A Framework for Security and Risk Analysis of Enrollment Procedures: Application to Fully-remote Solutions based on eDocuments

Marco Pernpruner[1,2][a], Giada Sciarretta[1][b] and Silvio Ranise[1,3][c]

[1]*Security & Trust Research Unit, Fondazione Bruno Kessler, Trento, Italy*

[2]*Department of Informatics, Bioengineering, Robotics and System Engineering, University of Genoa, Genoa, Italy*

[3]*Department of Mathematics, University of Trento, Trento, Italy*

Keywords: eDocuments, Enrollment, Onboarding, Risk Analysis, Security Analysis, Security Framework.

Abstract: More and more online services are characterised by the need for strongly verifying the real-world identity of end users, especially when sensitive operations have to be carried out: just imagine a fully-remote signature of a contract, and what could happen whether someone managed to perform it by using another person's name. For this reason, the identity management lifecycle contains specific procedures – called *enrollment* or *onboarding* – providing a certain level of assurance on digital users' real identities. These procedures must be as secure as possible to prevent frauds and identity thefts. In this paper, we present a framework composed of a specification language, a security analysis methodology and a risk analysis methodology for enrollment solutions. For concreteness, we apply our framework to a real use case (i.e., fully-remote solutions relying on electronic documents as identity evidence) in the context of a collaboration with an Italian FinTech startup. Beyond validating the framework, we analyse and highlight the essential role of mitigations on the overall security of enrollment procedures.

1 INTRODUCTION

Verifying a person's real identity has always been a risky operation, considering that paper-based identity documents could be counterfeited with little or even no possibility of detection. This is one of the main reasons why electronic documents (also called eDocuments) have been created: these documents are equipped with contactless chips that can be read wirelessly, and in some cases they are also capable of performing cryptographic operations. In addition, eDocuments are tamper-resistant and provide a high level of assurance on the attested data. As a result, many companies that previously required customers to physically visit a branch to perform some sensitive operations (such as opening a new bank account or signing a contract) can now leverage the intrinsic security assured by eDocuments to provide fully-remote procedures and thus improve the usability on their clients' side. Although NIST defines these procedures as *enrollment* (Grassi and Fenton, 2017), they are commonly referred to as *onboarding*, especially when they lead to customer acquisitions.

However, simply relying on eDocuments' security is not enough: enrollment procedures must be sturdily designed and adopt suitable security mechanisms to prevent impersonation attacks. Moreover, such procedures – as well as physical ones – are subject to many guidelines and regulations that have been internationally issued to prevent frauds and identity thefts (e.g., *Anti Money-Laundering* in the financial sector).

In this paper, we present a framework for security and risk analysis of enrollment procedures, that is based on the following components:

- *Specification Language*: starting from the analysis of on-the-market solutions, we have derived a list of basic actions that are commonly involved in enrollment procedures. In order to allow an easy description and analysis of the procedures, we describe them by means of a simple specification language with precise semantics;
- *Security Analysis Methodology*: we describe a security analysis methodology for enrollment procedures by defining the concept of *identification factors* (that we have borrowed and adapted from

[a] https://orcid.org/0000-0001-8936-2726
[b] https://orcid.org/0000-0001-7567-4526
[c] https://orcid.org/0000-0001-7269-9285

Pernpruner, M., Sciarretta, G. and Ranise, S.
A Framework for Security and Risk Analysis of Enrollment Procedures: Application to Fully-remote Solutions based on eDocuments.
DOI: 10.5220/0010554502220233
In *Proceedings of the 18th International Conference on Security and Cryptography (SECRYPT 2021)*, pages 222-233
ISBN: 978-989-758-524-1

the authentication context), threat model and security goal;

- *Risk Analysis Methodology*: we describe a risk analysis methodology to provide a prioritization of the attackers detected during the security analysis according to their severity.

Our framework can be generally adopted to describe and analyse any enrollment procedure, by simply adjusting – if needed – the set of actions involved (i.e., the specification language) and the capabilities of the attackers on these actions (i.e., the threat model). For concreteness, we have applied the framework in the context of a realistic use case (i.e., fully-remote solutions relying on eDocuments), where we have also analysed the role of security mitigations on the overall security of enrollment procedures.

Structure of the Paper. Section 2 provides some essential concepts to understand what follows; Section 3 identifies our requirements; Section 4 defines our framework and its components; Section 5 discusses some security mitigations that we have identified; Section 6 shows the application of the framework to a real use case and highlights the essential role of security mitigations; Section 7 discusses some state-of-the-art solutions considered throughout this work and Section 8 finally draws conclusions.

2 BACKGROUND

Enrollment. The National Institute of Standards and Technology (NIST) has recently released the *Special Publication 800-63* suite (NIST, 2017) dealing with digital identity. According to the definitions in (Grassi et al., 2017b; Grassi and Fenton, 2017), the *enrollment* can be considered as the procedure allowing users to obtain a credential from a specific service, after the verification of their identity. An enrollment flow is composed of three main parts: (i) *Resolution*: the service collects the personal information of the applicant and some identity evidence (e.g., a digital copy of an identity document through a camera or a scanner); (ii) *Validation*: the service verifies the genuineness of the identity evidence provided (e.g., by querying authoritative sources or visually checking the documents), thus obtaining guarantees on the identity information; (iii) *Verification*: the service checks that the identity evidence belongs to the applicant (e.g., by collecting a picture of her and comparing it with the one embedded in the document). All these parts are extremely sensitive and potentially exposed to many security issues: in case *Validation* and *Verification* were not properly secured, users could provide wrong or fictional personal information, altered identity evidence or even identity evidence belonging to someone else. Therefore, enrollment procedures must be properly designed to rely on cutting-edge techniques and assure that the user is really who she claims.

Electronic Documents. When dealing with paper-based identity documents, the detection of tampered or misused identity evidence is not trivial, since criminals have always found new ways to counterfeit them. For these reasons, electronic versions of identity documents have recently been produced: passports became *electronic passports* (ePassports), while *electronic identity cards* (eID cards) replaced paper-based identity cards. These eDocuments are equipped with a contactless chip that can be read wirelessly and contains the citizen's personal data; in this way, criminals would need to forge not only the printed data, but also those contained within the chip (that is much more difficult, as they are digitally signed).

