication costs of improved PPGPT protocol. Computation is performed over 1% of the human genome.

4.1.2 *Efficient RFLP-based PPGPT with PSI-CA*

We now present a very efficient technique for Privacy-Preserving Genetic Paternity Testing (PPGPT). To construct it, we take advantage of domain knowledge in genomics and build upon effective *in vitro* techniques (RFLP or SNP) rather than generic computational techniques. First, we design a protocol that implements RFLP-based GPT. Next, we propose a cryptographic technique for secure computation of this protocol that realizes PPGPT. Finally, we show that the technique used for computing RFLP-based GPT can be easily adapted to perform SNP-based GPT.

As discussed in Sec. 3.1, RFLPs use specific restriction enzymes (e.g., HaeIII, PstI, and HinfI), to digest a genome into hundreds of smaller fragments. Following the deterministic and well-known process, enzymes cut the DNA at each occurrence of a given pattern (e.g., "`CTGCAG`" with PstI). Next, a subset of these fragments is selected using a small number of probes for well-known markers, which are located in known areas of the genome. In an RFLP-based paternity test, this process is applied to the DNA of the two tested individuals. If resulting fragments have comparable lengths, then the test returns a positive with certain confidence, based on the exact number of fragments of the same length.

There are a few slightly different ways to select the type and the number of markers, thus identifying exactly which fragments to compare. For the sake of reliability, one needs to use markers that are rare enough (i.e., occur in unrelated individuals with very low probability) while common enough to occur in at least one of the tested subjects. Currently, public databases and scientific literature offer thousands available probes for RFLP in human genomes [11, 65, 74]. However, to reduce the cost of in vitro tests, only a small subset of them is actually used [20]. Different laboratories consider various accuracy/cost trade-offs. Some compare as few as 9-15 DNA markers, returning a positive result whenever fewer than two fragments do not match [13], with an estimated 99.9% accuracy. Meanwhile, others use up to 25 markers and return a positive whenever fewer than two fragments do not match, thus providing significantly higher accuracy, i.e., about 99.999% [24, 53].

In the United States, these testing methodologies follow precise regulations issued by the American Association of Blood Banks (AABB) and are considered legally admissible as evidence in the court of law. Since our PPGPT technique closely mimics the *in vitro* procedure, it achieves the same level of accuracy. Nevertheless, as the cost of RFLP emulation on digitalized genomes is not significantly affected by the number of selected markers, we can anticipate increasing the number of markers to improve accuracy. We could perform tests with 50 markers and show that this only adds a small cost. However, selection of additional markers is out of the scope of this paper, as their introduction does not change the algorithm's functionality presented below.

RFLP-based Protocol. This protocol involves two individuals, on private input of their respective fully sequenced genomes. We distinguish between Client and Server, to denote the fact that only the former learns the test outcome. The protocol is run on common input of: a threshold τ, a set of enzymes $E = \{e_1, \ldots, e_j\}$, and a set of markers $M = \{mk_1, \ldots, mk_l\}$. Each participant also inputs its digitized genome.

1. First, participants emulate the digestion process of each enzyme $e_i \in E$ on their genome. Consider, for instance, the PstI enzyme: whenever the string `CTGCAG` occurs, the enzyme cuts the genome in two fragments, so that the first ends with `CTGCA` and the second starts with `G`. As a result, genomes are digested into a large number of fragments of variable length.
2. Next, participants probe the fragments using markers in M. During this process, each participant selects up to l fragments $\{frag_1, \ldots, frag_l\}$ (e.g., $l = 25$), corresponding to M. All remaining fragments are discarded. Public markers are chosen such that each appears in at most one sequence.
3. Client builds the set $F_C = \{(|frag_i^{(c)}|, mk_i)\}_{i=1}^{l}$. For each marker i not corresponding to any fragment, $frag_i^{(c)}$ is replaced with the empty string. Similarly, Server builds $F_S = \{(|frag_i^{(s)}|, mk_i)\}_{i=1}^{l}$
4. Client and Server run the PSI-CA protocol described in Fig. 1, on respective inputs: F_C and F_S. Client learns $pt = |F_C \cap F_S|$, i.e., how many of its and Server's fragments are of the same size.
5. Client learns the test result by comparing pt to threshold τ.

Why Compare Lengths? It might seem that comparing string *lengths* is unreliable since two same-length strings might encode completely different content, while our protocol would consider these strings as matching. In practice, however, this well-established technique yields false positives with ***extremely low probability***. Sequences are selected using markers, i.e., according to (part of) their content. Selection of markers, in turn, guarantees that they appear only in one specific position in the entire genome. Edges of each fragment are content-dependent as well, since enzymes digest them according to a specific pattern of nucleotides. Therefore, two unrelated sequences of the same length would not be compared and two same-length sequences containing the same marker should be indeed considered matching.

Furthermore, this approach boosts the resilience of PPGPT against sequencing errors. Only errors occurring in the pattern digested by enzymes (or in the markers) influence the result of the RFLP-based PPGPT. However, since patterns and markers are relatively short compared to the size of the genome, this happens with very low probability, since sampling errors are uniformly distributed. However, if we let participants compare hashes of fragments, rather than their length, even a moderate error rate would severely increase the probability of false negatives, since even a single sequencing error would affect the final outcome of the test. Moreover, the main purpose of the PPGPT presented in this paper is not to improve accuracy of the *in vitro* test currently used, but to efficiently and securely replicate it *in silico*.

PSI-CA or PSI? The use of PSI-CA, rather than PSI, is needed to minimize information learned by Client from protocol execution. With PSI, if the number of matches is sufficiently high (even if the test is negative), Client would learn the lengths of several Server's fragments: it could then use this information to perform a paternity test between the party previously playing the role of Server and any other individual (although with slightly lower reliability).

SNP-based Protocol. SNP-based tests are replacing RFLP-based tests due to their better performance [8]. While this technique is not yet considered legally admissible in court, it is expected to eventually supersede its RFLP-based counterpart. Our RFLP-based protocol can be extended to perform paternity testing using SNPs: instead of selecting fragments using enzymes and markers, the SNP-based test selects fragments using a set of known SNPs. Since the rest of the protocol is unchanged and the size of the set of SNPs is usually 52 elements [8], the new protocol performs almost identically to the RFLP-based PPGPT protocol with 50 fragments.

Performance Evaluation. We now measure performance of the RFLP-based protocol on the Intel Core i5-560M testbed. The (offline) time needed to emulate the enzyme digestion process on the full genome is 74 seconds. This computation is performed only once, thus, it does not affect the time required to perform the interactive protocol. Finally, in order to assess the practicality of the protocol on embedded devices, we also measured its performance on a modern smartphone — a Nokia N900 equipped with ARM Cortex A8 CPU running at 600 MHz. Table 3 summarizes the online cost of the RFLP-based protocol, measuring computation and communication overhead, using different numbers of markers, on both i5-560M and A8 processors.

