Identifying Personal DNA Methylation Profiles by Genotype Inference

Michael Backes*, Pascal Berrang*, Matthias Bieg†, Roland Eils†‡,
Carl Herrmann†‡, Mathias Humbert*, Irina Lehmann§
*CISPA, Saarland University, Saarland Informatics Campus
†German Cancer Research Center (DKFZ)
‡University of Heidelberg
§Helmholtz Centre for Environmental Research Leipzig, UFZ, Leipzig

Abstract—**Since the first whole-genome sequencing, the biomedical research community has made significant steps towards a more precise, predictive and personalized medicine. Genomic data is nowadays widely considered privacy-sensitive and consequently protected by strict regulations and released only after careful consideration. Various additional types of biomedical data, however, are not shielded by any dedicated legal means and consequently disseminated much less thoughtfully. This in particular holds true for DNA methylation data as one of the most important and well-understood epigenetic element influencing human health.**

In this paper, we show that, in contrast to the aforementioned belief, releasing one's DNA methylation data causes privacy issues akin to releasing one's actual genome. We show that already a small subset of methylation regions influenced by genomic variants are sufficient to infer parts of someone's genome, and to further map this DNA methylation profile to the corresponding genome. Notably, we show that such re-identification is possible with 97.5% accuracy, relying on a dataset of more than 2500 genomes, and that we can reject all wrongly matched genomes using an appropriate statistical test. We provide means for countering this threat by proposing a novel cryptographic scheme for privately classifying tumors that enables a privacy-respecting medical diagnosis in a common clinical setting. The scheme relies on a combination of random forests and homomorphic encryption, and it is proven secure in the honest-but-curious model. We evaluate this scheme on real DNA methylation data, and show that we can keep the computational overhead to acceptable values for our application scenario.

I. Introduction

Since the first whole-genome sequencing in 2001, the cost of molecular profiling has been plummeting, enabling a significant progress in biomedical science and the rise of precision medicine [1]. This scientific breakthrough is triggered by the increasing availability of biomedical data, whose main negative counterpart is the new threat towards health privacy. The extent of the threat, and mechanisms to mitigate it, have been extensively studied regarding the genomic data. The various attack vectors and protection techniques have been well surveyed and categorized back in 2014 already [2]. The genome is especially privacy sensitive as it uniquely identifies someone, it is very stable over our whole lifetime, and it is correlated among relatives [3]. This may explain why the security community has been, so far, focusing essentially on enhancing the privacy of genomic data, and not the other types of biomedical data, such as epigenetic data, despite their vital functions for human health and their rapidly growing availability [4].

DNA methylation is one of the most important and best understood epigenetic elements influencing human health. It is an essential regulator of gene transcription. As a consequence, aberrant DNA methylation patterns (such as hypermethylation and hypomethylation) have been associated with a large number of cancer types [5], [6], [7]. Because of its crucial role in human health, DNA methylation data might constitute highly sensitive data as well, whose privacy should be protected using dedicated legal or technical means. However, epigenetic data might not even be considered as genetic information in the strict legal sense, and thus not be protected by legal frameworks, such as the US Genetic Information Nondiscrimination Act (GINA) [8], [9].

Contrary to the genome, DNA methylation data, and more generally epigenetic data, vary quite significantly over time, mainly because they are highly influenced by environmental factors. This may explain why DNA methylation data are simply released (without identifiers) on open online platforms with nonrestricted access. In order to prevent privacy breaches, the genomic data corresponding to the DNA methylation data are generally *not* made publicly available, and follow stricter privacy rules. However, it is well-known that DNA methylation is also influenced by genetic factors [10]. As a consequence, correlations between DNA methylation and the genome could be exploited in order to re-identify anonymous DNA methylation profiles by using some public genomic database (e.g., OpenSNP [11]). Unfortunately, previous work has only tackled potential re-identification risks and countermeasures from a relatively high-level qualitative perspective (see Section IX). In this work, we provide the first detailed quantitative assessment of the identification risks inherent to DNA methylation data and, moreover, propose a provably secure technical mechanism to enable privacy-preserving methylation-based diagnosis.

a) Contributions: Specifically, we present a Bayesian inference framework to predict part of the genotype from DNA methylation data. We then propose an algorithm that matches DNA methylation profiles to the genotypes whose posterior probabilities are maximized given these methylation profiles. By using a rich methylation-genotype dataset, we show that only a few tens of methylation regions are sufficient to accurately match DNA methylation to genotypes. Furthermore, we present a statistical method that enables us to reject the small fraction of cases where the matching algorithm does not provide 100% accuracy, e.g., when the genotype corresponding to the methylation profile is not part of the genotype dataset. We also observe that, in such cases, if a relative is part of the genotypes' dataset, it is the one (wrongly) matched to the methylation profile. By including all genotypes contained in phase 3 of the 1000 Genome Project, we show that the attack success is very robust to an increase in the size of the genotype dataset. All accuracy, false-positive and true-positive rates remain constant for a size of the genotype dataset varying from 75 to 2579.

Given the extent of the threat, we propose a novel cryptographic scheme for privately classifying tumors, which enables a privacy-preserving medical diagnosis in a common clinical setting. With our method, neither a curious third-party running the machine-learning algorithm can learn the personal DNA methylation data, nor the data owner (e.g., the patient) can learn the detailed machine-learning model. In particular, we adapt existing homomorphic schemes for privately evaluating random forests with encrypted data, and prove the resulting scheme secure in the honest-but-curious adversarial model, which constitutes the state-of-the-art adversary model in this problem setting. We evaluate the classifier performance on real methylation data, and show that it can precisely classify brain tumors in 9 subtype classes based on 900 methylation levels in less than an hour, which represents a fully tolerable computational time for the considered application scenario.

b) Organization: In Section II, we introduce the relevant concepts and properties of DNA methylation. In Section III, we present the considered adversarial model. We then detail the analytical method behind our identification attack in Section IV. We describe our dataset in Section V before using it to evaluate the success of our attack in Section VI. In Section VII, we present our private classification algorithm and evaluate its performance in Section VIII. We review the most relevant previous work in Section IX, before concluding in Section X. We provide the detailed security proofs of our cryptographic scheme in the Appendix.

II. Background

Methylation of the DNA is one of the most important epigenetic modifications in the genome, with profound consequences on the structure and the activity of the DNA molecule [12], [13]. It has been observed in numerous species (animals and plants), but some species lack this mechanism. It consists in the addition of a methyl group to the cytosines or adenine by specific enzymes called methyltransferases; however in humans, only cytosine methylation in CpG-dinucleotides, leading to the formation of 5-methylcytosine, has been observed. Given its mostly repressive effect on gene expression, DNA methylation at the promoter of genes is a mechanism by which genes can be silenced during development, for example to maintain the pluripotent state of stem cells [14].

Aberrant changes in the DNA methylation patterns, which are frequently observed in cancer, can lead to the hyperactivation of genes such as oncogenes, or the silencing of tumor suppressor genes [5]. While the changes in the DNA methylation pattern can be dramatic in cancer, DNA methylation in normal tissues can also be modified due to, for example, environmental influences. It has been shown in diverse studies that environmental cues such as pollution, exposure to stress or cigarette smoke leads to changes in the DNA methylation of the genome for persons exposed to these influences [15], [16], [17], [18]. Recently, several studies analyzed the influence of these external effects on the methylation patterns in a cohort of mothers and children and found massive number of differentially methylated regions when comparing children of smoking and non-smoking mothers, with downstream effects on the expression of genes involved in important pathways of lung development and maturation [15], [16].

Besides external factors, the genotype of an individual can also affect the methylation of certain regions [19], [10], [20]. Individuals carrying particular alleles at some single nucleotide polymorphisms (SNPs) can exhibit specific DNA methylation patterns at some loci. Such SNPs having an influence on the methylation are called methylation quantitative trait loci (meQTLs), and have been studied previously to uncover the mechanisms by which single nucleotide polymorphisms can have a effect on the methylation patterns. An obvious effect is when the polymorphism affects a CpG dinucleotide. If the polymorphism affects the cytosine (C) or the guanine (G), the CpG dinucleotide is lost, leading to a loss of methylation at this site. However, other polymorphisms beyond these "CpG destroying SNPs" can lead to methylation changes. Given this possible link between varying genotypes and DNA methylation, the question is to what extent knowledge of the DNA methylation pattern could be used to reverse-engineer the meQTLs and predict genotypes based on the methylation.

III. Threat Model

We assume that the adversary gets access to one or multiple individual profiles of genome-wide DNA methylation levels, as well as to a set of genotypes. There are around 28 million CpG sites per individual and about 150 million known genomic variants to which the adversary can potentially have access. Then, we study various scenarios that could occur in practice. A typical example is to map a given anonymized DNA methylation profile to a genotype in order to re-identify it. Indeed, genomic data can facilitate de-anonymization, because there are already many profiles publicly available online with real identifiers, but also because it includes information about phenotypic traits, and kinship that can be further matched to side channels such as surname-genome associations databases [21] or online social networks [22]. Moreover, the genome is very stable over our whole lifetime, and thus cannot be revoked.

Note that we assume the adversary to have no prior knowledge about the presence of the target's genotype in the set of genotypes. Thus, the adversary also wants to determine whether the genomic profile that most likely matches to DNA methylation profile belongs to the same person. In other words, the adversary also tests if the owner of the DNA methylation profile is also part of the genomic dataset. We also study if familial relationships can mislead the adversary about the genotype corresponding to the methylation profile.

In the private classification model, we consider an honest-but-curious adversary as this assumption is standard in previous works on privacy-preserving medical diagnosis in a clinical setting [23], [24], [25], [26]. Indeed, it seems reasonable to assume that involved parties in the healthcare setting, such as hospitals or medical practitioners, will follow the protocol honestly. We leave the strengthening of our protocols to work with active adversaries for future investigations.

IV. Attack Methodology

We present here our de-anonymization attack from a theoretical perspective. The attack relies upon the matching of one or multiple DNA methylation profiles to their corresponding genotypes. To do so, the adversary first infers the probability of a genotype given only methylation data, and second maps the methylation profile to the genotype that maximizes the average posterior probabilities between genotypic positions and methylation sites. Once the best matching has been found by the adversary, he also wants to be sure that the methylation and genotypic samples in the matching pair belongs to the same person. Indeed, it could be that an individual is part of the DNA methylation dataset but not of the genotype dataset, or vice versa. To verify this, the adversary relies on a test statistic related to the matching score that provides him with a degree of certainty about whether the matching between methylation data and genotype is significant enough to be considered correct. If there is not enough certainty, the adversary can conclude that the corresponding genotype is most likely not part of the dataset.

A. Learning the Attack Model

The probabilistic relationships between methylation levels and genotypes are derived by relying on a separate training dataset $\mathcal{T} = \{(\vec{m}_i, \vec{g}_i)\}_{i=1}^{t}$ containing t pairs of DNA methylation levels' profiles and their corresponding genotypes. In practice, methylation profiles $\vec{m}_i$ and genotypes $\vec{g}_i$ have tens of millions of different positions. Specifically, the training phase aims: (i) to determine the meQTLs, i.e., the positions q in the genotype influencing the methylation levels in a region r, and (ii) to learn the magnitude of this influence. During this training phase, we select a subset $\mathcal{G}$ of n independent meQTLs g_i^q, and determine, for each of them, the single most correlated methylation region m_i^r over all the t pairs. In case more than one methylation region is most correlated with the same meQTL, we pair the highest correlated methylation region with the given meQTL first, and then pair the other methylation region with the second most correlated meQTL, and so on and so forth. This eventually provides us with a set of methylation region-meQTL position pairs $\mathcal{Q} = \{(r_j, q_j)\}_{j=1}^{n}$, where $\forall (r_j, q_j), (r_k, q_k) \in \mathcal{Q} : r_j \neq r_k \Leftrightarrow q_j \neq q_k$.

Once we have identified the positions in the genotype that influence most DNA methylation, we are interested in inferring the posterior probability of every meQTL g_j^i given the corresponding methylation region m_j^i, $\Pr(G_j^i = g_j^i \mid M_j^i = m_j^i)$. In this probability, G_j^i denotes the discrete random variable of the meQTL at position q_i of individual j, where $g_j^i \in \{0, 1, 2\}$ for any q_i and j, and M_j^i denotes the continuous random variable representing the methylation levels of individual j averaged over all CpG sites within region r_i, where $m_j^i \in [0, 1]$. By Bayes theorem, we have that:

$$\Pr(G_j^i = g_j^i \mid M_j^i) = \frac{p(M_j^i \mid G_j^i = g_j^i) \Pr(G_j^i = g_j^i)}{\sum_{g_j^i} p(M_j^i \mid G_j^i = g_j^i) \Pr(G_j^i = g_j^i)} \tag{1}$$

The prior genotype probabilities $\Pr(G_j^i = g_j^i)$ can be retrieved from population statistics databases, such as dbSNP,[1] or directly computed on any dataset of populations with similar ethnicity background. Moreover, we can learn the conditional probability distributions $p(M_j^i \mid G_j^i = g_j^i)$, for all $g_j^i \in \{0, 1, 2\}$, by relying on our training dataset $\mathcal{T}$, focusing only on the meQTL-methylation pairs contained in $\mathcal{Q}$. In this process, we must select the continuous distribution function that best fits the methylation-meQTL data. We discuss what distribution function fits best in Section VI.

B. Matching Attack

After having trained $p(M_j^i \mid G_j^i = g_j^i)$ for all pairs in $\mathcal{Q}$ and, for each pair, all three possible genotype values, on the training dataset $\mathcal{T}$, we can predict the posterior probabilities $\Pr(G_j^i = g_j^i \mid M_j^i)$ of the n meQTLs in $\mathcal{G}$ given methylation profiles in another dataset, referred to as the test set in the following. The test set consists of two independently chosen subsets: (i) a set $\mathcal{S} = \{(\vec{s}_i)\}_{i=1}^{n_g}$ containing $n_g \geq 1$ genotypes, and (ii) a set $\mathcal{E} = \{(\vec{e}_i)\}_{i=1}^{n_m}$ containing $n_m \geq 1$ methylation profiles. Note that individuals in $\mathcal{S}$ and $\mathcal{E}$ may be different, and that the adversary wants to infer the links between $\mathcal{S}$ and $\mathcal{E}$. In this endeavor, the adversary must compute, for all meQTLs in $\mathcal{G}$ and $n_g \times n_m$ pairs of individuals' in the test set, the posterior probabilities of the actual value of the genotypes given the methylation sites (by using the previously learned probabilities), i.e., $p_{j,k}^i := \Pr(G_j^i = s_j^i \mid M_k^i = e_k^i)$.

