Privacy-Preserving Mobile Video Sharing using Fully Homomorphic Encryption

Utsav Goswami*, Kevin Wang†, Gabriel Nguyen‡ and Brent Lagesse§
Computing and Software Systems
University of Washington Bothell
Bothell, WA USA
Email: *utsavg@uw.edu, †wangk24@uw.edu, ‡gaben@uw.edu, §lagesse@uw.edu

***Abstract*—Increased availability of mobile cameras has led to more opportunities for people to record videos of significantly more of their lives. Many times people want to share these videos, but only to certain people who were co-present. Since the videos may be of a large event where the attendees are not necessarily known, we need a method for proving co-presence without revealing information before co-presence is proven.**

In this demonstration, we present a privacy-preserving method for comparing the similarity of two videos without revealing the contents of either video. This technique leverages the Similarity of Simultaneous Observation technique for detecting hidden webcams and modifies the existing algorithms so that they are computationally feasible to run under fully homomorphic encryption scheme on modern mobile devices.

The demonstration will consist of a variety of devices preloaded with our software. We will demonstrate the video sharing software performing comparisons in real time. We will also make the software available to Android devices via a QR code so that participants can record and exchange their own videos.

I. Introduction

The continued development of small, high quality cameras has led to billions of people carrying around cameras nearly everywhere they go. Furthermore, these cameras are being integrated into emerging life-logging systems where photos, audio, and video are recorded by people constantly to provide an archive of their lives and augment their memory [1], [2], [3]. Many of the videos that are being recorded in these systems and in general contain semi-private information. These are videos that a person would like to share with others who they may or may not know, but not do so indiscriminately.

The goal of our system is to enhance the ability for users to share videos with people that they may or may not know, but only if those people were co-present when the video was taken. For example, Alice and Bob are attending a party for a mutual acquaintance, Carol, but did not meet each other at the party. Alice and Bob do not know each other, but both of them take videos of the festivities. At a later time, Alice is looking for additional videos of the party because she is trying to create a video collage for Carol. Bob does not want to provide the video to Alice if she was not actually at the party and likewise, Alice does not want to reveal her video to Bob if he was not there. Our system enables Alice to use her video as proof to Bob that she was present without revealing any of the content of the video.

No existing work meets the requirements of our video sharing system. The focus of some previous work has been on obscuring the video itself whereas we want all videos that are shared to be in original condition [4]. Also, these systems are designed to release information to the user, albeit in an obfuscated form, even if they were not present at the event. Other systems are designed to limit the sharing of video to known groups or users [5], [6] whereas we need videos to be distributed without having to have a pre-existing relationship with the other users. In many cases, a cloud server performs the processing on plaintext data and breaks our requirement that an honest, but curious, server should also not learn the content of the video [7].

We are able to accomplish this through the use of fully homomorphic encryption. This enables our system to operate on encrypted data so that no information about the videos is revealed to either party. The resulting computation produces a similarity value that can be used to decide whether or not to share the video. Our proposed demo will allow participants to record video during the demo session and then compare the videos to see if they should be shared without having to reveal any information about the video to the other person.

II. Design

A. Attacker Model

The attacker can either initiate a request or can respond to a request. In either case, the attacker can encrypt a real video or create an arbitrary collection of bytes to attempt to defeat the system. We assume that the cryptosystem is strong enough to prevent an attacker from decrypting the data [8].

B. System Design

Our system leverages two main technologies to enable video sharing. The first is *Similarity of Simultaneous Observation* (SSO) [9]. The algorithms presented in [9] enable a user to determine with high accuracy if two videos are of the same scene at the same time. The second is fully homomorphic encryption (FHE) [10] which enables us to execute algorithms on encrypted data. A cryptosystem is said to be fully homomorphic if it is homomorphic for all algorithms.

Due to complexities of developing efficient algorithms under a FHE cryptosystem, we have had to modify the original SSO algorithm so that it could be used in our system. Since

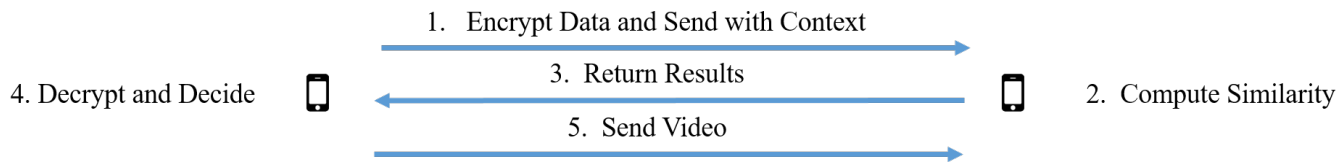

Fig. 1. Video Sharing Protocol

FHE computational libraries have not reached the maturity of traditional mathematical libraries, many functions that developers take for granted are not natively available under an FHE cryptosystem. As a result, we had to use a combination of pre-processing, function approximations through Taylor expansion, and the use of different distance measures in order to achieve similar results the original SSO paper.

