


A Formal Analysis of CIE Level 2 Multi-Factor Authentication via SMS OTP

Roberto Van Eeden¹, Matteo Paier^{1,2} ^a and Marino Miculan^{1,3} ^b

¹University of Udine, Dept. of Mathematics, Computer Science and Physics, Italy

²IMT School for Advanced Studies, Lucca, Italy

³Ca' Foscari University of Venice, Dept. of Environmental Sciences, Informatics and Statistics, Italy

Keywords: Formal Methods, Security Protocols, Digital Identity, Identity Management.

Abstract: We analyze the security of Level 2 multi-factor authentication (MFA) based on SMS One-Time Passcode (OTP) of Italian Electronic Identity Card (CIE). We propose a novel threat model encompassing password compromise, network disruptions, user errors, and malware attacks. The combinations of the adversary's attack capabilities yield a plethora of possible attack scenarios, which we systematically generate, formalise and verify in ProVerif. Our analysis reveals that CIE MFA based on SMS OTP is vulnerable to attacks with read access to the mobile device or keyboard, or to phishing, but even to mere read access to the user's computer screen. To address the latter vulnerability, we propose a minor modification of the protocol. The threat model we introduce paves the way for the analysis of other CIE MFA protocols.

1 INTRODUCTION

The Italian electronic identity card (*carta d'identità elettronica*, CIE, (Italian Ministry of the Interior, 2024)) is the official Italian digital identity token replacing paper identity cards since 2016. CIE, offering robust digital online authentication and legally valid electronic signatures, is envisioned to become the primary tool for accessing public services like education, healthcare, and tax offices. Its compliance with the EU's eIDAS regulation (EU Parliament and Council, 2014) enables cross-border use within the European market, e.g. for accessing foreign services or executing internationally binding transactions.


Given its critical importance, CIE provides various multi-factor authentication protocols. While Level 1 authentication requires the knowledge of the user's credentials only, Level 2 authentication requires also to prove the possession of a previously registered mobile device. This is verified during the authentication process, either through a dedicated application (CieID) or by receiving an One-Time Passcode (OTP) via SMS. The latter is the sole practical way for users without access to a smartphone — a situation quite common especially among the elderly.


This paper analyzes the security of CIE's Level 2 multi-factor authentication mechanisms, focusing particularly on the SMS OTP method (L2SMS).

To this end, we propose a threat model for CIE encompassing the core threats outlined in NIST SP 800-63B relevant to MFA protocols (NIST, 2020). These threats include compromised passwords (the very reason MFA exists), network disruptions (delayed or dropped messages), overall security of TLS channels, human error induced by phishing attacks, malware compromising user devices. In particular, we characterize the system as a set of *input and output interfaces*, following (Jacomme and Kremer, 2021); then, an *attack scenario* is defined by the control the attacker may gain on these interfaces. Depending on the attacker's capabilities, we obtain hundreds of different attack scenarios.

In order to analyse the security of L2SMS, we have systematically and automatically generated the formalisation of all these scenarios in the applied π -calculus, a formal language designed for representing security protocols in the Dolev-Yao model. Then, these formalisations have been thoroughly analysed with ProVerif, a protocol verification tool in the symbolic Dolev-Yao model (Blanchet et al., 2016).

This analysis reveals L2SMS's vulnerability to attacks gaining read access to the mobile device or the PC keyboard (e.g., using a keylogger or planting a

^a  <https://orcid.org/0009-0000-7588-7169>

^b  <https://orcid.org/0000-0003-0755-3444>

malware in the PC), or through phishing. Even the mere access to the user’s computer screen allows an attacker to gain unauthorized Level 2 access to services knowing only Level 1 credentials. Leveraging the attack trace found by ProVerif in this latter case, we propose a minor modification to strengthen the protocol’s security. Formal analysis of the modified protocol confirms its effectiveness in addressing the read-screen vulnerability.

The results of this work can provide useful indications for improving the CIE platform and services. Moreover, the threat model we introduce in this paper paves the way for further in-depth analysis of other CIE MFA protocols. Finally, this methodology can be applied to other similar digital identity cards.

This paper is organized as follows. In Section 2 we recall CIE main components according to an interface-based model, and the Level 2 authentication protocol via SMS OTP. In Section 3 we define the threat model for CIE, discuss the formalization in ProVerif of the various attack scenarios, and analyse the verification results. To address the screen-reading vulnerability, in Section 4 we propose a slight correction to the protocol. Finally, in Section 5 we draw some conclusions, recall related work, and give directions for future work.

2 CIE ARCHITECTURE AND SMS OTP AUTHENTICATION

CIE authentication uses the Security Assertion Markup Language (SAML) 2.0 open standard (Lockhart and Campbell, 2008) for exchanging authentication and authorization identities between security domains. The main parties involved are the CIE *Identity Provider* (IdP) and federated public administration *Service Providers* (SPs). The IdP authenticates users and generates security assertions.

