Efficient Secure Outsourcing of Genome-wide Association Studies

Wenjie Lu
Department of Computer Science
University of Tsukuba
Ten'nohdai 1-1-1, Tsukuba, Japan
Email: riku@mdl.cs.tsukuba.ac.jp

Yoshiji Yamada
Life Science Research Center,
Mie University
Kurimamachiya-cho 1577, Tsu, Japan
Email: yamada@gene.mie-u.ac.jp

Jun Sakuma
Department of Computer Science
University of Tsukuba / JST CREST
Ten'nohdai 1-1-1, Tsukuba, Japan
Email: jun@cs.tsukuba.ac.jp

***Abstract*—A genome-wide association study aimed at finding genetic variations associated with a particular disease is a common approach used in genetic epidemiology. We present a new efficient secure outsourcing computation of GWAS using homomorphic encryption based on ring-LWE. Our method works by virtue of the fact that integer vectors can be packed into a single ciphertext and a scalar product of integer vectors can be evaluated using a single homomorphic multiplication. We demonstrate by experimentation that secure outsourcing computation of a χ^2 test for independence with $5,000$ samples can be processed in one second including communication time, which is 250 times faster than an existing FHE solution.**

I. Introduction

A genome-wide association study (GWAS) aimed at identifying genetic variations associated with a particular disease is a common approach used in genetic epidemiology. Because of recent advances of DNA sequencing technologies, the cost of DNA sequencers is dropping rapidly. Actually, DNA sequencers are expected to be used widely in clinical/biological laboratories and hospitals in the near future. Personal genome information will be produced continually and ubiquitously. Related management methodologies incorporating large-scale personal genome data are currently underway. Outsourced analysis of genomic data in a cloud environment can be a promising solution to many related difficulties, but rigorous privacy protection is necessary, especially when the cloud environment is not trusted.

The objective of this study is to introduce a protocol for secure outsourced analysis of large-scale genome data for genetic epidemiology. For our secure outsourcing of GWAS, we consider three stakeholders, *data holders*, *researchers*, and the *cloud*. The data holders (hospitals, research institutes, or subjects) provide private genetic or clinical data to the cloud. A researcher is an entity that wishes to conduct a GWAS. The cloud is an untrusted entity that provides the researcher and data holders with computational resources such as computation power and storage. Our secure outsourcing of GWAS is processed as follows. We suppose that the public key of the researcher is distributed to all data holders before protocol execution using some key exchange protocol. All private information of the data holders is encrypted and submitted to the cloud through a secure channel; the data holders can go offline after data contribution. After the data collection phase, the cloud processes calculations for the GWAS independently with the encrypted data at the researcher's request. The researcher finally obtains the GWAS result.

Genetic variations (typically, single nucleotide polymorphisms, SNPs) that are significantly associated with a target disease are identified using statistical hypothesis testing. Kamm et al.[1] presented a secure multiparty computation of the χ^2 test for independence for case–control studies. Their protocol is based on secret-sharing and assumes presence of multiple cloud servers for secure outsourcing. Lauter et al.[2] introduced a secure outsourcing method of statistical tests for clinical epidemiology, including the Hardy–Weinberg equilibrium, χ^2 test for independence, and the Cochran–Armitage test for trend. Their method is based on a homomorphic encryption scheme aimed at outsourcing to a single cloud server.

In this manuscript, we present a new efficient secure outsourcing of GWAS in terms of communication and computation based on a homomorphic encryption scheme. The effectiveness of our proposed method is demonstrated with artificial datasets and a real dataset collected for a case–control study for genetic epidemiology[3]. We compare our method with that described by Lauter et al. using experimentation.

II. Genome-wide Association Studies

A case–control study is a common approach used in genetic epidemiology. It compares two groups of subjects: a case group that includes subjects with a target disease, and a control group that includes non-diseased subjects. Intuitively, if an allele is found at a specific marker locus more often in the case group than the control group, then the SNP is suspected to be associated with the target disease. Genetic association is typically investigated using the χ^2 test for independence. We first consider a biallelic locus with allele A and a. The left side of Table I presents the frequencies of alleles for the case and control groups at the marker locus. The association between the target diseases and genetic variants is investigated by testing the independence. If no mutual association exists between them (*null hypothesis*), then the observed frequencies are expected to be consistent with the expected frequencies, as shown on the right side of Table I. The significance of association is evaluated with a p-value, which is obtained from the χ^2 statistic with 1 degree of freedom as

$$\chi_c^2 = \sum_{k=1}^{4} \frac{(o_k - e_k)^2}{e_k}. \quad (1)$$

DOI 10.1109/SPW.2015.11

TABLE I. OBSERVED AND EXPECTED ALLELE FREQUENCY IN A CASE–CONTROL STUDY.

