Privacy-Preserving Face Recognition in Large-Scale Video Surveillance Systems

William Koch
École Polytechnique*

Supervisor: Matthieu Rambaud
Referent Instructor: Emmanuel Haucourt

December 14, 2020

Abstract

The use of video surveillance, let alone face recognition technologies, in public spaces is highly controversial. While some claim that face recognition is essential to effectively combat crime and terrorism, others fear the potential abuse of such systems and how it may impact the freedom of the individual.

We propose a video surveillance framework which maintains the advantages of face recognition while making mass-surveillance as we know it impossible. By leveraging recent work on actively secure multi-party computation in a two-party setting, the privacy of our framework relies on involving two independent parties in the computations. This ensures that a person (such as a terrorist) can only be identified and tracked if all parties agree to contribute their results. It follows that a party cannot solely decide to track specific individuals as this would be detected by the other party. We furthermore address important issues specific to this application such as the implications of corrupted parties and the possibility of reconstructing movement profiles.

Our results show that the offline phase is currently too expensive for large-scale real-time applications, and may need to involve a trusted crypto provider instead. The online phase however performs very well when performing matches against a criminal database with 150 entries. Using only four threads, ten faces can be checked every second without revealing the identity. Due to the parallelizability of the operations, our framework can already work over predefined limited time frames.

*Thesis written at Télécom Paris

Contents

1 Introduction

1.1 Use Case and Relevance

Video surveillance systems are currently used in many different situations, and we can distinguish between the use in private and in public spaces. However, it is often the latter that is subject to many debates. Video surveillance in public spaces such as airports, train stations or city squares highlights the dilemma between **privacy** and **security**. In times where a nation faces terrorism and increasing violence, politicians tend to favor security over privacy, as seen with the Patriot Act in the US after the 9/11 attacks for example. On the other hand, some cities such as San Francisco have banned the use of face recognition in their video surveillance systems entirely for fear of mass surveillance [9].

The fundamental application that is driving our work are full- or partial-coverage video surveillance systems on a local level (such as a city). While many of our results can certainly also be applied to single CCTV cameras, we specifically address and mitigate privacy concerns in larger connected systems. The objective is to respond to the dilemma between privacy and security by making mass surveillance cryptographically impossible while maintaining the same security benefits.

1.2 Our Contributions

Our primary contribution consists of defining the requirements and analyzing the potential threats of privacy-preserving face recognition, specific to video surveillance in public spaces. We use these results to propose a first framework for video surveillance which relies on a system of checks and balances by leveraging existing work on **actively secure** multi-party computation. While there exist other privacy-preserving implementations for face recognition, our implementation is, to the best of our knowledge, the first to block mass-surveillance by introducing an independent observing party which participates in the computations and has a copy of the database. This allows the observing party to detect if the other performs any modifications to the database.

While none of the involved parties is able to reconstruct biometric data from the CCTV cameras directly, all parties are aware of the number of people visible in the camera frame. Especially with very few people at night, a party might be able to reconstruct trajectories of individuals and thus gain information on the identity of the person. We introduce an obfuscation method that addresses this concern.

Lastly, we use these results to build a simplified example to demonstrate the feasibility of the framework and measure its performance over a local network.

1.3 Related Work

SCiFI [11] proposes a solution to a similar problem in face recognition. They introduce a system which allows a client (who has the biometric face data) to perform matches with a database on the server, without the server sharing the database or the client sharing the biometric data. However, the proposed system is only semi-honest and does not address the problem of mass-surveillance, as the server can arbitrarily choose its database. [3] also focuses on access control systems using Eigenenfaces for secure biometric authentication. As before, the system is only semi-honest and matching a single face with a database of 20 entries takes them approximately 25 seconds. [12] also uses the Eigenface recognition algorithm, and relies on Homomorphic Encryption and Garbled Circuits to perform the secure computations. [22] uses similar methods, but relies on a fully trusted authority for authentication purposes.

M. Upmanyu et al. [18] introduce faster privacy preserving methods than with MPC for video surveillance, tracking and face detection (using Viola-Jones), but do not address face recognition or active adversaries.

Many previous papers also propose privacy-preserving video surveillance systems by either directly

masking peoples' faces and thus making face recognition impossible, or by introducing usage control mechanisms that only give an operator access to video streams in specific pre-defined situations [1].

2 Preliminaries

2.1 Multi-Party Computation

In reality, we generally always rely on a trusted party to perform computations on sensitive data from multiple parties. This allows the parties to compute the result while only revealing their data to the trusted party, instead of everyone. However, finding a party whom everyone trusts equally is often difficult and does not guarantee that the trusted party will not leak secret data in the future. Even if a trusted party is found, a single attack on this party could leak all private data. Multi-party computation attempts to solve this problem by eliminating the need for a trusted party.

In the late 1970s, the first protocols for the secure computation of functions were introduced. In 1979, Shamir et al. proposed a protocol for mental poker [14], a previously unanswered question to whether two people could play a fair game of poker without a deck of cards or involving any other person.

More general protocols followed that were designed to compute nearly any arbitrary function. The common objective is for parties $P_1, ..., P_n$, $n > 1$, to evaluate a defined function f on secret inputs x_i, $1 \leq i \leq n$, where only P_i knows the value of x_i. All parties wish to compute the result $z = f(x_1, ..., x_n)$ without revealing any information about their secret value to anyone.

2.1.1 Security Models

When performing computations that involve multiple parties, there is a common distinction between different security models. It is important to distinguish between the behavior of different parties, as it can have serious consequences on the security of the application. In section 3.3, we will show how choosing the wrong security model can render our proposed framework redundant.

The security requirements of the application directly impact the protocol choice as well as the overall performance and communication costs.

There are two aspects that need to be considered. On the one hand, we look at the worst potential behavior of a single party. Here, we distinguish between semi-honest and malicious adversaries. Protocols are respectively considered to be passively or actively secure. On the other hand, there is a big difference between settings where the majority is honest, or corrupted (dishonest).

Semi-honest Adversary (Passive Security)

Semi-honest adversaries follow the protocol honestly. This means that they truthfully provide data according to the protocol and do not manipulate data in order to gain information for which they have no authorization. However, such an adversary is still interested in gaining as much information as possible on secret data. Multiple semi-honest adversaries might therefore decide to collude

Malicious Adversary (Active Security)

Malicious adversaries deviate from the standard protocol. In practice this means that they can modify their data before sharing it with other parties which corrupts the output of the computation. While actively secure protocols in itself do not strictly prevent such adversaries from deviating from the protocol, they aim at detecting deviations with a high probability and can then prevent the adversary from gaining too much information. Protocols such as SPD$\mathbb{Z}_{2^k}$ often use so-called information-theoretic MACs to authenticate secret shares of all involved parties.