Informally, when a user logs in to a SP using CIE, the SP: (1) generates a random operation ID; (2) generates a signed SAML Authn request for the login operation (including the required security level); and (3) redirects to the CIE IdP service with the Authn request as a parameter. The CIE IdP authenticates the user at the level required by the SP and, after confirmation, generates a matching signed SAML authentication assertion that the SP can validate.

All messages between the IdP server and the devices involved in the login process are exchanged over TLS (unless otherwise noted).

The CIE identity provider offers three distinct authentication levels for secure online access:

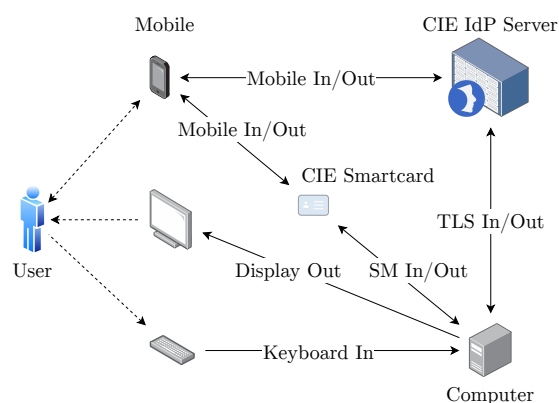


Figure 1: CIE system interface model.

Level 1 (Single Factor). Basic username and password login;

Level 2 (Two Factor). Authentication by means of a one-time passcode (OTP) received via SMS, or via the CieID app on a registered device. In the second case, the second factor PIN must be entered after receiving a push notification or scanning a QR code;

Level 3 (Two Factor with Smartcard). Authentication using the physical CIE smartcard and PIN. Two options exist: using a dedicated USB NFC card reader to interact with the smartcard or using the CieID app on an NFC-enabled smartphone.

For optimal security, most service providers enforce Level 2 or 3 multi-factor authentication, reserving Level 1 for low-risk scenarios with limited personal data involvement.

Following the approach used in (Jacomme and Kremer, 2021), our analysis adopts a model where devices, computers, servers, etc., are represented as abstract sets of interfaces facilitating data input and output. Each interface can be compromised independently from the others, yielding many different attack scenarios. The interface model for CIE is shown in Figure 1, which we describe next.

The PC used by the user (here the *computer*) is modelled by the following interfaces:

Display. Represents the user-facing screen, capable only of data output;

Keyboard. Represents user’s input to the computer;

TLS. Represents encrypted communications with remote entities like the IdP server. This interface offers both input and output at once, because we assume that attackers cannot compromise only one direction (incoming or outgoing) of TLS sessions;

SM. Represents bidirectional communication with the CIE smartcard via secure messaging (SM).

While the NFC reader connects through USB, SM communication itself is encrypted and authenticated. We consider the possibility of attackers compromising the USB interface without affecting SM sessions, necessitating its modelling as a separate, potentially compromised interface.

For *mobile* devices, the diverse interfaces involved in CIE authentication (SMS, TLS, NFC communication with the smart card, etc.) are condensed into a single interface with both input and output capabilities, as in (Jacomme and Kremer, 2021). This simplifies comparisons between different attack strategies required to compromise the various protocols.

The *Smartcard* has two interfaces: the SM In/Out with the PC, and the NFC In/Out with the mobile device. Although in the following we will not consider authentication schemata using the smartcard, we include it in the model for sake of completeness.

The *IdP server* is represented by the TLS interfaces for communicating with the user's computer, and the interfaces (SMS or TLS) to communicate with the mobile device.

Level 2 Authentication via SMS OTP. In this paper we focus on the two-factor authentication scheme L2SMS, which requires both knowledge of login credentials and ownership of the SIM card linked to the account. The sequence diagram is shown in Figure 2.

The user initiates the process by entering their username and password, which are sent to the IdP server over a TLS channel. Upon verification, the IdP server generates a unique OTP code, tied to the specific login operation, and sends it via SMS to the registered phone number. The user reads the OTP code from the mobile device, and enters it on their computer, which shows it on the computer screen and transmits it to the IdP server for comparison. If the codes match, the IdP server successfully completes the two-factor authentication process.

3 FORMAL ANALYSIS OF L2SMS

In this section we analyse the L2SMS protocol under various threat scenarios: the user's password is compromised, attackers possess varying levels of control over user devices via malware, and phishing attempts might be successful. By simulating these threats, the analysis comprehensively evaluates the effectiveness of this multi-factor approach in safeguarding user accounts. The formal analysis has been conducted within the symbolic Dolev-Yao model, as implemented by ProVerif (Blanchet et al., 2016). The formalization is available at (Van Eeden et al., 2024).

3.1 Threat Model

In this subsection we consider various ways in which the attacker can compromise the interfaces of the CIE architecture, following (Jacomme and Kremer, 2021).

A system compromise by an attacker can be represented by the level of control that the attacker has acquired over the system's interfaces, potentially influencing data flow through inputs and outputs. This influence can be partial, in the sense that the attacker may be able to only read from *or* write to specific data streams within an interface, like its inputs or outputs. In a secure system, the attacker does not have access to any interface, neither in writing nor in reading. Conversely, in a fully compromised system, the attacker possesses read-write access on all interfaces.