	observed		expected		
	a	A	a	A	total
case	$o_1 = 2r_0 + r_1$	$o_2 = 2r_2 + r_1$	$e_1 = \frac{(2n_0+n_1)2r}{2n}$	$e_2 = \frac{(2n_2+n_1)2r}{2n}$	$2r$
control	$o_3 = 2s_0 + s_1$	$0_4 = 2s_2 + s_1$	$e_3 = \frac{(2n_0+n_1)2s}{2n}$	$e_1 = \frac{(2n_2+n_1)2s}{2n}$	$2s$
total	$2n_0 + n_1$	$2n_2 + n_1$			$2n$

The p-value is obtainable by comparison of χ_c^2 to a χ^2 distribution. Our method can be extended readily to other types of hypothesis testing with slight modification of the encoding scheme. However, because of space limitations, we describe our method specifically for the χ^2 test for independence with allele frequencies.

III. HOMOMORPHIC ENCRYPTION

We introduce a homomorphic encryption scheme based on the ring-LWE assumption. Letting n be the lattice dimension of the scheme, where n is given as an integer of 2-power, then the message space is given as $R_t := \mathbb{Z}_t[x]/\langle x^n + 1\rangle$, where t is a prime. For our implementation, we used HElib [4], which implements the Brakerski–Gentry–Vaikuntanathan (BGV) scheme[5]. The BGV scheme supports evaluation of L-level arithmetic circuits. Presuming that $\mathrm{E}_{pk}(m_1)$, $\mathrm{E}_{pk}(m_2)$ are ciphertexts encrypted by BGV's scheme, then the homomorphic properties are

$$\mathrm{D}_{sk}\left(\mathrm{E}_{pk}(m_1) \oplus \mathrm{E}_{pk}(m_2)\right) = m_1 + m_2 \mod (x^n + 1, t)$$
$$\mathrm{D}_{sk}\left(\mathrm{E}_{pk}(m_1) \otimes \mathrm{E}_{pk}(m_2)\right) = m_1 \times m_2 \mod (x^n + 1, t),$$

where $\mathrm{D}_{sk}(\cdot)$ is decryption with corresponding secret key sk.

The BGV scheme is secure and correct under the ring-LWE assumption, as presented in [6]

$$n > \frac{(L(\log n + 23) - 8.5)(\kappa + 110)}{7.2}. \tag{2}$$

Therein, L stands for the level parameter; κ denotes a security level. If lattice dimension n and level parameter L follow eq. 2, then κ-bit security is guaranteed.

IV. SECURE OUTSOURCING OF GWAS

A. Problem Setting

The genotypes and phenotypes (disease status) of the subjects are distributed over two or more data holders. The researcher wishes to conduct statistical testing with the subjects' genotypes and phenotypes. The cloud provides researchers and data holders with computational resources such as computation power and storage.

In our settings, the researcher uses key-exchange protocol to share his public key with the data holders. Data holders need to use secure channel (e.g. SSH) to upload their encrypted data to the cloud. We suppose all stakeholders agree with a fixed database schema and subject identities so that the cloud can correctly merge data collected from data holders. All subjects are anonymously numbered by unique identities (subject IDs) and all the entities collaboratively utilize the subject IDs. In addition, we assume that all entities behave *semi-honestly*. If the cloud server is in collusion with the researcher, the researcher can obtain the subjects' raw genetic information. We therefore need a stronger assumption that the cloud is not in collusion with the researcher; the cloud only responds to specified queries (e.g. χ^2 statistics).

The objective of secure outsourcing of GWAS is to conduct hypothesis testing efficiently with a guarantee of security. In typical GWAS, genetic variants that occur at 5% or greater in a population are targeted. The number of such variants is on the order of three hundred thousand. Consequently, our goal is to process three hundred thousand SNPs with thousands of subjects with a single cloud server during a reasonable computation period under the security model described above. More specifically, throughout the process of outsoruicng, the cloud learns nothing except the number of the subjects and attributes; the researcher learns nothing except the contingency table described in Figure I.

