

# 저작권양도서

## (Copyright Transfer Form)


소속 : 인하대학교 정보통신공학과


성명 : 서지형, 홍준성, 박준형 학번 : 12191778, 12181852, 12191768


논문제목 : 동형암호 기반 원스톱 모바일 얼굴 검증 출입 통제 시스템

본인은 상기 논문을 2024학년도 1학기 정보통신프로젝트 최종 보고서 겸 결과 논문으로 제출하고자 합니다. 본 논문의 내용은 저자가 직접 연구한 결과인 것과 이전에 출판된 적이 없음을 확인합니다. 또한 공저자와 더불어 인하대학교 정보통신공학과에서 발간하는 논문집에 본 논문을 수록하는 것을 허락하며 제반 저작권을 정보통신공학과에 양도합니다.

2024년 7월 11일

주저자 : 서지형 

공저자 : 홍준성 

공저자 : 박준형 

정보통신공학과장 귀하

# 동형암호 기반 원스톱 모바일 얼굴 검증 출입 통제 시스템

## One-stop Mobile Face Verification Access Control System Based on Homomorphic Encryption

서지형, 홍준성, 박준형  
(Jihyeong Seo, Junseong Hong and Junhyung Park)

**요약:** 일반적으로 FaceNet을 활용한 동형암호 기반의 얼굴 검증 시스템은 사용자가 직접 본인의 비밀키를 가지고 암호화되어 있는 벡터 간 거리를 복호화하여 유효성을 판단한다. 하지만 서버 측에서 사용자 얼굴의 유효성을 필요로 하거나, 검증을 시도하는 단말기의 신뢰성을 충분히 확보하지 못할 때, 유효성 여부가 조작될 수 있는 취약점이 발생한다. 따라서 본 논문에서는 서버 측에서 암호문을 사용자에게 송신할 때, 사용자가 인위적으로 얼굴의 유효성 여부를 조작하지 못하도록 하는 마스킹 기법을 제안하고자 한다. 이를 통해 사용자의 단말기에 대한 신뢰도를 확보하고, 서버 별도의 인증 단말기 없이 개인 단말기만으로 얼굴 등록부터 검증까지 수행하여 출입을 통제할 수 있는 시스템을 구현하고자 한다.

**Abstract:** The homomorphic encryption-based face verification system using FaceNet determines validity by decrypting the encrypted vector distance with its own private key. However, vulnerability occurs when the server needs the validity of the user's face or when the reliability of the device attempting verification is not sufficiently secured, potentially leading to manipulated validity. Therefore, this paper proposes a method to prevent user from artificially manipulating the validity of its face when the server transmits the ciphertext to the user. By doing so, we aim to ensure the reliability of the user's device and implement a system that only the server can controls access using only the user's device, from face registration to verification, without the need to prepare additional authentication devices from the server side.

**Keywords:** Homomorphic Encryption, Face Verification, Mobile Device, Access Control



### I. 서론

일상 생활 속에서 생체 정보를 이용한 인증 기술의 비중은 점차 늘어나고 있다. 하지만 여러 생체 인증 기술 중에서 자주 사용되고 있는 얼굴 검증 시스템만 보더라도 얼굴 등록부터 검증 단계까지 중요한 개인정보인 얼굴 이미지가 외부로 쉽게 노출된다. 이에 따라 얼굴 이미지를 외부로 노출시키지 않고, 얼굴 검증을 수행할 수 있도록 하는 것이 중요한 과제가 될 것이다.

먼저 본 논문에서 사용한, 얼굴 검증을 위한 분류 모델 FaceNet은 얼굴 이미지로부터 128차원의 얼굴 특징 벡터를 추출하여 유클리드 공간 상에서 Triplet Loss를 이용해 훈련된 얼굴 임베딩 벡터 생성 모델이며, 동형암호는 복호화하지 않고 암호화된 상태로 연산할 수 있도록 하는 암호화 체계이다.

일반적으로 동형암호를 얼굴 검증 과정에 적용하고자 할 때, 임베딩 벡터를 암호화한 후 임베딩 벡터 간의 거리를 동형암호로 연산한다면, 서버 측에 얼굴 데이터를 공개하지 않고 얼굴의 유효성 여부를 얻어낼 수 있다. 하지만 이러한 방식은 사용자가 직접 벡터 간의 거리를 복호화 해 유효성을 판단해야 하므로, 서버 측에서 유효성을 검토하고자 할 때, 사용자를 전적으로 신뢰해야만 하는 문

제점이 발생한다.

