Fingerprinting the eSIM Ecosystem: An Empirical Study of Provisioning Infrastructures

Matteo Fanfarillo*, Lorenzo Valeriani*, Francesco Gringoli†, Giuseppe Bianchi*
*Department of Electronic Engineering, CNIT & University of Rome Tor Vergata, Rome, Italy
Email: matteo.fanfarillo, lorenzo.valeriani@cnit.it, giuseppe.bianchi@uniroma2.it
†Department of Computer Engineering, University of Brescia, Brescia, Italy
Email: francesco.gringoli@unibs.it

***Abstract*—The embedded SIM (eSIM) is rapidly reshaping how mobile devices connect to networks, enabling seamless activation and provisioning without the need for physical SIM cards. Unlike traditional SIMs, the eSIM ecosystem introduces a significantly more complex landscape, where the direct relationship between customers and Mobile Network Operators is mediated by multiple intermediate stakeholders and infrastructures. In this work, our objective is to examine this landscape and shed light on the actors involved in the eSIM provisioning process, exploring how they interact and how their implementations may be related. To this end, we performed an Internet-wide empirical analysis of eSIM provisioning infrastructures, focusing on fingerprinting real-world implementations of Subscription Manager Data Preparation Plus servers. By analyzing certificates, endpoint behaviors, and implementation patterns, we investigate whether server implementations conform to standard specifications or deviate from them, and whether credentials and infrastructures are reused across different actors in the ecosystem. Our preliminary findings suggest that 50% of online servers rely on only six shared implementations and certificates, offering early insights into the consolidation of the eSIM provisioning landscape and laying the ground for a deeper understanding of its actors and relationships.**

***Index Terms*—eUICC, LPA, SM-DP+, RSP, fingerprinting, emulators.**

I. Introduction

The term *embedded SIM (eSIM)* refers to a virtual Subscriber Identity Module (SIM) card that fundamentally differs from traditional SIM cards. In fact, eSIMs eliminate the need for users to physically obtain and insert a new SIM card when changing Mobile Network Operators (MNOs). Instead, users receive an eSIM profile electronically, typically by email, which can be easily activated on their devices by scanning a Quick Response (QR) code. This simplified process significantly improves user convenience compared to visiting MNO stores in person, contributing to the increasing adoption of eSIM technology.

The concept of eSIM was first introduced in 2010 [1] following the emergence of devices based on the Internet of Things (IoT), which require machine-to-machine (M2M) communication over the air to function effectively [2], [3]. Since these devices rely on automated network connectivity management, traditional SIM cards would be significantly limiting. Later, in 2016, the Global System for Mobile Communications Association (GSMA) released the first official document detailing the technical specifications of the consumer variant of eSIM provisioning [4]. This variant addresses all mobile devices that are directly controlled by the end user, such as smartphones and tablets. Since then, the adoption of eSIM technology has gradually accelerated. Operators around the world soon began to offer eSIM tariff plans, called *profiles*, and device manufacturers started to integrate eSIM support into their latest phone models. In recent years, eSIM technology has been predominantly used for temporary needs, such as international travel. In the long term, eSIMs are expected to completely replace traditional SIM cards. According to *Research and Markets*, the global eSIM market will expand from 13.78 billion dollars in 2024 to 23.57 billion dollars by 2030 [5].

However, the transition from traditional SIMs to eSIMs also reshapes the overall provisioning ecosystem. In the legacy model, the relationship between the customer and the MNO was primarily direct. With eSIMs, this process is now mediated by several intermediate entities responsible for securely generating, storing, and delivering digital profiles. At the core of this workflow is the *Subscription Manager Data Preparation+ (SM-DP+)* server, which prepares and securely transmits encrypted eSIM profiles to user devices during activation. These servers are typically operated by MNOs, device vendors, or third-party service providers, making the ecosystem highly distributed and heterogeneous. Understanding who operates which SM-DP+ servers and how implementations are shared or reused among stakeholders is therefore essential to map the real-world eSIM landscape.

Given this context, this work focuses on analyzing the characteristics and fingerprinting of SM-DP+ servers involved in eSIM provisioning, with the goal of understanding their behavior, identifying groups that share the same implementation, and obtaining preliminary insights into the structure of the ecosystem. Specifically, this research aims to address the following questions:

- **RQ1.** How are SM-DP+ servers used and mapped to different Mobile Network Operators and other stakeholders within the eSIM provisioning ecosystem?
- **RQ2.** To what extent are SM-DP+ implementations (and configurations) shared, reused, or consolidated between different entities, and what insights can be derived from these relationships?

ISBN: 978-3-903176-79-9

To address these questions, we develop dedicated tools and methodologies for fingerprinting SM-DP+ server implementations in the wild, combining certificate analysis, endpoint behavior inspection, and protocol-level interaction patterns. Our approach enables us to identify relationships between stakeholders and uncover reuse trends across provisioning infrastructures.

The remainder of this paper is organized as follows. Section II introduces the fundamentals of eSIM technology. Section III examines the relationships between SM-DP+ servers, SM-DP+ providers, and MNOs, as well as the methodology used to derive this information. Section IV presents key observations on the implementations of real-world SM-DP+ servers and provides a global profiling of their behavior and configuration. Section V reviews the most relevant related works, while Section VI concludes the paper.

II. Background

A. eSIM Architecture

Throughout this work, we focus exclusively on the consumer variant of eSIM provisioning, which is the most familiar to the average end user of everyday mobile devices. In this context, the eSIM ecosystem consists of several key entities that participate in the provisioning process. These are described hereafter [6].

