# FULLY HOMOMORPHIC FACES

*Juan Ramón Troncoso-Pastoriza*[1], *Fernando Pérez-González*[1,2,3]

1. **Signal Theory and Communications Department**, University of Vigo, 36310, Spain
2. **Gradiant**, 36310, Spain
3. **Dept. of Electrical and Computer Engineering**, University of New Mexico, Albuquerque, NM, USA
{troncoso,fperez}@gts.uvigo.es

**ABSTRACT**

Face recognition is a prominent application of image processing. It is also a very sensitive application, and privacy concerns have been lately raised and tackled in several recent papers dealing with privacy-preserving face recognition systems. Nevertheless, the presented systems either use the knowledge of some information derived from the database templates in order to perform the recognition or require several interaction rounds between client and server. In this paper, we propose a private system that can cope with a simple verification algorithm executed in the server without interaction (using a quasi-fully homomorphic encryption and an efficient face features representation with Lloyd-Max quantized Gabor jets), in which both the templates and the queried face are encrypted; we show its performance in terms of time complexity and size of transferred encryptions, as well as in verification accuracy with respect to the non-private system. This opens the door to completely private and noninteractive outsourcing of face recognition.

***Index Terms***— Lattice cryptography, homomorphic processing, face verification, privacy

## 1. INTRODUCTION

Face recognition is a prominent image processing application with privacy constraints, due to the sensitiveness of the involved biometric signals. In a common privacy-aware face recognition scenario, a user presents his/her face for matching against a database of enrolled clients; the latter must not be disclosed to the new user, as this would harm the security of the system and the privacy of the enrolled users, while the face presented by the query user must not be disclosed to the recognition system, for preserving the user's privacy. Recent privacy-preserving efficient solutions for this scenario combine homomorphic encryption and garbled circuits [1, 2], focusing on private face identification using the Eigenfaces algorithm, based on applying a PCA projection matrix to the presented face.

However, this *traditional* scenario does not protect the privacy of the enrolled users; it discloses the stored templates and the projection matrix to the recognition system. Currently, outsourced scenarios, where Clouds or other *untrusted* environments are used not only for storing the databases but for performing certain operations, are becoming increasingly ubiquitous. If the matching database is stored in an untrusted third party together with the detection logic, enrolled users' privacy must also be protected, and that party must have access neither to the database contents nor to the fresh faces presented against the system for recognition. Additionally, it is desirable that the system can run autonomously without interaction rounds with the client, requiring the lowest computational effort from the client-side, that usually runs on an embedded or mobile device.

In this work we tackle this privacy-aware scenario, where we aim at face verification in an outsourced system that works with a fully encrypted template database and query faces (total privacy) and provides a verification result without interaction with the client. For that purpose, we provide a quasi-fully homomorphic extension of Gentry's fully homomorphic cryptosystem [3], and show its performance in the envisaged biometric scenario, opening up a wide new set of applications, and providing a first stone for the fully private noninteractive outsourced processing in untrusted environments.

As for the used notation, matrices and (row) vectors are respectively represented as uppercase and lowercase boldface letters, while random variables are represented as uppercase letters; $[a]_d$ represents the reduction of $a \mod d$; vector notation $\boldsymbol{a} = [a_0, \ldots, a_{n-1}]$ and polynomial notation $a(x) = \sum_{i=0}^{n-1} a_i \cdot x^i$ will be used indistinctly when appropriate.

The rest of the paper is organized as follows: Section 2 reviews Gentry's cryptosystem; section 3 presents the proposed extension with a lower bound on the number of achievable sequential homomorphic multiplications; section 4 presents the application to a fully-private noninteractive face verification scenario, and evaluates its performance in widely known test databases. Finally, section 5 draws some conclusions.

