Efficient Privacy-Preserving Face Recognition
(Full Version)[*]

Ahmad-Reza Sadeghi, Thomas Schneider, and Immo Wehrenberg

Horst Görtz Institute for IT-Security, Ruhr-University Bochum, Germany
{ahmad.sadeghi,thomas.schneider}@trust.rub.de[**],immo.wehrenberg@rub.de

Abstract. Automatic recognition of human faces is becoming increasingly popular in civilian and law enforcement applications that require reliable recognition of humans. However, the rapid improvement and widespread deployment of this technology raises strong concerns regarding the violation of individuals' privacy. A typical application scenario for privacy-preserving face recognition concerns a client who privately searches for a specific face image in the face image database of a server. In this paper we present a privacy-preserving face recognition scheme that substantially improves over previous work in terms of communication- and computation efficiency: the most recent proposal of Erkin et al. (PETS'09) requires $\mathcal{O}(\log M)$ rounds and computationally expensive operations on homomorphically encrypted data to recognize a face in a database of M faces. Our improved scheme requires only $\mathcal{O}(1)$ rounds and has a substantially smaller online communication complexity (by a factor of 15 for each database entry) and less computation complexity. Our solution is based on known cryptographic building blocks combining homomorphic encryption with garbled circuits. Our implementation results show the practicality of our scheme also for large databases (e.g., for $M = 1000$ we need less than 13 seconds and less than 4 MByte online communication on two 2.4GHz PCs connected via Gigabit Ethernet).

Keywords: Secure Two-Party Computation, Face Recognition, Privacy

1 Introduction

In the last decade biometric identification and authentication have increasingly gained importance for a variety of enterprise, civilian and law enforcement applications. Examples vary from fingerprinting and iris scanning systems, to voice and face recognition systems, etc. Many governments have already rolled out electronic passports [22] and IDs [31] that contain biometric information (e.g., image, fingerprints, and iris scan) of their legitimate holders.

In particular it seems that facial recognition systems have become popular aimed to be installed in surveillance of public places [20], and access and border

[*] This paper will appear at ICISC 2009 [36].

[**] Supported by EU FP6 project SPEED, EU FP7 project CACE and ECRYPT II.

control at airports [8] to name some. For some of these use cases one requires online search with short response times and low amount of online communication.

Moreover, face recognition is ubiquitously used also in online photo albums such as Google Picasa and social networking platforms such as Facebook which have become popular to share photos with family and friends. These platforms support automatic detection and tagging of faces in uploaded images.[1] Additionally, images can be tagged with the place they were taken.[2]

The widespread use of such face recognition systems, however, raises also privacy risks since biometric information can be collected and misused to profile and track individuals against their will. These issues raise the desire to construct privacy-preserving face recognition systems [14]. [3]

In this paper we concentrate on efficient privacy-preserving face recognition systems. The typical scenario here is a client-server application where the client needs to know whether a specific face image is contained in the database of a server with the following requirements: the client trusts the server to correctly perform the matching algorithm for the face recognition but without revealing any useful information to the server about the requested image as well as about the outcome of the matching algorithm. The server requires privacy of its database beyond the outcome of the matching algorithm to the client.

In the most recent proposal for privacy-preserving face recognition [14] the authors use the standard and popular Eigenface [38,37] recognition algorithm and design a protocol that performs operations on encrypted images by means of homomorphic encryption schemes, more concretely, Pailler [33,13] as well as a cryptographic protocol for comparing two Pailler-encrypted values based on the Damgård, Geisler and Krøigård [10,11,12] cryptosystem). They demonstrate that privacy-preserving face recognition is possible in principle and give required choices of parameter sizes to achieve a good classification rate. However, the proposed protocol requires $\mathcal{O}(\log N)$ rounds of online communication as well as computationally expensive operations on homomorphically encrypted data to recognize a face in the database of N faces. Due to these restrictions, the proposed protocol cannot be deployed in practical large-scale applications. In this paper we address this aspect and show that one can do better w.r.t. efficiency.

Basically one can identify two approaches for secure computation: the first approach is to perform the required operations on encrypted data by means of homomorphic encryption (see, e.g., [33,13]). The other approach is based on Garbled Circuit (GC) à la Yao [40,26]: the function to be computed is represented by a garbled circuit i.e., the inputs and the function are encrypted ("garbled"). Then the client obliviously obtains the keys corresponding to his inputs and decrypts the garbled function. Homomorphic Encryption requires low communication complexity but huge round and computation complexity whereas GC

[1] `http://picasa.google.com/features-nametags.html`; `http://face.com`

[2] Geotagging can be done either manually or automatically on iPhones using GPS `http://www.saltpepper.net/geotag`.

[3] Similar concerns motivated previous research directions on privacy-preserving iris scanning [9] or fingerprinting [39].

has low online complexity (rounds, communication and computation) but large offline communication complexity. We present a protocol for privacy-preserving face recognition based on a hybrid protocol which combines the advantages of both approaches. Additionally, we give a protocol which is based on GC only.

Contribution. We give an efficient and secure privacy-preserving face recognition protocol based on the Eigenfaces recognition algorithm [38,37] and a combination of known cryptographic techniques, in particular Homomorphic Encryption and Garbled Circuits. Our protocol substantially improves over previous work [14] as it has only a constant number of $\mathcal{O}(1)$ rounds and allows to shift most of the computation and communication into a pre-computation phase. The remaining online phase is highly efficient and allows for a quick response time which is especially important in applications such as biometric access control.

Related Work. *Privacy-Preserving Face Recognition* allows a client to obliviously detect if the image of a face is contained in a database of faces held by server. We give a detailed summary of previous work on privacy-preserving face recognition [14] in §3.1. Our protocol has a substantially improved efficiency.

The related problem of *Privacy-Preserving Face Detection* [3] allows a client to detect faces on his image using a private classifier held by server without revealing the face or the classifier to the other party.

In order to preserve privacy, faces can be de-identified such that face recognition software cannot reliably recognize de-identified faces, even though many facial details are preserved as described in [32].

2 Preliminaries

In this section we summarize our conventions and setting in §2.1 and cryptographic tools used in our constructions in §2.2 (additively homomorphic encryption (HE), oblivious transfer (OT), and garbled circuits (GC) with free XOR). A summary of the face recognition algorithm using Eigenfaces is given in §2.3. Readers familiar with the prerequisites may safely skip to §3.

2.1 Parameters, Notation and Model

We denote symmetric security parameter by t and the asymmetric security parameter, i.e., bitlength of RSA moduli, by T. Recommended parameters for short-term security (until 2010) are for example $t = 80$ and $T = 1024$, whereas for long-term security $t = 128$ and $T = 3072$ are recommended [18]. The statistical correctness parameter is denoted with κ [4] and the statistical security parameter with σ. In practice, one can choose $\kappa = 40$ and $\sigma = 80$.

[4] The probability that the protocol computes a wrong result (e.g., caused by an overflow) is bounded by $2^{-\kappa}$.

We work in the semi-honest model where participants are assumed to be honest-but-curious (details later in §3). Our improved protocols can be proven in this model based on existing proofs for the basic building blocks from which they are composed. We further note that efficient garbled circuits of [25] (and thus our work) requires the use of random oracles. We could also use correlation-robust hash functions [23], resulting in slightly more expensive computation of garbled circuits [35] (see below).

2.2 Cryptographic Tools

Homomorphic Encryption (HE). We use a semantically secure additively homomorphic public-key encryption scheme. In an additively homomorphic cryptosystem, given encryptions $[\![a]\!]$ and $[\![b]\!]$, an encryption $[\![a+b]\!]$ can be computed as $[\![a+b]\!] = [\![a]\!][\![b]\!]$, where all operations are performed in the corresponding plaintext or ciphertext structure. From this property follows, that multiplication of an encryption $[\![a]\!]$ with a constant c can be computed efficiently as $[\![c \cdot a]\!] = [\![a]\!]^c$ (e.g., with the square-and-multiply method).

As instantiation we use the Paillier cryptosystem [33,13] which has plaintext space $\mathbb{Z}_N$ and ciphertext space $\mathbb{Z}^*_{N^2}$, where N is a T-bit RSA modulus. This scheme is semantically secure under the decisional composite residuosity assumption (DCRA). For details on the encryption and decryption function we refer to [13]. The protocol for privacy-preserving face recognition proposed in [14] additionally uses the additively homomorphic cryptosystem of Damgård, Geisler and Krøigård (DGK) which reduces the ciphertext space to $\mathbb{Z}^*_N$ [10,11,12].

