Multimedia Tools and Applications
https://doi.org/10.1007/s11042-021-10533-8

Just look at to open it up:

A biometric verification facility for password autofill to protect electronic documents

Maciej Smiatacz[1] · **Bogdan Wiszniewski**[1]

Received: 28 February 2020 / Revised: 11 October 2020 / Accepted: 13 January 2021 /
Published online: 05 March 2021

Abstract
Electronic documents constitute specific units of information, and protecting them against unauthorized access is a challenging task. This is because a password protected document may be stolen from its host computer or intercepted while on transfer and exposed to unlimited offline attacks. The key issue is, therefore, making document passwords hard to crack. We propose to augment a common text password authentication interface to encrypted documents with a biometric facial identity verification providing highly personalized security mechanism based on pseudo-identities. In consequence the encrypted document can be unlocked with the legitimate user's face, while for everyone else stays encrypted with a hard to crack text password. This paper makes two contributions: (1) The proposed scheme enables password autofill without referring to any external service, which significantly limits the possibilities of an attack by adversaries when opening, reading and editing the protected document, (2) By the adoption of biometric verification techniques enabling fine-tuning of false acceptance and false rejection rates, it provides for responsible adaptation to users.

Keywords Document encryption · Automatic password generation · Biometric identity verification · Reactive documents · Natural user interface

1 Introduction

Internet enables individuals to link across boundaries to work together for a common purpose by sharing and exchanging electronic documents of any format and with an arbitrary

This work was supported in part by the National Science Center in Poland under grants DEC1-2011/01/B/ST6/06500 and N N516 367936

✉ Bogdan Wiszniewski
bogwiszn@pg.edu.pl

Maciej Smiatacz
macsmiat@pg.edu.pl

[1] Department of Intelligent Interactive Systems, Faculty of Electronics, Telecommunications and Informatics, Gdansk University of Technology, Pomorskie, Poland

content. When needed, the content may be encrypted with a text password or an equivalent identity verification mechanism. As long as the document stays in its home directory this mechanism may be reinforced by its host server, which is capable of successfully repelling online attacks by limiting the acceptable number of guesses. However, when being transferred over the net, either uploaded/downloaded by users or sent as an email attachment, the document has no external protection from any third party intercepting it and is on its own against far more powerful offline attacks. In such a case, the password protecting it should be sufficiently strong to withstand an unbounded number of possible guesses. A serious challenge addressed in this paper is to enable users to effortlessly encrypt and decrypt documents with such strong passwords without the need memorize them.

1.1 The document password challenge

A password protecting document content must be strong enough to withstand offline attacks for the entire life time of the document. This period may extend over several hours as well as several months; as long as the business process is not completed, the protected content of its documents may have value for potential attackers. Later, upon completion, the documents may be destroyed or archived on a well (online) protected server. Only then would their passwords no longer be needed.

The question is what the lower bound of the number of guesses may be assumed to be secure for offline attacks, as hardware and cracking methods constantly improve. The estimate given in [13] of 10^{14} guesses is probably far too low today, but on the other hand the potential of 32-character passwords composed of symbols from the ASCII 95 printable character set still seems inexhaustible in this respect. The standard Shannon's metric for the amount of information in a string of $L = 32$ characters selected at random from the set of $N = 95$ uncorrelated symbols estimates the number of bits of entropy as $H_{sh} = log_2 N^L = L \cdot log_2 N = 32 \cdot log_2 95 \approx 210$. In other words, when using the brute-force approach the required number of guesses could be as high as $2^{210} \approx 1,9 \cdot 10^{63}$. Passwords of this kind are supported today by many document writer/authoring tools, including MS Word, Libre Office and Acrobat Writer.

Although strong, such long and complex passwords are not only impossible to remember but also often difficult to type correctly, making them useless when unique passwords are needed for every document that the user may want to access.

A popular solution to this problem with regard to multiple accounts could be the automatic password generator, capable of generating arbitrarily complex random strings of symbols [7, 21, 32, 49]. It may be a part of a more general generative password manager [29] that can also take care of storing generated passwords in some encrypted database or reproducing them on demand to access the password protected local or remote entity. However, this common register-first-login-next pattern that works in handling multiple user accounts will not work for password encrypted documents exchanged in a business process, as the document's originator and recipient usually are not the same person and operate different personal devices. Once generated on one device, the password (or a secret used to generate it) should be securely transferred to another device. It may be particularly troublesome if encrypted documents are exchanged as email attachments instead of being stored in a shared repository protected online. Moreover, business processes may involve many collaborators exchanging multiple documents many times. Passwords used to encrypt these documents can often be of a one-time use and their number may vary in respect to the current dynamic context of communication between collaborators, e.g. when many originators send different

documents to one recipient or when just one originator sends multiple copies of a document to many recipients. Keeping track of all the passwords used in this process by each participant can be mentally difficult, error prone and annoying.

Another solution to mitigate the problem of memorizing strong passwords by human users could be biometric authentication. It relies on performing measurements of the subject to decide if the thus obtained data stay within the limits of his/her unique *identity template*, or biometric reference data, regarding his/her fingerprint, voice or face. Although biometric authentication serves essentially the same purpose as the password based authentication, it may not be 100% accurate always. This is because it involves a non-zero Hamming distance decision threshold to accept the input (biometric) representation of the user, whereas the password based identity verification can accept only the exact submitted password string. One serious problem is that if permanent (not resettable) biometric reference data are stolen the attacker can use them to access other biometrically protected documents destined for the same user. Moreover, a widespread use of biometric verification for protecting documents exchanged in office practice could be limited by the fact that popular document writing and authoring tools can offer so far only the less convenient text based password protection.

1.2 A reactive document solution

The challenge of using strong text passwords to make documents resistant to offline attacks is addressed further in this paper by the solution layering passwordless biometric identity verification atop automatic password generation. In that way, the protected document content can be encrypted with a strong password, whereas the document as a whole is a reactive entity that can automatically generate the required password only if the user in front of the supporting personal device camera is the legitimate person. We call the proposed solution scheme *For Your Eyes Only (FYEO)*; its contribution can be summarized in the following points:

1. Biometric verification is based on a biometric identity template of the legitimate user. The template is irreversible, so no meaningful biometric data of the legitimate user can be reconstructed by an impostor to deceive the document to open up, and the password cannot be generated unless the biometric verification of the recipient is positive.
2. Arbitrary strong passwords involving all printable characters can be generated using the biometric identity template as a source of entropy. Any customary document writing/reading tool that supports text-based password protection of the document content can be incorporated into the scheme.
3. The biometric identity template is delivered directly to the recipient's device with the password encrypted content, so the document can be decrypted offline, without the support of any external authorization server.
4. The algorithm for generating a password is a secret that would be much more difficult to reconstruct by an impostor than to guess the otherwise strong password (if at all possible, due to the potential infinite number of algorithms to try).

In Section 2 the overall structure of the FYEO scheme is outlined and various threat scenarios are analyzed. Given several unique properties of the identity template it is argued in Section 3 that sending the template along with the encrypted document in one bundle is safe. In Section 4 the FYEO scheme is validated using a generic cost-effective feature based identity verification mechanism and a few common sense text password generation methods. Section 5 provides a short survey of current research and technologies for biometric

protection of digital content that the scheme proposed in the paper draws on, namely automatic password generation tools, entropy sources and face unlocking systems. Section 6 concludes the paper by outlining implementation of the FYEO scheme in two experimental document exchange platforms developed by authors.

2 The FYEO scheme

The process of encrypting and decrypting a document to be exchanged in this scheme requires its sender and recipient to perform several simple steps described below. They all involve a facial identity template (FIT) file generated separately for each prospective document recipient on a dedicated server. The server is isolated from the Internet to prevent attackers from intercepting photos of the subject that might be used later to deceive the encrypted document to open up. As mentioned before, the FIT file content is not reversible, i.e. once disposed on the server the original photos cannot be reconstructed from it. However, it suffices to verify the identity of the legitimate user and provides a source of entropy for generating a strong text password.

2.1 FYEO processes

Prior to sending any decrypted document to its user, each prospective recipient has to register at the FYEO enrollment point, shown in Fig. 1. A series of face shots is taken, with the exact number depending on the actual quality assessed by the *image to biometric data encoder*. The respective FIT file is generated and stored in the *biometric data repository*. The personal identifier PID is also generated to provide the user with a secret value that can affect the output of the password generation process.

The encryption process is outlined in Fig. 2. The yet unsecured document is picked by a sender (originator) from the *repository of original documents*, or is created as a new one. The recipient's FIT file and PID code are fetched from the *biometric data repository* and the password generation algorithm is picked from the *repository of biometric password generators*. Each algorithm is identified by its unique algorithm identifier AID. The biometric password generator can be implemented as a plug-in to enable unlimited expansion of the FYEO system. With that, new algorithms may be added and older ones removed when considered obsolete or expired. A password is generated by the *generation algorithm* module based on the PID code and the content of the FIT file, passed next to the *document writer* module to encrypt the original document. The *encrypting application* combining these two modules can incorporate any third party document writer tool, if only the latter can support document encryption based on text passwords. Finally the FIT file, the AID code and the encrypted document are bundled in one package to be uploaded to the target user computer or sent out as an email attachment.

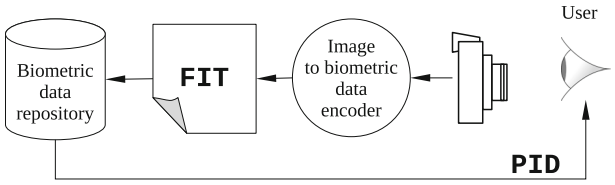

Fig. 1 FYEO enrollment process

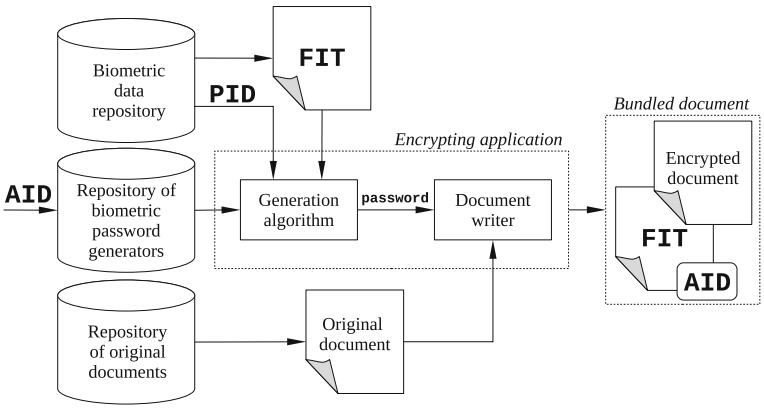

Fig. 2 FYEO encryption process (originating device)

The decryption process is performed on the recipient's (receiving) device, as shown in Fig. 3. Upon attempting to open the received FYEO document bundle, the detach process is started to extract the FIT file, the AID code and the encrypted document itself. Next, the validity of the AID code is checked in the current user's context (e.g. the expiry date, valid time of the day and date of opening the document, the allowed geographical location of the opening device, its required type or model, etc.) and availability of the corresponding password generator plug-in on the receiving device. If these conditions are met the user is prompted by a receiving device to take a face shot. The obtained image is processed by the *biometric verifier* module, which verifies if the actual image of the recipient matches the received FIT content and can be accepted. Next a text password is generated by the *generation algorithm* module based on the PID code provided by the recipient and the content of the detached FIT file and passed to the *document reader* module to open the encrypted document. The *decrypting application* combining the three aforementioned modules can incorporate any third party document reader tool, which may not necessarily be the same as the one used by the originator to encrypt the original document.

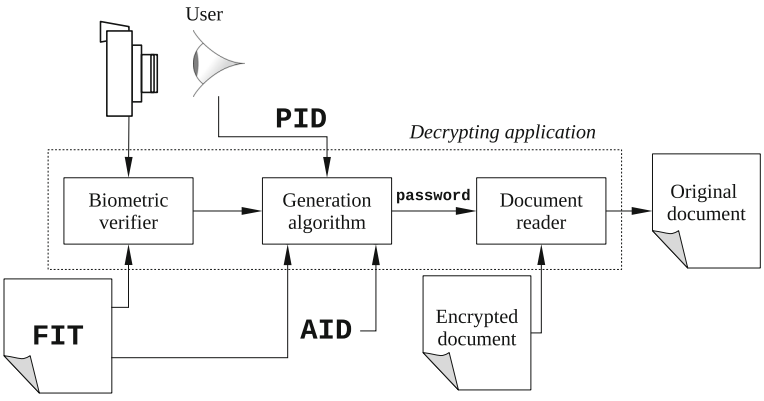

Fig. 3 FYEO decryption process (receiving device)

2.2 Threat scenarios

Below we analyze several threats that password protected documents can face during their lifetime and identify the related incidents, in particular the ones that may occur during transfer of the document over the Internet. We also assess the adequate countermeasures that the FYEO scheme can provide in that regard. They are listed in decreasing order of the chance that each related risk may occur given the number of safeguards that may be currently available to the document. On this basis we will specify further in Section 3 the desired properties of biometric data in FIT files that can be used to reduce the risk of successful offline attacks on text passwords generated based on such data.

2.2.1 Illegal interception of the document bundle

The FYEO document bundle may be stolen by a third party while on transfer or just sent by mistake to a wrong (but malicious) recipient. Its only protection would then be the password, which the attacker may try to guess. If he/she is unaware of the FYEO protection the attacker most likely would attempt the *brute force* or *dictionary* attacks on the hash extracted from the document. If the password is strong enough, e.g. is a long string of characters selected at random from a rich set like ASCII95, the brute force or dictionary attacks would fail. If however, the attacker is aware of the FYEO scheme, he/she may attempt to generate the password based on the content of the intercepted FIT file. Due to lack of the password generation software, however, the value of AID will be useless to the attacker, so the alternative to guessing of the password string might be guessing its generation algorithm. But the set of non-trivial algorithms that based on a single fixed length FIT file can generate strings of length L from a finite alphabet of N characters may potentially be infinite. So a sensible attacker would rather return to guessing the password string directly, by searching through N^L strings, instead of attempting to reproduce the algorithm used to generate it. But with N sufficiently large it would be hard anyway.