B. Data Encoding

Presuming a case–control study in which genotypes of a SNP locus and disease status (disease or non-disease) of a target disease are collected from M subjects, then, fortunately, genetic association can be tested independently for each SNP. Therefore, the process of hypothetical testing for a single SNP by case–control study is described in the following.

Let A and a be the alleles of the biallelic locus. Consequently, the genotype at the locus is either AA, Aa, or aa. We represent the allele frequency as a M-dimension integer vector $\boldsymbol{x}^A$. Here, x_i^A, the i-th element of $\boldsymbol{x}^A$, represents the frequency of allele A at the marker locus: $x_i^A = 2, 1$, and 0, respectively, for AA, Aa, and aa. Presuming that the disease status of each subject is represented by a binary variable, then "disease" is represented by 1 (case); "non-disease" is represented by 0 (control). The status "disease" for all the subjects is therefore represented by the M dimension binary vector $\boldsymbol{y}^{case}$.

Using this representation, the frequencies in Table I are evaluated by the scalar product of the vectors. The frequencies of A and a are evaluated respectively by $\boldsymbol{x}^A \cdot \mathbf{1}$ and $2n - \boldsymbol{x}^A \cdot \mathbf{1}$, where $\mathbf{1}$ is the vector of which the elements are 1. The frequency of A in the case and control group are given respectively as $\boldsymbol{x}^A \cdot \boldsymbol{y}^{case}$ and $\boldsymbol{x}^A \cdot (\mathbf{1} - \boldsymbol{y}^{case})$. The remaining frequencies are also derived from simple calculations.

C. Packing for Homomorphic Encryption

BGV's encryption scheme takes polynomials as plaintexts. To encrypt vectors representing genotypes and phenotypes, vectors are transformed into polynomials [2], [7]. The transformation introduced by Yasuda et al. [7] was designed originally for secure Hamming distance evaluation of binary vectors. In their method, a binary vector with multiple elements can be packed into a polynomial. Presuming that $\boldsymbol{u}$ is a binary vector with length ℓ and that $\boldsymbol{v}$ is a binary vector with length m ($m \leq \ell < n$), then the two vectors are transformed into

polynomials in two ways.

$$\rho_{fw}(\boldsymbol{u}) := \sum_{i=0}^{\ell-1} u_i x^i \in R_t, \rho_{bw}(\boldsymbol{v}) := -\sum_{j=0}^{m-1} v_j x^{n-j} \in R_t$$

In those equations, u_i is the i-th element of $\boldsymbol{u}$; v_j is the j-th element of $\boldsymbol{v}$. We respectively designate the former and latter as transformation *forward* and *backward* packing. One virtue of this packing is that homomorphic multiplication of the ciphertexts with packing leads to a scalar product

$$\begin{aligned}\mathrm{E}_{pk}(\rho_{fw}(\boldsymbol{u})) \otimes \mathrm{E}_{pk}(\rho_{bw}(\boldsymbol{v})) =& \mathrm{E}_{pk}(\sum_{i=0}^{\ell-1} u_i x^i \times -\sum_{j=0}^{m-1} v_j x^{n-j}) \\ =& \mathrm{E}_{pk}(\sum_{h=0}^{\ell-m} \left\langle \boldsymbol{u}_{(h,m)}, \boldsymbol{v} \right\rangle x^h + \mathcal{U}).\end{aligned}$$

Therein, $\boldsymbol{u}_{(h,m)}$ denotes the subvector of $\boldsymbol{u}$ with size m starting from the h-th element of $\boldsymbol{u}$. Also, $\mathcal{U}$ denotes terms with degree deg. $> \ell - m$. If $m = \ell$, then the constant term of the resulting polynomial corresponds to scalar product $\boldsymbol{v} \cdot \boldsymbol{u}$. This evaluation is efficient because the computation of scalar product is done only through a single homomorphic multiplication.

D. Scalar Product with Forward–backward Packing

For our secure outsourcing GWAS described above, we use the immediate scalar product evaluation via the Forward–backward packing, which works not only with binary vectors but also with integer vectors. Given $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{Z}_p^d$, the constant term of $\mathrm{E}_{pk}(\rho_{fw}(\boldsymbol{u})) \otimes \mathrm{E}_{pk}(\rho_{bw}(\boldsymbol{b}))$ corresponds to $\boldsymbol{u} \cdot \boldsymbol{v} \mod t$ when $dp^2 < t$ and $d \leq n$.