We observe that in general, gaining write access to a system interface is more difficult than simply reading data: writing to an interface requires higher privileges or the exploitation of specific vulnerabilities, whereas reading data demands less elevated permissions. As a consequence, while vulnerabilities granting write-only access do exist, they are uncommon, and do not apply to the CIE: in our model, once an attacker gains write access, it has also read access to the interface. This aligns with the general principle that passive attacks are easier than active attacks.

In the light of these observations, the control that an attacker can gain over a given interface can be classified into these levels:

- in:RO: attacker has acquired read-only access to data flowing into the interface;
- in:RW: attacker has acquired read-write access to data flowing into the interface;
- out:RO: attacker has acquired read-only access to data flowing out the interface;
- out:RW: attacker has acquired read-write access to data flowing out the interface.

We can denote by $\mathcal{A}_{d:a}^{if}$ the level of control exercised by an attacker over the interface *if*, where:

- $d \in \{\text{in}, \text{out}, \text{io}\}$ represents the direction which the attacker gains control over: the input (in) or the output (out). We use io as a shorthand to denote both directions;
- $a \in \{\text{RO}, \text{RW}\}$ represents the access control acquired by the attacker: read-only (RO) or read-write (RW).

Then, a *threat scenario* \mathcal{S} where the attacker acquires *access control levels* $d_1:a_1, \dots, d_n:a_n$ on interfaces if_1, \dots, if_n respectively, is represented by the set

$$\mathcal{S} = \{\mathcal{A}_{d_1:a_1}^{if_1}, \dots, \mathcal{A}_{d_n:a_n}^{if_n}\}.$$

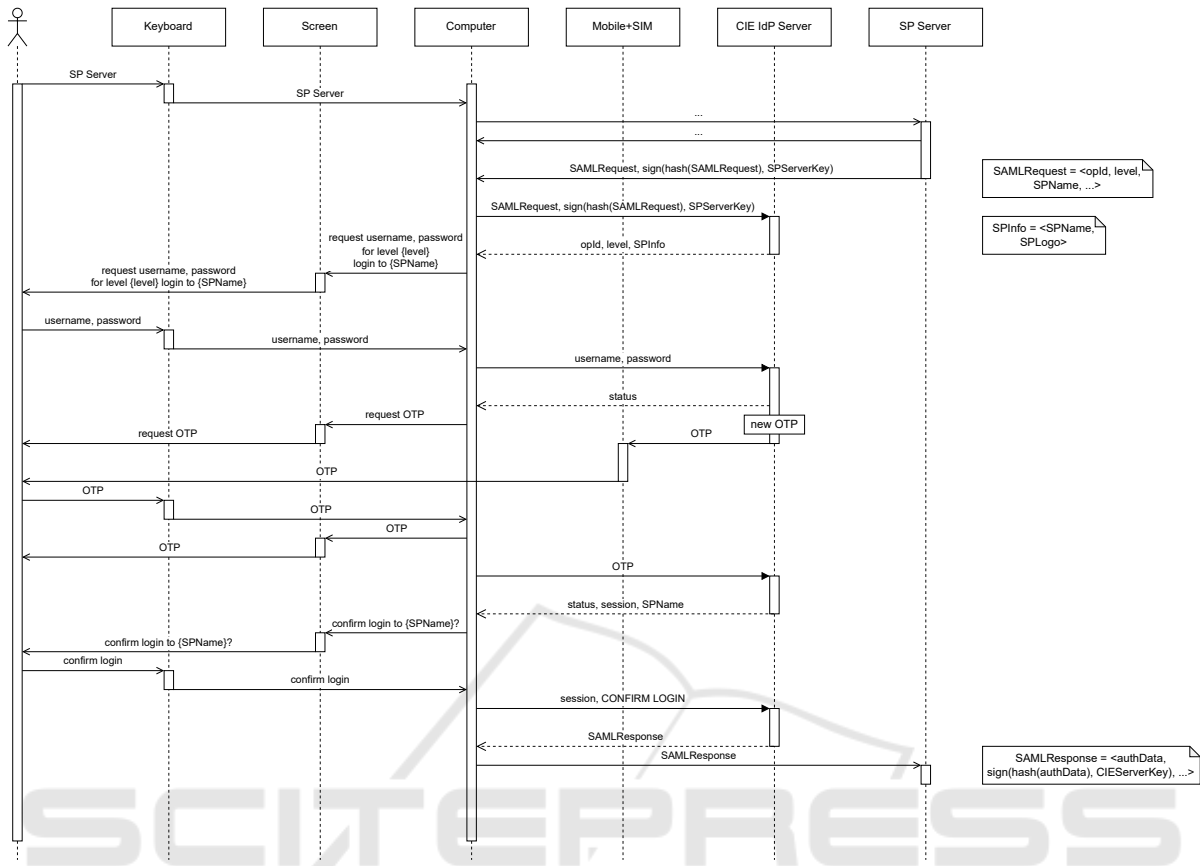


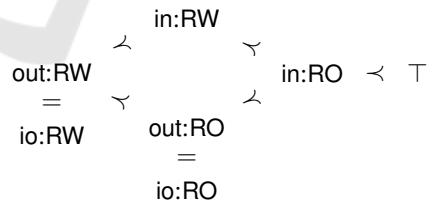
Figure 2: Sequence diagram of L2SMS, the CIE level 2 authentication via SMS OTP protocol.