Honest Majority

As the name indicates, the majority of parties is honest in an honest majority setting. Unlike semi-honest adversaries, honest parties are not willing to share their own secret shares in order to gain

information on the secret input.

Dishonest Majority

A dishonest majority implies that the majority of parties is either semi-honest or malicious. Note that a two-party setting with one dishonest party is a dishonest majority scenario.

2.1.2 Oblivious Transfer

Oblivious transfer (OT) is an important concept which is used in the pre-processing phase of MPC protocols. The problem can be characterized as follows: consider parties Alice and Bob where Alice owns inputs x_0 and x_1. Bob wishes to query the value at index $b \in \{0, 1\}$ without letting Alice know which input he chose, and without Alice needing to reveal x_{1-b} so that Bob only learns x_b.

The OT protocol suggested by [6] involves three rounds of communication and RSA encryption to solve this problem:

Oblivious Transfer

1: Alice generates a key pair (e, n, d) and two random messages m_0 and m_1. Note that the public key is (e, n). She sends $((e, n), m_0, m_1)$ to Bob.
2: Bob chooses $b \in \{0, 1\}$ and generates a random message k. He computes $q = m_b + \text{Enc}_e(k)$ and sends q to Alice.
3: Alice computes $k'_i = \text{Dec}_d(q - m_i)$ for $i \in \{0, 1\}$, randomly chooses $s \in \{0, 1\}$ and transmits $(m'_0 = x_0 + k'_0, m'_1 = x_1 + k'_1)$.
4: As Bob knows k, b, s (and therefore also knows $k'_b = k$, but not k'_{1-b}) he can easily obtain $x_b = m'_b - k$.

2.1.3 SPD$\mathbb{Z}_{2^k}$

SPD$\mathbb{Z}_{2^k}$ [4] is a multi-party computation protocol that computes in rings modulo 2^k instead of fields and provides active security in a dishonest majority setting. Our decision for using this specific protocol in our framework is explained in more detail in section 3.1. We also use the efficient protocols from [5] that build on SPD$\mathbb{Z}_{2^k}$.

Notation

The scheme uses a parameter k determining the size of the ring 2^k and a security parameter s. Let N denote the number of parties, $\alpha \in \mathbb{Z}_{2^{k+s}}$ the global key and m the MAC. The congruence relation $x \equiv y \mod 2^k$ will also be written as $x \equiv_k y$.

If x is a secret, we can not perform direct operations on x itself, as this would reveal the actual value. Instead, the concept of *secret sharing* is used. The secret shares of x are denoted $[x] = (x^i, \alpha^i, m^i)_{i=1}^N$ where x^i, α^i, m^i correspond to the shares of x, α, m that P_i knows.

We furthermore have that

- $x' = \sum_{i=1}^N x^i \mod 2^{k+s}$ and $x \equiv_k x'$, with $x^i \in \mathbb{Z}_{2^{k+s}}$
- $\alpha = \sum_{i=1}^N \alpha^i \mod 2^{k+s}$, with $\alpha^i \in \mathbb{Z}_{2^s}$
- $m = \alpha \cdot x'$, i.e. $\sum_{i=1}^N m^i \equiv_{k+s} \left(\sum_{i=1}^N x^i\right) \cdot \left(\sum_{i=1}^N \alpha^i\right)$

Addition and Subtraction

Both addition and subtraction can be performed without any communication between the party. In order to compute $[z] = [x] + [y]$, each party simply computes the sum of its respective shares such as $z^i = x^i + y^i$.

Multiplying a secret share $[x]$ by a public constant α can be performed locally as well by the same logic. Each party i computes $z^i = \alpha \cdot x^i$.

Multiplication

Multiplication requires one round of communication and can be efficiently computed using multiplication triples. These can be generated by the parties in a preprocessing phase using oblivious transfer which is explained in depth in the SPD$\mathbb{Z}_{2^k}$ paper [4]. A triple $([a], [b], [c])$ satisfies the condition $c = a \cdot b$. Note that a triple is discarded after every multiplication.

Multiplication $z = x \cdot y$

1: Get new triple $([a], [b], [c])$
2: Compute $[\epsilon] = [x] - [a]$, $[\delta] = [y] - [b]$ and open ϵ, δ
3: Locally compute $[z] = [c] + \epsilon \cdot [b] + \delta \cdot [a] + \epsilon \cdot \delta$

We can easily verify the correctness:

$$\begin{aligned} z &= c + \epsilon \cdot b + \delta \cdot a + \epsilon \cdot \delta \\ &= a \cdot b + (x - a) \cdot b + (y - b) \cdot a + (x - a) \cdot (y - b) \\ &= x \cdot b + y \cdot a + x \cdot y - x \cdot b - y \cdot a \\ &= x \cdot y \end{aligned}$$

Comparison

The advantage of working over rings modulo 2^k is that we can more easily exploit properties of the most significant bit (MSB). Let Π_{MSB} be the protocol for securely extracting the most significant bit of a shared value $[a]$ (please refer to [5] for details). If we want to compare two shared values $[a]$ and $[b]$, $a < b$, the problem breaks down to checking whether $a - b < 0$. Using the two's complement representation of signed integers, a value is negative if the most significant bit is one, and positive otherwise. Thus $\Pi_{\text{LTZ}}([a]) := \Pi_{\text{MSB}}([a])$ where $\Pi_{\text{LTZ}}([a])$ evaluates $[a] < 0$.

Using these results, we have $\Pi_{\text{LT}}([a], [b]) = \Pi_{\text{LTZ}}([a] - [b])$ for $a, b \in [-2^{k-2}, 2^{k-2})$. To address the potential overflow incurred by $[a] - [b]$ when $a, b \in [-2^{k-1}, 2^{k-1})$, [5] adapts the protocol.