Oblivious Transfer (OT). For our construction we use parallel 1-out-of-2 Oblivious Transfer for m bitstrings of bitlength ℓ, denoted as OT^m_ℓ. It is a two-party protocol where the server $\mathcal{S}$ inputs m pairs of ℓ-bit strings $S_i = \langle s^0_i, s^1_i \rangle$ for $i = 1, .., m$ with $s^0_i, s^1_i \in \{0,1\}^\ell$. Client $\mathcal{C}$ inputs m choice bits $b_i \in \{0,1\}$. At the end of the protocol, $\mathcal{C}$ learns $s^{b_i}_i$, but nothing about $s^{1-b_i}_i$ whereas $\mathcal{S}$ learns nothing about b_i. We use OT^m_ℓ as a black-box primitive in our constructions. It can be instantiated efficiently with different protocols [29,1,27,23]. It is possible to pre-compute all OTs in a setup phase while the online phase consists of 2 messages with $\Theta(2mt)$ bits. Additionally, the number of computationally expensive public-key operations in the setup phase can be reduced to a constant number with the extensions of [23].

Garbled Circuit (GC). Yao's Garbled Circuit approach [40], excellently presented in [26], is the most efficient method for secure evaluation of a boolean circuit C. We summarize its ideas in the following. First, the circuit **constructor** (server $\mathcal{S}$), creates a *garbled circuit* $\widetilde{C}$ with algorithm $\mathsf{CreateGC}$: for each wire W_i of the circuit, he randomly chooses a *complementary garbled value* $\widehat{w}_i = \langle \widetilde{w}^0_i, \widetilde{w}^1_i \rangle$ consisting of two secrets, $\widetilde{w}^0_i$ and $\widetilde{w}^1_i$, where $\widetilde{w}^j_i$ is the *garbled value* of W_i's value j. (Note: $\widetilde{w}^j_i$ does not reveal j.) Further, for each gate G_i, $\mathcal{S}$ creates and sends to the **evaluator** (client $\mathcal{C}$) a *garbled table* $\widetilde{T}_i$ with the following property: given a set of garbled values of G_i's inputs, $\widetilde{T}_i$ allows to recover the garbled value of the corresponding G_i's output, and nothing else. Then garbled values corresponding to $\mathcal{C}$'s inputs x_j are (obliviously) transferred to $\mathcal{C}$ with a parallel oblivious transfer protocol OT (see below): $\mathcal{S}$ inputs complementary garbled values $\widetilde{W}_j$ into the protocol; $\mathcal{C}$ inputs x_j and obtains $\widetilde{w}_j^{x_j}$ as outputs. Now, $\mathcal{C}$ can evaluate the garbled circuit $\widetilde{C}$ with algorithm EvalGC to obtain the garbled output simply by evaluating the garbled circuit gate by gate, using the garbled tables $\widetilde{T}_i$. Finally, $\mathcal{C}$ determines the plain values corresponding to the obtained garbled output values using an output translation table received by $\mathcal{S}$. Correctness of GC follows from method of construction of garbled tables $\widetilde{T}_i$.

Implementation Details. For most efficient implementation of the garbled circuit we use several extensions of Yao's garbled circuit methodology as summarized in [35]: the *"free XOR"* trick of [25] allows "free" evaluation of XOR gates (no communication and negligible computation); for each non-XOR gate (e.g., AND, OR, ...) we use *garbled row reduction* [30,35] which allows to omit the first entry of the garbled tables, i.e., for each non-XOR gate (e.g., AND, OR, ...) with 2 inputs a garbled table of $\Theta(3t)$ bits is transferred; *point-and-permute* [28] allows fast GC evaluation, i.e., evaluation of a 2 input non-XOR gate requires in the random oracle model one invocation of a suitably chosen cryptographic hash function such as SHA-256. In the standard model, when the cryptographic hash function is correlation robust two invocations are needed [35].

Efficient Circuit Constructions. We use the following efficient circuit building blocks from [24] operating on ℓ-bit numbers: Addition ADD_ℓ, Subtraction SUB_ℓ, Comparison CMP_ℓ, and Multiplexer MUX_ℓ circuits of size ℓ non-XOR gates, and Multiplication circuits $\mathsf{MUL}_{\ell\times\ell}$ of size $|\mathsf{MUL}_{\ell\times\ell}| = 2\ell^2 - \ell$ non-XOR gates. Circuits can be automatically generated from a high-level description of how the circuit is composed from these building blocks using the compiler of [34].

2.3 Face Recognition using Eigenfaces

One of the well-known and efficient algorithms for face recognition is the so-called *Eigenfaces* algorithm introduced in [38,37]. This algorithm achieves reasonable classification rates of approximately 96% [14] and is simple enough to be implemented as privacy-preserving protocol (cf. §3). The Eigenfaces algorithm transforms face images into their characteristic feature vectors in a low-dimensional vector space (face space), whose basis consists of *Eigenfaces*. The Eigenfaces are determined through Principal Component Analysis (PCA) from a set of training images; every face is then represented as a vector in the face space by projecting the face image onto the subspace spanned by the Eigenfaces. Recognition is done by first projecting the face image into the face space and afterwards locating the closest feature vector. For details on the enrollment process we refer to [14] and the original papers on the Eigenface algorithm [38,37].

In the following we briefly summarize the recognition process of the Eigenfaces algorithm. A detailed pseudocode description of the algorithm and the naming conventions and sizes of parameters are given in Appendix §A.

Inputs and Outputs: The algorithm obtains as input the query face image $\mathit{\Gamma}$ represented as a pixel image with N pixels. Additionally, the algorithm obtains the parameters determined in the enrollment phase as inputs: the average face $\mathit{\Psi}$ which is the mean of all training images, the Eigenfaces $u_1, .., u_K$ which span the K-dimensional face space, the projected faces $\mathit{\Omega}_1, .., \mathit{\Omega}_M$ being the projections of the M faces in the database into the face space, and the threshold value τ.

The output r of the recognition algorithm is the index of that face in the database which is closest to the query face $\mathit{\Gamma}$ or the special symbol $\perp$ if no match was found, i.e., all faces have a larger distance than the threshold τ.

Recognition Algorithm: The recognition algorithm consists of three phases:

1. Projection: First, the average face $\mathit{\Psi}$ is subtracted from the face $\mathit{\Gamma}$ and the result is projected into the K-dimensional face space using the Eigenfaces $u_1, .., u_K$. The result is the projected K-dimensional face $\bar{\mathit{\Omega}}$.
2. Distance: Now, the square of the Euclidean distance D_i between the projected K-dimensional face $\bar{\mathit{\Omega}}$ and all projected K-dimensional faces in the database $\mathit{\Omega}_i$, $i = 1, .., M$, is computed.
3. Minimum: Finally, the minimum distance D_{min} is selected. If D_{min} is smaller than threshold τ, the index of the minimum value, i.e., the identifier i_{min} of the match found, is returned to $\mathcal{C}$ as result $r = i_{min}$. Otherwise, the image was not found and the special symbol $r = \perp$ is returned.

3 Privacy-Preserving Face Recognition

Privacy-Preserving Face Recognition allows a client to obliviously detect if the image of a face is contained in a database of faces held by a server. This can be achieved by securely evaluating a face recognition algorithm within a cryptographic protocol. In the following we concentrate on the Eigenface algorithm described in §2.3 which was also used in [14]. Our techniques can be extended to implement different recognition algorithms as discussed in §5.3.

3.1 Privacy-Preserving Face Recognition using Eigenfaces

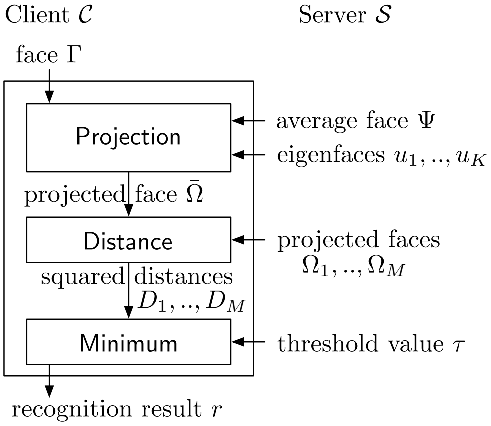
The inputs and outputs of the Eigenfaces algorithm are distributed between client $\mathcal{C}$ and server $\mathcal{S}$ as shown in Fig. 1(a). Both parties want to hide their inputs from the other party during the protocol run, i.e., $\mathcal{C}$ does not want to reveal for which face she is searching while $\mathcal{S}$ does not want to reveal the faces in his database or the details of the applied transformation into the face space.