2.2.2 Theft of the receiving device

If a personal device of the legitimate recipient is stolen, the adversary may know the owner's identity and most likely will attempt using the FYEO decrypting application. He/she may look for images of the identified user on the Web, as well as in the local directories of the stolen device. Two types of "signal tapping" attacks are possible in such a case [34]. One is the *sensor attack*, when the attacker submits a biometric input (e.g. generates images) without being able to observe the discretization output of the biometric verifier module, as shown in Fig. 3. Another is the *discretizer attack*, when the attacker is able to observe the binary feature representation corresponding to the biometric input. For the sensor attack a large set of biometric inputs may be attempted one after another until one of them is accepted, whereas for the discretizer attack resubmitting past unsuccessful binary representations may be avoided and the attack completed faster. But the sensor attack may be too costly for the attacker if the biometric verifier of the FYEO decrypting application can accept only a very narrow variability range of images of the legitimate subject and reject any of his/her general photos available on the Web or in the stolen device. In other words it should exhibit a relatively high *false rejection rate* (FRR). On the other hand, a discretizer attack would require the attacker to have prior access to images originally submitted to the biometric data encoder at the FYEO enrollment point shown in Fig. 1 in order to experiment what input could generate the FIT file of interest. That however would not be possible with

the enrollment point operated offline (out of the Internet). The biometric verifier module of the decrypting FYEO application installed on the device would not help much either if it had no generation capability of FIT files. Moreover, the FYEO decrypting application may be protected against hacking by hardening its code at various levels through the application of multiple layers of diversification and obfuscation, as developers of critical software applications often do [19].

On top of that some sort of "semi-online" protection maybe implemented on the local device by the decrypting application, which could corrupt or delete the encrypted document after exceeding a certain number of failed attempts to verify biometrically the candidate subject.

2.2.3 Takeover of the receiving device

A more dangerous scenario involves the attacker taking over the receiving device of the legitimate user where the encrypted document is supposed to be opened. In such a case it might be possible to covertly retrieve the PID code of the latter by logging the keys struck on a keyboard to capture his/her images that the biometric verifier of the decrypting application could accept. This threat may also involve a phishing attack with a legitimate email message intercepted by the attacker (or a false one) prepared to retrieve the PID code and images of the subject when opening it. Like for other threats related to the takeover of the receiving device, which may be possible for any document protection method (like capturing the screen or the printer of the hacked device when the opened document is read or printed), the primary preventive measure will be responsible access management to the personal device by its owner; it should include keeping its OS up-to-date and installing only trusted applications, implementing dedicated frameworks for security policy enforcement [55] or even adopting hardware based security technologies [59]. However, thanks to the fact that opening of the encrypted document in the FYEO scheme does not require connection to any external authentication service, a simple countermeasure to this type of attacks could be just not opening the encrypted documents when the device is connected to any network. Upon completing his/her work with the document the user's PID code, his/her device's camera as well as any other sensitive information will no longer be needed – thus, not being available to potential attackers when the network connection is reestablished.

2.2.4 Takeover of the originating device

Minimizing threats to originating devices depends on the particular user scenario. If the encrypted document is to be sent for a one-time use by one or more recipients in a multicast fashion, or made ready for download by the remote users, the originating device would be most likely a document server operated by users' organization. Its defenses could then be based on the robust online protection mechanisms mentioned before. However, if the previously received document is encrypted again for forwarding to the subsequent recipient, the previously receiving personal device may become the originating one. In this case a reasonable security policy would be the same as indicated before for the receiving device: responsible (access) management and performing document encryption only in the offline mode to prevent eavesdropping.

Upon taking over the originating device, the encryption process potentially may be attacked at several points, as shown in Fig. 2. One is the biometric database, which does not contain any useful data except user PIDs. They could be securely stored in a digital wallet or other local encrypted database, for example the Apple KeyChain password manager

does so with passwords, account names, and credit card numbers [2]. Another is the library of unsecured documents that may contain confidential content, e.g. working copies of documents currently being developed by the prospective originator. Their protection could rely on either encryption with temporary passwords stored locally in the aforementioned digital wallet, or creating each such document on the fly just before encrypting and sending, prior to reconnecting the device to the Internet. These two precautionary measures are quite general and may be recommended for any password based encryption of documents exchanged over the net; they are not specifically required for the FYEO scheme to work.

On the other hand, the repository of biometric password generators is specific to FYEO. Their protection against hackers may rely on the online protection mechanisms provided by the server from where documents are originated. However, if a less powerful personal device is used, all password generator plug-ins could be kept on the external memory card used only when needed and only when the device is in the offline mode.

2.2.5 Hacked enrollment point

Photos of each prospective recipient of biometrically encrypted FYEO documents may potentially be stolen during the enrollment process shown in Fig. 1 and used next to deceive the decrypting application shown in Fig. 3. Since the enrollment point does not require any network connection to work, it should stay offline all the time. As soon as the FIT file is generated, all images of the registered subject must be effectively deleted from the server's disk. Upon completion of enrollment, they will no longer be needed for the FYEO scheme to work.

3 Crafting the FIT file

In order to counteract the threat scenarios outlined in the previous section the content of the FIT file shall exhibit several specific properties, along with other general precautions associated with the use of personal devices. These properties are twofold. First, the FIT file should be a *renewable biometric template*. According to the ISO24745 standard [20], it should be possible to create different multiple FIT files from one or more images of the same subject and each one may be used individually to verify his/her identity without revealing information about the original reference. Second, the amount of correlation between the numerical representation of each single FIT content should be minimized to provide a high entropy source for generating strong text passwords. The password should be sufficiently strong to reduce the probability of successful brute-force attacks [11]. Further below, we will examine how the aforementioned properties of FIT files can be achieved to make the FYEO scheme proposed in Section 2 compliant with the ISO24745 standard and its underlying reference architecture for the protection of biometric information based on the concept of pseudo identities [9].

3.1 Pseudo identity

According to the reference architecture mentioned before, the FIT file should provide *pseudo identity* (PI) data that do not reveal any information allowing attackers to retrieve the original biometric measurement data, i.e. images captured during the enrollment process shown in Fig. 1. On the other hand the PI data should represent the protected identity of the subject in the form enabling his/her verification by capturing the relevant biometric sample

of the claimant, i.e. a photo shot during the decryption process shown in Fig. 3. Verification of the subject may be further supported by other subject-dependent *auxiliary data* (AD) embedded in the FIT file and *supplementary data* (SD) representing some knowledge or application-based secrets provided by the subject independent of the FIT file. Note in Fig. 3 that the required SD component of FYEO combines both, the *knowledge-based secret* represented by the PID code known to the subject and a number of *application-based secrets* known to the password generation plugin that checks the validity of the AID code. Moreover, the plugin code itself may constitute one more *possession-based secret*, if stored by the recipient separately from the receiving device on a memory card for higher security [9].

The reference architecture is intended by its authors to be technologically neutral and to be able to accept any biometric verification method that could make it impossible or at least computationally difficult to:

R1: retrieve the original biometric sample from the protected identity template;
R2: uniquely link subjects within and across databases through comparison of templates;
R3: search for subjects with very similar biometric characteristics;
R4: derive comparison scores to thwart discretizer attacks on the decrypting application.

Furthermore, the postulated biometric processing technique should:

R5: enable fine-tuning of false acceptance (FAR) and false rejection (FRR) rates,

to adjust the trade-off between individual security and comfort of using the biometric verification scheme implemented on the basis of the selected method. In other words, while the verification performance should be preferably in line with state-of-the-art biometric verification methods, a limited amount of verification performance degradation (increased FRR) is acceptable as long as this is balanced with the gain in security (decreased FAR).

Based on the above we incorporated in the prototype implementation of our FYEO scheme the face identity verification engine developed by us earlier for protecting personal devices, and derived from our general Classification Framework (CF) [47]. It implemented a feature extraction method based on the 2D Gabor filter and Local Ternary Patterns. These methods have been found particularly appropriate for facial expression recognition [26]. Their suitability has been proved regarding less powerful mobile devices for the significantly lower computation time. Since computations of our engine combine the outputs of the Gabor filter and Local Ternary Patterns, we will refer to it as the *Dual-channel Face Embedding Extractor* (DFEE) engine. As argued further below DFEE satisfies requirements R1-R4 for PIs. Moreover, as demonstrated by the series of experiments reported in Section 4.1, it allowed for effective tuning of the FAR/FRR rates specified by requirement R5. That greatly facilitated the incorporation in the FYEO scheme of some additional secret knowledge on the subject's illumination and pose. The latter is the exact opposite of what has been the ultimate objective of state-of-the-art *face recognition* engines, aimed rather at *reducing* the FRR parameter to make classifiers robust for illumination or pose changes [10, 60].

3.1.1 Dual-channel face embedding extractor engine

The architecture of the DFEE engine is outlined in Fig. 4. It uses a series of face shots from the camera to extract PI data from the person's face image and the negative set of images of other persons representing the "rest of the world" AD data. Extracted PI and AD data are written to the output FIT file. As argued further below this content is irreversible, i.e. acquiring or reconstructing original images of a person based on PI and AD data would be

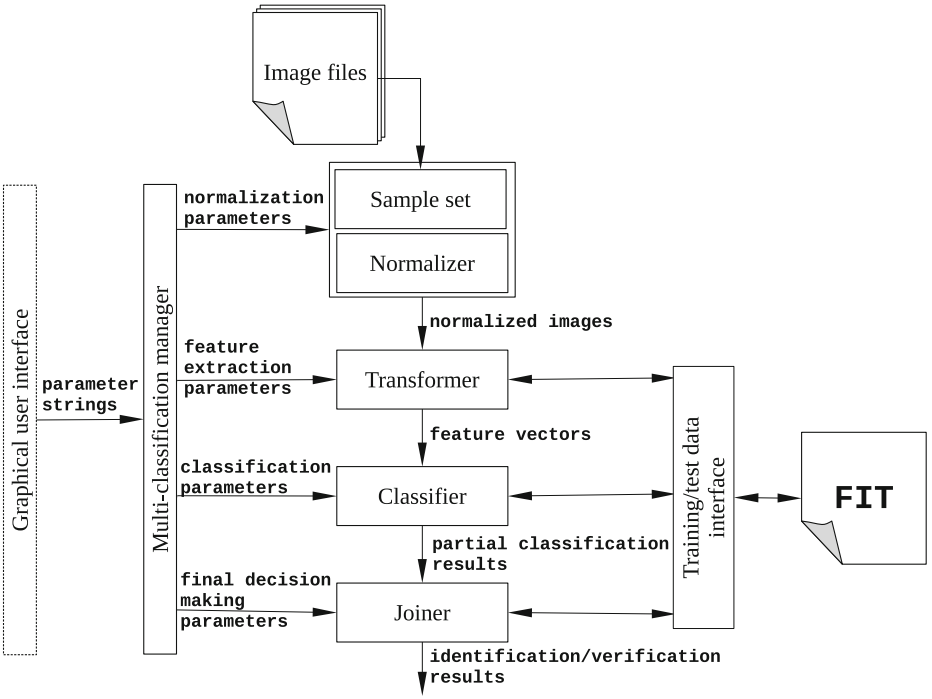

Fig. 4 Architecture of the DFEE engine

impossible. The *sample set* module reads *image files* and assigns class labels to the training samples. Data are processed in several steps: first normalization (preprocessing) of each input image is performed by the *normalizer* module under the control of the *sample set* module. Next the *transformer* module performs transformations including feature selection and extraction, and finally the *classifier* (matcher) module performs classification. In that process, several "pipelines" are created; each one defined by a specific *transformer-classifier* pair unaware of the existence of other pipelines. The *multi-classification manager* module controls the training process. Upon creation, this object receives an additional parameter in the form of a class inherited from the *joiner* (decision) module. The latter is responsible for combining decisions coming from different pipelines into the final one.

Normalizer Proper extraction of the region of interest requires detection of eye centers in each input image. It is done automatically with the cascaded classifier [53]. Processed images are scaled to the size of 64×64 pixels. The precision of face image normalization strongly affects the overall performance of the system, because the texture features used at the later stage are sensitive to scaling and translations.

Transformers The cropped images are passed to the two feature extraction pipelines; one of them calculates *Local Ternary Patterns (LTP)* derived from *Local Binary Patterns (LBP)* [52], and another performs *Gabor filtering* [25]. Additionally, each pipeline creates the secondary feature space by using *Kernel Linear Discriminant Analysis (KLDA)* [27].

The LBP operator proposed in [36] considers a small, e.g. 3×3 neighborhood of each pixel, and performs binarization in this area. The image is characterized by a set of histograms of LBP codes, calculated for specific regions, e.g. uniformly distributed rectangular areas, positioned around the nodes of a regular grid. Consequently, the LTP operator generates sequences containing N ternary values $(-1, 0, 1)$, which form the local texture descriptor (N is the number of neighbors). The LTP histogram is twice that big as its LBP origin.

Gabor filtering [25] is based on the concept of the Gabor kernel defined as:

$$\Psi_{\mathbf{k}}(\mathbf{p}) = \frac{||\mathbf{k}||^2}{\sigma^2} e^{-\frac{||\mathbf{k}||^2||\mathbf{p}||^2}{2\sigma^2}} \left(e^{i\mathbf{kp}} - e^{-\frac{\sigma^2}{2}} \right) \tag{1}$$

where $\mathbf{p}$ denotes the point within the filter mask for which the value of $\Psi_{\mathbf{k}}$ is calculated and $\mathbf{k}$ is the vector that encodes the wavelength together with the orientation of the filter. Typically, five frequencies (wavelengths) and eight orientations are taken into account, resulting in a 40-element filter bank. The convolution of $\Psi_{\mathbf{k}}$ with the image provides the local texture description in the form of the 40-element vector $\mathbf{j}$ of complex numbers called a *jet*. The face verification engine computes the jets for the same regions for which the LTP histograms are created.

