Secure Paternity Testing with Homomorphic Encryption

Nicole Anne Balde and Richard Bryann Chua*

Department of Physical Sciences and Mathematics,
University of the Philippines Manila, Manila, Philippines
{nibalde,rlchua}@up.edu.ph

Abstract. With the rise of consumer-centric genomic testing services, there is an increased risk of data breaches and privacy concerns regarding sensitive genomic data. Paternity testing, one of the most common genetic tests, is no exception to this risk. To address this privacy risk, Fully Homomorphic Encryption (FHE) offers a viable approach to preserve privacy while allowing computations to be performed directly on encrypted data. In our work, we implemented a secure paternity testing system with FHE using the CKKS scheme to allow for a secure, privacy-preserving genomic computations. Results show that using a large scaling factor yields a higher accuracy but resulted in a slight increase in runtime. A key contribution of our work is the successful implementation of a single-key FHE paternity testing system, which is more lightweight compared to multi-key FHE systems since it requires less computational resources and memory overhead. This demonstrates that it is possible to perform secure paternity testing while maintaining the accuracy and privacy of sensitive genomic data.

Keywords: Fully Homomorphic Encryption, Paternity Testing, Encrypted Computation, Genomic Computation

1 Introduction

With the rapid development in genome sequencing technologies, genetic analysis costs have dropped dramatically [24], resulting in increased availability of services that provide direct-to-consumer genetic testing (DTC-GT). These consumer-centric services allow consumers to request tests directly from providers and offer a variety of products such as ancestry or genealogy, paternity, and genetic predisposition to certain health conditions, opening up discussions about the possibility of a more personalized and proactive approach to one's health [25]. With DTC-GT being consumer-centric, comparatively cheaper than conventional testing, and with its non-invasive procedure for sample collection, such as saliva kits, the global estimate for the DTC-GT market in 2017 was around 359 million, with more than 12 million consumers using the services by then [14]. One of the more well-known tests involving DNA, paternity testing, has also seen an observable increase in its use for personal reasons since the early 2000s [1].

J. Caro et al. (eds.), *Proceedings of the Workshop on Computation: Theory and Practice (WCTP 2025)*, Atlantis Highlights in Computer Sciences 24,
https://doi.org/10.2991/978-94-6239-638-8_29

With the amount of information held within a person's DNA, it becomes important to ensure that this data is kept secure and private. Any compromised data can be used to identify possible relatives and serve as an identifier for the consumer who requested the service [27]. How each company secures its data varies and the risks of data breaches pose a great concern to many consumers of DTC companies. For example, in 2023, 23andMe experienced a data breach spanning 5 months which exposed the information of nearly half of the company's clients accounting for about 7 million people [9]. Hackers were reported to have accessed 14,000 accounts and were able to steal the data of 6.9 million people through the DNA relatives feature offered by the company [13].

These risks highlight the need for more secure methods to protect genomic data. One promising solution to this is the use of a cryptographic technique known as Fully Homomorphic Encryption (FHE) [20]. In our work, we used fully homomorphic encryption to perform the paternity testing computations on encrypted data using the CKKS scheme as a solution to the privacy and security concerns surrounding DTC-GT services. Compared to previous works like [23] and [30] which primarily used a multi-key FHE framework for genomic testing and determining whether a possible kin exists in a given database, our work differs through its use of a single-key FHE implementation for paternity testing. The use of a single-key implementation allows for the creation of a lightweight application as there is lower computational costs and no additional memory overhead needed to maintain and manage multiple keys.

The rest of the paper is structured as follows: Section 2 presents the literature review, where we looked at relevant studies. We provide information on homomorphic encryption in Section 3 and paternity testing computation in Section 4. We discussed how we used FHE in paternity testing in Section 5.2 and evaluated our implementation in Section 6. Section 7 concludes the paper and provides potential direction for future works.

2 Literature Review

Many types of genomic testing services are currently marketed under the DTC-GT model. Phillips [27] identified nine common genetic tests offered by DTC companies: health testing, carrier testing, nutrigenetic testing, ancestry, genetic relatedness, athletic ability, child talent, surreptitious testing (infidelity tests) and matchmaking. Of these, the most common and well-known genetic test is the ancestry and genetic relatedness test, particularly the paternity test [32, 18, 33]. It is estimated that the global market for DTC-GT by 2027 will reach $2.3 billion with the ancestry and relationship testing portion reaching $960.8 million by the end of the 2020-2027 analysis period [28]. DTC-GT services have generally been received positively. Lee et al. [18] found that their use revolves around the search for biological relatives, confirming their ethnicity and ancestry and gaining information regarding their health and genetic dispositions [21, 3]. However, concerns regarding the privacy and security of consumers' genetic information have also grown alongside the industry's expansion and rise. While

attempts at securing personal information is possible, recent studies have shown that it is possible to re-identify individuals even from large anonymized datasets. Holthouse et al. in [16] pointed out on the recent 23andMe data breach that data-rich environments such as that of DTC-GT platforms often present themselves as highly attractive targets. With features such as the DNA Relative and Family Tree offered by the company, profiles are interconnected with each other based on the results of the relation tests. This interconnectedness poses a significant security risk as even a small amount of compromised data under the hands of an attacker can yield significant amounts of data from the profiles connected to the compromised users.

