Privacy-Preserving Matching of DNA Profiles

Fons Bruekers[1], Stefan Katzenbeisser[1,2], Klaus Kursawe[1], Pim Tuyls[1]

[1]Philips Research Europe, Information and System Security Group,
Eindhoven, The Netherlands
{fons.bruekers,klaus.kursawe,pim.tuyls}@philips.com

[2]Technische Universität Darmstadt, Security Engineering Group,
Darmstadt, Germany
skatzenbeisser@acm.org

May 8, 2008

Abstract

In the last years, DNA sequencing techniques have advanced to the point that DNA identification and paternity testing has become almost a commodity. Due to the critical nature of DNA related data, this causes substantial privacy issues. In this paper, we introduce cryptographic privacy enhancing protocols that allow to perform the most common DNA-based identity, paternity and ancestry tests and thus implement privacy-enhanced online genealogy services or research projects. In the semi-honest attacker model, the protocols guarantee that no sensitive information about the involved DNA is exposed, and are resilient against common forms of measurement errors during DNA sequencing. The protocols are practical and efficient, both in terms of communication and computation complexity.

1 Introduction

Since the first deployment of DNA tests for person identification in 1994, advances in forensic sciences have decreased the effort of collecting DNA samples to the point that DNA-based identification and paternity testing have almost become a commodity. Most western governments keep databases with DNA profiles of criminal offenders and suspects: In the US, the FBI maintains the CODIS system [14, 6], storing (as of February 2007) more than 4.3 million profiles; the British authorities alone maintain a database containing DNA samples of more than five percent of the population. Besides the professional law-enforcement domain, DNA-based tests are increasingly available in private life: fathership tests are offered for as little as \$150, online genealogy projects use DNA testing to derive family trees [11, 1], and large-scale scientific studies try to unravel the history of mankind (e.g., the GenoGraphic project launched by the National Geographic Society [24]). With upcoming technical advances, such as microarrays, the effort of collecting and processing DNA samples will further decrease to the point where dedicated and expensive lab equipment may become unnecessary. DNA tests may be offered by pharmacies or stores, or even performed at home.

Besides person identification, DNA-related data is increasingly used in health care to gain a precise diagnosis and optimize treatments. DNA may, for example, code a pre-disposition to develop a specific disease; knowledge of this genetic disposition allows for preventive measures. In addition, DNA information is used to test for drug allergies and estimate the individual success rate of a specific treatment.

Due to its dual role of simultaneously providing both privacy-sensitive health data and identification information, special care must be taken when DNA-related data is stored and processed for identification purposes. While person identification is usually done with a ‘Short Tandem Repeat’ (STR) profile extracted from parts of the DNA that are considered non-functional, it has turned out that potentially sensitive medical information can be derived from STRs as well. For example, [16] notes that particular STRs may be linked to genetic diseases, since they are closely located to a particular gene that is responsible for a genetic disorder. The results in [10] show how STRs can be used to analyze the genome in order to locate genes responsible for a particular disease. Further, a series of examples displaying the links between STRs used for forensics and diseases can be found in [5]. Given the current scientific knowledge on the human genome, it can thus not be excluded that DNA profiles intended solely for identification purposes reveal sensitive medical data as well.

This increases the already eminent privacy problem connected to the use of DNA for identification purposes. Apart from the fact that even samples collected for the sole purpose of person identification may contain highly sensitive medical information, the wide range of possibilities DNA identification offers, calls for a very careful policy on when to voluntarily reveal DNA-related information.

While there is little doubt that DNA forensics is a powerful tool for police investigations, the existence of DNA databases does create desires for use-cases that were not intended at the time of data collection; examples are known from various other domains, such as road taxing information (which is now used to track suspects) or Internet data retention (which was introduced as a tool against terrorism, and now is used to track peer to peer users). For DNA, first proposals have been issued to use genetic information to determine the likelihood of a person becoming a criminal [2]; for example, [17] quotes a US state senator arguing that DNA profiles may help predict which probationers will likely commit further crimes. A summary of privacy issues in forensic use of DNA can be found in [13]. Thus, we believe it is important to technically restrict the usage of DNA forensics to a minimum and to require a watchdog organization to prevent abuse of the data—for example, while the police may have a database of encrypted DNA, the watchdog organization may hold the keys necessary to make use of the data.

In addition to forensic uses, DNA testing can allow for useful services to end consumers and researchers. Already now, several providers offer DNA based ancestry discovery and paternity testing; more advanced services (e.g., exact determining the exact ethnicity of a person) are also becoming available. In the future, the number of such services is likely to increase, including for example DNA based health recommendations. While we focus on identity testing here, our techniques can also be applied in those settings.

Related Work on DNA protection. There has been a considerable interest in the protection of genomic sequences for research purposes. Traditionally, protection was mainly achieved through anonymization techniques (see [19] for an overview). However, it was recently shown in [20] that re-identification of anonymized records is possible with high probability in case the anonymization preserved genealogical relations. To provide better protection, cryptographic privacy enhancing technologies have been investigated recently. For example, [15] aims at supporting large-scale biomedical research projects performing frequency counts of mutations; in their approach, genomic sequences are encrypted using a homomorphic public-key encryption scheme and the queries are performed directly on encrypted data. In [25] the authors concentrate on running queries formulated as regular expressions obliviously on DNA data; the approach can be extended to allow privacy-preserving fuzzy string searching.

While the approaches described above were designed for the protection of full DNA sequences in a research setting, the authors of [3] considered the protection of forensic DNA databases: each entry in the database is encrypted with a key that is derived (e.g., by a Fuzzy Extractor [9, 18]) from the DNA sample itself. If a DNA sample of a suspect is to be tested against the entries in the database, a key is first extracted from the sample; the test proceeds by trying to decrypt each entry in the database with the derived key. A match is obtained if at least one entry in the database can successfully be decrypted. This approach has the disadvantage that it is only applicable to a forensic testing scenario and cannot easily be extended to other genealogical tests considered in the present paper. Similarly, the concept of negative databases [8] can be used to test a single profile against a database so that the content of the database cannot be efficiently enumerated. However, this approach cannot be extended in a straightforward way to handle error-prone profiles and more complex tests like parental tests.