Entity	Offline (Time)		Online (Time/size)		
(markers)	i5-560M	A8	i5-560M	A8	Size
Client (25)	3.4 ms	323 ms	3.4 ms	323 ms	3 KB
Server (25)	3.4 ms	323 ms	3.4 ms	323 ms	3.5 KB
Client (50)	6.7 ms	645 ms	6.7 ms	645 ms	6 KB
Server (50)	6.7 ms	645 ms	6.7 ms	645 ms	7 KB

Table 3: Computation and communication costs of RFLP-based PPGPT technique, testing 25 and 50 fragments.

For the sake of completeness, we compared our results to prior work on privacy-preserving paternity testing, presented in Figure 3 of [7]. Following a conservative approach, we instantiate: (i) the cheapest protocol variant, which tolerates no error, and (ii) the most efficient additively homomorphic cryptosystem among those suggested, i.e., modified ElGamal [23]. Also, we only count the number of modular exponentiations. Given that the paternity test is performed over n alleles (with n ranging from 13 to 67 for increasing accuracy) we estimate the following costs. In step (2) of the protocol, the party obtaining the test result computes $8n$ modified ElGamal encryptions, thus, incurring $24n$ (short) modular exponentiations. In the i5-560M testbed, this takes from 43ms to 224ms, depending on n. In step (3), the other party needs to obtain the encrypted sum using homomorphic properties: it does so by performing $30n$ exponentiations. This takes between 54 and 262ms on the i5-560M testbed. Even ignoring all other operations in [7] and without precomputation, our most accurate test (using 50 markers) is about 5 times faster than the least accurate test in [7] (using 13 alleles).

4.1.3 PPGPT using Private Equality Testing

We now discuss another approach to PPGPT that uses Private Equality Testing (PET) and homomorphic encryption.

Comparing Two Genomes. As mentioned in Section 4.1.1, about 99.5% of any two human genomes are identical. Therefore, the most natural way of performing paternity test appears to be by determining how many nucleotides are shared between them. We already outlined a mechanism for privacy-preserving computation of such a test by representing a genome as a (unordered) set where each element correspods to a position-numbered nucleotide and then using PSI-CA to obtain the number of matching nucleotides. We now consider another technique. If we represent a genome as an ordered vector, then we can count the number of matching nucleotides by testing pairwise equality of vector elements. In the privacy-preserving "world", this problem can be solved using so-called Private Equality Testing (PET).

Given a probabilistic additively homomorphic encryption scheme, $(K; Enc; Dec)$, such as modified ElGamal [23], Paillier [66], or DGK [15], PET can be realized as follows. We assume Client and Server, on input of two items c and s, respectively, want to verify whether or not $c = s$:

1. Client generates (pk, sk) and sends Server $Enc(c)$;
2. Server replies with $(Enc(c) \cdot Enc(-s))^r$, for some random r in the message space;
3. Client learns that $c = s$ if Server's answers decrypts to zero, and nothing otherwise;

Note that "$\cdot$" denotes the operation on two ciphertexts that results in the ciphertext of the sum of their plaintexts, e.g., c and $-s$. Also, $Enc(x)^r$ is the operation on the ciphertext that yields modular exponentiation with exponent r of the corresponding plaintext x.

By extending this PET to n parallel executions (where n is the the genome size in nucleotides), Client would learn how many nucleotides they have in common. However, it would also learn *which* ones are shared. To prevent the latter, we modify the protocol as follows. Assuming that c_i and s_i represent the i-th nucleotide in Client's and Server's genomes, respectively:

1. Client generates (pk, sk) and sends Server:

$$\{Enc(c_1), \ldots, Enc(c_n)\}$$

2. Server replies with:

$$\Pi\left(\{(Enc(c_1) \cdot Enc(-s_1))^{r_1}, \ldots, (Enc(c_n) \cdot Enc(-s_n))^{r_n}\}\right)$$

where $\Pi(\cdot)$ is a random permutation and $r_1, \ldots, r_n$ are random in the message space.

3. Using $\Pi(\cdot)$, Client learns the number of ciphertexts that decrypt to yield a zero and, yet, does not learn *which* nucleotides match.

RFLP-based Paternity Test. We can also use PET to obtain PPGPT based on RFLP. Recall from Section 4.1.2 that, after enzyme digestion and marker probing, Client and Server obtain, respectively, $\{|frag_i^{(c)|}\}_{i=1}^{l}$ and $\{|frag_i^{(s)}|\}_{i=1}^{l}$. Similar to the technique discussed above, they can compute the number of fragments of the same length by using PET. (Recall that a list of fragments is an ordered vector). The resulting protocol is as follows:

1. Client generates (pk, sk) and sends Server:

$$\{Enc(|frag_1^{(c)}|), \ldots, Enc(|frag_l^{(c)}|)\}$$

2. Server replies with:

$$\Pi\Big(\{(Enc(|frag_1^{(c)}|) \cdot Enc(-|frag_1^{(s)}|))^{r_1}, \ldots,$$
$$\ldots, (Enc(|frag_l^{(c)}|) \cdot Enc(-|frag_l^{(s)}|))^{r_l}\}\Big)$$

where $\Pi(\cdot)$ is a random permutation and $r_1, \ldots, r_l$ are random in the message space.

3. By decryption, Client learns the number of matching fragment lengths (and nothing else) and determines the test outcome.

4.2 Personalized Medicine

Personalized Medicine (PM) is increasingly used to provide patients with drugs designed for their specific genetic features. As discussed in Sec. 1, in the context of PM, drugs are associated with a unique genetic fingerprint. Their effectiveness is maximized in patients with a matching DNA [39]. To this end, genomes need to be compared against the fingerprint and a patient need to surrender her DNA to a physician or a pharmaceutical company.

One privacy-preserving approach is to let the patient independently run specialized software over her genome and identify a match (or lack thereof) with a given drug's fingerprint. This way, the patient would learn whether the drug is appropriate. However, pharmaceuticals may consider DNA fingerprints of their drugs to be trade secrets and thus might be unwilling to reveal them. At the same time, for every new drug, pharmaceuticals are required to obtain approval from appropriate government entities, e.g., the Food and Drug Administration (FDA) in case of the United States.

We now introduce a technique for ***Privacy-Preserving Personalized Medicine Testing*** (P^{3}MT), involving the following steps:

- Following positive clinical trials, a pharmaceutical company obtains FDA approval on a specific DNA fingerprint fp and receives a corresponding authorization, $auth$.
- The pharmaceutical and the patient engage in a protocol, where the former inputs $(fp, auth)$ and the latter inputs her genome.
- At the end of the protocol, the pharmaceutical learns whether the patient's genome matches fingerprint fp, provided that $auth$ is a valid authorization of fp.

Privacy requirements are that: (1) the company learns nothing about patient genome besides the part matching the (authorized) fingerprint, and (2) the patient learns nothing about fp or $auth$.

4.2.1 P^{3}MT Instantiation

We now present a specific P^{3}MT instantiation. It involves: (1) an authorization authority (e.g., the FDA) denoted as CA, (2) a pharmaceutical — Client, and (3) a patient — Server.

