Towards Post-Quantum Secure eSIM Provisioning Protocols

Harrison Banda[1], Hannes Bartz[2], Juliane Krämer[1], Michael Meyer[1], and Vincent Quentin Ulitzsch[3]

[1] University of Regensburg, Regensburg, Germany
{harrison.banda,juliane.kraemer,michael1.meyer}@ur.de
[2] German Aerospace Center (DLR), Oberpfaffenhofen, Germany
hannes.bartz@dlr.de
[3] Massachusetts Institute of Technology, Cambridge, MA, USA
viniul@mit.edu

Abstract. The eSIM specification enables remote SIM provisioning without the need to hand out a physical SIM card. Instead of a physical SIM card, the subscriber downloads a SIM profile, which contains a subscriber's identity and authentication key material, to their embedded UICC, a discrete, embedded chip in the user's phone. This provisioning process is specified in the remote SIM provisioning (RSP) protocol and is secured through contemporary public-key cryptography. However, the potential advent of general-purpose quantum computers threatens the security of RSP.

This paper provides a quantum threat analysis of the RSP protocol, considering both *Harvest Now, Decrypt Later* attacks and active quantum attacks. To alleviate this threat, this paper introduces `PQC-RSP`, a post-quantum secure version of the RSP protocol. The main challenge we address is the introduction of key encapsulation mechanisms (KEMs) into the RSP protocol, both in a PQC-only and hybrid version, and the security implications of this modification. We prove the security of `PQC-RSP` and review the performance overhead introduced through the usage of post-quantum cryptography.

1 Introduction

Subscribers identify themselves towards a mobile network using their international mobile subscriber identity (IMSI) number and a related authentication key K_i [7]. This information is stored in a subscriber's *profile*. Traditionally, this profile has been stored on a smartcard, the subscriber identity module (SIM). The subscriber's profile was hardcoded into the SIM card and handed out to the user. The embedded SIM (eSIM) ecosystem is poised to replace SIM cards. Instead of requiring a physical SIM card, eSIM allows to remotely provision the SIM profile onto an embedded universal integrated circuit card (eUICC), a discrete, embedded secure element in a subscriber's mobile device (e.g., their smartphone) [6,24].

The remote provisioning procedure is specified through the remote SIM provisioning (RSP) protocol [24,25]. In the consumer version of this protocol, the user connects to a provisioning server (through an intermediary software) and downloads the profile from this server to their eUICC. RSP aims at providing confidentiality and integrity against an active man-in-the-middle network adversary. To this end, the provisioning process is secured by contemporary public-key cryptography: first, mutual authentication is ensured through a certificate-based handshake. Second, the eUICC and the provisioning server perform an authenticated Diffie–Hellman key exchange to establish a shared secret, with which they encrypt the subscriber's profile. This ensures that only the eUICC and the provisioning server ever have access to the subscriber profile, and that the subscriber only downloads a valid profile.

However, the potential advent of general-purpose quantum computers threatens the security guarantees of RSP. Indeed, this threat is recognized by the GSM Association (GSMA), which recommends to adapt the RSP towards quantum resistance [26].

Particularly alarming are *Harvest Now, Decrypt Later* attacks: in these attacks, an attacker records encrypted communication now and decrypts it once they have access to a general-purpose quantum computer. This retroactive decryption can pose a threat if by the time of decryption, the acquired information is still valuable to an attacker. In contrast, active quantum attacks are even more devastating, but only apply as soon as such a quantum computer exists.

To alleviate quantum attacks, post-quantum cryptography (PQC), i.e., cryptography that is assumed to resist quantum attacks, must be introduced into the eSIM protocol *now*. The main challenge herein is that there is no efficient direct post-quantum replacement for Diffie–Hellman key exchange as used in RSP. Instead, PQC mainly realizes key agreement through key encapsulation mechanisms (KEMs), whose usage requires modifications to the RSP protocol. Additionally, security protocols are often recommended to support *hybrid* cryptography, in which the protocol combines both contemporary and post-quantum cryptography (see, e.g., [14]). Such a solution offers protection against *Harvest Now, Decrypt Later* attacks due to the post-quantum scheme, and the classical scheme still provides classical security in case practical attacks against the relatively new post-quantum schemes are found.

In order to achieve post-quantum security, the RSP protocol must therefore be significantly modified. Hence, a detailed analysis is required to prove the security of the modified protocol design.