Contribution. In the present paper we provide—to the best of our knowledge for the first time—efficient and practical privacy enhancing protocols that allow the secure matching of DNA profiles. In contrast to previous approaches, which considered protection of DNA sequences and databases, we aim at protection of STR profiles as they are currently used in forensics and genealogy. Our protocols support identity, parental and ancestor tests and thus cover a wide range of questions on person relationships, offering the possibility to implement privacy-enhanced Internet genealogy services or research projects.

In Section 2 we review Short Tandem Repeats, which are commonly used in forensic sciences to perform both paternity and identity tests. In Section 3 we present privacy-preserving protocols for the most common applications of DNA-based identity testing; furthermore we discuss their efficiency and privacy. Finally, Section 4 discusses active attackers and shows fundamental limits of STR privacy protection techniques.

2 Short Tandem Repeats and Identity Testing

Desoxyribo-Nucleic Acid (DNA) is found in basically every cell of a living organism and determines to a great extent its physical characteristics. DNA consists of complementary pairs of long strands of four different nucleotides (A,C,G,T); in total, human DNA consists of several billion nucleotides. Every person inherits half of its DNA from the father and half from the mother (except mitochondrial DNA and the male Y chromosome); siblings inherit different combinations from their parents and thus have different but related DNA.

Some parts of the DNA, which have no apparent functionality, are known to contain short sequences of nucleotides that repeat a number of times. This phenomenon is called a Short Tandem Repeat (STR). The actual number of repetitive nucleotides in a STR varies widely over the population and is thus useful for identification purposes. The length of individual STRs

Locus	Allele (Σ)	Repeat structure
TH01	...	...
	8	$[AATG]_8$
	8.3	$[AATG]_5ATG[AATG]_3$
	9	$[AATG]_9$
	9.3	$[AATG]_6ATG[AATG]_3$
	...	...

Table 1: Part of a specification how repetitive patterns are encoded as alleles (elements of Σ) for the locus TH01 [4].

can be determined using chemical analysis: the analyzed DNA molecule is cut with restriction enzymes in front of and immediately after the STR. Finally, the length of the obtained fragment (and thus the number of repetitions, called *alleles*) can be determined using gel electrophoresis. STRs appear at different positions (called *loci*) in the DNA molecule and can be analyzed by using different types of restriction enzymes [4].

For the 22 pairs of autosomal chromosomes, every STR locus appears twice, one originating from the father and one from the mother. Thus for these STRs we obtain two potentially different numbers of repetitions. For the two types of sex chromosomes (X and Y), the structures are different; when evaluating the Y chromosome, only one number of repeats is found. STRs are a common phenomenon: thousands of different STRs are known, but only few, the core STRs, are usually used for forensics. In the field of criminal forensics, the European *SGM plus* identification method uses 10 different STR loci and gender information, whereas the US CODIS system utilizes a set of 13 loci. For genealogy, usually 37 or 67 STRs on the Y-chromosome are utilized.

In the rest of the paper, we will denote a STR occurring twice in a cell as a multiset $\{x_1, x_2\}$ of two not necessarily distinct elements over a finite set $x_1, x_2 \in \Sigma$, where Σ is a set of symbols (alleles) coding possible repetition numbers. For notational compatibility, STRs on the Y-chromosome will be denoted as singleton sets $\{x\} \in \Sigma$. Note that it is possible for a STR that $x_1 = x_2$ if the number of repetitions in DNA material inherited from the father is identical to the number of repetitions in the material inherited from the mother. The set Σ includes small integers and a few fractional numbers, which are commonly used in forensic sciences to code incomplete repetition patterns; Σ is always a finite set, typically containing 50 to 100 different symbols. Table 1 illustrates for one locus (TH01) how repetitive DNA patterns are encoded as alleles in Σ.

Even though the number of repetitions in a single STR varies over the population, the distribution of alleles x and y is far from uniform over Σ; for each locus there are a few symbols that occur with overwhelming probability. It is known from large-scale statistical analysis over the DNA of the population [21] that one STR allele x, viewed as a random variable over Σ, contains about 2.5 bits of entropy.

In the rest of the paper, by imposing an arbitrary order on the N loci considered, we will denote the STR profile of a person by an N-tuple of multisets

$$S = \langle \{x_{1,1}, x_{1,2}\}, \{x_{2,1}, x_{2,2}\}, \ldots, \{x_{N,1}, x_{N,2}\} \rangle \,,$$

or an N-tuple of singleton sets

$$S = \langle \{x_1\}, \{x_2\}, \dots, \{x_N\} \rangle$$

in case of STRs on the Y chromosome. For a profile S, we will use the short notation $\langle \{s_{i,1}, s_{i,2}\} \rangle$ or $\langle \{s_i\} \rangle$, where $1 \le i \le N$ are indices of the loci; furthermore, we have $x_i, x_{i,j} \in \Sigma$.

In this paper, we present protocols that allow to test whether two or three STR profiles are 'related' without disclosing the profiles to each other (or more precisely without leaking any information on the individual profiles in an information theoretic sense or under computational assumptions). We consider the following major questions on person relationships, which can be effectively tested on STRs:

- **Identity testing.** In this scenario, two STR profiles $S = \langle \{s_{i,1}, s_{i,2}\} \rangle$ and $T = \langle \{t_{i,1}, t_{i,2}\} \rangle$ are available; e.g., one profile may come from a crime scene and one from a forensic database. The goal is to determine whether both profiles were taken from the same person. This is the case if the alleles for each locus are identical, i.e.,

  $$\bigwedge_{i=1}^{N} [\{s_{i,1}, s_{i,2}\} = \{t_{i,1}, t_{i,2}\}] = \text{TRUE}, \tag{1}$$

  where '$=$' denotes a binary operator testing equivalence of multisets.