## 2. GENTRY'S FULLY HOMOMORPHIC CRYPTOSYSTEM

We take one of the latest versions of Gentry's bootstrappable fully homomorphic cryptosystem, presented in [3]. The cryptosystem is GGH-type based on ideal lattices. The rationale behind GGH cryptosystems lies in choosing two bases for a given lattice $L$, $\boldsymbol{B}_{sk}$ and $\boldsymbol{B}_{pk}$, respectively the secret key (a good basis with quasi-orthogonal vectors) and the public key (a bad basis, normally chosen as the Hermite Normal Form, HNF, of the lattice) of the cryptosystem. The encryption $\boldsymbol{c}$ of a message $m$ is built adding an error vector $\boldsymbol{e}$ s.t. $||\boldsymbol{e}||_1 < \lambda_1(L)$, that encodes $m$, to a point in the lattice. For decrypting, $\boldsymbol{e}$ is recovered using the basis $\boldsymbol{B}_{sk}$ as $\boldsymbol{e}' = \boldsymbol{c} \mod \boldsymbol{B}_{sk}$.

The *somewhat homomorphic* scheme presented by Gentry in [3], following the same approach as Smart and Vercauteren [4], uses a principal-ideal lattice $J$, generated by a chosen polynomial $v(x)$ with $t$-bit signed random integer coefficients ($\boldsymbol{v}$ in its vector notation), in the ring of polynomials modulo $f_n(x) \triangleq x^n + 1$, with a specific structure for its HNF $\boldsymbol{B} = HNF(J) = \begin{pmatrix} d & \boldsymbol{0} \\ \boldsymbol{r} & \boldsymbol{I}_{n-1} \end{pmatrix}$, where $d$ can be defined as $d = \det(J)$ or, equivalently, as the resultant of the polynomials $v(x)$ and $f_n(x)$, and $r$ is a root of $f_n(x) \mod d$, that forms the vector $\boldsymbol{r} = [-r, -[r^2]_d, \ldots, -[r^{n-1}]_d]^t$. $\boldsymbol{B}$ is the *public-key* encryption matrix, completely determined by the integers $d, r$, while the private key is given by $v(x)$ and its scaled ($\mod f_n(x)$)-inverse $w(x)$ ($v(x) \times w(x) = d \mod f_n(s)$).

As defined, this cryptosystem is *quasi*-homomorphic under addition and multiplication, that are directly mapped from the cryptotext ring (errors w.r.t. lattice points) to the clear-text ring. There is, however, a restriction to this homomorphism, as both operations are only correctly mapped when the error lies within the same Voronoi region of the lattice $L$ after applying the operation.

## 3. EXTENDING GENTRY'S CRYPTOSYSTEM

Gentry's cryptosystem can only cope with binary numbers, allowing for homomorphic *and* and *xor* gates; hence, simple arithmetic circuits with $b$-bit numbers need a high amount of binary homomorphic operations that increase the noise within the Voronoi region of the lattice, whose volume bounds the number of operations that do not lead to a decoding error. Gentry and Halevi [3] empirically calculate the maximum depth of an executable polynomial, for bootstrapping the squashed decryption circuit and achieving a full homomorphism.

In this section we extend the plaintext-size, allowing for homomorphic additions and multiplications in $\mathbb{Z}_{2^k}$ (powers of two are chosen for convenience); we also give a theoretical bound on the maximum number of executable multiplications, that also supports Gentry's empirical study for $\mathbb{Z}_2$. The extension seeks to enhance the efficiency of arithmetic non-interactive operations and decrease the cipher expansion rate and to trade the full homomorphic property by the possibility of dealing with a limited but high number of sequential arithmetic processing without interaction.

### 3.1. Encryption

In Gentry's original cryptosystem, the encryption operation of a bit $b \in \mathbb{Z}_2$ uses a random noise vector $\boldsymbol{u} \in \{0, \pm 1\}^n$, with each entry chosen as 0 with probability $q$ and $\pm 1$ with probability $(1-q)/2$ each; we extend the encryption for coping with $m \in \mathbb{Z}_{2^k}$:

$$\boldsymbol{a} = 2^k \boldsymbol{u} + m \cdot \boldsymbol{e}_1; \quad \boldsymbol{c} = \boldsymbol{a} \mod \boldsymbol{B} = [a(r)]_d \cdot \boldsymbol{e}_1.$$

The vector $\boldsymbol{c}$, as in the original construction, has only one non-zero component, representative of the encryption: $c = [a(r)]_d = [m + 2^k \sum_{i=0}^{n-1} u_i r^i]_d$. The complexity of encrypting a $k$-bit number is the same as for encrypting a bit in the original system. Furthermore, the security in terms of Birthday-type attacks is not altered either, as the noise vector has the same bits of entropy; hence, given a security level $\lambda$, $q$ may still be chosen as $2^{(1-q)n} \cdot \binom{n}{qn} > 2^{2\lambda}$.