따라서 본 논문에서는 서버 측에서 암호화된 벡터 간 거리를 사용자에게 송신할 때, 사용자가 인위적으로 얼굴의 유효성 여부를 조작하지 못하도록 하는 마스킹 기법을 제안하고, 이를 활용해 보안성을 유지한 채 얼굴 등록부터 검증 단계까지 모바일 단말기로 수행할 수 있는 출입 통제 시스템을 구현하고자 한다.

### II. 본론

#### 1. 동형암호

동형암호는 복호화 하지 않고 암호화된 상태로 연산할 수 있도록 하는 암호화 체계이다. 암호화된 상태로 계산된 결과는 암호화된 상태 그대로 유지되며, 비밀키를 가지고 암호화된 결과를 복호화 해 결과를 확인할 수 있다.

##### 1-1. CKKS

본 논문에서는 직접적으로 CKKS를 사용하지 않는다. 하지만 TenSEAL 라이브러리에서 tensor에 대한 암호화 방식을 CKKS 방식을 이용하여 사용한다. TenSEAL에서는 입력된 이미지를 행렬로 인코딩 한 후 vertical scan을 통해 하나의 암호문으로 암호화 한다. 이 때 CKKS 방식의 일괄처리 기능을 이용

# 동형암호 기반 원스톱 모바일 얼굴 검증 출입 통제 시스템

## One-stop Mobile Face Verification Access Control System Based on Homomorphic Encryption

서지형, 홍준성, 박준형  
(Jihyeong Seo, Junseong Hong and Junhyung Park)

**요약:** 일반적으로 FaceNet을 활용한 동형암호 기반의 얼굴 검증 시스템은 사용자가 직접 본인의 비밀키를 가지고 암호화되어 있는 벡터 간 거리를 복호화하여 유효성을 판단한다. 하지만 서버 측에서 사용자 얼굴의 유효성을 필요로 하거나, 검증을 시도하는 단말기의 신뢰성을 충분히 확보하지 못할 때, 유효성 여부가 조작될 수 있는 취약점이 발생한다. 따라서 본 논문에서는 서버 측에서 암호문을 사용자에게 송신할 때, 사용자가 인위적으로 얼굴의 유효성 여부를 조작하지 못하도록 하는 마스킹 기법을 제안하고자 한다. 이를 통해 사용자의 단말기에 대한 신뢰도를 확보하고, 서버 별도의 인증 단말기 없이 개인 단말기만으로 얼굴 등록부터 검증까지 수행하여 출입을 통제할 수 있는 시스템을 구현하고자 한다.

**Abstract:** The homomorphic encryption-based face verification system using FaceNet determines validity by decrypting the encrypted vector distance with its own private key. However, vulnerability occurs when the server needs the validity of the user's face or when the reliability of the device attempting verification is not sufficiently secured, potentially leading to manipulated validity. Therefore, this paper proposes a method to prevent user from artificially manipulating the validity of its face when the server transmits the ciphertext to the user. By doing so, we aim to ensure the reliability of the user's device and implement a system that only the server can controls access using only the user's device, from face registration to verification, without the need to prepare additional authentication devices from the server side.

**Keywords:** Homomorphic Encryption, Face Verification, Mobile Device, Access Control

## I. 서론

일상 생활 속에서 생체 정보를 이용한 인증 기술의 비중은 점차 늘어나고 있다. 하지만 여러 생체 인증 기술 중에서 자주 사용되고 있는 얼굴 검증 시스템만 보더라도 얼굴 등록부터 검증 단계까지 중요한 개인정보인 얼굴 이미지가 외부로 쉽게 노출된다. 이에 따라 얼굴 이미지를 외부로 노출시키지 않고, 얼굴 검증을 수행할 수 있도록 하는 것이 중요한 과제가 될 것이다.

먼저 본 논문에서 사용한, 얼굴 검증을 위한 분류 모델 FaceNet은 얼굴 이미지로부터 128차원의 얼굴 특징 벡터를 추출하여 유클리드 공간 상에서 Triplet Loss를 이용해 훈련된 얼굴 임베딩 벡터 생성 모델이며, 동형암호는 복호화하지 않고 암호화된 상태로 연산할 수 있도록 하는 암호화 체계이다.