There are several studies exploring the use of FHE in genomic analysis specifically in genomic relatedness. Namazi et al. in [23] proposed a privacy-preserving multi-key homomorphic framework for genomic computations that can be used in a variety of genomic tests such as personalized medicine, paternity testing, similar patient search and record linkage. de Souza et al. proposed a homomorphich encryption-based solution for the iDASH 2023 competition [30] which focused on the private detection of relatives. Their work focused on checking if a relative exists in the database and not the identification or retrieval of the match.

3 Homomorphic Encryption

In traditional cryptography, data is secured when it is moved from one location to another (in transit) or when it is stationary in one location (at rest). While this has its own value in practice, it does not secure data when it is of its highest value, which is when it is being used. This has led to the development of technologies aimed to resolve this issue dubbed computing on encrypted data (COED) [29]. One of these technologies is Fully Homomorphic Encryption.

Homomorphic encryption was first introduced in 1978 by Ronald Rivest, Len Adelman, and Michael Dertouzos when they observed that two encrypted numbers under RSA encryption can be multiplied and return a result that is equivalent to the sum of its plaintext counterparts when decrypted. It was the first instance of a partially homomorphic encryption scheme [8]. Following the introduction of homomorphic encryption, early developments in the field revolved around two types of HE: partially HE and somewhat HE. The first allowed for an unlimited number of times for one type of operation to be performed. The latter, on the other hand, allowed for some operations a limited number of times [2]. It was not until 2009 that Craig Gentry proposed the first plausible Fully Homomorphic Encryption scheme, which supported an unlimited number of operations for an unlimited number of times, using lattice-based cryptography. This proposal served as an important breakthrough in the field of homomorphic encryption. Further developments following Gentry' work were made yielding more practical and efficient schemes. Some of these advancements can be seen in the Brakerski/Fan-Vercauteren (BFV) and Brakerski-Gentry-Vaikuntantan (BGV) schemes, which display a higher efficiency and reduced noise growth [17]. The

Gentry-Sahai-Waters (GWS) scheme avoids the use of relinearization, which is computationally expensive and also has slower noise growth [8]. In 2017, the Cheon-Kim-Kim-Song (CKKS) scheme was introduced and was the first FHE scheme to allow approximate arithmetic over complex numbers [7].

4 Paternity Testing

Paternity testing is a DNA test performed to determine the biological father of a person. There are three standard methods used in paternity testing: polymerase chain reaction (PCR), restriction fragment length polymorphism (RFLP) and short tandem repeat (STR). In the PCR method, special fluorescent tags are used to enable gene detection. In the RFLP method, DNA restriction endonucleases are used to cut the isolated sample DNA into fragments. These fragments are then separated by size using an electric current and are identified using DNA probes. In the STR method, STR markers are used to distinguish one individual from another using the variability in the STR regions [21].

Short Tandem Repeats (STR) are short repeating units of DNA sequences of varying lengths scattered across the human genome which vary greatly from individual to individual [15]. These sequences account for approximately 3% of the human genome [11] and are located in regions known as STR loci. For a given person, the amount of repeat units contained within an STR locus are referred to as an allele and is numerically designated by a floating point number. In this floating point number, the number of repeated units is indicated at the left of the decimal point while the number of partial repeat units is indicated at the right of the decimal point [4]. Partial repeat units are units that are not full copies of the repeated motif. If there are no partial repeat units, the allele is reported as an integer value since the number at the right of the decimal place is 0. For example, given $(AATG)_6(\text{-}ATG)(AATG)_3$ in the STR locus TH01, there are a total of nine full repeat units of AATG and one partial repeat unit (ATG) containing only three of the four bases resulting in an allele designation of 9.3 in the TH01 STR marker [5].

Alleles at various STR loci are inherited in the same way as any other Mendelian genetic marker in which each two-diploid parent contributes one of their two alleles to their offspring. One allele is donated by the mother and one allele is donated by the father on each STR locus, producing a diploid genotype that is represented as a pair of values for the given STR locus [22]. For instance, a person who has allele designations 10 and 15 at position D7S820 would have received the first allele from one parent and the second allele from the other parent. Thus, the hypothetical person would have an STR profile which has (10,15) at the STR marker D7S820. Due to its high variability, it is highly useful for identification, kinship determination and criminal investigations [26].

In a paternity test between a child and an alleged father with use of STR markers [10, 6], a paternity index (PI) is computed using the alleles for each STR

loci through the likelihood ratio:

$$PI = \frac{P(\text{genotypes} \mid \text{alleged father})}{P(\text{genotypes} \mid \text{random man})}, \tag{1}$$

where $P(\text{genotypes} \mid \text{alleged father})$ is the probability of observing the genotype of the child given that the alleged father is the true father and $P(\text{genotypes} \mid \text{random man})$ is the probability of having the same genotype if the allele of the child came from a random, unrelated man in the population.

