

An Active Genomic Data Recovery Attack

M. AKGÜN


Abstract— With the decreasing cost and availability of human genome sequencing, genomic privacy becomes an important issue. Several methods have been proposed in the literature to overcome these problems including cryptographic and privacy-preserving data mining methods: homomorphic encryption, cryptographic hardware. In recent work, Barman et. al studied privacy threats and practical solutions considering an SNP based scenario. The authors introduced a new protocol where a malicious medical center processes an active attack in order to retrieve genomic data of a given patient. The authors have mentioned that this protocol provides a trade-off between privacy and practicality. In this paper, we first give an overview of the system for SNP based risk calculation. We provide the definitions of privacy threats and briefly Barman et al.'s protocol and solution. The authors proposed to use a weighted sum of SNP coefficients for calculating disease tendency. They argue that the specific choice of the bases would prevent unique identification of SNPs. Our main observation is that this is not true. Contrary to the security claim, SNP combinations can be identified uniquely in many different scenarios. Our method exploits a pre-computed look-up table for retrieving SNPs' values from the test result. An attacker can obtain all SNP values of a given patient by using the pre-computed look-up table. We provide practical examples of weights and pre-computed tables. We also mention that even in the case where the table is large and the attacker can not handle it at one time, he can still gather information using multi queries. Our work shows that more realistic attack scenarios must be considered in the design of genetic security systems.

Index Terms—Genomic privacy, secure computation.

I. INTRODUCTION

RECENT DEVELOPMENTS in high throughput sequencing technologies led to a decrease in the cost of genomic sequencing. As a result of this, next-generation sequencing is deployed more and more in clinical diagnosis and treatment. Large scale genomic projects are announced which aim to sequence thousands of individuals (Genomics England [1]). Since genomic data includes sensitive information for individuals and their relatives, efficient use of this data with privacy-preserving techniques becomes an important issue.

METE AKGÜN, is with the Institute for Translational Bioinformatics at the University Hospital Tübingen, Tübingen, Germany, (e-mail: mete.akguen@uni-tuebingen.de).

 <https://orcid.org/0000-0003-4088-2784>

Manuscript received March 22, 2019; accepted August 16, 2019.

DOI: [10.17694/bajece.543555](https://doi.org/10.17694/bajece.543555)

In hospitals, there is a lack of expertise in protecting the genomic data of their patients. Due to the size of the data and the limited resources, it is often difficult for hospitals to safely store, process and maintain the genomic data of patients. The prevention of cyber-attacks by hospitals may not be possible due to insufficient high skilled workers and technology. The solution to this problem is the storage and processing of genomic data in a privacy-protected manner in a third-party service provider. In this case, service providers must process them without seeing the content of the data.

The human genome consists of four different nucleotides (A,C,G,T). These nucleotides form about 20.000 - 25.000 genes responsible for producing various types of proteins which are assigned inside the cells during whole life processes. About %99.5 of the genome is common in the human population where the remaining portion makes up the genetic variance. Most genetic variants in an individual are Single Nucleotide Polymorphisms (SNPs). A single nucleotide poly-morphism (SNP) can be defined as a variation occurring with some probability in a population where a single nucleotide differs from the reference genome. As a result of Genome-Wide Association Studies (GWAS), SNPs provide probabilistic information about the susceptibility of a disease. Generally, a few SNP combinations are evaluated together to calculate the overall inclination to a syndrome such as cardiovascular disease or Alzheimer. Since SNPs form the nonredundant part of the genome and contains minimalistic information, it makes sense to consider privacy-preserving protocols in terms of SNP's.

There are many different types of threats and security models where genomic privacy is a concern: querying on private genomic data, secure querying on public data, secure sequence alignment in public clouds [2]. Several methods have been proposed in the literature to overcome these problems including cryptographic and privacy-preserving data mining methods: homomorphic encryption [3], cryptographic hardware [4],[5].