As for *ePassports*, to allow only authorized people to access the personal data, the identity page contains a *machine-readable zone* (`MRZ`) that must be scanned by the reader to derive the key needed to mutually authenticate with the ePassport's chip; the reader can then interact with the ePassport in order to get the personal data. These data can be validated by checking the *Document Security Object* (`SOD`), a specific file containing the signature of all the personal data's hashes: the reader can thus verify the authenticity and integrity of the data. This procedure, currently used by customs and immigration offices at borders, is provided by the *MRTD* application (ICAO, 2015).

The MRTD application is also contained in *eID cards*, which – in addition – provide another module for online authentication and, indirectly, for identification: the *IAS ECC* application (GIXEL, 2009). In fact, given the key pair and the `X509` certificate (Cooper et al., 2008) they are equipped with, eID cards can be used in a *challenge-response* protocol: an application willing to authenticate the user can send a *challenge* (i.e., a piece of information associated with the ongoing operation) to the reader, and thus to the eID card. By signing the challenge through its private key, the eID card generates the *response*; this operation can be performed only once the user has inserted the `PIN` of the specific document. Since the eID card's public key is known, the application can verify the proper signature of the original challenge, finally authenticating the user.

Table 1: OWASP risk computation.

		— **Likelihood** →		
		Low	Medium	High
|	Low	Note	Low	Medium
Impact	Medium	Low	Medium	High
↓	High	Medium	High	Critical

OWASP Risk Rating Methodology. The Open Web Application Security Project (OWASP) has developed the *Risk Rating Methodology* (OWASP, 2018), an approach to estimate the severity of specific risks. The *risk* is defined as $likelihood \times impact$, where the *likelihood* rates the probability of exploiting a specific vulnerability, while the *impact* measures the consequences of an attacker successfully leveraging a vulnerability.

Computing the likelihood and the impact requires the identification of some suitable *factors*, which can be dependent on the considered scenario and have to be given a score ranging from 0 to 9; lower scores for likelihood factors mean *less likely*, while when dealing with the impact they denote *less serious consequences* in case the considered vulnerability was exploited. The official OWASP documentation provides some examples regarding factors, although they can be fully customised: for instance, *Ease of Discovery* may be a likelihood factor representing how easy the considered vulnerability can be discovered (where 1 = practically impossible, 3 = difficult, 7 = easy, 9 = automated tools are available for this purpose).

Once these factors have been identified and rated, the overall likelihood and impact result from the average of such scores, which are then assigned a qualitative label according to their value: Low if $value < 3$, Medium if $3 \leq value < 6$, High if $value \geq 6$.

Finally, the overall risk can be found by combining the likelihood and impact labels as in Table 1.

3 REQUIREMENTS

Enrollment procedures are now playing a key role in many sensitive contexts. In addition, the current situation connected with COVID-19 – preventing people from physically visiting public offices and company branches – further requires a way to be securely and easily identified from home, possibly by using already owned devices. One of the easier and less complex remote identification methods consists of a video call between a service operator and the person who has to be identified, with the former asking the latter for any document needed to finalise the operation. However, since human operators have to be physically involved in the call, this method may slow down the enrollment process, especially when a considerable amount of people needs to be identified in the same period. Moreover, as online services should be accessible by everyone, the enrollment procedure should be carried out as simply as possible (e.g., without requiring people to perform complex operations or to install software). From all these considerations, we can infer the following requirements that an enrollment procedure should meet. The procedure must:

R1. be carried out remotely and automatically, without human operators for identification;

R2. rely on devices that people already own;

R3. provide an adequate level of usability, thus allowing everyone to finalise it.

4 FRAMEWORK

In this section, we describe the framework that we have developed, composed of a specification language to model enrollment protocols (Section 4.1), a security analysis procedure (Section 4.2) and a risk analysis procedure (Section 4.3). Finally, we discuss the usefulness of the framework (Section 4.4).

4.1 Specification Language

The specification language aims at providing a simple and graphical way to model enrollment procedures. In this work, for the sake of concreteness, we have decided to instantiate the requirements defined in Section 3 by focusing on eDocuments, which allow secure remote identifications (satisfying *R1*). Moreover, the automatic data extraction from eDocuments enhances the overall level of usability and prevents mistakes (satisfying *R3*). However, to benefit from all these capabilities, people would need to own NFC readers enabling the interaction with the eDocuments. Instead of forcing people to buy a reader to plug into their personal computers, we require a mobile *application* (hereafter *app*) that can be used with smartphones or tablets equipped with NFC (satisfying *R2*).

Consequently, by focusing on eDocuments, we have identified a list of basic actions that are commonly involved in enrollment procedures based on such means, thus defining the specification language displayed in Table 2. When modelling an enrollment procedure, the involved actions are depicted in sequence separated by a semicolon; this is the visual representation of an *enrollment flow*, which can be defined as a finite set of actions from those in Table 2.

Some actions may require a *basic entity* as argument, which is represented between parenthesis.

Example 1. *An enrollment procedure requiring users to insert the eID card's* PIN (⌨(PIN)), *read the document through NFC* (Ⓝ(🪪)) *and finally confirm the correctness of the extracted data* (👤✓) *can be modelled as follows:*

⌨(PIN); Ⓝ(🪪); 👤✓

Although for concreteness we focus on enrollment procedures based on eDocuments, the specification language can be used with any enrollment procedure by properly extending – if needed – the set of involved basic actions.

4.2 Security Analysis

In order to perform a security analysis, we refer to the idea of *factors*. While *authentication factors* have been defined by NIST (Grassi et al., 2017a) for authentication, we are not aware of a similar notion for enrollment. To fill this gap, in this work we introduce *identification factors*. These factors are associated with the actions that the user is required to perform, though not all actions are necessarily associated with an identification factor. For instance, 🤝 is not related to any identification factor, while ⌨(PIN) is associated with {PIN}.

We define as *security goal* ($\mathcal{SG}$) the set of identification factors that should not be compromised for the enrollment procedure to be considered secure.