Under appropriate noise management, a useful property is derived from homomorphic addition.

$$\begin{aligned}& (\mathrm{E}_{pk}(\rho_{fw}(\boldsymbol{u}_1)) \oplus \mathrm{E}_{pk}(\rho_{fw}(\boldsymbol{u}_2))) \otimes \mathrm{E}_{pk}(\rho_{bw}(\boldsymbol{v})) \\ =\ & \mathrm{E}_{pk}((\sum_{k=0}^{d-1}(u_{1k} + u_{2k})x^k)) \otimes \mathrm{E}_{pk}(\rho_{bw}(\boldsymbol{v})) \\ =\ & \mathrm{E}_{pk}(\rho_{fw}(\boldsymbol{x}_1 + \boldsymbol{x}_2)) \otimes \mathrm{E}_{pk}(\rho_{bw}(\boldsymbol{y})) \\ =\ & \mathrm{E}_{pk}((\boldsymbol{x}_1 + \boldsymbol{x}_2) \cdot \boldsymbol{y} + \mathcal{U}).\end{aligned}$$

Therein, $\mathcal{U}$ is unconcerned terms with degree greater than 0.

E. Secure Outsourcing GWAS Protocol

Let Q data holders participate in the procedure. They separately hold genotype vector $\boldsymbol{x} \in \{0,1,2\}^M$ and phenotype vector $\boldsymbol{y} \in \{0,1\}^M$. Let $\pi : \{1,2,\ldots,Q\} \times \{1,2,\ldots,M\} \to \{0,1,2\}$ be an assignment function that represents the partition of genotype/phenotype held by the q-th data holder. For example, the partition of genotype vector $\boldsymbol{x}$ for the q-th data holder is represented as shown below.

$$\pi_{\boldsymbol{x}}(q,i) = \begin{cases} x_i \text{ if } q\text{-th data holder holds the } i\text{-th element of } \boldsymbol{x}, \\ 0 \text{ otherwise.} \end{cases}$$

We assume that each element of vectors is held by only one data holder, i.e., $\sum_q \pi_{\boldsymbol{x}}(q,i) = x_i$ holds for every i. The procedure of secure outsourcing GWAS is shown in Figure 1. At Step 3, the cloud merges all information into two ciphertexts. Frequencies $\boldsymbol{x}^A \cdot \mathbf{1}$, $\boldsymbol{y}^{case} \cdot \mathbf{1}$, $\boldsymbol{x}^A \cdot \boldsymbol{y}^{case}$ are evaluated via the scalar product at step 4. Given the fact that division on encrypted data can be costly, our protocol separately computes numerators and denominators in Eq 1. The researcher can download the encryption of these numerators and denominators and do the division locally after the decryption. The whole contingency table can be reconstructed with the outputs of our protocol which might cause privacy issue. We notice that this issue can be conducted by obfuscating methods such as differential privacy which is beyond the scope of this paper.

Procedure of Secure Outsourcing GWAS

1) Key Setup: The researcher generates a key pair (pk, sk) Then, the public key pk is distributed to the stakeholders via a key exchange protocol.
2) Upload ($q = 1, \ldots, Q$): The q-th data holder encodes its own information as

$$\begin{aligned}\phi_{q,\boldsymbol{x}^A} &= \sum_{i=0}^{M} \pi_{\boldsymbol{x}^A}(q,i)x^i, \\ \phi_{q,\boldsymbol{y}^{case}} &= -\sum_{i=0}^{M} \pi_{\boldsymbol{y}^{case}}(q,i)x^{n-i},\end{aligned}$$

and submits ciphertexts $\mathrm{E}_{pk}(\phi_{q,\boldsymbol{x}^A})$ and $\mathrm{E}_{pk}(\phi_{q,\boldsymbol{y}^{case}})$ to the cloud.
3) Join: The cloud joins the collected ciphertexts into two ciphertexts, one of which is for the genotypes; the other is for disease status.

$$\begin{aligned}ct_{\boldsymbol{x}^A} &= \oplus_{q=1}^{Q} \mathrm{E}_{pk}(\phi_{q,\boldsymbol{x}^A}) \\ ct_{\boldsymbol{y}^{case}} &= \oplus_{q=1}^{Q} \mathrm{E}_{pk}(\phi_{q,\boldsymbol{y}^{case}})\end{aligned}$$