- The *embedded Universal Integrated Circuit Card (eUICC)* is a hardware chip embedded in mobile devices that contains the eSIM software. It is remotely programmable and capable of storing and managing multiple eSIM profiles.
- The *Local Profile Assistant (LPA)* is an application running on the user's device that manages the eSIM profiles. It is responsible for the download, installation, and deletion of profiles. In addition, it acts as an intermediary, allowing communication between the eUICC and the provisioning server. Together, eUICC and LPA form what is referred to as the *user agent*.
- The *Mobile Network Operator (MNO)* is the telecom company that offers eSIM profiles, which contain the subscription data required to enable network connectivity for end users.
- The *Subscription Manager Data Preparation+ (SM-DP+)* server is the core server component responsible for securely delivering eSIM profiles to user agents via a defined message exchange protocol. It handles profile encryption, authentication, and download operations.
- The *Subscription Manager Discovery Service (SM-DS)* server is another server-side component that maintains a registry of available profiles for a given eUICC and facilitates their discovery [7].

The protocol that governs the functioning of eSIM and ensures seamless communication between these entities is called *Remote SIM Provisioning (RSP)* [6]. The RSP protocol secures communication between its components using the *Transport Layer Security (TLS)* protocol at the transport layer of the TCP/IP stack [8]. Only TLSv1.2 and TLSv1.3 are allowed for exchanges involving the LPA and the SM-DP+ server. Furthermore, a pairwise secure channel is mandated for communication between the eUICC and the LPA. This channel, which ensures the confidentiality and authentication of the message, typically relies on TLS.

B. Interaction Steps in RSP

The interaction takes place in four distinct phases [8].

- *Profile Ordering*: The MNO requests the SM-DP+ server to prepare an eSIM profile. The server responds with an *Activation Code*, which includes an identifier within the context of the MNO and the server: the *Matching ID*.
- *Download Initialization*: The MNO forwards the Activation Code to the LPA to begin the profile download process.
- *Common Mutual Authentication*: This phase involves authentication between the eUICC and the SM-DP+ server, with the mediation of the LPA.
- *Profile Download and Installation*: The server sends the encrypted profile to the LPA, which then forwards it to the eUICC for final installation.

1) Common Mutual Authentication steps: The main steps of the Common Mutual Authentication procedure are illustrated in Figure 1 (top).

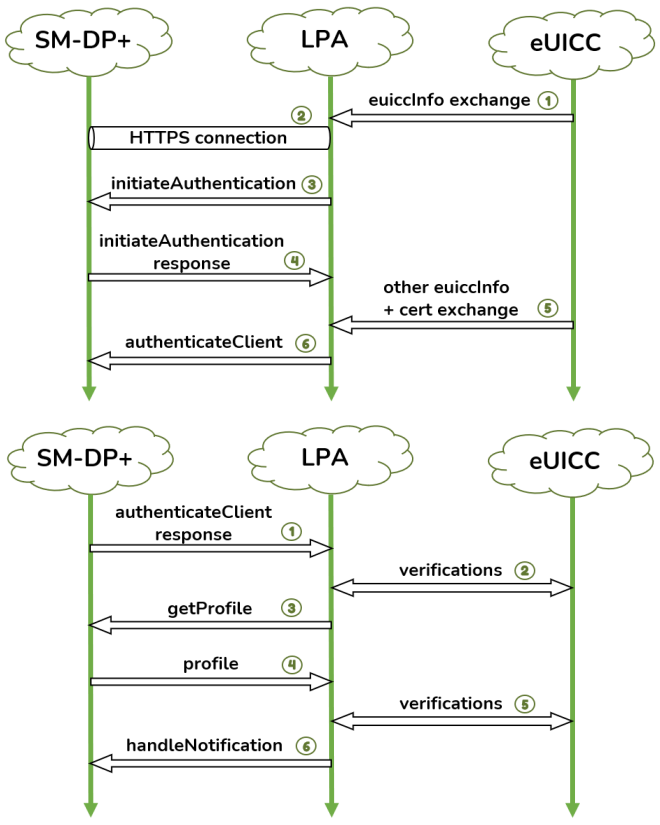

Fig. 1. Common Mutual Authentication procedure (top) + Profile Download and Installation procedure (bottom).

Following step (3), the SM-DP+ server evaluates the content of the *initiateAuthentication* message, particularly checking if it supports at least one of the RSP certificate chains presented by the user agent. Similarly, after step (4), the user agent validates the server response, including its certificate. Lastly, upon receiving the *authenticateClient* message in step (6), the SM-DP+ server performs its validation. If successful, the eUICC is authenticated.

This procedure remains identical when an SM-DS server replaces the the SM-DP+ server, as both utilize the Common Mutual Authentication process [6].

2) Profile Download and Installation steps: The main steps of the Profile Download and Installation procedure are shown in Figure 1 (bottom).

At the beginning of step (1), the SM-DP+ server verifies the availability of a profile corresponding to the Matching ID of the *authenticateClient* message. During step (2), the LPA and eUICC validate the profile against the so called *Profile Policy Rules (PPRs)* and verify the certificates and signatures. In step (4), the SM-DP+ server validates the *getBoundProfilePackage* message and responds with the *Bound Profile Package (BPP)*, which is the profile encrypted through a shared key. Finally, in step (5), LPA and eUICC complete the final checks and successfully install the profile.

This procedure is specific to SM-DP+ servers, as SM-DS servers do not support profile download [6].