A. Related Work

Ayday et al. [6] proposed a system based on homomorphic encryption to protect individual's privacy in disease risk tests. This work also proposes to use storage and processing unit to store sensitive data in encrypted form and disease risk tests are performed by authorized institutions using homomorphic encryption technique and secure integer comparison. In this solution, a storage and processing unit (SPU) stores all the

SNPs (approximately 40 million) of the patient. Ayday et. al solved the storage problem in [13] without sacrificing privacy. They classify SNPs as real SNPs and potential SNPs, where real SNPs are set of SNPs observed in the patient. SPU stores

the real SNPs instead of storing all SNPs. However, this constitutes a problem for privacy as SPU stores positions of

TABLE I
COMPARISON OF PREVIOUS SOLUTIONS

Work	Privacy			Authorization	Efficiency	Weighted Av
	Test Inference	Passive SNP	Active SNP			
Ayday et al. [6]	✓	X	X	X	X	✓
Ayday et al. [7]	✓	✓	X	X	X	✓
Danezis and De Cristofaro [8]	✓	X	X	X	✓	✓
Djatkiko et al. [9]	✓	X	X	X	X	✓
Zhang et al. [10]	✓	X	X	X	✓	✓
Fan and Mohanty [11]	✓	X	X	X	✓	✓
Perillo and De Cristofaro [12]	✓	X	X	✓	✓	✓

the real SNPs in plain text. Ayday et. al solved this problem by storing the real SNPs along with some redundant content from the set of potential SNPs.

In order to improve Ayday et al.'s scheme [6], Danezis and De Cristofaro [8] used Additively Homomorphic Elliptic Curve based El-Gamal (AH-ECC) [14] instead of the Paillier cryptosystem in order to decrease the computational overhead. The patient has a smartcard that participates in the protocol execution. The lost of the smartcard can cause privacy violation. Furthermore, the cloud provider knows the number of SNPs of each patient. This is also a data leakage. Perillo and De Cristofaro [12] proposed a cryptographic protocol for running different types of tests on individuals' genetic data. Their scheme is also based on the use of AH-ECC [14]. Differently it provides authorization which means SNP wights and locations are verified by central authority such as the FDA.

Djatkiko et al. [9] proposed a privacy-preserving algorithm to compute genomic tests that need the linear combination of SNP values. They applied partially homomorphic Paillier encryption and private information retrieval techniques to protect patients' privacy. The computational overhead of their solution is very high when compared to that of Ayday et al.'s solution.

Zhang et al. [10] proposed a framework for disease risk calculation using SNP values. Their framework reduces the storage overhead of previous solutions significantly by using bloom filters. It also reduces communication cost by indexing the encrypted genomic data.

Fan and Mohanty [11] proposed a solution for privacy preserving calculation of the susceptibility of a patient to a particular disease. The proposed scheme is based on Shamir's (1, n) secret sharing [15] which allows the computation of a certain number of multiplications and unlimited additions. It is more efficient than Ayday et al.'s solution [6] in terms of storage and computation time.

Readers are recommended to read surveys in [16], [2], [17], [18], [19] and [20] for more information on genomic privacy.

B. Our Contributions

All existing works provide security under semi-honest model in which the involving parties are not able to deviate the protocol description. It is very easy to provide security under this model with lower communication and computation complexities because adversaries are not allowed to change their inputs and to collude with other parties. This shows that all previous works are vulnerable to active SNP retrieval attacks in which an attacker can modify SNP weights in order to learn SNP values. The comparison of previous solutions is given in Table I.

Barman et al. [21] proposed a solution that makes all existing works secure to active SNP retrieval attacks. They studied privacy threats and practical solutions considering an SNP based disease risk calculation scenario. The authors introduced a new protocol where a malicious medical center processes an active attack in order to get SNP values of an individual. The authors mentioned that this protocol provides a tradeoff between privacy and practicality. In this study, we show that the solution offered by Barman et. al [21] does not prevent the leakage of SNP values. We show that SNP combinations can be uniquely identified in many different scenarios. Our method uses a pre-calculated lookup table to retrieve the values of the SNPs from the test result. The attacker can obtain all SNP values of a particular patient using the previously calculated lookup table. We present practical examples of weights and pre-calculated tables. We also observe that even if the lookup table is very large to handle, and the attacker can infer SNP values with multiple queries Our study shows that more realistic attack scenarios should be considered in the design of genetic security systems.

