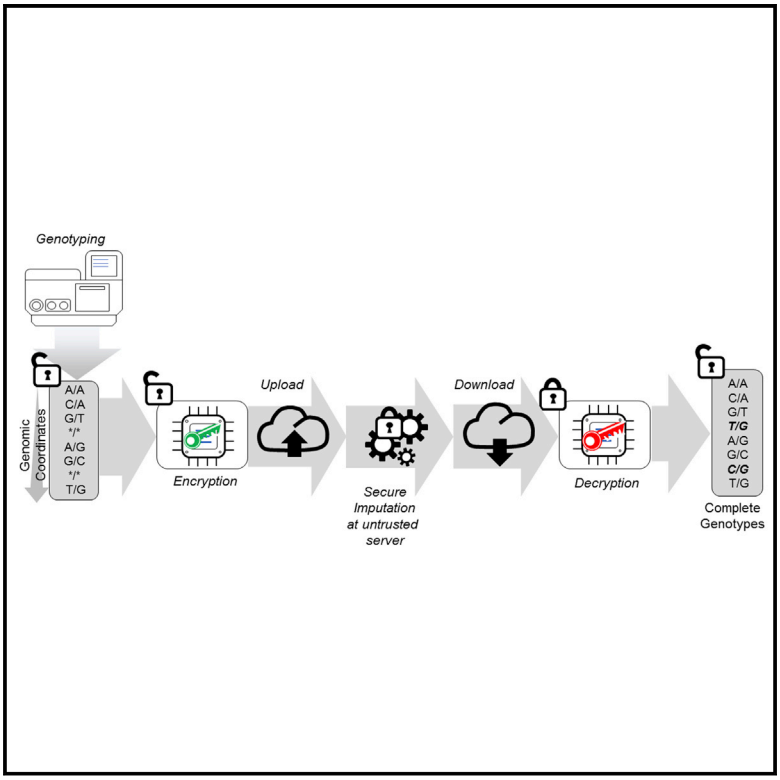
Methods

Cell Systems

Ultrafast homomorphic encryption models enable secure outsourcing of genotype imputation

Graphical abstract



Authors

Miran Kim, Arif Ozgun Harmanci, Jean-Philippe Bossuat, ..., Yongsoo Song, Juan Troncoso-Pastoriza, Xiaoqian Jiang

Correspondence

arif.o.harmanci@uth.tmc.edu (A.O.H.), xiaoqian.jiang@uth.tmc.edu (X.J.)

In brief

Kim et al. present fast and efficient methods for privacy-aware outsourcing of genotype imputation. The presented secure analysis frameworks can be adopted by other high-throughput genomic data analysis tools.

Highlights

- Fast homomorphic encryption enables secure and practical genotype imputations
- Secure methods require comparable resources as non-secure methods
- Accuracy of secure methods can be improved using population specific panels

Kim et al., 2021, Cell Systems *12*, 1108–1120
November 17, 2021
https://doi.org/10.1016/j.cels.2021.07.010

CellPress

CellPress
OPEN ACCESS

Cell Systems

Methods

Ultrafast homomorphic encryption models enable secure outsourcing of genotype imputation

Miran Kim,[1,16] Arif Ozgun Harmanci,[2,16,17,*] Jean-Philippe Bossuat,[3] Sergiu Carpov,[4,5] Jung Hee Cheon,[6,7] Ilaria Chillotti,[8] Wonhee Cho,[6] David Froelicher,[3] Nicolas Gama,[4] Mariya Georgieva,[4] Seungwan Hong,[6] Jean-Pierre Hubaux,[3] Duhyeong Kim,[6] Kristin Lauter,[9] Yiping Ma,[10] Lucila Ohno-Machado,[11] Heidi Sofia,[12] Yongha Son,[13] Yongsoo Song,[14] Juan Troncoso-Pastoriza,[3] and Xiaoqian Jiang[15,*]

[1]Department of Computer Science and Engineering and Graduate School of Artificial Intelligence, Ulsan National Institute of Science and Technology, Ulsan 44919, Republic of Korea
[2]Center for Precision Health, School of Biomedical Informatics, University of Texas Health Science Center, Houston, TX 77030, USA
[3]École polytechnique fédérale de Lausanne, Lausanne, Switzerland
[4]Inpher, EPFL Innovation Park Bàtiment A, 3rd Fl, 1015 Lausanne, Switzerland
[5]CEA, LIST, 91191 Gif-sur-Yvette Cedex, France
[6]Department of Mathematical Sciences, Seoul National University, Seoul 08826, Republic of Korea
[7]Crypto Lab Inc., Seoul 08826, Republic of Korea
[8]Zama, Paris, France and imec-COSIC, KU Leuven, Leuven, Belgium
[9]West Coast Head of Research Science, Facebook AI Research (FAIR), Seattle, Washington
[10]University of Pennsylvania, Philadelphia, PA 19104, USA
[11]UCSD Health Department of Biomedical Informatics, University of California, San Diego, San Diego, CA 92093, USA
[12]National Institutes of Health (NIH) - National Human Genome Research Institute, Bethesda, MD 20892, USA
[13]Samsung SDS, Seoul, Republic of Korea
[14]Department of Computer Science and Engineering, Seoul National University, Seoul 08826, Republic of Korea
[15]Center for Secure Artificial intelligence For hEalthcare (SAFE), School of Biomedical Informatics, University of Texas Health Science Center, Houston, TX 77030, USA
[16]These authors contributed equally
[17]Lead contact
*Correspondence: arif.o.harmanci@uth.tmc.edu (A.O.H.), xiaoqian.jiang@uth.tmc.edu (X.J.)
https://doi.org/10.1016/j.cels.2021.07.010

SUMMARY

Genotype imputation is a fundamental step in genomic data analysis, where missing variant genotypes are predicted using the existing genotypes of nearby "tag" variants. Although researchers can outsource genotype imputation, privacy concerns may prohibit genetic data sharing with an untrusted imputation service. Here, we developed secure genotype imputation using efficient homomorphic encryption (HE) techniques. In HE-based methods, the genotype data are secure while it is in transit, at rest, and in analysis. It can only be decrypted by the owner. We compared secure imputation with three state-of-the-art non-secure methods and found that HE-based methods provide genetic data security with comparable accuracy for common variants. HE-based methods have time and memory requirements that are comparable or lower than those for the non-secure methods. Our results provide evidence that HE-based methods can practically perform resource-intensive computations for high-throughput genetic data analysis. The source code is freely available for download at https://github.com/K-miran/secure-imputation.

INTRODUCTION

Whole-genome sequencing (WGS) (Ng and Kirkness, 2010; Shendure et al., 2017) has become the standard technique in clinical settings for tailoring personalized treatments (Rehm, 2017) and in research settings for building reference genetic databases (Schwarze et al., 2018; 1000 Genomes Project Consortium, 2015; Chisholm et al., 2013). Technological advances in the last decade enabled a massive increase in the throughput of WGS methods (Heather and Chain, 2016), which provided the opportunity for population-scale sequencing (Goldfeder et al., 2017), where a large sample from a population is sequenced for studying ancestry and complex phenotypes (Lango Allen et al., 2010 throughout the article Locke et al., 2015), as well as rare (Agarwala et al., 2013; Gibson, 2012; Chen et al., 2019) and chronic diseases (Cooper et al., 2008). Although the price of sequencing has been decreasing, the sample sizes are increasing to accommodate the power necessary for new studies. It is anticipated that tens of millions of individuals will have access to their personal genomes in the next few years.

The increasing size of sequencing data creates new challenges for sharing, storage, and analyses of genomic data.

Among these, genomic data security and privacy have received much attention in recent years. Most notably, the increasing prevalence of genomic data, e.g., direct-to-consumer testing and recreational genealogy, makes it harder to share genomic data due to privacy concerns. Genotype data are very accurate in identifying the owner because of their high dimensionality, and leakage can cause concerns about discrimination and stigmatization (Nissenbaum, 2009). Also, the recent cases of forensic usage of genotype data are making it very complicated to share data for research purposes. The identification risks extend to family members of the owner, since a large portion of the genetic data are shared with relatives. Many attacks have been proposed on genomic data sharing models, where the correlative structure of the variant genotypes provides enough power to adversaries to make phenotype inference and individual re-identification possible (Nyholt et al., 2009). Therefore, it is of the utmost importance to ensure that genotype data are shared securely. There is a strong need for new methods and frameworks that will enable decreasing the cost and facilitate the analysis and management of genome sequencing.

One of the main techniques used for decreasing the cost of large-scale genotyping is *in silico* genotype imputation; i.e., measuring genotypes at a subsample of variants, e.g., using a genotyping array, and then utilizing the correlations among the genotypes of nearby variants (the variants that are close to each other in genomic coordinates) and imputing the missing genotypes using the sparsely genotyped variants (Howie et al., 2011; Das et al., 2018; Marchini and Howie, 2010). Imputation methods aim at capturing the linkage disequilibrium patterns on the genome. These patterns emerge because genomic recombination occurs at hotspots rather than at uniformly random positions along the genome. The genotyping arrays are designed around the idea of selecting a small set of "tag" variants, as small as 1% of all variants, that optimize the trade-off between cost and imputation accuracy (Hoffmann et al., 2011; Stram, 2004). Imputation methods learn the correlations among variant genotypes by using population-scale sequencing projects (Loh et al., 2016). In addition to filling in missing genotypes, the imputation process has many other advantages. Combining low-cost array platforms with computational genotype imputation methods decreases genotyping costs and increases the power of genome-wide association studies (GWASs) by increasing sample sizes (Tam et al., 2019). Accurate imputation can also greatly help with the fine-mapping of causal variants (Schaid et al., 2018) and is vital for meta-analysis of the GWAS (Evangelou and Ioannidis, 2013). Genotype imputation is now a standard and integral step in performing GWAS. Although imputation methods can predict only the variant genotypes that exist in the panels, the panels' sample sizes are increasing rapidly; e.g., in projects such as TOPMed (Taliun et al., 2019; TOPMed, 2016) will provide training data for imputation methods to predict rarer variant genotypes, and this can increase the sensitivity of GWAS.

Although imputation and sparse genotyping methods enable a vast decrease in genotyping costs, they are computationally very intensive and require management of large genotype panels and interpretation of the results (Howie et al., 2012). The imputation tasks can be outsourced to third parties, such as the Michigan Imputation Server, where users upload the genotypes (as a Variant Call Format, VCF, file) to a server that performs imputation internally using a large computing system. The imputed genotypes are then sent back to the user. However, there are major privacy (Naveed et al., 2015) and data security (Berger and Cho, 2019) concerns over using these services, since the genotype data are analyzed in plaintext format where any adversary who has access to the third party's computer can view, copy, or even modify the genotype data. As genotype imputation is one of the central initial steps in many genomic analysis pipelines, it is essential that the imputation be performed securely to ensure that these pipelines can be computed securely as a whole. For instance, although several secure methods for GWAS have been developed (Cho et al., 2018), if genotype imputation (a vital step in GWAS analyses) is not performed securely, it is not possible to make sure GWAS analysis can be performed securely.