As an example, $\{\mathcal{A}_{out:RW}^{displ}, \mathcal{A}_{in:RO}^{mobile}\}$ represents the scenario where the attacker can both read and display arbitrary content on the computer screen, and read messages received by the mobile device.

In principle, all combinations of read-only/read-write access on the various interfaces are possible; for our CIE model, this would yield a plethora of over 2,000 attack scenarios of different granularity and severity. Actually, we can reduce this complexity by observing that acquiring control over an interface's output is generally more challenging than controlling its input, due to the different execution privileges required for each action. For example, reading inputs from a USB keyboard often only requires user-mode code execution, while intercepting or modifying data sent over USB by another program requires elevated privileges (e.g., admin access) or the exploitation of specific vulnerabilities in the target system.

In the light of this observation, the control access levels can be ordered according their strength (and hardness to achieve): out:RW is stronger than both out:RO and in:RW; both out:RO and in:RW are stronger than in:RO, which is stronger than no control access at all. This defines a preorder \preceq on the set of

control access levels: $l_1 \preceq l_2$ means that achieving the access level l_1 implies achieving also the access level l_2 . Overall, the order between levels is as follows, where T represents no access.



where $l_1 = l_2$ means $l_1 \preceq l_2$ and $l_2 \preceq l_1$, and $l_1 \prec l_2$ means $l_1 \preceq l_2$ and $l_1 \neq l_2$.

This order can be extended to scenarios in the natural way: $\mathcal{S}_1 \preceq \mathcal{S}_2$ if and only if, for all $\mathcal{A}_{l_2}^{if} \in \mathcal{S}_2$ there exists $\mathcal{A}_{l_1}^{if} \in \mathcal{S}_1$ such that $l_1 \preceq l_2$. The order is strict, denoted as $\mathcal{S}_1 \prec \mathcal{S}_2$, when $\mathcal{S}_1 \preceq \mathcal{S}_2$ and $\mathcal{S}_1 \neq \mathcal{S}_2$. Clearly, we have the following properties:

- If the protocol is secure in scenario \mathcal{S} , then it is secure in every scenario \mathcal{S}' such that $\mathcal{S} \preceq \mathcal{S}'$.
- Conversely, if a protocol is vulnerable in scenario \mathcal{S} , then it is vulnerable in every scenario $\mathcal{S}' \preceq \mathcal{S}$.

These properties enable efficient analysis by reducing the scope of scenarios. If an attack is found in a given scenario, we can skip analysing stronger ones. Conversely, a verified secure protocol in a scenario does not require checking in weaker ones. This reduces the number of scenarios to analyse to less than 270.

A particular case is that of TLS channels, used by all CIE protocols to securely communicate with the servers. The formalisation of TLS is out of the scope of this paper (see e.g. (Bhargavan et al., 2017)). Here we adopt a simplified model assuming the essence of TLS (secrecy, authentication, and order protection) while permitting message blocking or delays by attackers. The only meaningful access levels for TLS are `io:RO` and `io:RW`, because once an attacker gains access to one direction of a TLS channel, it can access also the other one. Furthermore, we consider also the possibility that the attacker can initiate TLS sessions with legitimate servers, and impersonate an IdP server for unsuspecting clients.

Phishing. Human errors and social engineering tactics remain significant threats to multi-factor authentication systems, even with robust protocols. Hence, evaluating the resilience of CIE’s multi-factor authentication against incorrect user actions is crucial.

Phishing attacks are modelled similarly to (Jacomme and Kremer, 2021), but we exclude the possibility of omitting fingerprint comparisons because the CIE authentication process never visually displays the fingerprint of the device attempting login—and hence fingerprint comparisons is not an option.

With phishing attacks growing more sophisticated and diverse, employing a wider range of tools like SMS, calls, websites, etc., untrained users become increasingly vulnerable. They may unknowingly grant access to malicious replicas of services or reveal sensitive information like OTP codes. This risk is modelled in simulated phishing scenarios by allowing the attacker to control the targeted server, i.e., the server that the user will initiate authentication with.

To minimize phishing risks the CielD app shows a three-second warning message urging users to compare the displayed URL with the official CIE IdP URL. However, the possibility of users disregarding this warning remains a concern. Even with this safeguard, advanced techniques like IDN homograph attacks can deceive users into believing a fraudulent page is genuine (Holgers et al., 2006). However, these countermeasures do not apply to Level 2 authentication via SMS OTP, which is always available anyway.

Formally, the presence of a phishing attack is denoted by including the special attack symbol PH in the attack scenario.