Our cryptographic building block is Authorized Private Set Intersection (APSI) [9, 18, 17], hence, our Client/Server/CA notation. We select one specific APSI construction in [17], illustrated in Fig. 2, since it currently offers lowest communication and computation complexity. (Moreover, it can be instantiated in the malicious model with only a small constant additional overhead.) For efficiency reasons, $R_{c:i}$'s and R_s are chosen uniformly at random from $W = [1..\lfloor\sqrt{N}/2\rfloor]$, rather than from $\mathbb{Z}_{N/2}$, as in the original version of the protocol. In fact, as proved in [33], the distribution of $g^x \bmod N$ with $x \leftarrow W$ is computationally indistinguishable from the distribution defined by g^x with $x \leftarrow [1..\phi(N)]$. This change does not affect protocol security arguments. Thus, we do not provide a new proof for APSI in this paper.

P^{3}MT involves two phases: *offline* and an *online*.

During the *offline* phase:

1. CA generates RSA public-private keypair $((N, e), d)$, publishes (N, e), and keeps d private.
2. Client prepares a fingerprint of drug $\mathcal{D}$: $fp(\mathcal{D}) = \{(b_j^*||j)\}$, where each b_j^* is expected at position j of a genome suitable for $\mathcal{D}$.
3. Client obtains from CA an authorization $auth(fp(\mathcal{D}))$, where $auth(fp(\mathcal{D})) = \{\sigma_j \mid \sigma_j = \mathrm{H}(b_j^*||j)^d \bmod N\}$.
4. Server runs the offline stage of the APSI protocol in Fig. 2, on input, $\mathcal{G} = \{(b_1||1), \ldots, (b_n||n)\}$, and publishes resulting $\{ts_1, \ldots, ts_n\}$.

During the *online* phase:

1. Client and Server run the online part of the APSI protocol in Fig. 2. Recall that Client's input is $(fp(\mathcal{D}), auth(fp(\mathcal{D})))$, and Server's is $\mathcal{G}$.
2. After the interaction, Client obtains $fp(\mathcal{D}) \cap \mathcal{G}$, and uses this information to determine whether Server is well-suited for drug $\mathcal{D}$.

We note that $auth$ is needed to limit the scope of the test on a patient DNA: the FDA can guarantee that: (i) fp only covers the appropriate set of required nucleotides, and (ii) pharmaceuticals cannot input arbitrary portions of a patient genome.

The proposed P^{3}MT protocol is resilient against (randomly distributed) sequencing errors. The size of the fingerprint input by Client in the protocol is negligible compared to the size of the entire genome. Thus, positions corresponding to Client input are affected by errors with extremely low probability.

Performance Evaluation. To estimate the efficiency of the P^{3}MT protocol, we consider two genetic tests commonly performed in the context of personalized medicine: the analysis of *hla-B* and *tpmt* genes. Our choice is also motivated by the size of their fingerprints that, according to genomics experts, is representative of most personalized medicine tests.

First, we look at the *hla-B*5701* allelic variant, one G→T mutation associated with extreme sensitivity to abacavir,

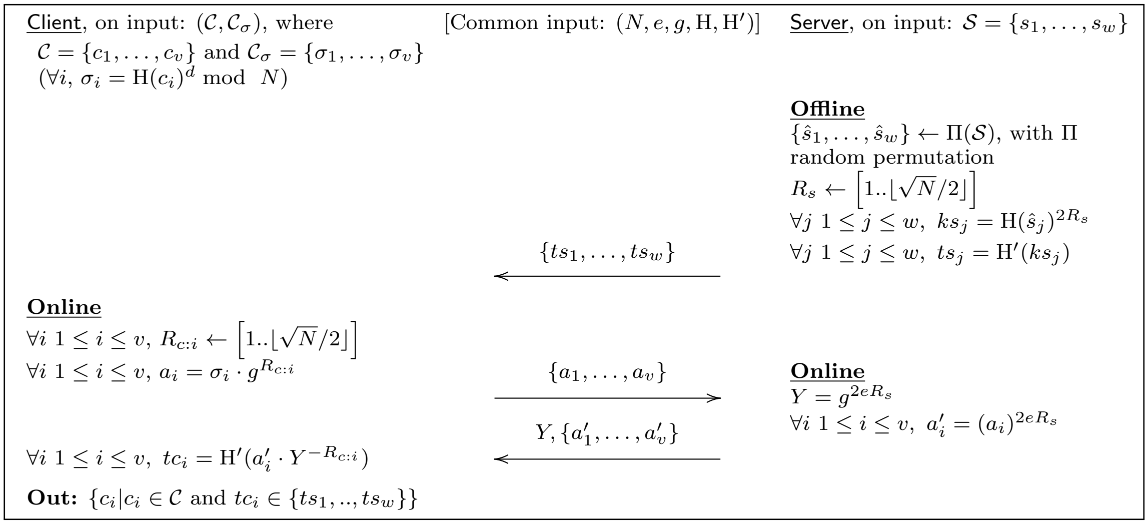

Client, on input: $(\mathcal{C}, \mathcal{C}_\sigma)$, where
$\mathcal{C} = \{c_1, \ldots, c_v\}$ and $\mathcal{C}_\sigma = \{\sigma_1, \ldots, \sigma_v\}$
$(\forall i,\ \sigma_i = \mathrm{H}(c_i)^d \bmod N)$
[Common input: $(N, e, g, \mathrm{H}, \mathrm{H}')$]
Server, on input: $\mathcal{S} = \{s_1, \ldots, s_w\}$
Offline
$\{\hat{s}_1, \ldots, \hat{s}_w\} \leftarrow \Pi(\mathcal{S})$, with Π random permutation
$R_s \leftarrow \left[1..\lfloor\sqrt{N}/2\rfloor\right]$
$\forall j\ 1 \le j \le w,\ ks_j = \mathrm{H}(\hat{s}_j)^{2R_s}$
$\forall j\ 1 \le j \le w,\ ts_j = \mathrm{H}'(ks_j)$
$\{ts_1, \ldots, ts_w\}$
Online
$\forall i\ 1 \le i \le v,\ R_{c:i} \leftarrow \left[1..\lfloor\sqrt{N}/2\rfloor\right]$
$\forall i\ 1 \le i \le v,\ a_i = \sigma_i \cdot g^{R_{c:i}}$
$\{a_1, \ldots, a_v\}$
Online
$Y = g^{2eR_s}$
$\forall i\ 1 \le i \le v,\ a'_i = (a_i)^{2eR_s}$
$Y, \{a'_1, \ldots, a'_v\}$
$\forall i\ 1 \le i \le v,\ tc_i = \mathrm{H}'(a'_i \cdot Y^{-R_{c:i}})$
Out: $\{c_i | c_i \in \mathcal{C}$ and $tc_i \in \{ts_1, .., ts_w\}\}$

Figure 2: APSI Protocol from [17] (simplified for semi-honest security). The protocol is run on common input of RSA modulus $N = pq$ (with p and q safe primes), public exponent e, a random element g in $\mathbb{Z}_N^*$ and two hash functions, H and H′, modeled as random oracles. All computation is mod N.

a drug used in HIV treatment [61]. In diploid organisms (such as humans), mutation may occur in either chromosome inherited from the parents. Thus, the related fingerprint contains 2 (*nucleotide*, *position*) pairs. We also consider the analysis of *tpmt* typically done before prescribing 6-mercaptopurine to leukemia patients. As shown in [85], two alleles are known to cause the *tpmt* disorder: (1) one presents a mutation G→C in position 238 of gene's c-DNA, (2) the other presents one mutation G→A in position 460 and one A→G in position 719.[4] Therefore, the resulting fingerprint contains these 6 (nucleotide, position) pairs.