  Due to the imperfection of the chemical process used to analyze DNA samples and infer the STR alleles, infrequent errors occur in STR profiles with a small probability. In addition, in the cell reproduction process infrequent mutations may occur, thereby interfering with the STR patterns. To account for these imperfections, identity tests usually allow a small number of mismatches at different loci. Instead of verifying the condition of Eq. (1) for each of the N loci, a match is already reported if they are satisfied on at least $N - t$ out of all N loci for a small number t. Note that the accuracy of the test degrades significantly with a growing number t of errors. Thus, as the total number N of tested loci is already extremely limited, identity tests usually do not allow more than two errors in order to allow reliable identification. The protocols presented in this paper are designed to support this level of error-resilience.

- **Common ancestor testing on the Y chromosome.** In this scenario, two Y-chromosome STRs $S = \langle \{s_i\} \rangle$ and $T = \langle \{t_i\} \rangle$ are available, e.g., one may be stored in an online genealogy database and one may be possessed by a person who wishes to determine his ancestry. Due to the stability of the Y chromosome during reproduction, the persons from which the profiles S and R are taken are considered to be related, if they share the same Y chromosome and thus have the same STR profile. If there is a distant relationship between S and R, some STR alleles may have changed due to mutations during reproduction. Thus, a positive test result is reported if the STRs agree on all but at most t loci, where t usually does not exceed three:

  $$\bigvee_{C \subseteq \{1,\dots,N\}, |C| \ge N-t} \quad \bigwedge_{i \in C} [\{s_i\} = \{t_i\}] = \text{TRUE}. \tag{2}$$

- **Paternity testing with one parent.** Two profiles $S = \langle\{s_{i,1}, s_{i,2}\}\rangle$ and $T = \langle\{t_{i,1}, t_{i,2}\}\rangle$ are available. The goal is to determine whether T is a profile of a person that can potentially be a parent of the person from which S was taken. As noted above, during the reproduction process, for each locus one STR allele is inherited from the parent. For testing a parent-child relationship, it thus suffices to determine whether

  $$\bigwedge_{i=1}^{N} [\{s_{1,i}, s_{2,i}\} \cap \{t_{1,i}, t_{2,i}\} \neq \emptyset] = \text{TRUE}, \tag{3}$$

  where '$\cap$' denotes the multiset intersection operation. Again, in order to enhance the robustness of the test, a limited number t of mismatches at different loci may be allowed; in that case, the logical operator $\bigwedge_{i=1}^{N}$ should be replaced (similar to the last case) by $\bigvee_{C\subseteq\{1,\dots,N\},|C|\geq N-t} \bigwedge_{i\in C}$.

- **Paternity testing with two parents.** In this case three profiles $M = \langle\{m_{i,1}, m_{i,2}\}\rangle$, $F = \langle\{f_{i,1}, f_{i,2}\}\rangle$ and $C = \langle\{c_{i,1}, c_{i,2}\}\rangle$ are available. The task is to determine whether profiles M and F come from persons who can be the parents of a person with profile C. This is indeed the case if for each locus in C, one allele comes from M and the *other* allele comes from F. Thus, M and F can be parents of C, if

  $$\begin{aligned} & \bigwedge_{i=1}^{N} [\{c_{i,1}, c_{i,2}\} \in (\{m_{i,1}, m_{i,2}\} \boxtimes \{f_{i,1}, f_{i,2}\})] \qquad (4) \\ = & \bigwedge_{i=1}^{N} ((c_{i,1} = m_{i,1} \vee c_{i,1} = m_{i,2}) \wedge (c_{i,2} = f_{i,1} \vee c_{i,2} = f_{i,2})) \vee \\ & \qquad ((c_{i,1} = f_{i,1} \vee c_{i,1} = f_{i,2}) \wedge (c_{i,2} = m_{i,1} \vee c_{i,2} = m_{i,2})) = \top, \end{aligned}$$

  where $A \boxtimes B$ denotes the set of all two-element multisets, where one element is taken from A and the other one from B, i.e., $A \boxtimes B = \{\{a, b\} \mid a \in A,\ b \in B\}$. Again, the test may be designed so that it allows a small number of mismatches.

3 Protocols for Secure STR Matching

In this section we consider efficient privacy-preserving implementations of the tests outlined in Section 2: two or three STR profiles are tested whether they are 'related' according to Eqs. (1)-(4), while being assured that the protocol execution does not leak information about the profiles except the result under computational assumptions. In particular, in case of a mismatch, no protocol participant should learn any information about the other participant's profiles, except that they do not match his own.

Our protocols require a semi-honest attack model, i.e., parties that correctly follow the protocol. Furthermore, for obvious reasons, the protocol cannot guarantee that the participants use proper input data; if a participant manages to use the DNA of her dog instead of her own, the results of the protocol will be correspondingly wrong. In some of our settings, no party has an interest in such a form of cheating; while, for example, an genealogy service may have an interest in keeping more data than actually needed, there is little reason to manipulate the protocol output. (However note that it sometimes is possible to derive information from the protocols input/output behavior alone; as we will show later in Section 4, such an information leak is unavoidable). In settings in which one party has an interest in manipulating the output—such as comparison of DNA found at a crime scene and that of a suspect—this party would usually not participate in the protocol in the first place. Rather, our scheme allows for a separation of duties. While the police can collect a database of encrypted DNA identities, a watchdog organization may hold the keys and thus be able to prevent abusive use of the DNA database.

3.1 Preliminaries

The privacy problem addressed in this paper can be formulated as an instance of secure multiparty computation [26, 7]. Technically, we show that secure evaluation of multivariate polynomials (using homomorphic encryption) can yield efficient STR matching protocols. The use of polynomial evaluation for comparison of private data (profile matching) was first proposed by Freedman et al. [12]. For a specific matching problem, one constructs a polynomial over the input values of each party, which evaluates to zero if and only if the inputs match. The matching protocol consists of a secure two-party computation to evaluate the polynomial on the private input values. In this paper we concentrate on protocols that are secure in the semi-honest adversary model; however, extensions to the malicious case are possible using standard constructions of secure multiparty computation [23], at the expense of efficiency.