3.2 Formalization in ProVerif

In this subsection we briefly describe the formalization of the L2SMS protocol and the interface model, in ProVerif, a state-of-the-art tool for the verification of security protocols. For an introduction to this tool, we refer the reader to (Blanchet et al., 2016).

Within a system, each interface can be modelled as a pair of private channels, one for input and one for output (Jacomme and Kremer, 2021). To simulate an attacker gaining read-only access to such an interface, we expose on a publicly accessible channel all message names that are transmitted through the corresponding private channels, during the execution of the protocol. Formally, let `iface` represent a secret interface channel and `public` represent a public channel accessible to the attacker. Granting the attacker read-only access to the interface is equivalent to transforming each instance of `out(iface, message)` into `out(public, message); out(iface, message)` and each instance of `in(iface, message)` into `in(iface, message); out(public, message)`. In contrast, granting the attacker read-write access to the channel involves divulging the private channel name itself to the attacker, effectively handing over complete control of the communication channel.

The secure and authenticated communication via TLS channels is modelled using a specialized function `TLS(id, id):channel`. To ensure legitimacy, only authorized parties can access this function to obtain unique channel names. When establishing a TLS connection, one of the participating hosts generates a fresh session name, attaching it to all subsequent messages to prevent mixing data across different sessions.

However, as ProVerif’s private channels are typically synchronous, this model would not allow the attacker to block or delay messages. To address this limitation, each TLS channel is extended with the interleaving concurrency semantics using the “parallel” operator from the π -calculus, allowing the attacker to manipulate the execution order of processes, thus simulating message interception or delays.

Moreover, in order to grant the attacker the ability to always initiate TLS sessions with the Identity Provider (IdP) server, a dedicated TLS manager process publicly discloses the channel names where one of the two parties involved is under attacker control.

To simulate phishing attacks, the model retrieves the target server URL from a publicly accessible channel. This represents potentially compromised online sources susceptible to manipulation. Normally, a diligent human user would compare the obtained URL with the legitimate IdP server’s known address.

However, in scenarios simulating phishing attacks, the model assumes that the human victim to the deception bypasses this verification step.

The main property under consideration is the *injective correspondence* between accesses made by a specific device to the account and the login attempts initiated by the user on that device: each successful access on the device must match a distinct login attempt initiated by the user on the same device.

The L2SMS protocol has been tested under every possible combination of malware and human threat scenarios. Overall, these combinations lead to 270 possible scenarios. The ProVerif code corresponding to each of these scenarios is automatically generated by means of the m4 macro processor (Kernighan and Ritchie, 1977). A set of preprocessor definitions has been associated with each level of malware interface access and human threat; the necessary checks and reveals are automatically generated based on the supplied definitions. A Python script iterates over the set of all possible combinations of scenarios, and invokes m4 and ProVerif as necessary.

To avoid unnecessary computations, we skip the verification of instances where the result is entirely implied by weaker or stronger scenarios. Namely, we skip scenarios where the results are already determined by others with either more attacker capabilities (stronger control of the system) or increased user vulnerabilities (exposed to additional threats).

3.3 Results

While the L2SMS protocol offers an additional layer of protection against unauthorized access compared to relying solely on passwords, its security raises some concerns. Table 1 shows the results of our formal analysis of the L2SMS authentication schema; the corresponding ProVerif formalizations are available at (Van Eeden et al., 2024).

While brute-force attacks using stolen credentials might be deterred, L2SMS falls short against more sophisticated techniques. Even relatively simple malware like USB keyloggers can compromise the system if they can intercept the one-time passcodes (OTPs) sent via SMS. Any attacker capable of capturing these codes essentially holds the key to bypassing L2SMS authentication, rendering it ineffective against such threats.

As an example, an attack trace for the $\mathcal{A}_{out:RO}^{displ}$ scenario (corresponding to, e.g., the presence of a screen capturing software) has been found by ProVerif; a sequence diagram describing this attack is shown in Figure 3. In this case, an attacker may initiate a login attempt shortly after the user; to proceed with the le-

Table 1: Analysis results under different threat scenarios.

Threat Scenario	L2SMS
	✓
$\mathcal{A}_{in:RO}^{mobile}$	✗
$\mathcal{A}_{out:RO}^{displ}$	✗
$\mathcal{A}_{out:RW}^{displ}$	✗
$\mathcal{A}_{in:RO}^{usb}$	✗
$\mathcal{A}_{io:RO}^{tls}$	✗
PH	✗

Results:

- ✓ correctness proven
- ✓ correctness proven in a stronger scenario
- ✗ attack found
- ✗ attack found in a weaker scenario

Scenarios:

- PH: Phishing
- $\mathcal{A}_{d,a}^{if}$: Attacker control over a system interface, where:
 - *if* is the name of the system interface
 - *d* represents control over interface’s inputs (in), outputs (out) or inputs and outputs (io)
 - *a* represents the access control over the interface: read-only (RO) or read-write (RW)

gitimate login attempt, the user may enter the OTP code relative to the malicious login attempt (mistaking it for the code relative to the legitimate login attempt), causing it to be reproduced on the computer display as it is typed (the OTP input field is not hidden). An attacker with $\mathcal{A}_{out:RO}^{displ}$ access and knowing only the credentials for level 1 access, would then be able to obtain a copy of the OTP code and successfully authenticate as the user at level 2.