The cloud prepares $ct_{\mathbf{1}} = \mathrm{E}_{pk}(\phi_{\mathbf{1}})$, $ct'_{\mathbf{1}} = \mathrm{E}_{pk}(\phi_{\mathbf{1}'})$ where $\phi_{\mathbf{1}} = \sum_{i=0}^{n-1} x^i$ and $\phi_{\mathbf{1}'} = -\sum_{i=0}^{n-1} x^i$.
4) The cloud calculates:

$$\begin{aligned}ct_{case} &= (ct_{\boldsymbol{y}^{case}} \otimes ct_{\mathbf{1}}) \oplus ct_{\boldsymbol{r}_1} \\ ct_A &= (ct_{\boldsymbol{x}_A} \otimes ct'_{\mathbf{1}}) \oplus ct_{\boldsymbol{r}_2} \\ ct_{case,A} &= (ct_{\boldsymbol{y}^{case}} \otimes ct_{\boldsymbol{x}_A}) \oplus ct_{\boldsymbol{r}_3}\end{aligned}$$

where $ct_{\boldsymbol{r}_1}$, $ct_{\boldsymbol{r}_2}$, and $ct_{\boldsymbol{r}_3}$ are encryption of random polynomials from R_t but with zero constant term.
5) Hypothesis testing: The researcher downloads ct_{case}, ct_A, and $ct_{case,A}$ from the cloud. Then, by decrypting the ciphertexts, the researcher obtains

$$\begin{aligned}2r &= \boldsymbol{y}^{case} \cdot \mathbf{1}, \\ 2n_2 + n1 &= \boldsymbol{x}^A \cdot \mathbf{1}, \\ o_2 = 2r_2 + r_1 &= \boldsymbol{x}^A \cdot \boldsymbol{y}^{case}.\end{aligned}$$

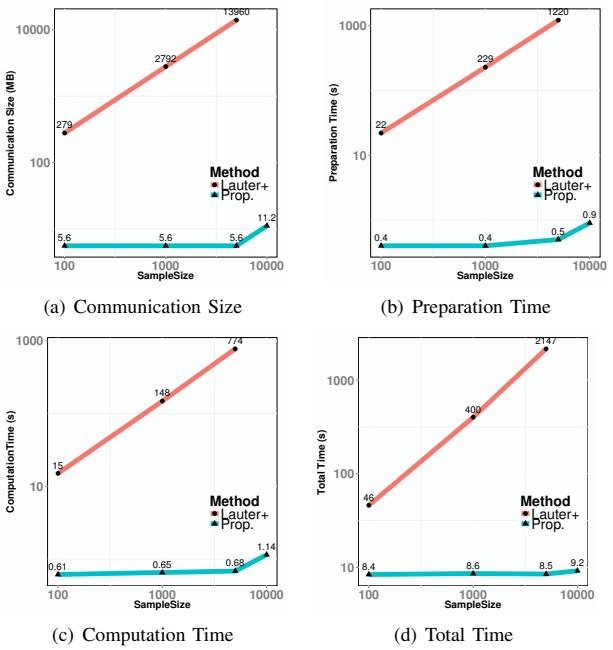
as the constant terms of the polynomials. χ^2 statistics can be evaluated using Eq. 1.

Fig. 1. Procedure for Secure Outsourcing of GWAS.

Because of space limitations, we omit the security proof and explain intuitively why the procedure is secure in the semi-honest model. The data holders receive no messages. They therefore learn nothing. The cloud receives the public key from the researcher and ciphertexts from the data holders. The cloud thus learns nothing under the ring-LWE assumption. Under the assumption that the cloud responds only to a specified query, the researcher obtains ciphertexts ct_A, ct_{case}, and $ct_{A,case}$ only. From the constant terms, the researcher learns $2r, 2n_0+n_1$ and o_1, which are the output of the protocol. The coefficients of the remaining terms reveal nothing to the researcher because of randomization.

V. Experiments and Discussion

We benchmarked the communication and computational costs of our procedure. All experiments were conducted on two computers with Core i7 2.3 GHz CPU and 16 GB RAM:



(a) Communication Size (b) Preparation Time

(c) Computation Time (d) Total Time

Fig. 2. Performance of proposed algorithm and the algorithm of Lauter [2] in terms of communication size (MB), preparation time(s) and computation time(s) for evaluating a single SNP.

one functions in the role of the data holders; the other serves as the cloud. Ethernet with bandwidth 100 Mbit/s was used for communication. All programs were implemented with C++; HElib [8] was used for homomorphic encryption. Lattice dimension n was set as $n = 8192$, message space parameter t =20,011, and maximum evaluation depth $L = 3$. According to Eq. 2, this setting offers at least 128-bit security. Lauter et al implemented their algorithm with a computational algebra system, Magma, in [2]. We reimplemented their algorithm with HElib in the same settings described above and compared it with our approach.