In order to test the current state-of-the-art methodologies for benchmarking the feasibility of the cryptographic methods for genotype imputation, we organized the genotype imputation track in iDASH2019 Genomic Privacy Challenges. This track benchmarked more than a dozen methods on a small scale (supplemental information; Table S1) to rank the most promising approaches for secure genotype imputation. The methods developed by the top winning teams led us (organizers and contestants) to perform this study to report a more comprehensive analysis of the secure genotype imputation framework, including benchmarks with state-of-the-art methods. We developed and implemented several approaches for secure genotype imputation. Our methods make use of the homomorphic encryption (HE) formalism (Gentry, 2009) that provides mathematically provable, and potentially one of the strongest security guarantees for protecting genotype data while imputation is performed in an untrusted semi-honest environment. To include a comprehensive set of approaches, we focus on three state-of-the-art HE cryptosystems, namely, Brakerski/Fan-Vercauteren (BFV) (Brakerski, 2012; Fan and Vercauteren, 2012), Cheon-Kim-Kim-Song (CKKS) (Cheon et al., 2017), and the fully homomorphic encryption over the torus (TFHE) (Chillotti et al., 2020; Boura et al., 2018). In our HE-based framework, genotype data are encrypted by the data owner before outsourcing the data. After this point, the data always remain encrypted, i.e., encrypted in transit, in use, and at rest; it is never decrypted until the results are sent to the data owner. The strength of our HE-based framework stems from the fact that the genotype data remain encrypted even while the imputation is being performed. Hence, even if the imputation is outsourced to an untrusted third party, any semi-honest adversaries learn nothing from the encrypted data. This property makes the HE-based framework very powerful. For an untrusted third party who does not have access to the private key, the genotype data are indistinguishable from random noise (i.e., practically of no use) at any stage of the imputation process. Our HE framework provides the strongest form of security for outsourcing genotype imputation compared with any other approach under the same adversarial model.

HE-based frameworks have been deemed impractical since their inception. Therefore, in comparison with other cryptographically secure methods, such as multiparty computation (Cho et al., 2018) and trusted execution environments (Kockan et al., 2020), HE-based frameworks have received little attention.

Recent theoretical breakthroughs in the HE literature and a strong community effort (HES, n.d. 2020) have since rendered HE-based systems practical. However, many of these improvements are only beginning to be reflected in practical implementations and applications of HE algorithms. In this study, we provide evidence for the practicality of the HE formalism by building secure and ready-to-deploy methods for genotype imputation. We perform detailed benchmarking of the time and memory requirements of HE-based imputation methods and demonstrate the feasibility of large-scale secure imputation. In addition, we compared HE-based imputation methods with the state-of-the-art plaintext, i.e., non-secure, imputation methods, and we found comparable performance (with a slight decrease) in imputation accuracy with the benefit of total genomic data security.

We present HE-based imputation methods in the context of two main steps, as this enables a general modular approach. The first step is imputation model building, where imputation models are trained using the reference genotype panel with a set of tag variants (variant genotypes on an Illumina array platform) to impute the genotypes for a set of target variants, e.g., common variants in the 1,000 Genomes Project (1000 Genomes Project Consortium, 2015) samples. The second step is the secure imputation step, where the encrypted tag variant genotypes are used to predict the target genotypes (which are encrypted) by using the imputation models trained in the first step. This step, i.e., imputation model evaluation using the encrypted tag variant genotypes, is where the HE-based methods are deployed. In principle, the model training step needs to be performed only once when the tag variants do not change, i.e., the same array platform is used for multiple studies. Although these steps seem independent, model evaluation is heavily dependent on the representation and encoding of the genotype data, and the model complexity affects the timing and memory requirements of the secure outsourced imputation methods. However, our results suggest that linear models (or any other model that can be approximated by linear models) can be almost seamlessly trained and evaluated securely, where the model builders (1st step) and model evaluators (2nd step) can work independently. However, our results also show that there is an accompanying performance penalty, especially for the rare variants, in using these models, and we believe that new and accurate methods are needed to provide both privacy and imputation accuracy. It should be noted that the performance penalty stems not from HE-model evaluation but from the lower performance of plaintext models. We provide a pipeline that implements both model training and evaluation steps so that it can be run on any selection of tag variants. We make the implementations publicly available, so that they can be used as a reference by the computational genomics community.

RESULTS

We present the scenario and the setting for secure imputation and describe the secure imputation approaches we developed. Next, we present accuracy comparisons with the current state-of-the-art non-secure imputation methods and the time and memory requirements of the secure imputation methods. Finally, we present the comparison of the time and memory requirements of our secure imputation pipeline with the non-secure methods.

Genotype imputation scenario

Figure 1A illustrates the secure imputation scenario. A researcher genotypes a cohort of individuals by using genotyping arrays or other targeted methods, such as whole-exome sequencing, and calls the variants using a variant caller such as the Genome Analysis Toolkit, GATK (Depristo et al., 2011). After genotyping, the genotypes are stored in plaintext, i.e., unencrypted and not secure for outsourcing. Each variant genotype is represented by one of the three values $\{0, 1, 2\}$, where 0 indicates a homozygous reference genotype, 1 indicates a heterozygous genotype, and 2 indicates a homozygous alternate genotype. To secure the genotype data, the researcher generates two keys: a public key for encrypting the genotype data and a private key for decrypting the imputed data. The public key is used to encrypt the genotype data into ciphertext, i.e., random-looking data that contain the genotype data in a secure form. It is mathematically provable (i.e., equivalent to the hardness of solving the ring learning with errors, or RLWE, problem, Lyubashevsky et al., 2010) that the encrypted genotypes cannot be decrypted into plaintext genotype data by a third party without the private key, which is in the possession of only the researcher. Even if an unauthorized third party copies the encrypted data without authorization (e.g., hacking, stolen hard drives), they cannot gain any information from the data as they are essentially random noise without the private key. The security (and privacy) of the genotype data are therefore guaranteed, as long as the private key is not compromised. The security guarantee of the imputation methods is based on the fact that genotype data are encrypted in transit, during analysis, and at rest. The only plaintext data that are transmitted to the untrusted entity are the locations of the variants, i.e., the chromosomes and positions of the variants. Since the variant locations are publicly known for genotyping arrays, they should not leak any information. However, when the genotyping is performed by sequencing-based methods, the variant positions may leak information, as we discuss more in the next sections.

The encrypted genotypes are sent through a channel to the imputation service. The channel does not have to be secure against an eavesdropper because the genotype data are encrypted by the researcher. However, secure channels should be authenticated to prevent malicious man-in-the-middle attacks (Gangan, 2015). The encrypted genotypes are received by the imputation service, an honest-but-curious entity, i.e., they will receive the data legitimately and extract all the private information they can from the data. However, a privacy breach is impossible as the data are always encrypted when they are in the possession of the imputation service. Hence, the only reasonable action for the secure imputation server is to perform the genotype imputation and to return the data to the researcher. It is possible that the imputation server acts maliciously and intentionally returns bad-quality data to the researcher using badly calibrated models. However, it is economically or academically reasonable to assume that this is unlikely, since it would be easy to detect this behavior on the researcher's side and to advertise the malicious or low quality of the service to other researchers. Therefore, we assume that the secure server is semi-honest, and it performs the imputation task as accurately as possible. However, more complex malicious entities that perform complex attacks (e.g., slight biases in the models) are harder to detect. We treat these scenarios as out of scope of

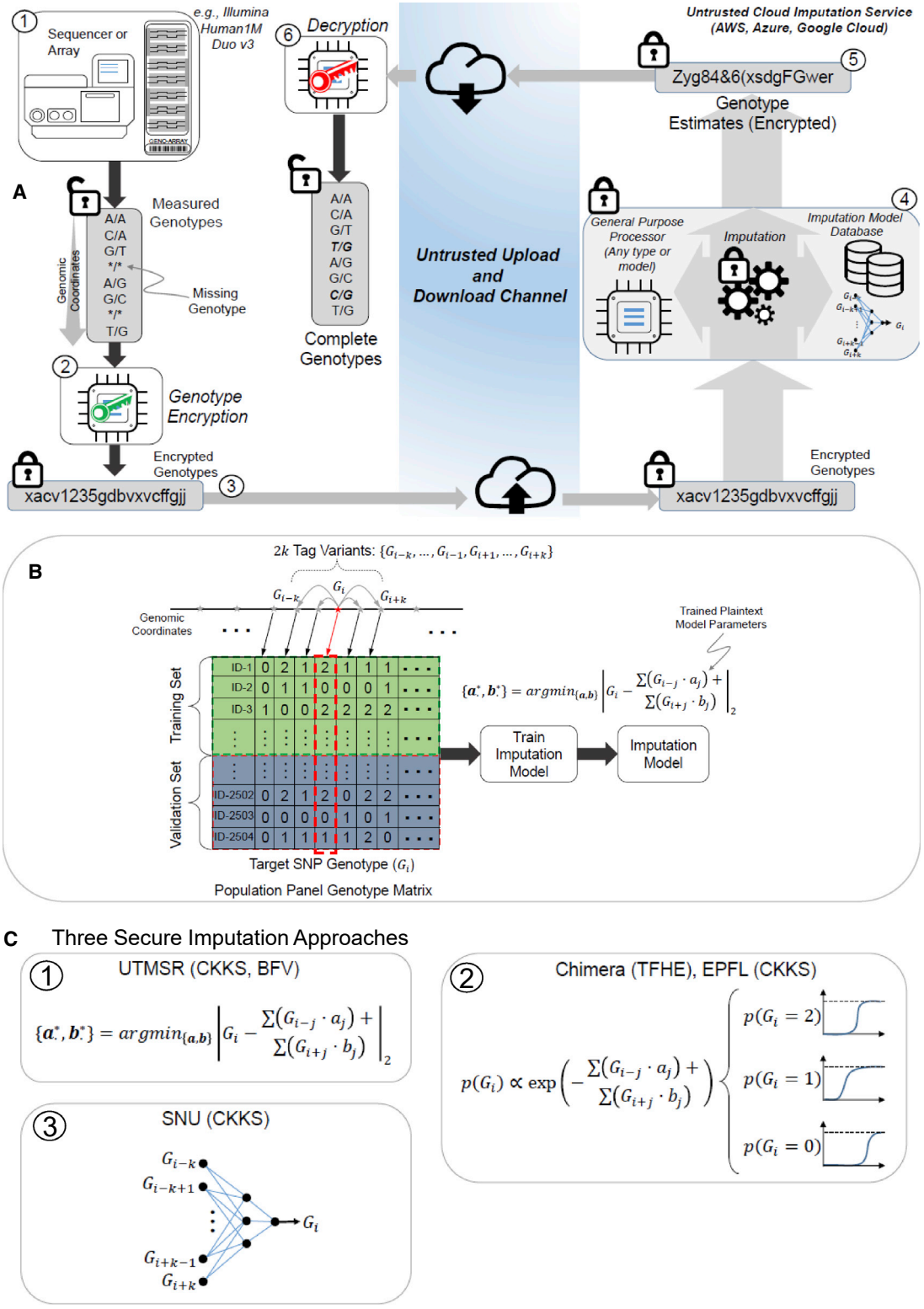


Figure 1. Illustration of secure genotype imputation

(A) Illustration of the genotype imputation scenario. The incomplete genotypes are measured by genotyping arrays with missing genotypes (represented by stars). Encryption generates random-looking strings from the genotypes. At the server, encrypted genotypes are encoded, then they are used to compute the missing variant genotype probabilities. The encrypted probabilities are sent to the researcher, who decrypts the probabilities and identifies the genotypes with the highest probabilities (italic values).

(legend continued on next page)

our current study. Providing secure services against malicious entities is a worthwhile direction to explore for future studies.