The role of KLDA is to emphasize the differences between the given class and the "rest of the world" by constructing a proper feature space. In order to use it one has to define metric $d(\mathbf{p}, \mathbf{q})$. We use χ^2 distance for histograms and $L2$ norm for feature vectors. The Gaussian kernel values [52] are computed for each pair of training images as $K(\mathbf{p}, \mathbf{q}) = e^{-d(\mathbf{p},\mathbf{q})/2\sigma^2}$, where σ is the parameter for controlling the balance between good generalization and low training error, and stored in $M \times M$ matrix $\boldsymbol{K}$. Next matrix $\bar{\boldsymbol{K}} = \boldsymbol{\Pi}\boldsymbol{K}\boldsymbol{\Pi}$ is constructed, where $\boldsymbol{\Pi} = \mathbf{I} - (1/M)\mathbf{1}_M\mathbf{1}_M^T$ and $\mathbf{I}$ denotes the identity matrix and $\mathbf{1}_M$ is the M-element vector with all elements equal to $1/M$. The eigendecomposition $\bar{\boldsymbol{K}} = \mathbf{U}\Lambda\mathbf{U}^T$ is performed. For each training image $\mathbf{x}$ the M-element vector $\mathbf{k_x}$ containing distances between this image and all other samples is derived. Then, the new representation of $\mathbf{x}$ is calculated as $\mathbf{y} = \boldsymbol{\Lambda}^{-1/2}\mathbf{U}^T\boldsymbol{\Pi}\mathbf{k_x}$. In the next step the mean within-class scatter matrix $\mathbf{S_W}$ and the mean between-class scatter matrix $\mathbf{S_B}$ are calculated from $\mathbf{y}$ vectors to facilitate the construction of matrix $\mathbf{A} = (\mathbf{S_W} + \epsilon\mathbf{I})^{-1}\mathbf{S_B}$, where typically $\epsilon = 0.001$. Eigenvectors of $\mathbf{A}$ are calculated and stored in matrix $\mathbf{V}$. The final transformation matrix $\mathbf{T}$ that converts $\mathbf{k_x}$ into the new representation is calculated from the equation $\boldsymbol{\omega_x} = \mathbf{T}\mathbf{k_x} = \mathbf{V}^T\boldsymbol{\Lambda}^{-1/2}\mathbf{U}^T\mathbf{k_x}$. The size of matrix $\mathbf{T}$ depends on the number of training samples, however the negative set may be constructed arbitrarily, in particular no rules exist how big it should be or what images should be used.

Classifier The simplest nearest neighbor classifier is used by both pipelines mentioned before. For each person P the set of M_P training images $\mathbf{t}_i$ is collected, and test sample $\mathbf{x}$ is accepted as representing the legitimate user if $\exists\mathbf{t}_i \; D(\mathbf{x}, \mathbf{t}_i) < \delta \cdot w$, where $i = 1, ..., M_p$ and $D(\mathbf{a}, \mathbf{b})$ is the Euclidean distance calculated between feature vectors extracted from images $\mathbf{a}$ and $\mathbf{b}$ and transformed by KLDA. Parameter δ is the maximum distance between training samples $\mathbf{t}_i$ describing a given person, and w is a tuning parameter that controls the balance between the actual values of FAR and FRR.

Joiner The final identity verification is based on a simple voting scheme: if both pipelines agree and provide the positive result, the user is allowed to open the file. Otherwise, the access attempt is rejected.

3.1.2 Pseudo identity conformance of DFEE

Before proceeding to the experimental measurement of how strong protection of FYEO documents can be when based on illumination and pose secrets (requirement R5), first let's check whether DFEE can qualify for inclusion in the FYEO scheme by satisfying requirements R1-R4, which specify two fundamental properties of biometric template protection schemes required by the ISO24745 standard, namely *irreversibility* and *unlinkability* [35].

Our biometric template is irreversible (requirement R1) because it is based on two many-to-one transformations: Gabor filtering and Local Ternary Patterns. The original image cannot be restored from the values of convolutions of intensity function with complex Gabor kernels, nor from the histograms of ternary codes, obtained as a result of several local thresholding operations. Moreover, thanks to the introduction of secrets related to illumination and pose discussed further, the cross-database matching of facial identity templates is practically infeasible, since it was shown that images of the same person lit in different ways are less similar to one another than photographs of different individuals taken under the same lighting conditions [1]. Additionally, a precise generative model of a FIT content could not be easily constructed even if the attacker would be able to acquire multiple templates created for the same person under different conditions. Although some of the early face recognition algorithms tried to utilize the fact that all of the images of a Lambertian surface taken from a fixed viewpoint but under varying illumination lie in a 3D linear subspace of the high-dimensional image space, a human face is actually *not* a Lambertian surface [5] (due to the presence of cast shadows, for example). Therefore, the features of unknown images cannot be deduced from the values stored in acquired FIT files, thus cross-database tracking of a particular person is not possible (requirements R2 and R3). Moreover, DFEE can produce renewable FITs, i.e. when a biometric database (see Fig. 2) is compromised, new instances of FIT file can be reissued at the enrollment point in different configurations of the subject. Besides pose and illumination, parameters of the grid used by the *normalizer* and *transformer* modules (shown in Fig. 4) may vary between databases. In particular the size of the grid, positions of its nodes and their order may vary, making FIT files incomparable. To diversify them further additional transformation functions can be applied during enrollment and verification: face images can be distorted in the signal domain prior to feature extraction, or the feature values can be transformed using a noninvertible function, which should be "smooth only locally and not globally" (examples of such transformations are provided in [40]). The parameters of the function can be governed by a random key and treated as a secret or at least stored separately.

Requirement R4, which is aimed at mitigating the risk of a successful discretizer attack is also satisfied by the DFEE engine. As outlined in Fig. 4, during the enrollment phase the respective FIT file is generated by DFEE, whereas during the decryption phase only a verification result is produced without generating (recoding) any new FIT file for comparison with the original one.

3.1.3 FIT file content

The algorithms described above produce a FIT file with PI data values representing the individual for whom the DFEE engine was trained. It contains numerous settings of a classifier as a collection of XML data, e.g.:

```
<?xml version="1.0"?>
<opencv_storage>
<input_type>VECTOR</input_type>
<sigma>10.</sigma>
<epsilon>1.0000000000000000e-003</epsilon>
<transform type_id="opencv-matrix">
  <rows>1</rows>
  <cols>100</cols>
  <dt>d</dt>
  <data>1.5633980072441953e-001 ... </data>
</transform>
<number_of_classes>2</number_of_classes>
<!-- user facial data -->
<ID0>0</ID0>
<name0>TARGET</name0>
<number_of_images0>12</number_of_images0>
<M00 type_id="opencv-matrix">
  <rows>640</rows>
  <cols>1</cols>
  <dt>d</dt>
  <data> 1.8662906625098184e+001 ... </data>
</M00>
...
<M011 type_id="opencv-matrix">
  <rows>640</rows>
  <cols>1</cols>
  <dt>d</dt>
<data>1.4558722036316027e+001 ... </data>
</M011>
<!-- facial data of others-->
<ID1>1</ID1>
<name1>OTHERS</name1>
<number_of_images1>88</number_of_images1>
<M10 type_id="opencv-matrix">
  <rows>640</rows>
  <cols>1</cols>
  <dt>d</dt>
  <data>1.4594344511234386e+001 ... </data>
</M10>
    ...
<M187 type_id="opencv-matrix">
  <rows>640</rows>
  <cols>1</cols>
  <dt>d</dt>
  <data>7.9162100568885831e+000 ... </data>
 </M187>
</opencv_storage>
```

The `<transform>` element contains data of the KLDA transformation matrix **T** defined in Section 3.1.1 before. Whereas data produced for the subsequent image files are stored in elements marked with tags of the form `<M[cl][im]>`, where `[cl]` denotes the class

number, i.e. '0' for the TARGET (the series of the subject images) and '1' for the OTHERS (the series of the "rest of the world" images) portion of the FIT file and [im] is the number of each image within the class. According to the ISO24745 standard and its underlying reference architecture mentioned before they constitute respectively the PI and AD data. It may be seen in the example listing that twelve images M00 ...M011 from the training set and 88 images M10 ...M187 from the negative set were processed by DFEE to produce this particular FIT file.

Further in Section 4 we will analyze to what extent illumination and pose of the person's face in the photo can control the data values in the TARGET part of the FIT file during the training phase at the enrollment point, so as to be able to effectively tune DFEE to accept that person's image during the verification phase only in very specific shots.

3.2 FIT based password autofill

As illustrated before a FIT file produced by the DFEE engine consists of byte strings that are predominantly made up of decimal digits. If used as a source of entropy for automatic password autofill, e.g. by drawing random digits from it to provide input to the relevant generation algorithm, the digits should exhibit randomness as close as possible to:

R6: the uniform distribution to guarantee maximum uncertainty of generated passwords.

In order to ensure that all non-numerical substrings including SPACE and TAB characters, as well as those parts of numerical strings whose correlation may result from the context of their occurrence in the FIT file must be removed. According to the example listing in Section 3.1.3 the latter are the content of <sigma>, <epsilon>, <rows> and <cols> elements, as well as sign, non-fractional parts of floating point numbers in <data> elements and their respective exponent parts. Formally, for each number represented by a string of the form F = "$[-]d_0.d_1 \ldots d_{16}E[+|-]00d_{17}$", where $d_{i=0,\ldots,16} \in [0, 9]$ and $d_{17} \in [0, 3]$, substrings of the form "$[-]d_0$." and "$E[+|-]00d_{17}$" have to be removed to get only each number's decimal fraction part. Thus obtained text file, each one of several thousand lines (32-digit strings) may constitute a pool of random digits to be used as inputs to various password generation algorithms.

However, when selecting the concrete generation method, one should make sure that the related algorithm would not introduce in its output any subtle patterns of its own. Such patterns could be implied by any particular statistical properties of thus obtained set of digits, as well as by intrinsic properties of the function calculated by the password generation algorithm. The latter may involve mathematical properties of the function's class that the attacker can discover such as converging to a repeatable cycle of output values or non-injectivity, as well as arithmetic properties of its implementation implied by the limitations of the underlying computer arithmetic such as round-up or truncation errors. To avoid that, generation algorithms should use techniques advocated in [11], which are known to preserve entropy present in a source of random data and to reduce any bias that may exist in it.

One group of these techniques applied to the FIT based pool of random digits calls for selecting a random block of bytes and combining them by using a mixing function [8]. A simple one could calculate byte-wisely just a modulo check of the block, whereas a stronger mixing function like SHA-256 [43] may calculate a more complex block hash. By the argument provided in [11], combining uncorrelated input bytes in that way would produce output symbols of lesser eccentricity. Probability of a single digit in the FIT based dataset may be formally modeled as $p = 0.1 \pm E_{in}$, where $E_{in} \ll p$ denotes eccentricity measuring imperfection of the desired ideal uniform distribution of input digits from the

block. Since the modulo of the sum of two decimal digits (bytes) is addition without carry in which the output symbol always changes with a change in either input symbol, the output symbol eccentricity E_{out} would be less than the eccentricity of n added input symbols on the order of $E_{out} \approx E_{in}^{n}$. This property of the simple mixing function is exploited by the sample automatic password generation algorithm shown in Fig. 5.

In step #1, a block of lines with decimal digits is selected at random in the FIT file crafted accordingly and in step #2 its modulo checksum is calculated as a decimal string. Finally in step #3 digits are drawn at random from that string and for each of them a symbol is selected from a subrange of ASCII95 symbols. For 32-character passwords 32 digits need to be drawn. This scheme can produce a considerably large set of password generation algorithms since the pseudo-random number generator controlling the selection of blocks and digits could be seeded with arbitrary values. Example passwords generated by the above scheme for various seeds are listed in Table 1.

A variant of the password generation method outlined in Fig. 5 may use in step #2 the much stronger SHA-256 mixing function and select in step #3 the output symbols from a slightly reduced set of printable characters, e.g. ASCII85. The thus generated passwords for the same seed values as used in Table 1 are listed in Table 2; they seem to be no less demanding than the former ones.

Another group of techniques preserving entropy in a random data source may explore one useful property of reversible compression – by being lossless it can effectively reduce eccentricity of the imperfect randomness of the input symbols to be compressed. By the argument provided in [11], if the same amount of information is to be present in the shorter compressed output as in the longer input, then on average, the probabilities of symbols in shorter sequences would be more uniformly distributed than the probabilities of symbols in the longer sequences. In other words, the overall statistical quality of the compressed output byte-stream can be improved. This idea is exploited by the sample automatic password generation algorithm shown in Fig. 6.