In the semi-honest model we are working in, parties are assumed to follow the protocol but try to learn additional information from the protocol trace beyond what can be derived from the inputs and outputs of the algorithm when used as a black-box. In particular this requires that all internal results of the Eigenfaces algorithm, including the values passed between the different phases $\bar{\Omega}$ and $D_1, .., D_M$, are "hidden" from both parties. For practical applications it is sufficient to assume that both parties are computationally bounded, i.e., no polynomial-time adversary can derive information from "hidden" values.

For implementing the privacy-preserving Eigenfaces algorithm and "hiding" the intermediate values, different techniques can be used as listed in Fig. 1(b).

To the best of our knowledge, the only previous work on privacy-preserving face recognition [14] uses homomorphic encryption (HE) to implement the Eigenfaces algorithm in a privacy-preserving way, i.e., computations are performed on homomorphically encrypted data and the intermediate values are homomorphically encrypted (denoted as $[\![\cdot]\!]$). We summarize this protocol in §3.2.

Our Hybrid protocol presented in §4.1 substantially improves the efficiency of this protocol by implementing the Projection and Distance phase using homomorphic encryption and the Minimum phase with a garbled circuit. An alternative protocol which implements the entire recognition algorithm as garbled circuit and hides intermediate values as garbled values (denoted as $\widetilde{\cdot}$) is presented in §4.2. Our improvements over previous work are summarized in §5.



(a) Protocol Structure

	[14]	This Work	
Protocol	HE (§3.2)	Hybrid (§4.1)	GC (§4.2)
Projection	HE	HE	GC
↓	$[\![\bar{\Omega}]\!]$	$[\![\bar{\Omega}]\!]$	$\widetilde{\bar{\Omega}}$
Distance	HE	HE	GC
↓	$[\![D_i]\!]_{i=1}^M$	$[\![D_i]\!]_{i=1}^M$	$(\widetilde{D}_i)_{i=1}^M$
Minimum	HE	GC	GC

(b) Protocols and Applied Techniques

Fig. 1. Privacy-Preserving Face Recognition using Eigenfaces

3.2 Previous Work: Privacy-Preserving Face Recognition using HE

In [14], the authors describe describe a protocol for privacy-preserving face recognition which implements the Eigenfaces recognition algorithm of §2.3 on homomorphically encrypted data. Their protocol is secure in the semi-honest model, i.e., players are honest-but-curious [14, Appendix A].

Projection. First, $\mathcal{C}$ and $\mathcal{S}$ jointly compute the projection of the face image $\mathit{\Gamma}$ into the eigenspace spanned by the Eigenfaces $u_1, .., u_K$ as follows: $\mathcal{C}$ generates a secret/public key pair of a homomorphic encryption scheme (cf. §2.2) and encrypts the face $\mathit{\Gamma}$ as $\llbracket \mathit{\Gamma} \rrbracket = (\llbracket \mathit{\Gamma}_1 \rrbracket, .., \llbracket \mathit{\Gamma}_N \rrbracket)$. $\mathcal{C}$ sends the encrypted face $\llbracket \mathit{\Gamma} \rrbracket$ along with the public key to $\mathcal{S}$. Using the homomorphic properties, $\mathcal{S}$ projects the encrypted face into the low-dimensional face space and obtains the encryption of the projected face $\llbracket \bar{\mathit{\Omega}} \rrbracket = (\llbracket \bar{\omega}_1 \rrbracket, .., \llbracket \bar{\omega}_K \rrbracket)$ by computing for $i = 1, .., K$: $\llbracket \bar{\omega}_i \rrbracket = \llbracket -\sum_{j=1}^{N} u_{i,j} \mathit{\Psi}_j \rrbracket \cdot \prod_{j=1}^{N} \llbracket \mathit{\Gamma}_j \rrbracket^{u_{i,j}}$. The first factor can already be computed in the pre-computation phase. Additionally we observe that the values $\llbracket \bar{\omega}_i \rrbracket$ can be accumulated in parallel by using a parallel fast exponentiation algorithm which re-uses the same squared values of $\llbracket \mathit{\Gamma}_j \rrbracket$ in the square-and-multiply method.

Distance. After $\mathsf{Projection}$, $\mathcal{C}$ and $\mathcal{S}$ jointly compute the encryption of the Euclidean distances between the projected face $\llbracket \bar{\mathit{\Omega}} \rrbracket$ and all projected faces $\mathit{\Omega}_1, .., \mathit{\Omega}_M$ in the database held by $\mathcal{S}$. This is done by computing for $i = 1, .., M$: $\llbracket D_i \rrbracket = \llbracket ||\mathit{\Omega}_i - \bar{\mathit{\Omega}}||^2 \rrbracket = \llbracket S_{1,i} \rrbracket \cdot \llbracket S_{2,i} \rrbracket \cdot \llbracket S_3 \rrbracket$, where $\llbracket S_{1,i} \rrbracket = \llbracket \sum_{j=1}^{K} \omega_{i,j}^2 \rrbracket = \prod_{j=1}^{K} \llbracket \omega_{i,j}^2 \rrbracket$ and $\llbracket S_{2,i} \rrbracket = \llbracket \sum_{j=1}^{K} (-2\omega_{i,j}\bar{\omega}_j) \rrbracket = \prod_{j=1}^{K} \llbracket \bar{\omega}_j \rrbracket^{-2\omega_{i,j}}$ can be computed by $\mathcal{S}$ from $\llbracket \bar{\mathit{\Omega}} \rrbracket$ without interaction with $\mathcal{C}$. We note that the values $\llbracket S_{1,i} \rrbracket$ can be pre-computed entirely and online computation of $\llbracket S_{2,i} \rrbracket$ can be speeded up by accumulating these values in parallel in order to re-use the same squares in the square-and multiply exponentiation algorithm. To obtain $\llbracket S_3 \rrbracket = \llbracket \sum_{j=1}^{K} \bar{\omega}_j^2 \rrbracket$ from $\llbracket \bar{\mathit{\Omega}} \rrbracket$, the following protocol is suggested in [14]: For $j = 1, .., K$: $\mathcal{S}$ chooses $r_j \in_R \mathbb{Z}_n$, computes $\llbracket x_j \rrbracket = \llbracket \bar{\omega}_j + r_j \rrbracket = \llbracket \bar{\omega}_j \rrbracket \cdot \llbracket r_j \rrbracket$ and sends $\llbracket x_j \rrbracket$ to $\mathcal{C}$. $\mathcal{C}$ decrypts $\llbracket x_j \rrbracket$, computes $\llbracket S_3' \rrbracket = \llbracket \sum_{j=1}^{K} x_j^2 \rrbracket$, and sends $\llbracket S_3' \rrbracket$ to $\mathcal{S}$. $\mathcal{S}$ finally computes $\llbracket S_3 \rrbracket = \llbracket S_3' \rrbracket \cdot \llbracket -\sum_{j=1}^{K} r_j^2 \rrbracket \cdot \prod_{j=1}^{K} \llbracket \bar{\omega}_j \rrbracket^{-2r_j}$. The complexity of this protocol is summarized in §B.1.

Minimum. As last step, $\mathcal{C}$ and $\mathcal{S}$ jointly compute the minimum value D from $\llbracket D_1 \rrbracket, .., \llbracket D_M \rrbracket$ and its index Id. If the minimum value D is smaller than the threshold value τ known by $\mathcal{S}$, then $\mathcal{C}$ obtains the result Id. To achieve this, [14] suggests the following protocol: Choose the minimum value and index from the list of encrypted value and id pairs $(\llbracket D_0 = \tau \rrbracket, \llbracket \mathsf{Id}_0 = \perp \rrbracket), (\llbracket D_i \rrbracket, \llbracket \mathsf{Id}_i \rrbracket)_{i=1}^{M}$. For this, they apply a straight-forward recursive algorithm for minimum selection based on a sub-protocol which compares two encrypted distances and returns a re-randomized encryption of the minimum and its index to $\mathcal{S}$. For this sub-protocol, an optimized version of the homomorphic encryption-based comparison protocol of Damgård, Geisler and Krøigaard (DGK) [10,11,12] is used.