Homomorphic Encryption. In the constructions, we use a homomorphic public key encryption scheme E^H, such as Paillier encryption [22] over a message space $\mathcal{M}$, which has the property that $E^H(x+y)$ can be efficiently computed from the individual encryptions $E^H(x)$ and $E^H(y)$ without knowledge of the secret key, where the addition operation is performed in a finite ring. Additionally, $E^H(rx)$ can be computed from r and $E^H(x)$ as $E^H(x)^r$.

Besides Paillier encryption, it is also possible to employ the following homomorphic variant of ElGamal [23] with $\mathcal{M} = \mathbb{Z}_q$. Let p be a large prime and g be a generator of prime order q (with $q \,|\, p-1$) of a suitably large subgroup of $\mathbb{Z}_p^*$. To generate a public-/private key pair, one chooses a random $1 \leq \alpha \leq q-1$ and computes $h = g^\alpha \bmod p$; the public key is given by the tuple (p, g, h), whereas the private key is α. To encrypt an element $m \in \mathbb{Z}_q$, one chooses a random element $r \in \mathbb{Z}_q$ and computes the ciphertext tuple $(c_1, c_2) = (g^r \bmod p,\ g^m h^r \bmod p)$. The scheme can thus be seen as plain ElGamal encryption, where a message m is encoded as g^m before encryption. It is easy to see that this encryption scheme is homomorphic with respect to addition in $\mathbb{Z}_q$: suppose that (c_1, c_2) is the encryption of a plaintext x and (c_1', c_2') is the encryption of x', then $(c_1 c_1' \bmod p, c_2 c_2' \bmod p)$ is an encryption of $x + x' \bmod p$.

Unfortunately this encryption scheme offers only limited decryption possibilities. As in the plain ElGamal scheme, to decrypt a ciphertext (c_1, c_2), one first computes $c_2(c_1^\alpha)^{-1} \bmod p$, yielding the encoded message g^m. Recovering m from this encoding requires taking a discrete logarithm in $\mathbb{Z}_q^*$, which is assumed to be intractable. However, if the encrypted message m is known to belong to a small message space, g^m can be decoded by brute force search. Similarly, given the private key, it can be tested efficiently whether (c_1, c_2) is the encryption of a specific message m. The protocols presented in the paper require only the latter capability: it is sufficient to distinguish, given the knowledge of the private key, an encryption of the message $m = 0$ from

1. T maps his alleles $\overline{t_{i,1}} = H_i(t_{i,1})$ and $\overline{t_{i,2}} = H_i(t_{i,2})$ for $1 \leq i \leq N$.
2. T computes the sum $A = \sum_{i=1}^{N} \overline{t_{i,1}} + \overline{t_{i,2}}$ and sends $E^H_{pk_T}(A)$ to S.
3. S similarly maps her alleles $\overline{s_{i,1}} = H_i(s_{i,1})$ and $\overline{s_{i,2}} = H_i(s_{i,2})$, computes the sum $B = \sum_{i=1}^{N} \overline{s_{i,1}} + \overline{s_{i,2}}$ and computes an encryption $E^H_{pk_T}(B)$.
4. S computes, using homomorphic properties of E^H, an encryption $E^H_{pk_T}(rZ_I) = E^H_{pk_T}(r(A-B))$, where r is a blinding factor, chosen uniformly at random. The obtained encryption is sent back to T.
5. T decrypts the result and reports a match if a zero is obtained.

Figure 1: Secure identity testing protocol.

an encryption of a message $m \neq 0$. This can be performed efficiently by ordinary ElGamal decryption to obtain the coded message g^m and by testing whether $g^m = 1$.

Representation of Alleles. To represent alleles as elements of the message space $\mathcal{M}$ of the underlying encryption scheme, we use families of random injective functions $\langle H_1, \ldots, H_N \rangle$, where each function is randomly drawn from the set of injective functions $\Sigma \to \mathcal{M}$. The cardinality $|\mathcal{M}|$ acts as security parameter. In the protocols described below, we use the function H_i to map the alleles of locus i. Due to the random choice of H_i, the mapping becomes dependent on the locus. The correctness of the protocols highly depends on the property that the same allele will be mapped to different messages at different loci. Given an STR profile $\langle \{t_{i,1}, t_{i,2}\} \rangle$ or $\langle \{t_i\} \rangle$, we will denote with $\overline{t_{i,j}} = H_i(t_{i,j})$ or $\overline{t_i} = H_i(t_i)$ the alleles, mapped according to the locus in which they appear.

In the analysis, we will assume that each H_i is a random injective function, which is known to all protocol participants. In practice it is possible, due to the small cardinality of Σ, to store each H_i as a table. Alternatively, a more space-efficient implementation can be derived from any collision-resistant hash function H that maps into the set $\mathcal{M}$ by letting $H_i(m) = H(\textit{pad} \,\|\, i \,\|\, m)$, where *pad* denotes some padding. Assuming that H is collision-resistant, the function H_i will, due to the small cardinality of Σ, be injective with high probability (note that any two inputs $a, a' \in \Sigma$ that violate the injectivity of H_i immediately yield a collision of H).