일반적으로 동형암호를 얼굴 검증 과정에 적용하고자 할 때, 임베딩 벡터를 암호화한 후 임베딩 벡터 간의 거리를 동형암호로 연산한다면, 서버 측에 얼굴 데이터를 공개하지 않고 얼굴의 유효성 여부를 얻어낼 수 있다. 하지만 이러한 방식은 사용자가 직접 벡터 간의 거리를 복호화 해 유효성을 판단해야 하므로, 서버 측에서 유효성을 검토하고자 할 때, 사용자를 전적으로 신뢰해야만 하는 문

제점이 발생한다.

따라서 본 논문에서는 서버 측에서 암호화된 벡터 간 거리를 사용자에게 송신할 때, 사용자가 인위적으로 얼굴의 유효성 여부를 조작하지 못하도록 하는 마스킹 기법을 제안하고, 이를 활용해 보안성을 유지한 채 얼굴 등록부터 검증 단계까지 모바일 단말기로 수행할 수 있는 출입 통제 시스템을 구현하고자 한다.

## II. 본론

### 1. 동형암호

동형암호는 복호화 하지 않고 암호화된 상태로 연산할 수 있도록 하는 암호화 체계이다. 암호화된 상태로 계산된 결과는 암호화된 상태 그대로 유지되며, 비밀키를 가지고 암호화된 결과를 복호화 해 결과를 확인할 수 있다.

#### 1-1. CKKS

본 논문에서는 직접적으로 CKKS를 사용하지 않는다. 하지만 TenSEAL 라이브러리에서 tensor에 대한 암호화 방식을 CKKS 방식을 이용하여 사용한다. TenSEAL에서는 입력된 이미지를 행렬로 인코딩 한 후 vertical scan을 통해 하나의 암호문으로 암호화 한다. 이 때 CKKS 방식의 일괄처리 기능을 이용

하여  $N \times N$  행렬을  $N$  개의 암호문으로 암호화하는 방식을 차용한다 [1].

## 1-2. TenSEAL

Microsoft SEAL 기반으로 만들어진 동형암호 라이브러리 TenSEAL은 텐서 간의 동형암호 연산에 특화되어 있는 Python API이며, 본 연구에서 벡터 간의 거리를 구하는 연산에서 활용하기에 가장 적합하다 판단하여 TenSEAL을 사용하게 되었다. 본 논문에서는 TenSEAL을 이용하여 다음과 같은 기능을 구현하였다 [2].

### 1-2-1. 암호키 생성

CKKS 방식으로 동형암호화 context를 생성한 뒤, 컨텍스트를 TenSEAL의 serialize함수를 이용하여 직렬화 한다. 이 직렬화한 벡터로부터 공개키와 비밀키를 생성한다.

### 1-2-2. 임베딩 벡터 추출

얼굴 이미지에서 추출한 얼굴 특징 벡터를 암호화한 뒤, 직렬화(serialize)를 시행하였다.

### 1-2-3. 임베딩 벡터의 암호화

등록된 얼굴로부터 추출한 특징 벡터와 검증에 사용될 얼굴로부터 추출한 특징 벡터간 거리차이를 계산한 distance 데이터를 ckks\_vector 형태로 복원한다. 서버에서 ckks\_vector를 복호화하는 과정을 통해 클라이언트에서 전송한 사용자 ID와 인덱스를 비교하여 일치여부를 확인한다.

## 2. FaceNet

본 논문은 얼굴 영상을 고정 길이의 유클리드 공간 임베딩으로 변환하여, 동일인의 얼굴은 가깝게, 다른 사람의 얼굴은 멀리 매핑되도록 학습하는 FaceNet 모델을 사용하여 학습한다. FaceNet 모델의 가장 큰 특징은 Triplet loss인데, 이 방식이 앞에서 소개한 동일인의 얼굴은 가깝게, 다른 사람의 얼굴은 멀리 매핑하는 손실함수이다. triplet loss는 앵커(anchor), 양성 샘플(positive), 음성 샘플(negative)라는 3가지 샘플을 이용하여 함수를 진행한다. 앵커와 양성 샘플은 같은 클래스, 앵커와 음성 샘플은 다른 클래스이다. triplet loss 함수는 anchor와 양성 샘플 간 거리는 가깝게, 앵커와 음성 샘플 간 거리는 멀리 하도록 정의한다. triplet loss 함수의 수식은 다음과 같다 [3].