Complexity of Minimum *protocol (cf. Table 1).* The Minimum protocol of [14] requires a logarithmic number of $6\lceil \log_2(M+1) \rceil + 1$ moves. Overall, $8M$ Paillier ciphertexts and $2\ell' M$ DGK ciphertexts are sent in the online phase, where $\ell' = 50$ is the length of the squared distances $D_1, .., D_M$ among which the minimum is selected (cf. Table 5). This results in a communication complexity of $(16 + 2\ell')MT$ bits. The asymptotic online computation complexity is dominated by approximately $2M$ Paillier decryptions and $\ell' M$ DGK decryptions for $\mathcal{C}$ and the same number of exponentiations for $\mathcal{S}$.

4 Our Protocols for Privacy-Preserving Face Recognition

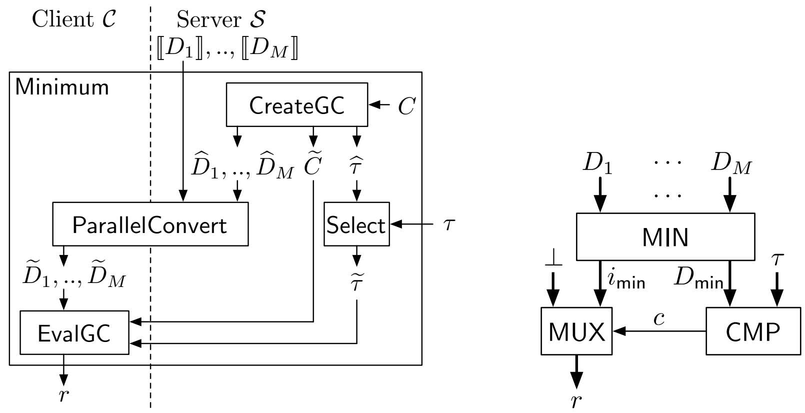
In the following we present two protocols which improve over the protocol of [14] (cf. §3.2) and are better suited for larger database sizes.

4.1 Privacy-Preserving Face Recognition using Hybrid of HE + GC

Our hybrid protocol for privacy-preserving face recognition improves over the protocol in [14] by replacing the $\mathsf{Minimum}$ protocol with a more efficient protocol based on garbled circuits. Additionally, the $\mathsf{Distance}$ protocol proposed in [14] can be slightly improved by packing together the messages sent from server $\mathcal{S}$ to client $\mathcal{C}$ into a single ciphertext as detailed in Appendix §B.2. We concentrate on the core improvements, the $\mathsf{Minimum}$ protocol, in the following.

Hybrid Minimum Protocol
The most efficient protocols for secure comparison in the setting with two computationally bounded parties is still based on Yao's garbled circuit (GC) approach [40,30,24] as briefly explained in §2.2. This also includes the natural generalization to selecting the minimum value and index of multiple values. As shown in [24], these GC based protocols clearly outperform comparison protocols based on homomorphic encryption [15,6,16,10,11,12]. In the following we show how the protocols of [24] can be adopted to yield a highly efficient, constant round $\mathsf{Minimum}$ protocol for our privacy-preserving face recognition protocol.

Overview. The high-level structure of our improved $\mathsf{Minimum}$ protocol is shown in Fig. 2(a) and consists of several building-blocks: the $\mathsf{ParallelConvert}$ subprotocol converts the homomorphically encrypted distances $[\![D_1]\!], .., [\![D_M]\!]$ held by server $\mathcal{S}$ into their corresponding garbled values $\widetilde{D}_1, .., \widetilde{D}_M$ output to client $\mathcal{C}$ (details below). These garbled values can then be used to evaluate a garbled circuit $\widetilde{C}_{\mathsf{Minimum}}$ which computes the $\mathsf{Minimum}$ phase of Algorithm 1 in Appendix §A (details on how the underlying circuit C_{Minimum} is constructed below). The garbled circuit $\widetilde{C}_{\mathsf{Minimum}}$ can be created already in the setup phase using algorithm $\mathsf{CreateGC}$ and sent to $\mathcal{C}$ before the online phase starts. The garbled values $\widetilde{\tau}$ which correspond to server's threshold value τ are selected by $\mathcal{S}$ (Select) and transferred to $\mathcal{C}$ as well (either in the setup phase or in the online phase depending on how often the database changes). Finally, $\mathcal{C}$ evaluates $\widetilde{C}_{\mathsf{Minimum}}$ on the garbled values $\widetilde{\tau}, \widetilde{D}_1, .., \widetilde{D}_M$ and obtains the correct output r.



(a) Protocol Structure with $C := C_{\mathsf{Minimum}}$.

(b) Circuit C_{Minimum}

Fig. 2. Improved Minimum Protocol

ParallelConvert *protocol.* An efficient ParallelConvert protocol is given in [24] which we summarize in the following (see [24] and [4] for a detailed description): $\mathcal{S}$ blinds the homomorphically encrypted ℓ'-bit values $[\![D_i]\!]$, $i = 1, .., M$ with a randomly chosen additive T-bit mask $R_i \in_R \mathbb{Z}_n$ and sends the blinded values $[\![D_i + R_i]\!]$ to $\mathcal{C}$ who can decrypt. Then, $\mathcal{C}$ and $\mathcal{S}$ jointly run a garbled circuit protocol in order to obliviously take off the mask R_i with a subtraction circuit. For improved efficiency, multiple values $[\![D_i]\!]$ can be packed together into a single ciphertext before blinding. To avoid an overflow when adding the T-bit random mask, the most significant κ bits are left as correctness margin, where κ is a statistical correctness parameter (e.g., $\kappa = 40$). This allows to pack $M' = \lfloor \frac{T-\kappa}{\ell'} \rfloor$ values into one ciphertext resulting in $m = \lceil \dfrac{M}{M'} \rceil$ packed Paillier ciphertexts for the M values. Overall, the ParallelConvert protocol consists of 3 moves.

Circuit C_{Minimum}. The circuit C_{Minimum} which computes the required functionality of the Minimum protocol is shown in Fig. 2(b): First, the minimum value $D_{\mathsf{min}} = min(D_1, .., D_M)$ and the corresponding index $i_{\mathsf{min}} \in \{1, .., M\}$ are computed with the MIN circuit. The MIN circuit is similar to the circuit evaluated in a first-price auction where the highest bid and the index of the highest bidder is selected [30]. An efficient construction of this circuit has size $|\mathsf{MIN}| \sim 2\ell' M$ non-XOR gates [24]. Afterwards, the minimum value D_{min} is compared with the threshold value τ using a comparison circuit CMP. The output c of the CMP circuit is 1 if $D_{\mathsf{min}} \leq \tau$ and 0 otherwise. Depending on c, the multiplexer MUX chooses either the minimum index i_{min} if $c = 1$ as output or the special symbol $\perp$ otherwise (e.g., $\perp = 0$). Overall, the circuit has size $|C_{\mathsf{Minimum}}| \sim 2\ell'(M-1)$ non-XOR gates.

Complexity. The complexity of our improved Minimum protocol and the Minimum protocol proposed in [14] is given in Table 1. For the computation complexity the table contains only the dominating costs: the number of Paillier and Damgård-Geisler-Krøigård (DGK) decryptions (Dec) and exponentiations (Exp) as well as the number of evaluations of a cryptographic hash function (Hash).

Table 1. Complexity of Minimum Protocols with Parameters M: # faces in database, ℓ': bitlength of values $D_1, .., D_M$, t: symmetric security parameter, T: asymmetric security parameter, κ: statistical correctness parameter, $m \sim \frac{\ell'}{T-\kappa}M$.