In the underlying APSI protocol (Fig. 2), cryptographic operations on Server genome do not depend on Client input. Therefore, they can be computed offline, once for all possible tests. Moreover, we have designed the P^{3}MT protocol to be as generic as possible. Our protocol runs on the whole Server's genome — with linear complexity — in order to address future scenarios where genomics advances will cause better understanding of many more regions of human genomes. To reduce offline costs, we apply reference-based compression [14, 6] – a technique commonly used to efficiently represent genomic information. In particular, Server input consists of all differences between its genome and the reference sequence. We emphasize that this technique does not require any biological correctness of the reference genome that is only used for compression [42]. This allows us to reduce the size of Server input to about 1% of the entire genome.

Table 4 summarizes execution time and bandwidth costs of the P^{3}MT protocol used for testing *hla-B* and *tpmt*. These costs cannot be meaningfully compared to prior work, since, to the best of our knowledge, there is no other technique targeting privacy-preserving personalized medicine testing. Furthermore, as mentioned in Sec. 2, there are no current techniques that enforce fingerprint *authorization* by a trusted entity, such as the FDA. Also, prior work is essentially designed for operation on DNA snippets, and it is unclear how to efficiently adapt it to full genomes. Although a detailed experimental study is out of scope of this paper, we intend to include it as part of future work.

Test	Party	Offline	Online	
		Time	Time	Size
*hla-b**5701	Client	–	0.82 ms	256 B
	Server	206 mins	0.82 ms	4.14 GB
tpmt	Client	–	2.46 ms	768 B
	Server	206 mins	2.46 ms	4.14 GB

Table 4: Computation and communication costs of P^{3}MT protocol for *hla-b* (2-nucleotide fingerprint) and *tpmt* (6-nucleotide fingerprint) tests.

4.3 Privacy-Preserving Genetic Compatibility Testing

Genetic Compatibility Testing (GCT) can predict whether potential partners are at risk of conceiving a child with a recessive genetic disease. This occurs when both partners carry at least one gene affected by mutation, i.e., they are either asymptomatic carriers or actual disease sufferers. As in the Beta-Thalassemia example discussed in Sec. 1, asymptomatic carriers usually need to learn whether their potential partner is also a carrier of the same disease, since this would pose a serious risk to their potential off-spring.

To achieve genetic compatibility testing with privacy we introduce the concept of ***Privacy-Preserving Genetic Compatibility Testing*** (PPGCT) that allows participants to run GCT without disclosing to each other: (1) any other genomic information, and (2) which disease(s) they are carrying or being tested for.

Current biological knowledge of the human genome allows screening for a genetic disease associated with one SNP in a specific gene. In other words, most well-characterized genetic diseases are caused by a mutation in a single gene. However, we anticipate that, in the near future, researchers will develop tests for more complex diseases (e.g., diabetes or hypertension) involving multiple genes and multiple mu-

[4] For more details on *tpmt* and c-DNA, refer to [63] and [56], respectively.

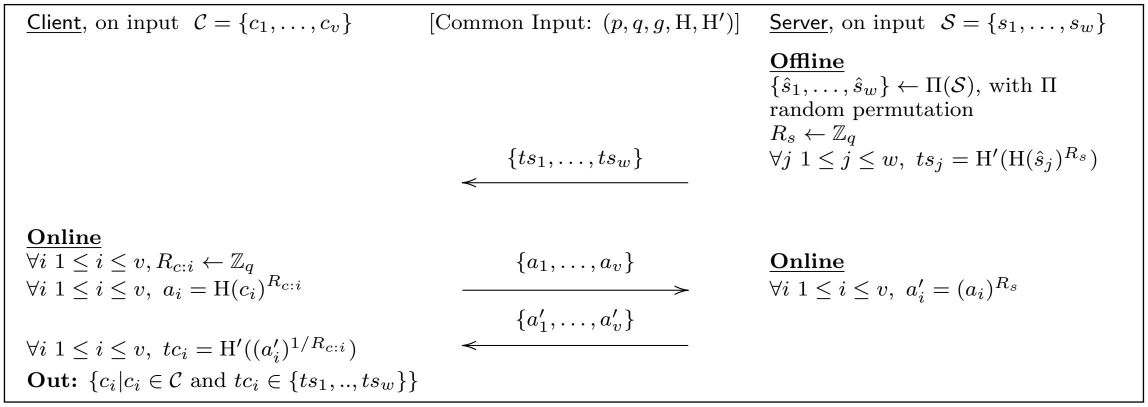

Client, on input $\mathcal{C} = \{c_1, \ldots, c_v\}$
[Common Input: $(p, q, g, \mathrm{H}, \mathrm{H}')$]
Server, on input $\mathcal{S} = \{s_1, \ldots, s_w\}$
Offline
$\{\hat{s}_1, \ldots, \hat{s}_w\} \leftarrow \Pi(\mathcal{S})$, with Π random permutation
$R_s \leftarrow \mathbb{Z}_q$
$\forall j\ 1 \leq j \leq w,\ ts_j = \mathrm{H}'(\mathrm{H}(\hat{s}_j)^{R_s})$
$\{ts_1, \ldots, ts_w\}$
Online
$\forall i\ 1 \leq i \leq v, R_{c:i} \leftarrow \mathbb{Z}_q$
$\forall i\ 1 \leq i \leq v,\ a_i = \mathrm{H}(c_i)^{R_{c:i}}$
$\{a_1, \ldots, a_v\}$
Online
$\forall i\ 1 \leq i \leq v,\ a'_i = (a_i)^{R_s}$
$\{a'_1, \ldots, a'_v\}$
$\forall i\ 1 \leq i \leq v,\ tc_i = \mathrm{H}'((a'_i)^{1/R_{c:i}})$
Out: $\{c_i | c_i \in \mathcal{C}$ and $tc_i \in \{ts_1, .., ts_w\}\}$

Figure 3: PSI Protocol from [44] (simplified for semi-honest security). It runs on common input of two primes p and q (s.t. $q|p-1$), a generator g of a subgroup of size q and two hash functions, H and H′, modeled as random oracles. All computation is mod p.

tations. Therefore, we aim to design PPGCT techniques not limited to single-mutation diseases. Additional motivating examples for PPGCT include compatibility testing for sperm and organ donors.

The proposed PPGCT protocol involves two participants: Client and Server. Client runs on input of a fingerprint of a genetic disease $\hat{\mathcal{D}}$. Server runs on input of its fully-sequenced genome $\mathcal{G}$. At the end of the interaction, Client learns the output of the test, i.e., whether Server carries disease $\hat{\mathcal{D}}$.