We derive a match score $w_{j,k}$ between individuals j and k by averaging the conditional probabilities $p_{j,k}^i$ over all n meQTL-methylation pairs in $\mathcal{Q}$, i.e., $w_{j,k} = \frac{1}{n} \sum_{i=1}^{n} p_{j,k}^i$. We then select the matching α^* over $(\max(n_g, n_m))! / (\max(n_g, n_m) - \min(n_g, n_m))!$ possible assignments that maximizes the sum of the individual match scores, i.e.,

$$\alpha^* = \arg\max_{\alpha} \sum_{k=1}^{n_m} \sum_{j=1}^{n_g} w_{j,k} \tag{2}$$

$$= \arg\max_{(j,k)} \frac{1}{n} \sum_{k=1}^{n_m} \sum_{j=1}^{n_g} \sum_{i=1}^{n} \Pr(G_j^i = s_j^i \mid M_k^i = e_k^i). \tag{3}$$

[1] https://www.ncbi.nlm.nih.gov/SNP/

This problem boils down to finding a best vertex matching on a weighted bipartite graph, with n_g vertices on one side representing the genotypes of n_g individuals, and n_m on the other side representing the methylation profiles of n_m individuals. Each edge between any two vertices pair (j, k) has a weight equal to $w_{j,k}$. As the number of possible assignments increases with $O(\max(n_g, n_m)^{\min(n_g,n_m)})$, the naive matching approach is computationally intractable if both n_g and n_m are big. Fortunately, there exist several algorithms in the literature that find the maximum weight assignment in polynomial time. In our experiments, we use the blossom algorithm [27], because it only has a complexity of $O((n_g+n_m)^3)$ and it can also be applied to general graphs. Of course, if $n_m = 1$ or $n_g = 1$, there is no need to use any maximum weight assignment algorithm as one can simply select the genotype $\vec{s}_j$, respectively methylation profile $\vec{e}_k$, maximizing $w_{j,1}$, respectively $w_{1,k}$, and the complexity is then linear in n_g, respectively n_m.

C. Statistical Validation of the Best Matching

In order to evaluate the significance of the match score between genotype $\vec{s}_j$ and methylation profile $\vec{e}_k$, we rely on the z-test and the corresponding z-score, defined as $z_{j,k} = (w_{j,k} - \mu(\vec{w}_k))\,/\sigma(\vec{w}_k)$, where $\vec{w}_k$ is the vector of match scores between methylation profile of individual k, $\vec{e}_k$, and all genotypes in $\mathcal{S}$, $\mu(\vec{w}_k)$ is its mean, and $\sigma(\vec{w}_k)$ is its standard deviation. The z-score can be similarly derived between the genotype $\vec{s}_j$ of individual j and all methylation profiles in $\mathcal{E}$. The only requirement is that the cardinality of the set over which we compute the mean and variance is large enough. The z-score allows us to determine, once a methylation profile is mapped to a genotype, whether these two profiles correspond to the same individual. Indeed, the pair that maximizes the match score might not be the one between the profiles of the same individual, especially when the individual's data is not part of one of the sets $\mathcal{E}$ or $\mathcal{S}$. In this case, we should be able to detect that the mapped pair does not contain the same individual. This is done by validating the mapped pair for a z-score greater than a given threshold.

If n_m-by-n_g matching becomes computationally infeasible, it is worth noting that it is also possible to map methylation profiles one-by-one to genotypes, i.e., carry out n_m times a one-by-n_g matching whose complexity is then linear in $n_m n_g$. Moreover, it can occur that the adversary has access to multiple methylation profiles of the same person but at different points in time. In this case, it can also be beneficial to rely on the one-by-n_g matching, which allows multiple methylation profiles to be mapped to the same genotype, contrary to the bipartite graph matching. In case the adversary is certain that there is only one methylation profile per individual, the n_m-by-n_g matching outperforms the one-by-n_g matching (see Section VI), but if he is not sure about the number of methylation profiles per individual, the n_m-by-n_g matching becomes more challenging to use.

V. Dataset

The dataset that was used in this study consists of meQTLs determined from a set of 75 individuals, 42 of which have parental relations (21 mother/child pairs) for which whole blood was available. The DNA methylation was determined using whole genome bisulfite sequencing (WGBS), allowing a genome wide measurement of the DNA methylation levels for all 28 million CpG dinucleotides. The sequencing data was processed using an in-house processing pipeline consisting of alignment of the sequencing reads, quality assessment, and methylation calling. Then, the genotype was determined at known SNP loci listed in the dbSNP database version 141, using the Bis-SNP tool, which calls SNP genotypes from WGBS data [28]. For the majority of individuals (67 out of 75), samples collected at the birth of the child, referred to as t_0, were available, but also at later times: one year (t_1), up to 8 years (t_8) for some individuals after birth.

Such a longitudinal dataset containing individuals with parental relations represents a very unique and valuable data source in the biomedical community. Note that this dataset cannot be released publicly yet, but will be certainly made available to researchers in a near future.

On a subset of these samples, we selected the CpGs based on their high variance across the dataset. CpG showing a very stable methylation profile across the subset of samples were discarded, as they are not expected to be under the influence of meQTLs. meQTLs were determined using a Spearman rank correlation test [29] (false discovery rate threshold after Benjamini-Hochberg correction [30] of 1%) for all SNPs located within 50 kb (kilobases) up-/downstream of the CpG showing highly variable methylation. This filtering process eventually output 568,103 meQTL-methylation pairs containing 502 methylation regions and 544,762 different SNPs. This implies an average number of approximately 1132 meQTLs per methylation region.

VI. Attack Evaluation

We present here our main experimental results, using the dataset described in the previous section. As explained in Section IV, the training phase relies on two different phases: (i) identify the meQTLs, i.e., the positions in the genotype that influence the methylation levels, and (ii) quantify the magnitude of this influence. As we carry out the first step similarly for all experiments, we present it first. This can also be seen as a data preprocessing step, which filters out non-relevant genotypic positions and methylation regions.

A. Generic Training Phase

We focus here on the meQTL-methylation pairs with a Spearman rank correlation coefficient larger than 0.49 (FDR threshold after Benjamini-Hochberg correction of 1%). This provides us with 326 methylation regions and 9,532 pairs, i.e., around 29 meQTLs per methylation region. Then, we keep only one most correlated meQTL for each methylation region, resulting in 326 pairs, as expected. Filtering out the meQTLs for which no information was available on dbSNP, we are left with 314 meQTL-methylation pairs. Finally, since we have to compute the variance (see below) of the conditional probability $p(M_j^i \mid G_j^i = g_j^i)$ for all possible values of g_j^i, we filter out meQTLs that do not have at least two samples per genotype value g_j^i. This eventually led us to a total of 293 meQTL-methylation pairs for the whole dataset.

Normal Distribution Function: The first step towards precisely modeling the influence of meQTLs on methylation regions is the selection of the continuous distribution function that best fits the observed data. We rely on the normal distribution which happens to be well fitted from both a visual and statistical perspective. First, in order to evaluate if the normal distribution approximation was statistically significant, we applied the one-sample Kolmogorov-Smirnov test to all 293 meQTL-methylation region pairs and possible genotype values, $g_j^i \in \{0,1,2\}$. The null hypothesis (the samples belonging to the normal distribution) was only rejected in a minority of cases at significant level 0.05 (134 out of 879). When we inspected those few cases manually, we found that all of those cases contained either a very few outliers or almost all of the methylation levels belonged to the exact same bin in the histogram and thus were almost exactly the same.

We also visually inspected the empirical conditional distributions $\hat{p}(M_j^i \mid G_j^i = g_j^i)$ for $g_j^i \in \{0,1,2\}$ and reached the same conclusion. Fig. 1 exemplarily shows Q-Q plots as well as the empirical distribution of methylation levels given each possible genotype of a representative pair (M_j^i, G_j^i) in our dataset. Moreover, it also displays the corresponding normal distributions induced by the unbiased estimators of the mean and standard deviation. The Q-Q plots depict on the x-axis the theoretical quantiles of a standard normal distribution. The y-axis displays the normalized quantiles of the sample distribution for each $G_j^i = g_j^i$. Given the minor discrepancies between the points and the diagonal, we can expect that the normal distribution will be a sufficiently good fit for the attack. Second, the part of the figure at the bottom right confirms that the normal distribution indeed is a good approximation for the conditional probability. More importantly, it also shows that the overlap between the distributions conditioned on different genotype values is small, which can be used to recover the correct genotype given the methylation level. This gives the intuition behind our re-identification attack.

B. Experiment-specific Training and Testing Sets

In this second phase, we quantify the magnitude of the influence of each meQTL on its corresponding methylation region. From now on, in order to illustrate the performance of the attack under different scenarios, we build our training dataset from different subsets of the whole dataset described in Section V. We consider three different training/testing experimental setups. In the first scenario, referred to as (a), we select one methylation profile per individual, i.e., 75 profiles, as follows: we pick the 67 profiles available at time t_0 and, in addition, the profiles of individuals not yet selected at t_0 (because of absence of data) at the smallest time point as possible: 1 at t_1, 1 at t_3, 3 at t_4, 2 at t_5, and 1 at t_6. We further select the 75 genotypes corresponding to these methylation profiles. Then, we randomly select 37 pairs for the training set, and 38 for the testing set, or attack set. We repeat the random splitting 100 times.

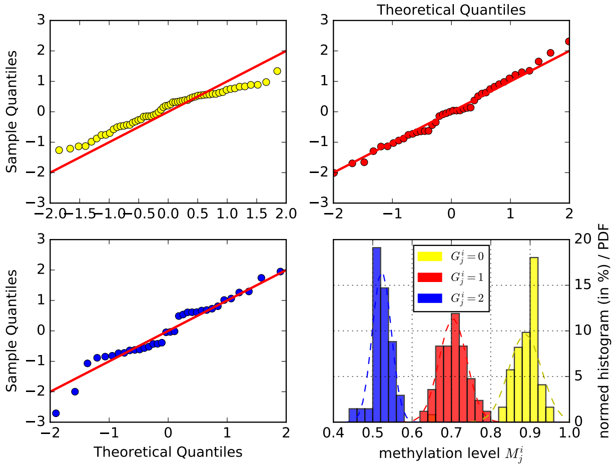

Fig. 1. Example of the empirical distribution $\hat{p}(M_j^i \mid G_j^i)$ of methylation levels conditioned on genotype values $g_j^i = \{0,1,2\}$ for the pair with meQTL rs10928633 (in chromosome 2, position 138625907) and methylation region $[138625907, 138626564]$ in the same chromosome. Red color (top-left plot) is $\hat{p}(M_j^i \mid G_j^i = 0)$, blue color (top-right plot) is $\hat{p}(M_j^i \mid G_j^i = 1)$, and green color (bottom-left plot) is $\hat{p}(M_j^i \mid G_j^i = 2)$.

In the second setup, (b), we want to make sure that there are no individuals in the training and testing sets who have familial relationships, i.e., we want to avoid a child being in the training set, and his mother being in the test set, or the other way around. We also aim at 37 samples in the training set and 38 in the test set. Thus, we first randomly select from 2 to 18 mother-child pairs to be included in the training set, which leads us to 4 to 36 samples. Then, we randomly select the remaining samples among the isolated individuals (i.e., those who have no child or mother in our dataset) to attain 37 samples. We repeat this random selection 100 times, and select the 38 remaining profiles to be part of the test set. This process ensures that there is no individual in the test set who is member of the same family as somebody in the training set.

The third experimental setup, (c), is used for the scenarios where we want to map more than one methylation profile at a time with the genotypes. In both previous settings, we consider $n_m = 1$ and $n_g = 75$ (or more, as we will see later), but we repeat the attack over all 38 methylation profiles independently. Now, we want to match $n_m > 1$ methylation profiles to $n_g = 75$ genotypes. We then select our samples in order to maximize the number of methylation profiles in the test set, as follows. We select all individuals at time t_1 and at time points $t > t_1$ that do not have methylation profiles at t_0 and t_1. This gives us 16 methylation profiles at t_1 plus 7 at later time points, thus 23 methylation profiles for the training set. Then, for the test set, we select all methylation profiles at t_0 whose owners do not overlap with those in the training set.

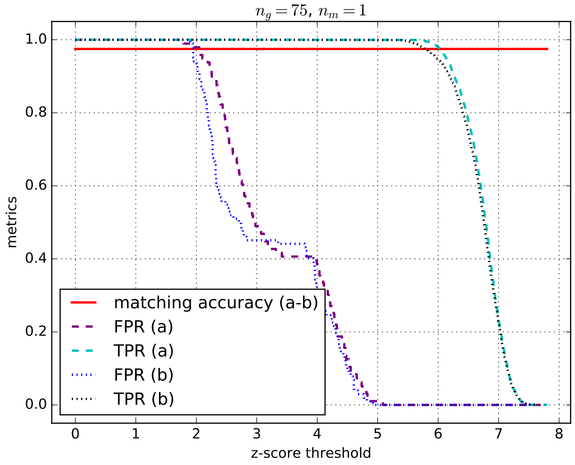

Fig. 2. Identification of one methylation profiles among 75 genotypes: Average accuracy of the matched pairs, and true-positive, false-positive rates for a varying z-score threshold.

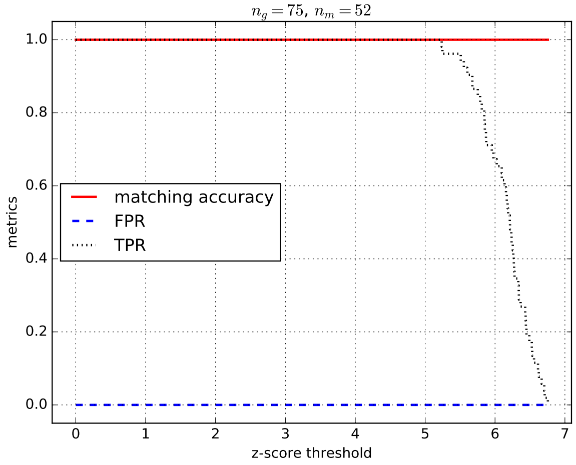

Fig. 3. Identification of 52 methylation profiles among 75 genotypes: Average accuracy of the matched pairs, and true-positive rate for a varying z-score threshold.

This leads to 52 methylation profiles for the test set.

Note that the requirement of having two samples per genotype value to learn the variance of the normal distribution is reducing the number of meQTL-methylation pairs when we apply it to the training set and not the whole dataset. The total number of pairs ranges from 237 to 248 with a median value 240 in setup (a). It ranges from 208 to 236 with a median of 222.5 for (b), and it is of 187 pairs for setup (c) for which there is only one run and the number of samples in the training set is smaller (due to stronger constraints).

C. Results

We start by showing the performance of the attack with all available meQTL-methylation pairs (given the aforementioned constraints), $n_m = 1$ and $n_g = 75$. We include all 75 individual genotypes to be potentially matched to the methylation profiles as we assume that this can only make the attack harder for the adversary than considering only the 38 or 52 genotypes corresponding to the methylation profiles of the test set. Of course, we only select the 38 methylation profiles present in the test set to run our experiments. Therefore, we try to match one methylation profile with 75 genotypes, 38 times, over 100 runs, i.e., 3,800 times, and average the results.

Fig. 2 shows: (i) the matching accuracy, i.e., the fraction of pairs containing genotypes and methylation profiles of the same individual, (ii) the true-positive rate (TPR) after applying the z-score test, i.e., the number of true matchings divided by the sum of the number of true matching pairs and the number of matching pairs that are wrongly identified as non-matching, and (iii) the false-positive rate (FPR) after applying the z-score test, i.e., the number of false mappings that are identified as true divided by the sum of the latter value and the number of true mappings identified as false. We could have also depicted other metrics, such as accuracy after z-score, but we consider the TPR and FPR as sufficient metrics to depict the success of the identification attack.

First, Fig. 2 shows that, on average, the attack accurately matches the methylation profile to its corresponding genotypes around 97.5% of the time. Then, we notice that, there exists a z-score for which, given a certain matching, we always reject all wrongly matched pairs (FPR = 0 for z-score approximately greater than 5), and never reject those that are correct (TPR = 1 for z-score approximately smaller than 5.5). This means that for the 2.5% of the pairs that are wrongly matched, we are able to identify that they are false positives. Finally, we notice that the matching accuracy is the same for both scenarios (a) and (b), and that the FPR and TPR are also very similar.