$$Loss = \max(|f(x_a) - f(x_p)| - |f(x_a) - f(x_n)| + margin, 0)$$

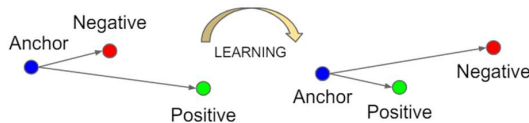


그림1. Triplet Loss

Python API를 이용하여 웹캠을 이용해 이미지를 불러온 뒤, 캡처된 이미지를 데이터 베이스에 저장한다. 이후 저장된 이미지에서 얼굴 이미지를 추출하고 Facenet 모델을 사용하여 얼굴특징

벡터(reg\_embedding)를 생성한다.

## 3. 정규화

이 구역은 본문에 관한 부분으로서 2단 형태로 정규화를 시행한다.

우선 등록된 얼굴로부터 추출한 특징벡터를 동형암호화한 벡터  $enc_{reg}$ 와 검증할 얼굴로부터 추출한 특징벡터를 동형암호화한 벡터  $enc_{v1}$  간의 거리를 L2 norm으로 계산한다.

$$dist = enc_{v1} - enc_{reg}$$

$$dist_{norm} = ||dist||$$

이후 facenet의 임계값인 100과 facenet을 통해 분산된 값의 최대값인 300을 이용하여 정규화한다.

$$dist_{normalize} = \frac{dist_{norm} - 100}{300}$$

첫 번째 정규화를 통해 dist값은 -1에서 1사이 값으로 정규화된다. -1이 true(동일 얼굴)이고, 1이 false(상이 얼굴)이다.

$$f(x) = -1/2x^3 + 3/2x$$

거리 지표를 더욱 정교하게 만들기 위해, 함수  $f(x)$ 를 연속적으로 3번 적용한다. 이 함수는 음수를 -1로, 양수를 1로 매핑하는 비선형 변환(non-linear transformation)의 특성을 가지고 있다.

마지막으로 아래와 같은 정규화식  $N(x)$ 를 적용해 유사하지 않은 임베딩들 간의 거리를 더욱 벌리고 유사한 임베딩들은 더욱 가깝게 만들어 주었으며 true를 1, false를 0으로 매핑하도록 정규화해 주었다.

$$N(x) = \frac{-f(f(f(\frac{x-100}{300}))) + 1}{2}$$

## 4. 동형암호 기반 FaceNet

일반적으로 FaceNet에 동형암호를 적용시키면 임베딩 벡터 간의 거리를 반드시 사용자 측에서 복호화 해야 하므로 서버 측에서 얼굴의 유효성 여부를 알고자 할 때, 그 정보를 신뢰하기 어렵다.

### 4-1. 얼굴 등록

먼저 공개키와 비밀키로 구성된 암호키를 생성한다. 그리고 클라이언트의 얼굴로부터 특징 벡터를 추출한 후 비밀키를 가지고 암호화한 다음 서버 측으로 송신하면 등록 절차가 마무리된다. 얼굴 등록 절차는 후술할 마스킹 기법을 적용한 모델에서도 동일하게 이뤄진다.

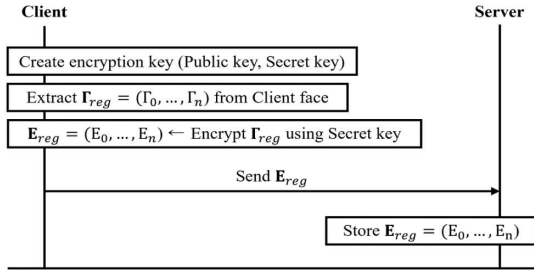


그림2. 얼굴 등록 단계

#### 4-2. 얼굴 검증

검증하고자 하는 얼굴로부터 특징 벡터를 추출한 후 비밀키를 가지고 암호화한 다음 서버 측으로 송신하면 서버는 이를 저장한 뒤 등록 단계에서 저장한 특징 벡터와 L2 norm 방식으로 벡터 간 거리를 계산한다. 이후 계산된 거리 값을 정규화한 뒤 클라이언트 측으로 송신한다. 클라이언트는 수신한 값을 비밀키를 가지고 복호화 한 뒤 해당 얼굴이 유효한지 판단한다.