Example 2. *Considering the flow in Example 1, the identification factors involved are the eID card and the related* PIN. *Therefore:*

$\mathcal{SG}$ = {🪪, PIN}

A *threat model ($\mathcal{TM}$) over the identification factors* is a pair ($\mathcal{ATT}, \mathcal{C}$), where $\mathcal{ATT}$ is the set of considered attackers (detailed below) and $\mathcal{C}$ represents their capabilities. More precisely, $\mathcal{C}$ is a collection of mappings corresponding to each attacker in $\mathcal{ATT}$: given an attacker and an identification factor, $\mathcal{C}$ returns a *padlock* representing the attackers' effects on that identification factor: the *green padlock* (🔒) denotes a non-compromised factor, while compromised factors are represented by a *red padlock* (🔓). Moreover, we use an *asterisk* (🔓*) to depict a factor that is compromised indirectly (e.g., in case an attacker deceives the victim into interacting with her eDocument instead of physically stealing it).

Example 3. *Following Example 2:*

- *a* person standing behind the user *could look at the* PIN *while the user is typing* (PIN → 🔓), *but he has no effect on the eID card* (🪪 → 🔒);
- *a* malicious application *on the user's mobile device could directly intercept the* PIN *while the user is typing* (PIN → 🔓) *and deceive the user into placing the eID card near the device, thus indirectly compromising it* (🪪 → 🔓*).

To avoid missing some attacks, $\mathcal{C}$ should be defined through a worst-case approach, unless some security mitigations are implemented in the protocol; in this case, their effects must be taken into consideration when defining the capabilities in $\mathcal{C}$. For instance, a *malicious application* running on the user's mobile device should be considered able to escalate the device's privileges and obtain root access, unless some mitigations prevent this behavior or warn the user.

To identify the set of attackers for our analysis, we take inspiration from the *Enrollment and Identity Proofing Threats* defined by NIST (Grassi and Fenton, 2017): among these, we do not consider attackers belonging to the *falsified identity proofing evidence* category, as they cannot be effective due to eDocuments' tamper-resistance, nor *enrollment repudiation* that for the moment is out of the scope of this work (it deals with an already enrolled user that denies having performed an enrollment). Instead, we focus on the *fraudulent use of another's identity* category, whose attackers aim to use identity documents belonging to a different individual to complete an enrollment. Therefore, in our analyses, $\mathcal{ATT}$ is the set of the following attackers:

- *Identity Document Thief (IDT)*: steals an identity document from its legitimate owner;
- *Eavesdropping Software (ES)*: intercepts the data typed on the device (e.g., keylogger);
- *Shoulder Surfer (SS)*: obtains secrets by looking at the user inserting sensitive information;
- *Social Engineer (SE)*: exploits human gullibility and confidence in others, thus deceiving people into revealing secret information or performing actions to their advantage;
- *Malicious Application (MA)*: runs on the attacker's or the victim's mobile device. In the former case, the app can fake or alter the communication between the legitimate app and the server; in the latter case, instead, the app can both interact with the victim to obtain her personal data or pictures, and access the device's internal storage to get the needed information autonomously.

An enrollment flow *violates the security goal* $\mathcal{SG}$ *under the threat model* $\mathcal{TM} = (\mathcal{ATT}, \mathcal{C})$ iff there is an attacker (or a combination of them) in $\mathcal{ATT}$ that – given its capabilities defined in $\mathcal{C}$ – compromises (directly or indirectly) all the identification factors contained in the $\mathcal{SG}$ associated to the flow.

Table 2: Specification language for the description of the flows.

	Basic entities		
🪪	The eID card	🛂	The ePassport
🪪	An additional personal document	PIN	The PIN of the eID card
▥	The MRZ printed on the eDocuments	👤	The selfie captured by a user
	Actions		
	The user may be required to...		
🤝	agree with the app's privacy policy and terms	☰	choose the type of eDocument she wishes to use and the interaction mode
👤✎	provide some extra information that is not included in the eDocument	👤✓	check and confirm the correctness of her personal data, as extracted from her eDocument
@	insert her email address and to verify it (e.g., through a link or an OTP sent by email)	💬	insert her phone number and to verify it (e.g., through an OTP sent by SMS)
👤📷	capture a photo selfie. In case the selfie also needs to contain an additional element, this will be specified as argument	N(•)	place the element specified as argument near the device, so that the app can interact with it through NFC
👤🎥	capture a video selfie	⌗(•)	scan the element specified as argument through the device's camera
📷(•)	take a picture of the argument	⌨(•)	insert the information specified as argument

Example 4. *Following Example 3, the* malicious application *(now defined as* MA*) violates the security goal $\mathcal{SG}$ since it manages to compromise all the identification factors involved (i.e., all these identification factors are marked with a red padlock:* 🔓 *or* 🔓* *).*

Definition 1. *The* security analysis problem *for an enrollment flow under a threat model $\mathcal{TM}$ is to find all (if any) minimal subsets $ATT \subseteq \mathcal{ATT}$ so that ATT violates $\mathcal{SG}$.*

Specifically, a subset $ATT \subseteq \mathcal{ATT}$ is *minimal* iff ATT violates $\mathcal{SG}$ and, for each $ATT' \subsetneq ATT$, ATT' does not violate $\mathcal{SG}$.

Example 5. *Given Example 4, the considered flow can be violated by $\{MA\}$; moreover, any further combination involving MA (e.g., $\{MA, IDT\}$ or $\{MA, ES\}$) is obviously successful, since MA's capabilities are unchanged or even enriched. However, only $\{MA\}$ will be considered since the other attackers do not represent minimal subsets.*

4.3 Risk Analysis

The risk analysis that we perform is based on the *OWASP Risk Rating Methodology* described in Section 2. By taking inspiration from (Pernpruner et al., 2020), in our analysis we consider the following likelihood factors[1]:

- *Technical Difficulty (TD)*: rates the technical difficulty of an attack to be successfully performed. More difficult attacks are connected with a lower likelihood;
- *Opportunity (O)*: provides a measure of the attacker's opportunity to perform the attack. This opportunity can be restricted by limitations affecting the chance of successfully completing an attack; for instance, a device protected with a screen lock would be more difficult (and thus less likely) to be used than unprotected ones;
- *Attack Vector (AV)*: considers the means through which the attack must be performed. An attack requiring a physical intervention would be less likely than a fully remote one;
- *User Interaction needed (UI)*: measures whether an attack requires an interaction with the victim to be successful. Attacks requiring a precise operation by the user would be less likely than others not requiring any interaction;
- *Spread of Attack (SA)*: deals with the spread of the considered attack, according to trusted statistics. Popular attacks increase the probability.