Fig. 3 shows the attack when there are more than one methylation profiles to match to their genotypes. Specifically, given the experimental setup (c), we have 52 methylation profiles that we try to match again to the whole 75 genotypes. First of all, we notice that the matching accuracy is 100%, i.e., that the attack correctly matches the 52 methylation-meQTL pairs. Then, by looking at the z-score to validate the matched pairs, we note that it starts rejecting valid pairs from around 5.2. As we only have correctly matched pairs after the matching algorithm, there is no point in displaying the FPR because there is no wrong pair to reject. We conclude from Fig. 2 and 3 that the attack is more successful when matching more than one methylation profile to multiple genotypes.

Next, we evaluate the impact of reducing the number of methylation-meQTL pairs on the attack success. In this endeavor, we gradually use an increasing number of observed methylation-meQTL pairs, from 1 to 237, in decreasing order of correlation. Fig. 4 shows the evolution of the matching accuracy and of the TPR after applying the z-test, for three possible FPR vaues: 0, 0.05, and 0.1. First, we notice that we reach the maximum matching accuracy with only 20 methylation-meQTL pairs, and almost 90% accuracy with 10

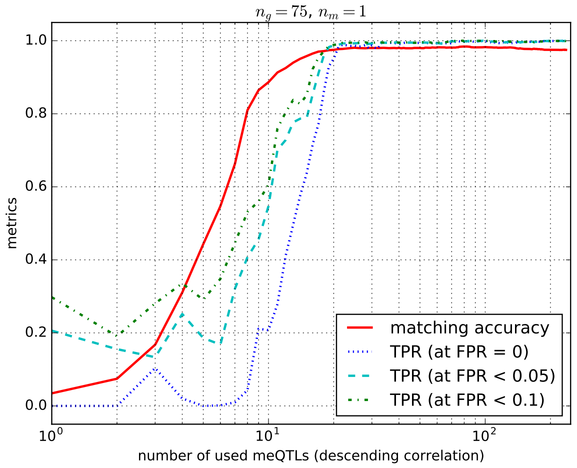

Fig. 4. Identification of one methylation profile among 75 genotypes with an increasing number of observed meQTLs/methylation regions (in descending levels of correlation): Average accuracy of the matched pairs, and true-positive rates at various false-positive levels.

pairs. Second, we see that we attain a TPR of 0.6 at a FPR of 0.05 when we apply the z-test (at 10 pairs). Furthermore, we reach a 0.95 TPR at 0.05 FPR with 20 methylation-meQTL pairs, and 0.99 with 30 pairs.

When evaluating the same experiment with a fixed threshold of 5.5 (as found suitable in Fig. 2), we notice that 80 methylation-meQTL pairs are necessary to achieve a TPR of almost 0.9 and a FPR of 0. This arises from the fact that a larger number of methylation-meQTL pairs provides more information and thus gives a more accurate match score, which also allows for higher z-score thresholds to perform better.

Similarly, Fig. 5 shows the evolution of the various metrics with respect to an increasing number of observed methylation-meQTL pairs, for $n_m = 52$. The less smooth behavior of the curves is due to the fact that we have one run here compared to 100 runs in the case where $n_m = 1$. We notice here that the matching accuracy and TPRs reach highest values for a number of methylation-meQTL pairs that is lower than when $n_m = 1$. Precisely, the attack reaches full accuracy and TPR at 0 false-positives with only 13 pairs. Again, we see that matching more than one methylation profiles to their corresponding genotypes induces higher attack success.

We evaluate now how the attack performance evolves when the genotype corresponding to the targeted methylation profile is not present in the genotype dataset. We have $n_g = 74$ genotypes if the targeted genotype is not present and, for the sake of comparison, we keep the same number when it is present, by removing another of the 74 genotypes at random. Fig. 6 shows the evolution of this performance with respect to an increasing probability that the targeted genotype is in the dataset, from 0 to 1, by intervals of 0.01. For each probability value x, we randomly generate a value v between 0 and 1, uniformly, and keep the targeted genotype in the dataset if and only if $v < x$. We repeat this sampling process 100 times

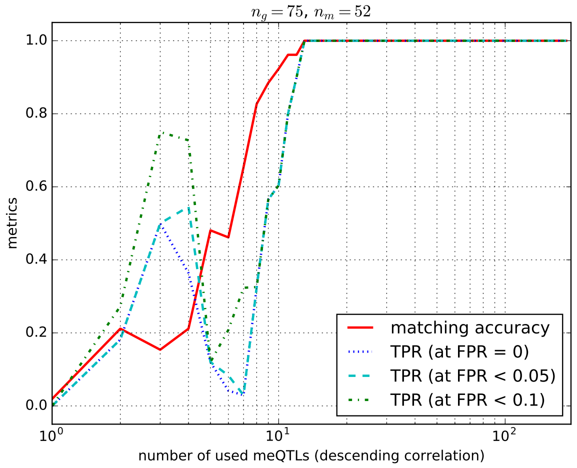

Fig. 5. Identification of 52 methylation profiles among 75 genotypes with an increasing number of observed meQTLs/methylation regions (in descending levels of correlation): Average accuracy of the matched pairs, and true-positive rates at various false-positive levels.

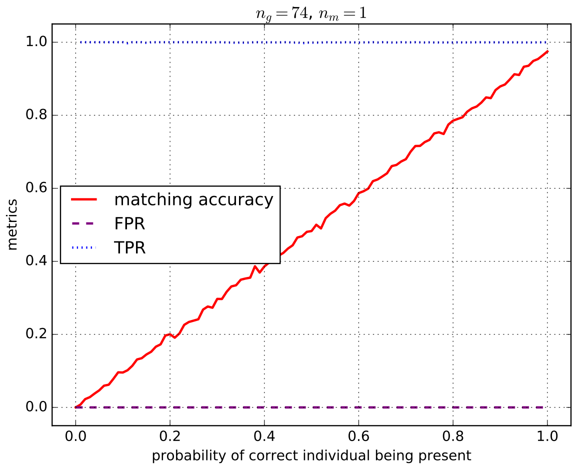

Fig. 6. Identification of one methylation profiles among 75 genotypes with an increasing probability of the correct matching genotype being present in the dataset: Average accuracy of the matched pairs, true-positive and false-positive rates.

and average its outcomes. As expected, the matching accuracy increases with the probability that the correct genotype is present in the dataset. The adversary cannot find the correct genotype if it is not there. The crucial point here is that the adversary can detect that the genotype is not present for any presence probability. Indeed, with the appropriate z-score (between 4.9 and 5.4), the adversary always rejects the wrongly matched genotypes (FPR=0) while accepting the correctly matched genotypes (TPR=1).

We also investigate the effect of a relative's genotype being in the genotype dataset, with a varying presence probability of the targeted genotype, as in Fig. 6. The relative here is either the mother or the child of this mother. Fig. 7 shows

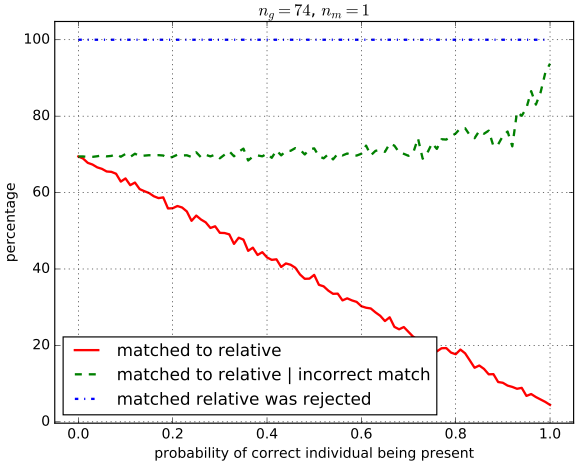

Fig. 7. Wrongly matched first-degree relatives in the identification of one methylation profiles among 74 genotypes with an increasing number of observed meQTLs/methylation regions (in descending levels of correlation): Average accuracy of the matched pairs, and true-positive rates at various false-positive levels.

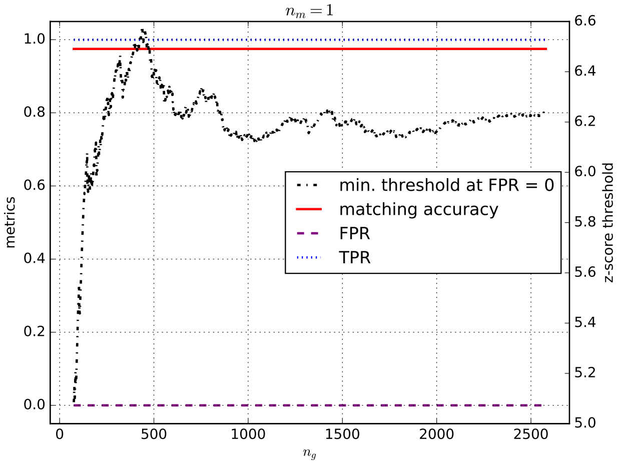

Fig. 8. Identification of one methylation profiles among an increasing number of genotypes, from 75 to 2579: Average accuracy of the matched pairs, true-positive and false-positive rates and minimum z-score threshold for a null false-positive rate.

the percentage of times the relative's genotype is matched to the methylation profile, in absolute value, and relative to the condition that the matched pair was wrong, and the percentage of times this wrongly matched pairs were rejected by the z-test. First, we observe a linear decrease of the probability of being matched to the relative with respect to the presence probability. We also see that this curve does not start at 1 but at around 0.7. This means that, when the targeted genotype is not in the dataset, the wrongly matched genotype is in 70% of the cases the relative's genotype, and in the 30% remaining cases the one of an unrelated individual.

In order to better understand these proportions, we display the fraction of familial matches among all wrong matches (green dashed curve). We observe that this fraction increases with the presence probability. In order to understand this behavior, we must recall that the matching accuracy also increases with the presence probability. This means, that the fewer wrong matched pairs there are, the more likely these are pairs containing the genotype of a relative and not of an unrelated individual. Also, it means that, when the chance that the targeted genotype is present in the dataset is high, the only genotype that can mislead the adversary's matching is the relative's genotype in the vast majority of cases.

Finally, we study the robustness of our attack for an increasing number of genotypes, from 75 to 2579, by including the 2504 genotypes of the 1000 Genomes Project (phase 3) [31]. Fig. 8 shows the evolution of the matching accuracy, of the false-positive and true-positive rates after the z-test, of the minimum z-score for reaching a null FPR. First, we notice that the matching accuracy remains constant, at 97.5%, for all genotype dataset's size n_g. Moreover, there always exists a z-score that enables us to reject all wrongly matched pairs while keeping all correctly matched pairs. We notably notice that this z-score evolves quite a lot until around $n_g = 1000$ and that it tends to converge to a fixed value when n_g gets closer to 2579. We conclude from this figure that the identification attack is very robust to an increase in the number of genotypes we have to match the methylation profile to.

We also evaluated this experiment with fixed thresholds on the z-score. When less than 100 genotypes are present, a threshold of 5.5 provides a TPR of 1 and FPRs below 0.05. When more than 100 genotypes are part of the test set, a threshold of 6 achieves the same effect. Since these observations conform with previous experiments, we believe that an adversary is able to determine a suitable threshold from her training data.

VII. Private Classification with Random Forests

As we have shown, publicly releasing methylation profiles has a huge detrimental effect on the patients' privacy, with a risk close to 100% to have one's methylation data re-identified. Therefore, we first strongly recommend to reconsider if the existing DNA methylation datasets should remain publicly available in online databases. Moreover, it is vital to understand the needs of the medical community for designing appropriate protection mechanisms that provide privacy guarantees and diagnosis utility to the patients. In this section, we propose a novel cryptographic scheme for privately classifying tumors based on random forests. We first describe the preliminaries on random forests, and then present our private random forest-based classifier.

Random forests are a promising technique used in the medical community for classifying diseases [32]. This ensemble method bases its classification on a multitude of classification trees in order to prevent overfitting and to reduce the prediction variance [33]. Danielsson et al. for example developed a random forest classifier tool enabling the identification of pediatric brain tumor subtypes with an accuracy of 98% [34].

In practice, when diagnosing a patient's disease, a sample is taken from the patient by a medical practitioner. Then, the sample needs to be analyzed either by the hospital or by a medical laboratory, resulting, e.g., in the DNA methylation profile of the patient. The actual classification based on these data can then be outsourced to a third-party company providing data-driven medicine, such as Sophia Genetics [35]. The DNA methylation profile is sent to the third party, which then provides the diagnosis to the physician or hospital. While the business model of this third party is inherently protected by keeping the classification model secret, the patient's privacy is clearly at risk, as his data are available to the third party.

Hence, when classifying a patient's disease, two privacy goals must be achieved: (1) protecting the company's classification model, and (2) protecting the patient's data from the third-party company. Note that, in order to construct its classifier, the company must have access to a training set of DNA methylation data in clear. Our scheme protects the data on which only classification has to be carried out (e.g., for diagnostic purposes). Finally, our scheme is flexible in the sense that it can release two outcomes: (i) only the class with the plurality vote (most frequently chosen by the random forest algorithm), or (ii) the class of every tree in the random forest, which enables the medical practitioner to carry a more fine-grained analysis of the distribution over the possible classes.

A. Preliminaries

1) Classification Trees: Classification trees (or decision trees) are a popular, predictive tool in machine learning, used to classify an input $\vec{v}$ into a set of different classes $Y = \{y_0, \ldots, y_k\}$. As the name suggests, a classification tree can be represented by a simple, usually binary tree, in which each interior node corresponds to an input value v_i. The two edges of each interior node partition the node's input domain into two distinct sets. Each leaf node of the tree is labeled with a class y_j. It is worth noting that a single class may occur at more than one leaf.

In order to classify an input using a classification tree, one starts at the root node and walks down the tree until a leaf node is reached. At each interior node, the decision which edge to select is determined by the partition to which the corresponding input value belongs. Finally, the class label of the leaf node determines the result of the classification task. In the following, we will focus on the most common form of classification trees as implemented in many libraries: binary classification trees in which the partitioning at each interior node is given by a comparison of the input value with a threshold w_i. The model of such a classification tree is completely described by the structure of the tree, the input values v_i corresponding to each node, as well as the thresholds w_i applied at each node.

2) Random Forests: Classification trees usually suffer from a high prediction variance and can easily suffer from overfitting to their training set. In order to reduce the prediction variance, random forests put together multiple noisy, but approximately unbiased classification trees.

In general, a random forest consists of B classification trees, where the number B is subject to tuning. The training of a random forest is performed on a training dataset $T = \{(\vec{x}_1, y_1), \ldots, (\vec{x}_n, y_n)\}$, consisting of n samples together with their corresponding class label. During the training, each tree is grown on n randomly chosen (with replacement) training samples using only a randomly chosen set of input predictors (components of the training samples) $K \subseteq \{1, \ldots, len(\vec{x})\}$. This random subset of input predictors is what distinguishes random forest from simple tree bagging and ensures the trees to be *de-correlated* so that the same input predictors are not used in all of the trees. This step is important to reduce the correlation of the trees, which then enables further reduction of the prediction variance [33].

Given a random forest model and an input $\vec{v}$, the classification algorithm evaluates each of the model's trees individually. Then, depending on the application, implementation or preference, the resulting class can be determined by plurality vote (or majority vote for binary classification), averaging the class predictions or providing class probabilities in terms of relative vote counts.