C. Certificate Chain in RSP

Within the RSP protocol, the GSMA defines a trust hierarchy among the key entities involved. At the top of the hierarchy is a trusted Root Certification Authority, referred to as the *Certificate Issuer (CI)*. In the initial version of the protocol [4], the CI issued certificates directly to SM-DP+ servers, SM-DS servers, and *eUICC Manufacturers (EUMs)*, who in turn embedded certificates in the eUICCs during manufacturing. In the current specification [6], the trust model allows for the inclusion of intermediate entities acting as subordinate Certificate Authorities, both under the CI and under the EUM.

At the RSP level, each entity holds a unique certificate, with the exception of the SM-DP+ server, which has an authentication certificate (`CERT.DPauth.SIG`) and a profile-binding certificate (`CERT.DPpb.SIG`). These ensure the authenticity of the server and the integrity of the profiles during deployment.

To maintain the highest possible level of security, different SM-DP+ servers should use different private keys and certificates. This segregation mitigates the risks associated with vulnerabilities such as DROWN, which exploit shared secrets on servers that use versions of TLS older than TLSv1.2 [9].

III. SM-DP+ Providers Analysis

A. Hostnames retrieval

To address **RQ1**, the first step is to identify all SM-DP+ servers around the world, enabling comprehensive large-scale testing. However, real-world SM-DP+ servers do not follow standardized URL patterns [10]. To overcome this, we combined two approaches: `(i)` purchasing eSIM profiles from multiple MNOs and extracting SM-DP+ hostnames from confirmation emails, and `(ii)` leveraging recently published online datasets that list hostnames on a global scale [11]–[13]. The resulting data set comprises 212 real-world SM-DP+ hostnames, 5 SM-DS hostnames, and 9 test SM-DP+ hostnames, for a total of 226.

B. Tools development

To determine which SM-DP+ provider operates each hostname, discover relationships between hostnames and IP addresses, and map MNOs to their respective SM-DP+ providers, we developed a module called `SM-DP+ Mapping Tools`. It internally performs DNS queries, enabling seamless large-scale analysis.

As a preliminary step, it is necessary to find out the number and identity of SM-DP+ providers responsible for the 212 real-world SM-DP+ hostnames. To this end, the *organizationName* field of the TLS certificate proves useful, as it directly indicates the SM-DP+ provider operating the server. Extracting such certificates requires a more sophisticated tool, which led us to develop the `eUICC+LPA Emulator` module. This emulator reproduces the behavior of a user agent by integrating both eUICC and LPA functionalities. It replicates, according to the specifications [6], the message exchange over TLS between the LPA and the SM-DP+ server during the Common Mutual Authentication and Profile Download and Installation phases, while also validating the response messages and authenticating the server. This makes the emulator particularly powerful, as it not only outputs server responses but also reveals the TLS certificate of the server obtained during the handshake and the `CERT.DPauth.SIG` certificate following the *initiateAuthentication* response.

We are also finalizing the development of the dual module to the `eUICC+LPA Emulator`, named `SM-DP+ Emulator`, which provides an executable for delivering profiles during the Common Mutual Authentication and Profile Download and Installation phases. However, this module falls outside the scope of this paper. Both emulators were developed using Python, a programming language selected for its robust libraries for message encoding/decoding (*asn1tools*, *base64*) and cryptographic operations (*asn1crypto*, *cryptography*, *hashlib*).

A scheme representing the architecture of our work is reported in Figure 2.

Validation: The `eUICC+LPA Emulator` was validated by communicating with the open source Osmo-smdpp server [14]. Validation was achieved when the emulator successfully exchanged the following messages in sequence: *initiateAuthentication*, *authenticateClient*, *getBoundProfilePackage*, and *handleNotification*. This was possible because Osmo-smdpp supports two test certificate chains provided by the GSMA [15], complete with public and private keys. The first chain, BRP, uses the Brainpool standard elliptic curve, while the second, NIST, employs an elliptic curve approved by the National Institute of Standards (NIST).

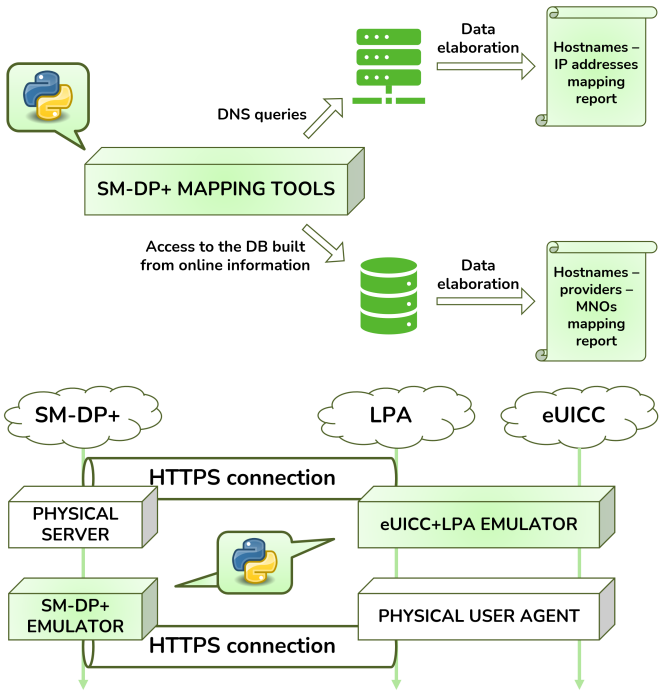

Fig. 2. A scheme describing how the `SM-DP+ Mapping Tools` module works (top) and how the two emulators are used (bottom).