	HE §3.2 [14]	Hybrid §4.1
Round Complexity	$6\lceil\log(M+1)\rceil + 1$ moves	3 moves
Asymptotic Communication Complexity [bits]		
online	$(2\ell' + 16)MT$	$2\ell' Mt + 2mT$
offline		$\mathrm{OT}_t^{\ell' M} + 3(3\ell' - 1)Mt$
Asymptotic Computation Complexity		
$\mathcal{C}$ online	$\approx 2M$ $\mathrm{Dec}_{\mathsf{Paillier}} + \ell' M$ $\mathrm{Dec}_{\mathrm{DGK}}$	m $\mathrm{Dec}_{\mathsf{Paillier}} + (3\ell' - 1)M$ Hash
$\mathcal{S}$ online	$\approx 2M$ $\mathrm{Exp}_{\mathsf{Paillier}} + \ell' M$ $\mathrm{Exp}_{\mathrm{DGK}}$	m $\mathrm{Exp}_{\mathsf{Paillier}}$

Our improved Minimum protocol requires a constant number of 3 moves for the ParallelConvert protocol ($\widetilde{\tau}$ can be sent with the last message). The online communication complexity is determined by the ParallelConvert protocol for converting M values of bitlength ℓ', i.e., m Paillier ciphertexts and the online part of the $\mathrm{OT}_t^{\ell' M}$ protocol which is asymptotically $2\ell' Mt + 2mT$ bits (cf. §2.2). The online computation complexity requires $\mathcal{S}$ to pack the m ciphertexts (corresponds to m exponentiations) and $\mathcal{C}$ to decrypt them. After the OT protocol, $\mathcal{C}$ needs to evaluate a garbled circuit consisting of approximately $(3\ell' - 1)M$ non-XOR gates ($\ell' M$ to subtract the random masks in the ParallelConvert protocol and $(2\ell' - 1)M$ for C_{Minimum}) which requires to invoke a cryptographic hash function (e.g., SHA-256) the same number of times. The offline communication complexity consists of the $\mathrm{OT}_t^{\ell' M}$ protocol and transferring the garbled circuit which requires $3t$ bits per non-XOR gate (see §2.2).

Improvements (cf. Table 1). Most notably, the *round complexity* of our improved Minimum protocol is independent of the size M of the database.

The *online communication complexity* of our protocol is smaller by a factor of approximately T/t, e.g., $1024/80 \approx 13$ for short-term security and 38 for long-term security (see §5.1 for details).

The *online computation complexity* of our protocol is substantially smaller as well, as the number of Paillier operations is reduced by a factor of approximately $2M/m = 2M' = \frac{2(T-\kappa)}{\ell'}$, e.g., $\frac{2(1024-40)}{50} \approx 40$ for short-term security and 121 for long-term security. The evaluation of the garbled circuit in our protocol (which requires one invocation of SHA-256 per gate) is computationally less expensive than the modular arithmetics needed for the DGK public-key cryptosystem used in [14] (see §5.2 for details).

4.2 Privacy-Preserving Face Recognition using GC

Alternatively, the entire face recognition algorithm based on Eigenfaces described in §2.3 can be implemented in a garbled circuit. In this approach, $\mathcal{S}$ constructs a garbled circuit which evaluates the functionality. This circuit is composed from multipliers, adders, and the minimum selection circuit of §4.1 in a straightforward way as described in §C. $\mathcal{S}$ sends the garbled circuit to $\mathcal{C}$ in the precomputation phase and $\mathcal{C}$ obtains the garbled input values corresponding to his query face $\varGamma$ via OT. Additionally, $\mathcal{S}$ sends the garbled values corresponding to his private inputs $(\varPsi, u_1, .., u_K, \varOmega_1, .., \varOmega_M, \tau)$ to $\mathcal{C}$. This can be done either in the offline phase if these parameters are fixed or in the online phase if the database is changed frequently. Finally, $\mathcal{C}$ evaluates the garbled circuit on the garbled inputs and obtains the classification result r.

Complexity. Our GC-based protocol for privacy-preserving face recognition requires a parallel OT protocol for $8N = 82,432$ garbled values as the query face $\varGamma$ consists of N pixels of 8 bit each. Additionally, server $\mathcal{S}$ transfers the garbled values corresponding to his $8N + 8KN + 32KM + 50 = 1,071,666 + 384 \cdot M$ input bits to client $\mathcal{C}$. The online phase of the protocol requires 2 moves for the online part of the OT protocol. As explained in §C, the evaluated garbled circuit C consists of approximately $19,866,112 + 25,660 \cdot M$ non-XOR gates.

5 Complexity Improvements

In the following we compare the communication- and round complexity of our improved protocols with the protocol proposed in [14]. The computation complexity is compared experimentally in §5.2.

Parameter Sizes. We compare the complexity of both protocols for different recommended sizes of security parameters – short-term (recommended use up to 2010), medium-term (up to 2030) and long-term security [18]. The sizes for the security parameters and corresponding parameter sizes for our Hybrid protocol are summarized in Table 2: we set statistical security parameter $\sigma = 80$ and statistical correctness parameter $\kappa = 40$. According to Table 5, the input length of the $\mathsf{Distance}$ protocol (§B.2) is $\ell = 32$ and that of the $\mathsf{Minimum}$ protocol (§4.1) is $\ell' = 50$.

5.1 Round Complexity and Asymptotic Communication Complexity

HE vs. Hybrid (Table 3). Our Hybrid protocol substantially improves the performance of the HE protocol proposed in [14]: the round complexity is reduced

Table 2. Size of Security Parameters (t: symmetric security parameter, T: asymmetric security parameter) and Corresponding Parameters for Hybrid Protocol (K': # blinded values packed into one ciphertext, k: # ciphertexts, M': # values packed into one ciphertext before blinding).

Security Level	Security Parameters		Distance (§B.2)		Minimum (§4.1)
	t	T	K'	k	M'
Short-Term	80	1024	8	2	19
Medium-Term	112	2048	17	1	40
Long -Term	128	3072	26	1	60

from logarithmic in the size of the database M down to a small constant of 6 moves. The online communication complexity of the Minimum protocol (§4.1) is reduced down to only 6.6% of the previous solution for short-term security. For medium- and long-term security the savings are even better. Our improvements of the Distance protocol (§B.2) down to 23% for short-term security are negligible w.r.t. the overall communication complexity as the communication complexity of this protocol is small (few KBytes) and independent of the *size* M of the database.

Table 3. Comparison of Round- and Communication Complexity – HE vs. Hybrid. M: # faces in database.

Protocol	HE §3.2 [14]			Hybrid §4.1 (Improvement)		
Round Complexity [moves]	$6\lceil\log(M+1)\rceil+4$			6 $(\mathcal{O}(\log M) \rightarrow \mathcal{O}(1))$		
Security Level	Short	Medium	Long	Short	Medium	Long
Asymptotic Communication Complexity (online)						
Projection [MB]	2.5	5.0	7.5	2.5	5.0	7.5
Distance [kB]	3.2	6.5	9.8	0.75 (23%)	1.0 (15%)	1.5 (15%)
Minimum [kB per face in DB]	15	29	44	0.99 (6.6%)	1.4 (4.8%)	1.6 (3.6%)

Hybrid vs. GC (Table 4). Our GC-based protocol requires only two moves for OT. In fact, the GC protocol could even be executed without any interaction when using a trusted hardware token [21] (this was called one-time program in [19]). If the database is static, i.e., no online updates are performed, the online communication complexity of this protocol does not depend on the size of the database, while with online updates it is by a factor of approximately 3 larger than that of the Hybrid protocol (see numbers in parentheses). The major drawback of the GC protocol is its huge offline communication complexity of several hundreds of Megabytes compared to few Kilobytes in the Hybrid solution.

Table 4. Comparison of Round- and Communication Complexity – Hybrid vs. GC.

Protocol	Hybrid §4.1			GC §4.2 (with online update)		
Round Complexity [moves]	6			2		
Security Level	Short	Medium	Long	Short	Medium	Long
Asymptotic Communication Complexity (online)						
base [MB]	2.5	5.0	7.5	1.6 (+10)	2.2 (+14)	2.5 (+16)
per face in DB [kB]	0.99	1.4	1.6	0 (+3.8)	0 (+5.3)	0 (+6.0)
Asymptotic Communication Complexity (offline) without OT						
base	8.0 kB	16 kB	20 kB	189 MB	265 MB	303 MB
per face in DB	6.4 kB	8.9 kB	10 kB	0.24 MB	0.34 MB	0.39 MB

5.2 Online Computation Complexity

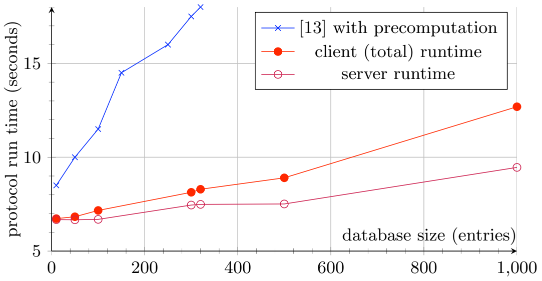
Hybrid protocol (§4.1). We have implemented our Hybrid protocol for privacy-preserving face recognition described in §4.1 in Python in order to quantify its online computation complexity. The decision for Python was made since it is platform independent and code can be run on various architectures without modification. In principle, interpreted Python programs run substantially slower than compiled code. We perform performance measurements on two standard PCs (AMD Athlon64 X2 5000+ (2.6GHz), 2 Cores, 4 GB Memory running on Gentoo Linux x86_64) communicating via TCP/IP6 over a Gigabit Ethernet connection. Both machines were clocked to 2.4GHz via CPU frequency scaling to make the performance comparable to [14]. The implementation is running in the cPython-2.6 interpreter and uses the gmpy module (version 1.04) that allows to access the GNU GMP library (version 4.3.1) within Python.