B. Private Classification with Random Forests

Next, we introduce our construction that enables to securely evaluate random forests between a third party and a querier. More specifically, we do not want the querier (referred to as client) to learn the structure of the trees, nor should the third party (referred to as server) learn anything about the input sample or the result of the classification.

We build our construction on top of the work of Bost et al. [23] and extend it to work with random forests. In their work, they introduced three major classification protocols, namely for hyperplane decision, Naïve Bayes, and classification trees, all satisfying the constraint to keep both the classifier model and the data confidential. Since classification trees are an important component of random forests, we first recap the details of the classification tree protocol, before extending it to random forests.

It is important to note that the classifier is trained upfront on data in the clear, whereas only the actual classification of new samples is performed securely on encrypted data.

1) Cryptosystem and Notation: In the following, we will rely on three different additively homomorphic public-key cryptosystems. An additively homomorphic public-key encryption scheme allows, given the two encrypted messages $Enc(a)$ and $Enc(b)$, to compute $Enc(a + b)$ using a public-key operation on the encrypted messages. Moreover, one of our cryptosystems is a leveled fully homomorphic encryption, which also allows to perform a bounded number of multiplications in sequence, i.e., to compute $Enc(a \cdot b)$ on the encrypted messages. Bounded means that the cryptographic scheme allows to evaluate polynomials only up to a certain multiplicative depth L. Below, we list the cryptosystems we use and also mention the corresponding plaintext spaces M:

1) the QR (Quadratic Residuosity) cryptosystem of Goldwasser-Micali [36] ($M = \mathbb{F}_2$, bits),

2) the Paillier cryptosystem [37] ($M = \mathbb{Z}_N$ with N being the public modulus of Paillier),
3) a leveled fully homomorphic encryption (FHE) scheme based on the Brakerski-Gentry-Vaikuntanathan [38] scheme as implemented by HELib [39] ($M = \mathbb{F}_2$).

We denote the client in our protocols by C and the server by S. $[b]_A$ denotes a bit b encrypted by the QR scheme under party A's key (so only A can decrypt the message using her secret key). Similarly, $[\![m]\!]_A$ denotes an integer m encrypted by the Paillier scheme, and $[\![b]\!]_A$ denotes a bit b encrypted by the leveled FHE scheme. SK_A^s is used for party A's secret key for the encryption scheme Paillier ($s = P$), QR ($s = QR$) or leveled FHE ($s = FHE$), and PK_A^s is the respective public key. For a distribution $\mathcal{D}$, $a \leftarrow \mathcal{D}$ means that we assign a a random sample from that distribution.

2) Cryptographic Assumptions and Adversarial Model: The security of our protocol relies on the semantic security [40] of the cryptosystems we use and, hence, also on the well-studied assumptions underlying those systems, namely the Quadratic Residuosity assumption, the Decisional Composite Residuosity assumption, and the Ring Learning With Errors (RLWE) assumption.

We prove our protocol to be secure in the two-party computation framework for passive adversaries (or honest-but-curious [40]), by relying on modular sequential composition of smaller protocols as described below.

3) Building Blocks: Specifically, we will reuse existing building blocks from the work of Bost et al. and also design a new one that is needed for our protocol: changing encryption ownership. Their work already introduced several smaller building blocks, such as different comparison protocols on encrypted data, or a protocol to evaluate the $\arg\max$ function on encrypted data. Those building blocks necessary for our own construction are briefly reviewed hereunder, before we introduce our own building blocks as well as the full construction.

a) Comparison Protocols: Bost et al. introduce five slightly different comparison protocols, two of which we will need in our construction. Let A, B be two parties. A has $\mathsf{PK}_B^P, \mathsf{PK}_B^{QR}$ and B has the corresponding secret keys $\mathsf{SK}_B^P, \mathsf{SK}_B^{QR}$.

The first comparison protocol (referred to as (1) later) assumes that A has two values $[\![a]\!]_B, [\![b]\!]_B$. This protocol then allows to compare a and b, such that A learns $[a \leq b]_B$ and B learns nothing about the comparison.

The second comparison protocol, (2), works the same way, the only difference being that B also learns $a \leq b$.

More details as well as the other comparison protocols can be found in [23].

b) $\arg\max$ on Encrypted Data: Based on their comparison protocol (2), Bost et al. develop a protocol to compute the $\arg\max$ on encrypted data. Let A, B be two parties. A has k encrypted values $([\![a_1]\!]_B, \ldots, [\![a_k]\!]_B)$ (where k is also known to B) and wants to know the $\arg\max$ over unencrypted values (i.e., the index i of the largest value a_i), but neither party should learn anything else.

Hence, this protocol allows to compute $\arg\max_{1 \leq i \leq k} a_i$ given only the values encrypted under B's key. In particular, during the computation, B should neither learn the values a_i, nor should B learn the order relations between the a_i's. The full details of this protocol are described in [23].

c) Changing the Encryption Scheme: In order to convert ciphertexts from one of the cryptosystems to another, Bost et al. rely on a simple protocol to change the encryption scheme. Since this protocol is crucial for essential parts of our construction, we will provide a more detailed description of the protocol.

First, we consider the case, for which $M_{s_1} = M_{s_2} = \mathbb{F}_2$, i.e., the two cryptosystems have the same message space: Let A, B be two parties, A having $\mathsf{PK}_B^{s_1}, \mathsf{PK}_B^{s_2}$ and a ciphertext $c = Enc_{s_1}(x)$. B has the corresponding secret keys $\mathsf{SK}_B^{s_1}, \mathsf{SK}_B^{s_2}$. The goal is to re-encrypt x using the cryptosystem s_2, without B learning x.

Intuitively, the protocol works as follows. First, A uniformly picks a random noise $r \leftarrow M_{s_1}$, encrypts it using $\mathsf{PK}_B^{s_1}$ and adds it to the ciphertext c, before sending the result to B. B then decrypts the ciphertext to $x + r \in M_{s_1}$, re-encrypts it using $\mathsf{SK}_B^{s_2}$ and sends $Enc_{s_2}(x + r)$ to A, who can strip off r using the homomorphic property of s_2. B only obtains $x + r$, which hides x information-theoretically (this can be seen as a one-time pad).

For the second case, when $M_{s_1} \neq M_{s_2}$, we only require the transformation from $M_{s_1} = \mathbb{F}_2$ to $M_{s_2} = \mathbb{Z}_N$, i.e., from FHE to Paillier. Here, the beginning of the protocol remains the same and A obtains $[\![x \oplus r]\!]_B$ with $x, r \in \mathbb{F}_2$. The important difference to the previous case now arises when A wants to strip off $r \in M_{s_1} = \mathbb{F}_2$ from the encryption. Since the additive operation on $\mathbb{F}_2$ is $\oplus$ and on $\mathbb{Z}_N$ is $+$, we have to emulate $\oplus$ in Paillier's message space. This can be easily done by computing:

$$[\![x]\!]_B = \begin{cases} [\![x \oplus r]\!]_B & \text{if } r = 0 \\ g([\![x \oplus r]\!]_B^{-1}) \mod N^2 & \text{if } r = 1 \end{cases}$$

Before giving the result to an adversary, who knows $[\![x \oplus r]\!]_B$, but not SK_B^P, the obtained result has to be refreshed to preserve semantic security. A pseudocode implementation as well as the security and correctness proofs of this protocol can be found in [23].

d) Private Evaluation of Classification Trees: The most useful protocol is the one for privately evaluating a classification tree. Here, the main idea is to represent the classification tree as a polynomial P, whose output is the result of the classification.

Let b_i be the boolean outcome of a comparison between the ith node's input value v_j and the corresponding threshold w_i, i.e., $w_i < v_j$. Then, given the class labels $Y = \{y_0, \ldots, y_k\}$, one can express a classification tree by a polynomial. The polynomial is constructed recursively by a procedure $\mathcal{F}(T)$. If T is a leaf node, $\mathcal{F}(T) = y$, where y is the class label at the leaf T. If T is an internal node, and T_1 is the child tree in case the corresponding b is true, and T_2 is the child tree in case b is false, then $\mathcal{F}(T) = b\mathcal{F}(T_1) + (1-b)\mathcal{F}(T_2)$ is the polynomial that evaluates T_1 if b and T_2 otherwise.

Using this polynomial, Bost et al. then introduce a protocol to evaluate the tree, while revealing only the outcome and the number of comparisons. Let S and C denote the server and client respectively. First, S and C make use of the comparison protocol (1), so that S learns the bits $[b_i]_C$ for every node. Then, they interact in the protocol to change the encryption scheme from QR to FHE, obtaining $\llbracket b_i \rrbracket_C$.

The server S can then evaluate the polynomial P using the homomorphic properties of the FHE scheme. However, since the plaintext space is only $\mathbb{F}_2$ and the class labels potentially take more than one bit, we would have to evaluate the polynomial for each bit individually. Fortunately, the so-called SIMD slots of the FHE scheme (described in details in [41]) allow the scheme to encrypt a vector of bits in one ciphertext and evaluate the polynomial on the whole vector at once, in parallel. Hence, for each class label y_i, the server encrypts its bit representation $y_{i0}, \ldots, y_{il}$ using these SIMD slots to $\llbracket y_{i0}, \ldots, y_{il} \rrbracket_C$ and can evaluate the polynomial for each bit in parallel.

The client can later decrypt the resulting class label and convert it back to the normal integer representation. A more detailed explanation, as well as proofs of correctness can be found in [23].

e) Changing Encryption Owner: Next, we will introduce our protocol to change the ownership of an encryption, which we will need in order to apply the $\arg\max$ protocol in a way that only the client learns the result of the plurality vote.

Given two parties A and B, out of which A holds the encrypted message $\llbracket x \rrbracket_B$, we want B to hold the same encrypted message, but this time under A's key. However, neither A nor B should learn the message x itself. In the following, we design a protocol to meet this goal and provide the proof in the appendix.

Let A have $\mathsf{PK}_B^P, \mathsf{SK}_A^P, \llbracket x \rrbracket_B$ and B have $\mathsf{SK}_B^P, \mathsf{PK}_A^P$. Then A first blinds the encrypted message by uniformly sampling a random noise r from the plaintext space, encrypting it and adding it to the ciphertext. Then, A also encrypts r using his own secret key and sends both $\llbracket x + r \rrbracket_B$ and $\llbracket r \rrbracket_A$ to B. B then decrypts the first ciphertext to $x + r$, which hides x in an information-theoretic way and encrypts it again using PK_A^P. Then B strips off r using the sent encryptions without learning r itself and obtains $\llbracket x \rrbracket_A$.

The complete protocol is shown in Protocol 1.

Protocol 1 Changing Encryption Owner

```
Input: $A : (\llbracket x \rrbracket_B, \mathsf{SK}_A^P, \mathsf{PK}_B^P)$, $B : (\mathsf{PK}_A^P, \mathsf{SK}_B^P)$
Output: $B : \llbracket x \rrbracket_A$
$A$: uniformly pick a random noise $r \leftarrow M_P = \mathbb{Z}_N$ (Paillier's message space), encrypt it using $\mathsf{PK}_B^P$ and compute $\llbracket x + r \rrbracket_B$
$A$: encrypt $r$ using $\mathsf{SK}_A^P$ to $\llbracket r \rrbracket_A$
$A$: send $(\llbracket x + r \rrbracket_B, \llbracket r \rrbracket_A)$ to $B$
$B$: decrypt $\llbracket x + r \rrbracket_B$ to get $x + r$ and encrypt it using $\mathsf{PK}_A^P$ to $\llbracket x + r \rrbracket_A$
$B$: compute $\llbracket x \rrbracket_A = \llbracket x + r \rrbracket_A \cdot \llbracket r \rrbracket_A^{-1}$ using the homomorphic property
```

Theorem 1. *Protocol 1 is secure in the honest-but-curious model.*

The proof of the theorem is provided in the appendix.

4) Private Random Forests: Now that we introduced all building blocks necessary to privately evaluate a random forest, we first give an intuition of our protocol before presenting its pseudocode in Protocol 2.

Intuitively, one could just evaluate each tree of a random forest individually, given the protocol introduced by Bost et al., and return the outcomes to the client. The client is then able to compute the plurality vote or any metric she is interested in. This, however, will not only leak the number of trees, but most likely also the number of nodes within each tree to the client. Indeed, the scheme of Bost et al. reveals the number of comparisons, thus the number of inner nodes to the client. We modify this idea to only leak the total number of trees and the total number of nodes. Moreover, we extend it by giving the option to only reveal the plurality-vote class to the client. To this end, we do not evaluate one tree after another, but we perform the evaluations of all trees in a batch, e.g., running the comparison protocol for the b_i's of all trees in a row. This way, the client cannot distinguish between different trees during the evaluation.

In order to allow the protocol to only reveal the plurality-vote class, we have to modify the protocol further. Intuitively, for the server S to determine the plurality-vote class, S needs to be able to count the votes for each class without learning the actual outcomes of the trees. We can achieve this by slightly changing the way the class labels are encoded into the SIMD slots: Instead of encoding each integer class label as its binary representation, we encode a class label y_i by only setting the ith bit to 1. While encoding k labels into a binary representation needs only $\lceil log_2(k) \rceil + 1$ bits, our method will take exactly k bits. However, if enough SIMD slots compared to the number of classes are available, this should not have a substantial effect on the protocol's performance. More specifically, a class label y_i is now encoded as $(y_{i1}, \ldots, y_{ik})$ with $y_{ij} = 1$ if $i = j$ and 0 otherwise.

After obtaining the outcomes of all trees, the server and client interact to change the outcomes' encryption schemes from FHE to Paillier, resulting in ciphertexts for each outcome and class label $\llbracket y_{ij} \rrbracket_C$ for $i \in \{1, \ldots, n\}, j \in \{1, \ldots, k\}$,

Protocol 2 Evaluate a Random Forest

```
Input: Client $C : (\mathsf{SK}_C^P, \mathsf{SK}_C^{QR}, \mathsf{SK}_C^{FHE}, \mathsf{PK}_S^P, \vec{v})$, Server $S : (\mathsf{PK}_C^P, \mathsf{PK}_C^{QR}, \mathsf{PK}_C^{FHE}, \mathsf{SK}_S^P, \mathcal{F} = \{t_1, \ldots, t_n\})$
Output: Client $C$ : the outcome of evaluating $\mathcal{F}$ on $\vec{v}$ in terms of a plurality vote or the individual votes
$S$: produces the polynomials $P_1, \ldots, P_n$ for each tree in $\{t_i\}_{i=1}^n$
$C$: sends the encrypted query $[\![v_0]\!]_C, \ldots, [\![v_m]\!]_C$ to $S$
$S$ and $C$ perform the comparison protocol (1) on a shuffled order of the nodes, so that $S$ obtains $[b_i]_C$ for every node in the trees
$S$: changes the encryption obtaining $[\![b_i]\!]_C$
$S$: computes each class label $y_i$ by setting only the $i$th bit to 1 and encrypts the class labels using FHE and SIMD slots to $[\![y_{i1}, \ldots, y_{ik}]\!]_C$ with $y_{ij} = 1$ if $i = j$ and 0 otherwise
$S$: evaluates the polynomials using the fully homomorphic encryption, obtaining the encrypted outcomes $\{[\![y_{j1}, \ldots, y_{jk}]\!]_C\}_{j=1}^n$ for each tree
if $C$ is allowed to get all individual outcomes then
    $S$: rerandomizes the encrypted outcomes, shuffles their order and sends them to $C$, who can decrypt them
else
    $S$: rerandomizes the encrypted outcomes and changes their encryption scheme to Paillier, resulting in $[\![y_{ij}]\!]_C$ for $i \in \{1, \ldots, n\}, j \in \{1, \ldots, k\}$
    $S$: sums the bits for each class separately, obtaining $[\![\sum_{i=1}^n y_{ij}]\!]_C = \sum_{i=1}^n [\![y_{ij}]\!]_C$ for every $j \in \{1, \ldots, k\}$, effectively computing the vote counts of each class
    $S$ and $C$ change the ownership of the vote counts, so that $C$ obtains $[\![\sum_{i=1}^n y_{ij}]\!]_S$ using our protocol
    $C$ and $S$ perform the $\arg\max$ protocol, so that $C$ learns only the outcome of the plurality-vote class
end if
```

where $y_{ij} = 1$ if the outcome of the ith tree was class j and $y_{ij} = 0$ otherwise. This encoding allows to sum up all votes for each class (or vote count), so that the server obtains $[\![\sum_{i=1}^n y_{ij}]\!]_C$ using Paillier's homomorphic property.