C. SM-DP+ providers and MNOs retrieval

Through the analysis of the TLS certificates retrieved using the `eUICC+LPA Emulator`, we identified 30 SM-DP+ providers responsible for the 212 real-world SM-DP+ hostnames. For each provider, using the `SM-DP+ Mapping Tools`, we analyzed the number of SM-DP+ hostnames associated with it, the number of physical end hosts it manages (as identified by their IP addresses), the number of MNOs it serves, and the number of countries involved, where each country corresponds to the origin of the respective MNOs. In total, we identified 69 physical end hosts and 286 MNOs, also with the help of online datasets [11]–[13], covering 88 different countries. All this information is summarized in Table I. It should be noted that the table contains a 31st entry, corresponding to the operators *SIMHUB China* (associated with the hostname "rsp.simhub.cn") and *eSIM.me Germany* (associated with the hostname "rsp.esim.me"). The related SM-DP+ servers, when contacted, do not return any response, resulting in a communication deadlock. This behavior prevents the retrieval of the TLS certificates of the servers and the identification of the associated SM-DP+ provider. Regarding the size of the reference sample, it is nearly exhaustive, as recent sources estimate the existence of over 260 MNOs/MVNOs that support eUICC and more than 25 RSP providers worldwide [16]. Therefore, the present work can be considered as a comprehensive profiling of real-world provisioning infrastructures within the eSIM ecosystem.

Thanks to the `SM-DP+ Mapping Tools`, it is also possible to determine how many SM-DP+ providers each operator relies on. Only one MNO (*Plus Poland*) relies on nine providers; 4 MNOs (*3 Austria*, *Proximum Belgium*, *Sun Mobile Hong Kong*, and *Vodafone Netherlands*) rely on five providers each; 3 MNOs (*China Mobile Hong Kong*, *Orange France*, and *Starhub Singapore*) rely on four providers each; 5 MNOs (*3 Hong Kong*, *China Unicom Hong Kong*, *Dtac Trinet Thailand*, *SoftBank Japan*, and *T-Mobile USA*) rely on three providers each; 41 MNOs rely on two providers each; the remaining 232 rely on a single provider.

Consequently, the `SM-DP+ Mapping Tools` suggests a many-to-many relationship between MNOs and SM-DP+ providers, highlighting the complexity of stakeholder interconnections within the eSIM ecosystem. Specifically, we observed that SM-DP+ hostnames associated with different MNOs often resolve to the same IP address, indicating that multiple operators may share a common SM-DP+ infrastructure. Conversely, some MNOs, such as *Vodafone Germany*, rely on multiple SM-DP+ providers.

TABLE I
SUMMARY OF THE IDENTIFIED SM-DP+ PROVIDERS. FOR EACH PROVIDER, WE REPORT THE NUMBER OF SM-DP+ HOSTNAMES, IP ADDRESSES, MNOS, AND THE NUMBER OF COUNTRIES WHERE THE MNOS ARE BASED.

SM-DP+ provider	#hostnames	#IPs	#MNOs	#countries
Beijing Watchdata	1	1	3	1
Bharti Airtel Limited	1	1	1	1
CAICT	2	2	2	1
CUCA	1	1	1	1
Eastcompeace Tech.	7	12	21	13
G+D Mobile Security	47	3	49	28
GlobalSign	1	1	1	1
IDEMIA	26	6	49	27
Invigo Offshore SAL	2	1	6	5
Kigen	2	2	2	2
Let's Encrypt	3	1	3	1
Linksfield	3	1	7	6
Monty	8	1	8	7
Nokia Solutions and Nertworks Oy*	1	1	1	1
Nordic eSIM	2	2	4	4
NovaCard JSC	8	2	11	2
Oasis Smart SIM Europe	2	2	17	14
Protahub	2	1	2	1
Redtea Mobile	3	2	29	16
Reliance Jio Infocom Limited	1	1	1	1
RiPSIM Technologies	1	2	2	1
Saudi Telecom Company	4	4	4	1
Simartis Telecom	1	1	1	1
Thales SA	54	5	70	38
TP Global Operations Limited	2	2	20	15
Valid USA	19	2	19	15
Vodafone Idea Limited	1	1	1	1
Worz Media	3	4	29	17
Wuhan Tianyu Information Industry	1	1	3	3
XH Smart Tech	1	1	1	1
n.d.	2	2	2	2

*It has been unreachable since November 2025.

A notable observation from Table I is that the DNS queries performed by the `SM-DP+ Mapping Tools` revealed a total of 69 IP addresses, corresponding to 69 distinct end hosts operated by 30 SM-DP+ providers. However, half of these end hosts are concentrated within only six providers: Eastcompeace Technology, G+D Mobile Security, IDEMIA, Saudi Telecom Company, Thales SA, and Worz Media. Even more striking is the distribution of MNOs: almost 25% rely on Thales SA, slightly more than 17% on IDEMIA, and slightly over 17% on G+D Mobile Security. These findings suggest a potential centralization of the eSIM ecosystem around a few key providers, which could have significant implications in terms of resilience and competition.

IV. SM-DP+ Fingerprinting Analysis

A. Methodology

Other tools development: Fingerprinting is an analysis technique used to profile a set of end hosts and classify their behavior. This provides insights into how many distinct configurations – and thus how many trust anchors – exist within the sample of end hosts. To address **RQ2**, our methodology consists of forcing the `eUICC+LPA Emulator` to establish a wide range of communication scenarios. In each scenario, the emulator customizes the request messages sent to the real counterpart in order to study the implementation of SM-DP+ servers worldwide, both when communication follows the specifications [6] and when malformed or inconsistent requests are introduced. To enable this, we extended the emulator into a more powerful module, named `eUICC+LPA Tester`.