The calculation for the paternity index is dependent on the pattern of inheritance. Given A, B, C and D which are possible allele values and probability p which denotes the allele frequency in a trio case wherein the genotypes of the mother, alleged father and child are known, the values of the numerator and denominator may fall under one of the cases identified in Table 1 [31].

Table 1. Numerator and Denominator values for PI Calculation

Mother	Child	Alleged father	Numerator	Denominator
AA	AA	AA	1	p_A
AA	AA	AB	$\frac{1}{2}$	p_A
AA	AA	BC	0	p_A
AB	AA	AA	$\frac{1}{2}$	$\frac{p_A}{2}$
AB	AA	AB	$\frac{1}{4}$	$\frac{p_A}{2}$
AB	AA	AC	$\frac{1}{4}$	$\frac{p_A}{2}$
AB	AA	BC	0	$\frac{p_A}{2}$
AA	AB	AB	$\frac{1}{4}$	$\frac{p_B}{2}$
AA	AB	BB	1	p_B
AA	AB	BC	$\frac{1}{2}$	p_B
AA	AB	CD	0	p_A
AB	AB	AA	$\frac{1}{2}$	$\frac{p_A+p_B}{2}$
AB	AB	AB	$\frac{1}{2}$	$\frac{p_A+p_B}{2}$
AB	AB	BC	$\frac{1}{4}$	$\frac{p_A+p_B}{2}$
AB	AB	AC	$\frac{1}{4}$	$\frac{p_A+p_B}{2}$
AB	AC	AC	$\frac{1}{2}$	p_C
AB	AC	CD	$\frac{1}{4}$	$\frac{p_C}{2}$
AB	AC	BC	$\frac{1}{4}$	$\frac{p_C}{2}$
AB	BC	CC	$\frac{1}{2}$	$\frac{p_C}{2}$
AB	BB	AB	$\frac{1}{4}$	$\frac{p_B}{2}$
AB	BC	BC	$\frac{1}{2}$	p_C
AB	BC	CD	$\frac{1}{4}$	$\frac{p_C}{2}$
AB	AB	CD	0	$\frac{p_A+p_B}{2}$
AC	AB	BB	$\frac{1}{2}$	$\frac{p_B}{2}$
AC	AB	BD	$\frac{1}{4}$	$\frac{p_B}{2}$
AC	AB	BC	$\frac{1}{4}$	$\frac{p_B}{2}$
AC	AB	CD	0	$\frac{p_B}{2}$

For example, consider locus, D8S1179, where the alleles of the mother is (10, 12), the child is (10, 14) and the alleged father, a Caucasian, is (14, 17). This example is shown in Table 2. To compute for the paternity index, the pattern of

Table 2. Sample Pattern of Allele Distribution

Individual	**Mother**	**Child**	**Alleged father**
D8S1179 Alleles	10,12	10,14	14,17
Allele Pattern	AB	AC	CD

inheritance must be identified first. Using Table 1, we can see that the pattern of inheritance for this example uses $1/4$ as the numerator and $p_C/2$ as the denominator where p_C is the allele frequency of the alleged father's first allele. Within the Caucasian population, the allele frequency of allele 14 is 0.1089 [12] and the paternity index is computed as follows:

$$PI = \frac{\frac{1}{4}}{\frac{0.1089}{2}} = \frac{0.25}{0.05445} = 4.591368. \tag{2}$$

If the mother's genotype is unknown or is not available, based on the pattern of inheritance, its corresponding PI computation formula may fall under one of the cases specified in Table 3 [31]. For mismatches wherein no alleles between

Table 3. PI Calculation For Duo (Unknown Mother) Cases

Child	**Alleged father**	**PI Formula**
AA	AA	$\frac{1}{p_A}$
AA	AB	$\frac{1}{2p_A}$
AB	AA	$\frac{1}{p_A}$
AB	AB	$\frac{(p_A+p_B)}{4(p_A \times p_B)}$
AB	BB	$\frac{1}{2p_B}$
AB	BC	$\frac{1}{4p_B}$

the child and the alleged father are shared, the PI value is set to zero [19]. As an example of a computation where the genotype of the mother is not available, consider the same locus, D8S1179, where the alleles of the child is (8, 13) and the alleged father, a Caucasian, is (13, 13). Table 4 reveals the pattern of inheritance identified for the given example. Using Table 3 as reference for the case formula, we can see that the pattern of inheritance identified in Table 4 for the given example uses $1/(2p_B)$ where p_B is the allele frequency of the alleged father's first and second allele. Within the Caucasian population, the allele frequency of

