**Proposal for comprehensive solutions for private association studies on encrypted genomic-phenomic data**

**Huang Lin**

**Feb. 15, 2016**

Supervisor: Huang Lin

Students: Carlessi Alexandre and Florian Briant

This project aims to redefine the contribution of JL’s ASIACCS16 submission [RAMTH2016]. Our final goal of this project is to provide comprehensive performance comparison of different privacy-preserving approaches for association studies (also including personalized medicine). Since privacy-preserving association studies fall into the framework of secure two-party computation, it can be realized by using various two-party computation building blocks such as additive homomorphic encryption [BCP2003], garbled circuit [Yao1986], or somewhat homomorphic encryption (SHE) [KD2014, NRLKMJ2015]. The original ASIACCS submission [RAMTH2016] has already provided both the design and implementation of additive homomorphic encryption based solution. This project will further provide the protocol designs and implementations for other two-party computation approaches. The write-up attached is the design of garbled circuit based solution [Lin2016]. In this semester, we will continue working on this project by not only providing the implementation of the gabled circuit based design, but also the design and implementation of SHE based solution. The major design challenge of SHE based solution is threefold:

1. Although the original ASIACCS submission [RAMTH2016] has provided a protocol for secure comparison of encrypted data based on additive homomorphic encryption, it is not clear whether or not there exists any more efficient SHE based encrypted data comparison protocol.
2. The original design encrypts individuals’ SNPs under different keys, which results in an interactive protocol among the participating parties in the subsequent privacy-preserving association study step. Our goal is to replace this interactive protocol with a newly designed proxy re-encryption algorithm to save the computation and communication overhead.
3. We will also investigate packing techniques [KD2014, NRLKMJ2015] used in somewhat homomorphic encryptions scheme to save the computation cost.

[RAMTH2016] Jean Louis Raisaro, Erman Ayday, Paul J. McLaren, Amalio Telenti, and Jean-Pierre Hubaux, Private Association Studies on Encrypted Genomic-Phenomic Data, 2016.

[BCP2003] E. Bresson, D. Catalano, and D. Pointcheval, “A simple public-key cryptosystem with a double trapdoor decryption mechanism and its applications,” Proceedings of Asiacrypt 03, pp. 37–54, 2003.

[Yao1986] A. Yao. How to Generate and Exchange Secrets. In 27th FOCS, pages  162–167, 1986.

[KD2014] Rohloff, K., & Cousins, D. B. (2014). A scalable implementation of fully homomorphic encryption built on NTRU. In Financial Cryptography and Data Security (pp. 221-234). Springer Berlin Heidelberg.

[NRLKMJ2015] Dowlin, N., Gilad-Bachrach, R., Laine, K., Lauter, K., Naehrig, M., & Wernsing, J. (2015). Manual for Using Homomorphic Encryption for Bioinformatics.

[Lin2016] Huang Lin, How to use secure two-party computation to perform private personalized medicine, 2016.

**How to use secure two-party computation to perform private personalized medicine**

**Huang Lin**

**Feb, 2016**

The primitive idea is to let CI encrypt the message using symmetric keys and deliver the ciphertext to the server (as shown in Fig.1). CI also delivers the necessary information to the clients. Then both the clients and server will get involved in a two-party computation to privately compute the output.

We assume that there are established secure and authenticated channels between CI and MU, also CI and SPU. Therefore, CI and the other two parties share symmetric keys, which could later be used for the both the encryption and authentication of the delivered messages. When CI delivers genomic data to SPU and MU for secure computation, he will first pick a fresh random IV and uses a symmetric encryption scheme *SEnc* to encrypt the data *D* using the shared symmetric key between CI and MU, i.e., . The IV will be delivered to MU through the secure channel between CI and MU. The symmetric encryption scheme SEnc could be based on a block cipher with OFB mode or CTR mode [KL2014]. The reason we choose such symmetric encryption scheme is that both modes allow the generation of key stream and performing encryption separately, which will imply simple circuit representation in the subsequent secure computation steps. We will further explain this reason in the following exposition.