### 3.2. Decryption

For the decryption, the original scheme uses an optimized procedure that only needs one of the odd coefficients of $\boldsymbol{w} \mod d$, denoted $w_i$. Hence, the decryption for a $k$-bit message $m$ becomes $m = [c \cdot w_i]_d w_i^{-1} \mod 2^k$. The only difference w.r.t. the original decryption is the product by $w_i^{-1} \mod 2^k$; being $w_i$ odd, it always exists: the choice of powers of two for the extended plaintext allows for keeping the same key generation process, while the added decryption complexity is negligible compared to $\mod d$ operations.

### 3.3. Homomorphically Achievable Polynomial Degree

Incorrect decryption may only happen when the error vector added to a lattice point lies outside the Voronoi region of the used lattice. This condition boils down to $||\boldsymbol{a} \cdot \boldsymbol{W}||_\infty < d/2$, where $\boldsymbol{W}$ is the rotation basis that generates $(w(x))$, having in each row the coefficients of $w(x) \cdot x^i \mod f_n(x)$. We can bound

$$||\boldsymbol{aW}||_\infty \leq ||\boldsymbol{a}||_\infty ||\boldsymbol{W}||_\infty = \max_i(|a_i|) \cdot \sum_{i=0}^{n-1} |w_i| \leq \sum_{i=0}^{n-1} |w_i| \sum_{i=0}^{n-1} |a_i|,$$

$$\sum_{i=0}^{n-1} |w_i| \sum_{i=0}^{n-1} |a_i| < d/2 \Rightarrow ||\boldsymbol{aW}||_\infty < d/2.$$

The number of non-zero elements ($Nz_j$) of a chosen $\boldsymbol{u}_j$ follows a Binomial distribution $Nz_j \sim Bi(n, 1-q)$. In a fresh encryption, each of these elements has modulus $2^k$, while the message has a modulus $|m| < 2^k$. Hence, $\sum_{i=0}^{n-1}(|a_i|) < 2^k(1 + Nz_j)$.

On the other hand, after a multiplication between two ciphertexts $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$ (in the polynomial quotient ring $\mathbb{Z}_d[x]/(f_n(x))$), the resulting point must also be within the Voronoi region. The product of two polynomials modulo $f_n(x)$ is equivalent to a cyclic convolution of their coefficient vectors (with a sign change for the overlapped subvector). Furthermore, as fresh encryptions have the same absolute value ($2^k$) for all the non-zero coefficients of $\boldsymbol{u}$, the $L_1$ norm of the resulting coefficient vector of the product of a given ciphertext $\boldsymbol{c}_1$ and a fresh encryption $\boldsymbol{c}_2$ is upper-bounded by $||\boldsymbol{c}_1||_1 \cdot 2^k(1 + Nz_2)$. In general, we have that, after $n_m$ successive products of a cipher by fresh encryptions, we can bound the probability of decryption error

$$P[\text{dec error}] = P[||\boldsymbol{aW}||_\infty \geq d/2] \leq$$

$$P\left[\underbrace{\sum_{i=0}^{n_m} \log(1 + Nz_i)}_{N_{n_m}} \geq \log\left(\frac{d}{2^{k(n_m+1)+1} \sum_{l=0}^{n-1} |w_l|}\right)\right],$$

where $N_{n_m}$ can be accurately approximated, using the CLT, by a Gaussian variable with mean $(n_m + 1) \cdot \mu = (n_m + 1) \cdot \cdot \sum_{i=0}^{n} \log_2(1+i) \binom{n}{i} (1-q)^i q^{n-i}$ and variance $(n_m + 1) \cdot \sigma^2 = (n_m + 1) \cdot \sum_{i=0}^{n} (\log_2(1+i) - \mu)^2 \binom{n}{i} (1-q)^i q^{n-i}$.