3.2 Identity Testing

We fix a family $\langle H_1, \ldots, H_N \rangle$ of random functions, as described in Section 3.1. By mapping all alleles to messages in $\mathcal{M}$, secure evaluation of Eq. (1) reduces to determining whether the sum

$$Z_I = \sum_{i=1}^{N} (\overline{s_{i,1}} - \overline{t_{i,1}}) + (\overline{s_{i,2}} - \overline{t_{i,2}}) = \sum_{i=1}^{N} (\overline{s_{i,1}} + \overline{s_{i,2}}) - \sum_{i=1}^{N} (\overline{t_{i,1}} + \overline{t_{i,2}}) \tag{5}$$

evaluates to zero. This can be efficiently tested (in one round with constant communication complexity) by the simple protocol depicted in Figure 1. Participant T maps all his alleles using H_i, adds the mapped alleles, encrypts the result with his public key pk_T and sends it to participant S. S in turn maps her own alleles and subtracts their encryptions from the value received by T. Finally, the result is multiplicatively blinded and sent back to T, who reports a match if he received an encryption of a zero. The protocol protects the privacy of both STR profiles (assuming semi-honest participants), as S obtains only a semantically secure encryption of the sum of T's mapped alleles and T obtains only a binary answer: a zero if there was a match or a uniformly chosen random number otherwise.

Correctness. It is easy to see that the above protocol is correct, i.e. it yields $Z_I = 0$ if and only if there is a match, except with negligible probability. If there is a match between the STR profiles of S and T, i.e., for all loci $1 \leq i \leq N$, the multisets $\{s_{i,1}, s_{i,2}\}$ and $\{t_{i,1}, t_{i,2}\}$ are equivalent, Eq. (5) obviously yields zero. Suppose now that there is a mismatch. Under the assumption that both $N = O(\log|\mathcal{M}|)$ and $|\Sigma| = O(\log|\mathcal{M}|)$, the above protocol yields a positive result only with negligible probability. This can be seen as follows: The probability of a false positive in the above protocol is equal to the probability that the following event happens

$$\overline{t_{N,2}} = \sum_{i=1}^{N}(\overline{s_{i,1}} + \overline{s_{i,2}}) - \sum_{i=1}^{N}\overline{t_{i,1}} - \sum_{i=1}^{N-1}\overline{t_{i,2}} \equiv \kappa.$$

Since the injective functions H_i for $i = 1, \dots, N$ are randomly chosen from the set of injective functions $\Sigma \rightarrow \mathcal{M}$, the values $\overline{s_{i,1}}, \overline{s_{i,2}}, \overline{t_{i,1}}$ for $i = 1, \dots, N$ and $\overline{t_{i,2}}$ for $i = 1, \dots, N-1$ are randomly distributed. Hence,

$$\text{Prob}[Z_I = 0] = \text{Prob}[H_N(t_{N,2}) = \kappa] = \frac{1}{|\mathcal{M}|}, \tag{6}$$

which is negligible.

Coping with Errors. The protocol as depicted above is sensitive to errors in the STR profile. As mentioned in Section 2, usually a small number of mismatches (one or two) need to be tolerated due to the chemical imperfections of the DNA sequencing process.

To cope with one mismatch, the above protocol can be extended in a straightforward manner: we modify the polynomial in such a way that we compute a sum Z_E as

$$Z_E = \sum_{(i,j) \in \{1,\dots,N\},\, i<j} z_i z_j,$$

where $z_i = (\overline{s_{i,1}} + \overline{s_{i,2}}) - (\overline{t_{i,1}} - \overline{t_{i,2}})$. By expanding the polynomial and writing it in terms of factors $\overline{s_{i,1}} + \overline{s_{i,2}}$ and $\overline{t_{i,1}} + \overline{t_{i,2}}$, Z_E can be computed efficiently by S if T pre-computes encryptions of the required terms $(\overline{t_{i,1}} + \overline{t_{i,2}})^k(\overline{t_{j,1}} + \overline{t_{j,2}})^l$ for $0 \leq k, l \leq 1$ and different indices i and j. The protocol is depicted in Figure 2. T starts by pre-computing the required encryptions of his mapped alleles and forwards them to S, who in turn can use the homomorphic properties of E^H to compute an encryption $E^H_{pk_T}(Z_E)$. Finally, R submits a blinded encryption $E^H_{pk_T}(rZ_E)$ to T, who decrypts the result. If he obtains a zero, a match is reported.

1. For each locus $1 \leq i \leq N$, T maps his alleles $\overline{t_{i,1}} = H_i(t_{i,1})$ and $\overline{t_{i,2}} = H_i(t_{i,2})$.
2. T computes encryptions of all products $E^H_{pk_T}((\overline{t_{i,1}} + \overline{t_{i,2}})^k(\overline{t_{j,1}} + \overline{t_{j,2}})^l)$ for $0 \leq k, l \leq 1$ with $(k, l) \neq (0, 0)$ and $1 \leq i < j \leq N$ and forwards them to R.
3. R maps her alleles $\overline{s_{i,1}} = H_i(s_{i,1})$ and $\overline{s_{i,2}} = H_i(s_{i,2})$ and uses the homomorphic property of the encryption to obtain an encrypted value $E^H_{pk_T}(Z_E)$, which she blinds with a uniformly random blinding factor r.
4. R forwards $E^H_{pk_T}(rZ_E)$ to T, who decrypts the value and reports a match if and only if he obtains a zero.

Figure 2: Secure identity testing protocol tolerating one allele error.

	Communication Complexity	Transmitted Data (in KB)		
		$N = 13$	$N = 37$	$N = 67$
Identity test, $t = 0$	1	0.3	0.3	0.3
Identity test, $t = 1$	$\frac{3}{2}N(N-1)$	58.5	499.5	1658.3
Identity test, $t = 2$	$\frac{7}{6}N(N-1)(N-2)$	500.5	13597.5	83833.7
Common Ancestor Test, $t = 1$	$\frac{3}{2}N(N-1)$	58.5	499.5	1658.3
Common Ancestor Test, $t = 2$	$\frac{7}{6}N(N-1)(N-2)$	500.5	13597.5	83833.7
One Parent Paternity Test, $t = 0$	$18N$	26.0	74.0	134.0
One Parent Paternity Test, $t = 1$	$40N(N-1)$	3120.0	13320.0	44220.0
Two Parent Paternity Test, $t = 0$	$23N + 1$	74.7	212.7	385.2

Table 2: Complexity of the matching protocols.