Before SPU and MU engage in the secure computation protocol, MU will first use and *IV* that he receives from CI to generate the symmetric encryption key for the block cipher. Then, the secure two party computation protocol will take the ciphertext sent by CI and the symmetric key stream generated in this step as input. The first-level circuit representation for this protocol will correspond to the decryption of symmetric encryption scheme, which only consists of XOR operation between the ciphertext and the encryption key stream, which wouldn’t cost much due to the free XOR technique in secure two party computation protocol. The rest circuit representation will correspond to the computation steps in personalized medicine. We note that this simple circuit representation for the symmetric decryption step is only possible when the symmetric encryption scheme is based on block cipher with OFB or CTR modes since their key generation and decryption algorithm can be separated and their decryption algorithm only consists of XOR operation between the ciphertext and the key stream. We can employ ObliVM framework [LWNHS2015] to implement the above secure computation protocol.

The security analysis of this scheme is also quite straightforward. We assume SPU and MU are two non-colluding parties in the system model, and hence MU won’t share with SPU. Therefore, the semantic security of the underlying symmetric encryption scheme guarantees that the will be indistinguishable from random bit strings. On the other hand, without colluding with SPU, MU obtains no useful information on the underlying data *D* other than IV in this communication round. Since we also assume secure and authenticated channels between CI and MU, also CI and SPU, hence all the delivered message is secure from any eavesdropper of the communication channels. Therefore, the security of the scheme is reduced to the security of the underlying secure two-party computation framework, which concludes the analysis.

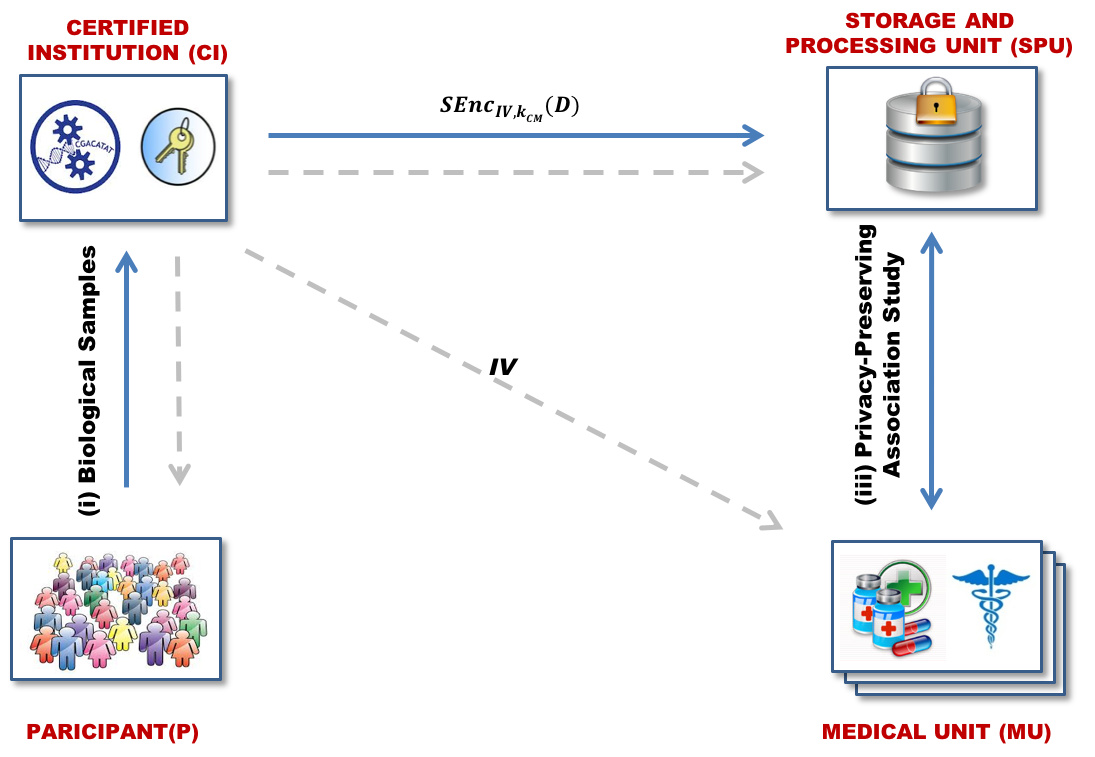


Figure 1. System Architecture

[KL2014] Katz, J., & Lindell, Y. (2014). Introduction to modern cryptography. CRC Press.