This paper is organized as follows. In Section II, we give an overview of the system model for genetic risk test calculation. In Section III, we give the definitions of privacy threats. In Section IV, we briefly define Barman et al.'s protocol [21] and their privacy solution. In Section V, we explain our observation that in fact, the solution is redundant. In Section

VI, we explain possible and existing countermeasures in order to eliminate active SNP retrieval attacks. Finally, in Section VII, we conclude the paper.

II. SYSTEM MODEL

In this section, we give the overview of the generic model described in the literature [6], [21] before. This model is constructed in order to calculate genetic risk test in a privacy-preserving way. In brief, a patient (P) sends his sample to the certificated institution (CI) for sequencing. The CI extracts genomic variants (SNPs) of the patient and encrypts SNPs. Then, the CI sends encrypted genomic data to the data center (DC). The CI is also responsible to distribute encryption keys to the related parties. The DC stores the encrypted genomic data. Medical center(s) (MC) communicate with the DC in order to compute genetic risk test in a privacy-preserving way. The system model is summarized in Figure 1.

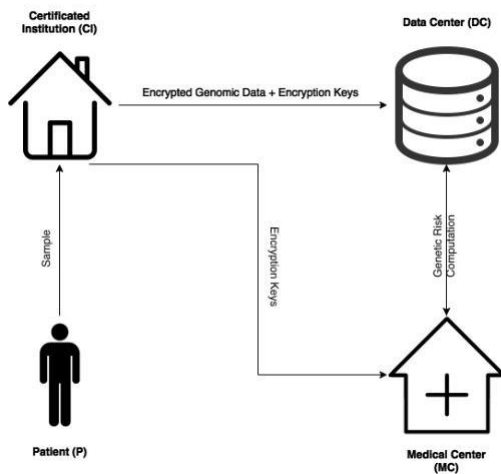


Fig.1. System model architecture

The genetic risk (G) is usually computed as a weighted sum of SNPs' values (Equation 1). W_i is the contribution (weight) of SNP_i . This computation can be done in a privacy-preserving way using secure multiparty computation or smart cards [22]. At the end of the test, the MC learns only the test result, but not the SNPs' values. Furthermore, DC does not learn the SNPs' weights.

$$\sum_{i=0}^n (W_i * SNP_i)$$

III. PRIVACY THREATS

Barman et al. [21] investigated privacy threats for system model architecture described in Section II. In the literature, P and CI are considered as honest parties and the MC and the DC are considered as honest-but-curious parties. Barman et al. [21] extended possible privacy threats by considering the MC and the DC as honest, semi-honest (passive) and dishonest (active). They describe three main attacks.

A. Test Inference Attacks

The semi-honest DC can learn which SNPs are used and how often they are used from test queries. Therefore, DC can infer the disease which a corresponding patient is suffering from. If the DC can re-identify P, this violates the privacy of the patient P. Danezis et al. [8] proposed to use all SNP values of a given patient in the genetic risk computation in order to prevent test inference attack. Another solution [23] proposed to use oblivious RAM to prevent the DC from learning access patterns of the MC.

B. Passive SNP Retrieval Attack

The MC can learn SNPs' values from the test result because the risk calculation is a linear equation and the MC knows some parameters used in this equation such as SNPs' weights. As the number of queries increases for a given patient P, P's privacy decreases. Ayday et al. [7] proposed to deliver test result as a range in order to prevent this attack.