Note that Z_E will (except with negligible probability) be zero if an error occurs in at most one locus of the profile: in this case only one of the values z_i will be non-zero. If there is more than one error, at least one product $z_i z_j$ will be nonzero, which results in Z_E being nonzero, except with negligible probability. This construction can be generalized in a straightforward manner for arbitrary t by summing over all products of t values z_i; however, the scheme soon gets inefficient due to large space requirements for transmitting the pre-computed products of mapped alleles.

Complexities. Table 2 gives an overview of the communication complexities of the matching protocols proposed in this section. As it can be seen, the complexity highly depends on the number of errors t that need to be tolerated during the matching process. Besides the communication complexity (measured in the number of transmitted encryptions with respect to the length N of the STR profile), we also list the number of transmitted bytes for practical length STR sequences ($N = 13$, $N = 37$ and $N = 67$), assuming a message space of 2048 bits for E^H.

1. For each locus $1 \leq i \leq N$, T maps his alleles $\overline{t_{i,1}} = H_i(t_{i,1})$ and $\overline{t_{i,2}} = H_i(t_{i,2})$.
2. T computes encryptions of all products $E^H_{pk_T}((\overline{t_{i,1}})^k(\overline{t_{i,2}})^l)$ for $0 \leq k, l \leq 2$ and $(k, l) \neq (0, 0)$ and forwards them to R.
3. R maps her alleles $\overline{s_{i,1}} = H_i(s_{i,1})$ and $\overline{s_{i,2}} = H_i(s_{i,2})$ and uses the homomorphic property of the encryption to obtain an encrypted value $E^H_{pk_T}(Z_O)$, which she blinds with a uniformly random blinding factor r.
4. R forwards $E^H_{pk_T}(rZ_O)$ to T, who decrypts the value and reports a match if and only if he obtains a zero.

Figure 3: Secure paternity testing protocol (one parent case).

3.3 Paternity Testing with One Parent

The problem of paternity testing with one parent can be formulated as a secure function evaluation problem as well. Evaluating Eq. (3) can be performed by testing whether the sum

$$Z_O = \sum_{i=1}^{N} \underbrace{(\overline{s_{i,1}} - \overline{t_{i,1}})(\overline{s_{i,1}} - \overline{t_{i,2}})(\overline{s_{i,2}} - \overline{t_{i,1}})(\overline{s_{i,2}} - \overline{t_{i,2}})}_{z_i} \tag{7}$$

is zero. This can be done, again using homomorphic encryption, by the efficient protocol depicted in Figure 3, which requires one round and linear communication complexity. These complexity bounds can be achieved by observing that Eq. (7) can be written as a multivariate polynomial of degree two in $\overline{t_{i,1}}$ and $\overline{t_{i,2}}$ for $1 \leq i \leq N$. To securely and efficiently evaluate Eq. (7), it is thus sufficient for T to provide encryptions of all mixed products $E^H_{pk_T}((\overline{t_{i,1}})^k(\overline{t_{i,2}})^l)$ for $0 \leq k, l \leq 2$ and $(k, l) \neq (0, 0)$ under his public key pk_T to R, who in turn can use the homomorphic properties of E^H to compute an encryption $E^H_{pk_T}(Z_O)$. Finally, R submits a blinded encryption $E^H_{pk_T}(rZ_O)$ to T, who decrypts the result. If he obtains a zero, a match is reported. The privacy of the profiles of both T and R are assured (in the semi-honest model), as R only obtains semantically secure encryptions and T receives a binary answer. Table 2 gives an overview of the communication complexity of the approach, compared with the protocols of Section 3.2.

Correctness. It is easy to see that the protocol is correct. If there is a paternity relationship between S and R, i.e., for all loci we have $\{s_{i,1}, s_{i,2}\} \cap \{t_{i,1}, t_{i,2}\} \neq \emptyset$, the sum of Eq. (7) will certainly evaluate to zero and the protocol reports a match. Suppose now there is no match. Then, for at least one locus $1 \leq i \leq N$, the elements in $\{s_{i,1}, s_{i,2}\}$ will be different from the elements in $\{t_{i,1}, t_{i,2}\}$. Subsequently, the i-th summand of Eq. (7) will, due to the random nature of the mapping H_i, be a random element of the message space $\mathcal{M}$. The probability Prob$[Z_O = 0]$ = Prob$[z_1 + \ldots + z_N = 0] = \frac{1}{|\mathcal{M}|}$ is negligible according to a similar reasoning as in Eq. (6); thus the protocol will, except with negligible probability, report a mismatch.

1. For each locus $1 \leq i \leq N$, T maps his alleles $\overline{t_i} = H_i(t_i)$.
2. T computes encryptions of all products $E^H_{pk_T}((\overline{t_i})^k(\overline{t_j})^l)$ for $0 \leq k, l \leq 1$ with $(k, l) \neq (0, 0)$ and $1 \leq i < j \leq N$ and forwards them to R.
3. R maps her alleles $\overline{s_i} = H_i(s_i)$ and uses the homomorphic property of the encryption to obtain an encrypted value $E^H_{pk_T}(Z_C)$, which she blinds with a uniformly random blinding factor r.
4. R forwards $E^H_{pk_T}(rZ_C)$ to T, who decrypts the value and reports a match if and only if he obtains a zero.

Figure 4: Secure common ancestor testing protocol.

Variations. The protocol can be made error-resilient in the same way as in Section 3.2. To cope with one error, the value Z_O is computed as $Z_O = \sum_{(i,j)\in\{1,\ldots,N\},\, i<j} z_i z_j$, which can be done by precomputing all required powers $(\overline{t_{i,1}})^k(\overline{t_{i,2}})^l(\overline{t_{j,1}})^m(\overline{t_{j,2}})^n$.