In comparison, the protocol in [14] was implemented in C++ using the GNU GMP library and executed on a single PC (2.4 GHz AMD Opteron with dual-core processor and 4 GB RAM under Linux) as two threads. This implementation neglects latencies of the communication stack and the network which would result in non-negligible slow-downs due to the logarithmic round complexity of their protocol.

Although our implementation is closer to a real-world setting and uses a substantially slower programming language, it still outperforms that of [14] especially for larger database sizes due to our algorithmic protocol improvements of the Minimum protocol as shown in Fig. 3(a). Surprisingly, our implementation is about 30% faster than the C++ implementation of [14] even in the unchanged parts of our protocol which use homomorphic encryption. Presumably this is due to faster multiplication in GMP version 4.3.

As shown in Fig. 3(b), our protocol scales well with increasing security level which is not the case for the protocol based on homomorphic encryption of [14] as the asymmetric security parameter T increases much faster than the symmetric security parameter t (cf. Table 2).

Overall, our implementation results confirm that our Hybrid protocol for privacy-preserving face recognition can be used for practical privacy-preserving face recognition even for large databases.



	Security Level		
Client	Short	Medium	Long
Projection	0.49	0.60	0.72
Distance	6.08	16.87	31.73
Minimum	1.86	2.71	4.49
Sum	8.43	20.18	36.95
Server	Short	Medium	Long
Projection	6.58	17.43	32.37
Distance	0.47	1.52	3.03
Minimum	0.06	0.21	0.54
Sum	7.11	19.15	35.94

(a) HE vs. Hybrid Protocol (Short-Term Security) (b) Hybrid Protocol for $M = 320$

Fig. 3. Comparison of Timing Complexity in [s]

Garbled Circuit protocol (§4.2). Unfortunately we were not able to compile the circuit that is evaluated in the GC-based protocol of §4.2 due to memory restrictions of the compiler of [34]. From our implementation of the GC-based Minimum phase of our Hybrid protocol we estimate the GC protocol to be slower than the Hybrid protocol (in the order of several minutes).

5.3 Future Work

The methods for constructing efficient protocols for privacy-preserving face recognition presented in this paper can be further improved into various directions. We plan to extend our prototype implementation into a general purpose framework for secure and efficient protocols based on a combination of homomorphic encryption and garbled circuits which will support future research.

Algorithmic Improvements for better classification accuracy might be achieved by using different face recognition algorithms. In the Projection phase, Fisherfaces [5] can be used instead of Eigenfaces to determine the projection matrix with Linear Discriminant Analysis (LDA) instead of Principal Component Analysis (PCA). In the Distance phase, a different distance metric than Euclidean distance could be used, e.g., Hamming distance or Manhattan (city block) distance. The Minimum phase could be based on meaning or scoring instead of minimum selection.

Further Protocol Improvements could be achieved with a different homomorphic encryption scheme that allows both, additions and multiplications [7,2,17] to avoid the additional communication round for computing Euclidean Distance.

Further Implementation Improvements can be achieved by exploiting parallelism on multi-core architectures or graphics processing units (GPUs).

Acknowledgements We thank Wilko Henecka for extending the compiler of [34] to generate the underlying circuits, authors of [14] for detailed information on their protocol, and anonymous reviewers of ICISC 2009 for helpful comments.

References

1. W. Aiello, Y. Ishai, and O. Reingold. Priced oblivious transfer: How to sell digital goods. In *Advances in Cryptology – EUROCRYPT'01*, volume 2045 of *LNCS*, pages 119–135. Springer, 2001.
2. F. Armknecht and A.-R. Sadeghi. A new approach for algebraically homomorphic encryption. Cryptology ePrint Archive, Report 2008/422, 2008. `http://eprint.iacr.org/`.
3. S. Avidan and M. Butman. Efficient methods for privacy preserving face detection. In *Advances in Neural Information Processing Systems (NIPS'06)*, pages 57–64. MIT Press, 2006.
4. M. Barni, P. Failla, V. Kolesnikov, R. Lazzeretti, A.-R. Sadeghi, and T. Schneider. Secure evaluation of private linear branching programs with medical applications. In *14th European Symposium on Research in Computer Security (ESORICS'09)*, LNCS. Springer, 2009. Full version available at `http://eprint.iacr.org/2009/195`.
5. P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720, 1997.
6. I. F. Blake and V. Kolesnikov. Strong conditional oblivious transfer and computing on intervals. In *Advances in Cryptology – ASIACRYPT'04*, volume 3329 of *LNCS*, pages 515–529. Springer, 2004.
7. D. Boneh, E.-J. Goh, and K. Nissim. Evaluating 2-dnf formulas on ciphertexts. In *Theory of Cryptography (TCC'05)*, volume 3378 of *LNCS*, pages 325–341. Springer, 2005.
8. O. Bowcott. Interpol wants facial recognition database to catch suspects. Guardian (October 20, 2008), `http://www.guardian.co.uk/world/2008/oct/20/interpol-facial-recognition`.
9. X. Boyen, Y. Dodis, J. Katz, R. Ostrovsky, and A. Smith. Secure remote authentication using biometric data. In *Advances in Cryptology – EUROCRYPT'05*, volume 3494 of *LNCS*, pages 147–163. Springer, 2005.
10. I. Damgård, M. Geisler, and M. Krøigård. Efficient and secure comparison for on-line auctions. In *Australasian Conference on Information Security and Privacy (ACISP'07)*, volume 4586 of *LNCS*, pages 416–430. Springer, 2007.
11. I. Damgård, M. Geisler, and M. Krøigård. A correction to "efficient and secure comparison for on-line auctions". Cryptology ePrint Archive, Report 2008/321, 2008. `http://eprint.iacr.org/`.
12. I. Damgård, M. Geisler, and M. Krøigård. Homomorphic encryption and secure comparison. *Journal of Applied Cryptology*, 1(1):22–31, 2008.
13. I. Damgård and M. Jurik. A generalisation, a simplification and some applications of paillier's probabilistic public-key system. In *Public-Key Cryptography (PKC'01)*, LNCS, pages 119–136. Springer, 2001.

