Implementation and Analysis of Homomorphic Facial Image Encryption and Manipulation

Aldrich Ellis C. Asuncion Brian Christopher T. Guadalupe

4th International Conference on Multimedia Systems and Signal Processing

Ateneo de Manila University

Homomorphic Encryption

- Allows for operations to be "performed" on encrypted data.
  - Allows operations to be performed on recovered plaintexts by performing corresponding operations on ciphertexts.
- **Fully homomorphic encryption (FHE)**
  - FHE allows for arbitrary computation on encrypted data through addition and multiplication of bits, but is very slow.
- **Partially homomorphic encryption (PHE)**
  - PHE only allows *some* operations to be performed securely.
  - PHE schemes are significantly less time-intensive than current FHE schemes.

Homomorphic Encryption in Image Processing

- CryptoImg (2016)
  - A **software library** which extends OpenCV, by Ziad, et al., which uses **homomorphic encryption** to perform image processing operations on remote servers without compromising user privacy
  - Uses a modified version of the **Paillier cryptosystem** to perform homomorphic operations on **floating-point numbers**
  - Implements **linear image processing operations**, such as image adjustment, spatial filtering, edge sharpening and histogram equalization

Motivation

- Can we perform more image operations, particularly **non-linear image operations**?
- Expand on current research by comparing different homomorphic cryptosystems and providing a foundation for more complex image processing tasks (e.g. facial detection and recognition)

Methodology

We considered the following homomorphic cryptosystems.

- ▷ Paillier cryptosystem
- ▷ Damgård–Geisler–Krøigaard (DGK) cryptosystem
- ▷ Brakerski–Gentry–Vaikuntanathan (BGV) cryptosystem

We consider the following image intensity transformations.

- ▷ Image negation: $T(r) = L - r$, L is the maximum intensity value
- ▷ Logarithm transformation: $T(r) = c \log(1 + r)$, $c = 30$
- ▷ Power-law transformation: $T(r) = cr^{\gamma}$, $c = 1$, $\gamma = 0.4$

We implement the three intensity transformations in each of the cryptosystems and test for **accuracy** and **time-efficiency**.

Methodology

- A wrapper library was implemented in Python 3.7.1 which combines the three cryptosystems in a unified interface.
- Existing implementations were ported for use in this study.
  - The Paillier cryptosystem is incorporated using the python-paillier library
  - The DGK cryptosystem is ported from an existing C++ implementation developed by Daniel Demmler.
  - The BGV cryptosystem is implemented using the Pyfhel library using a HElib backend.

On partially homomorphic cryptosystems

- The Paillier and DGK cryptosystems only support a limited set of homomorphic operations.
  1. Given two encrypted integers, can obtain encryption of their sum.
  2. Given an encrypted integer and an unencrypted integer, can obtain encryption of their sum.
  3. Given an encrypted integer and an unencrypted integer, can obtain encryption of their product.

Extensions to the Paillier and DGK cryptosystems

- There is a known method to extend Paillier to **floating-point numbers**, allowing secure addition/subtraction and secure plaintext multiplication. This can also be applied to DGK.
- In 2017, Boukoros, Karvelas, and Katzenbeisser presented a protocol to perform secure **division** in a two-party system.
- In 2009, Erkin, et al. presented a protocol to perform secure **squaring** in a two-party system.
- We can use the squaring protocol to perform secure **multiplication** in a two-party system.
  - Suppose we have the encryptions of a and b.
  - We can compute the encryptions of $(a+b)^2$, a^2, and b^2.
  - Then, compute the encryption of $\frac{1}{2}((a+b)^2 - a^2 - b^2) = ab$.

Implementation of image intensity transformations

- The logarithm of an encrypted number can then be computed using the following closed-form approximation:

$$\log(1+x) \approx \frac{a(x)}{b(x)} + \log 16,$$

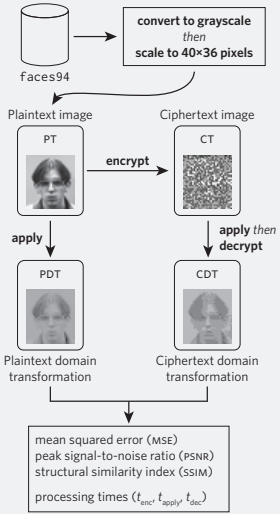
where

$$a(x) = 137x^5 + 26685x^4 + 617370x^3 - 6498630x^2 - 121239315x - 257804775$$
$$b(x) = 30(x^5 + 405x^4 + 27210x^3 + 488810x^2 + 2536005x + 3122577).$$