Table 4. Pattern of Allele Distribution for Sample Motherless Case

Individual	Child	Alleged father
D8S1179 Alleles	8,13	13,13
Allele Pattern	AB	BB

allele 13 is 0.3342 [12] and the paternity index is computed as follows:

$$PI = \frac{1}{(2)(0.3342)} = \frac{1}{0.6684} = 1.4961 \tag{3}$$

This computation is repeated over all STR markers being considered for the testing. In the FBI CODIS, the number of STR markers used for paternity testing is 20 [34, 12]. Once all PIs have been calculated, the product of the computed individual PIs across all STR markers is then calculated to get the combined paternity index (CPI):

$$CPI = \prod_{i=1}^{k} \mathrm{PI}_i. \tag{4}$$

Using the minimum standard for paternity inclusion, a score of 100 or greater is considered a possible biological father [6]. Implementing these formulas using fully homomorphic encryption, however, is difficult as the computation for the paternity index in each STR marker primarily uses conditional statements in determining the numerator and the denominator of the formula based on which case it falls under as listed in Table 1 or, in the case of Table 3, which PI formula to use. The division operation is also not supported in FHE schemes. Additional algorithms must be used to implement these operations which increases the overhead costs in the computation.

5 Homomorphic Paternity Testing

Namazi et al. in [23] proposed an FHE-compatible method of computing for the paternity index shown in Equation 5.

$$S_{PT} = \frac{1}{k} \cdot \sum_{i=1}^{k} f(x_{i,1},\, x_{i,2},\, x'_{i,1},\, x'_{i,2}) \tag{5}$$

where k denotes the number of STR markers being analyzed. $x_{i,1}$ and $x_{i,2}$ denote the alleles of the child at the ith STR marker while $x'_{i,1}$ and $x'_{i,2}$ denote the alleles of the alleged father at the same marker. The comparison of the STR markers is computed through the function f:

$$f(x_{i,1},\, x_{i,2},\, x'_{i,1},\, x'_{i,2}) = (x_{i,1} - x'_{i,1})(x_{i,1} - x'_{i,2})(x_{i,2} - x'_{i,1})(x_{i,2} - x'_{i,2}) \tag{6}$$

The function f evaluates the difference between the alleles of the child ($x_{i,1}$ and $x_{i,2}$) and the alleles of the alleged father ($x'_{i,1}$ and $x'_{i,2}$) for each of the

markers. The final score is the average of the computed comparisons over the k number of STR markers. In the FHE formula, the comparison function measures the similarity of the alleles of the alleged father and the child. In one STR marker, if both alleles match in at least one of the four subtraction terms, the difference yields a value of zero rendering the product for the four terms in that marker also zero. The more matches between the alleged father and child, the more instances of the function evaluating to zero which also lowers the overall score when the summation and averaging is applied. As such, a score closer to zero indicates higher similarity between the alleles of the alleged father and child which potentially indicates higher likelihood of paternity.

5.1 Dataset

We used a synthetic dataset generated using the allele frequencies provided in the 2015 FBI expanded CODIS Population Data and the 20 specified STR markers in the set composed of the original set with 13 STR loci and an additional 7 STR loci which is available at: https://ucr.fbi.gov/lab/biometric-analysis/codis/expanded-fbi-str-2015-final-6-16-15.pdf. The allele values were sampled from the documented allele frequency distributions of the Filipino population in Guam for 20 STR loci of the FBI extended CODIS and a dictionary containing said allele frequencies for each STR marker was used as the probability space for generating the alleles. Each locus had two alleles sampled independently to simulate the diploid nature of human DNA, wherein one allele is transmitted by each biological parent. The allele values generated represent the number of short tandem repeat (STR) units at each marker.

5.2 Homomorphic Paternity Testing Implementation

Our secure paternity testing system is composed of a client and a server. The client submits the encrypted STR profile of the child to the server. The server homomorphically computes the paternity scores between the encrypted STR profile of the child and the STR profiles of all father candidates in its database and returns the encrypted paternity scores to the client.

We implemented the Namazi et al. FHE paternity testing formula using the TenSEAL 0.3.16 and Python 3.12. TenSEAL is a Python library that provides a high-level interface for homomorphic encryption operations, particularly for the CKKS scheme. The following parameters were used in the CKKS scheme context of TenSEAL:

- Security Level: 128 bits
- Polynomial modulus degree: 16384, which provides 8192 slots for single instruction multiple data (SIMD) packing.
- Coefficient modulus chain: List of six prime sizes were used $[60, s, s, s, s, 60], s \in \{30, 40, 50, 60\}$. s was chosen to match the scaling factor used in the context. We tried different s in order to look at the effect of adjusting the scaling factor.

- Multiplicative depth: 4, since we are evaluating a product of four terms per marker which requires three multiplicative levels.
- Scaling factor: $2^s, s \in \{30, 40, 50, 60\}$, in order to test the performance of different scaling factors.