In step #1, a line containing decimal digits is selected at random in the input FIT based dataset. In step #2 the line is searched in a randomly selected direction (from left to right or from right to left) for digits that upon concatenation in step #3 (steps $\#3_{LR}$ or $\#3_{RL}$ respectively) would represent a decimal number constituting a code-point in the ASCII95 set of printable characters. Selection of the first digit is made at random, whereas the subsequent digits are selected on a random first-come basis until the legitimate code-point of a printable character is obtained. The number generated in this way, is a reversible compression of a two- or three-digit string drawn from the selected line; by the argument mentioned above

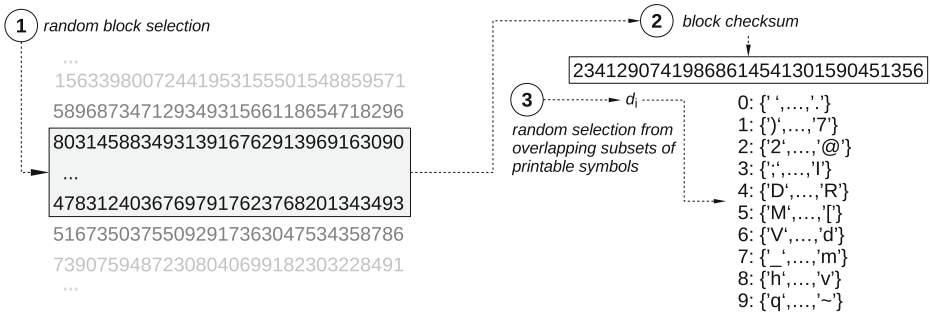

Fig. 5 Password generation based on simple mixing byte blocks

Table 1 Passwords generated when using a simple mixing function

Seed	Password	Seed	Password
117	[E<@u(y79Mm4}m>Gom)-xY}@y@was-;Q	23147	^7Z>oRNP-J4m7Fl9+2VwXakjR.7t(X:
1304	bJxQIf]vWOt-h]>3oRIEMPVku>R>zu6*	33672	o<9R;mdHv]?wM-I%_BUcJj>nmYD%HW6
1305	~Ir p_CF3<O b^W@EeslV'GSqRrqrsA(	36725	4D_5<PlgxhOf.DAF{sNyM<'yWM/bt,v@
3367	lP.DTSxQy_lbq*EY'Qllkm4L_-P[IiZ3	43756	;KY9#{b(@SZp$h.x6mVdOYkO=vK-{/&*
3816	,ylargDoHWf'2<Nrlb>dMsV#luPE!I<%	73147	'ME,J@_IEu{GRjWF jGH+yk*eYv5I^7]
3856	so4?ju%nl<z3*;ka.gmg,YA8JsGQTbQE	191217	(Z55U\i.8YE.v+d?Ij/;?;^xZwd>ZB7X
4335	*QfacWa:&(!'#(o_/XkjDm-4u0zk1+Sv	191312	ze5<bYxqD_54 Vl2&'s_"t6O4x,-hCIM
7305	K3t>dYJ*60vn Kec@c+#eU^Pu#-Osr\:	208113	t>I.+IR@cFu5cr>k?rlhWe@l-n;>-aCX
7335	O+aLh9<;j9<.Nq!Z4IN/)oldKZ,@rm<D	305873	sC5V@}>p< YCTv_Dgl3Y?!jp@P?;=U9'
8361	-M0}PHVN?j^7'G74RJ1 XD,ybddSM5RV	911312	O l26'E4rq?R_>9HQn^i<E\R5L}7b,>+
8877	f;OH_HSHfEKY8FW0.^7eKnlk8ZZfIv4O	1234567	35GENq$3fxNjjWC=3'cWAT>ij_IYM?mp
9857	74c,X;jXVGDhO)7gPQi{Yo[MAXXk:?]_	1304873	?_vm1q-,>r-tj]6KQWT@-D)H-GNXIF+j
12367	kG-3<Njirjrj!vykk xKB92DTq, d2b	130487308	nTXg0Xn)?5DHH{v>g)Rb2Ijdrv+V1LbD

it would represent a symbol of the reduced eccentricity in comparison to the eccentricity of each input digit used to calculate it. By modeling the probability of drawing a single input digit from the line selected randomly in step #1 as $p = 0.1 \pm E_{in}$, where $E_{in} \ll p$, concatenation of two or three decimal digits forming a string that represents either one of 68 two-digit code-points or one of 27 three digit code-points (out of the total of 95 printable ASCII output symbols) would yield eccentricity E_{out} of the output symbol reduced to $E_{out} \approx 0.79E_{in}^2 + 0.21E_{in}^3$ on average. Example passwords generated by this scheme for the same seeds as used in Tables 1 and 2 are listed in Table 3.

4 Scheme validation

In Section 3.1.2 we verified the properties of the DFEE engine qualifying its incorporation into the FYEO scheme with regard to irreversibility and unlinkability, preventing attackers from comparing intercepted FIT files when searching for the same or a similarly looking person on the Internet. The possible way around could be locating the subject behind his/her email address extracted from the intercepted FYEO bundle to look for any available photos of that person and to use them for signal tapping attacks. This, however, would require access to the FYEO decrypting application installed on the subject's personal device. And even so, such attacks would be limited to sensor attacks, as no FIT files can be generated by the underlying biometric verifier to prevent any comparisons of its output required by the discretizer attacks.

The above exhausts the scope of requirements R1-R4. However, checking whether the candidate biometric verification engine can meet requirements R5 and R6 require a series of measurement experiments; results obtained for the DFEE engine are reported below.

Table 2 Passwords generated when using the SHA-256 mixing function

Seed	Password	Seed	Password
117	1H.*G@5DCM3Ak;&11OiO1h/FQ0k"KNA2	23147	3ArU$1LrLVAM>hn2)I=%@PBZLA7K%-11
1304	@:V)L2Ig>UAMH#N2_m7%1LW*sA2?,(AS	33672	3&30r0P+'L3+XdPAiXBW1H7BN@5_P&@p
1305	2D[-u2.J<oAMQ8&2'>5+0eb:;11'I"AM	36725	1c-pt0KEZ&2e=j%@l%[O@:Nk,@qB.Y@:
3367	1Gq+$0f<GsA7K4,3&<F&3Ak5$@:D5R1,	43756	@Q-;$A7RYV2e4jX2dp<02DAQM2Dd0D0K
3816	A2Z;U@5p/!0K<Ms3A=nr2.U5X3Fam)2'	73147	2)/Mu1Lt',3&"f%0f:[>2)dNN2)$jIAn
3856	2.L,S2E*HOA2e$S2e+d'0f3?"3&'ZN0f	191217	3B:Y/2'Wm0@PTi!0f<HP3&tM,AiMJU2_
4335	2eFdVA7RE+2_[-D0O\I#1H@3M3AE9H0O	191312	1cRBN3&b8%@q0(+@Uh*$2ISO'0kFiO0J
7305	2'<WL@U'h,2Ing+@:D,QAMdpVA7I;M1c	208113	@5B'O11N."3ANEJ2DI""1,:^G@l?)N1,
7335	0ed/tA27^S2dnX)@UqJX@lIh.2'<QL1h	305873	0fV0M0ed,uAS!SV@lZP]3&'j/@pq;R1c
8361	3+=Qq2J"X(@QA!U@:'t-AiD_[@q7Q.3&	911312	@:j%Z1H&Z$2dnFL@kqUT1Li.M@l%RN3+
8877	ART.,3&3=%2e?SUAiMPX2Dnu#ARo7.0K	1234567	0KCjFAnEf-AmmQ*@Q65)2)A]#lhC2QAi
9857	@:D2V1c@9L0fCgCA7IMS3&'fV0kFoO1,	1304873	@50ZP1lOj)An5IlAhuA(2dnXV1c-sH3+
12367	3Fjg!AnF&30Ouss2.BuN2.C00@PqRO0J	130487308	2)[?K@PB'MARdGR@52=R@q0(\2DR!t@:

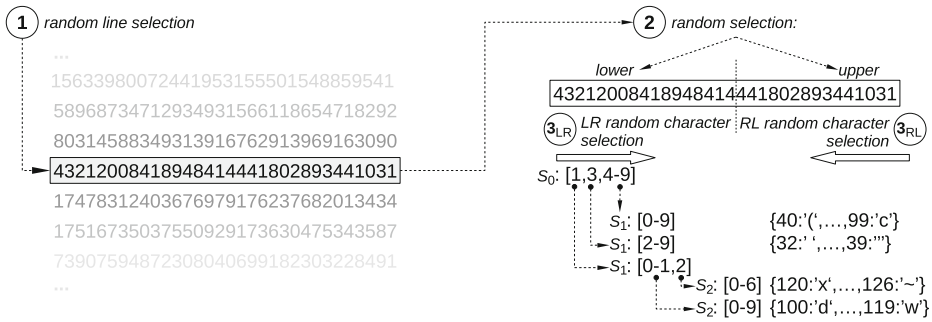

Fig. 6 Password generation based on reversible compression of byte streams

Our experiments have been a rather routine activity, aimed to validate the FYEO scheme against two crucial requirements specified in Section 3: R5 on finding the acceptable balance between the increased value of FRR limiting user's comfort of using the scheme and the decreased level of FAR indicating the gains in security of using it, and R6 on randomness of the generated password strings. Test data for our validation experiments were selected from widely available repositories of realistic images of different people, photographed in various lighting conditions and poses. The obtained FRR and FAR values were analyzed against requirement R5, whereas the data generated on their basis in the form of FIT files were processed using algorithms specified in Figs. 5 and 6 to generate sample passwords. These passwords were then examined using reliable tools to assess their strength.

Although results of these experiments presented further below one may consider not very innovative, they have been of key importance for conducting legitimate comparisons of our FYEO scheme with the related solutions described in Section 5 and for accepting it as a reliable security component in two experimental document exchange platforms implemented by us as mentioned in Section 6.

4.1 Feasibility of fine tuning the FAR/FRR rates

We wanted the biometric identity verifier incorporated in the FYEO scheme to accept only a few specific arrangements of the user's face in relation to the device's camera, while most of the others should be rejected. These two image aspects would be very convenient to model users' behavior in front of the camera of the personal (mobile) device during work; it

Table 3 Passwords generated when using reversible compression of random byte strings

Seed	Password	Seed	Password
117	+`{a-w=.YYppp[([\~f(p=wzTp/`p[(D	23147	rRCA#)M-f~pp~[[ppD`^pzO~D[p\TTDz
1304	zR!3]h,@D(<[(Y`[D`pT^'pppHwpOB~w	33672	L+=W3%D<zTw~D=`=`O`pI`<Hp\(`[p'f
1305	c4Jz^\ aYpY`\DT>I(pwpzp*/ppDppY[	36725	3lH150<IwfDw=DOI`Op>p[pzppz/`OwI
3367	eR!*8-(_[pz>~```\>p`p\TfppHIwp`pz	43756	7$3ufg\.`\I<pp[~p(^^*D`\z~/\ppTI
3816	ib&#!blmI*=Ip'YOHD~~^pp`BwBp>Bp*	73147	T9n=\/;R'/`=pT~Yp`DppI'fppp~'(^p
3856	_^}tP'YQ<zf>`=\B\\'//p^OD=*HpT^p	191217	7m" Jl^>O>T[~[p<Y=~=T^[p>YD~wpz\
4335	M"rlO=)"[p=O'f<Ip\T[pzpz\`Tf'fI'	191312	[fsF7;*[^DfH`~ppT`p*IwT(Opp/'Dp
7305	=Y}fL!r`ppp~wDp'ppO^H\YD`>TIf'pp	208113	={S]KE#v/ppw*p/pp/('/zOp>[pfBB>(
7335	=PV&)8CIp~`[/<Dpp'Dp>(^pp*pp`zT[	305873	\Y6Kzy8$`>*p`~f\^YI(p(\\~f~`[\(B
8361	2[0ODGJoI`\>[p'^*=Ipp`Op~Yp*`fIw	911312	:{qB.:YJ`Dp(</(z(p>Op'/'I\'pYf`p
8877	^#Q%-=*c'p(p^Bp/BwOH\p~\=IwpH[>D	1234567	^Ab7_';>DT'p'DOpz`Ip*'~`H[Opp'[/
9857	"Zi)6JG@IIY~I'TpOT[zp'\~wp>pp'>`	1304873	0ZD,J/48=w<H`[p*><`fp*(D(I\'/I`w
12367	".SAV(TXp==/Dzwpp\DO<`\(I(p/ODB'	130487308	,G+]Cz!R<I'[pp''~BYTIpp\`D<D<\<z

could matter in which hand the smartphone is held, at what angle the screen of the notebook or tablet is tilted, whether the personal computer is kept on the user's lap, etc. Therefore the objective of our experiments was to show that if the illumination and pose at the time of identity verification do not match the conditions present when training DFEE, the value of the FRR metric would increase considerably and make pointless the attacker's break-in attempts. Thus, paradoxically, our concern was to obtain high values of FRR and reach the point when despite that, the legitimate subject would be able to pass the identity check.

When examining how the illumination conditions can affect the process of biometric identity verification, we used the Extended Yale Face Database B (EYFD-B) [14] containing 5760 images of ten unique people, each one seen under 576 viewing conditions; samples are shown in Fig. 7. Respectively, to study the impact of the subject's pose on the biometric identity verification process we used images of the CMU-PIE database [45], including over 40,000 facial images of 68 people, each person imaged across 13 different poses, under 43 different illumination conditions, and with four different expressions in both *expression* and *talking* sessions; samples are shown in Fig. 8. In both types of experiments we used the negative set containing about 100 images randomly selected from the FERET database of 25389 files in total [38].

4.1.1 Illumination experiment

All EYFD-B images are divided into subsets, according to angle $\gamma = \max(\alpha, \beta)$ between the direction of the light source and the camera axis, with α measured longitudinally and β latitudinally in five 15° intervals from 0° to 75° inclusive. We selected images representing the concrete lighting directions specified by *azimuth* A and *elevation* E. Each one may be marked either with '+' or '−' indicating respectively its 'right' or 'left' and 'up' or 'down' values. Additionally, apart from the direction, in every subset deviation from the frontal illumination was assigned to one of two classes, namely *small deviation S* or *big deviation B*. For example $A- E+ B$ indicates the subset in which all faces are lit from the upper left with a high deviation angle. As shown in Table 4, in this experiment we succeeded in getting decreasing FRR values each time the actual illumination of the subject during the testing phase is similar to the training phase. The cells with the bold content indicate which testing sets yielded the lowest FRR value when a particular set of template images was used to train the system.