[LWNHS2015] Liu, C., Wang, X. S., Nayak, K., Huang, Y., & Shi, E. (2015, May). Oblivm: A programming framework for secure computation. In Security and Privacy (SP), 2015 IEEE Symposium on (pp. 359-376). IEEE.

**Improvement of privacy preserving genetic association studies based on a new variant of proxy re-encryption scheme**

**Huang Lin**

**Feb. 26, 2016**

This article aims to provide an improvement of privacy preserving genetic association studies scheme in [RAMTH2016]. The primitive idea of the original proposal includes the following operations: masking🡪partial decryption🡪decryption🡪key generation re-encryption🡪unmasking. Our goal is to replace this sequence of operations with a conditional proxy re-encryption scheme. The above operation sequence aims to transform genomic/phenomic data encrypted under different public keys to ciphertexts corresponding to an identical public key so that homomorphic operation can be perfomed over the transformed ciphertext. This process is conceptually close to the functionality of proxy encryption scheme which mainly aims to transform a ciphertext corresponding to a public key to that corresponding to another public key after a partial decryption of the ciphertext by the proxy.

In order for a proxy encryption to successfully replace the aforementioned operation sequence, there are mainly two requirements for the proxy encryption scheme:

1. The proxy encryption can support homomorphic operations. This is important because we have to make sure the transformed ciphertext can support homomorphic operation
2. The re-key can be secret shared by the MU and SPU, and the re-encryption process can be performed in a threshold manner. This property enables our modification to the original proxy re-encryption.

There already exists a proxy re-encryption scheme based on NTRU [NAL2015], which satisfies the above two requirements. The aforementioned operation sequence can be replaced by the modified proxy re-encryption scheme as follows:

1. Since CI is responsible for generating public/secret key pairs for all the individuals and also that of the final homomorphic encryption scheme. This implies at least at the beginning of the system, CI has both for individual *i* and the secret key for the final homomorphic operation. By invoking the rekey generation algorithm in [NAL2015], CI inputs both and to generate the rekey . For each individual, CI will perform the similar procedure, then he will apply the secret sharing procedure to split the rekeys into two shares and , which will be distributed to MU and SPU respectively. As usual, CI will also split the secret key into two shares, which will then be distributed to MU and SPU.
2. During the association study, for the input ciphertext from the individuals, both MU and SPU will first work together to transform the individual ciphertext to the ciphertext corresponding to . This step will include MU first performing partial re-encyption , and then SPU performing the next partial re-encryption as . Then the final transformed ciphertext would be . Then they will perform homomorphic operation over the transformed ciphertext and use the shares of to decrypt the final result.

[RAMTH2016] Jean Louis Raisaro, Erman Ayday, Paul J. McLaren, Amalio Telenti, and Jean-Pierre Hubaux, Private Association Studies on Encrypted Genomic-Phenomic Data, 2016.

[NAL2015] Nuñez, D., Agudo, I., & Lopez, J. (2015, April). NTRUReEncrypt: An efficient proxy re-encryption scheme based on NTRU. In Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security(pp. 179-189). ACM.

[BCP2003] Bresson, E., Catalano, D., & Pointcheval, D. (2003). A simple public-key cryptosystem with a double trapdoor decryption mechanism and its applications. In Advances in Cryptology-ASIACRYPT 2003 (pp. 37-54). Springer Berlin Heidelberg.

[SC2009] Shao, J., & Cao, Z. (2009). CCA-secure proxy re-encryption without pairings. In Public Key Cryptography–PKC 2009 (pp. 357-376). Springer Berlin Heidelberg.