14. Z. Erkin, M. Franz, J. Guajardo, S. Katzenbeisser, I. Lagendijk, and T. Toft. Privacy-preserving face recognition. In *Privacy Enhancing Technologies (PET'09)*, volume 5672 of *LNCS*, pages 235–253. Springer, 2009.
15. M. Fischlin. A cost-effective pay-per-multiplication comparison method for millionaires. In *Cryptographer's Track at RSA Conference (CT-RSA'01)*, volume 2020 of *LNCS*, pages 457–472. Springer, 2001.
16. J. A. Garay, B. Schoenmakers, and J. Villegas. Practical and secure solutions for integer comparison. In *Public Key Cryptography (PKC'07)*, volume 4450 of *LNCS*, pages 330–342. Springer, 2007.
17. C. Gentry. Fully homomorphic encryption using ideal lattices. In *ACM Symposium on Theory of Computing (STOC'09)*, pages 169–178. ACM, 2009.
18. D. Giry and J.-J. Quisquater. Cryptographic key length recommendation, March 2009. `http://keylength.com`.
19. S. Goldwasser, Y. T. Kalai, and G. N. Rothblum. One-time programs. In *Advances in Cryptology – CRYPTO'08*, volume 5157 of *LNCS*, pages 39–56. Springer, 2008.
20. T. Grose. When surveillance cameras talk, 2008. Time Magazine (February 11, 2008), `http://www.time.com/time/world/article/0,8599,1711972,00.html`.
21. V. Gunupudi and S. R. Tate. Generalized non-interactive oblivious transfer using count-limited objects with applications to secure mobile agents. In *Financial Cryptography and Data Security (FC'08)*, volume 5143 of *LNCS*, pages 98–112. Springer, 2008.
22. Interational Civil Aviation Organization (ICAO). Machine Readable Travel Documents (MRTD), Doc 9303, Part 1 Machine Readable Passports, Fifth Edition, 2003.
23. Y. Ishai, J. Kilian, K. Nissim, and E. Petrank. Extending oblivious transfers efficiently. In *Advances in Cryptology – CRYPTO'03*, volume 2729 of *LNCS*. Springer, 2003.
24. V. Kolesnikov, A.-R. Sadeghi, and T. Schneider. Improved garbled circuit building blocks and applications to auctions and computing minima. In *Cryptology and Network Security (CANS '09)*, LNCS. Springer, 2009. Full version available at `http://eprint.iacr.org/2009/411`.
25. V. Kolesnikov and T. Schneider. Improved garbled circuit: Free XOR gates and applications. In *International Colloquium on Automata, Languages and Programming (ICALP'08)*, volume 5126 of *LNCS*, pages 486–498. Springer, 2008.
26. Y. Lindell and B. Pinkas. A proof of Yao's protocol for secure two-party computation. ECCC Report TR04-063, Electronic Colloquium on Computational Complexity (ECCC), 2004.
27. H. Lipmaa. Verifiable homomorphic oblivious transfer and private equality test. In *Advances in Cryptology – ASIACRYPT'03*, volume 2894 of *LNCS*. Springer, 2003.
28. D. Malkhi, N. Nisan, B. Pinkas, and Y. Sella. Fairplay — a secure two-party computation system. In *USENIX*, 2004. `http://fairplayproject.net`.
29. M. Naor and B. Pinkas. Efficient oblivious transfer protocols. In *ACM-SIAM Symposium On Discrete Algorithms (SODA'01)*, pages 448–457. Society for Industrial and Applied Mathematics, 2001.
30. M. Naor, B. Pinkas, and R. Sumner. Privacy preserving auctions and mechanism design. In *ACM Conference on Electronic Commerce*, pages 129–139, 1999.
31. I. Naumann and G. Hogben. Privacy features of european eid card specifications. *Network Security*, 2008(8):9–13, 2008. European Network and Information Security Agency (ENISA).

32. E. M. Newton, L. Sweeney, and B. Malin. Preserving privacy by de-identifying face images. *IEEE Transactions on Knowledge and Data Engineering*, 17(2):232–243, 2005.
33. P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Advances in Cryptology – EUROCRYPT'99*, volume 1592 of *LNCS*, pages 223–238. Springer, 1999.
34. A. Paus, A.-R. Sadeghi, and T. Schneider. Practical secure evaluation of semi-private functions. In *Applied Cryptography and Network Security (ACNS'09)*, volume 5536 of *LNCS*, pages 89–106. Springer, 2009. `http://www.trust.rub.de/FairplaySPF`.
35. B. Pinkas, T. Schneider, N. P. Smart, and S. C. Williams. Secure two-party computation is practical. In *Advances in Cryptology – ASIACRYPT 2009*, LNCS. Springer, 2009. Full version available at `http://eprint.iacr.org/2009/314`.
36. A.-R. Sadeghi, T. Schneider, and I. Wehrenberg. Efficient privacy-preserving face recognition. In *12th International Conference on Information Security and Cryptology (ICISC '09)*, LNCS. Springer, 2009.
37. M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.
38. M. Turk and A. Pentland. Face recognition using eigenfaces. In *IEEE Computer Vision and Pattern Recognition (CVPR'91)*, pages 586–591. IEEE, 1991.
39. P. Tuyls, A. Akkermans, T. Kevenaar, G.-J. Schrijen, A. Bazen, and R. Veldhuis. Practical biometric authentication with template protection. In *Audio- and Video-Based Biometric Person Authentication*, volume 3546 of *LNCS*, pages 436–446. Springer, 2005.
40. A. C. Yao. How to generate and exchange secrets. In *IEEE Symposium on Foundations of Computer Science (FOCS'86)*, pages 162–167. IEEE, 1986.

A Face Recognition using Eigenfaces: Details

Algorithm 1 shows the pseudocode description of the Eigenfaces algorithm and Table 5 the naming conventions and sizes of the parameters.

Parameter	Size [14]	Description
M		number of faces in database
N	$= 10304$	size of a face in pixels
K	$= 12$	number of Eigenfaces
$\varGamma, \varPsi$	$\in [0, 2^8 - 1]^N$	face, average face
$u_1, .., u_K$	$\in [-2^7, 2^7 - 1]^N$	Eigenfaces
$\bar{\varOmega}, \varOmega_1, .., \varOmega_M$	$\in [-2^{31}, 2^{31} - 1]^K$	projected face, projected faces in database
$D_1, .., D_M$	$\in [0, 2^{50} - 1]$	squared distances between projected images
τ	$\in [0, 2^{50} - 1]$	threshold value

Table 5. Parameters and Sizes for Privacy-Preserving Face Recognition

Algorithm 1 Face recognition using Eigenfaces [38,37].

```
Input  face $\mathit{\Gamma}$, average face $\mathit{\Psi}$; Eigenfaces $u_1, .., u_K$; projected faces $\mathit{\Omega}_1, .., \mathit{\Omega}_M$; threshold value $\tau$
Output  recognition result $r \in \{1, .., M\} \cup \perp$

{Phase 1: Projection}
for $i = 1$ to $K$ do
    $\bar{\omega}_i = u_i^T(\mathit{\Gamma} - \mathit{\Psi})$
end for
projected face $\bar{\mathit{\Omega}} := (\bar{\omega}_1, .., \bar{\omega}_K)$

{Phase 2: Distance}
for $i = 1$ to $M$ do
    compute squared distance $D_i = ||\bar{\mathit{\Omega}} - \mathit{\Omega}_i||^2 = \sum_{j=1}^{K}(\bar{\omega}_j - \omega_{i,j})^2$
end for

{Phase 3: Minimum}
compute minimum value $D_{\mathsf{min}} = \min\{D_1, .., D_M\}$ and index $i_{\mathsf{min}}$: $D_{\mathsf{min}} = D_{i_{\mathsf{min}}}$
if $D_{\mathsf{min}} \leq \tau$ then
    Return $r = i_{\mathsf{min}}$
else
    Return $r = \perp$
end if
```

B Distance Protocol Based on Homomorphic Encryption

B.1 Complexity of Distance Protocol Based on Homomorphic Encryption (cf. §3.2).

The interactive part of the Distance protocol which computes the sum of squares $\llbracket S_3 \rrbracket$ has the following complexity: the first message consists of K Paillier ciphertexts $\llbracket x_j \rrbracket$, $j = 1, .., K$ of size $2T$ bit each (cf. §2.2), and the second message is one Paillier ciphertext $\llbracket S_3' \rrbracket$. $\mathcal{C}$ performs K Paillier decryptions of $\llbracket x_j \rrbracket$ and one encryption of $\llbracket S_3' \rrbracket$ while $\mathcal{S}$ computes K exponentiations with the exponents $-2r_j$ which are slightly longer than T bits. We will show how to improve this protocol later in §B.2.

B.2 Our Improved Sum of Squares Protocol

In the following we improve the Distance protocol proposed in [14] which computes the Euclidean distance. For this, we reduce the complexity of the subprotocol which computes the encrypted sum of squares $\llbracket S_3 \rrbracket = \llbracket \sum_{j=1}^{K'} \bar{\omega}_j^2 \rrbracket$ from $\llbracket \bar{\omega}_1 \rrbracket, .., \llbracket \bar{\omega}_{K'} \rrbracket$. Our improvements result from choosing shorter random masks and packing of multiple ciphertexts as described in the following.

Shorter random masks. In contrast to the protocol proposed in [14] our improved protocol blinds the values with random masks r_j which are substantially shorter than those proposed in [14] which are chosen from the full plaintext domain.