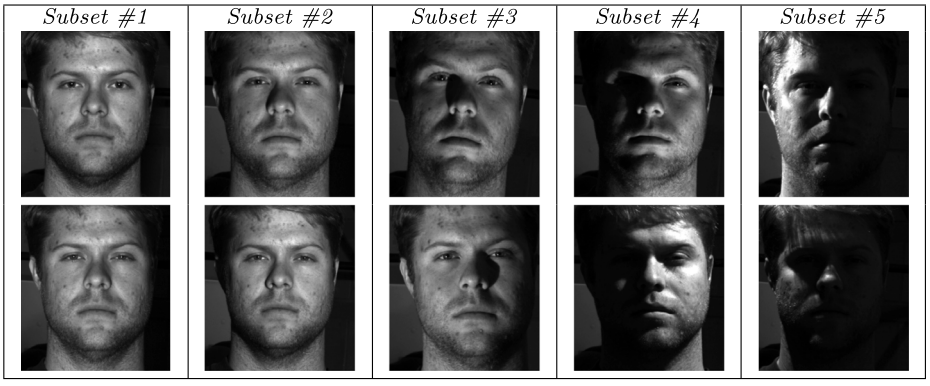

Fig. 7 EYFD-B database (illumination) samples

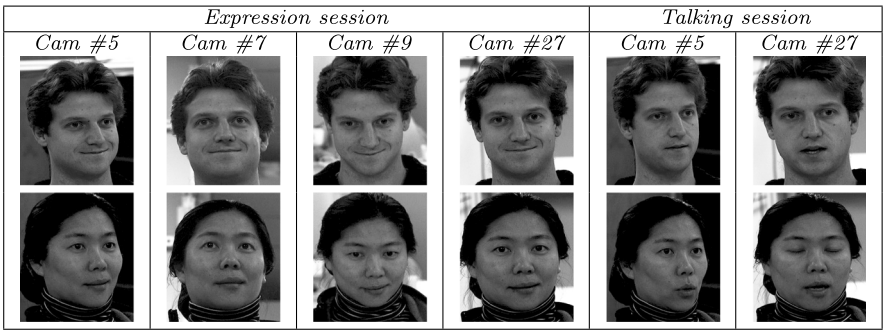

Fig. 8 CMU-PIE database (pose) samples

Note that when deviation from frontal illumination was high in the template images, the lowest values of FRR were achieved when the light was shed from the same direction as in the training images but the deviation from the frontal direction was small. For example the best results were obtained for the $A+\ E-\ S$ testing set when DFEE was trained with the $A+\ E-\ B$ set, for the $A-\ E+\ S$ testing set when trained with the $A-\ E+\ B$ set, and so on.

On the other hand, if the angle of light incidence was small in the template images, the value of FRR reached minimum if in the test images that angle was also small but the direction varied slightly (in most cases the light was shed from the top instead of from the bottom or vice versa). For example, the lowest FRR value was obtained for the testing set $A+\ E+\ S$ when DFEE was trained with the $A+\ E-\ S$ set.

Based on the above it may be argued that effective tuning of the FAR/FRR rates is possible with illumination of the subject and the related data can be feasibly embedded in the FIT files produced by the DFEE engine, as required by the standard for biometric template protection schemes based on pseudo identities (requirement R5).

4.1.2 Pose experiment

Two tests were performed using the CMU-PIE images, which results are listed in Table 5:

Test 1: Training set included images from the expression session plus one photo from the talking session, all captured with Camera #5 (side view). Testing set T_A included all images from the talking session captured with Camera #5; testing set T_B included images from expression and talking sessions captured with Cameras #7 and #27 to incorporate photos with approximately frontal views and Camera #9 to incorporate views slightly from above the subject.

Test 2: Training set included images from the expression session captured with Cameras #7 and #27 plus one photo from the talking session. Testing set T_C included all images from the talking session captured with Camera #27; testing set TD included images from expression and talking sessions captured with Cameras #5 and #9.

In this case the characteristics of the CMU-PIE database [45] allowed us to check only two poses; however, since a large number of images was available for each, we were able to divide the data into training and testing samples representing the same pose for the same person. All the results were similar – whenever the pose did not conform to the one used during the training process, FRR increased significantly and identity verification was practically impossible.

Table 4 Results of the illumination experiment

Training set (TrS)	Testing set (TeS)	FAR (%)	FRR (%)
$A+E-B$	$A+E-S$	**3.59**	**45.50**
	$A+E+B$	1.23	93.92
	$A+E+S$	2.32	86.87
	$A-E-B$	0.15	97.16
	$A-E-S$	1.18	82.88
	$A-E+B$	0.24	99.61
	$A-E+S$	0.93	88.65
$A+E-S$	$A+E-B$	7.40	77.46
	$A+E+B$	7.11	88.89
	$A+E+S$	**9.98**	**61.90**
	$A-E-B$	6.27	91.67
	$A-E-S$	8.59	62.04
	$A-E+B$	6.46	90.87
	$A-E+S$	8.26	78.89
$A+E+B$	$A+E-B$	3.25	93.99
	$A+E-S$	3.33	86.26
	$A+E+S$	**3.68**	**81.85**
	$A-E-B$	0.47	97.84
	$A-E-S$	0.32	99.10
	$A-E+B$	0.44	97.30
	$A-E+S$	0.10	100
$A+E+S$	$A+E-B$	0.52	95.24
	$A+E-S$	**3.74**	**58.10**
	$A+E+B$	1.98	90.63
	$A-E-B$	0.12	100
	$A-E-S$	3.18	74.04
	$A-E+B$	0.59	95.97
	$A-E+S$	4.26	73.89
$A-E-B$	$A+E-B$	0.17	94.59
	$A+E-S$	0.42	93.24
	$A+E+B$	0.11	99.32
	$A+E+S$	0.58	93.44
	$A-E-S$	**1.06**	**82.88**
	$A-E+B$	0.57	92.66
	$A-E+S$	1.56	84.86
$A-E-S$	$A+E-B$	2.84	94.27
	$A+E-S$	3.33	89.64
	$A+E+B$	1.47	99.32
	$A+E+S$	3.44	91.89
	$A-E-B$	2.48	97.30
	$A-E+B$	1.98	99.23
	$A-E+S$	**2.51**	**56.76**

Table 4 (continued)

Training set	Testing set	FAR (%)	FRR (%)
$A-E+B$	$A+E-B$	0.07	98.97
	$A+E-S$	0.39	93.69
	$A+E+B$	0.14	97.30
	$A+E+S$	0.76	91.12
	$A-E-B$	3.37	87.97
	$A-E-S$	1.25	91.89
	$A-E+S$	**1.93**	**83.24**
$A-E+S$	$A+E-B$	0.24	92.34
	$A+E-S$	0.95	78.15
	$A+E+B$	0.68	96.96
	$A+E+S$	1.54	61.00
	$A-E-B$	0.21	95.14
	$A-E-S$	**1.79**	**45.41**
	$A-E+B$	0.38	91.44

It may be argued that effective tuning of the FAR/FRR rates is possible with the pose of the subject and the related data can be feasibly embedded in the FIT files produced by the DFEE engine. Like in the case of illumination aspects of the image considered before, the requirement R5 can be satisfied with pose aspects as well.

4.1.3 Fine-tuning algorithm

As argued in Section 3.1 the orientation (lighting and pose) of the subject's face in the field of view of the camera cannot be recreated from the FIT file. In consequence, it can successfully constitute a knowledge-based secret conforming to the ISO24745 standard [20]. Moreover, the experiments described earlier in this Section show that an efficient tuning of the FAR/FRR rates is possible over a useful range of azimuth and elevation angles α and β. Their results listed in Table 4 can be comprehensively visualized with the plot in Fig. 9.

Consider each photo t from sets TrS and TeS in Table 4 to constitute point $\mathbf{t}_i =< r, \alpha_i, \beta_i >$ in a spherical coordinate system. Since in all photos the distance of the subject from the source of light was constant and the subject's face was always lit from the front all corresponding points $\mathbf{t}_i$ may be assumed to lie on the surface of the unit semi-sphere with the center at $< 0, 0, 0 >$. With $\mathbf{t}_0 = < 1, 0, 0 >$ as the reference point (all faces lit perpendicularly, i.e. at angles $\alpha_0 = \beta_0 = 0$) a deviation of photo $\mathbf{t}_1 = < 1, \alpha_1, \beta_1 >$ from $\mathbf{t}_0$

Table 5 Results of the pose experiment

Test #	Testing set	Poses compatible	FAR (%)	FRR (%)
1	T_A	yes	4.43	14.74
	T_B	no	2.36	89.29
2	T_C	yes	1.66	12.48
	T_D	no	0.62	94.29

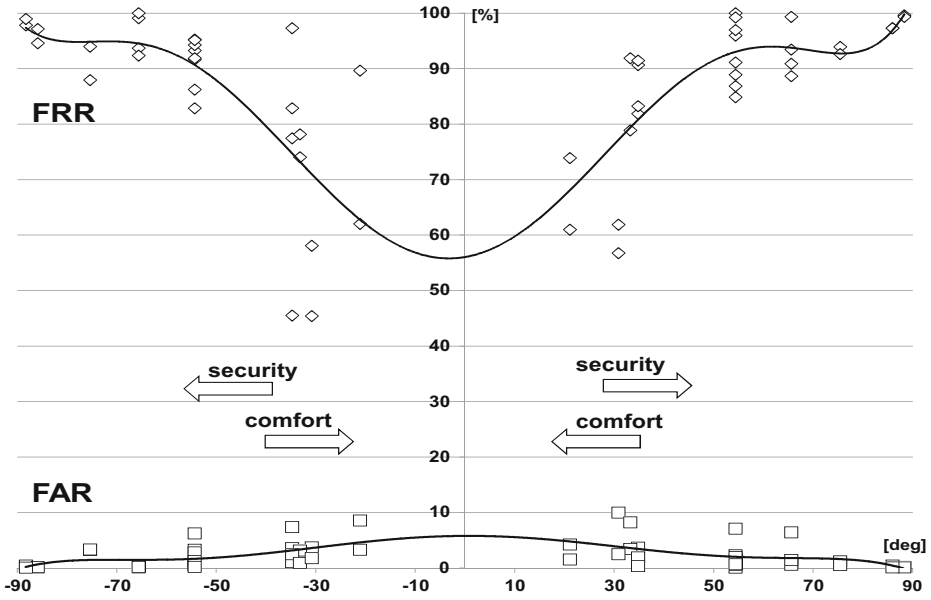

Fig. 9 FAR/FRR fine-tuning effectiveness

(faces lit at different azimuth and elevation angles α_1 and β_1) can be measured by central angle $\Delta\sigma$ between them and calculated as [58]:

$$\begin{aligned} \Delta\sigma &= \arccos\left(\sin\beta_0 \sin\beta_1 + \cos\beta_0 \cos\beta_1 \cos(\alpha_0 - \alpha_1)\right) \\ &= \arccos\left(\cos\beta_1 \cos\alpha_1\right) \end{aligned} \tag{2}$$

Values of $\Delta\sigma$ along axis X in Fig. 9 were calculated using the respective median values of angles α_1 and β_1 found for each set $TrS_i \cup TeS_i$ of photos, $i = 1, \ldots, 56$, listed in Table 4. For $\beta_i < 0$ the corresponding value of $\Delta\sigma$ was assumed negative. The trend lines for the corresponding FAR and FRR values interpolate the respective datasets with a polynomial of degree 6. It may be readily seen that for less significant deviations of $\mathbf{t}_i$ from the reference point $\mathbf{t}_0$ the values of FAR gently rise with the much steeper decrease of FRR values.

In other words, biometric identity verification based on less deviated photos can provide more comfort for the legitimate user, whose chance of being rejected is relatively lower and requires less effort (attempts) to "convince" the system to let him/her in. On the other hand, the related increased FAR values indicate that such biometric identity verification system is less secure, as chances of acceptance for impostors using photos of legitimate users in their more obvious poses grow.

An upheaval in thinking about the secure verification of biometric identity, which we wish to accentuate with Fig. 9, consists in the use of images that significantly deviate from the commonly understood "normality" represented by the close proximity of $\mathbf{t}_0$. Our experiments indicate that the limit of possible deviations is determined practically only by the level of FRR that the user can accept. This is because that although FAR values are lower for more significant deviations from $\mathbf{t}_0$ (the overall security of the system increases) the user's comfort decreases, as more attempts may be required from the legitimate user to get in.

The values of FAR and FRR marked in bold in Table 4 were found by us experimentally as the ones that can ensure a relatively high system security with acceptable effort of the

legitimate user. In that effect the FYEO enrollment point in Fig. 1 should implement steps for FAR/FRR tuning specified by Algorithm 1.

Algorithm 1 Fine tuning.

Require:
single photo t, collections S of N input photos and T of M training photos, FIT file F, tolerance parameter w, number of verification attempts M', time limit $\Delta\tau$ for verification attempts.

Ensure:
$N \geq M$

1: **repeat**
2: take photo t
3: **if** CheckValidity(t) is true **then** ▷ meaningful and normalizable content
4: add t to S
5: **until** SizeOf(S) is N
6: draw M photos from S to T
7: generate FIT file F for T
8: M' = VerificationAttempts(F) ▷ with $S - T$ photos or with new photos shot at the enrollment point (reproducibility test)
9: **if** $M' > M$ **then**
10: modify tuning parameter w
11: **go to** 8
12: **else if** Timeout($\Delta\tau$) **then**
13: **go to** 6
14: return F

The listed pseudocode may be implemented with any face extraction engine. In Step 3 image normalization operations are performed. In the case of our DFEE engine employed in the FYEO scheme proposed in Section 2 they involve detection of eye centers and scaling, as explained in Section 3.1.1. For other engines normalization may differ in some detail, according to the particular extraction model. In Step 8 the user's effort required to open the encrypted document is estimated. Upon enrolling, the user can instantly assess how easy it would be for him/her to remember and recreate important details of the pose in front of the camera when opening later his/her encrypted documents. If too many attempts are needed the tuning parameter w can be adjusted manually by the enrollment point operator in Step 10 – to best fit the individual characteristics of each user. If that does not help, the extractor should be retrained with other photos in Step 6. In the case of our DFEE engine parameter w controls the maximum allowable distance between training and testing images of the subject.