On the client side, the two allele values of the STR markers for each child query were first extracted as vectors from the query dataset and encoded using the CKKS scheme. Specifically, for every child, the STR marker values were represented by two alleles for every marker: one for Allele 1 (represented by A1) and another for Allele 2 (represented by A2). These allele values were stored as floating-point numbers. The child's A1 and A2 values for each marker were then independently replicated into a vector whose size is equal to the number of father candidates in the database. This allows one operation to be performed between the child vector containing the STR profile and all father profiles simultaneously. For example, if a child STR profile has three STR markers – D3S1358, D8S1179, and FGA – with A1 and A2 having the following allele values: (13, 14), (11, 12), and (20, 21) and there are four candidates for the father in the database, then a total of six vectors is created for the three STR markers:

- D3S1358_A1: [13.0, 13.0, 13.0, 13.0]
- D3S1358_A2: [14.0, 14.0, 14.0, 14.0]
- D8S1179_A1: [11.0, 11.0, 11.0, 11.0]
- D8S1179_A2: [12.0, 12.0, 12.0, 12.0]
- FGA_A1: [20.0, 20.0, 20.0, 20.0]
- FGA_A2: [21.0, 21.0, 21.0, 21.0]

These vectors are encrypted resulting in two encrypted vectors (A1 and A2) per marker per child. These encrypted vectors are then sent to the server for homomorphic computation.

On the server side, the homomorphic computation of the paternity scores is performed. The STR values of the father candidates is retrieved and cached in plaintext as arrays for quick access and computation. Specifically, the server extracts two arrays corresponding to each of the alleged fathers STR marker (A1 and A2 alleles) and stores them in the memory. In this case, given one STR marker, one array holds all A1 allele values across the father candidates present in the database for said STR marker while another array holds all A2 allele values for the same individuals for the same marker. These plaintext arrays are subsequently used in the homomorphic computation phase, in which they are "compared" against the encrypted child data received from the client. This enables FHE backend to perform the vectorized, parallel computations for the paternity testing.

With the encrypted child data and the stored plaintext father data in the server, the computation was performed in the server following the proposed paternity formula with the main function — comprised of the subtraction, multiplication and summation operations — done under FHE. Specifically, upon receiving the encrypted query data, the four subtraction terms were first calculated by getting the pairwise differences between each possible allele pairing for

the child and father in a given STR marker. Given the two child alleles ($x_{i,1}$ and $x_{i,2}$) and two father alleles ($x'_{i,1}$ and $x'_{i,2}$) in an STR marker i, the differences are computed via the following pairings:

- first child allele and first father allele: $(x_{i,1} - x'_{i,1})$
- first child allele and second father allele: $(x_{i,1} - x'_{i,2})$
- second child allele and first father allele: $(x_{i,2} - x'_{i,1})$
- second child allele and second father allele: $(x_{i,2} - x'_{i,2})$

The subtraction operation was performed element-wise using the encrypted child vectors and the father vectors. All four differences were then multiplied to get the product for the given STR marker. For example, given a child (C) with A1 and A2 allele values (9, 11) in the D16S539 STR Marker and three father candidates (FC) in the database with the allele values (11, 13), (9, 10) and (12, 14) respectively, the resulting vectors would be as shown in Table 5. Using these vectors, the differences in the pairwise subtraction, denoted by T_n in the example where n is the index of the subtraction term, are computed as follows:

Table 5. Sample Vectors for one STR marker

	Child (C)	**Father Candidates (FC)**
A1	[9.0, 9.0, 9.0]	[11.0, 9.0, 12.0]
A2	[11.0, 11.0, 11.0]	[13.0, 10.0, 14.0]

$$T_1 = C_{A1} - FC_{A1} = [9.0, 9.0, 9.0] - [11.0, 9.0, 12.0] = [-2, 0, -3]$$
$$T_2 = C_{A1} - FC_{A2} = [9.0, 9.0, 9.0] - [13.0, 10.0, 14.0] = [-4, -1, -5]$$
$$T_3 = C_{A2} - FC_{A1} = [11.0, 11.0, 11.0] - [11.0, 9.0, 12.0] = [0, 2, -1]$$
$$T_4 = C_{A2} - FC_{A2} = [11.0, 11.0, 11.0] - [13.0, 10.0, 14.0] = [-2, 1, -3]$$

Multiplying the differences in all four subtraction terms yields the following product for the D16S539 STR marker:

$$Product_{D16S539} = T_1 * T_2 * T_3 * T_4 = [0, 0, 45]$$

Each element in the resulting vector corresponds to the computed product for one father and one marker. In the given example, the resulting product vector shows that the product for the D16S539 marker is zero for the child and first father pair, zero for the child and second father pair and 45 for the child and third father pair. This step is repeated until all STR marker products have been computed. Once all products have been calculated, the products are then added to get the FHE computed score for the given child and their subsequent father pairs. This three-operation process is repeated until the FHE computed

scores for all child-father pairs have been computed. Since FHE cannot perform the division operation without an approximation algorithm, the final step is performed on the client-side after decryption. Once all FHE computations were completed, the resulting encrypted paternity scores for each child-father pair was returned by the server to the client and was decrypted in the client-side. The decrypted scores was then divided by the number of STR markers to attain the final paternity score for each child-father pair.