In the measurement of computational time, we measured time separately for Step 2 as preparation time and the time for processing from Step 3 to Step 4 as the computation time. The time for communication is dependent on the network bandwidth. We measured the amount of the communication (data size). The total execution time including, key setup, computation and communication was also measured.

Artificial Datasets. We benchmarked our proposed method and the method presented by Lauter et al. [2] for evaluation of a single SNP with different sample sizes $M = 100, 1000, 10,000$. The number of data holders $Q = 5$. Each holder has the same sample size M/Q. Results are presented in Figure 2. The method reported by Lauter et al. decomposes one genotype into three bits and encrypts them as three ciphertexts whereas our proposed method uses the packing technique, which enables encoding of $n = 8192$ genotype data into a single ciphertext. At Step 4, only three homomorphic multiplications are necessary to evaluate the χ^2 statistic for a single SNP. Experimental result Figure 2(c) shows that processing of the proposed algorithm at Step 3 and Step 4 costs less than one second given 5,000 samples, which is 1000+ times faster than the method reported by Lauter et al. From these results, secure outsourcing of GWAS with 3.0×10^6 SNPs of $10,000$ subjects is expected to be processed within 5 hours if 128 cores are used in parallel.

Real Dataset We demonstrated our procedure using actual datasets collected for genomic cohort studies for diabetes (DM). The study population comprised $n = 4257$ subjects collected from four medical institutes. Among those 4257 subjects, 1561 were diagnosed by medical doctors as having diabetes. For each subject, the genotypes of 312 SNPs were used in our analysis. We used the dominance/recessive model. Therefore, we tested 624 genetic features in total. Results show that we identified 39 SNPs having genetic association with diabetes with significance of $p < 0.05$. For processing hypothesis testing for this case–control study, 8.5 seconds were needed, including, communication time.

The secret-sharing solution [1] reported that computation of χ^2 statistics with 1080 subjects and 262264 SNPs cost over 2-3 hours. In terms of computational time, secret-sharing approach shows better performance, whereas it requires as least three cloud servers. In contrast, our our secure outsourcing procedure takes longer to execute than a solution based on secret sharing. However, our solution works with a single cloud server while solutions based on secure sharing require several non-colluding servers.

From these results, we can conclude that our secure outsouring procedure provides an efficient and practical secure outsourcing alternative for GWAS.

Acknowledgment The work is supported by JST CREST "Advanced Core Technologies for Big Data Integration". We appreciate reviewers for fruitful comments and discussion.

REFERENCES

[1] L. Kamm, D. Bogdanov, S. Laur, and J. Vilo, "A new way to protect privacy in large-scale genome-wide association studies," *Bioinformatics*, vol. 29, no. 7, pp. 886–893, 2013.

[2] K. Lauter, A. López-Alt, and M. Naehrig, "Private computation on encrypted genomic data," in *14th Privacy Enhancing Technologies Symposium, Workshop on Genome Privacy*, 2014.

[3] Y. Yamada, H. Matsuo, T. Segawa, S. Watanabe, K. Kato, T. Kameyama, K. Yokoi, S. Ichihara, N. Metoki, H. Yoshida *et al.*, "Assessment of genetic factors for type 2 diabetes mellitus," *International journal of molecular medicine*, vol. 18, no. 2, pp. 299–308, 2006.

[4] S. Halevi and V. Shoup, "Algorithms in helib," in *Advances in Cryptology–CRYPTO 2014*. Springer, 2014, pp. 554–571.

[5] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "(leveled) fully homomorphic encryption without bootstrapping," in *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*. ACM, 2012, pp. 309–325.

[6] C. Gentry, S. Halevi, and N. P. Smart, "Homomorphic evaluation of the aes circuit," in *Advances in Cryptology–CRYPTO 2012*. Springer, 2012, pp. 850–867.

[7] M. Yasuda, T. Shimoyama, J. Kogure, K. Yokoyama, and T. Koshiba, "Secure pattern matching using somewhat homomorphic encryption," in *Proceedings of the 2014 ACM CCSW 2013*. ACM, 2013, pp. 65–76.

[8] S. Halevi and V. Shoup, "HELib," http://shaih.github.io/HElib/index.html, accessed: 2014-12-10.