Contributions. Our contributions are summarized as follows:

(i) We provide a quantum threat analysis for the consumer RSP protocol (Section 3), considering *Harvest Now, Decrypt Later* attacks as well as active quantum-enabled attacks. We find that both attack scenarios have devastating consequences, emphasizing the need to transition to a post-quantum variant of RSP as soon as possible.

(ii) We provide `PQC-RSP` (Section 4.1), a post-quantum secure eSIM provisioning protocol, which we present in two variants: 1) the protocol `KEM-PQC-RSP` that

uses a post-quantum secure KEM and 2) the hybrid protocol `hybrid-PQC-RSP`, that uses a classical elliptic-curve Diffie-Hellman (ECDH) key exchange along with a PQC KEM. Both variants can be instantiated with either classical or post-quantum signature schemes to provide authentication.

(iii) While the security of the classical RSP protocol has been analyzed by Ahmed et al. [6], the protocol has to be modified to allow for the usage of PQC. Thus, it is of crucial importance to analyze whether these modifications introduce security flaws. To this end, we prove the security of `PQC-RSP` (Section 4.2), considering active and passive quantum adversaries. We show that `PQC-RSP` achieves mutual authentication, confidentiality, and integrity. In contrast to the analysis in [6] that relies on the symbolic verification tool ProVerif, our work provides a cryptographic approach that ties the protocol security directly to the assumptions of the PQC primitives.

(iv) `PQC-RSP` can be instantiated with any post-quantum KEM and signature scheme. We discuss the usage of several PQC schemes and their impact on the bandwidth requirements and performance of the RSP protocol (Section 4.3).

Related work. Ahmed et al. provide a security analysis of the consumer remote SIM provisioning protocol [6]. Their analysis shows that the RSP protocol is secure against a network adversary, provided that all participants are honest. Their analysis does not consider a quantum-enabled attacker, however.

Multiple works have touched on the topic of post-quantum security of telecommunication networks [19,32,41]. These works provide an overview of the areas in telecommunication protocols that are affected by quantum attacks and mention potential cryptographic primitives for their replacements. Prior work describes the necessary adapations towards post-quantum security for other protocols used in 5G, such as the Subscriber Concealed Identifier (SUCI) [40], or the Secure Channel Protocol '11' (SCP11) used for remote application management in the context of eSIM [10]. However, to the best of our knowledge, prior work has not presented a post-quantum-secure version of the GSMA consumer Remote SIM Provisioning protocol itself. In particular, existing work on post-quantum eSIM protocols addresses related secure-channel mechanisms such as SCP11 [10], but did not address the RSP-specific profile-binding and profile-download flow used to install a telecom subscriber profile.

The GSM association (GSMA) addressed the issue of post-quantum cryptography for telecom use cases in multiple whitepapers [23,26]. GSMA's guidelines for telecom use cases [26] mention that in order to make the RSP protocol secure, *"one could implement ML-KEM to ensure quantum resistance of the key exchange part."* To the best of our knowledge though, this work is the first to propose, in full detail, a post-quantum secure version of the RSP protocol with security proof, including the issue of integrating a hybrid key agreement.

Acknowledgments. This work was funded by the German Federal Ministry of Research, Technology and Space (BMFTR) under the projects 6G-RIC (16KISK033) and QUORYPTAN (16KIS2034).

2 Background

2.1 Cryptography

The security of currently used public-key cryptography (PKC) relies on the hardness of the integer factorization and (elliptic-curve) discrete logarithm problems, which are practically infeasible to solve with classical computers. It has been known for around 30 years, however, that these problems are efficiently solvable on a quantum computer, since Shor's quantum algorithm can solve both problems in polynomial time [37].

In 2016, as a consequence of this threat to PKC (and hence IT security at large), the National Institute of Standards and Technology (NIST) has initiated a standardization process for cryptographic schemes that are assumed to resist attacks with quantum computers [33]. This kind of cryptography is called post-quantum cryptography (PQC). After several rounds of public evaluation, the first three PQC algorithms have now been standardized by NIST: two signature schemes and one key encapsulation mechanism (KEM, see below); another signature scheme is currently being standardized, and another KEM was recently chosen for standardization. In an additional call for standardizing post-quantum signature schemes, 14 further algorithms are currently being studied [33].