The original SSO system used Kullback-Liebler Divergence (KLD), Jensen-Shannon Divergence (JSD), Dynamic Time Warping (DTW), and Pearson Correlation Coefficient (PCC) as the similarity measures. Due to poor classification performance, we eliminated DTW and PCC from this system. Additionally, the computation required by JSD required expensive approximations when done as a FHE algorithm that were prone to producing significant errors in certain regions of the input space, so we did not use it. We were able to approximate KLD under FHE through the use of precomputing and encrypting several values that would later be used under the FHE scheme. We also identified two additional similarity measurements, Bhattacharyya coefficient and Cramer distance that we were able to implement efficiently under the FHE cryptosystem all with mean approximation errors of less than 0.62% of the mean difference between a similar video and a different video.

Figure 1 demonstrates the protocol used to determine if the two videos are of the same scene once the user on the right has initiated a request. The process can be summarized as follows:

1) **Encrypt Data and Send with Context.** The user that is making the decision whether or not to share the video preprocesses the video and encrypts it. The encrypted data and the context is sent to the other user that is trying to prove they were co-present over a mutually authenticated TLS connection.
2) **Compute Similarity.** The other user computes all of the requisite similarity scores. Note that the other user cannot directly see what effect the values they use as inputs into the algorithms have. They can only see if they are sent the file or not.
3) **Return Results.** The encrypted results are returned to the decision-making user.
4) **Decrypt and Decide.** The decision-making user decrypts the results and provides them as inputs into the classification model.
5) **Send Video.** If the decision-making user is satisfied with the results, they send the video to the other participant.

III. Prototype Implementation and Demonstration

We have implemented our Proof of Presence Video Sharing system (PoP-Share) for mobile devices. Our mobile phone implementation is designed to run as an Android App. The app is built for Android 9.0 and runs on 64 bit CPUs. The app uses SEAL 3.3 [11] built using the Android NDK, so this implementation uses the CKKS [12] implementation of fully homomorphic encryption. We have implemented all of the FHE functionality in native C++ with a JNI wrapper to be accessed through the Android app. We currently are using a native Android GUI, but will be implementing the App with the Kivy GUI in the near future so the mobile phone user experience will match that of the PC implementation user experience. SEAL also runs on iOS, but we have not yet ported PoP-Share to that platform.

A. Performance

1) Timing: On an Android phone, the preprocessing time, which includes precomputation and encryption of all the data takes between 4 and 8 seconds on a Pixel 2 mobile phone to compare 60 seconds of video. While this is a large amount of time for pre-processing on the mobile phone, this is a process does not have to be run every time. We cache the results after a video has been processed, so this is only a one time cost that can be incurred during the time between recording a video and sharing the video, so it is unlikely to be noticed by the user since a user is not expected to immediately share the video with a person that they are unsure was at the event. The similarity computations on average complete in 200-400 milliseconds on the mobile phone.

B. Classification

As noted in table I, our system performs similarly to the original SSO system in terms of classification. We optimized our classifier for precision rather than F1 score because we considered the cost of a false positive to be much worse than a false negative.

TABLE I
Comparison of two SSO-based Systems

System	F1	Prec	Recall	Acc	Error
PoP-Share	96.63	97.73	95.56	95.16	4.84
Original SSO[9]	96.13	92.56	100.00	96.30	3.70