Due to the time-sensitive nature of the attack (i.e. there is a limited time window between OTP display and transmission to the IdP server), an attacker could achieve greater success by also manipulating message delivery. This manipulation could involve delaying or entirely dropping the message by, e.g., disrupting the network connection utilized by the computer.

Additionally, the choice of SMS communication introduces inherent weaknesses compared to the TLS channels employed by other protocols. This exposes L2SMS to a wider range of potential attacks. SIM swapping attacks, where an attacker gains control of a victim’s phone number by transferring it to a different SIM card, pose a significant risk. Moreover, on older UMTS and GSM networks L2SMS might be susceptible to SS7 attacks exploiting vulnerabilities in the signalling system and even weaker encryption schemes

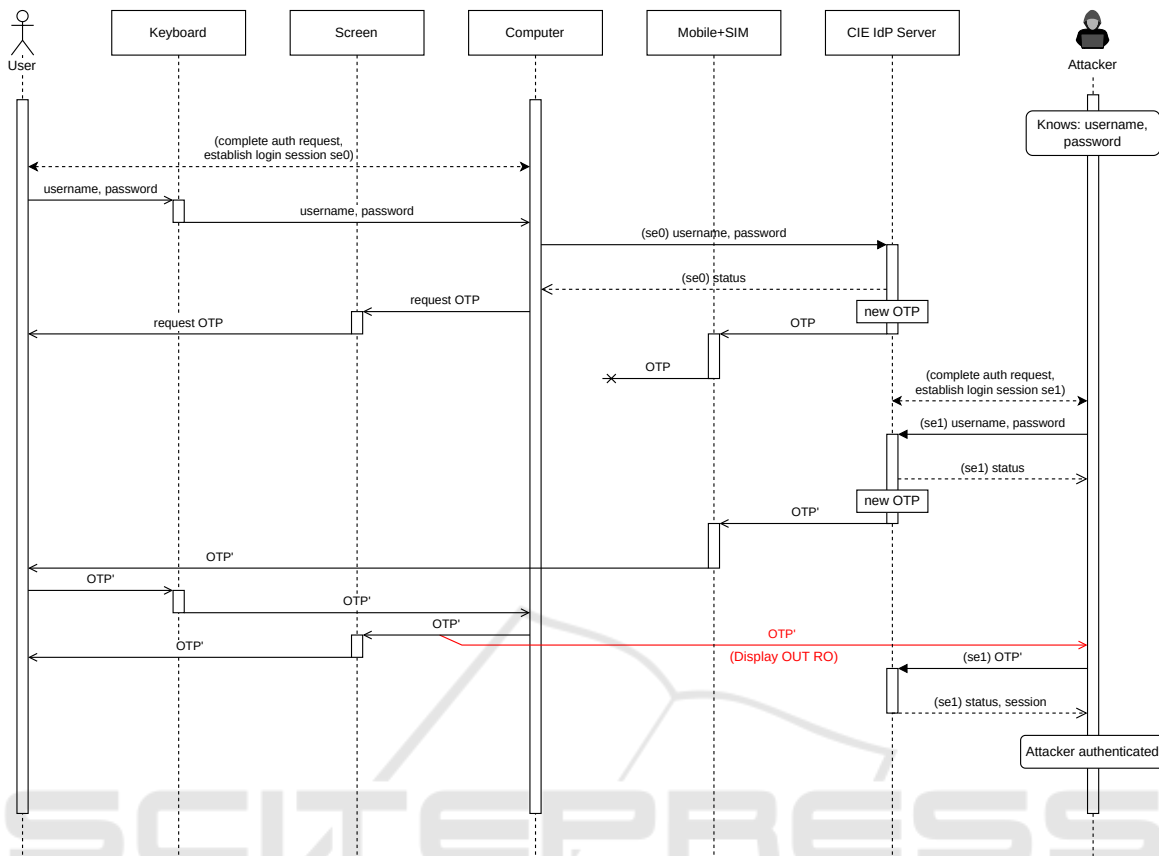


Figure 3: Overview of the L2SMS attack in scenario $\mathcal{A}_{out:RO}^{displ}$ of the attack trace found by ProVerif.

employed in these networks (Ullah et al., 2020).

Interestingly, according to our model, the implementation of L2SMS does not offer an advantage when dealing with malware controlling the computer display. This stems from the fact that, while not requiring users to scan codes displayed on the computer screen (relying solely on SMS communication with the mobile device), the OTP is shown to the user via the display channel. This makes it vulnerable to attacks exploiting compromised computer displays.