According to the German Federal Office for Information Security (BSI), cryptographically relevant quantum computers will exist in 15 years at the latest [15]. However, for many use cases, they already pose a major threat today, cf. Section 3.2. Therefore, post-quantum cryptography has to be integrated in all IT systems as soon as possible. Based on the algorithms standardized by NIST, this is possible also outside of an experimental context.

In theory, one could simply replace current key exchange or digital signature schemes, the main building blocks of public-key cryptography, by PQC alternatives. However, this poses different challenges: first, PQC schemes often involve a slower performance and larger bandwidth requirements than classical ones such as ECDH or ECDSA. Thus, the practicality of the proposed solutions must be carefully evaluated (see Section 4.3), especially when less powerful embedded devices are involved. Second, although CSIDH [18] and SWOOSH [21] provide quantum-resistant non-interactive key exchange schemes, their performance and bandwidth profiles are not practical for many use cases. In particular, CSIDH has been reported to be too slow for usage in TLS even on powerful servers [16], which also extends to more recent optimizations [17], and SWOOSH's key sizes are prohibitively large. Thus, in this work, we will analyze especially the usage of KEMs in our scenario, a variant of key exchange studied in the NIST process for PQC standardization. We introduce the concept in more detail below.

Key Encapsulation Mechanism (KEM). In the ubiquitously deployed key agreement scheme by Diffie and Hellman and its elliptic-curve variant ECDH, two parties agree on a common secret by performing similar computations with their own secrets. In contrast to that, currently discussed post-quantum key agreement solutions generally rely on a key encapsulation mechanism (KEM), where

one party generates a secret key and securely transmits it to the other party. A KEM consists of three algorithms:

- $KEM.KeyGen$ is a probabilistic key-generation algorithm that generates a public-private key pair (k_p, k_s).
- $KEM.Encapsulate$ is a probabilistic encapsulation mechanism that takes the public key k_p as input. It generates a secret k from some initial randomness, encrypts k using public-key cryptography, and then outputs the shared secret k and its corresponding ciphertext c. Note that each invocation of the $KEM.Encapsulate$ function produces a new random shared secret k. NIST requires that KEMs output 256-bit keys; some also allow for variable-length output [36].
- $KEM.Decapsulate$ is the decapsulation algorithm that takes as input a ciphertext c and a secret key k_s and either returns a shared secret k or reports a failure.

NIST requires that encapsulation with the public key and decapsulation with the corresponding private key produce the same shared secret, when the encapsulated ciphertext is given as an input to the decapsulation [33]. KEMs also allow for the possibility of a decryption failure. Then, the key establishment process, has to be repeated. This has to be taken into account when securing a system based on a KEM that has positive decryption failure rate. However, the failure probability is negligible for all NIST finalists and almost all practical purposes.

2.2 Cellular Network and eSIM Provisioning

Traditionally, cellular networks use SIM cards—physical smartcards—as material for user authentication and to establish secure communication. Each subscriber is identified by their international mobile subscriber identity (IMSI), and the SIM card and the the mobile network operator (MNO) have a shared subscriber key K_i. The subscriber can authenticate themselves through the Authentication and Key Agreement (AKA) protocol to the MNO. Subsequent communication between the subscriber and the operator is encrypted using keys derived from K_i. As a result, the confidentiality of the key K_i is crucial, since an attacker with access to K_i can fully decrypt a subscriber's communication with the operator. In the setting of physical SIM cards, only the MNO and the possessor of the SIM card have access to K_i, hence providing this confidentiality.

However, managing physical SIM cards with hardcoded subscriber keys became a burden for multiple stakeholders: device manufacturers need to provide mechanical support mechanisms (e.g., physical switch to eject) to support SIM cards; users need to manage SIM cards for a fleet of cellular-enabled IoT devices; operators rely on expensive logistics to send SIM cards to users.

The eSIM approach addresses these problems by replacing a physical SIM card by a logical *profile* on the eUICC in the user's device. Such a profile comprises of the subscriber's IMSI and subscriber key K_i, therefore maintaining compatibility with the traditional SIM card approach. The technical standard is

Table 1. Notation and Symbols.