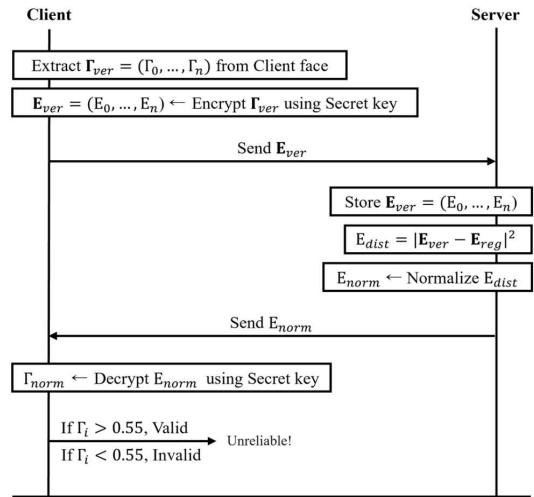


그림3. 얼굴 검증 단계

#### 5. 마스킹 기법

앞선 정규화 과정을 거친  $dist$ 는 0과 1사이에 놓이게 되고, 이 값이 1에 가까울수록 동일한 얼굴일 가능성이 높고, 0에 가까울수록 서로 다른 얼굴일 가능성이 높다. 본 연구에서는 이와 같이 동일한 얼굴일 때와 그렇지 않을 때,  $dist$ 의 분포가 확연히 다르게 나타난다는 점에 착안하여 클라이언트의 응답의 신뢰성을 보장하고자 하였다.

서버에서  $dist$ 를 암호화된 상태로 연산한 후에 클라이언트에 보내게 될 마스킹된 벡터 내 무작위로 선택된 한 인덱스에만 해당 값  $dist_{real}$ 을 넣고, 해당 인덱스 위치를 서버 내에 저장한다. 이후 벡터의 나머지 인덱스에는 서로 다른 얼굴일 경우 나타나는 0에 가까운 값  $dist_{fake}$ 들로 적절히 뽑아 넣어준 뒤 채워진 마스킹 벡터를 클라이언트로 보낸다.

0	1	...	$i$	...	$n$
$dist_{fake0}$	$dist_{fake1}$	...	$dist_{real}$	...	$dist_{fake(n)}$

그림4. 마스킹 벡터의 구조

실제 계산한  $dist_{real}$ 이 서로 다른 얼굴에서 나타난 값일 경우, 클라이언트는 마스킹 벡터를 복호화했을 때 특정한 기준을 충족하는 단 하나의 값인 실제  $dist_{real}$ 과 기준을 충족하지 못하는 나머지 다른 값들을 확인할 수 있다. 따라서 클라이언트는 이를 서버가 계산한 실제  $dist_{real}$ 로 판단하고, 해당 인덱스 위치를 서버에 전송한다. 서버에서는 클라이언트가 보내온 인덱스 위치가 서버에 저장된 인덱스 위치와 일치할 경우, 최종적으로 동일한 얼굴이라는 결과를 클라이언트에 알린다.

실제 계산한  $dist_{real}$ 이 서로 다른 얼굴로부터 나온 값일 경우, 클라이언트는 마스킹 벡터를 복호화했을 때, 기준을 충족하는 값을 찾을 수 없다. 실제  $dist_{real}$ 와 나머지  $dist_{fake}$ 들을 구분할 수 없다는 의미이다. 따라서 클라이언트는 해당하는 인덱스의 위치를 찾을 수 없었다고 서버에 알리고, 서버에서는 최종적으로 유효하지 못하다는 결과를 클라이언트에 알린다.