After the receipt of the encrypted genotypes by the server, the first step is recoding of the encrypted data into a packed format (Figure S1) that is optimized for the secure imputation process. This step is performed to decrease time requirements and to optimize the memory usage of the imputation process. The data are coded to enable analysis of multiple genotypes in one cycle of the imputation process (Dowlin et al., 2017). The next step is the secure evaluation of the imputation models, which entails securely computing the genotype probability for each variant by using the encrypted genotypes. The variants received from the researcher are treated as tag variants whose genotypes are used as features in the imputation model to predict the "target" variants, i.e., the missing variants (Figure 1B). For each target variant, the corresponding imputation model uses the genotypes of the nearby tag variants to predict the target variant genotype in terms of genotype probabilities. In other words, we use a number of nearby tag variants to build an imputation model for the respective target variant such that the tag variants that are nearby (in genomic coordinates) are treated as the features for assigning genotype scores for the target variant. After the imputation is performed, the encrypted genotype probabilities are sent to the researcher. The researcher decrypts the genotype probabilities by using a private key. The final genotypes can be assigned using the maximum probability genotype estimate, i.e., by selecting the genotype with the highest probability for each variant.

Genotype imputation models

We provide five approaches implemented by four different teams. For simplicity of description, we refer to the teams as Chimera, EPFL, SNU, and UTHealth-Microsoft Research (UTMSR) (see STAR Methods). Among these, CKKS is used in three different approaches (EPFL-CKKS, SNU-CKKS, and UTMSR-CKKS), and BFV and TFHE are each utilized by separate approaches (UTMSR-BFV and Chimera-TFHE, respectively). The teams independently developed and trained the plaintext imputation models using the reference genotype panel dataset. For each target variant, the tag variants in the vicinity of the target variant are used for imputing the target variant, i.e., the tag variants in the vicinity are used as features in the imputation models. The chimera team trained a logistic regression model and the EPFL team trained a multinomial logistic regression model. (Figure S4; Tables S3, S4, and S7); the SNU team used a 1-hidden-layer neural network (Figures 1C, S2, and S3; Table S5); and the UTMSR team trained a linear regression model (Figures 1C and S5).

Genotype representation

All methods treat the genotypes as continuous predictions, except for the Chimera and SNU teams who utilized a one-hot encoding of the genotypes (see STAR Methods), e.g., $0 \rightarrow (1,0,0)$, $1 \rightarrow (0,1,0)$, and $2 \rightarrow (0,0,1)$.

Tag variant (feature) selection

The selection of the tag variants is important as these represent the features that are used for imputing each target variant. In general, we found that the models that use 30–40 tag variants provide optimal results (for the current array platform) in terms of imputation accuracy (Tables S2, S5, S6, and S8). As previous studies have shown, tag variant selection can provide an increase in imputation accuracy (Yu and Schaid, 2007). Finally, we observed a general trend of linear scaling with the number of target variants (as shown in Figure S6 and other Tables S1–S9). This provides evidence that there is minimal extra overhead (in addition to the linear increasing sample size) for scaling to genome-wide and population-wide computations.

Training and secure evaluation of models

We present the accuracy comparison results further on. We include extended discussion of the specific ideas used for training and for secure evaluation of the genotype imputation models in supplemental information.

Accuracy comparisons with the non-secure methods

We first analyzed the imputation accuracy of the secure methods with their plaintext (non-secure) counterparts that are the most popular state-of-the-art imputation methods. We compared secure imputation methods with IMPUTE2 (Howie et al., 2009), Minimac3 (Das et al., 2016) (and Minimac4, which is an efficient re-implementation of Minimac3), and Beagle (Browning et al., 2018) methods. These plaintext methods utilize Hidden Markov models (HMMs) for genotype imputation (see STAR Methods). The population panels and the pre-computed estimates of the recombination frequencies are taken as input to the methods. Each method is set to provide a measure of genotype probabilities, in addition to the imputed genotype values.

To perform comparisons in a realistic setting, we used the variants on the Illumina Duo 1M version 3 array platform (Johnson et al., 2013). This is a popular array platform that covers more than 1.1 million variants and is used by population-scale genotyping studies such as HAPMAP (Belmont et al., 2003). We extracted the genotypes of the variants that were probed by this array platform and overlapped with the variants identified by the 1,000 Genomes Project population panel of 2,504 individuals. For simplicity of comparisons, we focused on chromosome 22. The variants that are probed by the array are treated as the tag variants that are used to estimate the target variant genotypes. The target variants are defined as variants on chromosome 22 whose allele frequency is greater than 5% as estimated by the 1,000 Genomes Project (1000 Genomes Project Consortium, 2015). We used the 16,184 tag variants and 80,882 common target variants. Then, we randomly divided the 2,504 individuals into a training genotype panel of 1,500 samples and a testing panel of 1,004 samples. The training panel is used as the input to the plaintext methods (i.e., IMPUTE2, Minimac3-4, and Beagle) and also for building the plaintext imputation models of the secure methods. Each method is then used to impute the

(B) Building of the plaintext model for genotype imputation. The server uses a publicly available panel to build genotype estimation models for each variant. The models are stored in the plaintext domain. The model in the current study is a linear model where each variant genotype is modeled using genotypes of variants within a k variant vicinity of the target variant.

(C) The plaintext models implemented under the secure frameworks.

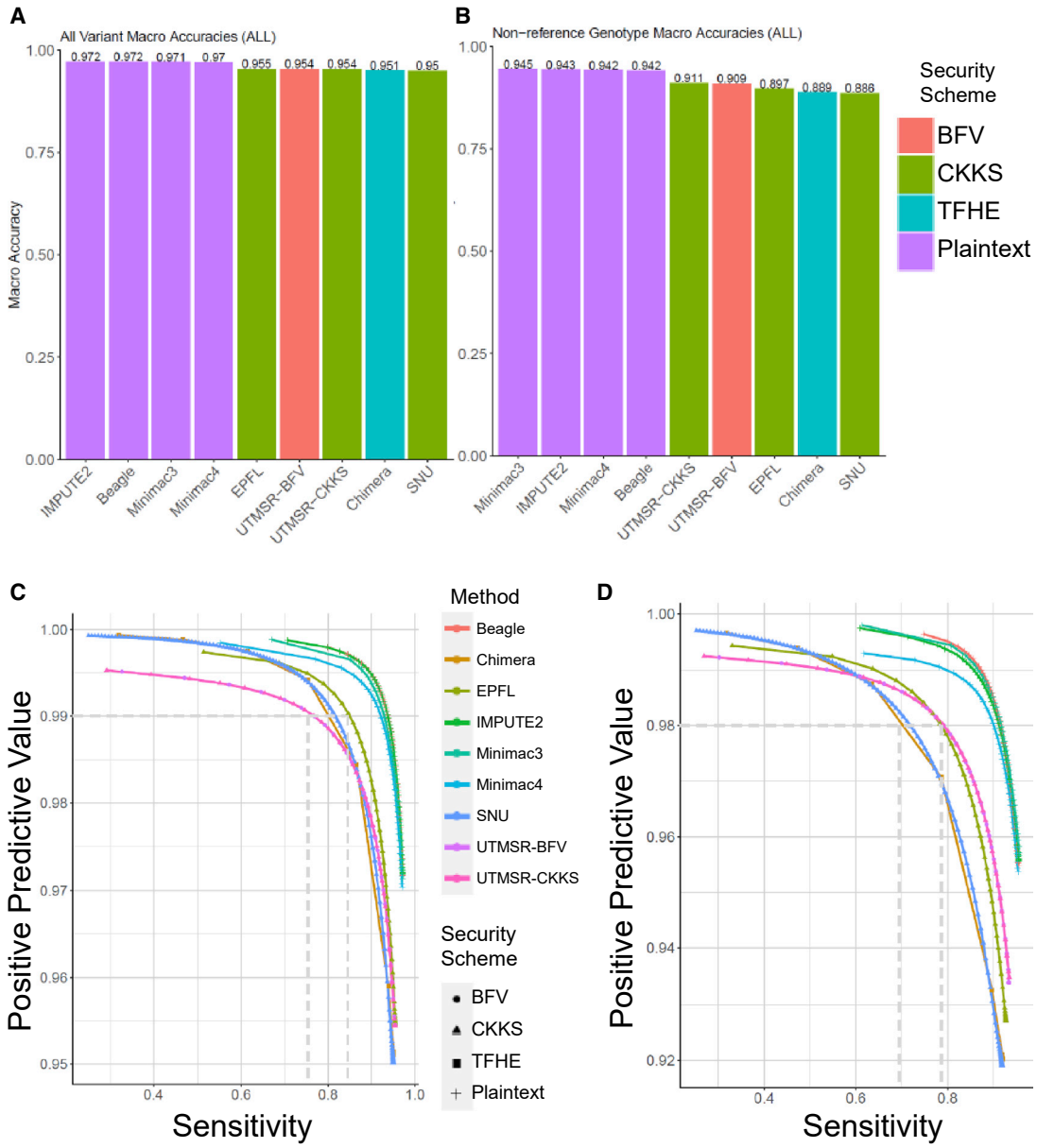


Figure 2. Accuracy benchmark

(A–D) Accuracy for all genotypes (A) and the non-reference genotypes (B) are shown for each method (x axis). The average accuracy value is shown at the top of each bar for comparison. Precision-recall curves are plotted for all genotypes (C) and the non-reference genotypes (D). Plaintext indicates the non-secure methods.

target variants using the tag variants. Figure 2A shows the comparison of genotype prediction accuracy computed over all the predictions made by the methods. The non-secure methods show the highest accuracy among all the methods. The secure methods exhibit very similar accuracy, whereas the closest method follows with only a 2%–3% decrease in accuracy. To understand the differences between the methods, we also computed the accuracy of the non-reference genotype predictions (see STAR Methods; Figure 2B). The non-secure methods show slightly higher accuracy compared with the secure methods. These results indicate that the proposed secure methods provide perfect data privacy at the cost of a slight decrease in imputation accuracy.