Comparison $u = a \overset{?}{<} b$

1: $[a_{k-1}] \leftarrow \Pi_{\text{MSB}}([a])$, $[b_{k-1}] \leftarrow \Pi_{\text{MSB}}([b])$
2: $[h] \leftarrow [a_{k-1}] + [b_{k-1}] - 2[a_{k-1}][b_{k-1}]$
3: $[e] \leftarrow \Pi_{\text{MSB}}([a] - [b])$
4: return $[d] \leftarrow [h][a_{k-1}] + [1 - h][e]$

Assume a and b have the same most significant bit. In that case, $[h] \leftarrow 2 \cdot [a_{k-1}] - 2 \cdot [a_{k-1}][a_{k-1}] = [0]$, and the protocol returns $[e]$. Furthermore, as a and b have the same sign, $a - b \in [2^{k-1}, 2^{k-1})$ returning $[e] \leftarrow \Pi_{\text{MSB}}([a] - [b])$ is safe by the arguments above.

If a and b have different signs, the result simply is most significant bit of a. The protocol returns the correct result in this case since we have $[h] \leftarrow [a_{k-1}] + [b_{k-1}] - 2[a_{k-1}][b_{k-1}] = [1] - 2 \cdot [0] = [1]$.

Information Theoretic MACs

Information theoretic MACs are designed to authenticate computations and force parties to follow the given protocol, as modifications to input data are detected with a high probability. In fact, [5] proves that a corrupted party can pass the MAC check with a probability of at most 2^{-s}.

Recall that every secret share $[x]$ consists of shares of the global key α and of the corresponding MAC m such that $m = \alpha \cdot x'$ where $x \equiv_k x'$, $x' = \sum_{i=1}^{N} x^i \mod 2^{k+s}$.

When we open x' and attempt to verify its correctness, we cannot reveal the global key α or the MAC m directly, as this would allow a corrupt party to modify his or her shares of the MAC in order to pass the MAC check. Instead, after opening x (which at this point may be corrupted), the parties each compute $z^i = m^i - x'\alpha^i$ locally, then commit to this value using a hash function and open it. If x has not been modified, it is easy to see that $\sum_i z^i = \sum_i m^i - x' \sum_i \alpha^i = m - \alpha x' = 0$.

Note that not committing to z^i using a hash function causes issues when some parties reveal their shares before others, allowing the others to modify their share of z^j with the additional knowledge they have. The last party N to reveal their share could in that case simply construct $z^N = 0 - \sum_{i=1}^{N-1} z^i$, allowing them to cheat arbitrarily.

Assume party k for fixed k decided to deviate from the protocol and reveal $x^k + \Delta$ instead of x^k. This results in the opening of $x' + \Delta$ instead of x' if all other parties are honest. Using information theoretic MACs however, we notice the following:

$$\begin{aligned}\sum_i z^i &= \sum_i m^i - (x' + \Delta) \sum_i \alpha^i \\ &= m - x'\alpha - \Delta\alpha\end{aligned}$$

If party k does not know α, there is thus a high probability that the MAC check will fail. However, if α were revealed to this party, it could simply modify $m^k \leftarrow m^k + \Delta\alpha$ to ensure the check passes.

Note also the additively homomorphic property of MACs. The authentication protocol for multiplication is a straightforward application of the three cases below. For secret shares $[x], [y]$ and a public constant c, we have:

$x + y$: $\forall i \in \{1, ..., N\}$, $(x^i, \alpha^i, m_x^i) + (y^i, \alpha^i, m_y^i) = (x^i + y^i, \alpha^i, m_x^i + m_y^i)$

$c \cdot x$: $\forall i \in \{1, ..., N\}$, $c \cdot (x^i, \alpha^i, m_x^i) = (c \cdot x^i, \alpha^i, c \cdot m_x^i)$

$x + c$: Fix $k \in \{1, ..., N\}$. Then party k computes $(x^k + c, \alpha^k, m_x^k + \alpha^k \cdot c)$, whereas parties $i \neq k$ compute $(x^i, \alpha^i, m_x^i + \alpha^i \cdot c)$

[5] also introduces a slightly adapted batch checking algorithm that can verify multiple values simultaneously.

2.2 Face Recognition

The face recognition task is a two step process. Given an image, we first need to identify (detect) the visible faces in the image and extract the cropped aligned faces. In the second matching stage, we try to compare two aligned faces for similarities to determine whether these people are the same or different.

2.2.1 Detection: MTCNN

MTCNN [23] uses a three staged process to identify bounding boxes and five facial landmarks, each stage involving the use of different convolutional neural networks. In the first stage, an image is given to a fully convolutional network called the Proposal Network in different sizes to obtain candidate windows. The second stage Refine Network takes these candidates and is trained to reject those that do not correspond to faces. Finally, the Output Network identifies facial regions and further refines the result.

This approach uses very shallow networks and has shown to be well suited for real-time applications, running at 99 FPS on an Nvidia Titan Black.

2.2.2 Matching: FaceNet

FaceNet [13] aims at building an embedding function f using a deep convolutional network which takes an image x of a face and projects it into the d-dimensional feature space $\mathbb{R}^d$. The network is trained so that all embeddings of the same face lie close together, whereas those of different faces are far apart. This is achieved by defining a loss function over image triplets, namely an anchor image, a different image corresponding to the same person, and an image corresponding to someone else. Given two aligned images of faces x and y, we say the faces match if the squared L_2 distance is below a threshold d, i.e. $\|f(x) - f(y)\|_2^2 < d$.

The model achieves an accuracy of 99.65% on the LFW dataset using the Inception ResNet v1 architecture [16] and VGGFace2 training dataset. This architecture builds on two concepts, which we will only discuss briefly here.

Inception Networks. Inception networks, as first described in [15], use convolutions that combine filters of different sizes in parallel. This improves the quality of the predictions by maintaining different levels of detail in a single layer and improves the efficiency of the network.

Residual Learning. [7] noticed an increase in training and test error when using deeper networks. According to the paper, the degradation problem is assumed to be caused by non-linear layers having difficulties approximating identity mappings. Traditionally, the next layer only takes the output of the previous layer as input, however He et al. use shortcut connections which combine the output of a traditional layer with an identity mapping.

2.2.3 Other Detection and Matching Methods

Detection. Viola-Jones [19] is a fast object-detection framework from 2001 which uses Haar-Like features to train cascaded classifiers. With AdaBoost, they find the best features from a set of 180,000 predefined different rectangular features. However, this method is rather susceptible to variations as pointed out by [21] and [23].

Matching. The Eigenface method [17], as used in many related works, is one of the oldest face recognition methods and projects faces into a "face space" whose basis is a set of eigenfaces. These are the result of applying the Principal Component Analysis (PCA) on the training set. However, this method can be sensitive to variations in lighting and orientation, as also noted by [3].