However, we cannot directly apply the $\arg\max$ protocol as this would reveal the classification result to the party holding the ciphertexts, i.e., the server. Hence, we leverage our encryption ownership protocol to transfer the vote counts to the client under the server's key. The client thus has $[\![\sum_{i=1}^n y_{ij}]\!]_S$, which allows him to determine the plurality-vote class by applying the $\arg\max$ protocol.

The complete protocol is provided in Protocol 2.

Theorem 2. *Protocol 2 is secure in the honest-but-curious model.*

We refer to the appendix for the proof.

VIII. Evaluation of the Private Classifier

Now that we have introduced our protocol for private classification on random forests, we will evaluate its performance on a dataset and classifier used in practice. More specifically, we base our performance evaluation on MethPed [34], [42], a random forest classifier for the identification of pediatric brain tumor subtypes based on DNA methylation data, which is available as an R package. From this package, we extract their random forest model and feed it into our protocol implementation for the performance evaluation.[2]

MethPed, in its standard configuration, trains a random forest model of 1000 trees based on its original training data, consisting of 472 clinically diagnosed brain tumor cases after data cleaning and k-nearest neighbor imputation of missing values [42]. The DNA methylation samples have been collected from several datasets, all of which are publicly available on the GEO database (GEO accession numbers GSE50022, GSE55712, GSE36278, GSE52556, GSE54880, GSE45353 and GSE44684). The random forest is then trained on a total of 900 methylation sites, which were shown to yield the highest predictive power in a large number of regression analyses.

Our protocol implementation is based on the original implementation of the work of Bost et al.[3]. We extended it by implementing the protocol for changing the encryption scheme from FHE to Paillier, as well as by adding our own protocol for changing the ownership of the encryption. Moreover, we fully implemented the random forest classification protocol (Protocol 2) and tested its correctness on sample inputs. Then, we ported the MethPed classifier into our implementation and included two methylation samples to evaluate the classifier on. The implementation of our private random forest classifier is written in C++ using GMP[4], Boost, Google's Protocol Buffers[5], and HELib [39]. The source code of our implementation can be found at https://github.com/paberr/ciphermed-forests.

In order to represent the methylation levels as integers in our protocol, we multiply them by 10^8 and store the result as an integer. Since the data we used is available at a precision of eight digits after the decimal point and methylation values are bounded by the range $[0, 1]$, we do not lose any precision.

A. Evaluation Setup

To evaluate the performance of our protocol, we ran the client and server of the classification task on different machines, both on the same network and on different networks. One client was run on a local computing server with approximately 775 GB RAM and four Intel Xeon E5-4650L processors, providing 64 cores (with hyperthreading enabled) running at 2.60 GHz. Another client was run on an Amazon AWS instance of the type `r4.2xlarge` with 61 GB RAM and 8 Intel Xeon E5-2686 v4 vCPUs and a network bandwidth up to 10 gigabit located in Frankfurt, Germany. The server was run on a local computing server with approximately 1.55 TB RAM and four Intel Xeon E7-8867 processors,

[2] The R implementation and the used methylation sites are available at http://bioconductor.org/packages/devel/bioc/html/MethPed.html.

[3] Available at https://github.com/rbost/ciphermed.

[4] https://gmplib.org

[5] https://code.google.com/p/protobuf/

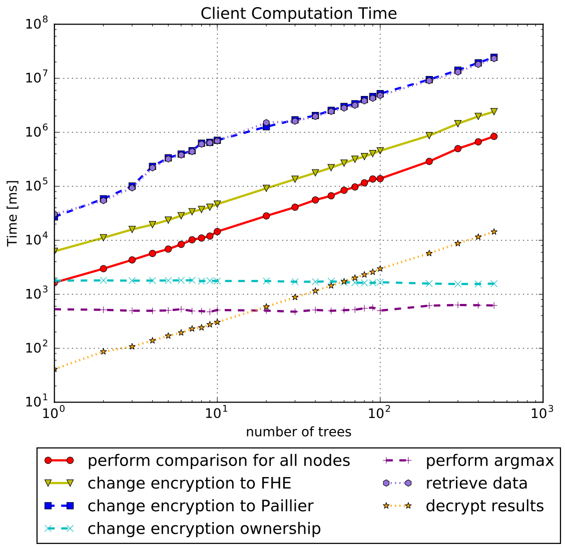

Fig. 9. Duration of different protocol steps on the client side for varying number of trees and both protocol variations.

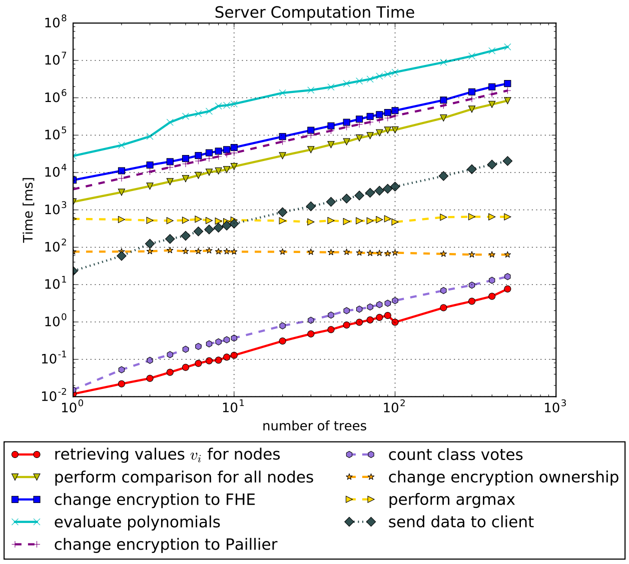

Fig. 10. Duration of different protocol steps on the server side for varying number of trees and both protocol variations.

providing 128 cores (with hyperthreading enabled) running at 2.50 GHz. Since our implementation does not make use of any multithreading technique, we used the large number of cores to run multiple experiments, i.e., classification tasks, at once.

Similar to Bost et al., we also used 1024-bit cryptographic keys and chose the statistical security parameter λ to be 100. HELib was configured to use 80 bits of security, roughly corresponding to a 1024-bit asymmetric key [23].

B. Performance Evaluation

We evaluate our protocol for a varying number of trees $n \in \{1, 2, \ldots, 9, 10, 20, \ldots, 90, 100, 200, \ldots, 400, 500\}$ and two independent classification queries provided in the MethPed R package [42]. We restricted the number of trees to a maximum of 500 in order to keep the computational costs low. We can still estimate the cost of running our protocol with 1000 trees by the general trend as seen in the following. Moreover, we evaluate both versions of our protocol, the first revealing only the plurality-vote class to the client, and the second revealing one outcome per tree to the client. For $n \leq 100$, we classify each of the samples five times, resulting in a total of 10 executions for each of our protocol instantiations. For $n > 100$, we classify each of the samples only once, due to the increased computational costs. The trees used for the classification consist of between 16 and 37 inner nodes, with an average of around 25 inner nodes.

In the following figures, a solid line is used for operations common to both our protocol instantiations, a dashed line is used for the instantiation returning the plurality-vote class, and a dotted line is used for the one outputting the outcome for each tree. The performance evaluation of common operations groups together the results of both instantiations, yielding 20 executions if $n \leq 100$, and 4 executions if $n > 100$.

Fig. 9 depicts the performance evaluation on the client side, both axes scaled logarithmically. Generally, the computational costs of most of our protocol steps scale approximately linearly in the number of trees. Only changing the ownership of the encryption and performing the $\arg\max$ seem to have a constant execution time. These two blocks scale linearly with the number of class labels, which are fixed (to the 9 types of brain tumors) in our experiments.

Next, we compare the execution time of both protocol instantiations. We see that both, helping to change the encryption scheme of the trees' outcomes from FHE to Paillier and retrieving all the tree's outcomes in the FHE cryptosystem, unexpectedly take almost the same amount of time, since essentially the same operations are required. Performing the plurality vote protocol then only adds a constant computational burden on the client's side, only negligibly increasing the total computation time.

In Fig. 10, we analyze the same scenarios on the server side. Unsurprisingly, the relationships between the number of trees in the random forest and the computational costs are the same as for the client. It is worth noting that the computationally most expensive operation is by far the FHE evaluation of the polynomials. Evaluating the polynomials takes almost an order of magnitude more time than the second most expensive protocol step. Thus, minimizing the number of trees and potentially also the number of inner nodes is a main concern when applying our protocol. Moreover, parallelizing the evaluation of the polynomials is a possible improvement, which we did not explore in our implementation.

In terms of the amount of exchanged data and the number of interactions, both protocol instantiations seem to be more or less equivalent as shown in Fig. 11. Revealing the individual outcomes to the client is not noticeably different from performing the plurality vote protocol. While time is mostly the major concern when running a classification task, the amount of data exchanged over the network should not be underestimated. For

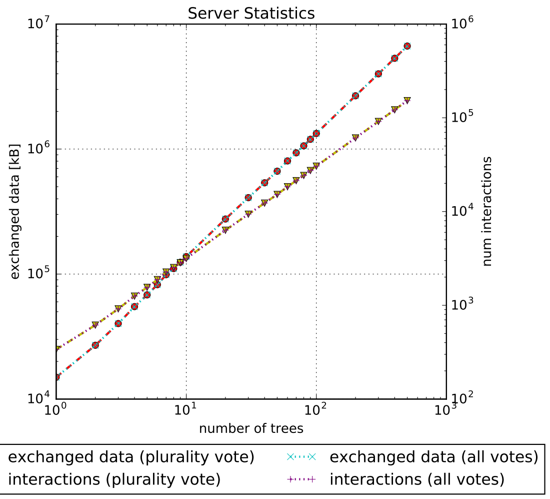

Fig. 11. Data exchange and number of interactions for varying number of trees and both protocol variations.

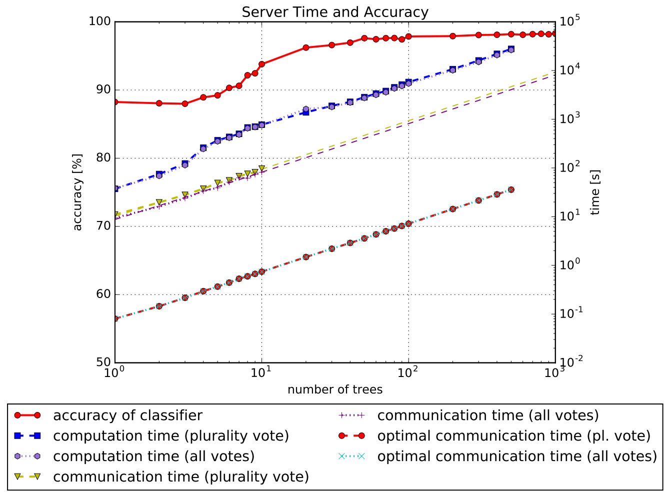

Fig. 12. Total duration of a classification task and accuracy of the random forest for varying number of trees and both protocol variations.

example, evaluating 50 trees involves exchanging around 0.67 GB of data over the network. Increasing the number of trees to 100, involves around 1.33 GB of data exchange.

Finally, in Fig. 12, we study the total time to run the protocol on the server side (excluding the time for sending packets over the network) in comparison with the accuracy of the random forest built on the given number of trees. The accuracy was determined based on the out-of-bag samples during the training phase and averaged over 10 different runs. Since our private classification uses the same precision for the methylation values as the R implementation and builds on exactly the same trees, the accuracy provided by our private classification technique is the same. While the computational costs clearly increase approximately linearly in the number of trees, the accuracy does not. While 1000 trees provide an accuracy of 98.3%, 50 trees are already sufficient to provide an accuracy of 97.6% at only an estimated 5% of the computational cost. We also depict the communication time between our Amazon AWS instance and the local computing server for a smaller range of number of trees. Evaluating 50 trees takes in total less than a hour, even when including the time for sending and receiving packets over the internet. We also evaluated the timing on the client's side, which exhibits the same behaviour as on the server's side.

We emphasise that our current implementation does neither aim at minimizing the number of interactions, nor does it make use of pipelining of interactions. Based on the measured throughput between the Amazon AWS instance and our computing server, we additionally depict the estimated optimal communication time over the network in Fig. 12. Improving the transmission of data in setup can potentially decrease the communication time for 500 trees down to 50 seconds.

Since, in the current medical scenario, it usually takes at least one day for a laboratory to analyze a sample, we assume a similar computational limit on the classification. Given such a limit, we conclude that a laboratory offering the privacy preserving analysis using our protocol would be able to provide a good trade-off between computational costs and accuracy. Moreover, the structure of random forests offers a great potential to parallelize some of the operations (e.g., the polynomial evaluation), which we leave for future work.

We note that both protocol instantiations take approximately the same time to run. While returning the selected classs for a number of 50 trees is about 2 minutes faster than returning the majority vote, this difference only accounts to about 6 minutes for 100 trees and to about 23 minutes for 500 trees. Hence, we suggest to select the instantiation based on the output the client needs and the classifier information the server agrees to reveal. If the client wants a fine-grained output to analyze the distribution of the different classes, then he may request to get access not to the plurality-vote class, but to the selected class of each tree. However, this will leak more information about the underlying random forest model than disclosing only the plurality-vote class.

IX. Related Work

We first summarize the two most closely related papers, which report about the risk of identification of DNA methylation data. The first (short) paper studying this risk shows that part of the genotype (around 1,000 positions), as well as alcohol consumption and smoking, can be inferred from certain methylation data [43]. They warn that such genotype inference could represent personally identifying information but do not study further how genotypes could be matched to methylation profiles, neither do they quantify with what success such an attack could be carried out, and under which conditions. Besides also identifying CpGs correlated with genomic variants, Dyke et al. propose high-level guidelines for methylation data disclosure that preserves privacy [9]. They notably mention the restriction of access to methylation data that are highly correlated with the genotype. Again, a concrete scenario is missing in order to evaluate the extent of the threat as well as the protection provided by their countermeasure.