3.4 Common Ancestor Testing

By using a similar approach as the paternity test with one parent, privacy-preserving common ancestor tests on the Y chromosome can be implemented. We again give only the protocol that allows to cope with at most one error ($t = 1$); extensions to larger t are straightforward. Testing the condition of Eq. (2) between the two Y-chromosome STR profiles $S = \langle\{s_i\}\rangle$ and $T = \langle\{t_i\}\rangle$ for the case of one error requires evaluating the polynomial

$$Z_C = \sum_{(i,j)\in\{1,\ldots,N\},\, i<j} (\overline{s_i} - \overline{t_i})(\overline{s_j} - \overline{t_j}).$$

This can again be done by pre-evaluating all required powers of $(\overline{t_i})^k(\overline{t_j})^l$ by the efficient protocol of Figure 4.

3.5 Paternity Testing with Two Parents

The problem of paternity testing with two parents can, in a similar way as the related problem with one parent, be posed as a secure function evaluation problem. Evaluating Eq. (4) straightforwardly translates into the problem of testing whether

$$\begin{aligned} Z_T \quad = \quad & \sum_{i=1}^{N} \Big[(\overline{c_{i,1}} - \overline{m_{i,1}})(\overline{c_{i,1}} - \overline{m_{i,2}}) + (\overline{c_{i,2}} - \overline{f_{i,1}})(\overline{c_{i,2}} - \overline{f_{i,2}})\Big] \cdot \\ & \Big[(\overline{c_{i,1}} - \overline{f_{i,1}})(\overline{c_{i,1}} - \overline{f_{i,2}}) + (\overline{c_{i,2}} - \overline{m_{i,1}})(\overline{c_{i,2}} - \overline{m_{i,2}})\Big] \end{aligned} \tag{8}$$

evaluates to zero. By expanding the factors, the above equation can be written as a polynomial in the values $\overline{c_{i,1}}$, $\overline{c_{i,2}}$, $A_i = \overline{m_{i,1}} + \overline{m_{i,2}}$, $B_i = \overline{m_{i,1}m_{i,2}}$, $C_i = \overline{f_{i,1}} + \overline{f_{i,2}}$ and $D_i = \overline{f_{i,1}f_{i,2}}$, in which all powers $(\overline{c_{i,1}})^k(\overline{c_{i,2}})^l$ with $0 \leq k, l \leq 4$ and terms $A_i, B_i, C_i, D_i, A_iB_i, C_iD_i, A_iC_i, A_iD_i, B_iC_i, B_iD_i, A_i^2, B_i^2, C_i^2, D_i^2$ appear. This allows for an efficient oblivious evaluation of the polynomial; depending on which party receives the answer of the matching process, the following evaluation strategies can be employed:

- **Result available to M.** M starts the protocol by choosing a pair of public/private keys, encrypting the values $A_i, B_i, A_i^2, B_i^2, A_iB_i$ and sending them to F, who in turn uses the homomorphic properties of the encryption to compute encryptions of the cross terms $A_iC_i, A_iD_i, B_iC_i, B_iD_i$; he then forwards all computed encryptions, including $C_i, D_i, C_i^2, D_i^2, C_iD_i$, to C, who finally uses again the homomorphic property to compute an encryption of Z_T. This result is randomized with a multiplicative blinding factor and sent back to M. Finally, M decrypts the result and checks the answer.

- **Result available to F.** This case is analogous to the previous one, with the roles of F and M interchanged.

- **Result available to C.** C starts the protocol by choosing a pair of public/private keys and encrypting all required powers $(\overline{c_{i,1}})^k(\overline{c_{i,2}})^l$. He forwards all encryptions to F, who computes all terms that involve the values C_i, D_i, sends the result on to M, who finally evaluates the polynomial under encryption, blinds the result and sends the encrypted value back to C, who decrypts to obtain the answer.

Note that the complexity of the protocol depends on the evaluation strategy. In case M or F receive the result, the protocol requires transmission of 19 encryptions for each locus; in case C gets the result, it requires 23 encryptions per locus.

4 Fundamental Limitations of STR Privacy

The protocols designed in this paper assumed semi-honest participants, who execute the protocols correctly and do not 'lie' about their input profiles. In this section we consider the impact of an attacker who (honestly) runs the matching protocol but is allowed to execute the protocol on arbitrarily chosen inputs. In this setting, we show a general impossibility: In case the involved protocol participants can 'lie' about their STR profiles on which the protocols are run and multiple dependent protocol runs are performed, *no protocol can exist* that perfectly assures participants' privacy in paternity tests. The intuitive reason for this result lies small length and limited entropy of STR profiles, as noted in Section 2. As this result is an inherent consequence of the problem statement, it is important to limit the abilities of the protocol participants to arbitrarily modify their input profiles. (This can e.g. be achieved by requiring the parties to commit to their inputs before the protocol runs and proving that the execution is correct with respect to the committed profiles.)

We will illustrate this result for the one-parent parental testing problem of Eq. (3). If an attacker has a parental relationship with the victim[1], each invocation of an STR matching protocol

[1] Note that the parent-child relationship required between the attacker and victim for a successful attack is transitive. For example, a mother can use it to determine the sequence of her child, and in turn use this to determine the sequence of the father. Using this, a sufficiently determined attacker could get the sequence of arbitrary persons, though the level of determination and minimum duration of 9 month needed for that attack would make it difficult in practice.

allows to derive information about the victims DNA. In particular, we will show that running $N|\Sigma|$ tests allows a malicious attacker with a parent-child relationship to the victim to reconstruct the entire STR sequence of the victim.

