2023 IEEE 20th International Conference on Mobile Ad Hoc and Smart Systems (MASS) | 979-8-3503-2433-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/MASS58611.2023.00062

FaceGroup: Continual Face Authentication via Partially Homomorphic Encryption & Group Testing

Tingcong Jiang, Chuanneng Sun, Salim El Rouayheb, and Dario Pompili
Department of Electrical and Computer Engineering, Rutgers University–New Brunswick, NJ, USA
{*tingcong.jiang, chuanneng.sun, salim.elrouayheb, pompili*}*@rutgers.edu*

***Abstract*—The Zero Trust (ZT) paradigm has recently emerged. The core idea of ZT is never to trust but always authenticate. By incorporating ZT into network architectures, neither users nor service providers need to trust each other, which significantly enhances the security level of these architectures. Cloud-based facial authentication is one of the plausible access control solutions to bring ZT into network architectures. Many of the state-of-art works on encrypted cloud-based authentication use Fully Homomorphic Encryption (FHE) to protect users' private data. FHE enables a cloud server to perform arithmetic operations on encrypted inputs without decrypting them but with a significant computing overhead. In this work, we introduce novel approaches to incorporate Partially Homomorphic Encryption (PHE) into cloud-based facial authentication by changing the distance metric from Euclidean distance to Manhattan distance. As a result, we reduce the computational overhead by a factor ranging from 20 to 55. In addition, we propose a novel two-stage architecture for group facial recognition, which can further reduce the total computation cost of authentications required to identify an individual from a crowd. Compared with conventional facial recognition methods, to find people of interest, group facial verification can cut the cost of calculating facial recognition by 55%. With such a lightweight design, FaceGroup is scalable and can be deployed on resource-constrained devices.**

***Index Terms*—Face Recognition, Biometrics, Access control, Siamese Network, Homomorphic Encryption, Group Testing**

I. Introduction

Overview: To provide a high standard of security and protection to users and service providers, Zero Trust (ZT) paradigm [1] has recently emerged. Unlike Federated Learning (FL) [2], [3], where only the server is not trusted, the basic assumption that ZT makes is that none of the users and service providers should be trusted. Therefore, ZT advocates the idea of never trusting but always authenticating, and continuous authentication methods have become a common type of implementation of ZT. Although there are many authentication methods for physical access control, the methods suitable for continual authentication should be able to conduct remotely. A popular method of continually remote biometric authentication is identifying user identities via facial authentication.

Facial authentication methods can be classified into two categories: *on-device* and *off-device authentication*, depending on where the computation takes place. On-device facial authentication methods use local devices for computation, while off-device methods transfer most of the computation to third-party servers. On-device methods provide fast authentication, but their performance can suffer when onboard computation is intensive. Off-device methods provide a high capacity for large-scale authentication but can be vulnerable to malicious attacks and communication latency. Both of these two security systems have flaws and are vulnerable in certain circumstances. In this paper, we propose a novel hybrid security system to mitigate the flaws of on-device and off-device systems.

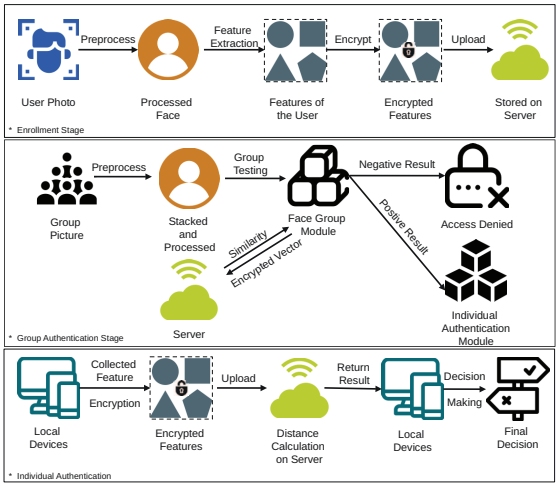


Fig. 1: Paradigm for FaceGroup system. Newly authorized personnel must enroll in the system by following the enrollment stage. Then, query faces are divided into batches, and the batches are processed by the group authentication stage. Only if the group authentication stage gives a positive result for a batch, every identity in the batch will be accurately authenticated in the individual authentication stage.

Motivation: A hybrid authentication system can address the limitations of on-device and off-device authentication systems. On-device authentication systems store data locally on cameras, requiring the access control system's authorization database [4] (i.e., the database that holds the information regarding who has access to rooms) to be updated for all devices whenever an authorized individual's access level changes. Unsuccessful updates to the authorization database can grant intruders access, making on-device methods unscalable and not compatible with the dynamic policy requirements made by ZT. Additionally, on-device methods have limited storage space and require upgrades when the number of authorized personnel increases.

Off-device authentication systems adopt a centralized structure, with heavy computation offloaded to cloud servers, enabling devices to be separated from the authorization database.

2155-6814/23/$31.00 ©2023 IEEE
DOI 10.1109/MASS58611.2023.00062

However, off-device computation leads to communication overhead and privacy concerns. Cameras need to transmit collected data to cloud servers, leading to high latency when authentication is frequently requested and increasing the risk of malicious attacks during data transmission. While encryption is a promising approach to keep data safe from leakage, it introduces high computational overhead, violating the original intention of off-device authentication systems.

Thus, a hybrid authentication system is needed that i) does not require large on-board storage space; ii) does not induce heavy on-device computation; iii) protects data from leakage.

Our Approach: We propose a novel authentication system called FaceGroup to address the challenges of current facial authentication methods. The system has three stages: i) the enrollment stage, ii) the group authentication stage, and iii) the individual authentication stage. Fig. 1 illustrates the system paradigm. During the enrollment stage, authorized personnel's images are preprocessed, encrypted, and stored on the server. To reduce computational costs, we propose a group testing mechanism inspired by biology and medical experiments [5], [6] in facial recognition, which enables batch face authentication. Furthermore, we propose a morphing neural network that merges multiple faces into one feature representation while preserving the characteristics of each face, reducing computation overhead and making off-device encrypted facial authentication practical for resource-constrained devices in real-life applications such as smart cities and access control.