Our random masks r_j are longer than the blinded ℓ-bit values $\bar{\omega}_j$ by σ' bits, i.e., $r_j \in_R \{0,1\}^{\ell+\sigma'}$. These smaller random masks reduces the computation complexity of the protocol.

Packing. The resulting blinded values $x_j = \bar{\omega}_j + r_j$ are σ' bit values (an overflow occurs with probability $2^{-\sigma'}$ which is negligible as described later). These blinded values can be packed together into a single ciphertext under encryption. This reduces the communication complexity as the packed ciphertext now carries multiple blinded values as well as the computation complexity of $\mathcal{C}$ as he needs to decrypt only a single ciphertext. The number of blinded values which can be packed into one ciphertext is

$$K' = \lfloor \frac{T}{\ell + \sigma'} \rfloor. \tag{1}$$

The statistical difference between the packed ciphertext and a random $K'(\ell + \sigma')$-bit string is $K' \cdot 2^{-\sigma'}$, as they differ only if one of the K' packed values overflows. If we upper-bound the statistical distance by $2^{-\sigma}$, where σ is a statistical security parameter (e.g., $\sigma = 80$) we obtain the following relation which determines σ' and K' in (1):

$$K' 2^{-\sigma'} \leq 2^{-\sigma}. \tag{2}$$

Our improved protocol for computing the encrypted sum of squares $[\![S_3]\!] = [\![\sum_{j=1}^{K'} \bar{\omega}_j^2]\!]$ from $[\![\bar{\omega}_1]\!], .., [\![\bar{\omega}_{K'}]\!]$ works as follows: For $j = 1, .., K'$, $\mathcal{S}$ chooses $r_j \in_R \{0,1\}^{\ell+\sigma'}$ and computes $[\![x]\!] = [\![\sum_{j=1}^{K'} 2^{(\ell+\sigma')(j-1)}(\bar{\omega}_j + 2^{\ell-1} + r_j)]\!] = [\![\sum_{j=1}^{K'} 2^{(\ell+\sigma')(j-1)}(2^{\ell-1} + r_j)]\!] \cdot \prod_{j=1}^{K'} [\![\bar{\omega}_j]\!]^{2^{(\ell+\sigma')(j-1)}}$. (Note that by adding $2^{\ell-1}$, the signed ℓ-bit integer values $\bar{\omega}_j \in [-2^{\ell-1}, 2^{\ell-1} - 1]$ are shifted into unsigned ℓ-bit integer values $\bar{\omega}'_j \in [0, 2^{\ell} - 1]$.) $\mathcal{S}$ sends $[\![x]\!]$ to $\mathcal{C}$ who decrypts and obtains x which is unpacked by parsing it into $(\ell + \sigma')$-bit chunks as $x = x_{K'} || .. || x_1$ with $x_j \in \{0,1\}^{\ell+\sigma'}$. Afterwards, $\mathcal{C}$ computes $[\![S'_3]\!] = [\![\sum_{j=1}^{K'} (x_j - 2^{\ell-1})^2]\!]$ and sends this to $\mathcal{S}$ who can compute $[\![S_3]\!]$ as in the protocol proposed in [14]: $[\![S_3]\!] = [\![S'_3]\!] \cdot [\![- \sum_{j=1}^{K'} r_j^2]\!] \cdot \prod_{j=1}^{K'} [\![\bar{\omega}_j]\!]^{-2r_j}$.

This protocol can easily be extended to compute the sum of $K > K'$ squares by executing it $k := \lceil \frac{K}{K'} \rceil$ times in parallel where the message sent from $\mathcal{C}$ to $\mathcal{S}$ consists of the single ciphertext $[\![S'_3]\!] = [\![\sum_{j=1}^{K} (x_j - 2^{\ell-1})^2]\!]$.

We note that our improved protocol for computing the sum of squares can easily be extended into an improved protocol for parallel squaring or parallel multiplications in a straight-forward way.

Correctness and Security. It is easy to verify the correctness of the improved sum of squares protocol. The security in the semi-honest model can be proven using standard techniques.

Complexity. The overall complexity of our improved sum-of-squares protocol and the protocol proposed in [14] is given in Table 6. For the computation complexity the table contains only the dominating costs – the number of Paillier encryptions (enc), decryptions (dec) and exponentiations with an exponent of length T (exp).

Table 6. Complexity of Protocols for Computing the Sum of Squares with parameters T: asymmetric security parameter, K: # values to be squared, $k < K$: # packed ciphertexts.

	[14]	This Work
Round Complexity [moves]	2	
Communication Complexity [bits]		
Message $\mathcal{C} \leftarrow \mathcal{S}$	$K \cdot 2T$	$k \cdot 2T$
Message $\mathcal{C} \rightarrow \mathcal{S}$	$2T$	
Asymptotic Computation Complexity		
$\mathcal{C}$ online	K $\mathrm{Dec}_{\mathrm{Paillier}}$ + 1 $\mathrm{Enc}_{\mathrm{Paillier}}$	k $\mathrm{Dec}_{\mathrm{Paillier}}$ + 1 $\mathrm{Enc}_{\mathrm{Paillier}}$
$\mathcal{S}$ online	K $\mathrm{Exp}_{\mathrm{Paillier}}$	$k+1$ $\mathrm{Exp}_{\mathrm{Paillier}}$

Overall, the first message of our improved protocol which is run k times in parallel consists of k Paillier ciphertexts $[\![x]\!]$ which are decrypted by $\mathcal{C}$. When $\mathcal{S}$ packs these ciphertexts together, the product $\prod_{j=1}^{K'} [\![\bar{\omega}_j]\!]^{2^{(\ell+\sigma')(j-1)}}$ can be computed efficiently such that its computation complexity corresponds to less than one exponentiation with an exponent of length T using Horner's method:

```
$s = 2^{\ell+\sigma'}$; $[\![x]\!] = [\![\bar{\omega}_{K'}]\!]$
for $j = K' - 1$ downto 1 do
    $[\![x]\!] = [\![x]\!]^s \cdot [\![\bar{\omega_j}]\!]$
end for
```

In the preprocessing phase, $\mathcal{S}$ can compute the sum $\sum_{j=1}^{K'} 2^{(\ell+\sigma')(j-1)}(2^{\ell-1} + r_j)$ also efficiently with Horner's method before encryption. Finally, $\mathcal{S}$ needs to perform the equivalent of k exponentiations with T-bit exponents due to the shorter random values r_j.

Improvements. Our improved protocol reduces the communication complexity (see §5 for details) as well as the online computation complexity (see §5.2 for details) of both parties by roughly a factor of K'.

C Privacy-Preserving Face Recognition using GC: Circuit

The circuit C which evaluated in our protocol for privacy-preserving face recognition based on Eigenfaces and GC (§4.2) is directly derived from the Eigenfaces algorithm Algorithm 1 described in §2.3.

In the Projection phase, the value $\Gamma - \Psi$ is computed which requires N subtractors for 8 bit strings. To compute each 32-bit value $\bar{\omega_i}$, $i = 1, .., K$, this difference is multiplied with the vector u_i^T consisting of N 8-bit values. This requires $KN(\mathsf{MUL}_{8\times 8} + \mathsf{ADD}_{32})$.

The Distance phase computes the squared Euclidean distance D_i (50-bit) between $\bar{\Omega} = (\bar{\Omega}_1, .., \bar{\Omega}_K)$ to each of the M projected faces $\Omega_i = (\omega_{i,1}, .., \omega_{i,K})$ in the database where each component has size 32-bit: $D_i = \sum_{j=1}^{K}(\bar{\omega}_j - \omega i, j)^2$. This requires $MK(\mathsf{SUB}_{32} + \mathsf{MUL}_{32\times 32} + \mathsf{ADD}_{50})$.

Finally, the Minimum phase selects the minimum value and index of these $\ell' = 50$-bit squared distances $D_1, \ldots, D_M$ and returns the minimum index if the minimum value is less than the threshold τ using the circuit C_{Minimum} described in §4.1. This circuit has size $C_{\mathsf{Minimum}} \sim 2\ell' M$ non-XOR gates.

Overall, the circuit C has size $|C| \sim 8N + KN(2 \cdot 8 \cdot 8 + 32) + MK(32 + 2 \cdot 32 \cdot 32 + 50) + 2\ell' M$ non-XOR gates, i.e., $|C| \approx 19866112 + 25660 \cdot M$ non-XOR gates when choosing the parameters according to Table 5.