본 연구에서는 클라이언트가 실제  $dist_{real}$ 를 복호화 해 서로 다른 얼굴이라는 판단을 내렸음에도 서버에 동일한 얼굴이었다는 거짓 정보를 내놓아 혼란을 야기하는 것을 방지하고자 하였다. 제안된 마스킹 기법을 활용하면 클라이언트는 서버가 실제  $dist_{real}$ 을 저장한 인덱스의 위치를 찾아내야 동일한 얼굴이며 조작된 결과가 아니라는 증명을 해낼 수 있다. 물론 실제  $dist_{real}$ 이 동일한 얼굴로부터 나온 값일 때, 인덱스의 위치를 찾았음에도 서버에 찾지 못했다고 거짓 정보를 보낼 수는 있다. 하지만 생체 인증이라는 시스템의 특징상 동일한 얼굴임에도 이를 부인함으로써 클라이언트가 얻을 수 있는 이익은 거의 없다고 판단해 해당 상황의 발생은 추가 고려하지 않았다.

#### 5-1. 통계적 구분 불가능성

제안된 마스킹 기법이 클라이언트 응답의 신뢰성을 보장하기 위해서 가장 중요한 요소는 크게 두 가지이다. 첫번째는 대상이 서로 다른 얼굴인 경우에 실제 계산된  $dist_{real}$ 과 특정한 통계적 분포로부터 얻어낸  $dist_{fake}$ 들이 구분 불가능한지이다. 두번째는 대상이 동일한 얼굴인 경우에 실제 계산된  $dist_{real}$ 과  $dist_{fake}$ 들을 구분할 수 있는지이다. 본 연구에서는 통계적 구분 불가능성을 이용해 첫 번째 요소를 만족시키고자 하였다 [4].

#### 5-2. 비유효값 분포 분석

$dist_{fake}$ 를 얻을 수 있는 통계적 분포를 생성하기 위해, 우선 한 사람의 얼굴을 기준으로 서로 다른 사람 939명의 얼굴로부터  $dist_{real}$ 을 구했고, 앞서 언급한 것과 같이 1보다는 0에 가까운 값들이 주로 나타나는 것을 확인할 수 있다.

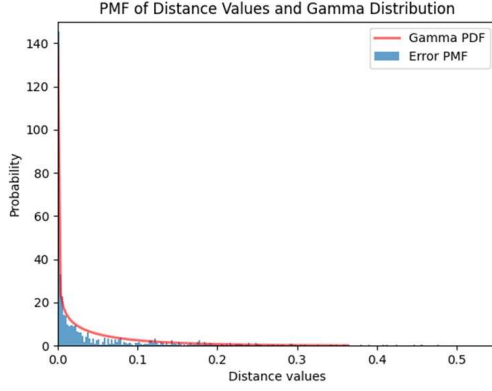


그림5. 비유효값의 분포 및 감마 분포

그림5의 서로 다른 얼굴의  $dist_{real}$  분포와 가장 유사한 형태를 띄는 분포를 찾아 해당 분포로부터 임의의  $dist_{fake}$ 를 추출하고자 하였다. 또한 본 연구에서는 거시적으로 파라미터를 조절하면서 적절한 감마 분포를 찾아 적용하고자 하였다. 다음은 매개 변수  $\alpha$ 와  $\beta$ 로 나타낼 수 있는 감마분포의 PDF(Probability Density Function)이다.

$$f(x) = \frac{x^{\alpha-1} e^{-\beta x} \beta^{\alpha}}{\Gamma(\alpha)}$$

본 연구에서 적용하고자 한 그림5의 감마분포의 파라미터는  $\alpha = 0.61, \beta = 10.0$ 이다. 최종적으로 해당 분포로부터 임의의  $dist_{fake}$ 를 추출하였다.

### 5-3. 마스킹 기법의 적용

마스킹 기법은 검증단계에서 사용되므로 얼굴 등록 단계는 이전과 동일하다.

#### 5-3-1. 얼굴 등록

먼저 공개키와 비밀키로 구성된 암호키를 생성한다. 그리고 클라이언트의 얼굴로부터 특징 벡터를 추출한 후 비밀키를 가지고 암호화한 다음 서버 측으로 송신하면 등록 절차가 마무리된다. 이는 앞서 설명했듯이 마스킹 기법을 적용하기 이전과 동일하다.