Suppose we have a black box matching protocol that, given STR profiles S and T, outputs *yes* if and only if the attacker S is in a parent-child relationship with the victim T. Suppose further that the attacker is in a parent-child relationship with the victim, i.e., S knows a STR profile that results in a *yes*, and that T can arbitrarily modify his input to the black box protocol. Denote with $S = \langle\{s_{i,1}, s_{i,2}\}\rangle$ and $T = \langle\{t_{i,1}, t_{i,2}\}\rangle$ the STR profiles of the attacker and the victim, respectively. By the assumption that the attacker and the victim are related, running one instance of the test will result in *yes*. To learn whether a specific allele $a \in \Sigma$ appears in the STR profile of T at locus i, i.e., $a \in \{t_{i,1}, t_{i,2}\}$, the attacker replaces both alleles $s_{i,1}$ and $s_{i,2}$ in his own profile with a, and reruns the matching protocol with the modified profile. If the result is still positive, he knows that a appears in T's STR profile at locus i. To learn the entire profile of T, the attacker runs the protocol with all values of $a \in \Sigma$ on all positions i. This results in an attack that requires at most $N|\Sigma|$ protocol runs to extract the complete profile of the victim (note that the attacker can optimize by stopping some tests early or incorporate knowledge on the distribution of alleles in Σ).

Similar attacks exist for the other matching problems as well. For example, in the parental test scenario with two parents, an attacker can run the test by feeding his own STR profile as inputs of both C and M. As, according to Eq. (3), every person can potentially be his own parent, this allows to run the above attack also in a two-parent test scenario.

5 Conclusions

We have presented a set of protocols that allow to run the most common DNA-based identity, paternity and ancestry tests in a privacy-preserving manner; the protocols can form the basis for privacy enhanced genealogical services or research projects. Our protocols take into account the special structure and properties of STR profiles, which allows for error-resilient, efficient and practical protocols. Furthermore, they offer full privacy in the semi-honest attacker model.

References

[1] www.23andme.com.

[2] L. Andrews. Predicting and punishing antisocial acts. In *Behavioral Genetics and Society: The Clash of Culture and Biology*. John Hopkins University Press, 1999.

[3] P. Bohannon, M. Jakobsson, and S. Srikwan. Cryptographic approaches to privacy in forensic DNA databases. In *Public Key Cryptography, Third International Workshop on Practice and Theory in Public Key Cryptography, PKC 2000*, volume 1751 of *Lecture Notes in Computer Science*, pages 373–390. Springer, 2000.

[4] J. Butler. *Forensic DNA Typing: Biology, Technology, and Genetics of STR Markers*. Academic Press, 2005.

[5] J. M. Butler. Genetics and genomics of core Short Tandem Repeat loci used in human identity testing. *Journal of Forensic Sciences*, 51(2):253–265, 2006.

[6] CODIS—combined DNA index system. `http://www.fbi.gov/hq/lab/codis`.

[7] R. Cramer, I. Damgård, J. B. Nielsen, and B. Pfitzmann. Multiparty computation from threshold homomorphic encryption. In *Advances in cryptology—EUROCRYPT 2001*, volume 2045 of *Lecture Notes in Computer Science*, pages 280–300. Springer, May 2001.

[8] George Danezis, Claudia Díaz, Sebastian Faust, Emilia Käsper, Carmela Troncoso, and Bart Preneel. Efficient negative databases from cryptographic hash functions. In *Information Security, 10th International Conference*, volume 4779 of *Lecture Notes in Computer Science*, pages 423–436. Springer, 2007.

[9] Y. Dodis, L. Reyzin, and A. Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In *Advances in Cryptology—EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 523–540. Springer, 2004.

[10] N. V. Morgan et al. A novel locus for Meckel-Gruber syndrome, MKS3, maps to chromosome 8q24. *Human Genetics*, 111(4-5):456–461, 2002.

[11] `www.familytreedna.com`.

[12] M. Freedman, K. Nissim, and B. Pinkas. Efficient private matching and set intersection. In *Advancves in Cryptology—EUROCRYPT'04*, volume 3027 of *Lecture Notes in Computer Science*, pages 1–19. Springer, 2004.

[13] E. Joh. Reclaiming “abandoned dna”: The fourth amendment and genetic privacy. Technical Report 40, School of Law, University of California, Davis, 2005.

[14] S. J. Niezgoda Jr. and B. Brown. The FBI laboratory's combined DNA index system program. In *Proceedings of the Sixth International Symposium on Human Identification*, 2005.

[15] M. Kantarcioglu, W. Jiang, Y. Liu, and B. Malin. A cryptographic approach to securely share and query genomic sequences. *IEEE Transactions on Information Technology in Biomedicine*, 2008, to appear.

[16] C. Kimpton. Report on the second EDNAP collaborative STR exercise. *Forensic Science International*, 71:137–152, 1995.

[17] S. Lehrman. Prisoners' dna database ruled unlawful. *Nature*, 394:818, 1998.

[18] J. Linnartz and P. Tuyls. New shielding functions to enhance privacy and prevent misuse of biometric templates. In *Audio-and Video-Based Biometrie Person Authentication, 4th International Conference*, volume 2688 of *Lecture Notes in Computer Science*, pages 393–402. Springer, 2003.

[19] 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.

[20] B. Malin. Re-identification of familial database records. In *Proceedings of the 2006 American Medical Informatics Annual Fall Symposium*, pages 524–528, 2006.

[21] Canadian Society of Forensic Science. Population studies data centre. `http://www.csfs.ca/databases`.

[22] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Advances in Cryptology—EUROCRYPT'99*, volume 1592 of *Lecture Notes in Computer Science*, pages 223–238. Springer, 1999.

[23] B. Schoenmakers and P. Tuyls. Practical two-party computation based on the conditional gate. In *Advances in Cryptology—ASIACRYPT 2004*, number 3329 in Lecture Notes in Computer Science, pages 119–136. Springer, 2004.

[24] National Geographic Society. The Genographic project. `https://www3.nationalgeographic.com/genographic`.

[25] J. R. Troncoso-Pastoriza, S. Katzenbeisser, and M. Celik. Privacy preserving error resilient DNA searching through oblivious automata. In *Proceedings of the 14th ACM conference on Computer and communications security (CCS'07)*, pages 519–528. ACM, 2007.

[26] A. C. Yao. Protocols for secure computations. In *Proceedings of the IEEE Symposium on Foundations of Computer Science*, pages 160–164, 1982.