It is worth mentioning that other implementations of the face verification engine may also be used by the FYEO scheme proposed in the paper. For example, embeddings provided by a deep neural network may be used instead of feature vectors, Euclidean distance may be substituted with cosine distance or similarity value calculated with the use of the probabilistic linear discriminant analysis (PLDA). However, in each case the general mechanism remains unchanged: the training data variance must be taken into account and fine tuning of the relevant tolerance parameter w is necessary to set the proper FAR/FRR ratio respectively for each enrolling person.

4.2 Text password quality

Crafting the FIT file as proposed in Section 3.2, consisting of extraction of decimal fraction strings of the respective numerical values in the transformation matrix and jets, is aimed at providing the pool of equally probable input symbols to draw from when generating passwords. Although the techniques outlined in Figs. 5 and 6 were argued to effectively reduce bias in their distribution, in addition, we experimentally measured the degree of imperfection of the entropy sources obtained from FIT files produced by the DFEE engine in the illumination and pose experiments described before and assessed the guessability of the text passwords generated thereof.

4.2.1 Randomness of FIT based datasets

To evaluate the entropy of the FIT based random digits we used 39 datasets with random decimal digits, crafted as proposed in Section 3.2 on the basis of 38 FIT files generated by the DFEE engine in the experiments reported in the previous subsection and one more FIT file generated for its regular user. These datasets were analyzed first with the EasyFit tool [33], to ascertain which probability distribution fits best their random content. Out of four matching probability distributions, the highest *goodness of fit* metric $GOF = 1$ was returned in two standard fitness tests (Kolmogorov-Smirnov and Anderson-Darling) for the discrete *uniform distribution* and the *negative binomial distribution*. Results of this experiment are summarized by two plots shown in Fig. 10.

The lower plot indicates frequencies of all ten digits in each respective dataset. It can be seen that the uniformity of their distribution is not perfect and a slight skew can be observed at both ends of the $[0, 9]$ range. It is most likely caused by the DFEE process arithmetic rounding down to 9 or up to 0 the least significant digit d_{16} in the respective decimal fraction portion of F strings. Clearly, mixing input digits in the password generation process proposed in Fig. 5 should help to level it properly.

The upper plot indicates the probability of finding successfully $s = 1$ times the specific digit $d \in [0, 9]$ in each analyzed dataset in $r = 9$ attempts, given the probability p (frequency f) of the digit in the file $p = f(d)$. As predicted by the negative binomial distribution formula $NB(r = 9, s = 1, p = f(d))$ [54] each digit $d \in [0, 9]$ is equally likely

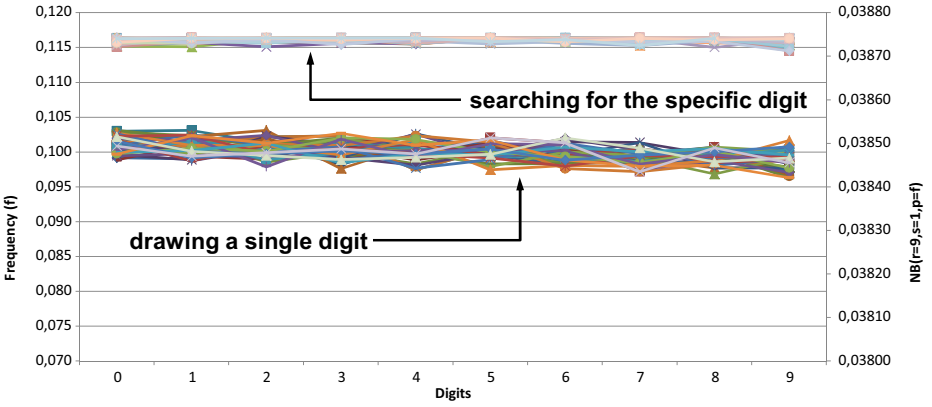

Fig. 10 Statistical properties of strings of digits in 39 preprocessed FIT files

to be found. However, when experimenting with the number of attempts $r < 9$ a slight skew at both extremes of the [0, 9] range was observed. This observation confirms the validity of the approach to password generation proposed in Fig. 6 – by choosing a high enough value r of digits searched to assemble the ASCII95 code-point their uniformity could be improved.

4.2.2 Password guessability

It is interesting to see if the passwords listed in Tables 1–3 have some patterns or repetitions that might lower their entropy compared to the ideal upper bound H_{sh} calculated for 32 character passwords in Section 1. Of course, these patterns (if any) would not be of the same nature as patterns that could be found in passwords made by people – including dictionary words or spatial patterns, mangled or recoded phrases, or some combination thereof. Nevertheless, we used two common techniques to assess guessability of our passwords in the typical context of use of password encrypted documents. One involves password strength meters often incorporated in popular document writing and authoring tools to aid users in assessing the strength of passwords considered for document encryption. Another considers the variety of attacks on the already encrypted document performed by *casual attackers*, who may have no hint on its password origins and may attempt customary attacks on the extracted document hash by using a program or script cycling through combinations of common words.

Password strength meters In the first experiment the popular `zxcvbn` password strength estimation tool was used as a valid representation of popular text password strength meters [57]. It models passwords as consisting of one or more concatenated patterns and may be considered a major improvement over traditional password strength estimators. Instead of counting the occurrence of special characters, mixed case characters, numeric digits, etc., it calculates a password's entropy to be the sum of its constituent patterns decreasing the total password entropy and unmatched gaps between them, which are treated as brute-force "patterns" increasing the entropy. The analysis is three-step. First, all the (possibly overlapping) patterns are detected and classified. Next, the entropy of each matched pattern is calculated independent of the rest of the password assuming pessimistically that the attacker knows the patterns that make up a password, but not necessarily how many or in which order. Finally, given the full set of possibly overlapping matches, the tool finds the simplest (of the lowest entropy) non-overlapping sequence by minimizing a certain heuristic formula including the terms measuring how many guesses would be required in the worst case, i.e. when the attacker knows the number of patterns in the sequence but not their order, and additionally, does not know the length of the pattern sequence. The tool does not model interdependencies between patterns, such as common phrases and other collocations, which based on the development of Section 3 is not the case for our passwords. Results of this analysis for passwords in Tables 1–3, sorted in the order of the values of seeds, are depicted in Fig. 11.

It may be seen that the tool was able to find some components in each tested password string that lowered its total entropy from the ideal upper bound value. Also, by reducing the size of the set of symbols used to compose the passwords from ASCII95 to ASCII85 the total entropy of each password of the 'sha256 (ASCII85)' series slightly decreased. However, the minimum entropy in the tested set remains well over $H_{min} = 140$ bits, i.e. if the attacker has any knowledge on the number of patterns considered by the `zxcvbn` tool the number of guesses will still excess $2^{140} = 1,39 \cdot 10^{42}$ attempts. When compared to the

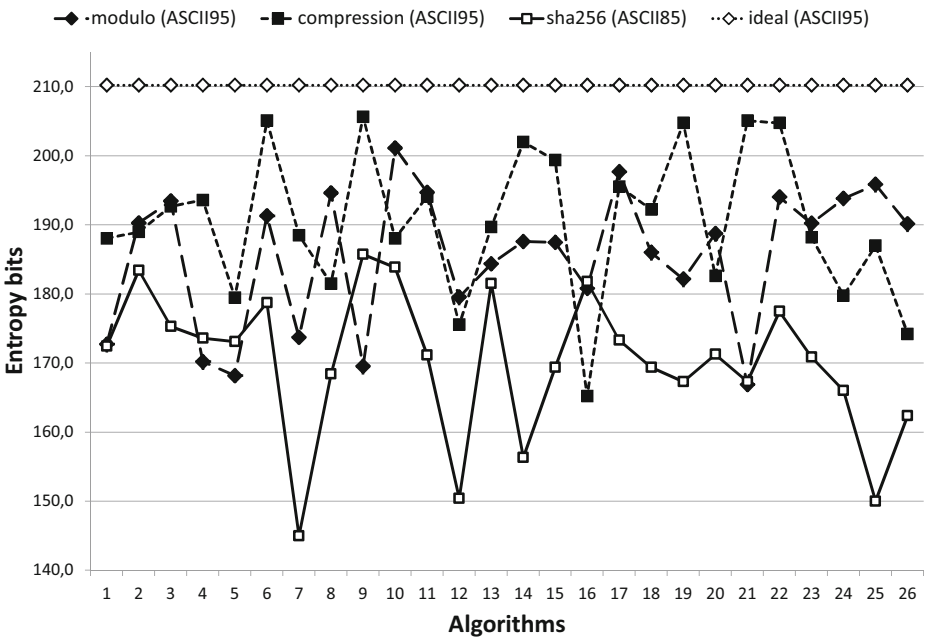

Fig. 11 Assessment of the sample passwords strength

recommended lower bound for password offline guessability estimate of 10^{14} guesses given in [13], the passwords listed in Tables 1–3 may be considered really strong.

Password cracking tools In the second experiment we used the Password Guessability Service (PGS), which offers a realistic simulation of professional guessing attacks [37]. The service can simulate several different classes of cracking algorithms by generating (guessing) candidate passwords to match the challenged one, including mangling rules to transform existing strings into new ones, composing characters using a Markov model of letters in natural language with finite automata representing password structures and setting-up probabilistic context-free grammar rules (PCFG) in which non-terminals represent contiguous substrings of a single character class of which the password may be composed of. Mangled wordlist attacks are simulated by PGS with two popular password cracking tools, John the Ripper (JTR) [48] and HashCat (HC) [50]. They both need an input wordlist (stolen passwords and dictionaries) and the set of predefined rules. JTR iterates through the entire wordlist using one mangling rule before proceeding to the subsequent rule, whereas HC iterates over all mangling rules for the first wordlist entry before continuing to the subsequent entry. Markov [28] and PCFG [56] based attacks require a prior analysis of the input wordlist, treated as training data, to properly weight entries or respectively assign probabilities to strings of letters based on their occurrence in the input set. Implementation of these two techniques requires significantly more CPU and RAM resources than JTR and HC. Therefore fewer guesses could be realistically attempted for more complex passwords. The volume of training data used by PGS includes three sets of plaintext passwords that are available online and three dictionary sources of nearly 50 millions of words in total, for which several different passwords-composition policies have been defined. They classify password generation rules with regard to the minimum password length and the number of

character classes occurring in the password (lowercase letters, uppercase letters, digits and symbols).

The passwords listed in Section 3.2 were submitted to PGS for guessability analysis with passwords in Table 1 classified as '4class8' (at least eight characters of all four character classes) and respectively in Tables 2 and 3 as '3class16' (at least 16 characters of at least three character classes). In about two weeks after submission guess exhaustion was reported by the PGS administrator with a short note that the JTR module performed on the order of $1.7 \cdot 10^9$ guesses and the Markov module performed on the order of $2.2 \cdot 10^9$ guesses. Not a single password submitted for evaluation was cracked.

4.3 Scoping assessment of FYEO security

Given the threat scenarios specified in Section 2.2 and the results of experiments reported in this section the overall security of the FYEO scheme may be finally assessed. We will argue that FYEO can effectively counteract the widespread tendency of users who often prioritize personal comfort over security. By being able to use passwords extremely difficult to remember and type they can achieve a high level of security in combination with highly personalized and user-friendly access control. Moreover, the biometric verification mechanism conforming to requirements R1-R5 discussed before is self-contained and does not require the introduction of any extra solutions to the underlying operating system of the user's device.

The steps and activities that the adversary may take to succeed are indicated in Fig. 12. They address only those aspects of document security that are directly related to the mechanisms introduced by FYEO. In particular, we assume that all customary security precautions have been properly taken by the receiving device owner/administrator, as the FYEO scheme is not intended to compensate for their lack or replace the existing ones. It is assumed that the user's device is protected against the hacking of its camera or screen to prevent interception of the user photo or the document content when opened on the device, and the FYEO application code is obfuscated to prevent interception of all internally generated data, including the password. The extent of work that attackers need to undertake to finally break all FYEO protections is classified below with regard to the degree of dedication of each individual attacker and the resources at his/her disposal.

Casual attackers may intercept in step #1 the FYEO bundle with the encrypted document while on transfer, but without being aware of the purpose of the attached FIT file. Then, most likely in step #2, he/she may attempt usual brute force or dictionary attacks on the password hash extracted from the document. Document protection would rely then solely on the hard to crack text password. Even when the interception occurred on the recipient's mail server (or somehow the attacker was able to establish the recipient's identity in step #3), other types of attacks still would not be possible unless the adversary got access to the password generation algorithm. That requires stealing or hacking the recipient's device by a highly motivated 'committed attacker'.