Next, we assessed whether the genotype probabilities (or scores) computed from the secure methods provide meaningful measures for choosing reliably imputed genotypes. For this, we calculated the sensitivity and the positive predictive value (PPV) of the imputed genotypes whose scores exceed the cutoff (see STAR Methods). To analyze how cutoff selections affect the accuracy metrics, we shifted the cutoff (swept the cutoff over the range of genotype scores) so that the accuracy is computed for the most reliable genotypes (high cutoff) and for the most

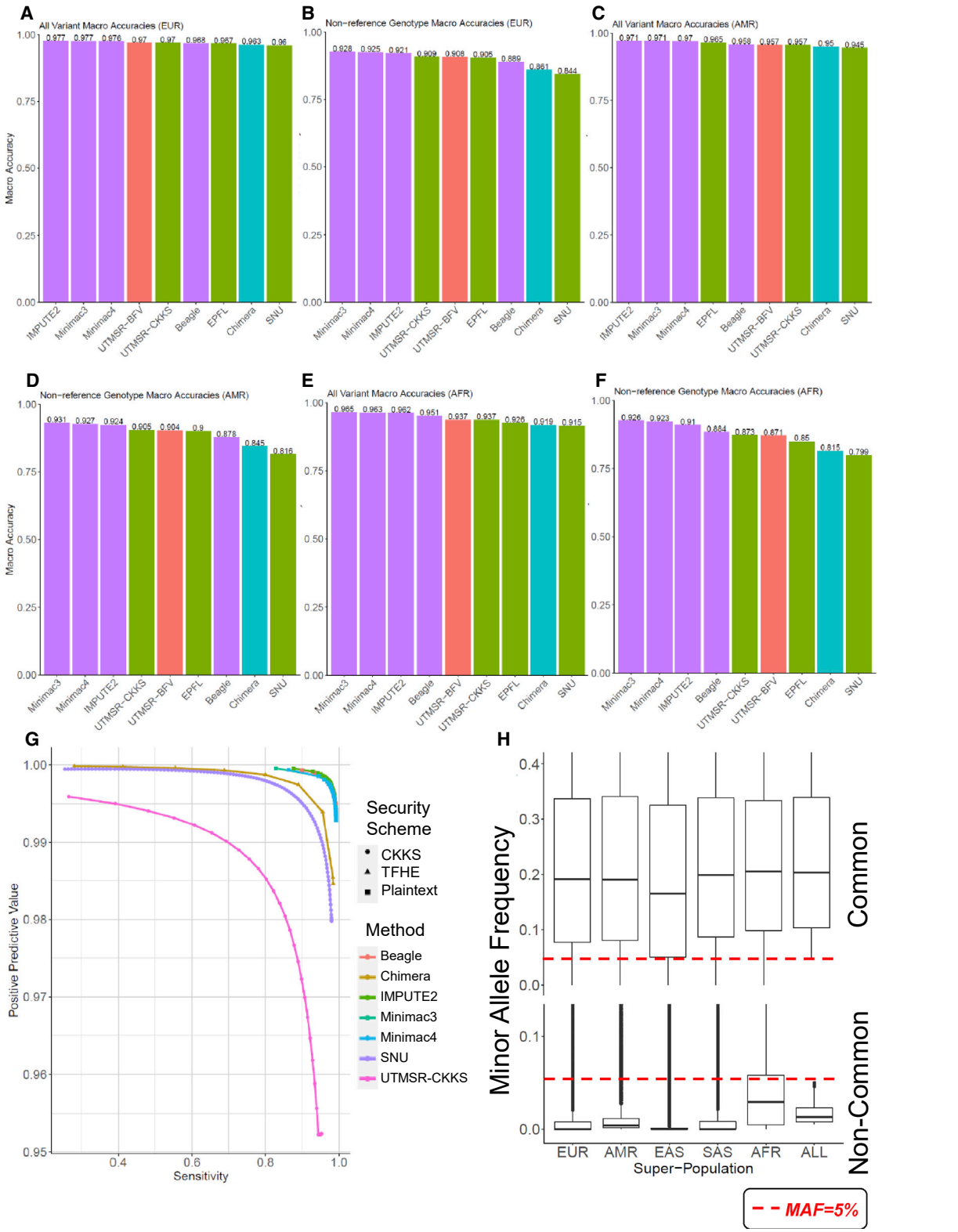


Figure 3. Genotype imputation accuracy benchmarking for stratified populations

(A–H) The population stratification of the accuracy is shown for EUR all genotypes (A) and non-ref genotypes (B), AMR all (C), and non-ref (D) genotypes, and AFR all (E), and non-ref genotypes.

(legend continued on next page)

inclusive genotypes (low cutoff). We then plotted the sensitivity versus the PPV (Figure 2C). Compared with the secure methods, the non-secure methods generally show higher sensitivity at the same PPV. However, secure methods can capture more than 80% of the known genotypes with 98% accuracy. The same results hold for the non-reference genotypes' prediction accuracy (Figure 2D). These results indicate that secure genotype predictions can be filtered by setting cutoffs to improve accuracy.

We also evaluated the population-specific effects on imputation accuracy. For this, we divided the testing panel into three populations—210 European (EUR), 135 American (AMR), and 272 African (AFR) samples—as provided by the 1,000 Genomes Project. The training panel yielded 389 AFR, 212 AMR, and 293 EUR samples. Figures 3A and 3B show genotype and non-reference genotype accuracy for EUR, respectively. We observed that the non-secure and secure methods are similar in terms of accuracy. We observed that the secure CKKS (UTMSR-CKKS) scheme with a linear prediction model outperformed Beagle in the EUR population, with marginally higher accuracy. We observed similar results for AMR populations where the non-secure methods performed at the top and secure methods showed very similar but slightly lower accuracy (Figures 3C and 3D). For AFR populations, the non-reference genotype prediction accuracy is lower for all the methods (Figures 3E and 3F). This is mainly rooted in the fact that the African populations show distinct properties that are not yet well characterized by the 1,000 Genomes Panels. We expect that the larger panels can provide better imputation accuracy.

To further investigate the nature of the imputation errors, we analyzed the characteristics of imputation errors of each method by computing the confusion matrices (Figure S7). We found that the most frequent errors are made when the real genotype is heterozygous, and the imputed genotype is a homozygous reference genotype. The pattern holds predominantly in secure and non-secure methods, although the errors are slightly lower, as expected, for the non-secure methods. Overall, these results indicate that secure imputation models can provide genotype imputations comparable with their non-secure counterparts.

To test the performance of the methods on rare variants, we focused on the 117,904 variants whose minor allele frequency (MAF) is between 0.5% and 5%. These variants represent harder to impute variants since they are much less represented compared with the common variants. The results show that the vicinity-based approaches that our methods use show a clear decrease in performance compared with the HMM-based approaches (Figure 3G). This is expected since our approaches depend heavily on the existence of well-represented training datasets. In the rare variants, however, the number of training examples for the non-reference genotypes goes as low as 1 or 2 examples over 1,000 individuals. That is why we observed a substantial decrease in the imputation power in our methods. Interestingly, we observed that the more complex methods (Chimera's logistic regression and SNU's neural network approach) provided comparably better accuracy than the ordinary linear model, which suggests that the more complex vicinity-based models can perform more accurate imputation for rare variants. In summary to this comparison, the rare variants represent challenging cases and a limitation for vicinity-based secure approaches.

It should be noted that a substantial portion of the rare variants are shown to be population specific (Bomba et al., 2017). To test for this, we analyzed the population specificity of the variants by computing the population-specific AF of these variants. We observed that most of the rare variants show enrichment in the African populations (Figure 3H) with a median MAF of around 2%–3% for AFR. Compared with the rare variants, the common variants showed a much more frequent and more uniform representation among the populations. These results highlight that rare variants can potentially be more accurately imputed using population-specific panels, which is in concordance with previous studies (Kowalski et al., 2019). Finally, from the perspective of downstream analyses, such as GWAS, high allele frequency variants are much more useful, since even the highly powered GWAS studies perform stringent MAF cutoffs at 2%–3% to ensure that the causal variants are not false positives (Sung et al., 2018).

Timing and memory requirements of secure imputation methods

One of the main critiques of HE methods is that they are impractical due to memory and time requirements. Therefore, we believe that the most important challenge is to make HE methods practical in terms of memory and time. To assess and demonstrate the practicality of the secure methods, we performed a detailed analysis of the time and memory requirements of secure imputation methods. We divided the imputation process into four steps (key generation, encryption, secure model evaluation, and decryption), and we measured the time and the overall memory requirements. Figure 4A shows the detailed time requirements for each step. In addition, we studied the scalability of secure methods. For this, we report the time requirements for 20,000 (20K), 40,000 (40K), and 80,000 (80K) target variants to present how the time requirements scale with the number of target variants. The secure methods spend up to 10 ms for key generation. In the encryption step, all methods were well below 2 s. The most time-consuming step of evaluation took less than 10 s, even for the largest set of 80K variants. Decryption, the last step, took less than 2 s. Except for the key generation and encryption, all methods exhibited linear scaling with the increasing number of target variants. Overall, the total time spent in secure model evaluation took less than 25 s (Figure 4B). This could be ignored when compared with the total time requirements of the non-secure imputation. Assuming that time usage scales linearly with the number of target variants (Figure 3A), 4 million variants can be evaluated in approximately 1,250 s, which is less than half an hour. In other terms, secure evaluation is approximately 312 μs. per variant per 1,000 individuals $(25\ \text{s} \times 1,000\ \text{individuals})$. It can be decreased even further by scaling

(F). Precision-recall curve for rare variants (G). The boxplots illustrate the super-population-specific minor allele frequency distribution (y axis) for the common (top) and un-common variants (bottom)

(H). ALL indicates the MAF distribution for all populations. The center and the two ends of the boxplots show the median and 25%–75% values of the MAF distributions.

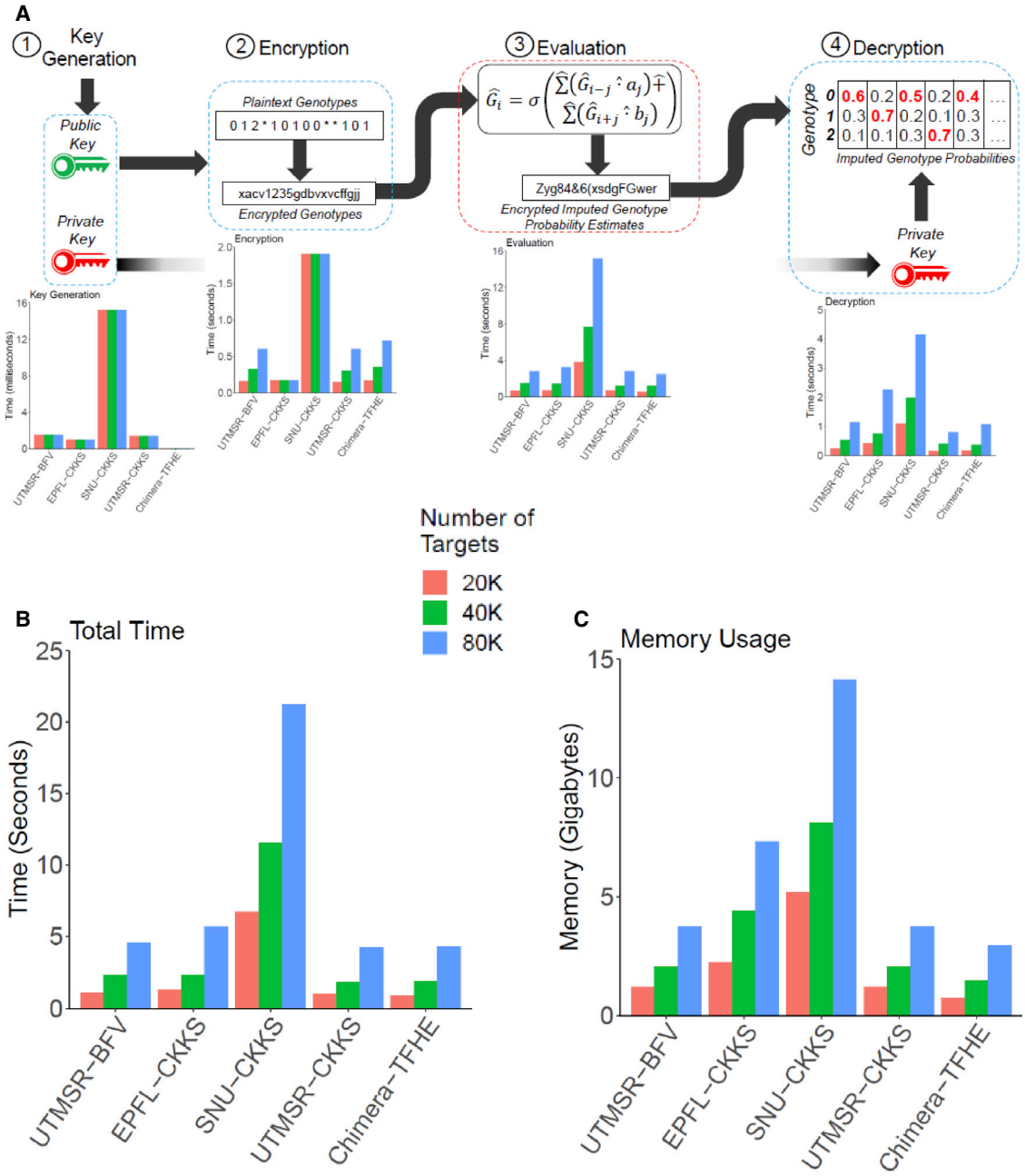


Figure 4. Memory and time requirements of the secure methods

(A–C) Each method is divided into 4 steps: (1) key generation, (2) encryption, (3) evaluation, and (4) decryption. The bar plots show the time requirements (A) using 20K, 40K, and 80K target variant sets. The aggregated time (B) and the maximum memory usage of the methods are also shown (C).

to a higher number of CPUs (i.e., cores on local machines or instances on cloud resources). In terms of memory usage, all methods required less than 15 gigabytes of main memory, and three of the five approaches required less than 5 gigabytes (Figure 4C). These results highlight the fact that secure methods could be deployed on even the commodity computer systems. The training of the methods on rare variants were performed to ensure the assigned scores are best tuned for the unbalanced training data in rare variants. The Chimera and SNU teams (best performing methods) have a diverse range of requirements for secure evaluation where the neural network approach (SNU) requires high resources, whereas the logistic regression approach has much more practicable resource requirements (Tables S9 and S10).