Once the group authentication stage detects an authorized face, the individual authentication stage determines which face is authorized. To protect against untrusted cloud servers, we adopt the Homomorphic Encryption (HE) approach [7] which allows mathematical operations to be applied to encrypted data and correctly decrypted. However, Fully HE (FHE), which supports both multiplication and addition, induces significantly high computational overhead. As FHE is computationally expensive, we use **Partially HE (PHE)**, which supports only one of the two operations but is more efficient. To enable the use of FHE in facial authentication, we propose a Manhattan distance-based approach that relies only on additions to compare query faces with stored faces.

Our *contributions* are summarized as,

- We propose a novel lightweight privacy-preserving cloud-based facial authentication method that utilizes PHE and significantly reduces computation costs.
- We propose a novel lightweight deep learning architecture to achieve group testing in facial authentication. To the best of our knowledge, we are the first work that achieves group testing in face authentication.
- We proposed a comprehensive group testing framework for face authentication, which can also be extended to other biometric authentication methods (e.g., fingerprint). It covers group testing procedures, design choices based on deployment needs, and security measurements.
- We evaluate our method across multiple platforms, datasets, distance metrics, and models to prove its effectiveness and robustness and show group testing can save 55% computation.

Paper Outline: First of all, in Sect. II, we introduce the related works in the fields of Homomorphic Encryption, continual face authentication, and group testing and compare our work with them. Secondly, we present our novel facial authentication procedure and facial group testing architecture in detail in Sect. III. Then, in Sect. IV, we extensively evaluate our proposed work under different platforms, datasets, and face feature extractors to show its scalability, effectiveness, and robustness. Finally, in Sect. V, we summarize our contributions and discuss future research directions.

II. Related Works

While existing works have focused on HE and facial authentication at a fundamental algorithmic level, few papers examine both aspects in conjunction. The key takeaway is that conventional approaches do not provide attractive compromises between accuracy, time, and security. In this section, we compare our work with related works and show our advantages over them w.r.t. computation cost and accuracy.

Siamese Network: Bertinetto et al. [8] introduced fully-convolutional Siamese Networks in 2016. A fully-convolutional Siamese Network is a structure that uses a pair of images as inputs. Then, the inputs are passed into the same Convolutional Neural Network (CNN) to generate two feature vectors. After, a loss is calculated based on the distance between the two vectors and the input labels. Intuitively, two similar samples should have similar feature vectors while two dissimilar samples should have distinct feature vectors, and the similarity is represented by the distance. The distance is generally calculated based on the Euclidean metric, but little study has been done on incorporating Manhattan distance with Siamese Network. In another work [9], the authors used Manhattan Distance in Long Short-Term Memory (LSTM) network in Natural Language Processing (NLP). However, they used Manhattan Distance only from a performance perspective and made no further investigation. In our work, we extensively experiment and analyze the differences between Euclidean distance and Manhattan distance w.r.t. accuracy, time cost, Siamese Network, HE, and practicability and show that Manhattan distance can achieve similar performance as Euclidean distance does in Siamese network training.

Group Testing: Aldridge et al. [10] gives an overview of the history of group testing and its applications. In general, group testing aims to reduce the number of tests needed to identify defectives, and group testing has a wide range of applications. The principle behind group testing is that we examine a subset of samples at once, and the exam result can tell whether there is a defect in the samples or not. According to Dorfman et al. [11], group testing can significantly reduce testing costs when the defect rate is low. Similarly, applying group testing in facial authentication can greatly reduce the number of tests required for identifying intruders, as the chances of encountering one are low. Kim et al. [12] introduced GroupFace, a face authentication framework that represents individuals using features from multiple groups and their own features. While resembling a group testing algorithm, GroupFace is primarily a feature extraction method that utilizes self-extracted features and

measures similarities to groups represented by latent variables. However, GroupFace suffers from scalability issues due to the need for numerous groups to represent a reasonable number of identities. Determining an optimal number of groups for deployment becomes challenging. Additionally, adding an extra group requires additional fully connected layers, resulting in a model with a high parameter count, making GroupFace less flexible and scalable. In our work, we propose a novel approach that combines multiple face images into a single sample for lightweight and scalable face verification. In our work, we proposed the first and novel approach that morphs multiple face images together as a sample for face verification, which is lightweight and scalable.

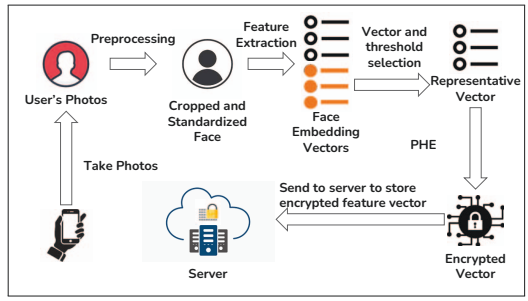
Homomorphic Encryption: In 2009 Gentry pioneered the first general fully HE scheme that can perform an arbitrary number of additions and multiplications [13]. While earlier implementations of HE suffered from performance drawbacks, recent advances in performance, usability, and rapid development have made it practical in certain applications [14]. On the other hand, Pascal Paillier invented the Paillier Cryptosystem (PC) that satisfies the properties of a PHE scheme. Therefore, PC is also known as Paillier Homomorphic Encryption supporting only addition operation on the ciphertext. In our work, we extensively experiment PC with the latest FHE schemes on different datasets and feature extractors to show the effectiveness of PC or PHE in general. Most importantly, although there are works done with HE, our work successfully brings HE into real-time processing on resource-constrained devices.