Committed attackers may succeed in tracking down the device after establishing the identity of the owner of the said device in step #4. If properly secured, however, the code of the *decrypting application* (see Fig. 3) would not generate the password without the prior positive biometric verification of the impostor. Therefore in step #4 the latter would have to attempt a *sensor attack* by using whatever general photos of the legitimate user (of the now known identity) he/she could find on the Internet or on the stolen device. The attack

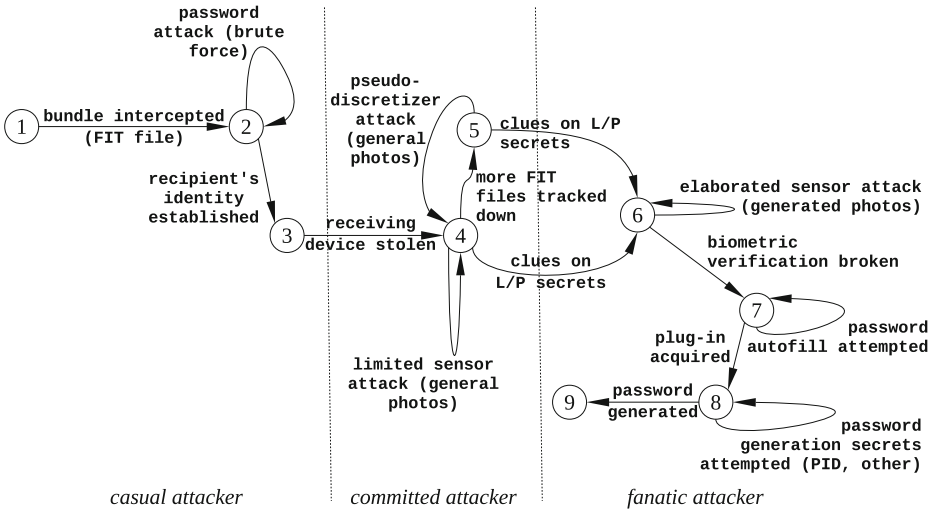

Fig. 12 FYEO document lines of defense

would be limited, since the number of such general photos would not suffice to break the illumination and pose 'barriers' embedded in the FIT file – if only the FAR/FRR rate was set properly as explained in Section 4.1. A 'committed attacker' could also attempt to find or intercept other original FIT files of the recipient (if any exist) of the previously established identity – or FIT files with similar content of similar looking people – to carry out a sort of the discretizer attack by attempting the respectively extended set of general photos for all intercepted FIT files in step #5. This attack shall be considered a 'pseudo-discretizer' attack, as the adversary would be able to check the set of all collected photos only against the intercepted FIT files; observing how the output characteristics of these files can be affected by the input photos would not be possible, as this feature is provided only by the enrollment point. Moreover, the password generation algorithm indicated by the AID code in the intercepted bundle can be properly generated only for the original FIT file, not its similar counterparts intercepted elsewhere. If during the experiments with multiple FIT files of the same person of known identity or the collected general photos or both, the adversary can possess or discover any fragmentary knowledge about the likely conditions of lighting or pose of the subject he/she may attempt generating images in step #6.

Fanatic attackers may exhibit extreme dedication and attempt the most recent techniques based on the *deep generative models* [41]. Although quite effective in terms of the overall performance of the generation process and quality of the produced images, these models need training sets with some essential 'ground truth' on the subject. The more the images from this training set deviate from the images registered at the enrollment point (unknown to the attacker), the more difficult the task of their generation will be. Note that all this time it will be a sensor attack, without the possibility of an effective assessment by the adversary whether the generated modifications to the input general images are getting closer to the intercepted FIT file. For that reason, the iteration in step #6 would resemble a dictionary attack on a text password, where instead of one-dimensional problem of generating variant text strings with a certain linear structure, the attacker would face a two-dimensional problem of generating variant images with a certain spacial composition of pixels. Undoubtedly,

the latter task will be at least as difficult as the former one, despite the use of the most efficient methods of generating artificially modified versions of input images. Finally, the ultimate protection against attacks using any acquired or generated images in steps #4, #5 and #6 in Figure 12 will be adding to the decrypting application a *liveness verification* module, making such attacks infeasible [46]. If, however, the attacker could still break all the barriers described above and somehow managed to pass the biometric verification process in step #7 the FYEO document has yet several more lines of defense. The first one may be a missing plug-in with the required password generation algorithm, kept by the user separately from his/her stolen device on a memory card. Then the adversary may attempt reconstructing the missing algorithm code. By the argument in Section 2.2.1 it would be rather a hopeless task due to the fact that one may write infinitely many correct programs to generate passwords based on the same (good enough) input entropy source. Therefore more reasonable action would be to generate the passwords directly, not the algorithms for generating them. Thanks to the password complexity, despite being able to pass biometric verification, the attacker will practically be again at step #2 in Fig. 12. On the other hand, if the plug-in could be acquired after all (e.g. stolen with the device), the generation of the proper password would still require additional knowledge of the user PID and other confidential data related to checking validity of the AID code. The ultimate way to break this FYEO document line of defense in step #8 may be to terrorize the user by the attacker. Such scenarios are, however, beyond the scope of this paper.

5 Related work

The passwordless approach introduced by the FYEO scheme draws on three concepts that until today have seen numerous practical implementations in the area of user authentication and which are still the subject of intensive research. They are password managers, capable of generating and retrieving complex passwords that need not be memorized by users, uncertain biometric data as high entropy sources for generating strong passwords and facial recognition unlocking systems providing a very convenient input mechanism for user credentials in keyboardless personal devices.

Below, we review the current state of the art of these concepts in more detail with regard to the findings reported in the paper and assess their significance to improving offline security of electronic documents.

5.1 Password managers

There is a rich variety of password managers to choose from today [23]. They may be stand alone programs, browser extensions, online services accessed through website portals, as well as locally accessed hardware devices that serve as keys. Their detailed survey is well beyond the scope of this paper, but it is worth summarizing briefly their core features in order to contrast them with the password generation capability provided by our FYEO scheme. They can generate on demand complex, unique and random password strings that may be stored afterwards in an encrypted database for retrieval. For that purpose, they typically require a user to generate and remember one “master” password to unlock and access the database, which can either be stored locally on the user’s device or stored remotely through an online file-hosting service. In the latter case the saved password database can be synced to the cloud to allow users to access their various site-specific passwords from different personal devices. The generative subclass of password managers provide the users

with an algorithm to reproduce passwords, so no database to store them is needed [31]. This is also the case of the FYEO scheme.

Typically a generative password manager requires only one secret to be provided as input by the user when generating password strings. The secret may be the master password, a graphical password or a digital object. The choice of how that secret is used to generate site-specific passwords is the primary difference between generative password managers. For example, the PwdHash generator [42] can generate site-specific passwords by combining a long-term user master password and data associated with the web site. The master password is used as the salt, allowing for generation of multiple passwords for one website, or the use of unique passwords for multiple accounts on one domain. The ObPwd generator [6] generates site-specific passwords as a function of a site-specific digital object together with optional parameters, including a long-term master password as a salt and the Web site URL. A digital object could be a file chosen by the user from a collection of images, music or videos and stored locally or accessed remotely, and constitutes a secret shared by the parties allowed to access the password protected entity. PALPAS [18] generates passwords complying with site-specific requirements using server-provided password policy data specified formally with password definition rules, a stored secret master password and a user-specific secret value as the salt, which is synchronized across all user devices using that server. AutoPass [30] can automatically generate passwords in a site-specific form and store password generation data on a dedicated server as part of the user-specific configuration data for that user and can synchronize the preserved passwords.

When applied to password protected documents, the aforementioned automatic password generation mechanisms requiring authentication servers could work for document centric business processes involving downloadable documents, but would be less practical when such processes are implemented using email attachments. For the latter, opening password protected documents each time they migrate to another device will generate additional network traffic, put extra overhead on the business process and unnecessarily attract the attention of potential attackers. On the contrary, implementation of the FYEO scheme does not require any external server, as all data needed to regenerate the password for opening the incoming document are in the FYEO bundle.

5.2 High entropy sources

The concept of using a user-selected digital object introduced by ObPwd and AutoPass has several advantages. One is convenience for users, who instead of creating and recalling text passwords satisfying complex composition rules can select and recall an object they are familiar with, like personal photos or favored music pieces. Another is that no long-term master passwords need to be stored, which might be easier to track down by potential attackers than one of many inconspicuously looking files in the user directory. Passwords generated based on the content of a digital object could certainly be more resistant to dictionary attacks since their numeric content representing pixels or signal samples could reasonably be considered a high-entropy source. Finally, the user may keep safe the password-generating object on a separate memory card or have online access to it, so a single object can provide for password regeneration to enable access from anywhere. However, ObPwd generates passwords from the user-selected file by computing a hash of its content without any deeper analysis of its statistical properties before qualifying it for the generation process like in the manner presented by us earlier in this paper.

It is widely believed that physical processes can provide a source of unpredictable numbers, so if the output is large enough strong passwords could be generated. In [6] it was

estimated that ObPwd would need at least 160 bytes of such output under the pessimistic assumption that on average, each byte would provide at least one bit of entropy. Typically the volume of such data would be much higher and ObPwd assumes chunks of data of up to 100 KB for performance reasons. However, a more thorough analysis of the underlying process semantics may be required if the same password shall be generated reproducibly, i.e. each time the user would like to use it again. A good candidate for that are biometric objects, providing streams of high entropy data and at the same time enabling biometric identity verification of the related subjects. Examples include physiological signals [39], brain signals (EEG) [12] or electrocardiograms (ECG) [22]. One problem with the required reproducibility of the password generation process based on such "internal" biometrics modalities is making it immune to noise and abnormal signal limitations. Besides that, acquiring EEGs or ECGs for the decryption process outlined in Fig. 3 would be cumbersome, if specialized devices and electrodes attached to the subject's body are required to pass authentication. Contrary to that, authentication performed by FYEO is contactless and yet involves data produced by a physical process filtered out with a complex non-linear filter.

5.3 Facial recognition unlocking systems

From the user point of view the proposed FYEO scheme has many similarities to existing biometric mechanisms for unlocking mobile devices, including the notable Apple's Face ID on iPhones and iPads or, more recently, Google's Face Unlock system on Pixel 4 smartphones. This is because mobile devices, like FYEO documents on transfer, may be stolen and thus exposed to offline attacks. Each mechanism utilizes the available camera system provided by the personal device to capture images of verified subjects. Also they both enable passwordless verification of users without any external server support and may be considered a very attractive alternative to typing complex passwords on small touchscreens of mobile devices. Putting aside the obvious advantage of making the user's face a password capable of unlocking the device and selected applications installed on it, a few comments on security and portability of the aforementioned mechanisms would be in place to compare them to our solution.

When it comes to resistance to offline attacks it is worth noting that the device may freely chose between biometric or password based identity verification. Typically, a biometrically protected device would require a password after a number of unsuccessful scans of the subject, after restarting it or after a prolonged period of inactivity. The same policy may be implemented by the FYEO decrypting application in Fig. 3 in case of device theft with the encrypted document, as described in Section 2.2.2. However, if the intercepted document is directly attacked by the adversary its only protection is the password. Unlike the user-provided passwords protecting personal devices, automatically generated FYEO passwords discussed in Section 3.2 would certainly be much harder to crack.

Biometric protection of the personal device may take advantage of the specially designed camera system. One example is Apple's Face ID, which uses an infrared dot projector and an infrared camera to read the dot pattern reflected off surfaces at various depths to create a 3D model of the subject's face. This 3D technology is widely believed to be very secure and highly accurate and can largely limit the risk of the system being bypassed using photos, compared to the 2D facial authentication based on facial data captured from 2D maps of the user's face. On the other hand the less secure 2D facial recognition unlock technology may be reinforced with a secure iris scanner, as in the Intelligent Scan feature offered by Samsung's Galaxy S9 and S9 Plus smartphones. Yet another solution can help a regular camera device with a light beam emitted by the built-in flash LED to distinguish a captured

facial image of the subject from a regular 2D photo by detecting corneal glint induced by the beam [61].

The biometric verifier of the FYEO scheme can take advantage of so equipped personal devices, if only their related operating system provides appropriate libraries for the on-device user authentication needs, for example the Biometric API of the AndroidX Biometric Library. In that case, instead of the FIT file any binary object could be inserted in the FYEO bundle to provide the source of entropy for generating the required password. However, it would be less safe, as the FIT file content is intrinsically related to the identity of the legitimate recipient. Note that when sending the document to a specific email address, the sender may not be sure that the person accepted by the remote face unlock system is the right one (e.g different users sharing the same account). Besides, the FYEO scheme in the form proposed in this paper provides for better portability of the related decrypting application, as it is less dependent on the actual camera system of the receiving device.

Finally, a common feature of the face unlocking systems of modern smartphones and the proposed FYEO scheme is the complete independence from the user identity verification process of any external server, which allows users to work offline. For that purpose the biometric verifier will need the facial data of the legitimate user to be stored locally. Face unlock systems of mobile devices protect that data by storing them in an encrypted form in their dedicated "secure enclave" piece of hardware. The data constitute a mathematical representation of key details of the face, which may gradually evolve as the system is used and re-trained to follow typical variations in a user's appearance. For security reasons the facial data must not suffice to reconstruct a person's original image, as discussed in Section 3.1. One of the biggest concerns after Face ID was released by Apple was the possibility that the user would unknowingly unlock the phone in the immediate vicinity of the camera. This was later rectified by requiring the user to take a proper pose in front of the camera. In that regard the proposed FYEO scheme requiring the specific pose and illumination of the user to unlock the encrypted document subsumes that Face ID refinement. Moreover, the FYEO scheme does not need any "secure enclave" to keep the FIT file for processing, since neither the original image of the subject nor the password generation algorithm can be reconstructed from its content.

6 Conclusion

Although the approach to ensuring offline security of digital content proposed in the paper incorporates several well-known concepts in the field of user authentication discussed before, their combination in the FYEO scheme for protecting electronic documents is innovative. It provides contactless authentication of legal users of the document based on their high entropy biometric data, for which extremely strong passwords can be automatically generated, a feature not yet available in the authoring and document exchange tools. Moreover, the entire authentication process is truly offline, as no external server or database are needed.