Resource usage comparison between secure and non-secure imputation methods

An important aspect of practicality is whether the methods are adaptable to different tag variants. This issue arises when a new array platform is used for genotyping tag variants with a

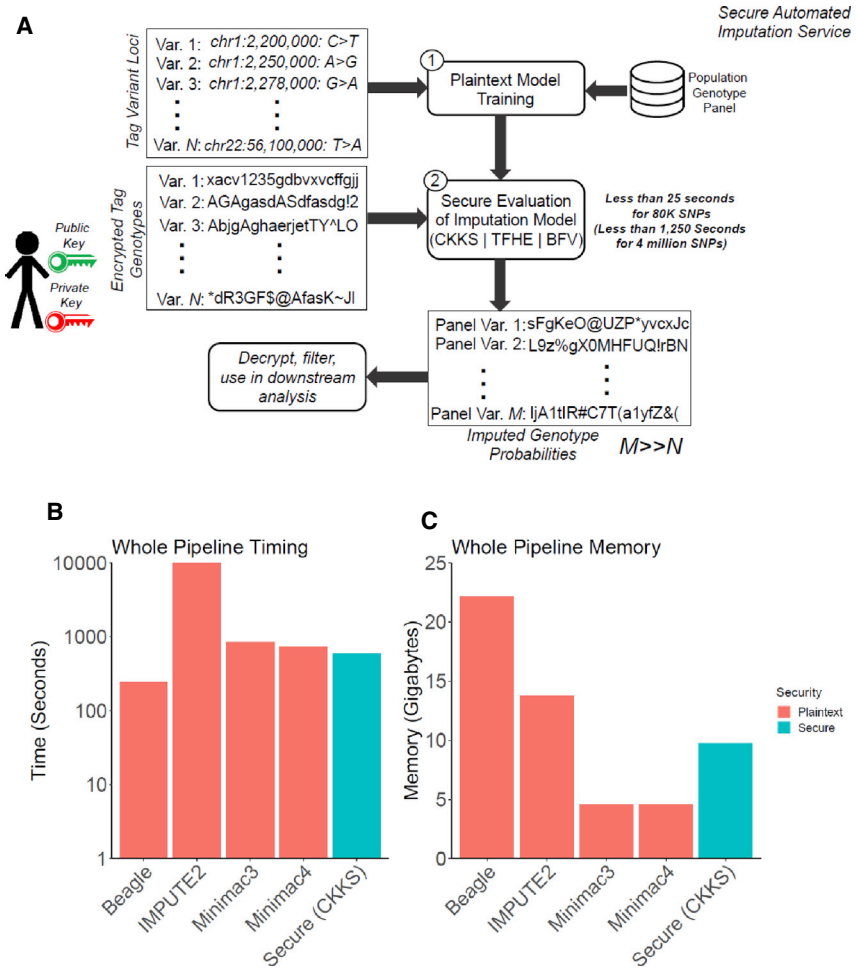


Figure 5. Comparison of time and memory requirements of methods

(A– C) The secure outsourced imputation service (A), time (B), and memory requirements (C) are illustrated in the bar plots where colors indicate security context. The y axis shows the time (in seconds) and main memory (in gigabytes) used by each method to perform the imputation of the 80K variants where the secure outsourced method includes the plaintext model training and secure model evaluation steps.

new set of tag variant loci. In this case, the current security framework requires that the plaintext models must be re-parametrized, and this may require a large amount of time and memory. To evaluate this, we optimized the linear models for the UTMSR-CKKS approach and measured the total time (training and evaluation) and the memory for the target variant set.

In order to make the comparisons fair with the HMM-based methods, we included the rare variants and common variants in this benchmark where the variants with MAF greater than 0.5% are used. In total, we used the 200,976 target variants in this range. In this way, we believe that we perform a fair comparison of resource usage with other non-secure methods. We assumed that the training and secure evaluation would be run sequentially, and we measured the time requirement of the secure approach by summing the time for key generation, encryption, secure evaluation, decryption, and the time for training. For memory, we computed the peak memory required for training and the peak memory required for secure evaluation. These time and memory requirements provided us with an estimate of the resources used by the secure pipeline (Figure 5A) that can be fairly compared with the non-secure methods.

We measured the time and memory requirements of all the methods by using a dedicated computer cluster to ensure resource requirements are measured accurately (see STAR Methods). For IMPUTE2, there was no option for specifying multiple threads. Hence, we divided the sequenced portion of chromosome 22 into 16 regions and imputed variants in each region in parallel using IMPUTE2, as instructed by the manual, i.e., we ran 16 IMPUTE2 instances in parallel to complete the computation. We then measured the total memory required by all 16 runs and used this as the memory requirement by IMPUTE2. We used the maximum time among all the 16 runs, as the time requirements by parallelized IMPUTE2. Beagle, Minimac3, and Minimac4 were run with 16 threads, as this option was available in the command line. In addition, Minimac4 requires model parametrization and preprocessing of the reference panel, which requires large CPU time. Therefore, we included this step in the timing requirements. Figures 5B and 5C show the time and memory requirements, respectively, of the three non-secure approaches and our secure method. The results show that the secure pipeline provides competitive timing (2nd fastest after Beagle) and memory requirements (3rd in terms of least usage after Minimac3 and Minimac4). Our results also show that Minimac3/Minimac4 and our secure approach provided a good trade-off between memory and timing, because Beagle and IMPUTE2 exhibit the highest time or highest memory requirements compared with other methods.

We also compared the secure models and found that different secure models exhibit diverse accuracy depending on allele frequency and position of variants (supplemental information).

DISCUSSION

We presented fully secure genotype imputation methods that can practically scale to genome-wide imputation tasks by using efficient HE techniques where the data are encrypted in transit, in analysis, and at rest. This is a unique aspect of the HE-based frameworks because, when appropriately performed, encryption is one of the few approaches that are recognized at the legislative level as a way of secure sharing of biomedical data, e.g., by HIPAA (Wilson, 2006) and partially by GDPR (Hoofnagle et al., 2019).

Our study was enabled by several key developments in the fields of genomics and computer science. First, the recent theoretical breakthroughs in the HE techniques have enabled

massive increases in the speed of secure algorithms. Although much of the data science community still regards HE as a theoretical and not-so-practical framework, the reality is far from this image. We hope that our study can provide a reference for the development of privacy-aware and fully secure approaches that employ HE. Second, the amount of genomic data have increased several orders of magnitude in recent years. This provides enormous genotype databases where we can train the imputation models and test them in detail before implementing them in secure evaluation frameworks. Another significant development is the recent formation of genomic privacy communities and alliances, i.e., Global Alliance for Genomic Health (GA4GH), where researchers build interdisciplinary approaches for developing privacy-aware methods. For example, our international study stemmed from the 2019 iDASH Genomic Privacy Challenge. We firmly believe that these communities will help bring together further interdisciplinary collaborations for the development of secure genomic analysis methods.

The presented imputation methods train an imputation model for each target variant. Our approach handles millions of models, i.e., parameters. Unlike the HMM models that can adapt seamlessly to a new set of tag variants (i.e., a new array platform), our approaches need to be retrained when the tag variants are updated. We expect that the training can be performed a-priori for a new genotyping array and that it can be reused in the imputation. The decoupling of the (1) plaintext training and (2) secure evaluation steps is very advantageous, because plaintext training can be independently performed at the third party without the need to wait for the data to arrive. This way, the users would have to accrue only the secure evaluation time, that is, as our results show, much smaller compared with the time requirements of the non-secure models, as small as 312 μs per variant per 1,000 individuals. Nevertheless, even with the training, our results show that the secure imputation framework can train and evaluate in run times comparable with plaintext (non-secure) methods. In the future, we expect many optimizations can be introduced to the models we presented. For example, we foresee that the linear model training can be replaced with more complex feature selection and training methods. Deep neural networks are potential candidates for imputation tasks, as they can be trained for learning the complex haplotype patterns to provide better imputation accuracy (Das et al., 2018). With the introduction of the graphical processing units (GPUs) on the cloud, these models can be trained and evaluated securely and efficiently. It is, however, important to be thorough about the security of the data because, as we mentioned before, even the number of untyped target variants that the researcher sends to the server can leak some information about the datasets. These stealthy leakages highlight the importance of using semantic security approaches. It is important to note that the secure evaluation steps implemented in our study replicate the results of the plaintext models almost exactly, which indicates that ''HE-conversion'' does not accrue any performance penalty.

Our study aims to spearhead the feasibility of secure genotype imputation in a high-throughput manner. As such, there are currently numerous limitations that must be overcome in future studies (supplemental information). For example, our approaches provide suboptimal accuracy when compared with non-secure methods, especially for rare variants. As we mentioned earlier, we foresee that our methods can be optimized in numerous ways. For instance, it has been previously shown that the vicinity-based methods can make use of tag single nucleotide polymorphism (SNP) selection to increase accuracy (Yu and Schaid, 2007). We are also foreseeing that new methods can be adapted on the hard-to-impute regions (Duan et al., 2013; Chen and Shi, 2019) to provide higher accuracy for these regions with complex haplotype structures.

Finally, we believe that the multitude of models and the secure evaluation approaches that we presented here can help provide a much needed reference point for the development and improvement of the imputation methods. Moreover, the developed models can be easily adapted to solve other privacy-sensitive problems by using secure linear, logistic, and network model evaluations, such as the secure rare variant association tests (Wu et al., 2011). Therefore, we believe that our codebases represent an essential resource for the computational genomics community. We have organized the codebases to ensure that they can be most accessible to the users without the necessary cryptography expertise. We are hoping that our codebase can provide a central role in the development of a community (similar to *dynverse* (dynverse, n.d.) or TAPE (TAPE, n.d. 2019; Rao et al., 2019) repositories for trajectory inference and protein embedding, respectively) where users can use the developed methods and datasets for uniform benchmarking of their new imputation methods.