C. Active SNP Retrieval Attack

In active SNP retrieval attack, the dishonest MC can manipulate SNPs' weights in order to retrieve SNPs' values from test results easily. For example, the MC sets all SNP weights to 0 except $W_j = 1$. The MC can retrieve the value of SNP_j which is equal to the test result. In another version of active SNP retrieval attack, the MC sets SNPs' weights as consecutive powers of a number. Consider a test with three SNPs, the MC sets SNPs' weights as the following: $W_0 = 4^0$, $W_1 = 4^1$ and $W_2 = 4^2$. The test result G is $(36)_{10} = (210)_4$. An attacker can retrieve SNPs' values from $G = (210)_4$ as the following: $SNP_2 = 2$, $SNP_1 = 1$ and $SNP_0 = 0$.

IV. BARMAN ET AL.'S PROTOCOL

Barman et al. [21] offer a solution to overcome the active SNP retrieval attack. According to the authors' definition: active SNP retrieval attack can be practiced by the MC by setting new SNP weights for a given test to retrieve the SNPs' raw values without being detected. Their solution is to force the MC to iteratively utter some SNP weights to the DC until the DC assures that the current test is legitimate. As the authors' mention, this solution weakens the MC's privacy while giving more power to the DC. Learning the test parameters might allow the DC to practice the test inference attack and also the test parameters might be private to the MC. So, the MC can abort the protocol if it thinks that it has to give too much information about the test parameters. The authors assume that only the MC can get the mapping information from the CI and the SNPs are stored as shuffled at the DC. The suggested protocol based on the described system model is as follows:

1. The MC wants to compute a genetic risk test based on R SNPs of a given patient but it adds D dummy SNPs with zero weight to its query. By adding D dummy SNPs, it prevents the DC from a test

inference attack and by adding zero weights to the dummy SNPs, it convinces the DC that the test is legitimate. Dummy SNPs have no effect on the test result. The authors call $N = R+D$ as the total length of the query.

2. The MC sends a request of N SNPs and a commitment for each SNP weight W_i , to the DC. They call $C_i = \text{Commit}(W_i) \forall i \in [0, N-1]$ as the commitment.

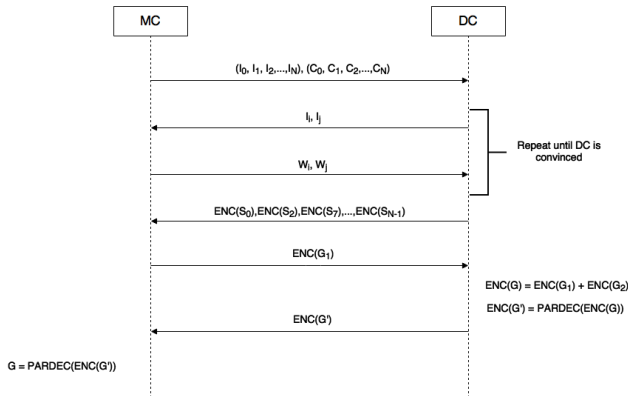


Fig.2. Barman et al.'s Protocol

3. The DC asks for the weights of random two indices $j, k \in [0, N-1]$. The MC responds with the relevant W_j and W_k to the DC.
4. The DC controls both the commitments C_j and C_k , and the weights W_j and W_k . If both weights are non-zero, and not different powers of the same number, the DC assures that the test is not an active SNP retrieval attack. Steps 3 and 4 are repeated until the DC is convinced or the MC aborts the protocol. For each iteration after the first, the DC can ask for only one new weight.
5. After believing that the test is not an active SNP retrieval attack, the DC sends the S ($S = N-2$ at most) encrypted SNPs corresponding to the weights not seen during the previous steps.
6. The MC homomorphically computes and sends the encryption of the first part of the test result, $ENC(G_1)$, according to the S SNPs.
7. The DC computes the encryption of the second part of the test result $ENC(G_2)$, according to the other encrypted SNPs whose weights are known from steps 3 and 4. The two partial results are homomorphically added into $ENC(G) = ENC(G_1) + ENC(G_2)$. $ENC(G')$, partial decryption of $ENC(G)$ is sent to the MC.
8. The MC decrypts the $ENC(G')$ and obtains G . The protocol ends

The authors declare that once the DC makes sure that the test is legitimate, it computes the encryption of the partial test result, $ENC(G_2)$. This guarantees that an active SNP retrieval attack cannot be performed, independent from the weights used for $ENC(G_1)$.