Despite the generally stronger security guarantees offered by other level 2 protocols, users currently lack the option to disable L2SMS logins even after registering other CieID authenticator devices. This means that even if users opt for more secure protocols, a compromised L2SMS remains a potential backdoor for attackers. Furthermore, the process of associating new devices for L2APP itself relies on an authentication scheme essentially equivalent to L2SMS, raising concerns about its overall effectiveness as an upgrade.

The most concerning aspect of L2SMS is its potential to undermine the security benefits of other protocols. If an attacker successfully compromises L2SMS, they gain access to the account regardless of

whether other, more robust protocols are also implemented. This essentially negates the added security assurances those protocols provide, exposing the entire system to increased risk.

4 PROTOCOL IMPROVEMENT

In this section we propose an improvement of the L2SMS protocol that addresses the vulnerability on the display channel, discovered in the previous analysis. We call this improved version L2SMS*.

It is important to notice that while MFA using SMS OTP offers an additional layer of security compared to no MFA at all, it has some inherent weaknesses. If the communication channel between the user and the service provider is not properly secured with TLS, attackers could potentially intercept the SMS code during transmission, rendering MFA useless. Similarly, malware or physical access to the user’s mobile phone could allow attackers to steal the incoming SMS code even before the user sees it. Finally, keyloggers installed on the user’s computer

Table 2: Comparison with the improved protocol. See Table 1 for the legend.

Threat Scenario	L2SMS	L2SMS*
	✓	✓
$\mathcal{A}_{in:RO}^{mobile}$	✗	✗
$\mathcal{A}_{out:RO}^{displ}$	✗	✓
$\mathcal{A}_{out:RW}^{displ}$	✗	✓
$\mathcal{A}_{in:RO}^{usb}$	✗	✗
$\mathcal{A}_{io:RO}^{tis}$	✗	✗
PH	✗	✗

could capture any information typed, including the SMS code entered for authentication.

However, all these weaknesses are not easily exploited. Communication channels can be secured by correctly setting up TLS communications (e.g., certificates are properly verified); SMS transmission security depends on the telecommunication operator, and hence it is outside our control; hardware keylogger require physical access to the user’s PC, and the implantation of software keyloggers requires admin-level access to the system.

Therefore, in securing L2SMS we prioritize the display channel. The attack shown in Figure 3 reveals that L2SMS is vulnerable because the computer displays the OTP code on the user’s screen. This exposes the OTP code to attackers with read-only screen access, e.g. using screen sharing software or performing “shoulder surfing”, allowing these attackers to steal the code and gain unauthorized access.

To address this vulnerability, we propose a simple variant of L2SMS, called L2SMS*, without the step where the OTP code is shown on the user’s screen. More formally, L2SMS* is obtained from L2SMS by omitting the unnecessary OTP display output after the user keyboard input.

The verification of this protocol yields the results presented in Table 2. Consistent with expectations, the previously identified vulnerability has been effectively mitigated: an attacker with read-only or read-write access to the display channel cannot gain unauthorized access. Additionally, no further concerns have been identified up to our model. Therefore, the verification process yields a positive outcome.

From an implementation point of view, there are several strategies to prevent attackers from observing the OTP as the user types it. One approach is to completely avoid providing any feedback during the input process. This means the user will not see any characters displayed on the screen, offering no clues to the attacker. Another option is to echo asterisks or bullets instead of the actual digits as they are typed. This

provides some basic user confirmation while keeping the actual code hidden from potential observers.

5 CONCLUSIONS

In this work we have studied the CIE architecture and its multi-factor authentication protocol based on SMS. First we have given an interface architecture for the CIE authentication process. Over this architecture we have defined a new threat model which takes into account threats such platform compromised by malware (at various degrees), wrong user behaviour (e.g., by phishing), and message delay or dropping. Then, we have carried out a systematic and extensive security analysis of the level 2 CIE multi-factor authentication via SMS OTP (L2SMS) under a multitude of attack scenarios. This analysis, carried out using the ProVerif tool, has shed light on potential weaknesses within the specific protocol. In particular, we have found that L2SMS is vulnerable to an attacker with mere read-only access to the user’s computer display. From the generated attack trace, we have suggested a viable improvement of the protocol, L2SMS*, to strengthen the security of the system. The formal analysis of the improved protocol shows that the vulnerability has been successfully tackled.

Our exploration has provided valuable insights and paves the way for further in-depth analysis of the remaining protocols within the system. Moreover, the methodology we have followed in this work can be applied to the analysis of other digital identity tools.

Related Work. Multi-factor authentication (MFA) protocols play a crucial role in securing online identities and transactions. A significant body of research has explored various aspects of MFA security.

(Jacomme and Kremer, 2021) present an extensive formal analysis of MFA protocols, with an automated, systematic generation of all combinations of threat scenarios and using ProVerif for automated protocol analysis. Their work highlights the importance of rigorous verification to identify potential vulnerabilities. Our approach builds upon this foundation by proposing an interface-based threat model specifically tailored to CIE’s architecture.