3 Application to Video Surveillance

3.1 General Framework

Idea and Involved Parties

The proposed framework (as shown in figure 1) relies on a subtle reinterpretation of the purpose of multi-party computation. We use a two-party setting, not only to hide the original input by performing secure distributed computations, but more importantly to introduce a system of checks and balances. Either party is able to verify that the other is not trying to gain more information than they are supposed to, because both parties need each other for the face recognition task. In other words, if party P_0 aims at tracking a set of faces F_0, and party P_1 a different set F_1, then only the intersection $F_0 \cap F_1$ can be identified and tracked. If parties P_0 and P_1 are carefully chosen, this is a very powerful property.

We will denote the set of faces that both parties agree to track by $C_{\text{DB}} = F_1 \cap F_2$. In the following, we may also refer to it as the *criminal database* as it is assumed that C_{DB} will only contain faces of criminals. However, we refrain from providing a more specific definition as this is subject to many factors and the chosen parties.

As the framework relies on the idea that both parties agree and verify that only the people in C_{DB} should be tracked, it follows that each party has access to C_{DB}. To guarantee the integrity of C_{DB}, we

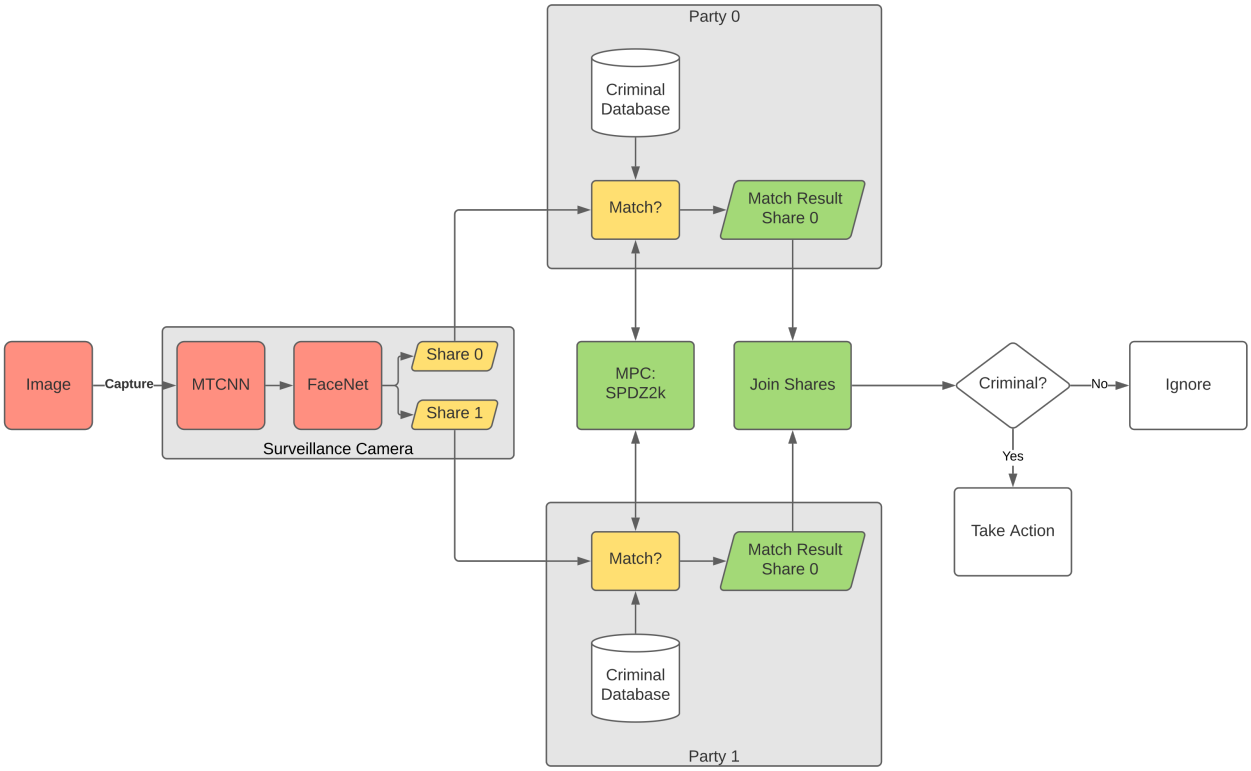


Figure 1: High-level process for privately performing face recognition. Red blocks contain identifiable information. Yellow blocks are sensitive as well, but reveal no information about the original data unless the parties collaborate. Green blocks are safe to share.

assume that each party owns a separate copy, C^0_{DB} and C^1_{DB} respectively. A well designed and cryptographically secure shared database would work as well though, instead of using separate copies.

Database Change Requests

Modifications to the criminal database can be done using a simple approval process. If party P_i, $i \in \{0, 1\}$ wants to add a criminal c to C^i_{DB}, it sends a request with information about c including its d-dimensional embedding from FaceNet to P_{1-i}. If P_{1-i} approves the modification and can verify the correctness of the embedding using images of c (but at least one), it adds the embedding to its database C^{1-i}_{DB} and notifies P_i. P_i can now add the embedding to its own database. At this point, we should have $C^0_{\text{DB}} = C^1_{\text{DB}}$.

In section 3.3 we discuss the potential scenario where P_i modifies his copy C^i_{DB} without the approval of P_{1-i}, and explain why this would be detected.

Surveillance Camera

For the following, we will assume that the surveillance cameras including their attached computing hardware are trusted environments and that security measures have been taken to prevent physical and network attacks. Nonetheless, it is important to keep in mind that the camera has access to the plaintext surveillance footage for short periods of time (since it captures the images) and is therefore not only the easiest point of attack, but also the most interesting.

The important stages of the face detection and recognition process occur on the surveillance camera itself. We run the MTCNN face detection algorithm on the original image captured by the camera, and the aligned faces are then passed through an Inception ResNet network (FaceNet) to project the faces into $\mathbb{R}^d$. The real-valued vectors are then converted to fixed-point numbers to simplify the conversion to integers, as required for MPC. The camera creates secret shares of each of the d-dimensional vectors and sends the shares to the respective parties over secure channels.

As none of these models add any new information to the plain image, there are no privacy concerns to this approach. Anyone who had access to the image would be able to extract the same information as the camera. Note again that FaceNet is trained to extract important features of faces, and **not** to recognize specific people.