Our cryptographic building block is Private Set Intersection (PSI) [18, 27, 44, 17]. We select the specific PSI construction in [44], shown in Fig. 4, since it achieves the best communication and computation complexity. It can also be instantiated in the malicious model with only a small constant additional overhead.

The PPGCT protocol involves the following steps:

1. Client builds a fingerprint corresponding to her genetic diseases $fp(\hat{\mathcal{D}}) = \{(b^*_j || j)\}$, where each b^*_j *is expected at position j of a genome with disease $\hat{\mathcal{D}}$.*
2. Client and Server run the PSI protocol in Fig. 4 on respective inputs: $fp(\hat{\mathcal{D}})$ and $\mathcal{G}$.
3. Client obtains $fp(\hat{\mathcal{D}}) \cap \mathcal{G}$, and uses this information to determine whether Server carries disease $\hat{\mathcal{D}}$.

The change from PSI-CA to PSI is motivated as follows. Depending on the disease being tested, a positive outcome occurs if the genome contains either: (1) the entire disease fingerprint, or (2) a given subset of nucleotides. In case of (1), the test result is positive only if: $fp(\hat{\mathcal{D}}) \subset \mathcal{G}$, i.e., $fp(\hat{\mathcal{D}}) \cap \mathcal{G} = fp(\hat{\mathcal{D}})$: if this happens, there is actually no difference between the output of PSI and that of PSI-CA. However, PSI-CA is preferred over PSI since, if the test is negative, less information about Server genome is revealed to Client. In case of (2), cardinality of set intersection is insufficient to assess the test result, since Client needs to learn which fingerprint nucleotides appear in Server's genome.

Similar to its P^{3}MT counterpart, the PPGCT protocol is resilient to uniformly distributed errors. In particular, since input size of Client is small, corresponding positions in Server genome are affected by errors with very low probability.

Open Problem: Unfortunately, a malicious Client could potentially *harvest* Server's genetic information (in addition to that needed for the compatibility test) by inflating its input. For instance, a healthy Client could learn whether or not Server carries a given genetic disease, unrelated to the compatibility testing.

Performance. As concrete examples, we use genetic compatibility tests for two genetic disorders: Roberts syndrome and Beta-Thalassemia. We chose them since they are fairly common and the size of their fingerprints is representative of that in most genetic compatibility tests.

Similar to P^{3}MT, we stress that cryptographic operations performed on Server genome, in the underlying PSI protocol, do not depend on Client input. Therefore, these operations can be pre-computed (just once) ahead of time.

First, we consider testing for Roberts syndrome. an autosomal genetic disorder, characterized by pre- and postnatal growth deficiency, limb malformations, and distinctive skull and facial abnormalities. As shown in [35], there are 26 single point mutations (in the *esco2* gene) causing this syndrome. Since humans are diploid organisms, we expect Roberts syndrome fingerprint to contain about 52 (nucleotide, location) pairs.

Next, we turn to Beta-Thalassemia. As pointed out in [28], more than 250 mutations in the *hbb* gene have been found to cause this disorder and most of them involve a change in a single nucleotide. Although reliable techniques to perform this test *in silico* are not yet available, it is reasonable to assume that the size of the Beta-Thalassemia fingerprint would include 2×250 = 500 (*nucleotide, location*) pairs.

Table 5 summarizes run time (computational) and bandwidth requirements for the PPGCT protocol for Roberts syndrome and Beta-Thalassemia, respectively. Following the same arguments as in P^{3}MT experiments, we let Server input the portion of its genome that differs from the reference genome, i.e., about 1%.

Performance of the PPGCT protocol cannot be meaningfully compared to prior work. As discussed in Sec. 2, it is not trivial to adapt current secure pattern matching techniques to genetic compatibility testing on fully sequenced genomes. An experimental study (including the adaptation of such techniques) is left for future work.

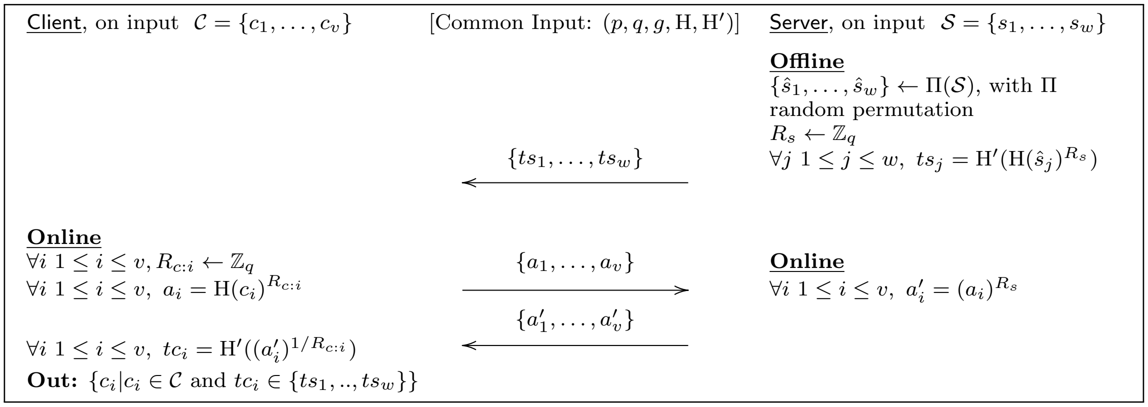

Client, on input $\mathcal{C} = \{c_1, \ldots, c_v\}$
[Common Input: $(p, q, g, \mathrm{H}, \mathrm{H}')$]
Server, on input $\mathcal{S} = \{s_1, \ldots, s_w\}$
Offline
$\{\hat{s}_1, \ldots, \hat{s}_w\} \leftarrow \Pi(\mathcal{S})$, with Π random permutation
$R_s \leftarrow \mathbb{Z}_q$
$\forall j \; 1 \le j \le w, \; ts_j = \mathrm{H}'(\mathrm{H}(\hat{s}_j)^{R_s})$
$\{ts_1, \ldots, ts_w\}$
Online
$\forall i \; 1 \le i \le v, R_{c:i} \leftarrow \mathbb{Z}_q$
$\forall i \; 1 \le i \le v, \; a_i = \mathrm{H}(c_i)^{R_{c:i}}$
$\{a_1, \ldots, a_v\}$
Online
$\forall i \; 1 \le i \le v, \; a'_i = (a_i)^{R_s}$
$\{a'_1, \ldots, a'_v\}$
$\forall i \; 1 \le i \le v, \; tc_i = \mathrm{H}'((a'_i)^{1/R_{c:i}})$
Out: $\{c_i | c_i \in \mathcal{C}$ and $tc_i \in \{ts_1, .., ts_w\}\}$

Figure 4: PSI Protocol from [44] (simplified for semi-honest security). It runs on common input of two primes p and q (s.t. $q|p-1$), a generator g of a subgroup of size q and two hash functions, H and H$'$, modeled as random oracles. All computation is mod p.

Test	Party	Offline	Online	
		Time	Time	Size
Roberts syndrome	Client	–	7.26 ms	62.5 KB
	Server	67 mins	7.26 ms	4.14 GB
Beta-Thalassemia	Client	–	70 ms	6.5 KB
	Server	67 mins	70 ms	4.14 GB

Table 5: Computation and communication costs of the PPGCT protocol for Beta-Thalassemia (500-nucleotide fingerprint) and Roberts syndrome (52-nucleotide fingerprint) tests.