We may bound the maximum number of bits to which we can extend the ciphertext for allowing a given number $n_m$ of successive multiplications with a given probability of error $p_e$:

$$k_{max} = \left\lfloor \frac{\log_2(d/||\boldsymbol{w}||_1) - 1}{n_m + 1} - \mu - \frac{Q^{-1}(p_e)\sigma}{\sqrt{n_m + 1}} \right\rfloor. \quad (1)$$

As expected, the maximum number of bits decreases with $1/(n_m + 1)$, and it is heavily influenced by the quotient $d/||\boldsymbol{w}||_1$, that intuitively indicates the effective radius of the Voronoi region, supporting noise addition. On the other hand, the choice of $t$ (bit-size of each $v_i$) determines the minimum value of this quotient: as the polynomial product of $v(x) \times w(x) = d \mod f_n(x)$, in vector notation this means that, using the Hölder inequality:

$$d = \boldsymbol{v} \cdot [w_0, -w_{n-1}, \ldots, -w_1]^t \leq ||\boldsymbol{v}||_\infty ||\boldsymbol{w}||_1 < 2^t ||\boldsymbol{w}||_1 \Rightarrow \frac{d}{||\boldsymbol{w}||_1} < 2^t.$$

Hence, for a good lattice, the maximum decodable noise norm (decryption radius) will be close to $t$ bits, and we can provide an estimation of the maximum plaintext bit-size for correct decryption after a given number of multiplications for a generic good lattice, just substituting $\log_2(d/||\boldsymbol{w}||_1)$ by $t$ in Eq. (1). Reciprocally, the inverse of this expression yields the maximum number of affordable multiplications without decryption error. It must be noted that $n_s$ consecutive homomorphic additions can increase at most in $\log_2(n_s)$ bits the size of the $\infty$-norm of the noise vector (Eq. (1) can take

**Table 1**: Lower bound on the maximum number of products and Gentry's empirically obtained maximum degree polynomial as a function of $t$, with $n = 128$

| $t$ | 64 | 128 | 256 | 384 |
|---|---|---|---|---|
| Lower bound | 10 | 22 | 46 | 69 |
| Empirical [3] | 13 | 33 | 76 | 128 |

this into account by subtracting $\log_2(n_s)$ from $t$). Hence, when determining the maximum degree of a polynomial run on fresh ciphered variables, the maximum number of multiplications is the determining factor. Gentry and Halevi provide an approximation of the maximum degree $deg$ of an elementary symmetric polynomial evaluated on $m$ encrypted binary variables, bounding the decryption radius by the approximated Euclidean norm of the polynomial output: $2^t \geq c^{deg}\sqrt{\binom{m}{deg}}$; the results deviate from this expression for large $m$ due to the overestimation of the effect of additions, as the combinatorial number of summed monomials grows above the dimensionality of the lattice, and cannot be considered independent anymore. Table 1 shows the validity of our bound compared to the experimental results obtained by Gentry.

Fig. 1 represents the number of sequentially performed products with new fresh ciphers before a decryption error occurs (for $n = 512$, $t = 380$ and $q = 1 - 20/512$, picking the minimum of 1000 trials), compared to the given lower bound for $p_e = 10^{-4}$. Our worst-case bound is fairly conservative for small plaintexts that allow for a high amount of products, but it becomes tight for medium-to-high $k$, even when the Gaussian approximation in those cases provides an overestimation of the decryption error. We have also obtained very similar results with bigger lattices (as Gentry and Halevi did for the binary case), due to the quotient $\log_2(d/||\boldsymbol{w}||_1)$ being virtually constant for all the found lattices, and the binomial distribution barely changing with high $n$ when fixing the rate $(1-q)\cdot n$.