The survey by (Sharif et al., 2022) examines technological trends in electronic identity schemes compliant with the eIDAS regulation. It emphasizes the growing adoption of hardware security tokens as a second authentication factor, aligning with the strong security posture of CIE’s Level 3 authentication. Our work focuses on the Level 2 protocols, which are more widely used in current applications.

The survey by (Sinigaglia et al., 2020) examines the adoption of MFA for online banking in practice. In particular, they report that the usage of out-of-band authentication via SMS has been deprecated by the guidelines provided by security organizations (e.g., (NIST, 2020)), since many attacks have targeted this authentication method to acquire sensitive data for MFA. Our analysis of L2SMS vulnerabilities confirms this trend, and contributes to a broader understanding of potential security risks associated with SMS-based MFA in real-world deployments.

Future Work. We are currently working on formalizing and verifying all the other multi-factor protocols available for level 2 and level 3 CIE authentication. Looking forward, several promising research directions emerge. Firstly, we would like to evaluate the resistance of smartphone-based login protocols towards attacks, considering the unique security models of popular mobile platforms. Secondly, we would like to assess the vulnerability of available CIE smartcards to side-channel attacks, where unintended information leakage occurs. Finally, verifying the adherence to security best practices of the implementation of official CIE authentication, to minimize potential attack vectors, can be an interesting research topic. By pursuing this research, we can continue to strengthen the security and reliability of CIE multi-factor authentication, safeguarding sensitive user data and transactions in the digital realm.

On a different direction, the interface-access threat model could be applied to other contexts. A particularly intriguing case is that of *containers*, since they have already a well-defined notion of interface through which they can interact. We plan to integrate the interface-access threat model to formal models of container compositions, such as (Burco et al., 2020).

ACKNOWLEDGEMENTS

This research has been partially supported by the Department Strategic Project of the University of Udine within the Project on Artificial Intelligence (2020-25), and the project SERICS (PE00000014) under the NRRP MUR program funded by the EU-NGEU.

We have contacted the Italian National Cybersecurity Authority (ACN <https://www.acn.gov.it>) about the results of this research. Moreover, during our analysis we discovered an Insecure Direct Object Reference vulnerability that allowed for second-factor bypass (level 1 downgrade attack) during CIE authentication. We want to thank the ACN for promptly patching the system after receiving our report.

REFERENCES

- Bhargavan, K., Blanchet, B., and Kobeissi, N. (2017). Verified models and reference implementations for the TLS 1.3 standard candidate. In *2017 IEEE Symposium on Security and Privacy (S&P)*, pages 483–502.
- Blanchet, B. et al. (2016). Modeling and verifying security protocols with the applied pi calculus and ProVerif. *Foundations and Trends® in Privacy and Security*, 1(1-2):1–135.
- Burco, F., Miculan, M., and Peressotti, M. (2020). Towards a formal model for composable container systems. In *Proceedings of the 35th Annual ACM Symposium on Applied Computing*, pages 173–175.
- EU Parliament and Council (2014). Regulation (EU) no 910/2014. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv:OJ.L_.2014.257.01.0073.01.ENG.
- Holgers, T., Watson, D. E., and Gribble, S. D. (2006). Cutting through the confusion: A measurement study of homograph attacks. In *USENIX Annual Technical Conference*, pages 261–266.
- Italian Ministry of the Interior (2024). Carta d’identità elettronica official website. <https://www.cartaidentita.interno.gov.it/en/home/>.
- Jacomme, C. and Kremer, S. (2021). An extensive formal analysis of multi-factor authentication protocols. *ACM Transactions on Privacy and Security (TOPS)*, 24(2):1–34.
- Kernighan, B. and Ritchie, D. (1977). The m4 macro processor. Technical report, Bell Laboratories, NJ.
- Lockhart, H. and Campbell, B. (2008). Security assertion markup language (SAML) v2.0 technical overview. *OASIS Committee Draft*, 2:94–106.
- NIST (2020). Digital identity guidelines: Authentication and lifecycle management. Special Publication 800-63B. <https://doi.org/10.6028/NIST.SP.800-63b>.
- Sharif, A., Ranzi, M., Carbone, R., Sciarretta, G., and Ranise, S. (2022). SoK: A survey on technological trends for (pre) notified eIDAS electronic identity schemes. In *Proceedings of the 17th International Conference on Availability, Reliability and Security*, pages 1–10.
- Sinigaglia, F., Carbone, R., Costa, G., and Zannone, N. (2020). A survey on multi-factor authentication for online banking in the wild. *Computers & Security*, 95:101745.
- Ullah, K., Rashid, I., Afzal, H., Iqbal, M. M. W., Bangash, Y. A., and Abbas, H. (2020). Ss7 vulnerabilities—a survey and implementation of machine learning vs rule based filtering for detection of ss7 network attacks. *IEEE Communications Surveys & Tutorials*, 22(2):1337–1371.
- Van Eeden, R., Paier, M., and Miculan, M. (2024). Analysis of CIE Level 2 SMS OTP authentication protocol in ProVerif. Available at <https://doi.org/10.5281/zenodo.10657295>.