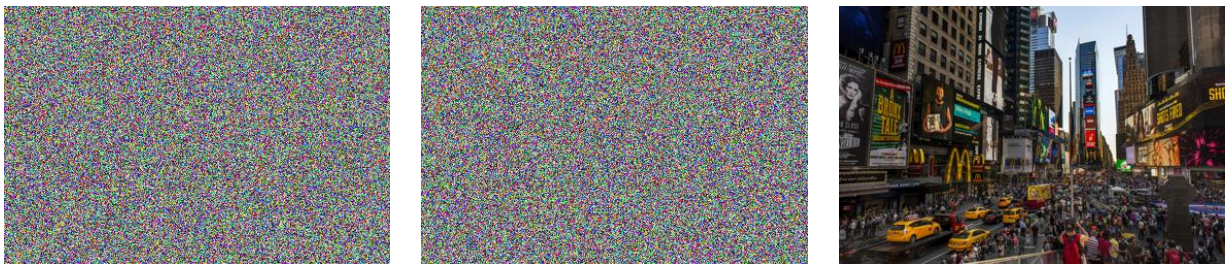
Furthermore, the surveillance footage itself can be secret shared. This is in particular useful when a severe crime has been committed and both parties have an interest in reconstructing the footage. Just like the shares of the embeddings for face recognition, the shares of the video footage are independent and identically distributed and therefore reveal no information about the original footage. Section 3.2 discusses this in more detail.

This approach requires every single surveillance camera to have its own GPU, but we believe that this is considerably more reasonable than making a forward pass through a deep neural network using actively secure multi-party computation when dealing with real-time constraints. When we refer to a *surveillance camera* in the future, we therefore refer to the camera including its computing hardware and networking capabilities.

Identifying Faces

Let $x \in \mathbb{Z}^d_{2^k} \mod 2^k$ be the projected embedding of a face from FaceNet. We denote the i^{th} index of x as x_i, such that $x'_i = \sum_{j=0}^{1} x_i^j \mod 2^{k+s}$ and $x_i \equiv_k x'_i$ where x_i^j corresponds to the share of x_i held by party j. In order to check if $x \in C_{\text{DB}}$, we securely compute for every $c \in C_{\text{DB}}$

$$[c_{\text{match}}] = \left(\sum_{i=1}^{d} ([x_i] - c)^2 \right) \overset{?}{<} t.$$



$x^0 = x - r \mod 256$ $x^1 = r$ $x^0 + x^1 \mod 256 = x$

Figure 2: The first two images correspond to the shares of parties P_0 and P_1, whereas the last image shows the reconstructed result if **both** P_0 and P_1 agree to reveal their shares. Photo: Brittany Petronella (`https://www.nycgo.com/images/venues/152/timesquare_brittanypetronella_0069.jpg`)

$[c_{\text{match}}]$ is then opened to reveal $c_{\text{match}} \in \{0, 1\}$. If $c_{\text{match}} = 1$, we can say with high certainty (given the accuracy of FaceNet) that x is the image of a criminal in the database, and further measures can be taken.

It is not sufficient to merely compute the distance securely, and then check whether the (then public) distance is below the threshold, as it does not offer sufficient privacy. If we know that a face $x \in \mathbb{Z}_{2^k}^d$ is distance a_c apart from c for all $c \in C_{\text{DB}}$, we can construct a d-sphere around each $c \in C_{\text{DB}}$ with radius a_c respectively. As the intersection of two n-spheres is a $(n-1)$-sphere, we only need a finite number of distances of x to criminals to be able to reconstruct x. Even if we were not able to uniquely reconstruct x due to insufficient data points, we might be able to gain sensitive information about the subspace that we can limit x to. This might leak information about ethnicity or gender.

3.2 Secret-Sharing the Surveillance Footage

Every frame of the video feed can be masked with a cryptographically secure random array of integers. To ensure perfect privacy of the image, every color value of each pixel needs to be masked with a separate random value.

$\Pi_{\textbf{imgshare}}(x)$ Generate secret shares of an image x of dimensions $w \times h$

1: Uniformly sample $r_i \leftarrow (\{0, ..., 255\})_{j=1}^3$ for every RGB pixel $i \in \{1, ..., w \cdot h\}$ using a cryptographically secure pseudorandom number generator. We set the array r to have indices r_i.
2: Send $x - r \mod 256$ to P_0 and r to P_1

The secret shares obtained by the mask reveal no information about the original image. Given some $x \in \mathbb{Z}_{256}$ and a uniformly sampled $r \leftarrow \mathbb{Z}_{256}$, we note that $x + r$ is uniformly distributed by properties of the ring. We can therefore conclude that $x + r$ reveals no information about x if r is unknown. Protocol describes the procedure, while figure 3.2 gives an idea of what is visible to the individual parties.

Reconstructing the image from two shares x^0 and x^1 from parties P_0 and P_1 respectively is simply the sum $x^0 + x^1 \mod 256$. Any party can deliberately add a cryptographically secure random mask to parts of its share if it wishes to hide certain areas of the image during reconstruction.

3.3 Addressing Corrupted Parties

In our implementation we use an actively secure multi-party computation protocol, meaning it is secure against corrupted parties. While using an actively secure protocol introduces a non-negligible additional overhead in computation and communication in the pre-processing phase, we show the severe potential consequences of **not** being actively secure.

Let $c \in C_{\text{DB}}$. Assume party P_i wishes to track any $p \in \mathbb{Z}_{2^k}^d - C_{\text{DB}}$ without the approval of P_{1-i}. It can easily compute $\Delta = p - c \in \mathbb{Z}_{2^k}^d$ and set its share x^i of a face x to be $x^i_{\text{corrupt}} \leftarrow x^i - \Delta$, so that $x^i_{\text{corrupt}} - c = x^i - (c + \Delta) = x^i - p$. While P_{1-i} believes that $[y] = [x] - c$ is being computed, we have that $[y] = [x] - p$, since

$$\sum_i y^i = x^i_{\text{corrupt}} + x^{1-i} - c = x^i + x^{1-i} - (c + \Delta) = x - p$$

Resuming the computations would result in party P_i tracking p instead of c. P_{1-i} helps the party to do so, however unknowingly. Using information theoretic MACs to authenticate the computations as illustrated in the preliminaries prevents this issue and uncovers corruptions with high probability.

3.4 Obfuscating Movement Profiles

When a party receives secret shares of two different face embeddings, it is impossible for this party to determine whether these shares belong to the same face or different ones, so at first sight it seems impossible for a party to privately track individuals who are not in C_{DB}.