Implementability of the FYEO scheme was assessed by us with two models of active document architectures. One implementation was based on the *Interactive Open Document Architecture (IODA)* [44] architecture, developed to augment the static content of digital documents with data and services that could turn them into executable papers, suitable for implementing reproducible research scenarios. The IODA executable paper is a reactive document, which can respond to activities initiated by its user. User scenarios requiring the

original document to be partially encrypted were exercised, with portions made available to selected users. For that, extracting of the FIT file (see Fig. 3) was implemented as one of document services. This solution could improve its security, as detaching the FIT file would require specialized software available only at the legitimate client's side. Another implementation involved the *Mobile Interactive Document Architecture (MIND)* [15] enabling proactive document attachments to migrate in a distributed system on their own by using standard email services. The MIND document is proactive and can initiate activities requiring reaction of its recipient. It embeds a migration path, which defines a business process involving collaborators in a virtual organization and combines a passive content with active services that support interaction of documents with users and their personal devices. The passive content of MIND documents may be password encrypted. As in the case of IODA, extraction of the FIT file from the bundle embedded in the MIND document was implemented as the document's service.

We continue our work towards a more secure transfer of electronic documents exchanged in collaborative processes by incorporating effective methods for determining liveness of the subject in front of the camera [46] combined with identifying various emotional states of the subject [24] developed in another project to make the FYEO scheme resistant to physical attacks on users and to extend the catalogue of factors affecting tuning of the FAR/FRR rates by facial expressions. Another issue to be addressed is extension of biometric verification to lighting conditions for outdoor use. The candidate techniques involve deep neural networks (DNN) [17, 51]; they seem to offer mechanisms to make some of their general classes irreversible and suitable for generating pseudo-identities, as required by the ISO24745 standard [3, 4]. Nevertheless, as observed by us in the experiments reported in Section 4 the state-of-the-art biometric recognition techniques do not necessarily comply with the irreversibility and unlinkability requirements for biometric verifiers [16].

Declarations

Conflict of Interests The authors declare that they have no conflict of interest.

References

1. Adini Y, Moses Y, Ullman S (1997) Face recognition: the problem of compensating for changes in illumination direction. IEEE Trans Pattern Anal Mach Intell 19(7):721–732. https://doi.org/10.1109/34.598229
2. Apple Inc. Manage passwords using keychains on Mac. https://support.apple.com/guide/mac-help/use-keychains-to-store-passwords-mchlf375f392/mac. Accessed: 2019-12-31
3. Arora S, Liang Y, Ma T (2015) Why are deep nets reversible: A simple theory, with implications for training. CoRR arXiv:1511.05653

4. Behrmann J, Grathwohl W, Chen RTQ, Duvenaud D, Jacobsen J-H (2019) Invertible residual networks. In: Chaudhuri K, Salakhutdinov R (eds) Proc. 36th Int. Conf. on Machine Learning, vol 97. PMLR, Long Beach, pp 573–582. http://proceedings.mlr.press/v97/behrmann19a.html
5. Belhumeur PN, Hespanha JP, Kriegman DJ (1997) Eigenfaces vs. Fisherfaces: recognition using class specific linear projection. IEEE Trans Pattern Anal Mach Intell 19(7):711–720. https://doi.org/10.1109/34.598228
6. Biddle R, Mannan M, van Oorschot PC, Whalen T (2011) User study, analysis, and usable security of passwords based on digital objects. IEEE Trans Inf Forensic Secur 6(3):970–979
7. Billa JB, Nawar A, Shakil MMH, Das AK (2019) PassMan: A new approach of password generation and management without storing. In: Proc. 7th Int. Conf. on Smart Computing Communications (ICSCC), pp 1–5
8. Bishop M (2018) Computer security, 2nd edn. Addison-Wesley Professional
9. Breebaart J, Busch C, Grave J, Kindt E (2008) A reference architecture for biometric template protection based on pseudo identities. In: Proc. Special Interest Group on Biometrics and Electronic Signatures (BIOSIG 2008), pp 25–38
10. Ding L, Ding X, Fang C (2012) Continuous pose normalization for pose-robust face recognition. IEEE Signal Process Lett 19(11):721–724. https://doi.org/10.1109/LSP.2012.2215586
11. Eastlake 3rd D, Schiller J, Crocker S (2005) Randomness requirements for security. RFC 4086, RFC Editor. https://doi.org/10.17487/RFC4096
12. Eswaran C, Palaniappan R, Phon-Amnuaisuk S, Ravi KVR (2007) Data encryption using event-related brain signals. In: 2007 Int. Conf. on Computational Intelligence and Multimedia Applications(ICCIMA), vol 1, pp 540–544
13. Florêncio D, Herley C, van Oorschot PC (2014) An administrator's guide to internet password research. In: Proc. 28th Large Installation System Administration Conf. (LISA14). USENIX Association, Seattle, pp 44–61
14. Georghiades AS, Belhumeur PN, Kriegman DJ (June 2001) From few to many: Illumination cone models for face recognition under variable lighting and pose. IEEE Trans Pattern Anal Mach Intell 23(6):643–660
15. Godlewska M, Wiszniewski B (2010) Distributed MIND – a new processing model based on mobile interactive documents. In: Proc. 8th Int. Conf. PPAM 2009, LNCS, vol 6068. Springer, pp 244–249
16. Gomez-Barrero M, Galbally J (2020) Reversing the irreversible: A survey on inverse biometrics. Comput Secur 90:101700. https://doi.org/10.1016/j.cose.2019.101700
17. Guo G, Zhang N (2018) What is the challenge for deep learning in unconstrained face recognition? In: 13th IEEE Int. Conf. on Automatic Face Gesture Recognition (FG 2018), pp 436–442
18. Horsch M, Hülsing A, Buchmann J (2015) PALPAS - PAsswordLess PAssword Synchronization. In: Proc. 10th Int. Conf. on Availability, Reliability and Security, ARES '15. IEEE Computer Society, Washington, pp 30–39
19. Hosseinzadeh S, Rauti S, Lauren S, Mäkelä JM, Holvitie J, Hyrynsalmi S, Leppänen V (2018) Diversification and obfuscation techniques for software security: A systematic literature review. Inf Softw Technol 104:72–93
20. ISO/IEC 24745 (2011) Information technology – Security techniques – Biometric information protection. Standard ISO/IEC 24745:2011(en), International Organization for Standardization, Geneva. https://www.iso.org/standard/52946.html
21. Jeong B, Vallat A, Csikszentmihalyi C, Park J, Pacheco D (2019) MementoKey: Keeping passwords in mind. In: Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems, CHI EA'19, pp LBW1316:1–LBW1316:6
22. Karimian N, Guo Z, Tehranipoor M, Forte D (2017) Highly reliable key generation from electrocardiogram (ECG). IEEE Trans Biomed Eng 64(6):1400–1411
23. Kissell J (2019) Take control of your passwords, 3rd edn. Take Control Books
24. Kołakowska A, Landowska A, Anzulewicz A, Sobota K (2017) Automatic recognition of therapy progress among children with autism. Sci Rep 7(1):1–14
25. Lades M, Vorbruggen JC, Buhmann J, Lange J, von der Malsburg C, Wurtz RP, Konen W (1993) Distortion invariant object recognition in the dynamic link architecture. IEEE Trans Comput 42(3):300–311
26. Lyons M, Akamatsu S, Kamachi M, Gyoba J (1998) Coding facial expressions with Gabor wavelets. In: Proc. 3rd. Int. Conf. on Face & Gesture Recognition, Nara, pp 200–205
27. M. S, Rätsch G, Weston J, Schölkopf B, Müller KR (1999) Fisher discriminant analysis with kernels. In: Neural Networks for Signal Processing IX: Proc. 1999 IEEE Signal Processing Society Workshop, pp 41–48

28. Ma J, Yang W, Luo M, Li N (2014) A study of probabilistic password models 2014 IEEE Symp. on Security and Privacy. IEEE Computer Society, San Jose, pp 689–704
29. Maclean R, Ophoff J (2018) Determining key factors that lead to the adoption of password managers. In: 2018 Int. Conf. on Intelligent and Innovative Computing Applications (ICONIC), pp 316–322
30. Maqbali FA, Mitchell CJ (2017) AutoPass: An automatic password generator. In: 2017 Int. Carnahan Conf. on Security Technology (ICCST), pp 1–6
31. Maqbali FA, Mitchell CJ (2016) Password generators: Old ideas and new. In: Proc. 10th IFIP WG 11.2 Int. Conf. WISTP 2016, LNCS, vol 9895. Springer, Heraklion, pp 245–253
32. Marky K, Mayer P, Gerber N, Zimmermann V (2018) Assistance in daily password generation tasks. In: Proc. of the 2018 ACM Int. Joint Conf. and 2018 Int. Symp. on Pervasive and Ubiquitous Computing and Wearable Computers, UbiComp'18, pp 786–793
33. MathWave Technologies (2017) Easyfit – distribution fitting made easy. Dnepropetrovsk, Ukraine. http://www.mathwave.com/en/home.html. [Software]
34. Meng-Hui L, Pong CY (2016) Entropy measurement for biometric verification systems. IEEE Trans Cybern 46(5):1065–1077
35. Nandakumar K, Jain AK (2015) Biometric template protection: Bridging the performance gap between theory and practice. IEEE Signal Proc Mag 32(5):88–100
36. Ojala T, Pietikainen M, Harwood D (1996) A comparative study of texture measures with classification based on featured distributions. Pattern Recogn 29(1):51–59
37. PGS (2019) The Carnegie Mellon University Password Research Group's Password Guessability Service. https://pgs.ece.cmu.edu
38. Phillips PJ, Moon H, Rizvi SA, Rauss PJ (October 2000) The FERET evaluation methodology for face-recognition algorithms. IEEE Trans Pattern Anal Mach Intell 22(10):1090–1104
39. Poon CY, Zhang YT, Bao SD (2006) A novel biometrics method to secure wireless body area sensor networks for telemedicine and m-health. IEEE Commun Mag 44:73–81
40. Ratha NK, Chikkerur S, Connell JH, Bolle RM (2007) Generating cancelable fingerprint templates. IEEE Trans Pattern Anal Mach Intell 29(4):561–572. https://doi.org/10.1109/TPAMI.2007.1004
41. Razavi A, van den Oord A, Vinyals O (2019) Generating diverse high-fidelity images with VQ-VAE-2. CoRR arXiv:1906.00446
42. Ross B, Jackson C, Miyake N, Boneh D, Mitchell JC (2005) Stronger password authentication using browser extensions. In: Proc. 14th Conf. on USENIX Security Symp., SSYM'05. USENIX Association, Berkeley, pp 17–31
43. Selvakumar AL, Ganadhas CS (2009) The evaluation report of SHA-256 crypt analysis hash function. In: 2009 Int. Conf. on Communication Software and Networks, pp 588–592
44. Siciarek J, Wiszniewski B (2011) IODA - an interactive open document architecture. Procedia Comput Sci 4:668–677
45. Sim T, Baker S, Bsat M (2002) The CMU pose, illumination, and expression (PIE) database. In: Proc. 5th IEEE Int. Conf. on Automatic Face and Gesture Recognition, FGR '02, pp 53–58
46. Smiatacz M (2012) Liveness measurements using optical flow for biometric person authentication. Metrol Measur Syst 19(2):257–268
47. Smiatacz M (2013) Eigenfaces, fisherfaces, laplacianfaces, marginfaces – how to face the face verification task. In: Proc. 8th Int. Conf. on Computer Recognition Systems, CORES 2013. Springer, Heidelberg, pp 187–196
48. Solar Designer (2013) John the ripper. Openwall Project. http://www.openwall.com/john/. [Software]
49. Srivastava S, Sivasankar M (2016) On the generation of alphanumeric one time passwords. In: Proc. Int. Conf. on Inventive Computation Technologies (ICICT), vol 1, pp 1–3
50. Steube J (2018) Hashcat. Hashcat Project. https://hashcat.net/oclhashcat/. [Software]
51. Taigman Y, Yang M, Ranzato M, Wolf L (2014) DeepFace: Closing the gap to human-level performance in face verification. In: Proc. 2014 IEEE Conf. on Computer Vision and Pattern Recognition, CVPR '14. IEEE Computer Society, Washington, pp 1701–1708
52. Tan X, Triggs B (2010) Enhanced local texture feature sets for face recognition under difficult lighting conditions. IEEE Trans Image Process 19(6):1635–1650
53. Viola P, Jones MJ (2004) Robust real-time face detection. Int J Comput Vis 57(2):137–154
54. Walkenbach J (2015) Excel 2016 bible. Wiley
55. Wang Y, Vangury K, Nikolai J (2014) Mobileguardian: A security policy enforcement framework for mobile devices. In: Proc. 2014 Int. Conf. on Collaboration Technologies and Systems (CTS), pp 197–202
56. Weir M, Aggarwal S, Medeiros B, Glodek B (2009) Password cracking using probabilistic context-free grammars. In: 2009 IEEE Symp. on Security and Privacy. IEEE Computer Society, Washington, pp 391–405

57. Wheeler DL (2016) zxcvbn: Low-budget password strength estimation. In: 25th USENIX Security Symp. USENIX Association, Austin, pp 157–173
58. Whittlesey MA (2019) Spherical geometry and its applications, 1st edn. Chapman and Hall/CRC
59. Yoon S, Jeon Y, Kim J (2015) Mobile security technology for smart devices. In: Proc. 2015 Int. Conf. on Information and Communication Technology Convergence (ICTC), pp 1171–1173
60. Zhang W, Zhao X, Morvan J, Chen L (2019) Improving shadow suppression for illumination robust face recognition. IEEE Trans Pattern Anal Mach Intell 41(3):611–624. https://doi.org/10.1109/TPAMI.2018.2803179
61. Zhao Y, Schneiderman HW, Andrew SM (2017) Facial recognition (U.S. Patent 8411909B1). published Apr. 2013, assigned to Google LLC