III. Proposed Work

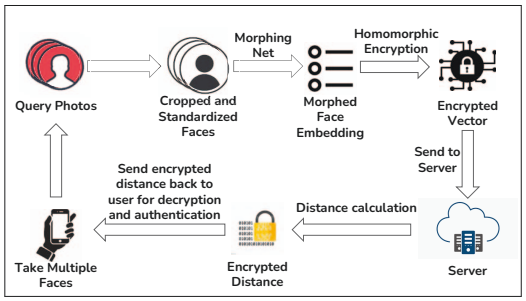
In this section, we introduce the proposed FaceGroup system, which comprises three stages: i) enrollment (Sect. III-A), ii) group verification (Sect. III-B), and iii) individual verification (Sect. III-C). In the enrollment stage, photos of individuals are collected and processed, PHE-encrypted, and stored on the cloud server before system deployment. Then, in the group verification stage, multiple photos are morphed and compared with the enrolled faces to detect if any individual of interest is present. The use of group testing in this stage significantly reduces computational costs while maintaining acceptable levels of accuracy. If an individual of interest is detected, the system proceeds to the individual verification stage, which employs a higher-accuracy yet more computationally intensive detection model to perform detection for each individual face. *To the best of our knowledge, we are the first work that achieves group testing in face authentication.*

A. Enrollment

Our authentication system, like many others, requires an initial enrollment phase (see Fig. 2a) to process and transmit the faces of authorized personnel to the server. During this phase, we capture several photos of users under different poses, crop them to highlight the facial regions and standardize them to mitigate camera noise. We choose an image standardization method that aligns with the preprocessing method used by the machine-learning-based feature extractor used in the



a



b

Fig. 2: Diagrams for (a) the enrollment phase, where faces of authorized personnel are processed and stored in the server; and (b) the authentication phase, which consists of group authentication and individual authentication.

latter phase, enabling faster training and convergence. Finally, we pass these pre-processed photos through the proposed anchor neural network, which is a deep neural network-based feature extractor that generates face embeddings, i.e., vector representations of the faces that are superior to the original photos for the downstream authentication step. Unlike traditional hand-crafted feature extraction methods [15], [16], FaceGroup's feature extractor is flexible and robust to outliers. The anchor network produces a feature embedding on each face, which is then encrypted to prevent data leakage when uploading the face embeddings of the persons of interest to the server. As discussed in Sect. I, we use PHE schemes to perform computation on encrypted data without incurring high computational costs. At the end of the enrollment phase, the encrypted face embeddings are stored on the cloud server and compared with any incoming face embeddings to determine if the incoming face is among the enrolled ones. To add new authorised personnel into the proposed system, instead of re-training/fine-tuning like conventional methods, the to-be-added personnel will go through this enrollment stage and upload the extracted feature vector with a proper label for future authentication purpose.

While PHE can lower the computational cost of using FHE, performing facial authentication on each individual face sequentially can still be computationally demanding, especially when there are numerous faces to verify. To address this issue, we will introduce group facial verification in the next section, which can further reduce the computational burden.

B. Group Facial Authentication

Kim et al. [12] made an initial attempt at group testing. However, their work lacked scalability and flexibility, making it impractical. To address these challenges, our research introduces an extended Siamese neural network [17], which enables the authentication of multiple individuals using a single detection. The objective of group testing is to achieve comparable results to conventional facial authentication while minimizing the number of required tests.

Architecture Design: The objective of our proposed architecture is to determine whether a given individual is a member of the enrolled identities. In order to achieve accurate results with group testing, we must find a practical and efficient method to combine multiple face images into a single feature embedding. Unlike liquids, images are not readily soluble, and there is no established mathematical model or theorem for merging multiple images while retaining their unique characteristics. To address this challenge, our approach, FaceGroup, employs a data-driven methodology that employs a deep learning model to extract features from N face images in a single forward pass. To tackle this challenge, we introduce a twin-neural network structure, also known as the Siamese network, as illustrated in Fig. 3a. The Siamese network comprises three components: i) a morphing network, ii) an anchor network, and iii) a classifier network. The morphing net is used to extract joint features from N face images of the incoming faces, while the anchor network extracts features of the person we are interested in. The pairwise distance between the two feature embeddings is then forwarded to the classifier to obtain a binary detection outcome. Both the morphing and anchor networks have the same architecture, differing only in their input layer configurations.

In general, an RGB image has a size of $(Channels \times Height \times Width)$, where $Channels = 3$. In a Convolutional Neural Network (CNN), kernels extract features from input images using a sampling window that has a size of $(Channels \times Kernel\ Height \times Kernel\ Width)$, sliding over the images. Our objective is to extract common features that can effectively represent N images simultaneously. Drawing inspiration from signal superposition, where kernels compute features from all channels, stacking face images at the channel dimension strengthens common features while averaging out outlier values. Thus, we stack N images on the channel dimension to enable the kernels to compute joint features of N images. However, blindly stacking images without alignment does not work, so we use [18] to crop, resize, and align the faces before stacking them together. This ensures that the stacked faces are aligned and fit our intuition. Once the faces are stacked, the input size for the morphing net is $(3N \times Height \times Width)$ for N face images, and for the anchor net, it is $(3 \times Height \times Width)$. However, the trained morphing net cannot work with a varying number of faces in the query images, which is a practical challenge since cameras may not always capture enough faces to form a fixed-size query image. To address this challenge, we use dummy images full of 0s as padding in query images. In Sect. IV, we conduct a comprehensive analysis of how the number of paddings affects the performance of our approach. Additionally, we provide a detailed guideline for choosing the query image size based on numerous experimental results.

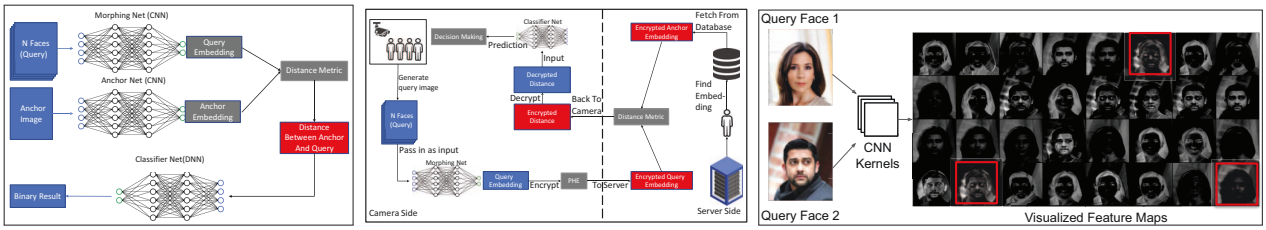
Deployment: Fig. 3b shows that during deployment, only the morphing net is deployed on the cameras, which capture incoming faces for authentication. To get the faces in a camera frame, we use existing face extraction techniques [18]. The extracted faces are then morphed by the morphing net and encrypted by PHE. The cameras then pass the morphed and PHE-encrypted feature vectors, which are called query embeddings, to servers for distance calculation. As all distance results are encrypted, the server has no knowledge of the identities involved, and the cameras have no access to authorized personnel data. This approach significantly enhances privacy preservation by blocking knowledge sharing between the cameras and servers.