- The power-law transformation was implemented based on the first five terms of the infinite series:

$$x^\gamma = \sum_{n=0}^{\infty} \frac{(\gamma \log x)^n}{n!},$$

Methodology



Tests for image quality and efficiency

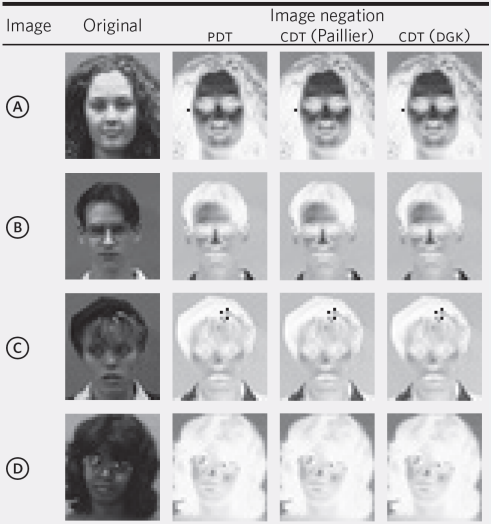
Mean squared error (MSE). Lower mean squared error between the PDT and the CDT indicate higher preservation of image quality.

Peak signal noise ratio (PSNR). An estimator for human visual perception of reconstruction quality, higher values of PSNR indicate higher image quality preservation.

Structural similarity index (SSIM). Gauges structural similarity between neighboring pixels in the PDT and CDT, Higher values of SSIM indicate higher structural similarity, and an SSIM of 1 indicates that the two images are identical.

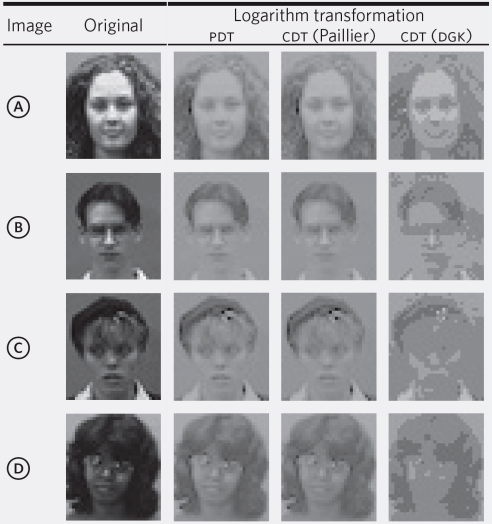
Time taken for processing. We record the time for encryption (t_{enc}), applying the intensity transformation (t_{apply}), and decryption (t_{dec}).

Image negation



For the image negation transformation, both the Paillier and DGK cryptosystems yielded accurate results
($\mathrm{MSE} = 0, \mathrm{PSNR} = \infty, \mathrm{SSIM} = 1$).
The Paillier cryptosystem took less time than the DGK cryptosystem to encrypt the images, apply the transformation, and decrypt the images (see paper).

Logarithm transformation



There was an increase in processing time when applying the logarithm transformation compared to the image negation transformation. The Paillier cryptosystem yielded accurate results (with $MSE \leq 30$ and $SSIM \approx 0.99$) , while the DGK cryptosystem yielded inaccuracies (with $MSE \leq 300$ and $0.61 \leq SSIM \leq 0.87$).

Power-law transformation

Image	Original	Power-law transformation		
		PDT	CDT (Paillier)	CDT (DGK)
Ⓐ				
Ⓑ				
Ⓒ				
Ⓓ				

There were inaccuracies in the output of both cryptosystems. Noise generated by computations have rendered the output images unusable.

Remarks

During testing, initial results obtained from BGV, particularly with image operations involving floating-point numbers like the logarithmic transformation, were unstable (i.e. the results vary drastically every time the same tests were done).

Conclusions based on preliminary results

- It is possible to extend the capabilities of partially homomorphic cryptosystems when assuming a two-party system.
- Closed-form approximations of functions can be applicable for implementation under a partially homomorphic cryptosystem.
- Between the Paillier and DGK cryptosystems, the Paillier cryptosystem is more applicable for non-linear intensity transformations.
- It is possible to perform reasonably accurate logarithm intensity transformations under a homomorphic cryptosystem under a two-party system.

Recommendations and Future Work

- Consider other approximations to logarithm / power-law transformations or other homomorphic cryptosystems, which may improve accuracy
- Possible extension to facial detection under homomorphic encryption
- Implementations of actual client-server systems can be explored

Thank you!