V. PROPOSED ACTIVE SNP RETRIEVAL ATTACK

In this section, we present an active SNP retrieval attack. We apply our attack to Barman et al.'s protocol. The authors suggest using specific bases for preventing unique identification of SNPs. We discover that this is redundant. An SNP combination can be identified uniquely in many different scenarios, we provide some examples. Our attack uses a pre-computed look-up table for retrieving SNPs' values from the test result. The attack can be described as follows.

1. The dishonest MC chooses R prime numbers as SNPs weights $(W_0, W_1, W_2, \dots, W_R)$.
2. The dishonest MC calculates test results for all possible SNPs' values and stores SNP's values and tests results in the table T . The sample look-up table with $R = 3$, $W_0 = 3$, $W_1 = 11$ and $W_2 = 23$ is shown in Table II.
3. The dishonest MC calculates commitments of SNPs weights.
4. The dishonest MC creates a query and adds D dummy SNPs with zero weights to its query.
5. The dishonest MC sends a request of $N = (R + D)$ SNPs and a commitment for each SNP weight W_i , to the DC.
6. The DC asks for the weights of random two indices $j, k \in [0, N-1]$. The MC responds with the relevant W_j and W_k to the DC.
7. The DC controls both the commitments C_j and C_k , and the weights W_j and W_k . If both weights are non-zero, and not different powers of the same number, the DC assures the test is not an active SNP retrieval attack. Steps 3 and 4 are repeated until the DC is convinced or the MC aborts the protocol. For each iteration after the first, the DC can ask for only one new weight.
8. After believing that the test is not an attack, the DC sends the S ($S = N-2$ at most) encrypted SNPs corresponding to the weights not seen during the previous steps. In our attack, the DC is convinced eventually because we choose at least two non-zero weights and SNPs' weights are guaranteed not to be different powers of the same number.
9. The dishonest MC homomorphically computes and sends the encryption of the first part of the test result, $ENC(G_1)$, according to the S SNPs.

10. The DC computes the encryption of the second part of the test result $ENC(G_2)$, according to the other encrypted SNPs whose weights are known from steps 3 and 4. The two partial results are homomorphically added into $ENC(G) = ENC(G_1) + ENC(G_2)$. $ENC(G')$, partial decryption of $ENC(G)$ is sent to the dishonest MC.
11. The dishonest MC decrypts the $ENC(G')$ and obtains G . Then, the dishonest MC retrieves SNPs' values by using the look-up table T.

The predetermined weight values give unique test results each SNPs' values as shown in Table II. Therefore, the success probability of our attack is 1. An attacker can retrieve all SNPs' values of a given patient by using a look-up table like Table II. In the protocol design, Barman et al. [21] considered two special attack scenarios. Our attack shows that more realistic attack scenarios must be considered when designing a security solution for the genetic system model shown in Figure 1.

TABLE II
ATTACK TABLE WITH SNP WEIGHTS ($W_0 = 3, W_1 = 11, W_2 = 23$)

SNP ₀	SNP ₁	SNP ₂	Weighted Sum
0	0	0	0
0	0	1	23
0	0	2	46
0	1	0	11
0	1	1	34
0	1	2	57
0	2	0	22
0	2	1	45
0	2	2	68
1	0	0	3
1	0	1	26
1	0	2	49
1	1	0	14
1	1	1	37
1	1	2	60
1	2	0	25
1	2	1	48
1	2	2	71
2	0	0	6
2	0	1	29
2	0	2	52
2	1	0	17
2	1	1	40
2	1	2	63
2	2	0	28
2	2	1	51
2	2	2	74