6 Evaluation

In our evaluation, we used a database containing 2000 father STR profiles and a query containing 200 child STR profiles. This generated a total of 400,000 computed paternity scores. We evaluated the performance of the FHE paternity testing system by measuring the accuracy of the decrypted paternity scores compared to their expected plaintext values and the runtime of the system.

6.1 Accuracy Analysis

We measured the accuracy of the decrypted paternity scores under each scaling factor by comparing them to the expected plaintext paternity scores. Four thresholds based on the number of decimal places were used to check for the accuracy of the decrypted scores. Table 6 summarizes the accuracy of the computations over different scaling factors. Of the four scaling factors, 2^{50} and 2^{60}

Table 6. Accuracy of encrypted paternity scores at different scaling factors and precision thresholds.

Global Scale	**Accuracy (%)**			
	0.1	**0.01**	**0.001**	**0.0001**
2^{30}	97.7532	45.6683	4.2462	0.4062
2^{40}	100	99.9950	97.9225	47.2473
2^{50}	100	100	100	100
2^{60}	100	100	100	100

have the highest accuracies based on the four thresholds. Extending the threshold further reveals the following accuracies for the scores computed using a scaling factor of 2^{50}:

- 1st to 4th decimal place:100%
- 5th decimal place: 99.9980%
- 6th decimal place: 98.3902%
- 7th decimal place: 52.4967%

For the 2^{60} scaling factor, the accuracies with the extended thresholds are as follows:

- 1st to 8th decimal place:100%
- 9th decimal place: 99.9843%
- 10th decimal place: 95.9740%
- 11th decimal place: 25.6928%

This suggests that the use of larger scaling factors allows for better precision in the results.

6.2 Runtime Analysis

The 400,000 computed paternity scores were returned by the server in about 350 seconds (5.8 mins). Using an initial scaling factor of 2^{40}, it took 125.61 seconds to encrypt the query, 218.96 seconds to perform all computations and 0.47 seconds to decrypt the encrypted paternity scores. The runtime for each child, on average, took 1.72 seconds with the lowest runtime taking 1.71 seconds and the highest runtime taking 1.82 seconds. Tables 7 and 8 summarizes the runtime breakdown of the computation over different scaling factors.

Table 7. Encryption, Decryption and Computation Time of the FHE paternity testing system at various scaling factors

Global Scale	Encryption Time (s)	Computation Time (s)	Decryption Time (s)
2^{30}	125.63	218.68	0.48
2^{40}	125.61	218.96	0.47
2^{50}	125.68	218.98	0.48
2^{60}	130.20	219.47	0.48

Table 8. Runtime of the FHE paternity testing system at various scaling factors

Global Scale	Total Runtime (s)	Average Runtime per Child (s)
2^{30}	346.67	1.72
2^{40}	346.89	1.72
2^{50}	346.99	1.72
2^{60}	352.20	1.75

Among the four scaling factors, the decryption time across the scaling factors was observed to be minimal and was consistently under 1 second. In terms of encryption and computation times, the performance across all tested scales remained relatively stable with a marginal increase in the 2^{60} scaling factor. The runtime for each child, on average, remained within the 1.72 second average with the value showing a small increment at the 2^{60} scaling factor. The total runtime

also showed a slight increase across all scales with a relatively higher increase at the 2^{60} scaling factor. The higher values at the 2^{60} scaling factor may be attributed to the effect of larger scaling factors on the ciphertext size. Higher scaling factors offer more precision however, this requires more bits to represent the encoded values which results in larger ciphertext sizes. This contributes to the observed increase in the overall runtime as well as in the encryption and computation time. The growth in the ciphertext sizes can be seen in Table 9 which shows the average ciphertext sizes generated during the encryption phase of the test using the different scaling factors.

Table 9. Average Ciphertext size at various scaling factors

Global Scale	Average Ciphertext Size (KB)
2^{30}	33267.08
2^{40}	41155.92
2^{50}	47060.95
2^{60}	51206.72

7 Conclusion

We were able to use homomorphic encryption to implement the paternity testing system in a way that preserves the privacy of the individuals involved. The system allows for the secure computation of paternity scores without revealing sensitive genetic information, thus providing a practical solution for privacy-preserving paternity testing. With a larger scaling factor, we get a higher accuracy in the computed paternity scores, but this comes at the cost of longer runtimes and larger ciphertext sizes. This shows that the use of FHE in paternity testing is practical and feasible, but careful consideration of the parameters is necessary to balance accuracy and performance. Exploring other possible parameter adjustments and combinations as well as the application of other optimization techniques should also be considered and explored in future works.