STAR★METHODS

Detailed methods are provided in the online version of this paper and include the following:

- KEY RESOURCES TABLE
- RESOURCE AVAILABILITY
  - Lead contact
  - Materials availability
  - Variant and genotype datasets
  - Accuracy benchmark metrics
  - Micro-AUC accuracy statistics
  - Measurement of time and memory requirements
  - Secure methods
  - UTMSR-BFV and UTMSR-CKKS
  - Chimera-TFHE
  - EPFL-CKKS
  - SNU-CKKS
  - Non-secure methods
  - Beagle
  - IMPUTE2
  - Minimac3 and Minimac4

SUPPLEMENTAL INFORMATION

Supplemental information can be found online at https://doi.org/10.1016/j.cels.2021.07.010.

ACKNOWLEDGMENTS

Authors thank the National Human Genome Research Institute (NHGRI) of National Institutes of Health for providing funding and support for iDASH Genomic Privacy challenges (R13HG009072). We also thank Luyao Chen for providing

technical support to set up the computational environment for unified evaluation. X.J. is CPRIT Scholar in Cancer Research (RR180012), and he was supported in part by Christopher Sarofim Family Professorship, UT Stars award, UTHealth startup, the National Institutes of Health (NIH) under award number R13HG009072, R01GM114612, and the National Science Foundation (NSF) RAPID #2027790. L.O.-M. is supported by the NIH under award number R13HG009072 and R01GM114612 for this work. EPFL team is funded in part by the grant #2017-201 (DPPH) of the Swiss PHRT and by the grant #2018-522 (MedCo) of the Swiss PHRT and SPHN. I.C. has been supported in part by ERC Advanced Grant ERC-2015-AdG-IMPaCT, by the FWO under an Odysseus project GOH9718N and by the CyberSecurity Research Flanders with reference number VR20192203. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the ERC or FWO. D.K., S.H. and J.H.C. were supported by the Institute for Information & Communications Technology Promotion (IITP) Grant through the Korean Government (MSIT), (Development and Library Implementation of Fully Homomorphic Machine Learning Algorithms supporting Neural Network Learning over Encrypted Data), under Grant 2020-0-00840. M.K. was supported by the Settlement Research Fund (No. 1.200109.01) of UNIST (Ulsan National Institute of Science & Technology) and Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (No.2020-0-01336, Artificial Intelligence graduate school support [UNIST]).

AUTHOR CONTRIBUTIONS

M.K., A.O.H., and X.J. designed the imputation scenario, implemented the evaluation metrics, conducted the benchmarking experiments with the baseline methods, and drafted the manuscript. M.K., A.O.H., X.J., Y. Song, and K.L. designed the baseline methods for UTMSR imputation pipelines. I.C., S.C., M.G., and N.G. trained and implemented the Chimera-TFHE pipeline and contributed the results to the manuscript. D.K., W.C., S.H., Y. Son, and J.H.C. trained and implemented the SNU-CKKS pipeline. Y.M., J.T.-P., D.F., J.-P.B., and J.-P.H. trained and implemented the EPFL-CKKS pipeline. H.S., L.O.-M., and X.J. oversaw the iDASH19 challenge, conceived the study, and edited the manuscript. All authors have read and approved the final manuscript.

DECLARATION OF INTERESTS

The authors declare no competing interests.

Received: August 25, 2020
Revised: April 21, 2021
Accepted: July 29, 2021
Published: August 30, 2021

REFERENCES

1000 Genomes Project Consortium (2015). A global reference for human genetic variation. Nature *526*, 68–74.

Agarwala, V., Flannick, J., Sunyaev, S.; GoT2D Consortium, and Altshuler, D. (2013). Evaluating empirical bounds on complex disease genetic architecture. Nat. Genet. *45*, 1418–1427.

Albrecht, M., Chase, M., Chen, H., Ding, J., Goldwasser, S., Gorbunov, S., Halevi, S., Hoffstein, J., Laine, K., Lauter, K., et al. (2018). Homomorphic encryption security standard. Technical report. https://homomorphicencryption.org/standard/.

Albrecht, M.R., Player, R., and Scott, S. (2015). On the concrete hardness of learning with errors. J. Math. Cryptol. *9*, 169–203. http://www.degruyter.com/view/j/jmc.2015.9.issue-3/jmc-2015-0016/jmc-2015-0016.xml.

Belmont, J.W., Hardenbol, P., Willis, T.D., Yu, F., Yang, H., Ch'Ang, L.Y., Huang, W., Liu, B., Shen, Y., Tam, P.K.H., et al. (2003). The international hapmap project. Nature *426*, 789–796.

Berger, B., and Cho, H. (2019). Emerging technologies towards enhancing privacy in genomic data sharing. Genome Biol. *20*, 128.

Bomba, L., Walter, K., and Soranzo, N. (2017). The impact of rare and low-frequency genetic variants in common disease. Genome Biol. *18*, 77.

Boura, C., Gama, N., Georgieva, M., and Jetchev, D. (2018). Chimera: combining ring-lwe-based fully homomorphic encryption schemes, Technical report, Cryptology eprint Archive, Report 2018/758. https://eprint.iacr.org/2018/758.

Brakerski, Z. (2012). Fully homomorphic encryption without modulus switching from classical GapSVP. In Advances in Cryptology – Crypto 2012, R. Safavi-Naini and R. Canetti, eds. (Springer), pp. 868–886.

Browning, B.L., Zhou, Y., and Browning, S.R. (2018). A one-penny imputed genome from next-generation reference panels. Am. J. Hum. Genet. *103*, 338–348.

Chen, J., Harmanci, A.S., and Harmanci, A.O. (2019). Detecting and annotating rare variants. Encyclopedia of Bioinformatics and Computational Biology *3*, 388–399.

Chen, J., and Shi, X. (2019). Sparse convolutional denoising autoencoders for genotype imputation. Genes *10*, 652. https://doi.org/10.3390/genes10090652.

Cheon, J.H., Kim, A., Kim, M., and Song, Y. (2017). Homomorphic encryption for arithmetic of approximate numbers. In International Conference on the Theory and Application of Cryptology and Information Security (Springer), pp. 409–437.

Chillotti, I., Gama, N., Georgieva, M., and Izabachène, M. (2020). TFHE: fast fully homomorphic encryption over the torus. J. Cryptol. *33*, 34–91.

Chisholm, J., Caulfield, M., Parker, M., Davies, J., and Palin, M. (2013). Briefing - Genomics England and the 100K Genome Project. Genomics Engl.

Cho, H., Wu, D.J., and Berger, B. (2018). Secure genome-wide association analysis using multiparty computation. Nat. Biotechnol. *36*, 547–551.

Cooper, J.D., Smyth, D.J., Smiles, A.M., Plagnol, V., Walker, N.M., Allen, J.E., Downes, K., Barrett, J.C., Healy, B.C., Mychaleckyj, J.C., et al. (2008). Meta-analysis of genome-wide association study data identifies additional type 1 diabetes risk loci. Nat. Genet. *40*, 1399–1401.

Das, S., Abecasis, G.R., and Browning, B.L. (2018). Genotype imputation from large reference panels. Annu. Rev. Genomics Hum. Genet. *19*, 73–96.

Das, S., Forer, L., Schönherr, S., Sidore, C., Locke, A.E., Kwong, A., Vrieze, S.I., Chew, E.Y., Levy, S., McGue, M., et al. (2016). Next-generation genotype imputation service and methods. Nat. Genet. *48*, 1284–1287.

Depristo, M.A., Banks, E., Poplin, R., Garimella, K.V., Maguire, J.R., Hartl, C., Philippakis, A.A., Del Angel, G., Rivas, M.A., Hanna, M., et al. (2011). A framework for variation discovery and genotyping using next-generation DNA sequencing data. Nat. Genet. *43*, 491–498.

Dowlin, N., Gilad-Bachrach, R., Laine, K., Lauter, K., Naehrig, M., and Wernsing, J. (2017). Manual for using homomorphic encryption for bioinformatics. Proc. IEEE *105*, 552–567.

Duan, Q., Liu, E.Y., Croteau-Chonka, D.C., Mohlke, K.L., and Li, Y. (2013). A comprehensive SNP and indel imputability database. Bioinformatics *29*, 528–531. https://doi.org/10.1093/bioinformatics/bts724.

Evangelou, E., and Ioannidis, J.P. (2013). Meta-analysis methods for genome-wide association studies and beyond. Nat. Rev. Genet. *14*, 379–389.

Fan, J., and Vercauteren, F. (2012). Somewhat practical fully homomorphic encryption. IACR Cryptol. Eprint Arch. *2012*, 144.

Gangan, S. (2015). A review of man-in-the-middle attacks. arXiv http://arxiv.org/abs/1504.02115.

Gentry, C. (2009). Fully homomorphic encryption using ideal lattices. In Proceedings of the Forty-first Annual ACM Symposium on Theory of Computing, STOC '09, pp. 169–178. http://doi.acm.org/10.1145/1536414.1536440.

Gibson, G. (2012). Rare and common variants: twenty arguments. Nat. Rev. Genet. *13*, 135–145.

dynverse. dynverse: benchmarking, constructing and interpreting single-cell trajectories. https://github.com/dynverse.

Goldfeder, R.L., Wall, D.P., Khoury, M.J., Ioannidis, J.P.A., and Ashley, E.A. (2017). Human genome sequencing at the population scale: a primer on

high-throughput DNA sequencing and analysis. Am. J. Epidemiol. *186*, 1000–1009.

Heather, J.M., and Chain, B. (2016). The sequence of sequencers: the history of sequencing DNA. Genomics *107*, 1–8.

HES (2020). Homomorphic encryption standardization (HES). https://homomorphicencryption.org.

Hoffmann, T.J., Kvale, M.N., Hesselson, S.E., Zhan, Y., Aquino, C., Cao, Y., Cawley, S., Chung, E., Connell, S., Eshragh, J., et al. (2011). Next generation genome-wide association tool: design and coverage of a high-throughput European-optimized SNP array. Genomics *98*, 79–89.

Hoofnagle, C.J., van der Sloot, B., and Borgesius, F.Z. (2019). The European Union general data protection regulation: what it is and what it means. Inf. Commun. Technol. Law *28*, 65–98.

Howie, B., Fuchsberger, C., Stephens, M., Marchini, J., and Abecasis, G.R. (2012). Fast and accurate genotype imputation in genome-wide association studies through pre-phasing. Nat. Genet. *44*, 955–959.

Howie, B., Marchini, J., and Stephens, M. (2011). Genotype imputation with thousands of genomes. G3 (Bethesda) *1*, 457–470.

Howie, B.N., Donnelly, P., and Marchini, J. (2009). A flexible and accurate genotype imputation method for the next generation of genome-wide association studies. PLoS Genet. *5*, e1000529.

Johnson, E.O., Hancock, D.B., Levy, J.L., Gaddis, N.C., Saccone, N.L., Bierut, L.J., and Page, G.P. (2013). Imputation across genotyping arrays for genome-wide association studies: assessment of bias and a correction strategy. Hum. Genet. *132*, 509–522.

Kockan, C., Zhu, K., Dokmai, N., Karpov, N., Kulekci, M.O., Woodruff, D.P., and Sahinalp, S.C. (2020). Sketching algorithms for genomic data analysis and querying in a secure enclave. Nat. Methods *17*, 295–301.