The `eUICC+LPA Tester` is supported by a Bash script that automatically executes the test cases. The script accepts as input the type of server to be tested. In addition to real-world SM-DP+ servers, the following types of server can also be tested: `a)` test SM-DP+ servers, which rely on test certificate chains; and `b)` SM-DS servers, which support the Common Mutual Authentication procedure but not the Profile Download and Installation phase [6]. The script iterates over all test cases and all known servers of the selected type. For each *test case–server* pair, it runs the `eUICC+LPA Tester`. During execution, all exchanged messages are logged in dedicated output files.

A Python script was also developed to analyze the output logs produced by the `eUICC+LPA Tester`. It classifies servers into groups based on their responses to each test case and generates a summary file. As in the case of the emulators, we are also finalizing the development of the server-side counterpart, named `SM-DP+ Tester`, which is however out of scope for this paper.

A schematic overview of the `eUICC+LPA Tester` workflow and the collection of results is shown in Figure 3.

Experiment setup: Our SM-DP+ fingerprinting analysis was carried out on two levels: `(i)` a high-level classification of the responses provided by the servers in each communication scenario; and `(ii)` a detailed inspection of the exact formatting of the response messages to better understand whether SM-DP+ servers from different providers share the same implementation. In addition, we analyzed the certificates exposed by the servers, both to check whether they are shared between SM-DP+ servers belonging to different providers and to study their validity periods.

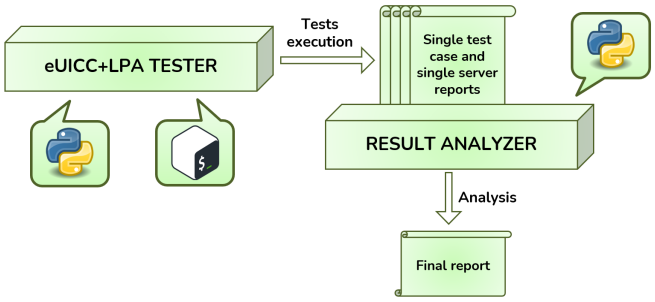

Fig. 3. Workflow of the `eUICC+LPA Tester` module and retrieval of results.

All testing was limited to the initial request messages of the Common Mutual Authentication procedure because real-world SM-DP+ and SM-DS servers rely on the official GSMA certificate chain, for which the private keys are not accessible. As a result, these servers are expected to reject the fake certificates presented by the `eUICC+LPA Tester` (or `eUICC+LPA Emulator`) in the second request and terminate the communication.

To analyze server behavior and perform fingerprinting, we designed 22 test cases grouped into four categories. The first, summarized in Table II, covers standard scenarios. The second, shown in Table III, includes cases where the *authenticateClient* message is deliberately tampered, while the third, shown in Table IV, does the same for the *initiateAuthentication* message. The fourth category, in Table V, evaluates message exchanges in an invalid order. Since it was not feasible to test all servers beyond the Common Mutual Authentication phase, the same test suite was applied to both SM-DP+ and SM-DS servers.

With reference to test case 1 in Table II, we observed that all servers support only TLS version 1.2 or higher. As a result, they are not susceptible to downgrade attacks or other transport-layer vulnerabilities within the TCP/IP stack.

B. Implementations Analysis

Real-world SM-DP+ servers: In this section, we describe the behavior of SM-DP+ servers (grouped by provider) across the test cases listed in Tables II, III, IV, V. The first providers to be tested were Eastcompeace Technology, IDEMIA, Redtea Mobile, Thales SA, and Worz Media, and are discussed first.

TABLE II
List of the standard test cases defined for SM-DP+ servers.

ID	Description
0	The client emulator behaves exactly as specified.
1	Attempting to establish a connection while restricting the use of TLS versions 1.3 and 1.2.

TABLE III
LIST OF THE TEST CASES IN WHICH *authenticateClient* HAS BEEN TAMPERED.

ID	Description
2	Sending *authenticateResponseError* instead of *authenticateResponseOk*.
3	Using a client-side certificate chain different from the default one.
4	Incorrect calculation of the signature (*euiccSignature1*).
5.1	Unexpected Specification Version Number (SVN), e.g., 0.0.0 (*euiccInfo2* subfield).
5.2	Unexpected values for *euiccCiPKIdListForVerification* and *euiccCiPKIdListForSigning* while keeping the default certificate chain (*euiccInfo2* subfield).
5.3	Setting *sasAcreditationNumber* to an empty string (*euiccInfo2* subfield).
6.1	Removing the *imei* from *ctxParams1* subfield.
6.2	Inserting an invalid Matching ID in *ctxParams1* subfield.
7.1	Invalid Transaction ID within the *euiccSigned1* field.
7.2	Directly setting an invalid *transactionId* within *authenticateResponseOk*.
8	Invalid *smdpAddress* within the *euiccSigned1* field.
9	Invalid *serverChallenge* within the *euiccSigned1* field.

TABLE IV
LIST OF THE TEST CASES IN WHICH *initiateAuthentication* HAS BEEN TAMPERED.

ID	Description
10	Malformed *euiccChallenge* (e.g., `0x00`).
11	Invalid *smdpAddress*.
12.1	Unexpected SVN, e.g., 0.0.0 (*euiccInfo1* subfield).
12.2	Unexpected values for *euiccCiPKIdListForVerification* and *euiccCiPKIdListForSigning* while keeping the default certificate chain (*euiccInfo1* subfield).

TABLE V
LIST OF THE TEST CASES IN WHICH THE MESSAGES ARE OUT OF ORDER.

ID	Description
13.1	Initiating the Common Cancel Session Procedure from the first message.
13.2	Initiating the Common Cancel Session Procedure from the second message.
14.1	Sending *getBoundProfilePackage* as the first message.
14.2	Sending *getBoundProfilePackage* as the second message.