Although a small size look-up table is sufficient to retrieve all SNPs of a given patient, an attacker may want to create a large size look-up table. The difficulty of creating unique look-up tables increases, as the number of SNPs queried increases. When the dishonest MC cannot create a unique table, it can create multiple partial unique tables in order to create unique test result for all possible SNPs values. When the MC calculates the test result G which is not unique as shown in Table III, the MC sends another query for the same SNPs by using weights of another table in order to calculate the unique test result. For example, an attacker queries SNP_i ,

SNP_j and SNP_k with SNP Weights ($W_0 = 3, W_1 = 5, W_2 = 7$) given in Table III. If the calculated test result is 10, the attacker cannot determine the SNPs' values because 10 is not unique. Then, the attacker queries SNP_i, SNP_j and SNP_k with SNP Weights ($W_0 = 3, W_1 = 5, W_2 = 11$) given in Table III. The calculated test result is 14. The attacker can retrieve SNPs' values ($SNP_i = 1, SNP_j = 0$ and $SNP_k = 1$). As a result, the attacker retrieves the SNPs' values by using two partial unique look-up tables.

VI. COUNTERMEASURES

Many studies on genomic privacy have focused on the semi-honest model as an adversarial model. There are a few studies evaluating the dishonest model in the literature. The attack described in this article is carried out by malicious attackers playing on the inputs.

TABLE III
ATTACK TABLE WITH SNP WEIGHTS ($W_0 = 3, W_1 = 5, W_2 = 7$) AND SNP WEIGHTS ($W_0 = 3, W_1 = 5, W_2 = 11$)

SNP ₀	SNP ₁	SNP ₂	Weighted Sum 1	Weighted Sum 2
0	0	0	0	0
0	0	1	7	11
0	0	2	14	22
0	1	0	5	5
0	1	1	12	16
0	1	2	19	27
0	2	0	10	10
0	2	1	17	21
0	2	2	24	32
1	0	0	3	3
1	0	1	10	14
1	0	2	17	25
1	1	0	8	8
1	1	1	15	19
1	1	2	22	30
1	2	0	13	13
1	2	1	20	24
1	2	2	27	35
2	0	0	6	6
2	0	1	13	17
2	0	2	20	28
2	1	0	11	11
2	1	1	18	22
2	1	2	25	33
2	2	0	16	16
2	2	1	23	27
2	2	2	30	38

Clinicians want to see the result of the risk calculation in order to give the right treatment or prevention. The clinician (adversary) knows the weight values used in the risk calculations and can select them as desired. The risk is usually calculated as the weighted sum of the SNP values. In this case, the clinician will obtain SNP values from the calculated risk value. It is possible to make it difficult for an attacker to obtain SNP values by controlling weight values as Barman et al [21]. But this is not the definitive solution. Appropriate weight values will always be selected to obtain SNP values.

Ayday et al. [7] proposed to give risk value as a range. In this solution, as the range value increases, patient privacy increases but the consistency of the test decreases.

As another solution, weight values can be stored encrypted and clinicians do not know these values. Thus, it becomes

impossible to obtain SNP values for calculations where more than one SNP value is used. In this solution, it is difficult to keep the weight values secretly in a central system. In real life, this solution is very difficult to implement.

Another solution is to give the test result to the patient privately. To do this, the test result must be given to the patient in an encrypted form and the patient must be able to decrypt it. The patient can share the test result with the clinician if he wishes. A method for transferring, storing and decrypting the data must be specified. These operations can be done safely using smart card technology. The partially decrypted test result is transferred to the smart card of the user. The user can read the test result privately using a reader and software on the personal computer. Smart cards are capable of decryption. Since the smart cards are tamper-proof devices, the test result can be reliably stored.

None of the proposed solutions can provide a definitive solution. The clinician learning the exact result of the test can always infer the SNP values.