During the authentication process, the camera sends query embeddings along with a query label that specifies the access level needed to gain access. Then, the server fetches feature vectors with labels that satisfied the query label from the database, and the fetched vectors are called anchor embeddings. The classifier outputs the probability that the identity corresponding to the anchor embedding is a member of the corresponding identities of query embeddings. Since the output is a probability, we can set a threshold to control how we interpret the probability. For example, most binary classifiers use a threshold of 0.5, where any output greater than or equal to 0.5 is considered positive and any output less than 0.5 is considered negative. However, lowering the threshold from 0.5 to 0.4 can increase the False Positive Rate (FPR) and decrease the True Positive Rate (TPR). In group testing, FPR is more important than TPR, which may sound counterintuitive, because it increases the probability of detecting a positive sample. The objective of group testing is not to obtain an exact detection result, but to eliminate negative samples as much as possible beforehand to reduce the total number of individual facial authentication. We provide a more detailed analysis of the affects brought by varying the threshold in Sect. IV.

C. Individual Facial Authentication

The individual facial authentication stage comprises two parts: first, we select a random vector as the representative vector, and second, we calculate a threshold that is the upper bound of the $x\%$ confidence interval based on the distances from the representative vector to all other enrolled face vectors. We discuss the choice of x in detail in Sect. IV. By using the $x\%$ confidence interval, we ensure that the threshold is less influenced by extreme or corner cases from the input images, leading to a lower False Positive Rate (FPR).

After encryption, the vector is ready to be stored on a cloud server, while the threshold is saved locally for future reference. As part of the authentication process, the server needs to compute the similarity score between the query face embedding and the enrolled face embeddings. Various norm metrics can be used for this purpose, with $L2$ Euclidean norm being a popular choice. For $L2$ norm, the distance between



a b c

Fig. 3: (a) Neural networks used in FaceGroup. (b) Computational diagram for FaceGroup. The faces of a group of faces are taken. Then, they are turned into one feature vector, which is encrypted by PHE. In the meanwhile, the server fetches the feature vector of the person of interest/authorized personnel for distance calculation. Lastly, the computed distance is sent back to the camera for decryption and decision-making. (c) The visualized kernel outputs from the first layer of the morphing net. We can observe that the morphing net is trying to morph the two faces. This is most obvious in the highlighted faces.

2 vectors $a, b \in \mathbb{R}^n$ can be calculated as, $D_{Euc}(a,b) = \sqrt{\sum_{i=1}^{n}(a_i - b_i)^2}$, where a_i and b_i are the ith element in vectors a and b. However, as Euclidean distance requires simultaneous addition and multiplication, which can only be performed using FHE, it is not compatible with our framework. In contrast, Manhattan distance only involves addition, making it a good candidate for our approach. The calculation of the Manhattan distance between the two vectors $a, b \in \mathbb{R}^n$ can be represented as, $D_{Manh}(a,b) = \sum_{i=1}^{n} |a_i - b_i|$.

From the security perspective, one potential security concern is that an attacker could send a dummy vector x, which they know, from a malicious camera. They could then use the returned distance vector z to recover feature vectors stored on the server by computing $z_i - x_i = y_i$. To address this challenge, we rearrange the elements of the vector z using a predefined order during the training process. This order is randomly generated but remains invariant throughout training, testing, and deployment. As a result, the classifier net is trained to implicitly adopt the predefined order, which is only known by the server. Therefore, even if an attacker hijacks a camera in our system, they cannot gain explicit knowledge about the correct permutation. To guess the correct order, an attacker would need $\frac{v!}{1*10^9}$ seconds, where v is the length of a feature vector even if the attacker can make $1 * 10^9$ guesses per second. To put the amount of time required to guess a correct permutation into perspective, when $v = 512$, the attacker needs $\frac{512!}{1*10^9} \approx 3.477 * 10^{1157}$ seconds or $1.102 * 10^{1150}$ years to try out all possibilities. Thus, it is infeasible for an attacker to obtain the correct rearrangement order from the camera side without leakage from the server.

We also argue that rearranging the positions of elements does not affect the performance of the classifier net, which is a multilayer perceptron (MLP), and threshold mechanism. Firstly, each layer in an MLP is represented by a matrix. When we focus on the first layer of the classifier net, the input feature vector z has shape vx1 and the output shape of the first layer is mx1, which v and m are chosen based on design choices. Then, the matrix representation M of the first layer is vxm. By calling the output vector as o, then $o_i = v \cdot M_i$ where o_i is the i^{th} entry of o and M_i is the i^{th} column of M. Therefore, if there is an exchange between the positions of z_i and z_j, we can get the same output by switching the positions of i^{th} and j^{th} rows of matrix M. Secondly, for the proposed threshold mechanism, the threshold is calculated as the sum of entries in z, the order does not affect the summation result. Therefore, the rearrangement of positions of entries in feature vector z works on both the classifier net and the threshold mechanism.

IV. Performance Evaluation

To show the practicability of our proposed architecture, we implement the proposed system and obtain results on different hardware. The results of the evaluation show that our proposed FaceGroup system is reliable, scalable, and practical in terms of computational cost and accuracy. Furthermore, we also evaluate our FaceGroup with different query sizes, padding sizes, and feature extractors. This section consists of the evaluation of two parts–the group authentication part (Sect. IV-A) and the individual authentication part (Sect. IV-B).