Instead, the impact factors are the following:

- *Attack Scale (AS)*: considers how many people would be affected by the attack. More involved users result in a higher impact;
- *Attack Detection (AD)*: deals with the possibility of detecting a successful attack. When an attack is discovered, it can be hindered, thus making easy-to-detect attacks have a lower impact.

[1] The OWASP Risk Rating Methodology uses the term *factors* to identify the elements that are considered to compute the likelihood and impact (cf. Section 2). It is highly important to be aware that such factors have a completely different meaning than *identification factors*.

In case some security mitigations are implemented in the protocol, their effects must be taken into consideration when assigning values to the factors above.

Definition 2. *The* risk analysis problem *for an enrollment flow under a threat model $\mathcal{TM} = (\mathcal{ATT}, \mathcal{C})$ is to find the risk associated with all the minimal subsets of attackers violating $\mathcal{SG}$.*

Example 6. *The risk of a single attacker (e.g., MA from Example 5) is determined as follows. First, we need to assign a justified value to each likelihood and impact factor; for instance, considering that MA can operate remotely, its Attack Vector is 7. By repeating this operation for each factor, we obtain all the values, which for MA can be the following: for the likelihood $TD = 3$, $O = 1$, $AV = 7$, $UI = 1$, $SA = 2$; for the impact $AS = 8$, $AD = 7$. Then, after computing the overall likelihood (2.80) and impact (7.50) by the average of the corresponding factors, we can assign them the corresponding labels (see Section 2):* Low *to the likelihood,* High *to the impact. The combination of these values, according to Table 1, results in a* Medium *risk.*

When dealing with a combination of attackers, finding a unique value for each factor is not trivial: in fact, every attacker that is part of the combination has his own values for all the factors. Therefore, we need to combine attackers' values as if we were considering a single attacker. Given the factors' meanings, we use the following logic:

- for *TD*, *AV*, *UI*, *AS* and *AD* the combined value should be the minimum of all the attackers' values for the considered factor;
- for *O* and *SA* the aggregated value should be less than the minimum of all the attackers' values for the considered factor.

Example 7. *Considering the combination $\{IDT, ES\}$, the thief (IDT) requires physical intervention ($AV = 1$) while the keylogger (ES) can act remotely ($AV = 7$); therefore, even the combination of such attacks must be performed (at least partially) through a physical intervention, thus the combined Attack Vector will be* 1*. On the other hand, considering that the thief (IDT) has $O = 3$ while the keylogger (ES) has $O = 4$, their combined Opportunity is further reduced to* 2 *(less than the minimum) since the two single attackers must collude in order to be effective.*

Once a single value for every factor has been computed, the procedure can continue as when considering a single attacker.

4.4 Usefulness of the Framework

Once the analyses have been performed, the outputs of the framework can help properly tune the security level of enrollment procedures: in many contexts, such as the financial one, different categories of customers may have different needs to address; the framework can help determine which solutions are worth adopting depending on the risk level one is willing to accept. Moreover, the possibility to analyse a protocol by changing the set of security mitigations implemented allows a *what-if analysis*, thus helping designers understand the risks and analyse the effects of mitigations on their protocols.

The framework can be used to model and analyse the security and risk of any enrollment procedure, since the set of actions involved (i.e., the specification language) and the capabilities of the attackers on the associated identification factors (i.e., the threat model) can be fully customised and adapted (if necessary) to the considered scenario.

In addition, the security and risk analysis problems that we have defined in Definitions 1 and 2 are decidable. In the former case, the security analysis problem considers finite sets, thus it is always possible to analyse all the involved attackers and combinations of them to identify those violating the security goal (if any). As for the latter case, instead, the risk analysis problem takes in input the list of successful attackers from the security analysis and evaluates the risk for each of them, thus working on finite sets again. These considerations make the framework suitable for implementation in an automated tool (currently an ongoing activity) to automatically assess the security of enrollment procedures.

5 MITIGATIONS

By both analysing state-of-the-art solutions and elaborating some ideas ourselves, we have identified a list of security mitigations that can be applied to enrollment procedures involving eDocuments (as per Section 3); these mitigations are displayed in Table 3 together with the effects they have on the attackers belonging to $\mathcal{ATT}$ (and thus on their capabilities in $\mathcal{C}$). To better understand them, it is important to notice that – depending on how the interaction with the eDocument happens – we can distinguish two scenarios:

- *MRTD scenario*: the applicant uses its eID card and allows the app to interact with it by scanning the `MRZ` (see Section 2);

Table 3: List of the identified mitigations and their effects.

#	Mitigations	Effects on attackers	Source
M1	Provide the app with a root detection library, in order to prevent the use of the app on rooted devices.	ES, MA ⬇ (makes the detection easier in case this attacker forces a root of the device)	From (OWASP, 2020) (§ 8.1).
M2	Leverage code obfuscation methodologies to prevent attackers from fully understanding the source code of the app (at least the most sensitive functions).	MA ⬇ (decrease the chance of understanding the code, and thus to alter the app)	From (OWASP, 2020) (§ 8.9).
M3	Implement, in the IAS ECC scenario, a binding between the challenge and the certificate used to establish a mutual TLS channel.	MA ⬇ (only in the IAS ECC scenario: prevent the attacker from injecting a stolen response in the legitimate app)	Originally used in OAuth 2.0 to bind access tokens (Campbell et al., 2020).
M4	Implement, in the MRTD scenario, a set of checks on the authenticity of the provided data.	MA ⬇ (only in the MRTD scenario: prevent the attacker from sending fake information)	
M5	Require users to capture a selfie at that moment, preventing the upload of existent files.	SE ✓ (cannot reuse files that have been deceivingly obtained by the user)	From many state-of-the-art solutions (see Section 7).
M6	Force the user to capture the selfie from the front camera.	SS ✓ (cannot capture pictures or video of people in the proximity)	From many state-of-the-art solutions (see Section 7).
M7	Request users to capture the selfie also in the IAS ECC scenario.	IDT, ES ✓ (cannot obtain the selfie of the user) SS, SE ⬇ (have less probability of obtaining the selfie of the user)	
M8	Implement liveness detection libraries to detect the misuse of static or modified pictures.	SE ✓ (cannot inject fake pictures on the legitimate app) SS ⬇ (decrease the chance of obtaining pictures or video of people in the proximity) NB: MA is not affected as they can bypass the liveness checks	From most state-of-the-art solutions (Jumio, 2020a; iDenfy, 2020b).
M9	Implement techniques to link the selfie captured during the procedure to the ongoing session, thus requiring the users to insert some *fresh* information.	MA ⬇ (cannot misuse already existent pictures or selfies that have already been used in other enrollment operations)	From some state-of-the-art solutions (Cassidy, 2018).