Kowalski, M.H., Qian, H., Hou, Z., Rosen, J.D., Tapia, A.L., Shan, Y., Jain, D., Argos, M., Arnett, D.K., Avery, C., et al. (2019). Use of >100,000 NHLBI Trans-Omics for Precision Medicine (TOPMed) Consortium whole genome sequences improves imputation quality and detection of rare variant associations in admixed African and Hispanic/Latino populations. PLoS Genet. *15*, e1008500.

Lango Allen, H.L., Estrada, K., Lettre, G., Berndt, S.I., Weedon, M.N., Rivadeneira, F., Willer, C.J., Jackson, A.U., Vedantam, S., Raychaudhuri, S., et al. (2010). Hundreds of variants clustered in genomic loci and biological pathways affect human height. Nature *467*, 832–838.

Locke, A.E., Kahali, B., Berndt, S.I., Justice, A.E., Pers, T.H., Day, F.R., Powell, C., Vedantam, S., Buchkovich, M.L., Yang, J., et al. (2015). Genetic studies of body mass index yield new insights for obesity biology. Nature *518*, 197–206.

Loh, P.R., Danecek, P., Palamara, P.F., Fuchsberger, C., A Reshef, Y., K Finucane, H., Schoenherr, S., Forer, L., McCarthy, S., Abecasis, G.R., et al. (2016). Reference-based phasing using the Haplotype Reference Consortium panel. Nat. Genet. *48*, 1443–1448.

Lyubashevsky, V., Peikert, C., and Regev, O. (2010). On ideal lattices and learning with errors over rings. Journal of the ACM *60*, 1–25.

Marchini, J., and Howie, B. (2010). Genotype imputation for genome-wide association studies. Nat. Rev. Genet. *11*, 499–511.

Naveed, M., Ayday, E., Clayton, E.W., Fellay, J., Gunter, C.A., Hubaux, J.P., Malin, B.A., and Wang, X. (2015). Privacy in the genomic era. ACM Comput. Surv. *48*, 1–44.

Ng, P.C., and Kirkness, E.F. (2010). Whole genome sequencing. In Genetic Variation (Springer), pp. 215–226.

Nissenbaum, H. (2009). Privacy in Context: Technology, Policy, and the Integrity of Social Life (Stanford University Press).

Nyholt, D.R., Yu, C.E., and Visscher, P.M. (2009). On Jim Watson's APOE status: genetic information is hard to hide. Eur. J. Hum. Genet. *17*, 147–149.

Rao, R., Bhattacharya, N., Thomas, N., Duan, Y., Chen, X., Canny, J., Abbeel, P., and Song, Y.S. (2019). Evaluating protein transfer learning with TAPE. bioRxiv. https://doi.org/10.1101/676825.

Rehm, H.L. (2017). Evolving health care through personal genomics. Nat. Rev. Genet. *18*, 259–267.

Schaid, D.J., Chen, W., and Larson, N.B. (2018). From genome-wide associations to candidate causal variants by statistical fine-mapping. Nat. Rev. Genet. *19*, 491–504.

Schwarze, K., Buchanan, J., Taylor, J.C., and Wordsworth, S. (2018). Are whole-exome and whole-genome sequencing approaches cost-effective? A systematic review of the literature. Genet. Med. *20*, 1122–1130.

Shendure, J., Balasubramanian, S., Church, G.M., Gilbert, W., Rogers, J., Schloss, J.A., and Waterston, R.H. (2017). DNA sequencing at 40: past, present and future. Nature *550*, 345–353.

Stram, D.O. (2004). Tag SNP selection for association studies. Genet. Epidemiol. *27*, 365–374.

Sung, Y.J., Winkler, T.W., de las Fuentes, L., Bentley, A.R., Brown, M.R., Kraja, A.T., Schwander, K., Ntalla, I., Guo, X., Franceschini, N., et al. (2018). A large-scale multi-ancestry genome-wide study accounting for smoking behavior identifies multiple significant loci for blood pressure. Am. J. Hum. Genet. *102*, 375–400.

Taliun, D., Harris, D.N., Kessler, M.D., Carlson, J., Szpiech, Z.A., Torres, R., Taliun, S.A.G., Corvelo, A., Gogarten, S.M., Kang, H.M., et al. (2019). Sequencing of 53,831 diverse genomes from the NHLBI TOPMed Program. bioRxiv. https://doi.org/10.1101/563866.

TAPE. (2019). Tasks assessing protein embeddings (TAPE), a set of five biologically relevant semi-supervised learning tasks spread across different domains of protein biology. https://github.com/songlab-cal/tape.

Tam, V., Patel, N., Turcotte, M., Bossé, Y., Paré, G., and Meyre, D. (2019). Benefits and limitations of genome-wide association studies. Nat. Rev. Genet. *20*, 467–484.

TOPMed. (2016). NHLBI trans-omics for precision medicine whole genome sequencing program. https://www.nhlbiwgs.org/.

Wilson, J.F. (2006). Health Insurance Portability and Accountability Act privacy rule causes ongoing concerns among clinicians and researchers. Ann. Intern. Med. *145*, 313–316.

Wu, M.C., Lee, S., Cai, T., Li, Y., Boehnke, M., and Lin, X. (2011). Rare-variant association testing for sequencing data with the sequence kernel association test. Am. J. Hum. Genet. *89*, 82–93.

Yu, Z., and Schaid, D.J. (2007). Methods to impute missing genotypes for population data. Hum. Genet. *122*, 495–504.

STAR★METHODS

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Deposited data		
The chromosome 22 genotype calls the 2,504 individuals in The 1000 Genomes Project's 3rd phase.	The 1000 Genomes Consortium	ftp://ftp-trace.ncbi.nih.gov/1000genomes/ftp/release/20130502/ALL.chr22.phase3_shapeit2_mvncall_integrated_v5a.20130502.genotypes.vcf.gz
Illumina Duo version 3 genotyping array documentation	Illumina Inc. Web Site	https://support.illumina.com/downloads/human1m-duo_v3-0_product_files.html
Source Data for Figures 1, 2, 3, 4, and 5 and supplemental information	This work	https://doi.org/10.5281/zenodo.4947832
Software and algorithms		
R Statistical Computing Platform	The R Foundation	https://www.r-project.org/
Source Code and Documentation for Secure Imputation Models	This work	https://doi.org/10.5281/zenodo.4948000
Source Code for generating Figures 1, 2, 3, 4, and 5 and the Figures S1–S8	This work	https://doi.org/10.5281/zenodo.4947832

We present and describe the data sources, accuracy metrics, and non-secure imputation method parameters. The detailed methods are presented in the supplemental information.

RESOURCE AVAILABILITY

Lead contact

Further information and requests for resources and reagents should be directed to and will be fulfilled by the lead contact, Arif Harmanci (arif.o.harmanci@uth.tmc.edu).

Materials availability

This study did not generate new materials.

Variant and genotype datasets

All the tag and target variant loci, and the genotypes are collected from the public resources. We downloaded the Illumina Duo 1M version 3 variant loci from the array's specification at the Illumina web site (https://support.illumina.com/downloads/human1m-duo_v3-0_product_files.html). The file was parsed to extract the variants on chromosome 22, which yielded 17,777 variants. We did not use the CNVs and indels while filtering the variants and we focused only on the single nucleotide polymorphims (SNPs). We then intersected these variants with the 1000 Genomes variants on chromosome 22 to identify the array variants that are detected by the 1000 Genomes Project. We identified 16,184 variants from this intersection. This variant set represents the tag variants that are used to perform the imputation. The phased genotypes on chromosome 22 for the 2,504 individuals in the 1000 Genomes Project are downloaded from the NCBI portal (ftp://ftp-trace.ncbi.nih.gov/1000genomes/ftp/release/20130502/ALL.chr22.phase3_shapeit2_mvncall_integrated_v5a.20130502.genotypes.vcf.gz). We filtered out the variants for which the allele frequency reported by the 1000 Genomes Project is less than 5%. After excluding the tag variants on the array platform, we identified 83,072 target variants that are to be used for imputation. As the developed secure methods use vicinity variants, the variants at the ends of the chromosome are not imputed. We believe this is acceptable because these variants are located very close to the centromere and at the very end of the chromosome. After filtering the non-imputed variants, we focused on the 80,882 variants that were used for consistent benchmarking of all the secure and non-secure methods.

Accuracy benchmark metrics

We describe the genotype level and variant level accuracy. For each variant, we assign the genotype with the highest assigned genotype probability. The variant level accuracy is the average variant accuracy where each variant's accuracy is estimated based on how well these imputed genotypes of the individuals match the known genotypes:

$$\text{Variant Acc.} = \left(\frac{1}{\#\text{ of Variants}}\right) \times \sum_i \left(\frac{\#\text{ Correctly Imputed Individuals for Variant } i}{\#\text{ of Individuals for Variant } i}\right).$$

Variant level accuracy is also referred to as the macro-aggregated accuracy.

At the genotype level, we simply count the number of correctly computed genotypes and divide this with the total number of genotypes:

$$\text{Genotype Acc.} = \frac{\sum_i(\#\text{ Correctly Imputed Individuals for Variant } i)}{\sum_i(\#\text{ of Individuals for Variant } i)}.$$

In the sensitivity vs positive predictive value (PPV) plots, the sensitivity and PPV are computed after filtering the imputed genotypes with respect to the imputation probability. We compute the sensitivity at the probability cutoff of τ is:

$$\text{Sens}_\tau = \frac{\sum_i(\#\text{ Correctly Imputed Individuals for Variant } i \text{ whose genotype probability}>\tau)}{\sum_i(\#\text{ of Individuals for Variant } i)}$$

Positive predictive value measures the fraction of correctly imputed genotypes among the genotypes whose probability is above the cutoff threshold:

$$\text{PPV}_\tau = \frac{\sum_i(\#\text{ Correctly Imputed Individuals for Variant } i \text{ whose genotype probability}>\tau)}{\sum_i(\#\text{ Individuals for Variant } i \text{ whose genotype probability}>\tau)}$$

Next, we swept a large cutoff range for τ from -5 to 5 with steps 0.01. We finally plotted the sensitivity versus PPV to generate the precision-recall curves for each method.

Micro-AUC accuracy statistics

For parameterizing the accuracy and demonstrating how different parameters affect algorithm performance, we used micro-AUC as the accuracy metric. This was also the original accuracy metric for measuring the algorithm performance in iDASH19 competition. Micro-AUC treats the imputation problem as a three-level classification problem where each variant is "classified" into one of three classes, i.e., genotypes, $\{0, 1, 2\}$. Micro-AUC computes an AUC metric for each genotype then microaggregates the AUCs for all the genotypes. This enables assigning one score to a multi-class classification problem. We use the implementation in scikit-learn package to measure the micro-AUC scores for each method (https://scikit-learn.org/stable/modules/generated/sklearn.metrics.roc_auc_score.html).

Measurement of time and memory requirements

For consistently measuring the time and memory usage among all the benchmarked methods, we used /usr/bin/time -f %e "\t" %M" to report the wall time (in seconds) and peak memory usage (in kilobytes) of each method.

Secure methods

We briefly describe the secure methods.