A. Group Authentication

By introducing group authentication, we aim to reduce the computational cost of performing individual authentication with acceptable sacrifice in accuracy. In this section, we will use results obtained from experiments to support our claim and show that group authentication can achieve a low False Negative Rate (FNR) so that we do not miss authorized persons at this stage.

Experiment Setup: We train our proposed group authentication model with CelebA Dataset [19], a dataset that consists of images of celebrities with various backgrounds, poses, lighting conditions, and noise levels. Before training, we use the feature extraction layers from a pre-trained Inception-Net V4 [20] model, a commonly used feature extractor, as the backbone of FaceGroup.

Results: In Fig. 4a, we observe a decrease in TPR and TNR as the number of faces in the query image increases. This is expected as it is more difficult to represent multiple individuals in a single feature vector with a fixed length. Additionally, our experiments indicate that models trained with Manhattan distance outperform those trained with Euclidean distance in terms of TPR, while the models trained with Euclidean distance outperform those trained with Manhattan distance in terms of TNR. Despite the query size increasing from 2 to 10,

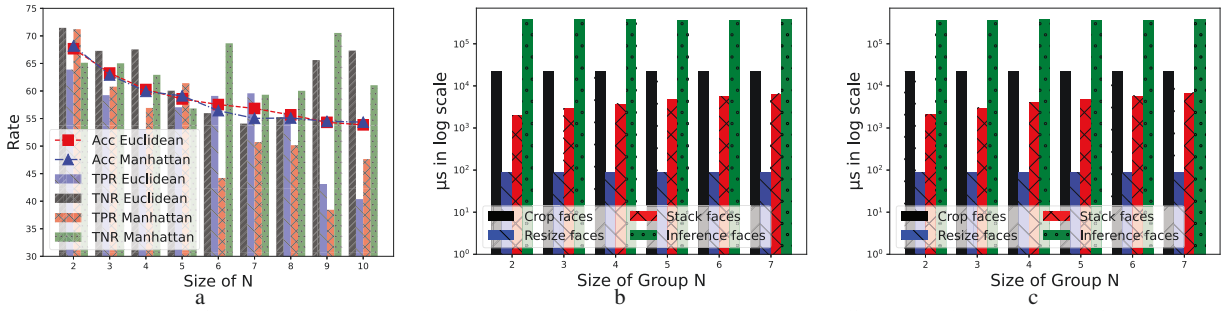


Fig. 4: (a) True Positive Rate (TPR) and True Negative Rate (TNR) under Euclidean and Manhattan distance metrics; Computational time in with (b) Euclidean distance and (c) Manhattan distance.

the accuracy of models trained with different distance metrics remains within an acceptable range. However, as the query size reaches 9 or 10, the ratio of TPR to TNR changes significantly compared to the ratios observed when the query size is less than 9. This is because of the insufficient kernels in the first input layer to capture the joint features of all faces in the query image. To ensure that our framework is indeed capturing joint features, we visualized the kernels from the first layer of a trained morphing net with a query size of 2 in Fig. 3c. The figure shows that the morphing net represents input faces as a single face while preserving the features of each individual face. For example, some kernels highlight the areas where the eyes overlap, while others highlight the hair areas. By observing these visualized kernels, we can quickly identify that there are two identities in the query image. Therefore, our proposed morphing network can capture not only joint features but also the unique characteristics of each individual.

Even though our models can capture the joint and unique features of each individual, each trained model can only work with a specific size of query images (the solution to this will be discussed in Sect. V). Therefore, in the cases of insufficient faces to construct a query image, we propose using dummy images filled with 0s as paddings. We evaluate our proposed model with query sizes ranging from 2 to 10 and padding sizes ranging from 1-4, as shown in Fig. 5. The figure shows that when the query size N is smaller than 7, a padding size $P >= N - 2$ can positively affect the accuracy. Moreover, if $N >= 7$, we can see that a padding size $P >= 3$ can increase the accuracy. However, when $P = N - 1$, there is no need to use padding as there is only one sample in the query image. Thus, we can use the individual face authentication method in such scenarios. If we set the possibility of using different padding sizes to be equal, the query size of 4 will have the highest expected positive effects on accuracy.

Lowering the threshold used to interpret the probability output from the classifier can solve the high FNR problem. Fig. 5c shows that decreasing the threshold decreases the FNR. Thus, controlling the threshold can control the FNR in our group authentication system. Although lowering the FNR may lead to a higher FPR, a reasonable reduction in the FNR can lower the expected number of tests. Before calculating the expected number of tests needed, we want to introduce a few terms: sensitivity S_e, specificity S_p, and prevalence p. *Sensitivity* is the probability that a specimen is tested to be positive when the specimen is exactly positive in a group test. On the other hand, *specificity* is the probability that a specimen is tested to be negative when the specimen exactly is negative in a group test. In other words, S_e is the TPR of a group testing, and S_p is the TNR of a group testing. Lastly, prevalence is the assumed ratio of the total number of positive specimens in a given population. Then, the expected number of tests per person can be written as,

$$E(\mathrm{D2}) = \frac{1}{n} + \mathbb{P}_n,\ \mathbb{P}_n = (1 - S_p)\,(1-p)^n + S_e\,(1 - (1-p)^n)\,, \tag{1}$$

in which n is the group size. In a simple example, we assume that there are 100000 people in a smart city, and there are 100 persons of interest out of these 100000 people. Furthermore, we assume that the second test is 100% accurate. Therefore, the prevalence is $100/100000 = 0.001$. By using the data obtained from Fig. 4a, we get that $S_e \approx 0.56$ and $S_p \approx 0.55$ when $n = 8$ and being trained with Euclidean distance. Thus, the expected number of tests needed per person is ≈ 0.45. Thus, the total number of tests needed to authenticate all 10 persons is $0.45 * 100000 = 45000$. This result is significant because we reduced the computation cost by 55% in this example by applying our proposed face group authentication method. Although group testing has a higher computation cost than FHE facial authentication, we reduced the number of tests needed, resulting in lower total computation costs. Additional parameters introduced by increasing the morphing net's n value are trivial when n is reasonably small, given the large number of parameters in the anchor net.