✓ = attacker made ineffective ⬇ = attacker's risk decreased

- *IAS ECC scenario*: the applicant uses its eID card and allows the app to interact with it by providing the `PIN`.

The proposed mitigations are fully compliant with the requirements in Section 3, with particular regard to *R1*: even those that are performed server-side (e.g., M8 and M9) can indeed be implemented through fully automated machine-learning algorithms that require human intervention only when certain security thresholds cannot be achieved.

Security designers do not necessarily need to implement all the proposed mitigations, as they can select those fitting best in their use cases. Of course, implementing only a subset of the proposed mitigations could affect the security and risk level of the procedure.

In addition to the description in Table 3, it is worth providing further details on some less-trivial mitigations, so as to better understand and thus manage to implement them in a proper way.

M3. To prevent the use of a stolen response by unauthorized parties, we suggest implementing a procedure similar to the one described for OAuth in RFC 8705 (Campbell et al., 2020). The mutual TLS channel that gets established does not aim to perform client authentication (self-signed certificates do not provide any guarantee), yet the binding of the app's self-signed certificate to the authentication challenge. This way, as only the app knows the private key corresponding to its self-signed certificate, the server can be sure that the response presented by a client contains a challenge that was previously issued to the same client, and thus that it has not been stolen or tampered with. This procedure involves the following additional steps:

1. before starting the enrollment procedure with the eID card, the app must generate a key pair with a self-signed certificate on the user's mobile device;
2. the app establishes a mutual TLS channel with the server using the generated key pair and certificate;
3. the server generates a challenge, associates the

client certificate with it and sends the certificate-bound challenge to the app;

4. the app interacts with the eID card to sign the challenge. The response is sent to the server after establishing a mutual TLS channel;
5. the server verifies the response and checks that the certificate used to establish the mutual TLS channel is the same that is bound to the challenge.

M4. In Section 2, we described how people's personal data can be verified server-side in the IAS ECC scenario through a challenge-response protocol. Moreover, we stated that the MRTD scenario provides guarantees on the extracted information only for in-person procedures, with physical readers verifying the *Document Security Object* (`SOD`, detailed in (ICAO, 2015)). However, as briefly explained in M4, we would need a way to trust the information extracted even during remote identification operations. To this end, we could rely on the `SOD` in a different way: since this object provides a digital signature of all the eDocument's data groups, the app could send it along with the extracted data, so to allow the server to perform the proper verifications on the authenticity and integrity of the received data. In this way, MA attackers trying to violate $\mathcal{SG}$ by sending fake information – without even interacting with a real eDocument – are prevented, since they would need a `SOD` that they cannot provide since those personal data are fictional and not extracted from an existing eDocument.

On the other hand, MA attackers that interact with an eDocument through MRTD can also read the `SOD`: for this reason, although the server can check the authenticity of the data (i.e., the *Validation* part of the enrollment flow, as detailed in Section 2), it cannot verify that they really belong to the applicant (i.e., the *Verification* part).

M7. As the picture contained in the eID card is not included in the `X509` certificate, M7 requires the reader to access both the IAS ECC and MRTD applications (to retrieve the `X509` certificate and the owner's picture, respectively). To do this, the `MRZ` value must be inferred by following a specific procedure (ICAO, 2015). Once this value has been derived, the reader can mutually authenticate with the eID card and access the picture as in the MRTD scenario.

M9. The most promising techniques to implement this mitigation are the following:

S1. Require the user to write an alphanumeric code suggested by the app on a piece of paper, and take a selfie with it; the code aims to enable the server to perform some verifications on the link with the ongoing operation. Specifically, the server should verify that: the user's face from the selfie matches the one from the eDocument's picture, the code written on the paper is correct, and the selfie has been sent within a certain interval.

S2. Require the user to capture a video of herself while reading some words aloud; these words would be suggested by the app to enable the server to perform some verifications on the link between the words and the ongoing operation. Specifically, the server should verify that: the user's face from the video-selfie matches the one from the eDocument's picture, the words read aloud are correct, and the video-selfie has been sent within a certain interval.

6 FRAMEWORK APPLICATION

In this section, we apply the framework described in the previous sections to a concrete enrollment procedure satisfying the requirements in Section 3 and specifically based on eDocuments. We have performed this work in collaboration with the Italian FinTech startup *CherryChain*[2], by focusing on a realistic use case inspired to their research and innovation activities: one of the main projects they are developing aims at realising a unique platform to connect service providers (e.g., energy suppliers) and consumers, thus handling the whole contracting lifecycle; the first step of this project is represented by the enrollment.

After describing the enrollment procedure that we have designed for this use case (Section 6.1), we perform the security and risk analysis on the designed procedure (Sections 6.2 and 6.3).