`1) Eastcompeace Technology`: servers fall into two groups: three of them provide empty responses, blocking the execution of the protocol; the other nine behave more typically, but do not detect a malformed *euiccChallenge*. `2) IDEMIA`: servers return "invalid EUM Certificate" in Test Case 1 and detect all misconfigurations in *initiateAuthentication* message. `3) Redtea Mobile`: similar to IDEMIA, with some implementation differences. `4) Thales SA`: servers return "execution error" in Test Case 1 and detects all anomalies in *initiateAuthentication* message. `5) Worz Media`: servers return "invalid EUM Certificate" in Test Case 1 but fail to detect a malformed *euiccChallenge*. `6) Beijing Watchdata`: server returns "invalid Matching ID" error in Test Case 1 and in all test cases in which the *authenticateClient* message is treated, but fails to validate malformed *euiccChallenge*. `7) Bharti Airtel Limited`: similar to Thales SA, with minor implementation differences. `8) CAICT`: one server reports "unsupported security configuration", the other "unknown profile ICCID", suggesting profiles must be pre-linked to a specific eUICC. `9) CUCA`: typically returns "unsupported security configuration", but also detects incorrect server addresses and unsupported SVN values in the *initiateAuthentication* message. `10) G+D Mobile Security`: similar to Thales SA but differs in verification order. `11) GlobalSign`: does not detect a malformed *euiccChallenge*. `12) Invigo Offshore SAL`: returns *HTTP_BAD_REQUEST*. `13) Kigen`: does not detect a malformed *euiccChallenge* and invalid certificate chains. `14) Let's Encrypt`: consistently returns an "invalid hostname" error, maybe due to misconfiguration. `15) Linksfield`: consistently returns empty messages. `16) Monty`: similar to IDEMIA but fails to detect an incorrect *smdpAddress* and invalid certificate chains. `17) Nokia Solutions and Networks Oy`: same as Workz Media. `18) Nordic eSIM`: similar to IDEMIA but fails to detect a malformed *euiccChallenge*. `19) Novacard JSC`: either same as Workz Media or returns an "invalid hostname" error (indicating misconfiguration). `20) Oasis Smart SIM Europe`: appears to accept unofficial certificate chains, as Test Case 3 yields "invalid Matching ID" rather than "invalid EUM Certificate", although we were unable to demonstrate this behavior. `21) Protahub`: similar to Redtea Mobile, with implementation differences. `22) Reliance Jio Infocom Limited`: same as Barthi Airtel Limited. `23) RiPSIM Technologies`: same as Worz Media. `24) Saudi Telecom Company`: same as Thales SA. `25) Simartis Telecom`: processes the *initiateAuthentication* message properly but returns an empty response after *authenticateClient* message. `26) TP Global Operations Limited`: similar to Redtea Mobile but with a different verification order. `27) Valid USA`: returns "invalid field value" in Test Case 1 and does not validate the *euiccChallenge*. `28) Vodafone Idea Limited`: similar to Thales SA but fails to detect a malformed *euiccChallenge* and an invalid SVN. `29) Wuhan Tianyu Information Industry`: consistently returns empty messages. `30) XH Smart Tech`: fails to establish a connection with the emulator. `31)` Last two SM-DP+ servers: do not respond at all.

From this analysis, three key observations can be made. `a)` Three SM-DP+ providers, which are CAICT, Eastcompeace Technology, and NovaCard JSC, operate heterogeneous sets of SM-DP+ servers. Despite being managed by the same provider, these servers exhibit not only different implementations but also diverse behaviors under specific conditions. `b)` The SM-DP+ servers managed by Nokia Solutions, Worz Media, and one NovaCard JSC server share the same RSP protocol implementation; however, a finer-grained fingerprinting analysis — based on spacing, indentation, and HTTP headers — reveals that servers belonging to different providers do not share an identical fingerprint, due to differences in their HTTP headers. The same applies to Barthi Airtel, Reliance Jio Infocom Limited, Saudi Telecom Company, and Thales SA. `c)` Figure 4 provides a visual summary of the observed behavior of the 69 SM-DP+ physical end hosts across the dataset.

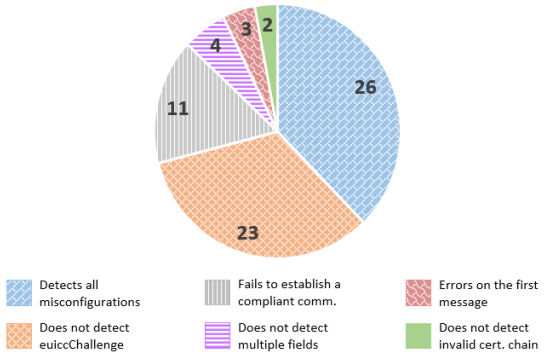

Fig. 4. Behaviors of the 69 identified real-world SM-DP+ end hosts.

SM-DS servers: Among the five identified SM-DS servers, four share the same implementation: in most test cases, they return a *HTTP_INTERNAL_ERROR*, suggesting that error handling mechanisms for many scenarios are not yet fully defined.

The fifth SM-DS server does not use the official GSMA certificate chain: it relies on a test certificate chain managed by Google (for which we do not have access to the private keys). Regardless of the test case, it always returns a generic error at the RSP protocol level.

Test SM-DP+ servers: Among the nine identified test SM-DP+ servers, only two (including Osmo-smdpp) allow for successful communication. The remaining servers either support only the Google-managed certificate chain or consistently respond with *HTTP_BAD_REQUEST*.

An interesting observation is that one particular test server blacklists the user agent (i.e. the emulator) after six consecutive failed connection attempts, preventing further communication. This never happens with real-world SM-DP+ servers, despite the potential effectiveness of blacklists in mitigating the impact of faulty or malicious user agents.