By reducing the number of tests needed, we also reduced the communication rounds required between cameras and servers. This 55% reduction in computation costs can also be applied to communication costs. Reducing communication costs is crucial in our proposed work, particularly in smart cities and smart buildings where there will be a high number of queries issued each second. Neglecting communication costs can lead to internet traffic congestion, especially for wireless connections that offer limited bandwidth. Thus, the 55% reduction in communication costs enhances the feasibility and scalability of our privacy-preserving group testing framework.

To show our architecture's real-time performance, we recorded the time for each step in FaceGroup in Fig. 4. As query size increases, the time taken for stacking faces increases, but inference time remains stable due to negligible

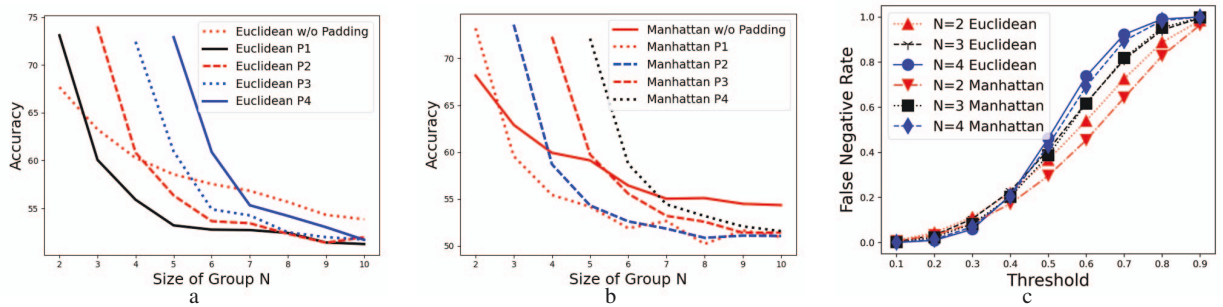


Fig. 5: The impact of different padding sizes on accuracy with (a) Euclidean and (b) Manhattan distances; (c) False Negative Rate (FPR) against the threshold value. When the threshold is increasing, the false negative rate increases.

TABLE I: Pictures and specifications on the three types of hardware we tested. The most powerful one is the workstation while the least powerful one is the Raspberry Pi.

Hardware	Workstation	Laptop	Raspberry Pi
Picture			
CPU	Intel i9-10900KF	AMD Ryzen 3 3200U	Broadcom BCM2837B0
CPU Frequency	3.7GHz	2.6GHz	1.4GHz
GPU	Nvidia RTX 3080	N/A	N/A

parameter increase. A complete cycle from capturing images to inferring results takes less than 500ms, making our framework real-time. We recommend equipping each camera with two models, N=4 and N=8, for optimal query size selection. The N=4 model is used for padding, and the N=8 model saves computation costs while maintaining accuracy when padding is not needed.

B. Individual Authentication

In the previous subsection, we can observe that group authentication achieves a good FNR so that the system does not miss authorized persons and the computational cost is low. In this subsection, we will show that individual authentication can achieve higher accuracy on authentication but with a much higher computational cost.

Experiment Setup: To show the computational performance of the proposed system, we run the algorithm on three different hardware–i) a powerful workstation, ii) a less powerful laptop, and iii) a Raspberry Pi 3 Model B+. The detailed specifications can be found in Tab. I. For FHE implementation, we used Pyfhel, which is a python implementation of Microsoft SEAL [21] that wraps the CKKS FHE scheme [22]. We used the python-paillier library [23] as Paillier PHE implementation.

Impact of Different Distance Metric: We evaluate how the interchange of Euclidean and Manhattan distance metrics affects our proposed work's accuracy and feature extractors in the training phase, using Labeled Faces in the Wild [24] and WebFace Dataset [25]. Both of the datasets cover face images with different illumination, poses, background, and noise conditions. Each iteration includes 20 test images per identity and enrollment sets with k images per identity, ranging from 2 to 9. We use two pre-trained machine learning models, VGG16 [26] and Inception Net V1 [27], which output feature vectors of 512 and 128 bits, respectively.

The results are shown in Figs. 6 and 7. Fig. 6b demonstrates the TPR of using the Manhattan distance and Euclidean distance in the Inception Net V1 and VGG16 with the LFW Dataset. At first glance, Fig. 6b does not make any sense because, with more enrolled samples, the TPR goes down. However, in deep, this has to do with how we choose the threshold. In our design, the threshold is the upper bound of the 90% confidence interval of the distances calculated in the enrollment phase. With a better choice of the threshold, our system can achieve a TPR over 90% with FPR close to 0%, as shown in Fig. 7b. Also, Fig. 7a shows that our choice of threshold can reduce the FPR. Because Inception Net V1 was trained with a face identification dataset and VGG16 was trained with an object detection dataset, Inception Net V1 is much better than VGG16 in the face authentication area no matter which distance metric we use. Furthermore, Manhattan distance outperforms Euclidean distance on both datasets with such a powerful feature extractor. Figs. 6a and 6c confirm our assumption that Manhattan distance is comparable to Euclidean distance even if the performance of the feature extractor is ordinary. The takeaway is that the accuracy of our proposed system is not significantly affected by the distance metric. Rather, the accuracy of our proposed system is mainly affected by the chosen feature extractor. To further justify our assumption, we use ROC curves to demonstrate how the TPR and FPR are affected by thresholds in Fig. 7. From Fig. 7, the performance difference between using Manhattan and Euclidean Distance is acceptable.

HE Computation Cost: To justify our choice of using Manhattan distance w.r.t. computation cost and HE schemes, we conducted experiments with three groups: Euclidean distance with FHE, Manhattan distance with FHE, and Manhattan distance with PHE. We used two feature extractors, VGG16 and Inception Net V1, to extract feature vectors from 30 randomly chosen images from LFW Dataset in each group and recorded the average encryption, distance calculation, and decryption time. We run this same experiment on Raspberry Pi, laptop, and desktop testbeds to further justify our design.