6.1 Specification Language

In Section 5, we have identified the two possible scenarios when dealing with eDocuments; each of them can be associated with an enrollment flow that we now describe by detailing all the actions involved. Regardless of the scenario, each flow begins with some common actions: first, the user is requested to agree with the privacy policy and terms of service (🤝). Then, she has to choose the type of document she would like to use (in this case, eID card) and whether she wants to provide the `PIN` or not (▤), thus being redirected to the IAS ECC or MRTD scenario, respectively. The next actions, instead, are strictly connected with the considered flow:

[2] https://www.cherrychain.it/

MRTD : 🤝; ☰; ⛶(▥); Ⓝ(🪪); 👤✓; @; 💬; 👤📷
IAS ECC : 🤝; ☰; ⌨(PIN); Ⓝ(🪪); 👤✎; 👤✓; @; 💬; 👤📷

6.1.1 Flow for the MRTD Scenario

After the common actions, the user is required to scan the MRZ of her eID card through the device's camera (⛶(▥)) and read this document through the device's NFC capabilities (Ⓝ(🪪)). The residence address can be extracted from the eID card, so the user only has to confirm the correctness of all the data or modify them in case, e.g., the residence address has changed (👤✓). After providing and verifying both the email address (@) and the phone number (💬), the user is required to capture a selfie (👤📷); the flow is then completed.

6.1.2 Flow for the IAS ECC Scenario

After the common actions, the user is required to insert the PIN (⌨(PIN)) and read her eID card through her mobile device's NFC capabilities (Ⓝ(🪪)). She is then asked to manually insert the residence address (👤✎) – not present in the X509 certificate, thus cannot be extracted from the eID card in this scenario – and confirm the correctness of all the data (👤✓). Once provided and verified both the email address (@) and the phone number (💬), the user is required to capture a selfie (👤📷); the flow is then completed.

6.2 Security Analysis

In this section, we perform a security analysis of the enrollment procedure described in Section 6.1. According to the considered flow, we can identify a list of identification factors involved:

- *MRTD*: the eID card (🪪) and the selfie (👤) that is later compared to the eID card's picture $\rightarrow \mathcal{SG}_{MRTD} = \{$🪪,👤$\}$
- *IAS ECC*: the eID card (🪪), its PIN and the selfie (👤) that is later compared to the eID card's picture $\rightarrow \mathcal{SG}_{IAS\_ECC} = \{$🪪,PIN,👤$\}$

To perform the security analysis, we have to understand how each attacker in $\mathcal{ATT}$ affects the identification factors; such capabilities (represented by $\mathcal{C}$) are displayed in Table 4 through the *padlock notation* described in Section 4.2.

6.2.1 Analysis for the MRTD Scenario

The flow belonging to this scenario relies on the eID card (🪪) and the selfie (👤) as identification factors, thus we have to focus only on these two columns in Table 4; according to them, the only minimal subset that manages to compromise the protocol is $\{MA\}$. In fact, although M4 and M9 considerably affect this attacker (by enabling the server to verify the integrity of the received data and the connection between the selfie and the ongoing enrollment operation, respectively), MA still manages to bypass them: in the former case, by getting the victim's personal data and SOD by interacting with her eID card through the malicious app on her device, then reusing these data on his own device; in the latter case, by maliciously interacting with the victim and deceiving her into providing the fresh selfie as required. Moreover, MA also manages to bypass M8 by pretending to be sending a fresh selfie.

Table 4: Attackers' capabilities.

Attackers	**Identification Factors**		
	🪪	PIN	👤
IDT	🔓	🔒	🔒
ES	🔒	🔓	🔒
SS	🔒	🔓	🔒
SE	🔒	🔓	🔒
MA	🔓*	🔓	🔓

🔒 = safe 🔓 = compromised 🔓* = indirectly compromised

Instead, some mitigations are completely effective against some other attackers: by preventing the reuse of selfies obtained through user's deception, M5 and M8 make SE ineffective; therefore, these mitigations thwart the combination $\{IDT, SE\}$ that would otherwise be able to obtain both the eID card and the selfie. Similarly, M6 and M8 defeat SS by preventing it from taking pictures or videos of users in proximity; therefore, also the combination $\{IDT, SS\}$ – which would be otherwise able to obtain the eID card and capture an unauthorized picture of the user as if it was a user's selfie – is not effective.

6.2.2 Analysis for the IAS ECC Scenario

The flow belonging to this scenario relies on the eID card (🪪), the PIN and the selfie (👤) as identification factors, thus we have to focus on all the columns in Table 4; according to them, even this scenario can be violated only by the minimal subset composed of $\{MA\}$. Regarding the mitigations, M3 prevents $\{MA\}$ from injecting a stolen signed challenge in legitimate apps, since the TLS binding requires the app that gets the challenge to be the same as the app that sends the response back. However, MA could be successful anyway when a signed challenge is captured by a malicious app in the user's smartphone and then injected into another malicious app under the control of the attacker. In this case, being both apps in control of the attacker, MA could use the same key pair to establish the mutual TLS channel, thus managing to enroll using information belonging to the victim. In the flow connected with this scenario, MA can also bypass M8 by pretending to be sending a fresh selfie.

Instead, M7 – in combination with M5, M6 and M8 – meets our expectations and makes $\{IDT, ES\}$, $\{IDT, SS\}$ and $\{IDT, SE\}$ ineffective. In fact, while obtaining the eID card and its PIN was generally enough, M7 also requires a user's selfie.

It is important to notice that the selfie would not be mandatory in this scenario: in fact, we could leverage the definition of *derived credential* (Grassi et al., 2017b), which is a credential issued on the basis of an already available credential that had previously been issued after identity proofing, in order not to duplicate this process. In this case, the possession of an eID card and the knowledge of the related PIN would prove the control of an authenticator (i.e., the eID card itself) that has been previously issued by a trusted entity (a Government office). Therefore, the *Verification* phase described as *part (iii)* in Section 2 could be fulfilled by simply inserting the eID card's PIN instead of providing additional elements such as a selfie. Nevertheless, the considerations above prove that the request for a user's selfie has indeed relevant consequences and significantly helps improve security.

6.3 Risk Analysis

The results of the risk analysis (in Table 5) highlight that MA is connected with a Medium risk in both scenarios. By comparing the risk values to those of MA with no mitigation implemented (Table 6), we can notice the considerable effectiveness of the proposed mitigations: beyond making most of the successful attackers ineffective (see Section 6.2) they even reduce the risk of MA, though without completely thwarting it. For the sake of clarity, the risk values positively affected by the implemented mitigations are highlighted in bold in Table 5. Further details on the values assigned to the likelihood and impact factors in Tables 5 and 6 are detailed on the complementary website[3].