References

1. ABC News: Paternity testing grows in popularity (2005). URL https://abcnews.go.com/GMA/story?id=1185107&page=1. Available at: https://abcnews.go.com/GMA/story?id=1185107&page=1
2. Acar, A., Aksu, H., Uluagac, A.S., Conti, M.: A survey on homomorphic encryption schemes: Theory and implementation. ACM Computing Surveys (CSUR) **51**(4), 1–35 (2018)
3. Baptista, N.M., Christensen, K.D., Carere, D.A., Roberts, J.S., Green, R.C., for the PGen Study Group: Adopting genetics: motivations and outcomes of personal genomic testing in adult adoptees. Genetics in Medicine **18**(9), 924–932 (2016). DOI 10.1038/gim.2015.192. URL https://doi.org/10.1038/gim.2015.192

4. Butler, J.M.: Genetics and genomics of core STR loci used in human identity testing. Journal of Forensic Sciences **51**(2), 253–265 (2006). DOI 10.1111/j.1556-4029.2006.00046.x. URL https://doi.org/10.1111/j.1556-4029.2006.00046.x
5. Butler, J.M.: Short tandem repeat (STR) loci and kits. In: Advanced Topics in Forensic DNA Typing, pp. 99–139. Elsevier (2012)
6. Butler, J.M.: Chapter 14 - relationship testing: Kinship statistics. In: J.M. Butler (ed.) Advanced Topics in Forensic DNA Typing: Interpretation, pp. 349–401. Academic Press (2015). DOI 10.1016/B978-0-12-405213-0.00014-2. URL https://doi.org/10.1016/B978-0-12-405213-0.00014-2
7. Cheon, J.H., Kim, A., Kim, M., Song, Y.: Homomorphic encryption for arithmetic of approximate numbers. In: Advances in Cryptology–ASIACRYPT 2017: 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, December 3-7, 2017, Proceedings, Part I, vol. 23, pp. 409–437. Springer International Publishing (2017)
8. Creeger, M.: The rise of fully homomorphic encryption: Often called the holy grail of cryptography, commercial FHE is near. Queue **20**(4), 70 (2022). DOI 10.1145/3561800. 22 pages
9. DeGeurin, M.: Hackers got nearly 7 million people's data from 23andme. the firm blamed users in 'very dumb' move. The Guardian (2024). URL https://www.theguardian.com/technology/2024/feb/15/23andme-hack-data-genetic-data-selling-response
10. El-Alfy, S.H., Abd El-Hafez, A.F.: Paternity testing and forensic DNA typing by multiplex STR analysis using ABI PRISM 310 genetic analyzer. Journal of Genetic Engineering and Biotechnology **10**(1), 101–112 (2012). DOI 10.1016/j.jgeb.2012.05.001. URL https://doi.org/10.1016/j.jgeb.2012.05.001
11. Fan, H., Chu, J.: A brief review of short tandem repeat mutation. Genomics, Proteomics & Bioinformatics **5**(1), 7–14 (2007). DOI 10.1016/S1672-0229(07)60009-6
12. FBI Extended Codis Population Data: Expanded STR population data 2015 (2016). URL https://ucr.fbi.gov/lab/biometric-analysis/codis/expanded-fbi-str-2015-final-6-16-15.pdf
13. Franceschi-Bicchierai, L.: 23andme admits it didn't detect cyberattacks for months. TechCrunch (2024). URL https://techcrunch.com/2024/01/25/23andme-admits-it-didnt-detect-cyberattacks-for-months/
14. Garner, S.A., Kim, J.: The privacy risks of direct-to-consumer genetic testing: A case study of 23andme and ancestry. Washington University Law Review **96**(6), 1219–1270 (2019). URL https://openscholarship.wustl.edu/law_lawreview/vol96/iss6/6/
15. Gill, P., Bleka, O., Hansson, O., Benschop, C., Haned, H.: Forensic genetics: the basics. In: Forensic Practitioner's Guide to the Interpretation of Complex DNA Profiles, chap. 1. Elsevier (2020). DOI 10.1016/B978-0-12-820562-4.00009-2. URL https://doi.org/10.1016/B978-0-12-820562-4.00009-2
16. Holthouse, R., Owens, S., Bhunia, S.: The 23andme data breach: Analyzing credential stuffing attacks, security vulnerabilities, and mitigation strategies (2025). DOI 10.48550/arXiv.2502.04303. URL https://doi.org/10.48550/arXiv.2502.04303
17. Kim, A., Polyakov, Y., Zucca, V.: Revisiting homomorphic encryption schemes for finite fields. In: T. Iwata, Y. Sasaki, T. Peyrin, H.C. Wu (eds.) Advances in Cryptology – ASIACRYPT 2021: 27th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 6–10, 2021, Proceedings, Part III, *Lecture Notes in Computer Science*, vol. 13092, pp.

