fingercode vector. For the fingerprint dataset, we used "NIST special database 9", which contains fingerprint images of 2000 different individuals. The image size is 832 * 768 pixels, and the images are scanned at 500 dpi. The dataset contains 2 different fingerprint instances of the same finger per individual, resulting in a total of 4000 fingerprints. In order to calculate an appropriate threshold for distance that can be used to decide whether or not to authenticate a user given his fingerprint, we measured a 1-norm distance between two different instances of fingerprint of the same finger. To check how much distance gets larger when fingerprint of wrong individual is used for verification, we also measured a 1-norm distance between i-th person's fingerprint and (i+1)th person's fingerprint. Using these 2 sets of 2000 different distances, we could determine a suitable threshold for "NIST special database 9" that can maximize true positive rate while minimizing false positive rate. We then compared true positive rates and false positive rates of 1-norm distance encrypted fingerprint authentication scheme to those of 2-norm distance scheme (the original filterbank-based fingerprint matching algorithm). While we mainly tested our scheme on the specified setting, we also tested on different settings such as fingerprint vectors of length 640 doubles.

3 RESULTS

3.1 Matching performance

We used an Intel Core i5-7500 CPU with 3.40GHz * 4 and 64-bit Ubuntu 16.04 LTS. As expected, our extension of FHEW library calculates exact 1-norm distance given two encrypted fingercode vectors since FHEW library is fully homomorphic. Hence, if any degrade in performance was introduced, it must came from the relaxation that we introduced by rounding vectors of doubles to vectors of integers and calculating 1-norm distance instead of 2-norm distance. However, under the setting we tested, the result shows that such degrade in performance is negligible. We plotted FAR (False Acceptance Rate) and FRR (False Rejection Rate) versus threshold for both our relaxed version of 1-norm distance and original version of 2-norm distance. The plots are shown in Figures 2 and 3. By carefully setting the threshold value, we could achieve comparable performance to the original 2-norm distance algorithm. For example, setting 1-norm distance threshold to 100 gave false negative of 1621, true positive of 379, false positive of 134, and true negative of 1865. Setting 2-norm distance threshold to 32.14 gave false negative of 1633, true positive of 367, false positive of 134, and true negative of 1865.

3.2 Time performance

Under our setting, executing one of 5 gate computations that FHEW library provides took an average of 0.1934 seconds. Homomorphic XOR gate computation took an average of 0.5594 seconds, just about 3 times longer than one of 5 basic gate computations since 3 gate computations were required to execute XOR gate computation. Encrypting fingercode vector was almost instantaneous. Encryption of 640-integer length fingerprint vector took an average of 0.02393 seconds. Given two encrypted fingercode vectors of length 16, time to calculate 1-norm of difference between two vectors required an average of 209 seconds, meaning an average of 209 seconds before the user could be authenticated. We have also verified that

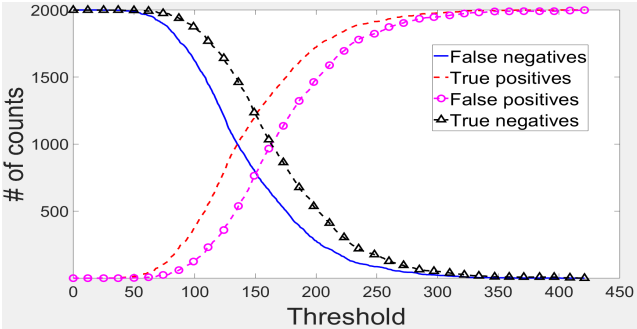

Figure 2: FARs and FRRs versus threshold when 1-norm distance is used, rounded doubles to integers.

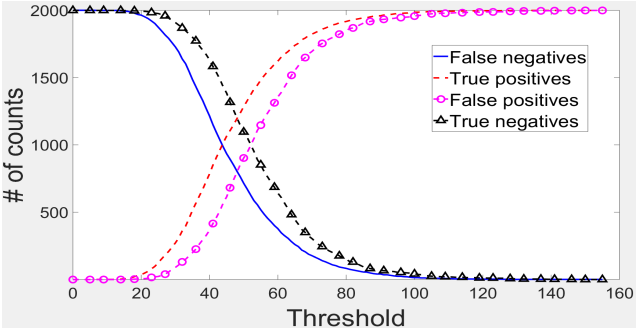

Figure 3: FARs and FRRs versus threshold when 2-norm distance is used.

time taken to compute 1-norm distance between two vectors grows linearly as vector length increased.

4 CONCLUSION

In this paper, we extended FHEW library to calculate a 1-norm distance between two encrypted fingercodes. We then applied extended library to the filterbank-based fingerprint matching algorithm. Even though it takes relatively longer time (about 209 seconds) than conventional matching algorithms without encryption, this authentication time could be acceptable for some applications such as user registration and user identification services without requiring real-time processing. For future work, we will extend our implementation based on FHEW to cover various algorithms for biometric data.

ACKNOWLEDGMENTS

This research was supported in part by the MIST (2015-0-00914) and the ICT R&D program (No.2017-0-00545).

REFERENCES

[1] Ilaria Chillotti, Nicolas Gama, Mariya Georgieva, and Malika Izabachene. 2016. Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds. In *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 3–33.

[2] Craig Gentry. 2009. *A Fully Homomorphic Encryption Scheme.* Ph.D. Dissertation. Stanford, CA, USA. Advisor(s) Boneh, Dan. AAI3382729.

[3] Anil K Jain, Salil Prabhakar, Lin Hong, and Sharath Pankanti. 2000. Filterbank-based fingerprint matching. *IEEE Transactions on Image Processing* 9, 5 (2000), 846–859.