As for M9, additional considerations are worthy: during the risk analysis, no difference has been detected between the effects of *S1* and *S2* on the overall risk, since in both cases MA manages to interact with the victim to obtain everything he needs. Specifically: in case of *S1*, MA has to deceive the victim into providing the attacker with a selfie of her, together with the code connected with the attacker's malicious enrollment attempt written on a piece of paper; in case of *S2*, MA has to deceive the victim into providing the attacker with a video-selfie of her, in which she also reads the words connected with the attacker's malicious enrollment attempt. Therefore, the choice above can be made depending on other factors such as usability, technical difficulty and/or server load. Nevertheless, no matter which of the above techniques is adopted, the overall risk is Medium in both scenarios. In general, all the proposed mitigations significantly reflect on the attackers' likelihood, while having little effect on the impact. Since the likelihood is already Low, affecting the impact (which would be a quite complex operation) would be the only way to further decrease the overall risk.

7 STATE-OF-THE-ART SOLUTIONS

Many works dealing with security methodologies applied to authentication protocols can be found in the academic literature, either from a formal (Jacomme and Kremer, 2021; Fett et al., 2016; Feng et al., 2021) or an implementation perspective (Sudhodanan et al., 2016; Yang et al., 2016). However, the same cannot be said for enrollment: to the best of our knowledge, no previous security methodology for such protocols has been developed so far. Therefore, our state-of-the-art review has focused on governmental (Section 7.1) and commercial (Section 7.2) enrollment solutions, which we have taken into consideration during both the design of the flows and the identification of the mitigations.

7.1 Governmental Solution: SPID

SPID (*Sistema Pubblico di Identità Digitale*, Public Digital Identity System) (AgID, 2020) is the Italian digital identity to access Public Administration's online services in a frictionless and secure way. Due to its sensitiveness, SPID can be released only by accredited Identity Providers; each of them may set up their own procedures to properly identify users (e.g., physical identification, remote identification through webcam or digital signature). Given the focus of this work, to extract basic actions and mitigations to consider we have further studied the procedures relying on eDocuments as identification means (e.g., the Poste Italiane solution (Poste Italiane, 2020)).

7.2 Commercial Solutions

Many companies provide commercial solutions to perform fully-remote identity verification, such as (Authenteq, 2020), (HooYu, 2020), (iDenfy, 2020a), (Jumio, 2020b), (Onfido, 2020), (Thales Group, 2020), (Veriff, 2020), and many other. Unfortunately,

[3] https://st.fbk.eu/complementary/SECRYPT2021

Table 5: Risk analysis of the enrollment procedure with all the mitigations implemented.

Scenario	**Attackers**	**Likelihood**							**Impact**				**Risk**
		TD	*O*	*AV*	*UI*	*SA*	Overall		*AS*	*AD*	Overall		
MRTD	$\{MA\}$	**3**	**1**	7	**1**	**2**	**2.80**	**Low**	**8**	**7**	**7.50**	High	**Medium**
IAS ECC	$\{MA\}$	**2**	**1**	7	1	**2**	**2.60**	**Low**	8	**5**	**6.50**	High	**Medium**

Table 6: Risk analysis of the enrollment procedure without mitigations, considering only MA.

Scenario	**Attackers**	**Likelihood**							**Impact**				**Risk**
		TD	*O*	*AV*	*UI*	*SA*	Overall		*AS*	*AD*	Overall		
MRTD	$\{MA\}$	6	9	7	7	6	7.00	High	9	8	8.50	High	Critical
IAS ECC	$\{MA\}$	3	2	7	1	4	3.40	Medium	8	6	7.00	High	High

companies rarely release details about their procedures, considering the commercial nature of their business. For these reasons, since we are not their customers, we can only refer to the available documentation that can be found online. The basic procedure shared between the various solutions (with slight changes) usually requires the applicant to take a picture of her document (either electronic or non-electronic) and then a selfie, so that the photo extracted from the former can be compared to the latter. After collecting the needed information, some checks are performed server-side; among these, the most common is the *liveness check* that we have implemented in our procedure as well.

Besides a basic procedure, some solutions also allow full customisation according to the customers' needs: in case of (Veriff, 2020), for instance, by adding or removing steps that the applicant must perform (e.g., video-selfie or selfie with the eDocument) and by enabling or disabling checks to perform (e.g., device verification, biometric comparison, liveness check and automated data extraction).

7.2.1 FinTech Solutions

To comply with Know Your Customer (KYC) guidelines, online banks have to identify prospective customers with a high degree of assurance before let them open a bank account. According to (Built for Mars, 2020), some institutes still perform verifications manually, hence increasing the customers' waiting before their account is activated. Anyway, the majority of the analysed banks leverage enrollment procedures that are mostly performed in an automatic way; manual intervention can still be needed, but only in some cases (exceptions or over-threshold risks). To this end, some institutes implement commercial solutions (e.g., (Monzo, 2020) relies on Jumio and (Revolut, 2020) on Onfido), while some other implement custom enrollment procedures.

8 CONCLUSIONS

This paper has focused on enrollment procedures, that are gaining more and more popularity since they allow a solid identification of the applicants. Given the sensitive contexts in which these operations are placed, they must provide a high level of assurance on the applicants' identities. Therefore, we have presented a framework for security and risk analysis of any enrollment procedure; this framework is composed of: (i) a *specification language* to provide a clear and graphical description of enrollment protocols; (ii) a *security analysis methodology* to obtain a list of the attackers that are able to compromise the protocols; (iii) a *risk analysis methodology* to sort the successful attackers according to their severity, thus enabling a prioritization of them. The outputs of the framework can be used to adjust the security level of enrollment procedures, also allowing to perform *what-if analyses* by changing the set of mitigations to consider and verifying the effects on security.

In the context of a collaboration with the Italian FinTech startup *CherryChain*, we have applied the proposed framework to fully-remote solutions relying on eDocuments as identity evidence. The collaboration with *CherryChain* was extremely important to both parties: on the one hand, it contextualized our work in a practical use case; on the other hand, our framework allowed *CherryChain* to verify the security of the protocols they were designing, also identifying the mitigations to implement after discussing their benefits in terms of security and feasibility.