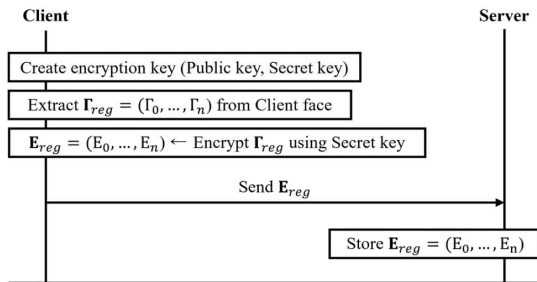


그림6. 얼굴 등록 단계

#### 5-3-2. 얼굴 검증

검증하고자 하는 얼굴로부터 특징 벡터를 추출한 후 비밀키를 가지고 암호화한 다음 서버 측으로 송신하면 서버는 이를 저장한 뒤 등록 단계에서 저

장한 특징 벡터와 L2 norm 방식으로 벡터 간 거리를 계산한다. 이후 계산된 거리 값을 정규화한 뒤 임의의 인덱스를 랜덤하게 선택한 후 해당 인덱스에는 계산된 거리 값을 넣어주고, 해당 인덱스를 제외한 나머지 인덱스에는 앞에서 설정한 임의의 감마분포로부터 비유효값들을 추출해 넣어준다.

위와 같은 과정에 의해 마스킹 벡터를 완성하고, 해당 마스킹 벡터를 클라이언트 측으로 송신한 후 클라이언트는 비밀키를 가지고 이를 복호화한다.

이때 얼굴이 유효하다면 복호화 한 마스킹 벡터에서 0.55 이상인 1에 가까운 값이 서버에서 선택한 인덱스에서 나타날 것이고, 해당 인덱스의 위치를 서버에 반환해 유효성 및 무결성을 증명해낼 수 있다. 만약 유효한 얼굴이 아니라면 모든 인덱스의 값이 0.55 미만인 0에 가까운 값들로 나타날 것이므로 서버에서 지정한 인덱스를 찾아낼 수 없어 유효성 및 무결성을 증명해낼 수 없다.

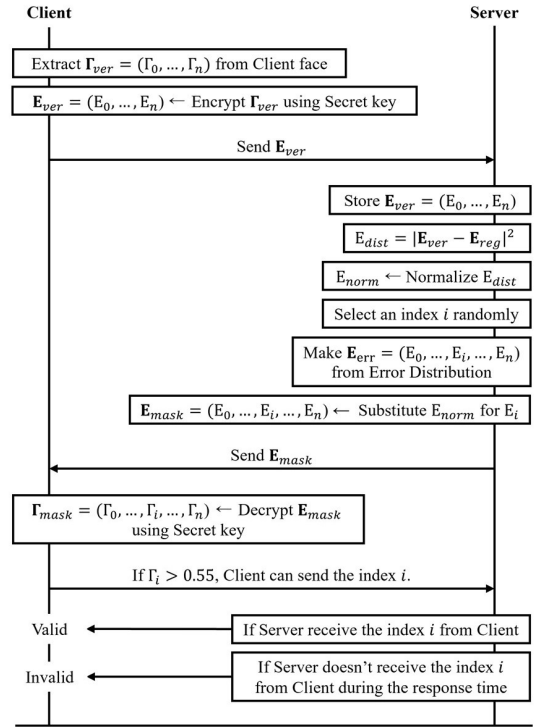


그림7. 얼굴 검증 단계

## III. 최종 구현

### 1. 모바일 출입 통제 시스템 구현

마스킹 기법을 적용한 얼굴 검증 시스템을 실제 어플리케이션으로 구현해 해당 시스템이 원활하게 작동하는지 확인하고자 했다. 구현과정에서 MTCNN 방식으로 얼굴을 검출해내어 Facenet 모델로 얼굴 임베딩 벡터를 한번에 추출해주는 프레임워크인 deepFace를 사용하였다 [5].





그림8. 로그인 화면

그림9. 메인 화면

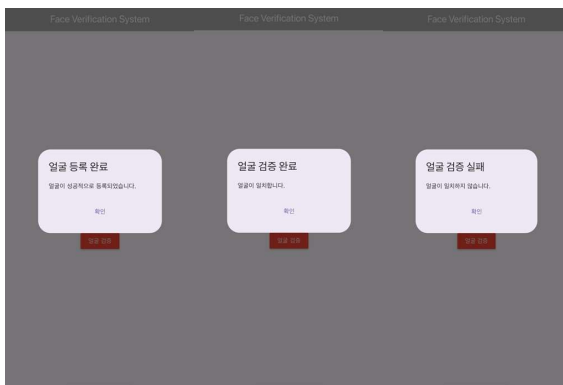


그림10. 얼굴 등록 및 검증 화면

그림9와 같이 암호키 생성(공개키, 비밀키), 얼굴 등록, 얼굴 검증으로 총 세가지 기능을 구현하였으며 실제 얼굴을 가지고 시연한 결과는 그림10과 같다.