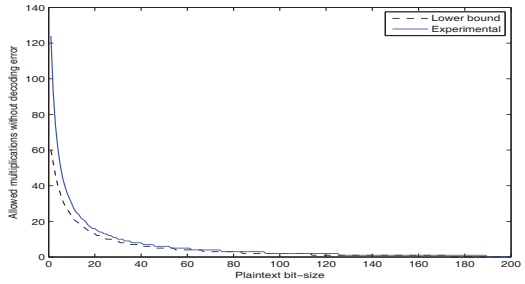


**Fig. 1**: Minimum number of multiplications (Eq. (1)) without decoding error after 1000 trials as a function of $k$

## 4. FULLY PRIVATE NONINTERACTIVE FACE VERIFICATION

In order to test the benefits and versatility of the extended cryptosystem in a typical scenario, we have tested it for outsourced face verification with privacy constraints. In this scenario, a query user presents his face features and a tentative ID against a database; the system must determine if those features actually correspond to the previously enrolled ID. The target of the outsourced privacy-preserving system is to conceal both the presented face features and the database templates to the party that runs the verification process, while the database templates are also not disclosed to the query user.

We have chosen the face representation in [5], that employs quantized Gabor features using a Lloyd-Max quantization based on an accurate model of Gabor coefficients' moduli. Other privacy-preserving systems presented in the literature, like [1], are based on Eigenfaces. In the clear, Gabor filters provide a slightly more complex solution with a better performance (about 8% increase) in known databases like LFW [6], due to the biological models that support the use of Gabor filters. Unlike in [5], we work with indices of quantized coefficients instead of the actual quantized values; this allows for a hugely reduced plaintext size without much degradation in system performance (cf. Section 4.1), and benefits from an inherent normalization of the Jets provided by the quantization process itself. The verification algorithm is based on either average correlation (cosine distance) or average Euclidean distance.

**Table 2**: Average correct recognition rate for LFW's view 2 for Gabor coefficients quantized with $N$ levels and for Eigenfaces

| Gabor unquant. | $N = 8$ | $N = 4$ | $N = 2$ | Eigenfaces |
|---|---|---|---|---|
| 65.95% | 62.90% | 62.60% | 60.65% | 60.02% |

In the enrollment phase, the presented feature vectors are encrypted and stored in a central database for later use as templates. The verification threshold $\eta$ is a system parameter also kept encrypted. In the verification phase, a user presents an ID to be matched together with the encrypted quantized Gabor coefficients $\boldsymbol{g}$ from his face. The database holder homomorphically calculates the encryption of the *soft* score $\sum_{i=0}^{N_{templates}} \text{dist}(template_i, \boldsymbol{g}) - N_{templates}\eta$, that is provided as the output of the verification process. If a binary hard score is needed, then known comparison protocols could be used afterwards, involving interaction with the client, but we are aiming at a fully noninteractive solution, testing the raw performance of the extended cryptosystem in this scenario.

### 4.1. Performance and evaluation results

We have tested the application of the developed encrypted verification system using the Euclidean distance metric in several commonly used databases; due to space constraints, we show only the results on the challenging LFW (Labelled Faces in the Wild). Regarding the recognition accuracy, Fig. 2 and Table 2 respectively show the ROC curves and correct recognition rate for the biometric system using 2,4 or 8-level Lloyd-Max quantization of Gabor coefficients' moduli, compared to the original unquantized clear-text system and an Eigenfaces detector. Correct recognition rate for the quantized Gabor system (8-levels) is around 63%, approximately the same performance as baseline V1-like recognition systems [7]; the degradation with respect to the original unquantized system (around 3%) stems from using just quantization indices instead of the truly quantized values; nevertheless, original performance can be recovered even for 4 quantization levels by applying precalculated weights to the Gabor coefficients [5]. Conversely, Eigenfaces' performance goes down to near 60%, on the order of baseline pixel-space recognition systems.