VII. CONCLUSION

A recent study [21] proposed a new threat model where malicious medical center tries to retrieve genomic data of a given patient. The authors also proposed a solution to this type of attack. They claim that their solution may be vulnerable to more sophisticated attacks involving multiple queries. We show that in fact there is a simpler type of attack. The attacker can learn genomic data using a simple pre-computed look-up table. It remains for future work to develop a security solution to prevent our active SNP retrieval attack. The researchers have to make a trade-off between privacy and efficiency in order to reduce the effects of active SNP retrieval attacks.

REFERENCES

- [1] "Genomics England — 100,000 Genomes Project," accessed: 2015-07-05. [Online]. Available: <http://www.genomicsengland.co.uk/>
- [2] M. Akgun, A. O. Bayrak, B. Ozer, and M. S. Sagiroglu, "Privacy preserving processing of genomic data: A survey," *Journal of Biomedical Informatics*, vol. 56, no. 0, pp. 103–111, 2015.
- [3] M. Goodrich, "The mastermind attack on genomic data," in *Security and Privacy, 2009 30th IEEE Symposium on*, May 2009, pp. 204–218.
- [4] M. Canim, M. Kantarcioglu, and B. Malin, "Secure management of biomedical data with cryptographic hardware," *Trans. Info. Tech. Biomed.*, vol. 16, no. 1, pp. 166–175, Jan. 2012.
- [5] C. Uhler, A. B. Slavkovic, and S. E. Fienberg, "Privacy-preserving data sharing for genome-wide association studies," *Journal of Privacy and Confidentiality*, vol. 5, no. 1, pp. 137–166, 2013.
- [6] E. Ayday, J. L. Raisaro, P. J. McLaren, J. Fellay, and J.-P. Hubaux, "Privacy-preserving computation of disease risk by using genomic, clinical, and environmental data," in *Proceedings of the 2013 USENIX Conference on Safety, Security, Privacy and Interoperability of Health Information Technologies*, ser. *HealthTech'13*. Berkeley, CA, USA: USENIX Association, 2013, pp. 1–1.
- [7] E. Ayday, J. L. Raisaro, J. Hubaux, and J. Rougemont, "Protecting and evaluating genomic privacy in medical tests and personalized medicine," in *Proceedings of the 12th annual ACM Workshop on Privacy in the Electronic Society, WPES 2013*, Berlin, Germany, November 4, 2013, 2013, pp. 95–106.