The result is shown in Fig. 8 and Tab. II. The figure is divided into six groups–(i) Manhattan with Inception Net V1

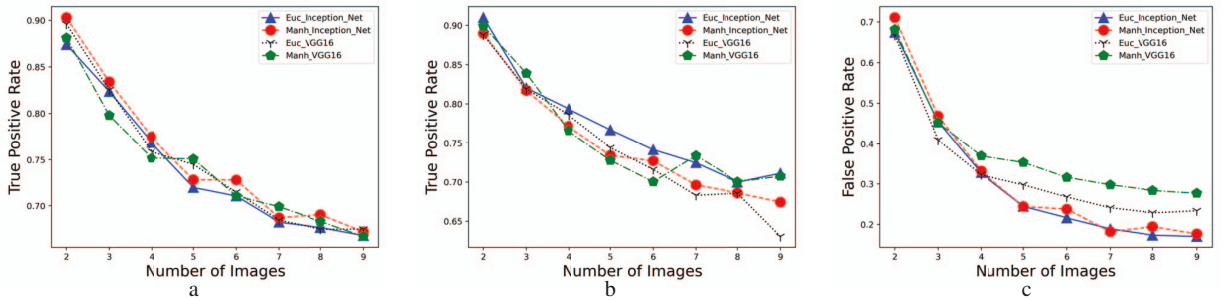


Fig. 6: True Positive Rate (TPR) in (a) WebFace dataset and (b) LFW dataset with different metrics and feature extractors; (c) False Positive Rate (FPR) in WebFace Dataset with different metrics and feature extractors.

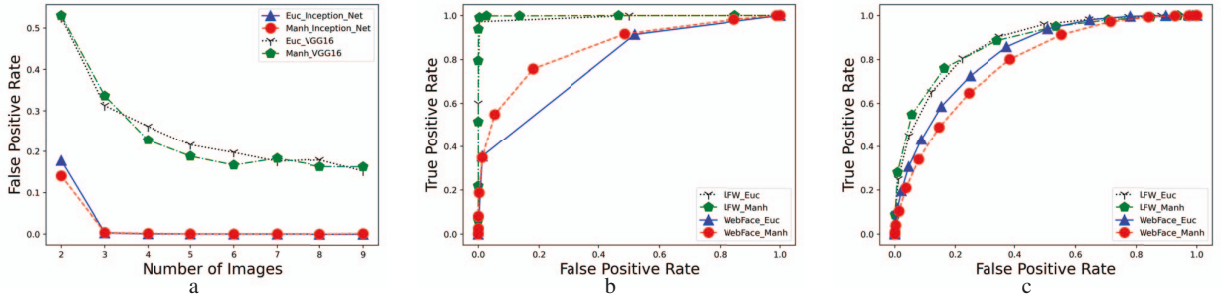


Fig. 7: (a) FPR in LFW Dataset with different metrics and feature extractors; (b) Receiver Operating Characteristic (ROC) curve of using Inception Net V1; (c) ROC curve of using VGG16.

TABLE II: Encryption, distance calculation, decryption time in million seconds (ms) with different configurations recorded on three hardware–a powerful workstation (we call it PC in this table), a less powerful laptop, and a Raspberry Pi 3 Model B+ (RPi in the table). Running FHE on the Raspberry Pi exceeds the memory limit and therefore we put N/A here.

Feature Size	Encryption Method	Distance Metric	Execution Time (ms)								
			Encryption			Distance Calculation			Decryption		
			PC	Laptop	RPi	PC	Laptop	RPi	PC	Laptop	RPi
128	PHE	Manh	2.64	5.08	102.04	1.34	2.59	78.30	1.11	1.54	55.55
128	FHE	Manh	109.92	5532.05	N/A	0.94	2.19	N/A	13.95	20.51	N/A
128	FHE	Euc	105.70	5465.96	N/A	120.73	225.62	N/A	14.83	25.91	N/A
512	PHE	Manh	487.80	776.92	430.71	8.49	10.72	322.13	4.17	4.27	211.28
512	FHE	Manh	19958.69	1046582.70	N/A	5.70	7.76	N/A	51.50	77.51	N/A
512	FHE	Euc	19941.31	1046540.76	N/A	481.95	898.93	N/A	58.64	99.79	N/A

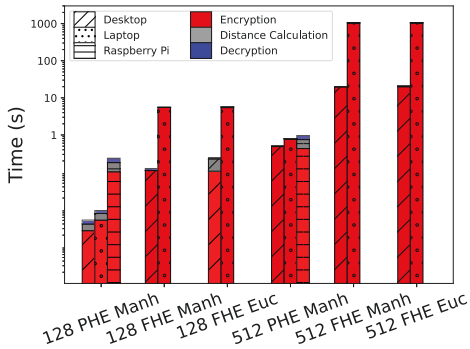


Fig. 8: The total time taken by the three phases with different combinations of HE schemes, distance metrics, and feature extractors profiled on the desktop, laptop, and Raspberry Pi. Running FHE on the Raspberry Pi exceeds the memory limit, leaving the corresponding bar blank.

and PHE, (ii) Manhattan with Inception Net V1 and FHE, (iii) Manhattan with VGG16 and FHE, (iv) Manhattan with VGG16 and PHE, (v) Euclidean with Inception Net V1 and FHE, and (vi) Euclidean with VGG16 and FHE. There are two interesting observations from Fig. 8. First of all, no matter which encryption method we use, the distance calculation time is almost identical across the Manhattan distance groups, indicating Manhattan distance can significantly reduce the computation overhead caused by the HE schemes. Secondly, the encryption and decryption time is consistent across groups using the same encryption method, regardless of the feature extractor chosen. This increases the flexibility of our system for engineers to customize feature extractor selection without affecting encryption and decryption time. Moreover, PHE can handle 512 bits feature vectors within 0.5 seconds, while FHE can take up to 20 seconds, limiting its scalability. Thus, our work is more scalable than cloud-based facial authentication implementations using FHE.