5. SECURITY DISCUSSION

We now discuss security properties of protocols presented in this paper. In general, security of each protocol is based on that of the underlying building blocks. Therefore, we omit proof details to ease presentation. Also, out cryptographic building blocks (PSI-CA, APSI, and PSI) can be generally used in a *black-box* manner. One can select any instantiation without affecting security of our protocols, as long as the chosen construction yields secure PSI/APSI/PSI-CA functionality. However, we pick specific instantiations to maximize protocol efficiency. As discussed earlier, we consider semi-honest adversaries (participants). Nevertheless, we are not restricted to this model, since our cryptographic building blocks are (provably) adaptable to the malicious participant model, incurring a small constant extra overhead.

PPGPT. We now show that RFLP-based PPGPT protocol (Sec. 4.1) is secure against semi-honest adversaries. We assume that PSI-CA performs secure computation of the $\mathcal{F}_{\mathsf{PSI\text{-}CA}}$ functionality, in the presence of semi-honest participants. We select the construction in [19], that is secure under the One-More-DH assumption in the Random Oracle Model (ROM).
We divide the protocol in two phases. In the first, both Client and Server privately and independently perform the RFLP-related computation on their respective inputs. (This covers steps 1 to 3 of PPGPT). At the end of this phase, Client and Server construct sets F_C and F_S, respectively. Clearly, during this phase, neither participant learns anything about the other's input. During the second phase (steps 4-5), participants use F_C and F_S as their respective inputs to PSI-CA. Given the security of the latter, Client only learns $|F_S \cap F_C|$. PSI-CA protocols may reveal $|F_S|$ to Client and $|F_C|$ to Server. However, $|F_S|=|F_C|=l$, which is already known to both parties.

P^{3}MT. Similarly, security of the P^{3}MT protocol (in Sec. 4.2), against semi-honest Client and Server, stems from security of the underlying protocol — APSI. That is, if APSI performs secure computation of the $\mathcal{F}_{\mathsf{APSI}}$ functionality in the presence of semi-honest participants, then P^{3}MT is also secure. This holds since a semi-honest participant with a non-negligible advantage in distinguishing between real and simulated executions of P^{3}MT would have the same advantage in distinguishing between real and simulated executions of APSI. Although one can use APSI as a black box, for efficiency reasons, we prefer instantiations that allow pre-computation on Server input. In our instantiation, we select the APSI construction in [17], proven secure under the RSA and DDH assumptions (in ROM).

PPGCT. Finally, security of the PPGCT protocol (Sec. 4.3) against semi-honest adversaries relies on that of the underlying PSI protocol, to which it is immediately reducible. (In other words, a semi-honest participant with a non-negligible advantage in distinguishing between real and simulated executions of PPGCT would have the same advantage in distinguishing between real and simulated executions of PSI.) Again, although one can use PSI as a black box, for efficiency reasons, we need PSI instantiations that allow pre-computation on Server input, such as OPRF-based constructs [37, 18, 17, 44]. We chose the PSI from [44], proven secure under the One-More-DH assumption (in ROM).

6. CONCLUSIONS AND FUTURE WORK

This paper identified and explored three popular privacy-sensitive genomic applications: (i) paternity tests, (ii) personalized medicine and (iii) genetic compatibility testing. Unlike most previous work, we focused on fully sequenced genomes. This scenario poses new challenges, both in terms of privacy and computational cost. For each application, we proposed an efficient construction, based on well-known cryptographic tools: Private Set Intersection (PSI), Private Set Intersection Cardinality (PSI-CA), and Authorized Private Set Intersection (APSI). Experiments show that these protocols incur online overhead sufficiently low to be practical today. In particular, our protocol for privacy-preserving paternity testing is significantly less expensive — in both computation and communication — than prior work. Furthermore, all protocols presented in this paper have been carefully constructed to mimic the state-of-the-art of (*in vitro*) biological tests currently performed in hospitals and laboratories.

Items for future work include, but are not limited to:

- Introducing privacy-preserving genetic paternity testing based on STR and/or SNP comparison.
- Exploring privacy-preserving techniques to realize genetic ancestry testing, i.e., to discover whether or not individuals are related up to a certain degree (e.g., see http://23andme.com.)
- Exploring probabilistic privacy-preserving genetic paternity and ancestry testing based on MinHash techniques, as discussed in [5].
- Extending the paternity test protocol to allow both participants to determine whether the other party introduced correct input according to some auxiliary authorization. (Note that APSI does not suffice since one of the parties might alter its input so that the test is negative).
- Investigation of additional privacy-sensitive applications for fully-sequenced genomes, such as certified forensic identification, where the subject of investigation must prove the authenticity of its input; privacy-preserving organ recipients compatibility, where a subject efficiently identifies a matching sample without revealing information about her genome.
- Extending our experiments to include adaptation of secure pattern matching and text processing to personalized medicine and genetic compatibility testing on full genomes.

Acknowledgements. We are grateful to Christophe Magnan for useful hints about the testing environment and to anonymous ACM CCS'11 reviewers for helping us improve the paper. Work of Pierre Baldi is supported, in part, by grants: NIH LM010235 and NIH-NLM T15 LM07443.

7. REFERENCES

[1] A. Abbott. Special section on human genetics: With your genes? Take one of these, three times a day. *Nature*, 425(6960), 2003.

[2] M. Adams et al. The Genome Sequence of Drosophila melanogaster. *Science*, 287(5461), 2000.

[3] JS Beckmann and M. Soller. Restriction fragment length polymorphisms and genetic improvement of agricultural species. *Euphytica*, 35(1):111–124, 1986.

[4] M. Blanton and M. Aliasgari. Secure outsourcing of dna searching via finite automata. In *DBSec*, pages 49–64, 2010.

[5] C. Blundo, E. De Cristofaro, and P. Gasti. EsPRESSo: Efficient Privacy-Preserving Evaluation of Sample Set Similarity. http://arxiv.org/abs/1111.5062, 2011.

[6] M.C. Brandon, D.C. Wallace, and P. Baldi. Data structures and compression algorithms for genomic sequence data. *Bioinformatics*, 25(14), 2009.

[7] F. Bruekers, S. Katzenbeisser, K. Kursawe, and P. Tuyls. Privacy-Preserving Matching of DNA Profiles. http://eprint.iacr.org/2008/203, 2008.

[8] C. Børsting, J.J. Sanchez, H.E. Hansena, A.J. Hansena, H.Q. Bruuna, and N Morlinga. Performance of the SNPforID 52 SNP-plex assay in paternity testing. *Forensic Science International: Genetics*, 2(4):292–300, 2008.

[9] J. Camenisch and G. Zaverucha. Private intersection of certified sets. In *Financial Cryptography*, 2009.

[10] B. Carlson. SNPs – A shortcut to personalized medicine. *Genetic Engineering & Biotechnology News*, 2008.