C. Certificates Analysis

Certificate sharing: Our analysis shows that only real-world SM-DP+ servers operated by the same provider can share the same `CERT.DPauth.SIG` or TLS certificate. In contrast, a single end host may expose multiple `CERT.DPauth.SIG` and TLS certificates. For example, in the case of NovaCard JSC, we identified two SM-DP+ servers with distinct IP addresses (i.e., two end hosts) exhibiting three different `CERT.DPauth.SIG` certificates. Similarly, G+D Mobile Security servers were found to operate under two distinct IP addresses while relying on five different TLS certificates. In general, each SM-DP+ provider can be associated with an arbitrary number of IP addresses (x), `CERT.DPauth.SIG` certificates (y), and TLS certificates (z). Importantly, certificates are never shared across different providers, which is consistent with expectations [9], as also highlighted in Section II-C.

Certificate validity periods: The validity of the certificates can be determined from the *notBefore* and *notAfter* fields. For `CERT.DPauth.SIG`, almost 80% of the end hosts employ certificates with a lifetime of 36 months, 20% use certificates valid for 120 months, and only one case was found with a validity of 12 months. Regarding TLS certificates, roughly 95% are valid for 13 months, while the remaining 5% show durations of 3, 12, or 24 months, with a single outlier from G+D Mobile Security exhibiting a validity of 108 months.

V. Related Work

Security analysis of the RSP protocol: Security research on the RSP protocol is currently limited due to its relative novelty. One of the first comprehensive analyses was conducted in 2022 [8]. The authors formally analyze the RSP protocol from the Common Mutual Authentication to the Profile Download and Installation phase. Their work defines 15 security goals, explores two protocol variants, and evaluates 11 scenarios, each assuming that at most one protocol entity is compromised. The work [17] highlights the implicit trust granted to any SM-DP+ server that holds a valid GSMA certificate. The authors argue that compromising such a server could have serious implications and propose the SIM Profile Transparency Protocol (SPTP), aimed at detecting malicious provisioning of SIM profiles. [18] presents a formal verification of the Common Mutual Authentication procedure using Burrows–Abadi–Needham (BAN) logic [19]. The authors identify key vulnerabilities, including the lack of end-to-end encryption, echoing prior concerns raised in [20]. A recent study [21] presents an empirical investigation of how the adoption of eSIM affects user privacy, focusing on routing transparency, reseller access, and profile control.

Broader studies on eSIM technology: Several studies examine the broader implications of the adoption of eSIM. Yuan *et al.* [22] investigate the integration of eSIM in IoT architectures, with a focus on authentication and profile management procedures. A complementary line of research explores how MNOs can support consumer devices enabled by eSIM, focusing on architectural and operational considerations [23]. Another notable work proposes an automated and cost-efficient framework for remote provisioning of IoT devices through eSIM using the IoT-SAFE protocol [24]. The proposed system leverages blockchain for secure profile verification and Software-Defined Networking (SDN) for dynamic security management. Experimental results demonstrate that the approach is scalable, secure, and significantly reduces provisioning time compared to conventional methods.

Empirical analysis of cellular infrastructures: Empirical analysis of mobile ecosystems enables users and researchers to detect and expose security issues in deployed infrastructures. This is especially relevant when misconfigurations affect critical services operated by MNOs. 5GMap [25], for example, is a tool that enables 5G deployment evaluations and has revealed configuration flaws in cryptographic support at the PDCP and NAS layers, including cases where encryption support was inadvertently disabled. Similarly, recent studies [26], [27] investigate the security posture of real-world VoWiFi services,

highlighting weaknesses in the exposure and configuration of ePDG endpoints advertised by several MNOs.

eSIM open-source implementations: During the development of the `eUICC+ LPA Emulator`, the open-source LPA implementation `lpac` [28] became publicly available. Unlike the software modules introduced in this paper, `lpac` does not include the functionalities or capabilities of the eUICC. Consequently, both the `eUICC+LPA Emulator` and the `eUICC+LPA Tester` represent a novel contribution, as no prior implementation combines the features of an eUICC with those of an LPA. Similarly, although there is an open source SM-DP+ server (Osmocom) [14], the `SM-DP+ Tester` will constitute a valuable addition to analyze real-world user agent implementations. Beyond enabling automated test campaigns even for users without knowledge of the internal details of the emulator and tester, the framework also introduces a set of novel test cases that extend the state of the art.

VI. Conclusion

The eSIM ecosystem remains a broad and relatively underexplored domain. In this work, we adopted a practical and systematic approach by developing a suite of software modules, including `SM-DP+ Mapping Tools`, `eUICC+LPA Emulator`, and `eUICC+LPA Tester`, which were used to perform extensive server-side profiling of the ecosystem. Our analysis revealed a many-to-many relationship between the 30 identified SM-DP+ providers and the 286 MNOs, and showed that SM-DP+ servers operated by different providers do not share fingerprints. Nevertheless, all servers exhibited largely consistent behavior, although little differences (and anomalies) emerged in some negative test cases.

Beyond these modules, we are finalizing complementary components — the `SM-DP+ Emulator` and `SM-DP+ Tester` — designed to support in-the-wild analysis of the client side of the eSIM ecosystem. We plan to integrate all of these modules into a unified tool and release the corresponding software. A key innovation of this tool lies in its ability to run automated test campaigns, enabling interactions with real-world SM-DP+ servers and user agents in diverse communication scenarios. To the best of our knowledge, it also represents the first framework capable of performing a global-scale profiling of the majority of existing SM-DP+ providers, SM-DP+ servers, and eSIM-enabled MNOs, thereby creating a worldwide dataset.