However, if only one person is visible in the camera frame, then all secret shares can be tied to this one person (without knowing who this person is). If this person now moves out of the visible range of one surveillance camera and moves into another, all parties now know that the former camera did not identify any people, whereas one person is visible in the other frame. Both parties can therefore deduce that some (unknown) person moved from one location to another.

On a larger scale, it can be possible to create movement profiles of individuals when only a few people are visible (at night, or in areas with a low population density). These movement profiles are unique for every person as they can reveal information about where we live, where we buy our groceries, which bars and restaurants we frequent and which friends we visit. With sufficient information about certain individuals, it may be possible to deduce the true identity behind a given motion path.

We illustrate the problem with a small example in a strongly simplified model using discrete time steps. For each camera $i \in \mathbb{Z}$, we denote by v_i the number of people visible to camera i. We assume that a person p_j with location $l_j \in \mathbb{R}$ is only visible by exactly one camera at a time, i.e. p_j is visible to camera $\lfloor l_j \rfloor$.

At each time step, a person visible to camera i can either remain visible to camera i, move left and become visible to camera $(i-1)$ or move right and become visible to camera $(i+1)$. v_i updates accordingly. We informally model the movements of a person in motion with the following probabilities:

- $\mathbb{P}(\text{"repeat last movement"} \mid \text{"moving"}) = 0.85$
- $\mathbb{P}(\text{"stop"} \mid \text{"moving"}) = 0.1$
- $\mathbb{P}(\text{"change direction"} \mid \text{"moving"}) = 0.05$

whereas a person standing still is assumed to behave as follows:

- $\mathbb{P}(\text{"remain standing still"} \mid \text{"standing"}) = 0.8$
- $\mathbb{P}(\text{"move left"} \mid \text{"standing"}) = 0.1$
- $\mathbb{P}(\text{"move right"} \mid \text{"standing"}) = 0.1$

When no information about the previous motion is given (i.e. at $t = 0$), we assume the person either moves left, stays or moves right with equal probability.

At time $t = 0$, each party can arbitrarily associate a unique id to each observation j, namely $\text{id}_{j,0} \in \mathbb{N}$. We give the actual person p_j behind this observation (whose true identity remains unknown to the parties) the id $\text{id}_j = \text{id}_{j,0}$. Clearly, at time $t = 0$ we have $|\text{id}_{j,0} - \text{id}_j| = 0$.

Now, at every time step t, the parties may arbitrarily associate $\mathrm{id}_{j,t} \in \mathbb{N}$ to each observation j. We consider the worst case scenario where a party chooses $\arg\max_{\mathrm{id}_{j,t}} \left(\mathbb{P}[|\mathrm{id}_{j,t} - \mathrm{id}_j| = 0 \big| \mathrm{id}_{j,0}, ..., \mathrm{id}_{j,t-1}]\right)$.

In figure 3, we illustrate the problem in a simple 5-camera setup with a total of two people visible across all cameras. Each box $i \in \{0, ..., 4\}$ represents a camera, and the number in the box corresponds to the number of people v_i visible to camera i. Blank boxes indicate that no person was detected. The relative positioning of the boxes represents the actual field of vision and location of the cameras. We notice how we can uniquely reconstruct two motion paths until (and including) time step $t = 2$, given our observations. At $t = 3$, there are two possible scenarios - either "orange" moved one further to the right, or "orange" stopped. However, the former is considerably more likely. In fact, "orange" continues moving to the right with probability $p = 0.8$, but combined with our other observations, we can compute that "orange" moved to the right with probability

$$\begin{aligned} p &= \frac{\mathbb{P}(\text{"orange moves right"}) \cdot \mathbb{P}(\text{"blue remains still"})}{\mathbb{P}(\text{"orange moves right"}) \cdot \mathbb{P}(\text{"blue remains still"}) + \mathbb{P}(\text{"orange stops"}) \cdot \mathbb{P}(\text{"blue moves right"})} \\ &= \frac{0.8 \cdot 0.85}{0.8 \cdot 0.85 + 0.1 \cdot 0.1} \\ &= 0.986. \end{aligned}$$

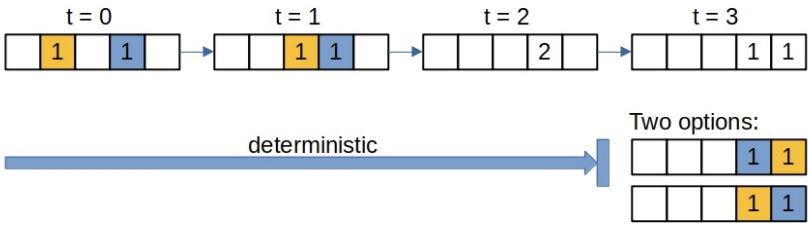


Figure 3: Attempting to reconstruct movement profiles by assigning colors to observed people

Obfuscation

As the complexity of finding a solution to this problem increases with the number of people visible to the camera, we simply adapt the surveillance camera to detect "dummy faces". This comes at negligible additional cost, since the surveillance camera can simply choose a d-dimensional vector with constant values. Given an obfuscation parameter $q \geq 0$, the camera i then chooses to create $\max(0, q - v_i)$ dummy embeddings. Instead of observing v_i people, the parties now observe $\max(q, v_i)$ people. The parties will not be able to distinguish a share of a dummy embedding from a share of an actual embedding, as the values are uniformly distributed over the ring $\mathbb{Z}_{2^k}$.

If we take the exact same example from before to include dummy observations with $q = 1$, it becomes considerably harder to reconstruct the motion path of an arbitrary person visible to camera 2 at $t = 0$. To accurately compute the possibility of each path occurring, we would additionally need to estimate the probabilities for all $0 \leq v_i \leq q$. The possible scenarios are shown in figure 4, where the orange marker corresponds to the location of one of the two observed people at $t = 0$.

The first piece of useful information that we get is at $t = 2$, where camera 4 observes two people with absolute certainty. Reconstructing the motion path is more difficult since the dummy faces hide many potential motion dynamics that could help either of the parties understand the observations.