In a similar vein as our approach, Schadt et al. propose a Bayesian method to predict from and match genotypes to RNA-expression profiles [44]. By using 1,000 eQTLs (expression quantitative trait locus), they were able to correctly match RNA expression profiles and genotypes of more than 300 individuals. Furthermore, they simulated dataset of 300 million individuals and showed that the matching accuracy was still as high as 97%. Franzosa et al. study whether individuals possess microbial patterns that could be used to uniquely identify them [45]. Their results demonstrate that more than 80% of individuals can still be uniquely identified among a population of hundreds of individuals, up to one year later in the case of the gut microbiome. Backes et al. also study how microRNA expression profiles can be tracked over time [46]. They demonstrate that such data can be linked with a success rate of 90%, and that success rates remain constant up to one-year time shift between two profiles. They further propose two countermeasures: one based on hiding part of microRNA expressions, and the other based on probabilistic sanitization of the microRNA expression profiles. Backes et al. further show that microRNA-based datasets are prone to membership inference attacks by relying on the average statistics of their microRNA expression values [47].

Gymrek et al. show that genotypes can be re-identified by querying genetic genealogy databases (containing surnames) with short tandem repeats on the Y chromosome [21]. By combining the inferred surnames with other types of metadata, such as age and state, they are able to trace back with high success the identitites of multiple contributors in public databases. Humbert et al. show that single nucleotide polymorphisms (SNPs), which are more commonly available online, can be also exploited to infer various phenotypic traits, such as eye color or blood type, in order to further re-identify anonymous genotypes, by typically using side channels such as online social networks [22]. Both these works clearly illustrate that, once the genotype corresponding to a DNA methylation profile has been identified, it becomes relatively simple to recover the real identity of the owner of this methylation profile.

Finally, there have been several works on privacy-preserving disease prediction by relying on encrypted genomic data. Bost et al. develop three main private classification protocols (including decision trees) that protect both the patients' data and the classifier model [23]. They prove their protocols to be secure in the honest-but-curious adversarial model, and evaluate its performance on real medical datasets. We build upon their constructions for our own private random forest classifier. Duverle et al. propose a new protocol that enables to privately compute statistical tests on patients' data by relying on exact logistic regression [48]. Their performance evaluation shows that they can perform statistical tests with more than 600 SNPs across thousands of patients in several hours.

Ayday et al. have developed schemes for private disease susceptibility tests by using homomorphic encryption and proxy-encryption [24], [49]. The considered tests are based on linear combinations of the SNPs (and other environmental and clinial factors in [49]) contributing to a given disease, and do not involve complex machine-learning classifiers. Danezis and De Cristofaro improve upon the protocol of [24] by using an alternative SNP encoding and make the patient-side computation more efficient [26]. McLaren et al. use a similar security architecture as the one initially proposed by Ayday et al. to develop a practical privacy-preserving scheme of genome-based prediction of HIV-related outcomes [25]. All these papers assume an honest-but-curious adversary, which is considered as realistic in the healthcare environment.

X. Conclusion

In this work, we have first demonstrated that DNA methylation datasets can be re-identified by having access to an auxiliary database of genotypes. Following a Bayesian approach, we have shown that we could reach an accuracy of 97.5% to 100% depending on the attack scenario, with a few hundreds of methylation regions and genotype positions. Then, by using a statistical test upon our matching outcomes, we have empirically demonstrated that the very few wrongly matched pairs could be correctly identified and rejected, yielding a false-positive rate of 0 and true-positive rate of 1 for appropriate statistical thresholds. We have further shown that our identification attack was very robust to a decrease of methylation-meQTL pairs. When matching 52 methylation profiles with 75 genotypes, we could reach a full accuracy with only 13 meQTLs and methylation regions. We have also observed that, especially when the targeted genotype is present in the genotype dataset, the very few wrongly matched pairs contain the genotype of the relative (in more than 90% of the cases). Finally, we have shown that our attack was robust to an increase of the database size to more than 2500 genotypes.

Facing this severe threat to epigenetic privacy, we have proposed a novel cryptographic scheme for privately classifying tumors based on methylation data. Our protocol relies on random forests and homomorphic encryption, and it is proven secure in the honest-but-curious adversarial model. We have implemented our private classifier in C++ and evaluated its performance on real data. We have shown that it can accurately classify brain tumors in nine classes of tumor subtypes based on 900 methylation levels in less than an hour. This constitutes an acceptable computational overhead in the considered clinical setting at hand. As a meta-consequence, we highly recommend to remove DNA methylation profiles from public databases as these are extremely prone to re-identification, especially given that genotypes are also increasingly available online, sometimes with their owners' identifiers [11].

As future work, we plan to study if the identification attack is as successful when meQTL-methylation pairs are learned from a different tissue's data. At the defense side, we would like to study other machine-learning algorithms, and to propose private schemes for those that are efficient in classification with methylation data. Differentially private approaches could also be studied, although differential privacy may degrade utility too much for typical medical needs [50].

ACKNOWLEDGEMENTS

We would like to thank Jonas Schneider for his helpful feedback on the security proofs. This work was supported by the German Federal Ministry of Education and Research (BMBF) through funding for the Center for IT-Security, Privacy and Accountability (CISPA) (FKZ: 16KIS0656) and by the German Research Foundation (DFG) via the collaborative research center Methods and Tools for Understanding and Controlling Privacy (SFB 1223), project A5.

REFERENCES

[1] "President obama's precision medicine initiative," https://www.whitehouse.gov/the-press-office/2015/01/30/fact-sheet-president-obama-s-precision-medicine-initiative, accessed: 2016-10-14.

[2] Y. Erlich and A. Narayanan, "Routes for breaching and protecting genetic privacy," *Nature Reviews Genetics*, vol. 15, pp. 409–421, 2014.

[3] M. Humbert, E. Ayday, J.-P. Hubaux, and A. Telenti, "Addressing the concerns of the Lacks family: quantification of kin genomic privacy," in *Proceedings of the 2013 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2013, pp. 1141–1152.

[4] "The growth of the gene expression omnibus," https://plot.ly/~sambucas/34/the-growth-of-the-gene-expression-omnibus/, accessed: 2016-10-14.

[5] M. Esteller and J. G. Herman, "Cancer as an epigenetic disease: DNA methylation and chromatin alterations in human tumours," *The Journal of Pathology*, vol. 196, no. 1, pp. 1–7, jan 2002.

[6] P. M. Das and R. Singal, "DNA methylation and cancer," *Journal of clinical oncology*, vol. 22, no. 22, pp. 4632–4642, 2004.

[7] B. Vogelstein, N. Papadopoulos, V. E. Velculescu, S. Zhou, L. A. Diaz, and K. W. Kinzler, "Cancer genome landscapes," *Science*, vol. 339, no. 6127, pp. 1546–1558, 2013.

[8] M. A. Rothstein, Y. Cai, and G. E. Marchant, "The ghost in our genes: legal and ethical implications of epigenetics," *Health matrix (Cleveland, Ohio: 1991)*, vol. 19, p. 1, 2009.

[9] S. O. Dyke, W. A. Cheung, Y. Joly, O. Ammerpohl, P. Lutsik, M. A. Rothstein, M. Caron, S. Busche, G. Bourque, L. Rönnblom *et al.*, "Epigenome data release: a participant-centered approach to privacy protection," *Genome biology*, vol. 16, pp. 1–12, 2015.

[10] J. L. McClay, A. A. Shabalin, M. G. Dozmorov, D. E. Adkins, G. Kumar, S. Nerella, S. L. Clark, S. E. Bergen, C. M. Hultman, P. K. E. Magnusson, P. F. Sullivan, K. A. Aberg, and E. J. C. G. van den Oord, "High density methylation QTL analysis in human blood via next-generation sequencing of the methylated genomic DNA fraction," *Genome Biology*, vol. 16, no. 1, p. 291, 2015.

[11] "openSNP," https://opensnp.org, accessed: 2016-11-05.

[12] P. a. Jones, "Functions of DNA methylation: islands, start sites, gene bodies and beyond." *Nature reviews. Genetics*, vol. 13, no. 7, pp. 484–92, jul 2012.

[13] D. Schübeler, "Function and information content of DNA methylation," *Nature*, vol. 517, no. 7534, pp. 321–326, jan 2015.

[14] K. L. Sheaffer, R. Kim, R. Aoki, E. N. Elliott, J. Schug, L. Burger, D. Schubeler, and K. H. Kaestner, "DNA methylation is required for the control of stem cell differentiation in the small intestine," *Genes & Development*, vol. 28, no. 6, pp. 652–664, mar 2014.

[15] T. Bauer, S. Trump, N. Ishaque, L. Thu rmann, L. Gu, M. Bauer, M. Bieg, Z. Gu, D. Weichenhan, J.-P. Mallm, S. Ro der, G. Herberth, E. Takada, O. Mu cke, M. Winter, K. M. Junge, K. Gru tzmann, U. Rolle-Kampczyk, Q. Wang, C. Lawerenz, M. Borte, T. Polte, M. Schlesner, M. Schanne, S. Wiemann, C. Geo rg, H. G. Stunnenberg, C. Plass, K. Rippe, J. Mizuguchi, C. Herrmann, R. Eils, and I. Lehmann, "Environment-induced epigenetic reprogramming in genomic regulatory elements in smoking mothers and their children," *Molecular Systems Biology*, vol. 12, no. 3, pp. 861–861, mar 2016.

[16] S. Trump, M. Bieg, Z. Gu, L. Thürmann, T. Bauer, M. Bauer, N. Ishaque, S. Röder, L. Gu, G. Herberth, C. Lawerenz, M. Borte, M. Schlesner, C. Plass, N. Diessl, M. Eszlinger, O. Mücke, H.-D. Elvers, D. K. Wissenbach, M. von Bergen, C. Herrmann, D. Weichenhan, R. J. Wright, I. Lehmann, and R. Eils, "Prenatal maternal stress and wheeze in children: novel insights into epigenetic regulation," *Scientific Reports*, vol. 6, p. 28616, jun 2016.

[17] J. van Dongen, M. G. Nivard, G. Willemsen, J.-J. Hottenga, Q. Helmer, C. V. Dolan, E. A. Ehli, G. E. Davies, M. van Iterson, C. E. Breeze, S. Beck, P. A. Hoen, R. Pool, M. M. van Greevenbroek, C. D. Stehouwer, C. J. van der Kallen, C. G. Schalkwijk, C. Wijmenga, S. Zhernakova, E. F. Tigchelaar, M. Beekman, J. Deelen, D. van Heemst, J. H. Veldink, L. H. van den Berg, C. M. van Duijn, B. A. Hofman, A. G. Uitterlinden, P. M. Jhamai, M. Verbiest, M. Verkerk, R. van der Breggen, J. van Rooij, N. Lakenberg, H. Mei, J. Bot, D. V. Zhernakova, P. van't Hof, P. Deelen, I. Nooren, M. Moed, M. Vermaat, R. Luijk, M. J. Bonder, F. van Dijk, M. van Galen, W. Arindrarto, S. M. Kielbasa, M. A. Swertz, E. W. van Zwet, A. Isaacs, L. Franke, H. E. Suchiman, R. Jansen, J. B. van Meurs, B. T. Heijmans, P. E. Slagboom, and D. I. Boomsma, "Genetic and environmental influences interact with age and sex in shaping the human methylome," *Nature Communications*, vol. 7, p. 11115, 2016.

[18] L. G. Tsaprouni, T.-P. Yang, J. Bell, K. J. Dick, S. Kanoni, J. Nisbet, A. Viñuela, E. Grundberg, C. P. Nelson, E. Meduri, A. Buil, F. Cambien, C. Hengstenberg, J. Erdmann, H. Schunkert, A. H. Goodall, W. H. Ouwehand, E. Dermitzakis, T. D. Spector, N. J. Samani, and P. Deloukas, "Cigarette smoking reduces DNA methylation levels at multiple genomic loci but the effect is partially reversible upon cessation," *Epigenetics*, no. December, pp. 00–00, oct 2014.

[19] A. L. Teh, H. Pan, L. Chen, M. L. Ong, S. Dogra, J. Wong, J. L. MacIsaac, S. M. Mah, L. M. McEwen, S. M. Saw, K. M. Godfrey, Y. S. Chong, K. Kwek, C. K. Kwoh, S. E. Soh, M. F. F. Chong, S. Barton, N. Karnani, C. Y. Cheong, J. P. Buschdorf, W. St??nkel, M. S. Kobor, M. J. Meaney, P. D. Gluckman, and J. D. Holbrook, "The effect of genotype and in utero environment on interindividual variation in neonate DNA methylomes," *Genome Research*, vol. 24, no. 7, pp. 1064–1074, 2014.

[20] T. R. Gaunt, H. A. Shihab, G. Hemani, J. L. Min, G. Woodward, O. Lyttleton, J. Zheng, A. Duggirala, W. L. McArdle, K. Ho, S. M. Ring, D. M. Evans, G. Davey Smith, and C. L. Relton, "Systematic identification of genetic influences on methylation across the human life course," *Genome Biology*, vol. 17, no. 1, p. 61, 2016.

[21] M. Gymrek, A. L. McGuire, D. Golan, E. Halperin, and Y. Erlich, "Identifying personal genomes by surname inference," *Science*, vol. 339, pp. 321–324, 2013.

[22] M. Humbert, K. Huguenin, J. Hugonot, E. Ayday, and J.-P. Hubaux, "De-anonymizing genomic databases using phenotypic traits," *Proceedings on Privacy Enhancing Technologies (PoPETs)*, 2015.

[23] R. Bost, R. A. Popa, S. Tu, and S. Goldwasser, "Machine learning classification over encrypted data," in *22nd Network and Distributed System Security Symposium (NDSS' 15)*, 2015.

[24] E. Ayday, J. L. Raisaro, J.-P. Hubaux, and J. Rougemont, "Protecting and evaluating genomic privacy in medical tests and personalized medicine," in *Proceedings of the 12th ACM workshop on Workshop on privacy in the electronic society*. ACM, 2013, pp. 95–106.

[25] P. J. McLaren, J. L. Raisaro, M. Aouri, M. Rotger, E. Ayday, I. Bartha, M. B. Delgado, Y. Vallet, H. F. Günthard, M. Cavassini *et al.*, "Privacy-preserving genomic testing in the clinic: a model using HIV treatment," *Genetics in Medicine*, 2016.

[26] G. Danezis and E. De Cristofaro, "Fast and private genomic testing for disease susceptibility," in *Proceedings of the 13th Workshop on Privacy in the Electronic Society*. ACM, 2014, pp. 31–34.

[27] J. Edmonds, "Paths, trees, and flowers," *Canadian Journal of Mathematics*, vol. 17, no. 3, pp. 449–467, 1965.

[28] Y. Liu, K. D. Siegmund, P. W. Laird, and B. P. Berman, "Bis-SNP: Combined DNA methylation and SNP calling for Bisulfite-seq data," *Genome Biology*, vol. 13, no. 7, p. R61, 2012. [Online]. Available: http://genomebiology.com/2012/13/7/R61

[29] C. Spearman, "The proof and measurement of association between two things," *The American journal of psychology*, vol. 15, no. 1, pp. 72–101, 1904.

[30] Y. Benjamini and Y. Hochberg, "Controlling the false discovery rate: a practical and powerful approach to multiple testing," *Journal of the royal statistical society. Series B (Methodological)*, pp. 289–300, 1995.

[31] "Igsr: The international genome sample resource," http://www.internationalgenome.org/data, accessed: 2016-11-05.

[32] T. K. Ho, "Random decision forests," in *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*, vol. 1. IEEE, 1995, pp. 278–282.

[33] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*. Springer series in statistics Springer, Berlin, 2001, vol. 1.