그림8의 화면에서 로그인을 하고 암호키를 생성하면 해당 사용자의 공개키를 서버로 보내고, 서버는 해당 사용자의 공개키로만 사용자 얼굴의 유효

성을 연산할 수 있다. 이때 만약 다른 사용자의 공개키로 연산하면 정상적인 범위의 값이 나오지 않는다.

구현 과정에서 예상된 한계점 중 하나인 소요시간 측면의 문제점을 다시 한번 상기할 수 있었다. 암호키 생성 단계에서 공개키의 파일 크기가 매우 커 공개키를 서버로 보내는 과정에서의 소요시간이 상당히 길었으며, 검증 단계에서도 동형암호 연산으로 인해 발생하는 느린 연산의 특징으로 소요시간에 큰 영향을 주었다. 하지만 이는 앞으로 동형암호 분야의 연구가 활발히 이뤄지면서 자연스럽게 해결될 문제일 것이다.

## 2. 한계점

임의의 비유효값 분포의 통계적 구분 불가능성을 보완하기 위해 Gaussian Noise를 더해주고자 했으나, Threshold인 0.55를 넘는 값이 다수 발생하는 등 어플리케이션 구현 과정에서 예러가 발생해 이를 제외해주는 등 통계적 구분 불가능성을 만족시키는데 다소 완벽하지 못했다. 하지만 방향성을 제시하고 그 예를 보여주는 것이 목표였기에 이를 감안하고 실험을 진행하였다. 또한 CNN 모델로부터 임베딩 벡터를 추출하는 과정을 서버가 아닌 클라이언트 측에서 수행하도록하면 클라이언트가 처리할 연산량이 비교적 많아지지만, 서버 측에서 수행하도록 할 경우 CNN 모델 자체를 동형암호 기반으로 재설계된 모델의 재학습 필요 등의 문제점으로 현재로서는 마스킹 기법으로 해결하는 방법이 가장 현실적이었다.

## IV. 결론

결과적으로 사용자는 개인 단말기를 사용하고 동형암호로 개인 정보를 보호받을 수있어 시스템과 서버를 신뢰할 수 있고, 서버는 마스킹 기법을 사용하여 사용자를 신뢰할 수 있어 양방향으로 신뢰도가 보장된다. 또한 서버 측에서 따로 인식 단말을 마련하지 않아도 QR코드, GPS 등의 추가적인 방법들로 서버가 요청할 경우에만 얼굴 검증을 시도할 수 있도록 해, 사용자의 모바일 단말기만으로 얼굴 검증을 하되 서버가 통제권을 갖도록 할 수 있다.

## 참고문헌

- [1] J. H. Cheon et al., "Homomorphic encryption for arithmetic of approximate numbers," in Advances in Cryptology-ASIACRYPT 2017: 23rd International Conference on the Theory and Applications of Cryptology and Information Security, Hong Kong, China, Dec. 3-7, 2017, vol. 23, Springer International Publishing, 2017.
- [2] A. Benaissa et al., "Tenseal: A library for encrypted tensor operations using homomorphic encryption," 2021.
- [3] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015.

- [4] P. N. Vasudevan, "Topics in Information Security," National University of Singapore, Sep. 14, 2021.
- [5] S. Serengil and A. Özpınar, "A Benchmark of Facial Recognition Pipelines and Co-Usability Performances of Modules," Bilişim Teknolojileri Dergisi, vol. 17, no. 2, pp. 95-107, 2024.



**서 지 형**

2019년~ 현재 인하대학교 정보통신공학과 학사과정 재학중.  
관심분야는 인공지능, 프론트엔드 개발, 보안



**홍 준 성**

2018년~ 현재 인하대학교 정보통신공학과 학사과정 재학중.  
관심분야는 인공지능, 데이터베이스



**박 준 형**

2019년~ 현재 인하대학교 정보통신공학과 학사과정 재학중.  
관심분야는 인공지능, 프론트엔드 개발, 데이터 시각화