With these findings, it remains an open problem to find the optimal obfuscation parameter q, such that given privacy parameters T and α, as well as a probability measure $\mathbb{P}$,

$$\mathbb{P}\left[\sum_{t=0}^{T-1} |\mathrm{id}_{j,t} - \mathrm{id}_j| = 0\right] < \alpha$$

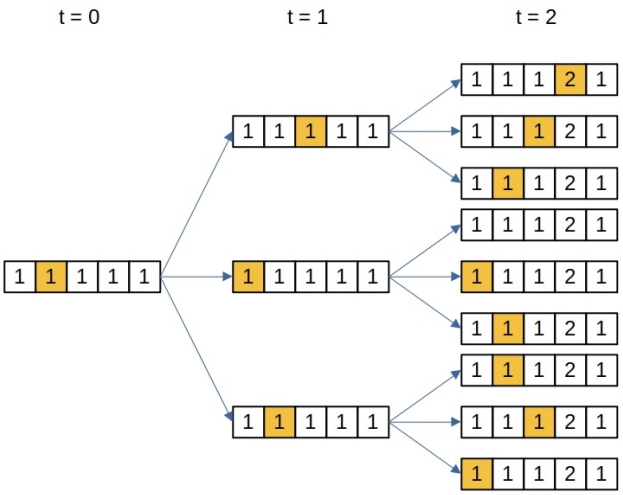


Figure 4: Attempting to reconstruct movement profiles from observations and "dummy faces"

meaning that the probability of being able to correctly reconstruct the motion path of a specific person after T observations should be less than α.

3.5 Issues and Limitations

The proposed framework relies on the fact that parties P_0 and P_1 do not trust each other and have no interest in exchanging secrets. This is of course an idealized assumption which does not always hold in reality, whether it is caused by corruption, change in legislature or simply social engineering.

As already briefly pointed out, one of the weakest points of the system are the surveillance cameras. While the parties P_0 and P_1 most likely data centers with strict security measures, the surveillance cameras are in public spaces, more or less easily accessible. Attacks on the camera could render the entire framework redundant, as it could allow an attacker to obtain direct access to the video feed. Powerful players such as P_0, P_1 or other organizations could have the resources to do so. On the other hand, given that this requires every camera to be compromised individually, we consider the risk of a large-scale attack to be rather low. It would be worth investigating the use of trusted execution environments, such as Intels SGX.

There is not only the security concern of the surveillance cameras, but also the cost aspect. Unlike conventional CCTV cameras, our proposed framework requires each camera to have a good GPU to detect faces and then project them into a Euclidean space. This will likely be one of the main cost drivers of the surveillance cameras.

The privacy preserving obfuscation method described in the previous section also comes at a price. We are effectively performing computations on "empty" faces, thus increasing energy consumption and bandwidth usage. The advantages and costs of this method still need to be explored in more detail.

4 Benchmarks

All benchmarks have been conducted using the MP-SPDZ library's [10] implementation of SPD$\mathbb{Z}_{2^k}$ on an AWS c5a.8xlarge instance. We use MP-SPDZ as it provides a simple interface to quickly change the protocol.

We focus on a database with a fixed size of 150 entries. This is sufficient for a city of the size of Frankfurt to match faces against a database of suspects for murder and sexual abuse against children [8][1].

[1]Disclaimer: We only provide this information to give an idea of the order or magnitude. Under no circumstance

FaceNet generates 128-dimensional embeddings by default, which are the ones we primarily use in our benchmark tests. We furthermore attempted to train a model with 32-dimensional embeddings on the VGGFace2 dataset [2] with approximately 3.3 million faces, however the training crashed after only 24 hours. At the point where the training was interrupted, we had reached a validation accuracy of approximately 70%, far from the 99.65% achieved by the default model. However, by reducing the size of the vector, we also reduce the time of the online phase from $0.363s$ to $0.105s$ on one thread.

The online phase is very efficient as the SPD$\mathbb{Z}_{2^k}$ protocol allows us to precompute the expensive triplets and authentication shares during an offline phase. As seen in the benchmark tests, we can match approximately 5 or 10 people against a criminal database every second using only two or four threads. As the task is highly parallelizable, an entire city could achieve real-time processing in the online phase with a sufficient number of threads. With the introduction of more efficient integer arithmetics on GPUs [20], it is reasonable to assume that surveillance systems involving only a handful of cameras can already be implemented today if we have generated sufficient preprocessing data in the offline phase.

n_{threads}	Online Phase		Total	
	Time (s)	Global Data Sent (MB)	Time (s)	Global Data Sent (MB)
1	0.363	4.229	571.51	104,551
2	0.179	4.230	279.93	104,634
4	0.102	4.231	143.26	104,805
8	0.066	4.232	81.36	105,146
16	0.096	4.235	64.64	105,817

The total cost combines the cost of the online and the offline phase. While the online phase is efficient, we immediately see that the offline phase is a tremendous bottleneck, making real-time applications difficult, if not impossible, even when running the offline phase over night when only a few people are outside. However, if sufficient data for the online phase is generated in advance for a predefined time period in the offline phase, the framework can process video material in real-time for this time period. Using a semi-honest model on the other hand would greatly decrease the offline cost, but as described in section 3.3, this could have severe consequences for the privacy of the framework. The experimental results confirm our approach to performing as many computations as possible on the surveillance camera itself.

5 Outlook

In the next step, we need to investigate the offline phase in more detail and determine if we can improve it by tailoring it to our application. Alternative solutions, such as using trusted crypto providers to generate multiplication triples and possibly even information theoretic MACs could yield promising results.

While this framework is initially designed for privacy preserving face recognition, the possibilities do not end there. Just as we performed face recognition on secret shares of vectors, it is also possible to feed secret-shared images through entire pre-trained neural networks without revealing the original image. This opens many interesting applications, such as training models to detect

1. (camera) vandalism

should this be considered legal advice, and we certainly do not intend to discriminate against victims of other severe crimes by our selection.

2. violence
3. car crashes (to immediately be able to call an ambulance)

We did not explore these in more detail due to a lack of training data, but we would like to point them out nonetheless. Just as with our proposed framework for face recognition, it is possible to implement such models while guaranteeing that they will only reveal the information that they were specifically designed to detect, nothing else.

In a next step, we need to consider post-quantum cryptography. This includes replacing the standard asymmetric cryptography, whose security is threatened by quantum computers, with PQC-secure methods, such as lattice based approaches. For all other operations, we need to understand how we might need to adapt the security parameters k and s to maintain the same level of security.

6 Conclusion

The framework as such appears to be promising, as it can - as we have shown in theory - provide people with perfect privacy while allowing law enforcement agencies to track down those who have committed severe crimes. Our obfuscation method aims at adding an additional level of privacy by hiding even the most subtle details which, as we have shown, can leak sensitive information. While our suggested implementation may not be feasible yet for face recognition in public spaces over unlimited time-periods, we have successfully shown that it works over short pre-defined periods.