[34] A. Danielsson, S. Nemes, M. Tisell, B. Lannering, C. Nordborg, M. Sabel, and H. Carén, "MethPed: a DNA methylation classifier tool for the identification of pediatric brain tumor subtypes," *Clinical epigenetics*, vol. 7, no. 1, p. 1, 2015.

[35] "Sophia Genetics," http://www.sophiagenetics.com/, accessed: 2016-11-05.

[36] S. Goldwasser and S. Micali, "Probabilistic encryption and how to play mental poker keeping secret all private information," in *Proceedings 14th ACM Symposium on the Theory of Computing*, 1982.

[37] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 1999, pp. 223–238.

[38] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "Fully homomorphic encryption without bootstrapping," Cryptology ePrint Archive, Report 2011/277, 2011, http://eprint.iacr.org/2011/277.

[39] S. Halevi and V. Shoup, "HElib – an implementation of homomorphic encryption," 2014, https://github.com/shaih/HElib.

[40] O. Goldreich, "Foundations of cryptography. basic applications," 2004.

[41] N. P. Smart and F. Vercauteren, "Fully homomorphic SIMD operations," *Designs, codes and cryptography*, vol. 71, no. 1, pp. 57–81, 2014.

[42] M. T. Ahamed, A. Danielsson, S. Nemes, and H. Carén, "MethPed: an R package for the identification of pediatric brain tumor subtypes," *BMC bioinformatics*, vol. 17, no. 1, p. 262, 2016.

[43] R. A. Philibert, N. Terry, C. Erwin, W. J. Philibert, S. R. Beach, and G. H. Brody, "Methylation array data can simultaneously identify individuals and convey protected health information: an unrecognized ethical concern," *Clinical epigenetics*, vol. 6, p. 28, 2014.

[44] E. E. Schadt, S. Woo, and K. Hao, "Bayesian method to predict individual SNP genotypes from gene expression data," *Nature genetics*, vol. 44, pp. 603–608, 2012.

[45] E. A. Franzosa, K. Huang, J. F. Meadow, D. Gevers, K. P. Lemon, B. J. Bohannan, and C. Huttenhower, "Identifying personal microbiomes using metagenomic codes," *Proceedings of the National Academy of Sciences*, p. 201423854, 2015.

[46] M. Backes, P. Berrang, A. Hecksteden, M. Humbert, A. Keller, and T. Meyer, "Privacy in epigenetics: Temporal linkability of MicroRNA expression profiles," in *Proceedings of the 25th USENIX Security Symposium*, 2016.

[47] M. Backes, P. Berrang, M. Humbert, and P. Manoharan, "Membership privacy in MicroRNA-based studies," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2016, pp. 319–330.

[48] D. A. Duverle, S. Kawasaki, Y. Yamada, J. Sakuma, and K. Tsuda, "Privacy-preserving statistical analysis by exact logistic regression," in *Security and Privacy Workshops (SPW), 2015 IEEE*. IEEE, 2015, pp. 7–16.

[49] E. Ayday, J. L. Raisaro, P. J. McLaren, J. Fellay, and J.-P. Hubaux, "Privacy-preserving computation of disease risk by using genomic, clinical, and environmental data," in *Presented as part of the 2013 USENIX Workshop on Health Information Technologies*, 2013.

[50] M. Fredrikson, E. Lantz, S. Jha, S. Lin, D. Page, and T. Ristenpart, "Privacy in pharmacogenetics: An end-to-end case study of personalized warfarin dosing," in *Proceedings of the 23rd USENIX Security Symposium*, 2014, pp. 17–32.

[51] R. Canetti, "Security and composition of multiparty cryptographic protocols," *Journal of CRYPTOLOGY*, vol. 13, no. 1, pp. 143–202, 2000.

[52] Y. Lindell and B. Pinkas, "Secure multiparty computation for privacy-preserving data mining," *IACR Cryptology ePrint Archive*, vol. 2008, p. 197, 2008. [Online]. Available: http://eprint.iacr.org/2008/197

Appendix

Although we assume the same security model as in the work by Bost et al. [23], we recap here the necessary concepts.

A. Secure two-party computation framework

Both, our protocol to change the ownership of an encryption and the protocol to privately evaluate a random forest model are two-party protocols. Let the two parties be denoted by A and B. In order to show that all computations are done privately, we assume the honest-but-curious (semi-honest) model as described in [40].

Let $f = (f_A, f_B)$ be a (probabilistic) polynomial function and Π be a protocol computing f. Using A's input a and B's input b, the two parties want to compute $f(a, b)$ by applying the protocol Π with the security parameter λ.

We denote the view of a party $P \in \{A, B\}$ during the execution of Π by the tuple $V_P(\lambda, a, b) = (1^\lambda; a; r^P; m_1^P, \ldots, m_t^P)$ where r is P's random tape and $m_1^P, \ldots, m_t^P$ are the messages received by P. We define the outputs of parties A and B for the execution of Π as $\mathsf{Out}_A^\Pi(\lambda, a, b)$ and $\mathsf{Out}_B^\Pi(\lambda, a, b)$. The global output is defined as the tuple $\mathsf{Out}^\Pi(\lambda, a, b) = (\mathsf{Out}_A^\Pi(\lambda, a, b), \mathsf{Out}_B^\Pi(\lambda, a, b))$.

To ensure the private, secure computation, we require that whatever A can compute from its interactions with B can be computed from its input and output, yielding the following security definition.

Definition 1. *A two-party protocol Π securely computes the function f if there exist two probabilistic polynomial time algorithms S_A and S_B (also called simulators) such that for every possible input a, b of f,*

$$\{S_A(1^\lambda, a, f_A(a, b)), f(a, b)\} \equiv_c \{V_A(\lambda, a, b), \mathit{Out}^\Pi(\lambda, a, b)\}$$

and

$$\{S_B(1^\lambda, b, f_B(a, b)), f(a, b)\} \equiv_c \{V_B(\lambda, a, b), \mathit{Out}^\Pi(\lambda, a, b)\}.$$

$\equiv_c$ means computational indistinguishability against probabilistic polynomial time adversaries with negligible advantage in the security parameter λ.

B. Cryptographic assumptions

In this section, we briefly review the cryptographic assumptions underlying the cryptosystems we use.

Assumption 1 (Quadratic Residuosity Assumption [36]). *Let $N = p \times q$ be the product of two distinct odd primes p and q. Let $\mathbb{QR}_N$ be the set of quadratic residues modulo N and $\mathbb{QNR}_N = \{x \in \mathbb{Z}_N^* \mid x \text{ is not a quadratic residue modulo } N, \text{ but } \mathcal{J}_N(x) = +1\}$ be the set of quadratic non residues, where $\mathcal{J}_N(x)$ is the Jacobi symbol.*

$\{(N, \mathbb{QR}_N) \mid |N| = \lambda\}$ and $\{(N, \mathbb{QNR}_N) \mid |N| = \lambda\}$ are computationally indistinguishable with respect to probabilistic polynomial time algorithms.

Assumption 2 (Decisional Composite Residuosity Assumption [37]). *Let $N = p \times q$ with $|N| = \lambda$ be the product of two distinct odd primes p and q. We call z a Nth residue modulo N^2 if there exists $y \in \mathbb{Z}_{N^2}$ such that $z = y^N \mod N^2$. Nth residues and non Nth residues are computationally indistinguishable with respect to probabilistic polynomial time algorithms.*

Assumption 3 (RLWE [38]). *Let $f(x) = x^d + 1$ where $d = d(\lambda)$ is a power of 2. Let $q = q(\lambda) \geq 2$ be an integer. Let $R = Z[x]/(f(x))$ and let $R_q = R/qR$. Let $\chi = \chi(\lambda)$ be a distribution over R. The $RLWE_{d,q,\chi}$ problem is to distinguish between two distributions: In the first distribution, one samples (a_i, b_i) uniformly from R_q^2. In the second distribution, one first* *draws $s \leftarrow R_q$ uniformly and then samples $(a_i, b_i) \in R_q^2$ by sampling $a_i \leftarrow R_q$ uniformly, $e_i \leftarrow \chi$, and setting $b_i = a_i \cdot s + e_i$. The $RLWE_{d,q,\chi}$ assumption is that the $RLWE_{d,q,\chi}$ problem is infeasible.*

C. Modular Sequential Composition

In order to ease the security proof of our construction, we rely on sequential modular composition as defined in [51]. The idea is that two parties run a protocol Π and use calls to an ideal functionality f while running Π. This can be imagined as A and B privately computing f by sending their inputs to a trusted third party T and receiving the results from it. If we can now show that Π respects security and privacy in the honest-but-curious model and if we have a protocol ρ that securely and privately computes f in the same model, we can replace f by executions of ρ in Π. The resulting protocol Π^ρ is then still secure in the aforementioned model.

We call $(f_1, \ldots, f_m)$*-hybrid model* the semi-honest model augmented with an incorruptible trusted party T for evaluating the functionalities. The parties A and B run a protocol Π that contains calls to T for these functionalities. For each call, the parties send their input to T and wait until they receive the respective results. It is crucial that both parties must not communicate until receiving the result, since we only consider sequential composition here. T does not keep state between different calls to the functionalities. Therefore the protocol may contain multiple calls even for the same function, which all are independent.

Let Π be a two-party protocol in the $(f_1, \ldots, f_m)$-hybrid model and $\rho_1, \ldots, \rho_m$ be secure protocols in the semi-honest model computing $f_1, \ldots, f_m$. We define $\Pi^{\{\rho_1, \ldots, \rho_m\}}$ as the protocol where all ideal calls of Π have been replaced by executions of the corresponding protocol: if party P_j needs to compute f_i with input x_j, it halts, starts an execution of ρ_i with the other party, gets the result β_j from ρ_i and continues as if β_j was received from T.

Theorem 3 (Modular Sequential Composition Theorem [51], [52])**.** *Let $f_1, \ldots, f_m$ be two-party probabilistic polynomial time functionalities and $\rho_1, \ldots, \rho_m$ be protocols that compute respectively $f_1, \ldots, f_m$ in the presence of semi-honest adversaries.*

Let g be a two-party probabilistic polynomial time functionality and Π a protocol that securely computes g in the $(f_1, \ldots, f_m)$-hybrid model in the presence of semi-honest adversaries.

Then $\Pi^{\rho_1, \ldots, \rho_m}$ securely computes g in the presence of semi-honest adversaries.

D. Changing Encryption Owner

Proof of Theorem 1. The function f this protocol computes is:

$$f((\llbracket x \rrbracket_B, \mathsf{SK}_A, \mathsf{PK}_B), (\mathsf{PK}_A, \mathsf{SK}_B)) = (\emptyset, \llbracket x \rrbracket_A)$$

For the sake of simplicity, we do not take into account the randomness used for the encryptions of r for A and $c' = x + r$ for B. The distribution of these coins for one party is completely independent of the other elements taken into account in the simulations, so we omit them in our security proof.

A's view is $V_A = (\mathsf{SK}_A, \mathsf{PK}_B, \llbracket x \rrbracket_B; r; \emptyset)$. A does not output anything. The simulator $S_A(\mathsf{SK}_A, \mathsf{PK}_B, \llbracket x \rrbracket_B)$ runs as follows:

1) Picks uniformly at random $\tilde{r} \leftarrow M_P$.
2) Outputs $(\mathsf{SK}_A, \mathsf{PK}_B, \llbracket x \rrbracket_B; \tilde{r}; \emptyset)$

Since r and $\tilde{r}$ are sampled from the same distribution, independently from any other parameter,

$$\{(\mathsf{SK}_A, \mathsf{PK}_B, \llbracket x \rrbracket_B; \tilde{r}; \emptyset), f(\llbracket x \rrbracket_B, \mathsf{SK}_A, \mathsf{PK}_B, \mathsf{PK}_A, \mathsf{SK}_B)\} = \{(\mathsf{SK}_A, \mathsf{PK}_B, \llbracket x \rrbracket_B; r; \emptyset), f(\llbracket x \rrbracket_B, \mathsf{SK}_A, \mathsf{PK}_B, \mathsf{PK}_A, \mathsf{SK}_B)\}.$$

Moreover, it holds that

$$\{(\mathsf{SK}_A, \mathsf{PK}_B, \llbracket x \rrbracket_B; r; \emptyset), f(\llbracket x \rrbracket_B, \mathsf{SK}_A, \mathsf{PK}_B, \mathsf{PK}_A, \mathsf{SK}_B)\} = \{(\mathsf{SK}_A, \mathsf{PK}_B, \llbracket x \rrbracket_B; r; \emptyset), (\emptyset, \llbracket x \rrbracket_A)\}$$

and we can conclude

$$\{S_A(\mathsf{SK}_A, \mathsf{PK}_B, \llbracket x \rrbracket_B), f(\llbracket x \rrbracket_B, \mathsf{SK}_A, \mathsf{PK}_B, \mathsf{PK}_A, \mathsf{SK}_B)\} \equiv_c \{V_A(\llbracket x \rrbracket_B, \mathsf{SK}_A, \mathsf{PK}_B, \mathsf{PK}_A, \mathsf{SK}_B), \mathsf{Out}(\llbracket x \rrbracket_B, \mathsf{SK}_A, \mathsf{PK}_B, \mathsf{PK}_A, \mathsf{SK}_B)\}.$$

B's view is $V_B = (\mathsf{PK}_A, \mathsf{SK}_B; \llbracket x + r \rrbracket_B, \llbracket r \rrbracket_A)$. B outputs $\llbracket x \rrbracket_A$. We build a simulator $S_B(\mathsf{PK}_A, \mathsf{SK}_B)$ as follows:

1) Pick uniformly at random $\tilde{r} \leftarrow M_P$ and $\tilde{c} \leftarrow M_P$.
2) Generate the encryptions $\llbracket \tilde{r} \rrbracket_A$ and $\llbracket \tilde{c} \rrbracket$ using PK_A.
3) Output $(\mathsf{PK}_A, \mathsf{SK}_B; \llbracket \tilde{c} \rrbracket_B, \llbracket \tilde{r} \rrbracket_A)$

By semantic security of the encryption scheme (in our concrete case the Paillier cryptosystem), it holds that (proof see below)

$$\{(\mathsf{PK}_A, \mathsf{SK}_B; \llbracket \tilde{c} \rrbracket_B, \llbracket \tilde{r} \rrbracket_A), f(\llbracket x \rrbracket_B, \mathsf{SK}_A, \mathsf{PK}_B, \mathsf{PK}_A, \mathsf{SK}_B)\} \equiv_c \tag{4}$$

$$\{(\mathsf{PK}_A, \mathsf{SK}_B; \llbracket x + r \rrbracket_B, \llbracket r \rrbracket_A), f(\llbracket x \rrbracket_B, \mathsf{SK}_A, \mathsf{PK}_B, \mathsf{PK}_A, \mathsf{SK}_B)\} \tag{5}$$

and hence (using also the correctness of the scheme)

$$\{S_B(\mathsf{PK}_A, \mathsf{SK}_B), f(\llbracket x \rrbracket_B, \mathsf{SK}_A, \mathsf{PK}_B, \mathsf{PK}_A, \mathsf{SK}_B)\} \equiv_c \{V_B(\llbracket x \rrbracket_B, \mathsf{SK}_A, \mathsf{PK}_B, \mathsf{PK}_A, \mathsf{SK}_B), \mathsf{Out}(\llbracket x \rrbracket_B, \mathsf{SK}_A, \mathsf{PK}_B, \mathsf{PK}_A, \mathsf{SK}_B)\}.$$

We will prove the computational indistinguishability of (4) and (5) in more detail by giving a reduction to the semantic security. To this end, we assume that we have a distinguisher $\mathcal{D}$ that can distinguish (4) and (5). Specifically, given $\{(\mathsf{PK}, \mathsf{SK}', \llbracket y \rrbracket_{\mathsf{SK}'}, \llbracket r \rrbracket_{\mathsf{SK}}), \llbracket x \rrbracket_{\mathsf{SK}}\}$ $\mathcal{D}$ outputs 1 if y, r and x are independent uniformly random values and 0 if $r = y - r'$ for a random r' and $x = y - r = r'$. Then, we construct a reduction $\mathcal{R}$ as follows:

1) On input PK, generate a new key pair $(\mathsf{SK}', \mathsf{PK}') \leftarrow \mathsf{KeyGen}(1^\lambda)$.
2) Pick uniformly at random $y, \tilde{r} \leftarrow M$.
3) Choose challenger messages $m_0 = y - \tilde{r}$, $m_1 = \tilde{r}$ and give them to the semantic security challenger.