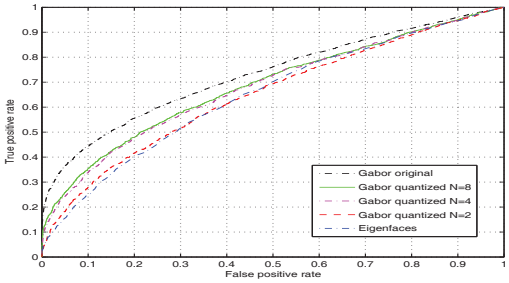


**Fig. 2**: ROC curves averaged over 10 folds of LFW's view 2

**Table 3**: Efficiency figures for the privacy-preserving face-verification algorithm: client times, server homomorphic processing (HP) time and needed bandwidth

| Execution times | Client | | Server | Bandwidth |
|---|---|---|---|---|
| | Cipher | Decrypt | (HP) | |
| Proposed | 0.4s | 0.0026s | 12.3s | 127MB |
| Gentry (binary) | 1.3s | 0.052s | 883.4s | 380MB |
| PaillierCT | 15.4s | 0.0043s | 57.99s | 5.3MB |
| PaillierE | 22s | 43.2s | 479.0s | 13.3MB |

We take the 8-level quantization for its good compromise between clear-text cardinality and recognition performance. Lattice dimensions have been fixed to $n = 512$, with $t = 380$ and $q = 1 - 20/n$, for a security parameter of $\lambda \approx 70$. We work with 5200-dimensional Gabor vectors for each face ($13 \times 10$ localizations, $8$ orientations and $5$ scales) with 3-bit coefficients, so calculating the Euclidean distance between two vectors needs one multiplication per pair of values, 5199 additions and one subtraction. Hence, starting from 8-level coefficients, the resulting score is correctly represented using $\lceil \log_2(5200 \cdot 2 \cdot 8^2) \rceil = 19$ bits), so we use $k = 19$ bits for the extended cryptosystem. Taking into account the $\log_2(5200) = 12.3$ bits of decrease for the effective decryption radius, Eq (1) yields 13 supported consecutive multiplications, so the extended cryptosystem can perfectly cope with the whole distance calculations, without incurring in decryption errors (but with negligible probability).

For implementation we have used the GMP and NTL libraries for C++, and tested the time efficiency without any kind of parallelization in one core of an Intel i5 at 3.30GHz with 8GB of RAM. Table 3 shows the efficiency figures for the proposed algorithm compared to the expected running times of a *traditional* implementation based on an additive homomorphism (Paillier-based [8], using a 2048-bit modulus), with either clear-text templates (PaillierCT, partial privacy) and with encrypted templates (PaillierE, total privacy using interactive multiplication protocols); in both Paillier-based systems the client provides the encryptions of both his face coefficients and their squared value, in the most favorable case for Paillier's homomorphism; we have also included, for reference, the estimated execution time of Gentry's original binary cryptosystem using binary circuits for addition and multiplication; this system cannot provide valid outputs without using homomorphic deciphering circuits, as the degree of the distance circuit exceeds the noise capacity of the used lattice; each of these circuits (for bootstrapping the cipher of a bit), that needs to be applied after each binary multiplication gate, runs in about 8 seconds in our test machine; with about $3.2 \cdot 10^5$ products, the computational load using the binary version of Gentry would become infeasible; we do not include them into the time evaluation, but they are an inherent limitation of the original binary cryptosystem.

Thanks to the extension the system becomes feasible both in terms of bandwidth and server processing time overcoming the pointed out limitation; the use of homomorphic operations in $\mathbb{Z}_{2^k}$ instead of $\mathbb{Z}_2$ reduces the server computation time in almost two orders of magnitude (furthermore, binary encryptions do not provide a correct output without the needed deciphering circuits), while the bandwidth is divided by a factor of three.

In terms of computational efficiency, the extended cryptosystem yields a clear advantage w.r.t. any of the others, even for Paillier with clear-text templates. The load for the client is decreased in two orders of magnitude w.r.t. Paillier, while the server's load decreases in a factor of almost 50. This is due to the lighter homomorphic operations for Gentry's even when they work with larger ciphertexts. Conversely, the transferred encryptions for the proposed system are less than one order of magnitude higher than for encrypted Paillier templates, due to the larger expansion factor that lattice cryptosystems like Gentry's present; this is the main fact that holds back the performance of the homomorphism; the presented extension advances in this path, reducing the expansion factor and greatly increasing the efficiency of the operations performed noninteractively at the server. Furthermore, when the scenario of interest is an outsourced system that processes private data, the initial bandwidth is not critical: the more operations can be performed *unattendedly*, the more versatile and powerful the system becomes.