[11] Center for Applied Genomics, University of Toronto. Database of Genomic Variants. http://projects.tcag.ca/variation, 2011.

[12] F.S. Collins and V.A. McKusick. Implications of the Human Genome Project for medical science. *Jama*, 285(5), 2001.

[13] L. Cunningham. High-stakes Test. Daily Business Review, 2003.

[14] K. Daily, P. Rigor, S. Christley, X. Xie, and P. Baldi. Data structures and compression algorithms for high-throughput sequencing technologies. *BMC bioinformatics*, 11(1), 2010.

[15] I. Damgård, M. Geisler, and M. Krøigård. Homomorphic encryption and secure comparison. *Journal of Applied Cryptology*, 1(1):22–31, 2008.

[16] G. Danezis, C. Diaz, S. Faust, E. Käsper, C. Troncoso, and B. Preneel. Efficient negative databases from cryptographic hash functions. *ISC*, pages 423–436, 2007.

[17] E. De Cristofaro, J. Kim, and G. Tsudik. Linear-complexity private set intersection protocols secure in malicious model. In *Asiacrypt*, pages 213–231, 2010.

[18] E. De Cristofaro and G. Tsudik. Practical Private Set Intersection Protocols with Linear Complexity. In *Financial Cryptography*, 2010.

[19] E. De Cristofaro and G. Tsudik. Fast and Private Computation of Set Intersection Cardinality. Cryptology ePrint Archive, Report 2011/141, 2011. http://eprint.iacr.org/2011/141.

[20] N.C. Dracopoli, J.L. Haines, and B.R. Korf. *Current protocols in human genetics*. John Wiley & Sons, 1994.

[21] R.M. Durbin, D.L. Altshuler, G.R. Abecasis, D.R. Bentley, A. Chakravarti, A.G. Clark, F.S. Collins, F.M. De La Vega, P. Donnelly, M. Egholm, et al. A map of human genome variation from population-scale sequencing. *Nature*, 467(7319):1061–1073, 2010.

[22] M.L. Durham. NOTE AND COMMENT: How Research Will Adapt to HIPAA: A View from Within the Healthcare Delivery System. *Am. JL and Med.*, 28:491–509, 2002.

[23] T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE transactions on Information Theory*, 31(4):469–472, 1985.

[24] DJ Endean. RFLP analysis for paternity testing: observations and caveats. In *The International Symposium on Human Identification*, pages 55–76, 1989.

[25] J.H. Fowler, J.E. Settle, and N.A. Christakis. Correlated genotypes in friendship networks. *Proceedings of the National Academy of Sciences*, 108(5), 2011.

[26] M.J. Freedman, Y. Ishai, B. Pinkas, and O. Reingold. Keyword search and oblivious pseudorandom functions. In *TCC*, pages 303–324, 2005.

[27] M.J. Freedman, K. Nissim, and B. Pinkas. Efficient private matching and set intersection. In *Eurocrypt*, pages 1–19, 2004.

[28] Genetics Home Reference. HBB Gene. http://ghr.nlm.nih.gov/gene/HBB.

[29] R. Gennaro, C. Hazay, and J.S. Sorensen. Text Search Protocols with Simulation Based Security. In *PKC*, pages 332–350, 2010.

[30] R. Gibbs and A. Singleton. Application of genome-wide single nucleotide polymorphism typing: Simple association and beyond. *PLoS Genet*, 2(10), 10 2006.

[31] G.S. Ginsburg and H.F. Willard. Genomic and personalized medicine: foundations and applications. *Translational Research*, 154(6):277–287, 2009.

[32] A. Goffeau, BG Barrell, H. Bussey, RW Davis, B. Dujon, H. Feldmann, F. Galibert, J.D. Hoheisel, C. Jacq, M. Johnston, et al. Life with 6000 Genes. *Science*, 274(5287), 1996.

[33] O. Goldreich, R. Israel, and V. Rosen. On the security of modular exponentiation with application to the construction of pseudorandom generators. *Journal of Cryptology*, 16, 2000.

[34] Oded Goldreich. *Foundations of cryptography: Basic applications*, chapter 7.2.2. Cambridge Univ Press, 2004.

[35] M. Gordillo, H. Vega, A.H. Trainer, F. Hou, N. Sakai, R. Luque, H. Kayserili, S. Basaran, F. Skovby, R. Hennekam, et al. The molecular mechanism underlying Roberts syndrome involves loss of ESCO2 acetyltransferase activity. *Human molecular genetics*, 17(14), 2008.

[36] J.F. Gusella, N.S. Wexler, P.M. Conneally, S.L. Naylor, M.A. Anderson, R.E. Tanzi, P.C. Watkins, K. Ottina, M.R. Wallace, A.Y. Sakaguchi, et al. A polymorphic DNA marker genetically linked to Huntington's disease. *Nature*, 306(5940):234–238, 1983.

[37] C. Hazay and Y. Lindell. Efficient protocols for set intersection and pattern matching with security against malicious and covert adversaries. In *TCC*, pages 155–175, 2008.

[38] C. Hazay and T. Toft. Computationally secure pattern matching in the presence of malicious adversaries. *Asiacrypt*, pages 195–212, 2010.

[39] JW Ho, S. Choi, Y. Lee, TC Hui, SS Cherny, MM Garcia-Barceló, L. Carvajal-Carmona, R. Liu, S. To, T. Yau, et al. Replication study of SNP associations for colorectal cancer in Hong Kong Chinese. *British Journal of Cancer*, 2010.

[40] MA Hoffman. The genome-enabled electronic medical record. *Journal of Biomedical Informatics*, 40(1):44–46, 2007.

[41] N. Homer, S. Szelinger, M. Redman, D. Duggan, W. Tembe, J. Muehling, J.V. Pearson, D.A. Stephan, S.F. Nelson, and D.W. Craig. Resolving individuals contributing trace amounts of DNA to highly complex mixtures using high-density SNP genotyping microarrays. *PLoS Genetics*, 4(8), 2008.

[42] Markus Hsi-Yang Fritz, Rasko Leinonen, Guy Cochrane, and Ewan Birney. Efficient storage of high throughput dna sequencing data using reference-based compression. *Genome Research*, 21(5):734–740, May 2011.

[43] International Human Genome Sequencing Consortium. Initial sequencing and analysis of the human genome. *Nature*, 409:860–921, 2001.

[44] S. Jarecki and X. Liu. Fast Secure Computation of Set Intersection. In *SCN*, 2010.

[45] S. Jha, L. Kruger, and V. Shmatikov. Towards practical privacy for genomic computation. In *S&P*, pages 216–230, 2008.

[46] J. Kaiser. A plan to capture human diversity in 1000 genomes. *Science*, 319, 2008.

[47] M. Kantarcioglu, W. Jiang, Y. Liu, and B. Malin. A cryptographic approach to securely share and query genomic sequences. *Transactions on Information Technology in Biomedicine*, 12(5):606–617, 2008.

[48] F. Kastrinos, B. Mukherjee, N. Tayob, F. Wang, J. Sparr, V.M. Raymond, P. Bandipalliam, E.M. Stoffel, S.B. Gruber, and S. Syngal. Risk of pancreatic cancer in families with Lynch syndrome. *JAMA: The Journal of the American Medical Association*, 302(16), 2009.