Starting from this work, we plan to enrich the specification language that has been used in this paper to support a wider range of enrollment procedures, even based on different requirements. Moreover, to simplify and automatize the analysis process, we are currently formalising the proposed framework through formal definitions and pseudocodes that can be easily implemented within an automatic tool. Finally, it would be interesting to extend our work by taking inspiration from some of the considerations presented in a report (ENISA, 2021) released by the European Union Agency for Cybersecurity (ENISA) after this work was already completed.

ACKNOWLEDGEMENTS

The research has been partly supported by *CherryChain S.r.l.* in the context of a research and innovation project funded by the Autonomous Province of Trento, and by *IPZS* (*Poligrafico e Zecca dello Stato Italiano*, the Italian Government Printing Office and Mint) within the joint laboratory *DigiMat Lab*.

REFERENCES

AgID (2020). SPID. https://www.spid.gov.it/.

Authenteq (2020). Identity Verification Without Compromise. https://authenteq.com/.

Built for Mars (2020). Opening an account. https://builtformars.co.uk/banks/opening/.

Campbell, B., Bradley, J., Sakimura, N., and Lodderstedt, T. (2020). RFC 8705. https://tools.ietf.org/html/rfc8705.

Cassidy, N. (2018). Onfido launches the market's most robust video liveness detection. https://onfido.com/resources/blog/news-onfido-launches-the-market-s-most-robust-liveness-detection.

Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and Polk, W. (2008). RFC 5280. https://tools.ietf.org/html/rfc5280.html.

ENISA (2021). *Remote ID Proofing*. https://www.enisa.europa.eu/publications/enisa-report-remote-id-proofing.

Feng, H., Li, H., Pan, X., and Zhao, Z. (2021). A Formal Analysis of the FIDO UAF Protocol. In *Network and Distributed System Security Symposium*, NDSS 2021. https://www.ndss-symposium.org/ndss-paper/a-formal-analysis-of-the-fido-uaf-protocol/.

Fett, D., Küsters, R., and Schmitz, G. (2016). A Comprehensive Formal Security Analysis of OAuth 2.0. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16. https://doi.org/10.1145/2976749.2978385.

GIXEL (2009). *European Card for e-Services and National e-ID Applications: Identification Authentication Signature European Citizen Card (IAS ECC)*.

Grassi, P. A. and Fenton, J. L. (2017). Digital Identity Guidelines: Enrollment and Identity Proofing. SP 800-63A. https://doi.org/10.6028/NIST.SP.800-63a.

Grassi, P. A., Fenton, J. L., Newton, E. M., Perlner, R. A., Regenscheid, A. R., Burr, W. E., and Richer, J. P. (2017a). Digital Identity Guidelines: Authentication and Lifecycle Management. NIST Special Publication 800-63B. https://doi.org/10.6028/NIST.SP.800-63b.

Grassi, P. A., Garcia, M. E., and Fenton, J. L. (2017b). Digital Identity Guidelines. SP 800-63-3. https://doi.org/10.6028/NIST.SP.800-63-3.

HooYu (2020). Identify. https://www.hooyu.com/h/hooyu-identify/.

ICAO (2015). MRTD. Doc 9303. https://www.icao.int/publications/pages/publication.aspx?docnum=9303.

iDenfy (2020a). Identity Verification Service. https://www.idenfy.com/.

iDenfy (2020b). Liveness Detection to prevent Spoofing Attack. https://www.idenfy.com/blog/spoofing-attack-prevention/.

Jacomme, C. and Kremer, S. (2021). An Extensive Formal Analysis of Multi-Factor Authentication Protocols. *ACM Trans. Priv. Secur.*, 24(2). https://doi.org/10.1145/3440712.

Jumio (2020a). Certified Liveness Detection. https://www.jumio.com/technology/live-detection/.

Jumio (2020b). Jumio. https://www.jumio.com/.

Monzo (2020). Monzo. https://monzo.com/.

NIST (2017). Digital Identity Guidelines document suite. SP 800-63. https://pages.nist.gov/800-63-3/.

Onfido (2020). User onboarding. https://onfido.com/use-cases/user-onboarding/.

OWASP (2018). OWASP Risk Rating Methodology. https://owasp.org/www-community/OWASP_Risk_Rating_Methodology.

OWASP (2020). *Mobile Application Security Verification Standard*. Version 1.2. https://github.com/OWASP/owasp-masvs/releases/download/v1.2/OWASP_MASVS-v1.2-en.pdf.

Pernpruner, M., Carbone, R., Ranise, S., and Sciarretta, G. (2020). The Good, the Bad and the (Not So) Ugly of Out-of-Band Authentication with eID Cards and Push Notifications: Design, Formal and Risk Analysis. In *Proceedings of the Tenth ACM Conference on Data and Application Security and Privacy*, CODASPY '20. https://doi.org/10.1145/3374664.3375727.

Poste Italiane (2020). PosteID abilitato a SPID. https://posteid.poste.it/.

Revolut (2020). Get more from your money. https://www.revolut.com/.

Sudhodanan, A., Armando, A., Carbone, R., and Compagna, L. (2016). Attack Patterns for Black-Box Security Testing of Multi-Party Web Applications. In *Network and Distributed System Security Symposium*, NDSS 2016. https://www.ndss-symposium.org/wp-content/uploads/2017/09/attack-patterns-black-box-security-testing-multi-party-web-applications.pdf.

Thales Group (2020). ID Verification Suite. https://www.thalesgroup.com/en/markets/digital-identity-and-security/government/identity-verification.

Veriff (2020). Veriff. https://www.veriff.com/.

Yang, R., Lau, W. C., and Liu, T. (2016). Signing Into One Billion Mobile App Accounts Effortlessly with OAuth2.0. In *Black Hat Europe 2016*. https://www.blackhat.com/docs/eu-16/materials/eu-16-Yang-Signing-Into-Billion-Mobile-Apps-Effortlessly-With-OAuth20-wp.pdf.