## 5. CONCLUSIONS

We have presented an extension of Gentry's fully-homomorphic cryptosystem, in which the homomorphic decryption capability is traded for high gains in efficiency when executing low-to-medium degree arithmetic operations. We provide a bound for the number of allowed sequential multiplications, and show the performance of the cryptosystem in a practical scenario dealing with Lloyd-Max quantized moduli of Gabor coefficients for face verification. Contrary to traditional systems based on additive homomorphisms, the presented one allows for a completely private verification, with both encrypted templates and queried faces, opening up the possibility of outsourced noninteractive face recognition within an untrusted environment like a Cloud, being the only needed interaction in that case the initial transmission of the encrypted inputs.

Several future research lines can be highlighted: the specification of the homomorphic decryption circuit for the non-binary case; achieving other ways of decreasing the cipher expansion of the cryptosystem while keeping the good homomorphic properties, either increasing the plaintext size or decreasing the public key size for bigger lattices; finally, providing a noninteractive solution for comparison operations and other nonlinear operations that cannot be directly mapped by the nonbinary homomorphism is also challenging.

## 6. ACKNOWLEDGMENTS

We thank Gonzalo Jiménez for his work in implementing the cryptographic primitives and Daniel González for his helpful comments on Gabor-based biometric systems. This work was partially funded by Xunta de Galicia under projects "Consolidation of Research Units" 2010/85, SAFECLOUD (ref. 09TIC014CT), SCALLOPS (ref. 10PXIB322231PR) and VISAGE (ref. 10TIC008CT), by the Spanish Ministry of Science and Innovation under project COMONSENS (ref. CSD2008-00010) of the CONSOLIDER-INGENIO 2010 Program, and PRISMED (ref. IPT-2011-1076-900000) of the INNPACTO 2011 Subprogram, and by the Iberdrola Foundation through the Prince of Asturias Endowed Chair in Information Science and Related Technologies.

## 7. REFERENCES

[1] Z. Erkin, M. Franz, J. Guajardo, S. Katzenbeisser, I. Lagendijk, and T. Toft, "Privacy-preserving face recognition," in *PETS'09*, 2009, number 5672 in LNCS, pp. 235–253.

[2] A.-R. Sadeghi, T. Schneider, and I. Wehrenberg, "Efficient privacy-preserving face recognition," in *ICISC 2009*. 2010, vol. 5984 of *LNCS*, pp. 229–244, Springer.

[3] C. Gentry and S. Halevi, "Implementing gentry's fully-homomorphic encryption scheme," in *EUROCRYPT 2011*, 2011, vol. 6632 of *LNCS*, pp. 129–148.

[4] N.P. Smart and F. Vercauteren, "Fully Homomorphic Encryption with Relatively Small Key and Ciphertext Sizes," in *PKC 2010*, Paris, France, May 2010, vol. 6056 of *LNCS*, pp. 420–443.

[5] J.R. Troncoso-Pastoriza, D. González-Jiménez, and F. Pérez-González, "A new model for Gabor Coefficients' Magnitude in Face Recognition," in *IEEE ICASSP 2010*, Dallas, USA, March 2010, IEEE.

[6] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller, "Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments, Technical Report 07-49," 2007.

[7] N. Pinto, JJ. Dicarlo, and DD. Cox, "Establishing good benchmarks and baselines for face recognition," in *IEEE ECCV*, 2008, Faces in 'Real-Life' Images Workshop.

[8] P. Paillier, "Public-Key Cryptosystems Based on Composite Degree Residuosity Classes," in *EUROCRYPT'99*. 1999, vol. 1592 of *LNCS*, pp. 223–238, Springer.