[49] J. Katz and Y. Lindell. *Introduction to modern cryptography*. Chapman & Hall/CRC, 2008.

[50] J. Katz and J. Malka. Secure text processing with applications to private dna matching. In *CCS*, pages 485–492, 2010.

[51] L. Kissner and D. Song. Privacy-preserving set operations. In *Crypto*, pages 241–257, 2005.

[52] J.D. Kulynych and D. Korn. The New HIPAA (Health Insurance Portability and Accountability Act of 1996) Medical Privacy Rule. *Circulation*, 108, 2003.

[53] E.S. Lander. DNA fingerprinting on trial. *Nature*, 339(6225):501–505, 1989.

[54] V.I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet Physics Doklady*, volume 10, pages 707–710, 1966.

[55] S. Levy, G. Sutton, P.C. Ng, L. Feuk, A.L. Halpern, B.P. Walenz, N. Axelrod, J. Huang, E.F. Kirkness, G. Denisov, et al. The diploid genome sequence of an individual human. *PLoS biology*, 5(10), 2007.

[56] R. Lewis and A. Reynolds. *Human genetics: concepts and applications*. McGraw-Hill, 2003.

[57] B. Malin. An evaluation of the current state of genomic data privacy protection technology and a roadmap for the future. *Journal of the American Medical Informatics Association*, 12(1):28–34, 2005.

[58] A.L. McGuire and R.A. Gibbs. COLUMN: Currents in Contemporary Ethics: Meeting the Growing Demands of Genetic Research. *JL Med. & Ethics*, 34:809–828, 2006.

[59] V.A. McKusick and S.E. Antonarakis. *Mendelian inheritance in man: a catalog of human genes and genetic disorders*. John Hopkins University Press, 1994.

[60] A.J. Menezes, P.C. Van Oorschot, and S.A. Vanstone. *Handbook of applied cryptography*. CRC, 1997.

[61] S.A. Migueles, M.S. Sabbaghian, W.L. Shupert, M.P. Bettinotti, F.M. Marincola, L. Martino, C.W. Hallahan, S.M. Selig, D. Schwartz, J. Sullivan, et al. HLA B* 5701 is highly associated with restriction of

virus replication in a subgroup of HIV-infected long term nonprogressors. *Proceedings of the National Academy of Sciences of the United States of America*, 97(6), 2000.

[62] National Center for Biotechnology Information (US). Single Nucleotide Polymorphism Database. http://www.ncbi.nlm.nih.gov/projects/SNP/.

[63] National Center for Biotechnology Information (US). TPMT thiopurine S-methyltransferase. http://www.ncbi.nlm.nih.gov/gene?term=thiopurine%20S-methyl%20transferase.

[64] National Center for Biotechnology Information (US). Restriction Fragment Length Polymorphism (RFLP). http://www.ncbi.nlm.nih.gov/projects/genome/probe/doc/TechRFLP.shtml, 2011.

[65] NCBI. Genome Mapping. http://www.ncbi.nlm.nih.gov/projects/genome/probe/doc/ApplMapping.shtml, 2011.

[66] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *EUROCRYPT*, pages 223–238, 1999.

[67] A. Prat and J. Baselga. The role of hormonal therapy in the management of hormonal-receptor-positive breast cancer with co-expression of her2. *Nat Clin Prac Oncol*, 5(9):531–542, 2008.

[68] ScientificMatch.com. scientificmatch.com, 2011.

[69] R. F. Service. The race for the $1000 genome. *Science*, 311:1544–1546, 2006.

[70] N. Siva. 1000 Genomes project. *Nature biotechnology*, 26(3), 2008.

[71] T. Smith and M. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147:195–197, 1981.

[72] P.D. Stenson, M. Mort, E.V. Ball, K. Howells, A.D. Phillips, N.S.T. Thomas, and D.N. Cooper. The human gene mutation database: 2008 update. *Genome Medicine*, 1(1):1–6, 2009.

[73] The Federal Bureau of Investigation. Combined DNA Index System (CODIS). http://www.fbi.gov/about-us/lab/codis, 2011.

[74] T. Tokino, E. Takahashi, M. Mori, A. Tanigami, T. Glaser, JW Park, C. Jones, T. Hori, and Y. Nakamura. Isolation and mapping of 62 new RFLP markers on human chromosome 11. *American journal of human genetics*, 48(2), 1991.

[75] J.R. Troncoso-Pastoriza, S. Katzenbeisser, and M. Celik. Privacy preserving error resilient dna searching through oblivious automata. In *CCS*, pages 519–528, 2007.

[76] J. Vaidya and C. Clifton. Secure set intersection cardinality with application to association rule mining. *Journal of Computer Security*, 13(4):593–622, 2005.

[77] M. Wadman. Genetics bill cruises through senate. *Nature*, 453, 2008.

[78] J. Wang, W. Wang, R. Li, Y. Li, G. Tian, L. Goodman, W. Fan, J. Zhang, J. Li, J. Zhang, et al. The diploid genome sequence of an Asian individual. *Nature*, 456(7218):60–65, 2008.

[79] R. Wang, Y.F. Li, X.F. Wang, H. Tang, and X. Zhou. Learning your identity and disease from research papers: information leaks in genome wide association study. In *CCS*, pages 534–544, 2009.

[80] R. Wang, X. Wang, Z. Li, H. Tang, M.K. Reiter, and Z. Dong. Privacy-preserving genomic computation through program specialization. In *CCS*, pages 338–347, 2009.

[81] R.H. Waterston, K. Lindblad-Toh, E. Birney, J. Rogers, J.F. Abril, P. Agarwal, R. Agarwala, R. Ainscough, M. Alexandersson, P. An, et al. Initial sequencing and comparative analysis of the mouse genome. *Nature*, 420(6915):520–562, 2002.

[82] A.D. Weston and L. Hood. Systems biology, proteomics, and the future of health care: toward predictive, preventative, and personalized medicine. *Journal of proteome research*, 3(2):179–196, 2004.

[83] D.A. Wheeler, M. Srinivasan, M. Egholm, Y. Shen, L. Chen, A. McGuire, W. He, Y.J. Chen, V. Makhijani, G.T. Roth, et al. The complete genome of an individual by massively parallel DNA sequencing. *Nature*, 452(7189):872–876, 2008.

[84] A.C. Yao. Protocols for secure computations. In *FOCS*, pages 160–164, 1982.

[85] C.R. Yates, E.Y. Krynetski, T. Loennechen, M.Y. Fessing, H.L. Tai, C.H. Pui, M.V. Relling, and W.E. Evans. Molecular diagnosis of thiopurine S-methyltransferase deficiency: genetic basis for azathioprine and mercaptopurine intolerance. *Annals of internal medicine*, 126(8), 1997.

[86] X. Zhou, B. Peng, Y.F. Li, Y. Chen, H. Tang, and X.F. Wang. To Release Or Not To Release: Evaluating Information Leaks in Aggregate Human-Genome Data. In *ESORICS*, pages 607–627, 2011.