4) Receive c from the challenger, compute $\llbracket \tilde{r} \rrbracket_{\mathsf{PK}}$ and query the distinguisher $\mathcal{D}(\{(\mathsf{PK}, \mathsf{SK}', \llbracket y \rrbracket_{\mathsf{SK}'}, c), \llbracket \tilde{r} \rrbracket_{\mathsf{PK}}\})$, which returns b.
5) Return b to the challenger.

Since we simulate both cases ((4) and (5)) perfectly to the distinguisher, its success probability in distinguishing (4) and (5) transfers exactly to our reduction in the semantic security game. Since Paillier encryption is shown to be semantically secure under the Decisional Composite Residuosity Assumption, the distinguisher must have at most negligible success probability. And hence our scheme is secure. □

E. Private Random Forest Evaluation

The correctness of our protocol follows from the correctness of the private classification tree protocol in [23]. Moreover, we will provide a security proof for the protocol revealing only the plurality-vote class. Since our second protocol instantiation – revealing all trees' outcomes – is essentially only a shorter version of the main protocol, we do not provide a separate security proof for this protocol.

Proof of Theorem 2. Let A be the server S and B be the client C. We prove the security of our protocol (see Protocol 2) in the hybrid model using the following 5 ideal functionalities, which we let execute by a trusted third party:

- the comparison protocol in step 3: $f_1(\llbracket x \rrbracket_B, \llbracket y \rrbracket_B, l, \mathsf{SK}_B^{QR}, \mathsf{PK}_B^{QR}, \mathsf{SK}_B^{P}, \mathsf{PK}_B^{P}) = ([x \leq y]_B, \emptyset)$
- the protocol to change the encryption scheme in step 4: $f_2([b]_B, \mathsf{SK}_B^{QR}, \mathsf{PK}_B^{QR}, \mathsf{SK}_B^{FHE}, \mathsf{PK}_B^{FHE}) = ([\![\![b]\!]\!]_B, \emptyset)$
- the protocol to change the encryption scheme in step 10: $f_3([\![\![y_1, \dots, y_k]\!]\!]_B, \mathsf{SK}_B^{FHE}, \mathsf{PK}_B^{FHE}, \mathsf{SK}_B^{P}, \mathsf{PK}_B^{P}) = (\{\llbracket y_1 \rrbracket_B, \dots, \llbracket y_k \rrbracket_C\}_{i=1}^{k}, \emptyset)$
- the protocol to change the ownership of the encryption in step 12: $f_4(\llbracket x \rrbracket_B, \mathsf{SK}_A^{P}, \mathsf{PK}_B^{P}, \mathsf{PK}_A^{P}, \mathsf{SK}_B^{P}) = (\emptyset, \llbracket x \rrbracket_A)$
- the $\arg\max$ protocol in step 13: $f_5(\{\llbracket a_i \rrbracket_A\}_{i=1}^{k}, l, \mathsf{SK}_A^{P}, \mathsf{PK}_A^{P}, \mathsf{SK}_A^{QR}, \mathsf{PK}_A^{QR}) = (\emptyset, \arg\max_i \{a_i\}_{i=1}^{k})$

We will conclude using Theorem 3, our own security proofs for those steps, as well as the proofs in [23].

The whole protocol computes the function:

$$\begin{aligned} f(&\{P_i\}_{i=1}^{n}, \{w_h\}_h, \{\llbracket v_i \rrbracket_B\}_{i=1}^{g}, l, \\ &\mathsf{SK}_A^{P}, \mathsf{PK}_A^{P}, \mathsf{SK}_A^{QR}, \mathsf{PK}_A^{QR}, \\ &\mathsf{SK}_B^{P}, \mathsf{PK}_B^{P}, \mathsf{SK}_B^{QR}, \mathsf{PK}_B^{QR}, \\ &\mathsf{SK}_B^{FHE}, \mathsf{PK}_B^{FHE}) \end{aligned}$$

where $\{P_i\}_{i=1}^{n}$ are the polynomials, $\{w_h\}_h$ are the thresholds for each inner node, g is the number of features of the client's sample, $\{\llbracket v_i \rrbracket_B\}_{i=1}^{g}$ is the input by the client. f_A returns nothing, while f_B returns the plurality-vote class of the random forest evaluation.

A's view now is:

$$\begin{aligned} V_A = (&\{P_i\}_{i=1}^{n}, \{w_h\}_h, \{\llbracket v_i \rrbracket_B\}_{i=1}^{g}, l, \\ &\mathsf{SK}_A^{P}, \mathsf{SK}_A^{QR}, \mathsf{PK}_B^{P}, \mathsf{PK}_B^{QR}, \mathsf{PK}_B^{FHE}; \\ &\mathsf{coins}; \\ &\{[b_h]_B\}_h, \{[\![\![b_h, \dots, b_h]\!]\!]_B\}_h, \\ &\{\llbracket y_{ij} \rrbracket_B\}_{i \in \{1,\dots,n\}, j \in \{1,\dots,k\}}) \end{aligned}$$

where coins is the random tape for encryptions and $\{[b_h]_B\}_h$ the comparison result for each node. We simulate A's real view with the following simulator S_A:

1) Generate a random bit $\tilde{b}_h$ for each inner node in the random forest.
2) Generate random bits y_{ij} for $i \in \{1, \dots, k\}, j \in \{1, \dots, n\}$.
3) Generate a random tape $\widetilde{\mathsf{coins}}$ of the required length. The length can be determined based mainly on the polynomials, which encode the number of trees, number of classes and the number of nodes in the tree.
4) Output

$$\begin{aligned} H_0 = (&\{P_i\}_{i=1}^{n}, \{w_h\}_h, \{\llbracket v_i \rrbracket_B\}_{i=1}^{g}, l, \\ &\mathsf{SK}_A^{P}, \mathsf{SK}_A^{QR}, \mathsf{PK}_B^{P}, \mathsf{PK}_B^{QR}, \mathsf{PK}_B^{FHE}; \\ &\widetilde{\mathsf{coins}}; \\ &\{[\tilde{b}_h]_B\}_h, \{[\![\![\tilde{b}_h, \dots, \tilde{b}_h]\!]\!]_B\}_h, \\ &\{\llbracket \tilde{y}_{ij} \rrbracket_B\}_{i \in \{1,\dots,n\}, j \in \{1,\dots,k\}}) \end{aligned}$$

Since $\widetilde{\mathsf{coins}}$ and coins come from the same distribution, H_0 is indistinguishable from:

$$\begin{aligned} H_1 = (&\{P_i\}_{i=1}^{n}, \{w_h\}_h, \{\llbracket v_i \rrbracket_B\}_{i=1}^{g}, l, \\ &\mathsf{SK}_A^{P}, \mathsf{SK}_A^{QR}, \mathsf{PK}_B^{P}, \mathsf{PK}_B^{QR}, \mathsf{PK}_B^{FHE}; \\ &\mathsf{coins}; \\ &\{[\tilde{b}_h]_B\}_h, \{[\![\![\tilde{b}_h, \dots, \tilde{b}_h]\!]\!]_B\}_h, \\ &\{\llbracket \tilde{y}_{ij} \rrbracket_B\}_{i \in \{1,\dots,n\}, j \in \{1,\dots,k\}}) \end{aligned}$$

Moreover, by the semantic security of QR and FHE (we abstain from the trivial reduction proof here), we can deduce that H_1 is computationally indistinguishable from:

$$\begin{aligned} H_2 = (&\{P_i\}_{i=1}^{n}, \{w_h\}_h, \{\llbracket v_i \rrbracket_B\}_{i=1}^{g}, l, \\ &\mathsf{SK}_A^{P}, \mathsf{SK}_A^{QR}, \mathsf{PK}_B^{P}, \mathsf{PK}_B^{QR}, \mathsf{PK}_B^{FHE}; \\ &\mathsf{coins}; \\ &\{[b_h]_B\}_h, \{[\![\![b_h, \dots, b_h]\!]\!]_B\}_h, \\ &\{\llbracket \tilde{y}_{ij} \rrbracket_B\}_{i \in \{1,\dots,n\}, j \in \{1,\dots,k\}}) \end{aligned}$$

And by the semantic security of Paillier, we get that H_2 is computationally indistinguishable from:

$$\begin{aligned} H_3 = (&\{P_i\}_{i=1}^{n}, \{w_h\}_h, \{\llbracket v_i \rrbracket_B\}_{i=1}^{g}, l, \\ &\mathsf{SK}_A^{P}, \mathsf{SK}_A^{QR}, \mathsf{PK}_B^{P}, \mathsf{PK}_B^{QR}, \mathsf{PK}_B^{FHE}; \\ &\mathsf{coins}; \\ &\{[b_h]_B\}_h, \{[\![\![b_h, \dots, b_h]\!]\!]_B\}_h, \\ &\{\llbracket y_{ij} \rrbracket_B\}_{i \in \{1,\dots,n\}, j \in \{1,\dots,k\}}) \end{aligned}$$

Hence, we showed that

$$\begin{aligned}&V_A(\{P_i\}_{i=1}^n, \{w_h\}_h, \{[\![v_i]\!]_B\}_{i=1}^g, l,\\ &\quad \mathsf{SK}_A^P, \mathsf{PK}_A^P, \mathsf{SK}_A^{QR}, \mathsf{PK}_A^{QR},\\ &\quad \mathsf{SK}_B^P, \mathsf{PK}_B^P, \mathsf{SK}_B^{QR}, \mathsf{PK}_B^{QR},\\ &\quad \mathsf{SK}_B^{FHE}, \mathsf{PK}_B^{FHE})\\ &\equiv_c S_A(\{P_i\}_{i=1}^n, \{w_h\}_h, \{[\![v_i]\!]_B\}_{i=1}^g, l,\\ &\quad \mathsf{SK}_A^P, \mathsf{SK}_A^{QR}, \mathsf{PK}_B^P, \mathsf{PK}_B^{QR}, \mathsf{PK}_B^{FHE})\end{aligned}$$

B's view is

$$\begin{aligned}V_B = (&\{v_i\}_{i=1}^g, l, c, n, k\\ &\mathsf{PK}_A^P, \mathsf{PK}_A^{QR}, \mathsf{SK}_B^P, \mathsf{SK}_B^{QR}, \mathsf{SK}_B^{FHE};\\ &\mathsf{coins};\\ &\{[\![\sum_{i=1}^n y_{ij}]\!]_A\}_{j=1}^n, \arg\max_j\{\sum_{i=1}^n y_{ij}\}_{j=1}^n)\end{aligned}$$

where c is the number inner nodes over all trees, n is the number of trees, k is the number of classes, $[\![\sum_{i=1}^n y_{ij}]\!]_A$ is the encrypted vote count per class and $\arg\max_j\{\sum_{i=1}^n y_{ij}\}$ is the result of the $\arg\max$ protocol and hence the output of B.

We simulate B by the simulator S_B as follows:

1) Generate n random Paillier encryptions $\{[\![\tilde{y}_j]\!]_A\}_{j=1}^n$.
2) Generate a random value between $v \leftarrow \{1, \ldots, n\}$.
3) Generate a random tape $\widetilde{\mathsf{coins}}$ of the required length, which can be determined by c, n and k.
4) Output

$$\begin{aligned}H_0' = (&\{v_i\}_{i=1}^g, l, c, n, k\\ &\mathsf{PK}_A^P, \mathsf{PK}_A^{QR}, \mathsf{SK}_B^P, \mathsf{SK}_B^{QR}, \mathsf{SK}_B^{FHE};\\ &\widetilde{\mathsf{coins}};\\ &\{[\![\tilde{y}_j]\!]_A\}_{j=1}^n, v)\end{aligned}$$

Given that $\widetilde{\mathsf{coins}}$ and coins both are sampled from the same distribution with the same length, we can conclude that $H_0' \equiv_c H_1'$, with H_1' below:

$$\begin{aligned}H_1' = (&\{v_i\}_{i=1}^g, l, c, n, k\\ &\mathsf{PK}_A^P, \mathsf{PK}_A^{QR}, \mathsf{SK}_B^P, \mathsf{SK}_B^{QR}, \mathsf{SK}_B^{FHE};\\ &\mathsf{coins};\\ &\{[\![\tilde{y}_j]\!]_A\}_{j=1}^n, v)\end{aligned}$$

Next, we show the indistinguishability of H_1' and V_B by giving a reduction to the semantic security of Paillier. To this end, we assume that we have a distinguisher $\mathcal{D}$ that can distinguish H' and V_B. Specifically, given

$$\begin{aligned}(&\{v_i\}_{i=1}^g, l, c, n, k\\ &\mathsf{PK}_A^P, \mathsf{PK}_A^{QR}, \mathsf{SK}_B^P, \mathsf{SK}_B^{QR}, \mathsf{SK}_B^{FHE};\\ &\mathsf{coins};\\ &\{[\![y_j]\!]_A\}_{j=1}^n, v)\end{aligned}$$

$\mathcal{D}$ outputs 1 if $v = \arg\max_j\{y_j\}_{j=1}^n$ and 0 otherwise. Then, we construct a reduction $\mathcal{R}$ as follows:

1) On input PK, pick uniformly at random $x, y, z \leftarrow M$, such that $x \neq y \neq z$.
2) Order the chosen values (w.l.o.g., we from here on assume $x < y < z$).
3) Generate new keys $\mathsf{PK}_A^{QR}, \mathsf{SK}_B^P, \mathsf{SK}_B^{QR}, \mathsf{SK}_B^{FHE}$.
4) Choose challenger messages $m_0 = x$, $m_1 = z$ and give them to the semantic security challenger.
5) Receive c from the challenger and query the distinguisher $\mathcal{D}(\emptyset, 0, 0, 2, 0, \mathsf{PK}, \mathsf{PK}_A^{QR}, \mathsf{SK}_B^P, \mathsf{SK}_B^{QR}, \mathsf{SK}_B^{FHE}; \emptyset; \{[\![y]\!]_{\mathsf{PK}}, c\}, 2)$, which returns b.
6) Return b to the challenger.

Since we simulate both cases perfectly to the distinguisher, its success probability transfers exactly to our reduction in the semantic security game. Since Paillier encryption is shown to be semantically secure under the Decisional Composite Residuosity Assumption, the distinguisher must have at most negligible success probability.

Given the correctness of the protocol as well as the computational indistinguishability of both simulators and views, we can apply Theorem 3. We replace the ideal calls by our provable secure building blocks. Theorem 3 then gives us the security of our scheme in the semi-honest model. □