UTMSR-BFV and UTMSR-CKKS

The UTMSR (UTHealth-Microsoft Research) team uses a linear model with the nearby tag variants as features for each target variant. The plaintext model training is performed using the GNU Scientific Library. The collinear features are removed by performing the SVD and removing features with singular values smaller than 0.01. The target variant genotype is modeled as a continuous variable that represents the "soft" estimate of the genotype (or the estimated dosage of the alternate allele) and can take any value from negative to positive infinity. The genotype probabilities are assigned by converting the soft genotype estimation to a score in the range [0,1]:

$$p(g) = \exp(-1 \times |\tilde{g} - g|), g \in \{0, 1, 2\}, \quad \text{(Equation 1)}$$

where g denotes one of the genotypes and $\tilde{g}$ represents the decrypted value of the imputed genotype estimate. Suppose that each variant genotype is modeled using genotypes of variants within k variant vicinity of the variant. In plaintext domain, the imputed value can be written as follows:

$$\tilde{g}_j = w_{j,0} + \sum_{r=1}^{k}\left(w_{j,r}^{-} \times g_{j-r}\right) + \sum_{r=1}^{k}\left(w_{j,r}^{+} \times g_{j+r}\right), \quad \text{(Equation 2)}$$

where $w_{j,0}$ is the intercept of the linear model, and $w_{j,r}^{-}$ and $w_{j,r}^{+}$ denote the linear model weights for the j^{th} target variant's r^{th} upstream and downstream tag variants, respectively.

The secure outsourcing imputation protocols are implemented on two popular ring-based HE cryptosystems – BFV (Brakerski, 2012; Fan and Vercauteren, 2012) and CKKS (Cheon et al., 2017). These HE schemes share the same parameter setup and key-generation phase but have different algorithms for message encoding and homomorphic operations. In a nutshell, a ciphertext is generated by adding a random encryption of zero to an encoded plaintext, which makes the ring-based HE schemes secure under the

RLWE assumption. More precisely, each tag variant is first encoded as a polynomial with its coefficients, and the encoded plaintext is encrypted into a ciphertext using the underlying HE scheme. The plaintext polynomial in the BFV scheme is separated from an error polynomial (inserted for security), whereas the plaintext polynomial in the CKKS scheme embraces the error. Then Equation 2 is homomorphically evaluated on the encrypted genotype data by using the plain weight parameters. We exploit parallel computation on multiple individual data, and hence it enables us to obtain the predicted genotype estimates over different samples at a time. Our experimental results indicate that the linear model with 32 tag variants as features for each target variant shows the most balanced performance in terms of timing and imputation accuracy in the current testing dataset (see Table S8 and Figure S5). Our protocols achieve at least a 128-bit security level from the HE standardization workshop paper (Albrecht et al., 2018). We defer the complete details to the "UTHealth-Microsoft Research team solution" section in the supplementary document.

Chimera-TFHE

The Chimera team used multi-class logistic regression (logreg) models trained over one-hot encoded tag features: each tag SNP variant is mapped to 3 Boolean variables. Chimera's model training and architecture performed the best (with respect to accuracy and resource requirement) among six other solutions in the iDASH2019 Genotype Imputation Challenge.

We build three models per target SNP (one model per variant), i.e., target SNPs are also one-hot-encoded. These models give the probabilities for each target SNP variant. The maximal probability variant is the imputed target SNP value. A fixed number d of the nearest tag SNPs (in relation to the current target SNP) are used in model building. We train the models with different values of d in order to study the influence of neighborhood size: from 5 to 50 neighbors with an increment of 5. The most accurate model, in terms of micro-AUC score, is obtained for a neighborhood size $d = 45$. The fastest model with an acceptable accuracy (micro $-$ AUC> 0.99) is obtained for $d = 10$. Although, the execution time of the fastest model is only ≈ 2 times faster compared to the most accurate model (refer to Table S2).

During the homomorphic evaluation, only the linear part of the logreg model is executed, which means in particular that we do not homomorphically apply the sigmoid function on the output scores. We use the coefficient packing strategy and pack as many plaintext values as possible in a single ciphertext. The maximum number of values that can be packed in a *RingLWE* ciphertext equals the used ring dimension, which is $n = 1024$ in our solution. We chose to pack one or several columns of the input (tag SNPs) into a single ciphertext. Since the TFHE library *RingLWE* ciphertexts encrypt polynomials with Torus ($\mathbb{T} = \mathbb{R}\mathrm{mod}1$) coefficients, we downscale the data to Torus values (multiples of 2^{-14}) and upscale the model coefficients to integers.

In our solution, we use linear combinations with public integer coefficients. The evaluation is based on the security of *LWE* and only the encryption phase uses *RingLWE* security notions with no additional bootstrapping or key-switching keys. The security parameters have been tuned to support binary keys. Of course, as neither bootstrapping nor key-switching is used in our solution, the key distribution can be changed to any distribution (including the full domain distribution) without any time penalty. Our scheme achieves 130 bits of security, according to the LWE estimator (Albrecht et al., 2015). More information about the our solution is described in the supplementary document ("Chimera-TFHE team solution").

EPFL-CKKS

EPFL uses a multinomial logistic regression model with $d - 1$ neighboring coefficients and 1 intercept variable for each target variant, with three classes {0,1,2}. The plaintext model is trained using the `scikit-learn` python library. The input variants are represented as values {0,1,2}. There is no pre-processing applied to the training data. For a target position j, the predicted probabilities for each class label are given by:

$$P\left[y = g \middle| z_r^{(p,j)}\right] = \frac{e^{w_0^{(\cdot,j,g)} + \sum_{r=1}^{d-1} w_r^{(\cdot,j,g)}}}{\sum_{i=0}^{2} e^{w_0^{(\cdot,j,i)} + \sum_{r=1}^{d-1} w_r^{(\cdot,j,i)} z_r^{(p,j,\cdot)}}}, \qquad \text{(Equation 3)}$$

where $\{w_0^{(\cdot,j,g)}, \ldots, w_{d-1}^{(\cdot,j,g)}\}$ are the trained regression coefficients for label $g \in \{0,1,2\}$ and position j, and $\{z_1^{(p,j,\cdot)}, \ldots, z_{d-1}^{(p,j,\cdot)}\}$ are the neighboring variants for patient p around target position j. The hard prediction for position j is given by $y^{(p,j,g)} = \mathrm{argmax}_g(P[y = g | z_r^{(p,j)}])$. The variants $\{z_1^{(p,j,\cdot)}, \ldots, z_{d-1}^{(p,j,\cdot)}\}$ are sent encrypted and packed to the server, using the CKKS homomorphic cryptosystem, and the exponents in Equation 3 are computed homomorphically. The client decrypts the result and can obtain the label probabilities and hard predictions for each position. For the prediction, we use several numbers of regression coefficients, ranging from 8 to 64; as this number increases, both the obtained accuracy and the computational complexity increase (see Table S6). We use a single parametrization of the cryptosystem (see the "EPFL-Lattigo team solution" section in the supplemental information) for all the regression sizes, which keeps the cipher expansion asymptotically constant. The security of this solution is based on the hardness of the RLWE problem with Gaussian secrets.

SNU-CKKS

The SNU team applies one-hidden layer neural network for the genotype imputation. The model is obtained from Tensorflow module in plain (unencrypted) state, and the inference phase is progressed in encrypted stated for given test SNP data encrypted by the CKKS HE scheme. We encode each ternary SNP data into a 3-dimensional binary vector, i.e., $0 \rightarrow (1,0,0)$, $1 \rightarrow (0,1,0)$ and $2 \rightarrow (0,0,1)$. For better performance in terms of both accuracy and speed, we utilize an inherent property that each target SNP is mostly

related by its adjacent tag SNPs. We set the number of the adjacent tag SNPs as a pre-determined parameter d, and run experiments on various choices of the parameter ($d = 8k$ for $1 \leq k \leq 9$). As a result, we check that $d = 40$ shows the best accuracy in terms of micro-AUC. Since the running time of computing genotype score grows linear to d, the fastest result is obtained at $d = 8$. We refer the intermediate value $d = 24$ to the most balanced choice in terms of accuracy and speed.

The security of the utilized CKKS scheme relies on the hardness of solving the RLWE problem with ternary (signed binary) secret. For the security estimation, we applied the LWE estimator (Albrecht et al., 2015), a sage module that computes the computational costs of state-of-art (R)LWE attack algorithms. The script for the security estimation is attached as a figure in the "SNU team solution" section in the supplementary document.

Non-secure methods

We describe the versions and the details of how the non-secure methods were run. The benchmarks were performed on a Linux workstation with 769 Gigabytes of main memory on an Intel Xeon Platinum 8168 CPU at 2.7 GHz with 96 cores. No other tools were run in the course of benchmarks.

Beagle

We obtained the jar formatted Java executable file for Beagle version 5.1 from the Beagle web site. The population panel (1,500 individuals) and the testing panel data are converted into VCF file format as required by Beagle. We ran Beagle using the chromosome 22 maps provided from the web site. The number of threads is specified as 16 threads at the command line (option 'nthreads=16'). We set the 'gp' and 'ap' flags in the command line to explicitly ask Beagle to save genotype probabilities that are used for building the sensitivity versus PPV curves. Beagle supplies the per genotype probabilities for each imputed variant. These probabilities were used in plotting the curves.

IMPUTE2

IMPUTE2 is downloaded from the IMPUTE2 website. The haplotype, legend, genotype, and the population panels are converted into specific formats that are required by IMPUTE2. We could not find a command line option to run IMPUTE2 with multiple threads. To be fair, we divided the sequenced portion of the chromosome 22 (from 16,000,000 to 51,000,000 base pairs) into 16 equally spaced regions of length 2.333 megabases. Next, we ran 16 different IMPUTE2 instances in parallel, as described in the IMPUTE2 manual. The output from the 16 runs is pooled to evaluate the imputation accuracy of IMPUTE2. IMPUTE2 provides per genotype probabilities, which were used for plotting the precision-recall curves.

Minimac3 and Minimac4

Minimac3 and Minimac4 are downloaded from the University of Michigan web site. We next downloaded Eagle 2.4.1 phasing software for phasing input genotypes. "Eagle+Minimac3" and "Eagle+Minimac4" were used in the Michigan Imputation Server's pipeline that is served for the public use. The panels are converted into indexed VCF files as required by Eagle, Minimac3, and Minimac4. We first used the Eagle protocol to phase the input genotypes. The phased genotypes are supplied to Minimac3 and Minimac4, and final imputations are performed. Eagle, Minimac3, and Minimac4 were run with 16 threads using the command line options ('–numThreads=16' and '–cpus 16' options for Eagle and Minimac3, respectively). Minimac3 and Minimac4 reports an estimated dosage of the alternate allele, which we converted to a score as in the above equation for UTMSR's scoring.

Minimac4 algorithm requires a preprocessing of the reference haplotype with a parameter estimation step. We observed that the parameter estimation step add a substantial amount of processing time and Minimac4 requires the parameter estimates to perform imputation.

Data and code availability

- Source data statement. Accuracy and resource benchmarking related source data have been deposited at https://doi.org/10.5281/zenodo.4947832. The 1000 Genomes project dataset are publicly available from NCBI portal at NCBI:ftp://ftp-trace.ncbi.nih.gov/1000genomes/ftp/release/20130502/ALL.chr22.phase3_shapeit2_mvncall_integrated_v5a.20130502.genotypes.vcf.gz). The Illumino array platform metadata is available from https://support.illumina.com/downloads/human1m-duo_v3-0_product_files.html.
- Code statement. The original source code, documentation, and usage examples for the imputation models are deposited at github: https://github.com/K-miran/secure-imputation and are also archived and deposited at https://doi.org/10.5281/zenodo.4948000.
- Scripts statement. The source and scripts for generating the figures and associated instructions are archived and deposited under https://doi.org/10.5281/zenodo.4947832 and are co-located with the figure-related datasets.
- Any additional information required to reproduce this work is available from the lead contact.