7 Acknowledgments

First and foremost I am extremely grateful to my supervisor Matthieu Rambaud at Télécom Paris for his continued excitement, insightful suggestions and support by pointing me to interesting papers, even when it was long past midnight. I would like to thank my professor and referent instructor Emmanuel Haucourt, who guided me through the formalities of the bachelor thesis, and professors Benjamin Smith and Daniel Augot for helping me find a suited supervisor for my idea. I also appreciate all the patient and friendly support I received on MP-SPDZ from Marcel Keller on Github. Lastly, I extend my sincere thanks to Jörn-Marc Schmidt who first introduced me to MPC and FHE, and without whom I never would have written my thesis on secure computations.

8 About the Research Environment

This thesis was written as part of an individual project during a two-month internship at Télécom Paris in France, under the supervision of Matthieu Rambaud. Due to SARS-CoV-2, most of the internship was conducted remotely, but the progress in person was regularly discussed in person when possible.

References

[1] Pascal Birnstill and Alexander Pretschner. "Enforcing privacy through usage-controlled video surveillance". In: Aug. 2013, pp. 318–323. DOI: 10.1109/AVSS.2013.6636659.

[2] Qiong Cao et al. *VGGFace2: A dataset for recognising faces across pose and age*. 2018. arXiv: 1710.08092 [cs.CV].

[3] E. Carniani et al. *Enhancing Video Surveillance with Usage Control and Privacy-Preserving Solutions*. Tech. rep. Istituto di Informatica e Telematica, Consiglio Nazionale delle Ricerche & Scuola Superiore Sant'Anna, NoES-TeCIP.

[4] Ronald Cramer et al. *SPDZ2k: EfficientSPD $\mathbb{Z}_{2^k}$ MPC mod 2^k forDishonestMajority*. Cryptology ePrint Archive, Report 2018/482. https://eprint.iacr.org/2018/482. 2018.

[5] Ivan Damgård et al. *New Primitives for Actively-Secure MPC over Rings with Applications to Private Machine Learning*. Cryptology ePrint Archive, Report 2019/599. https://eprint.iacr.org/2019/599. 2019.

[6] Shimon Even, Oded Goldreich, and Abraham Lempel. "A Randomized Protocol for Signing Contracts". In: *Commun. ACM* 28.6 (June 1985), pp. 637–647. ISSN: 0001-0782. DOI: 10.1145/3812.3818. URL: https://doi.org/10.1145/3812.3818.

[7] Kaiming He et al. *Deep Residual Learning for Image Recognition*. 2015. arXiv: 1512.03385 [cs.CV].

[8] Polizei Hessen. *Frankfurt am Main: Polizeiliche Kriminalstatistik 2019, Aufgliederung Tatverdächtige gesamt. Police crime statistics 2019, breakdown of suspects in total*. URL: https://k.polizei.hessen.de/753315062. (accessed: 22.11.2020).

[9] Serge F. Kovaleski Kate Conger Richard Fausset. *San Francisco Bans Facial Recognition Technology*. URL: https://www.nytimes.com/2019/05/14/us/facial-recognition-ban-san-francisco.html. (accessed: 10.11.2020).

[10] Marcel Keller. *MP-SPDZ: A Versatile Framework for Multi-Party Computation*. Cryptology ePrint Archive, Report 2020/521. https://eprint.iacr.org/2020/521. 2020.

[11] M. Osadchy et al. "SCiFI - A System for Secure Face Identification". In: *2010 IEEE Symposium on Security and Privacy*. 2010, pp. 239–254. DOI: 10.1109/SP.2010.39.

[12] Ahmad-Reza Sadeghi, Thomas Schneider, and Immo Wehrenberg. *Efficient Privacy-Preserving Face Recognition*. Cryptology ePrint Archive, Report 2009/507. https://eprint.iacr.org/2009/507. 2009.

[13] Florian Schroff, Dmitry Kalenichenko, and James Philbin. "FaceNet: A unified embedding for face recognition and clustering". In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2015). DOI: 10.1109/cvpr.2015.7298682. URL: http://dx.doi.org/10.1109/CVPR.2015.7298682.

[14] Adi Shamir, Ronald L. Rivest, and Leonard M. Adleman. "Mental Poker". In: *The Mathematical Gardner*. Ed. by David A. Klarner. Boston, MA: Springer US, 1981, pp. 37–43. ISBN: 978-1-4684-6686-7. DOI: 10.1007/978-1-4684-6686-7_5. URL: https://doi.org/10.1007/978-1-4684-6686-7_5.

[15] Christian Szegedy et al. *Going Deeper with Convolutions*. 2014. arXiv: 1409.4842 [cs.CV].

[16] Christian Szegedy et al. *Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning*. 2016. arXiv: 1602.07261 [cs.CV].

[17] M. A. Turk and A. P. Pentland. "Face recognition using eigenfaces". In: *Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. 1991, pp. 586–591. DOI: 10.1109/CVPR.1991.139758.

[18] M. Upmanyu et al. "Efficient privacy preserving video surveillance". In: *2009 IEEE 12th International Conference on Computer Vision*. 2009, pp. 1639–1646. DOI: 10.1109/ICCV.2009.5459370.

[19] Paul Viola and Michael Jones. "Rapid object detection using a boosted cascade of simple features". In: *Comput. Vis. Pattern Recog* 1 (Jan. 2001).

[20] Takeshi Yamanouchi. *Chapter 36. AES Encryption and Decryption on the GPU*. URL: https://developer.nvidia.com/gpugems/gpugems3/part-vi-gpu-computing/chapter-36-aes-encryption-and-decryption-gpu. (accessed: 28.11.2020).

[21] Bin Yang et al. *Aggregate channel features for multi-view face detection*. 2014. arXiv: 1407.4023 [cs.CV].

[22] X. Yang et al. "Efficient and Privacy-Preserving Online Face Recognition Over Encrypted Outsourced Data". In: *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. 2018, pp. 366–373. DOI: 10.1109/Cybermatics_2018.2018.00089.

[23] K. Zhang et al. "Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks". In: *IEEE Signal Processing Letters* 23.10 (Oct. 2016), pp. 1499–1503. ISSN: 1070-9908. DOI: 10.1109/LSP.2016.2603342.