- [8] G. Danezis and E. D. Cristofaro, "Fast and private genomic testing for disease susceptibility," in *Proceedings of the 13th Workshop on Privacy in the Electronic Society, WPES 2014*, Scottsdale, AZ, USA, November 3, 2014, 2014, pp. 31–34. [Online]. Available: <http://doi.acm.org/10.1145/2665943.2665952>
- [9] M. Djatmiko, A. Friedman, R. Boreli, F. Lawrence, B. Thorne, and S. Hardy, "Secure evaluation protocol for personalized medicine," in *Proceedings of the 13th Workshop on Privacy in the Electronic Society*, ser. *WPES '14*. New York, NY, USA: ACM, 2014, pp. 159–162. [Online]. Available: <http://doi.acm.org/10.1145/2665943.2665967>
- [10] J. Zhang, L. Zhang, M. He, and S. Yiu, "Privacy-preserving disease risk test based on bloom filters," in *Information and Communications Security - 19th International Conference, ICICS 2017*, Beijing, China, December 6-8, 2017, *Proceedings*, 2017, pp. 472–486. [Online]. Available: https://doi.org/10.1007/978-3-319-89500-0_41
- [11] G. Fan and M. Mohanty, "Privacy-preserving disease susceptibility test with shamir's secret sharing," in *Proceedings of the 14th International Joint Conference on e-Business and Telecommunications (ICETE 2017) - Volume 4: SECURE*, Madrid, Spain, July 24-26, 2017., 2017, pp. 525–533.
- [12] M. Perillo and E. D. Cristofaro, "PAPEETE: private, authorized, and fast personal genomic testing," in *Proceedings of the 15th International Joint Conference on e-Business and Telecommunications, ICETE 2018 - Volume 2: SECURE*, Porto, Portugal, July 26-28, 2018., 2018, pp. 650–655.
- [13] E. Ayday, J. L. Raisaro, and J.-P. Hubaux, "Personal Use of the Genomic Data: Privacy vs. storage Cost," in *IEEE Global Communications Conference, Exhibition and Industry Forum – GLOBECOM*, 2013.
- [14] T. E. Gamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Trans. Information Theory*, vol. 31, no. 4, pp. 469–472, 1985.
- [15] A. Shamir, "How to share a secret," vol. 22, no. 11, pp. 612–613, 1979.
- [16] Y. Erlich and A. Narayanan, "Routes for breaching and protecting genetic privacy," *Nature Reviews Genetics*, vol. 15, no. 6, pp. 409–421, 2014.
- [17] M. Naveed, E. Ayday, E. W. Clayton, J. Fellay, C. A. Gunter, J.-P. Hubaux, B. A. Malin, and X. Wang, "Privacy in the genomic era," *ACM Computing Surveys*, vol. 48, no. 1, pp. 1–44, 2015. [Online]. Available: <http://dl.acm.org/citation.cfm?doi=2808687.2767007>
- [18] M. Z. Hasan, M. S. R. Mahdi, and N. Mohammed, "Secure count query on encrypted genomic data: A survey," *IEEE Internet Computing*, vol. 22, no. 2, pp. 71–82, 2018. [Online]. Available: <https://doi.org/10.1109/MIC.2018.112102323>
- [19] M. M. A. Aziz, M. N. Sadat, D. Alhadidi, S. Wang, X. Jiang, C. L. Brown, and N. Mohammed, "Privacy-preserving techniques of genomic data - a survey," *Briefings in Bioinformatics*, vol. 20, no. 3, pp. 887–895, 2019.
- [20] A. Mittos, B. Malin, and E. D. Cristofaro, "Systematizing genome privacy research: A privacy-enhancing technologies perspective," *PopETs*, vol. 2019, no. 1, pp. 87–107, 2019. [Online]. Available: <https://doi.org/10.2478/popets-2019-0006>
- [21] L. Barman, M. T. Elraini, J. L. Raisaro, J. Hubaux, and E. Ayday, "Privacy threats and practical solutions for genetic risk tests," in *2015 IEEE Symposium on Security and Privacy Workshops, SPW 2015*, San Jose, CA, USA, May 21-22, 2015, 2015, pp. 27–31. [Online]. Available: <https://doi.org/10.1109/SPW.2015.12>
- [22] M. Akgun, B. Erguner, A. O. Bayrak, and M. S. Sagiroglu, "Human genome in a smart card," in *HEALTHINF 2014 - Proceedings of the International Conference on Health Informatics, ESEO*, Angers, Loire Valley, France, 3-6 March, 2014, 2014, pp. 310–316. [Online]. Available: <http://dx.doi.org/10.5220/0004799903100316>
- [23] N. P. Karvelas, A. Peter, S. Katzenbeisser, E. Tews, and K. Hamacher, "Privacy-preserving whole genome sequence processing through proxy-aided ORAM," in *Proceedings of the 13th Workshop on Privacy in the Electronic Society, WPES 2014*, Scottsdale, AZ, USA, November 3, 2014, 2014, pp. 1–10. [Online]. Available: <http://doi.acm.org/10.1145/2665943.2665962>

BIOGRAPHIES



METE AKGÜN was born in Giresun, Turkey, in 1980. He received his B.Sc. degree in electrical engineering (with high honor) from Bahçeşehir University, Istanbul, in 2005, the M.Sc. and the Ph.D. degrees in computer engineering from Boğaziçi University, İstanbul, in 2009 and 2015, respectively. He worked as a research engineer in TÜBİTAK between 2006 and 2019. Since 2018, he is a postdoctoral researcher at University of Tübingen, Germany. His research interests include security, data privacy and bioinformatics.