Acknowledgments: The first author was supported by *Agenzia per la cybersicurezza nazionale* under the programme for promotion of XL cycle PhD research in cybersecurity — CUP number: E83C24002610001. The views expressed are those of the authors and do not represent the funding institution.

References

[1] TCA, "Introduction to: eSIM", Trusted Connectivity Alliance, Technical report, Nov. 2022.

[2] A. Vesselkov, H. Hammainen, and P. Ikalainen, "Value networks of embedded SIM-based remote subscription management", *2015 Conference of Telecommunication, Media and Internet Techno Economics (CTTE)*, 2015, pp. 1-7.

[3] GSMA, "Remote Provisioning Architecture for embedded UICC v4.3", Global System for Mobile Communications Association, Technical Spec., Jan. 2023.

[4] GSMA, "RSP Technical Spec. version 1.0", Global System for Mobile Communications Association, Technical Spec., Jan. 2016.

[5] Research and Markets, "eSIM market", 2025, [Online]. Available: https://www.researchandmarkets.com/report/e-sim.

[6] GSMA, "RSP Technical Spec. version 3.0", Global System for Mobile Communications Association, Technical Spec., Oct. 2022.

[7] Septs and LaForge, "eUICC and eSIM developer manual", 2024, [Online]. Available: https://euicc-manual.osmocom.org.

[8] A. Ahmed, A. Peltonen, M. Sethi, and T. Aura, "Security analysis of the consumer Remote SIM Provisioning protocol", *ACM Transactions on Privacy and Security*, vol. 27, num. 3, Aug. 2024.

[9] N. Aviram *et al.*, "Drown: Breaking TLS using SSLv2", *Proc. of the 25th USENIX Security Symposium*, USENIX Association, 2016, pp. 689-706.

[10] 3GPP, "Access to the 3GPP Evolved Packet Core (EPC) via non-3GPP access networks", 3rd Generation Partnership Project, Technical Spec., Jan. 2016.

[11] J. Lang, "eSIM profile database", 2024, [Online]. Available: https://osmocom.org/projects/sim-card-related/wiki/.

[12] NekokoLPA, "Nekoko LPA's Profile Stats", 2024, [Online]. Available: https://lpa.nekoko.ee/stats.

[13] CursedHardware, "gsma-rsp-certificates", 2024, [Online]. Available: https://github.com/CursedHardware/gsma-rsp-certificates/.

[14] Osmocom, "pysim", 2025, [Online]. Available: https://github.com/osmocom/pysim.

[15] GSMA, "RSP test certificates definitions version 1.4", Global System for Mobile Communications Association, Technical Spec., July 2020.

[16] 1nce, "Remote SIM Provisioning in IoT: Definition, Technical Aspects and Key Players", 2025, [Online]. Available: https://1nce.com/it-it/euicc-sim-card-for-iot-esim/remote-sim-provisioning-in-iot-definition-tech-aspects-and-key-players.

[17] A. Ahmed, M. Thakur, S. Paavolainen, and T. Aura, "Transparency of eSIM profile for the consumer remote SIM provisiong protocol", *Annals of Telecommunications*, vol. 76, num. 3, Apr. 2021, pp. 187-202.

[18] J. Lastre, Y. Ko, H. Kwon, B. Kim, and I. You, "Formal verification of consumer Remote Sim Provisioning Common Mutual Authentication using BAN logic", *2025 1st Internat. Conference on Consumer Technology*, 2025, pp. 1-4.

[19] P. Syverson, "The use of logic in the analysis of cryptographic protocols", *Proocedings. 1991 IEEE Computer Society Symposium on Research in Security and Privacy*, 1991, pp. 156-170.

[20] K. Ermoshina, F. Musiani, and H. Halpin, "End-to-end encrypted messaging protocols: An overview", *Internet Science*, Springer Internat. Publ., 2016, pp. 244-254.

[21] M. Motallebighomi, J. Veara, E. Bitsikas, and A. Ranganathan, "eSIMplicity or eSIMplification? Privacy and Security Risks in the eSIM Ecosystem", *34th USENIX Security Symposium (USENIX Security 25)*, 2025, pp. 5425-5444.

[22] H. Yuan, A. Baloian, J. Janak, and H. Schulzrinne, "eSIM technology in IoT architecture", 2024, [Online]. Available: https://arxiv.org/abs/2401.04302.

[23] B. Abdou, "Commercializing eSIM for network operators", *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*, 2019, pp. 616-621.

[24] P. Krishnan *et al.* "eSIM and blockchain integrated secure zero-touch provisioning for autonomous cellular-IoTs in 5G networks", *Computer Communications*, vol. 216, 2024, pp. 324-345.

[25] A. Paci, M. Chiacchia, and G. Bianchi, "5GMap: User-driven audit of access security configurations in cellular networks", *Proc. of the 19th Wireless On-demand Network Systems and Services Conference*, Jan. 2024, pp. 97-104.

[26] F. Zampognaro, D. Verde, and G. Bianchi, "VoWiFi security: An exploration of non-3GPP untrusted access via public ePDG URLs", *Italian Conference on Cybersecurity*, 2024.

[27] G.K. Gegenhuber, F. Holzbauer, P.E. Frenzel, E. Weippl, A. Dabrowski, Diffie-Hellman Picture Show: Key Exchange Stories from Commercial VoWiFi Deployments", *Proc. of the 33rd USENIX Security Symposium (USENIX Security 24)*, 2024, pp. 451–468.

[28] eSTK.me Group, "lpac", 2025, [Online]. Available: https://github.com/estkme-group/lpac.