As for results for laptops, although we can see a huge jump in encryption, distance calculation, and decryption time

compared with results recorded on the desktop, the run time of PHE is at the sub-second level. In comparison, FHE roughly takes 5.5 seconds to encrypt a 128 bits-long feature vector and over 1000 seconds or 17 minutes to process a 512 bits-long feature vector, which is not practical in any sense. From the perspective of time complexity, PHE roughly doubled the time taken from the desktop setup to the laptop setup. In contrast to PHE, FHE roughly takes more than 50 times the original time taken from the desktop to the laptop. Our proposed work is also feasible on resource-constrained devices, as PHE completes the work cycle at a sub-second level, while FHE fails due to insufficient memory space. Also, Manhattan distance with FHE takes significantly less time to calculate than Euclidean distance. These results show that our proposed work applies to hardware with very limited computation resources, and this exciting finding further endorses our choice of Manhattan distance over Euclidean distance.

In conclusion, on average, group authentication has worse accuracy than the individual authentication system, regardless of the distance metrics. On the other hand, group authentication can process face samples in batches, which significantly reduces the number of tests needed. Furthermore, to reduce the chances of falsely classifying a positive sample as negative, we can change the classification threshold to control the false negative rate. As a result, easy-to-classify samples are pre-screened by the group testing stage while hard-to-classify samples will be tested by the more accurate individual face authentication stage.

V. Conclusion and Future Works

We have shown feasibility for key parts of our continual facial authentication architecture. We have also demonstrated using existing PHE is more than capable of processing real-time comparisons of facial feature matrices. The results show that we can process a single verification in under a second. Moreover, we have shown the potential of Manhattan distance in machine learning w.r.t. model training, computation cost, and flexibility. Most importantly, we have shown that group testing in facial authentication with practicability, flexibility, and scalability is practical. Our trained group testing model can save at least 55% computation cost.

For future work, we plan to i) introduce transformer architecture [28] into the morphing network so that we do not need to train one model for different numbers of faces; ii) conduct in-the-wild experiments to collect performance data (e.g. computation and storage requirements) from real-world environments; iii) do an in-depth analysis of how our proposed work can withstand Byzantine attacks.

References

[1] S. Rose, O. Borchert, S. Mitchell, and S. Connelly, "Zero trust architecture," 2020-08-10 04:08:00 2020.

[2] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*, pp. 1273–1282, PMLR, 2017.

[3] C. Sun, T. Jiang, S. Zonouz, and D. Pompili, "Fed2kd: Heterogeneous federated learning for pandemic risk assessment via two-way knowledge distillation," in *2022 17th Wireless On-Demand Network Systems and Services Conference (WONS)*, pp. 1–8, IEEE, 2022.

[4] R. S. Sandhu and P. Samarati, "Access control: principle and practice," *IEEE communications magazine*, vol. 32, no. 9, pp. 40–48, 1994.

[5] C. Gollier and O. Gossner, "Group testing against covid-19," tech. rep., EconPol Policy Brief, 2020.

[6] M. Aldridge, O. Johnson, J. Scarlett, *et al.*, "Group testing: an information theory perspective," *Foundations and Trends® in Communications and Information Theory*, vol. 15, no. 3-4, pp. 196–392, 2019.

[7] C. Fontaine and F. Galand, "A survey of homomorphic encryption for nonspecialists," *EURASIP Journal on Information Security*, vol. 2007, pp. 1–10, 2007.

[8] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, "Fully-convolutional siamese networks for object tracking," 2016.

[9] J. Mueller and A. Thyagarajan, "Siamese recurrent architectures for learning sentence similarity," in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, p. 2786–2792, AAAI Press, 2016.

[10] M. Aldridge, O. Johnson, and J. Scarlett, "Group testing: An information theory perspective," *Foundations and Trends® in Communications and Information Theory*, vol. 15, no. 3–4, p. 196–392, 2019.

[11] R. Dorfman, "The detection of defective members of large populations," *The Annals of mathematical statistics*, vol. 14, no. 4, pp. 436–440, 1943.

[12] Y. Kim, W. Park, M.-C. Roh, and J. Shin, "Groupface: Learning latent groups and constructing group-based representations for face recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5621–5630, 2020.

[13] C. Gentry, *A Fully Homomorphic Encryption Scheme*. PhD thesis, Stanford, CA, USA, 2009.

[14] A. Acar, H. Aksu, A. S. Uluagac, and M. Conti, "A survey on homomorphic encryption schemes: Theory and implementation," *ACM Computing Surveys (Csur)*, vol. 51, no. 4, pp. 1–35, 2018.

[15] Z. Xu, J. R. Bradley, and D. Sinha, "Latent multivariate log-gamma models for high-dimensional multitype responses with application to daily fine particulate matter and mortality counts," *The Annals of Applied Statistics*, vol. 17, no. 2, pp. 1175–1198, 2023.

[16] H. Wang, J. Hu, and W. Deng, "Face feature extraction: A complete review," *IEEE Access*, vol. 6, pp. 6001–6039, 2018.

[17] G. Koch, R. Zemel, R. Salakhutdinov, *et al.*, "Siamese neural networks for one-shot image recognition," in *ICML deep learning workshop*, vol. 2, p. 0, Lille, 2015.

[18] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[19] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

[20] C. Szegedy, S. Ioffe, and V. Vanhoucke, "Inception-v4, inception-resnet and the impact of residual connections on learning," *CoRR*, vol. abs/1602.07261, 2016.

[21] "Microsoft SEAL (release 3.6)." https://github.com/Microsoft/SEAL, Nov. 2020. Microsoft Research, Redmond, WA.

[22] J. H. Cheon, A. Kim, M. Kim, and Y. Song, "Homomorphic encryption for arithmetic of approximate numbers," in *Advances in Cryptology – ASIACRYPT 2017* (T. Takagi and T. Peyrin, eds.), (Cham), pp. 409–437, Springer International Publishing, 2017.

[23] C. Data61, "Python paillier library." https://github.com/data61/python-paillier, 2013.

[24] G. B. Huang, M. Mattar, H. Lee, and E. Learned-Miller, "Learning to align from scratch," in *NIPS*, 2012.

[25] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Learning face representation from scratch," 2014.

[26] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015.

[27] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," 2014.

[28] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.