608–639. Springer-Verlag, Berlin, Heidelberg (2021). DOI 10.1007/978-3-030-92078-4_21
18. Lee, H., Vogel, R.I., LeRoy, B., Zierhut, H.A.: Adult adoptees and their use of direct-to-consumer genetic testing: Searching for family, searching for health. Journal of Genetic Counseling **30**(1), 144–157 (2021). DOI 10.1002/jgc4.1304. URL https://doi.org/10.1002/jgc4.1304
19. Lee, J.C.I., Tsai, L.C., Chu, P.C., Lin, Y.Y., Lin, C.Y., Huang, T.Y., Yu, Y.J., Linacre, A., Hsieh, H.M.: The risk of false inclusion of a relative in parentage testing - an in silico population study. Croat. Med. J. **54**(3), 257–262 (2013)
20. Li, J.: Security implications of direct-to-consumer genetic sevices. In: 2015 IEEE First International Conference on Big Data Computing Service and Applications, pp. 147–153 (2015). DOI 10.1109/BigDataService.2015.26
21. Ma, H., Zhu, H., Guan, F., Cherng, S.: Paternity testing. Journal of American Science **2**(4), 76–92 (2006). URL https://www.jofamericanscience.org/journals/am-sci/0204/12-0205-mahongbao-am.pdf
22. Montelius, K., Stenersen, M., Sajantila, A.: Disaster victim management: DNA identification. In: J. Payne-James, R.W. Byard (eds.) Encyclopedia of Forensic and Legal Medicine (Second Edition), second edition edn., pp. 262–267. Elsevier, Oxford (2016). DOI https://doi.org/10.1016/B978-0-12-800034-2.00236-6. URL https://www.sciencedirect.com/science/article/pii/B9780128000342002366
23. Namazi, M., Farahpoor, M., Ayday, E., Pérez-González, F.: Privacy-preserving framework for genomic computations via multi-key homomorphic encryption. Bioinformatics **41**(3) (2025). DOI 10.1093/bioinformatics/btae754. URL https://doi.org/10.1093/bioinformatics/btae754. Published January 31, 2025
24. National Human Genome Research Institute: The cost of sequencing a human genome (2023). URL https://www.genome.gov/about-genomics/fact-sheets/Sequencing-Human-Genome-cost. Available at: https://www.genome.gov/about-genomics/fact-sheets/Sequencing-Human-Genome-cost
25. National Human Genome Research Institute: Direct-to-consumer genetic testing FAQ for healthcare professionals (2023). URL https://www.genome.gov/For-Health-Professionals/Provider-Genomics-Education-Resources/Healthcare-Provider-Direct-to-Consumer-Genetic-Testing-FAQ. Available at: https://www.genome.gov/For-Health-Professionals/Provider-Genomics-Education-Resources/Healthcare-Provider-Direct-to-Consumer-Genetic-Testing-FAQ
26. Norrgard, K.: Forensics, DNA fingerprinting, and codis. Nature Education **1**(1), 35 (2008). URL https://www.nature.com/scitable/topicpage/forensics-dna-fingerprinting-and-codis-736/
27. Phillips, A.M.: Only a click away – DTC genetics for ancestry, health, love... and more: A view of the business and regulatory landscape. Applied and Translational Genomics **8**, 16–22 (2016). DOI 10.1016/j.atg.2016.01.001. URL https://doi.org/10.1016/j.atg.2016.01.001
28. Research and Markets: Global direct-to-consumer (DTC) genetic testing market research report 2021: Ancestry and relationship to account for $960 million of the total $2.3 billion - forecast to 2027 (2021). URL https://www.businesswire.com/news/home/20210713005608/en/Global-Direct-to-Consumer-DTC-Genetic-Testing-Market-Research-Report-2021-Ancestry-Relationship-to-Account-for-%24960-Million-of-the-Total-%242.3-Billion---Forecast-to-2027---ResearchAndMarkets.com
29. Smart, N.: Computing on encrypted data. IEEE Security & Privacy **21**(4), 94–98 (2023). DOI 10.1109/MSEC.2023.3279517

30. de Souza, F.D.M., de Lassus, H., Cammarota, R.: Private detection of relatives in forensic genomics using homomorphic encryption. BMC Medical Genomics **17**(273) (2024). URL https://bmcmedgenomics.biomedcentral.com/articles/10.1186/s12920-024-01288-4
31. Stephenson, F.H.: Forensics and paternity. In: Calculations for Molecular Biology and Biotechnology, pp. 423–446. Elsevier (2010)
32. Thelingwani, R.S., Jonhera, C.A., Masimirembwa, C.: Analysis of data and common mutations encountered during routine parentage testing in zimbabwe. Scientific Reports **14**(1), 1385 (2024). DOI 10.1038/s41598-024-51987-8. URL https://www.nature.com/articles/s41598-024-51987-8
33. Tiner, J.C., Mechanic, L.E., Gallicchio, L., Gillanders, E.M., Helzlsouer, K.J.: Awareness and use of genetic testing: An analysis of the health information national trends survey 2020. Genetics in Medicine **24**(12), 2526–2534 (2022). DOI 10.1016/j.gim.2022.08.023. URL https://doi.org/10.1016/j.gim.2022.08.023
34. Welch, L.A., Gill, P., Phillips, C., Ansell, R., Morling, N., Parson, W., Palo, J.U., Bastisch, I.: European network of forensic science institutes (ENFSI): Evaluation of new commercial STR multiplexes that include the european standard set (ESS) of markers. Forensic Sci. Int. Genet. **6**(6), 819–826 (2012)