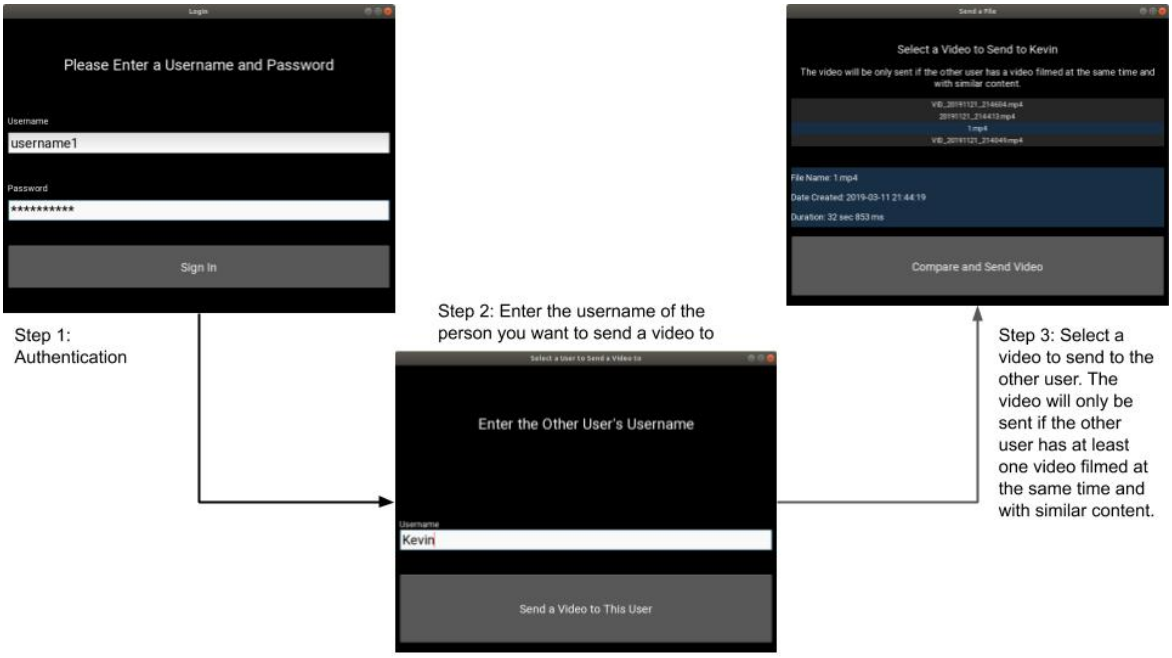

Fig. 2. Flow of User Interaction

C. Showcase Plan

We will bring several devices such as laptops and mobile phones that have our software preloaded on them that we will use to demonstrate the video sharing software. In addition, we will also provide a QR code that will link to an APK that Android users will be able to install on their mobile phones so that they can also participate in the demo with their own devices and try it in other areas besides in front of our poster. We intend to connect to the venue's Wi-Fi to perform the demonstration.

IV. Conclusion

The main idea we want visitors to our demo to take away is that it is possible to build privacy-preserving systems to share data on mobile devices without revealing any information until you are convinced the transaction is appropriate. We have demonstrated this by creating a system that enables video exchange using the video itself as a proof of presence. No information is revealed until the participant chooses to release the video due to the application of a fully homomorphic cryptosystem. Additionally the algorithms for determining similarity have been designed in such a way that they can run on a mobile phone in hundreds of milliseconds.

References

[1] C. Gurrin, A. F. Smeaton, and A. R. Doherty, "LifeLogging: Personal Big Data," *Foundations and Trends® in Information Retrieval*, vol. 8, no. 1, pp. 1–125, 2014. [Online]. Available: http://dx.doi.org/10.1561/1500000033

[2] S. Jiang, Z. Li, P. Zhou, and M. Li, "Memento: An Emotion-driven Lifelogging System with Wearables," *ACM Trans. Sen. Netw.*, vol. 15, no. 1, pp. 8:1–8:23, Jan. 2019. [Online]. Available: http://doi.acm.org/10.1145/3281630

[3] C. Dobbins and S. Fairclough, "A mobile lifelogging platform to measure anxiety and anger during real-life driving," in *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, Mar. 2017, pp. 327–332.

[4] M. Upmanyu, A. M. Namboodiri, K. Srinathan, and C. V. Jawahar, "Efficient privacy preserving video surveillance," in *2009 IEEE 12th International Conference on Computer Vision*, Sep. 2009, pp. 1639–1646.

[5] J. Fan, H. Luo, M.-S. Hacid, and E. Bertino, "A Novel Approach for Privacy-preserving Video Sharing," in *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, ser. CIKM '05. New York, NY, USA: ACM, 2005, pp. 609–616, event-place: Bremen, Germany. [Online]. Available: http://doi.acm.org/10.1145/1099554.1099711

[6] B. Thuraisingham, G. Lavee, E. Bertino, J. Fan, and L. Khan, "Access control, confidentiality and privacy for video surveillance databases," in *Proceedings of the eleventh ACM symposium on Access control models and technologies - SACMAT '06*. Lake Tahoe, California, USA: ACM Press, 2006, p. 1. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1133058.1133061

[7] K. Yang, Z. Liu, X. Jia, and X. S. Shen, "Time-Domain Attribute-Based Access Control for Cloud-Based Video Content Sharing: A Cryptographic Approach," *IEEE Transactions on Multimedia*, vol. 18, no. 5, pp. 940–950, May 2016.

[8] M. Albrecht, M. Chase, H. Chen, J. Ding, S. Goldwasser, S. Gorbunov, S. Halevi, J. Hoffstein, K. Laine, K. Lauter, S. Lokam, D. Micciancio, D. Moody, T. Morrison, A. Sahai, and V. Vaikuntanathan, "Homomorphic Encryption Security Standard," HomomorphicEncryption.org, Toronto, Canada, Tech. Rep., Nov. 2018.

[9] K. Wu and B. Lagesse, "Do You See What I See? Detecting Hidden Streaming Cameras Through Similarity of Simultaneous Observation," *IEEE Pervasive Computing and Communications*, p. 10, 2019.

[10] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *Proceedings of the 41st annual ACM symposium on Symposium on theory of computing - STOC '09*. Bethesda, MD, USA: ACM Press, 2009, p. 169. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1536414.1536440

[11] *Microsoft SEAL (release 3.3)*, Jun. 2019. [Online]. Available: https://github.com/Microsoft/SEAL

[12] J. H. Cheon, A. Kim, M. Kim, and Y. Song, "Homomorphic Encryption for Arithmetic of Approximate Numbers," Tech. Rep. 421, 2016. [Online]. Available: https://eprint.iacr.org/2016/421