Symbol	Description
$mnoID$	MNO identifier
$iccid$	Unique identifier of the SIM profile
$\mathrm{E}(M;k)$	Encryption of message M with symmetric key k
$\mathrm{MAC}(M;k')$	MAC of message M with symmetric key k'
$\mathrm{Sign}(M;SK)$	Signature of message M with private key SK
subject of $Cert$	Subject identifier to which $Cert$ was issued
$check(Cert_a, Cert_b)$	Check that $Cert_a$ is signed by the subject key of $Cert_b$

defined by GSMA. As a part of the eSIM ecosystem, GSMA defines the Remote SIM Provisioning (RSP) architecture [24], which installs an eSIM profile over the internet. The RSP protocol has a machine-to-machine and a consumer version. In this paper, we focus on the consumer RSP protocol. We will briefly outline the setting of this protocol here, and describe it in more detail in Section 3.1.

By default, each eUICC comes without any profile. When an end user wants to subscribe to a cellular network provider using an eSIM, they contact the MNO (out-of-band, for example through a website). The MNO then provides the profile on the subscription manager data preparation (SM-DP+) server with an associated eUICC identifier U. The provider hands out a QR code containing the address of the SM-DP+ server and provisioning token. After scanning the code, the user equipment (UE)'s local profile assistant (LPA) downloads and installs the profile onto the eUICC. Each eSIM profile uses about 10 to 100 KB in size [1,2], and the maximum number of installed profiles depends on the installed profiles and the eUICC's internal storage size.

Because a SIM profile contains critical information, such as the key K_i, the RSP has security requirements to protect private data from various external attackers. The communication between the eUICC and the SM-DP+ over the LPA for requesting and downloading eSIM profiles uses cryptographic primitives such as an ECDH key exchange and digital signatures. Essentially, each party authenticates themselves, and the profile is provided in encrypted form, hence only the eUICC and the MNO obtain the cleartext profile. Thus, even if an attacker gains full control of the UE containing an eUICC, after or during the RSP protocol execution, they cannot gain control of the provisioned eSIM profile.

3 Quantum Analysis of the RSP protocol

In this section, we will first describe the current protocol to provision a profile to the eUICC, followed by a quantum threat analysis of the protocol. Our description and notation will closely follow [6].

3.1 The eSIM Provisioning Protocol

As outlined above, the purpose of the RSP protocol is to provide a profile to a subscriber's phone. The MNO provides the profile on the SM-DP+ server, and

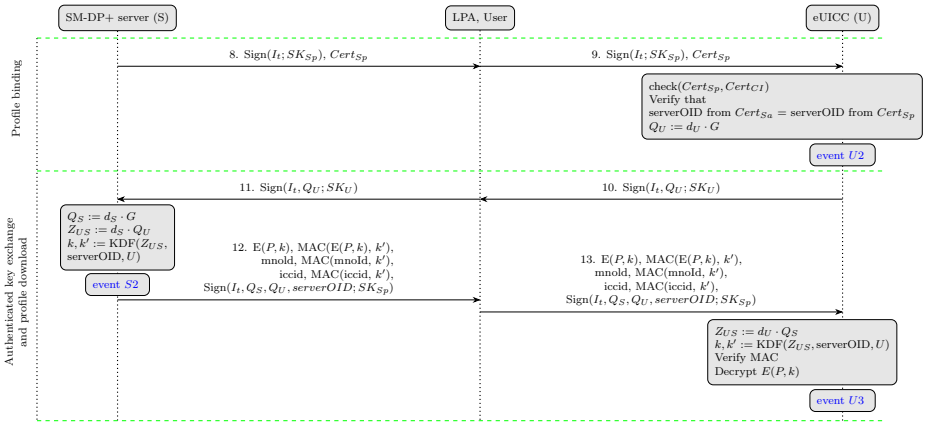

Fig. 1. Profile binding, authenticated key exchange and profile download phase in the RSP protocol. For brevity, signature checks are not depicted in this figure. Adapted from [6].

the eUICC inside the UE communicates with the SM-DP+ server via the LPA software application. The goal is to provide the eUICC with the eSIM profile, which should only be seen in clear by the eUICC and the SM-DP+ server.

We will briefly describe all five phases of the RSP protocol, but only give full details on the phases that are relevant for our analysis with regard to post-quantum security, i.e., the profile binding and authenticated key exchange phases. Figure 1 details these phases following the exposition and notation of [6] for compatability, and Table 1 summarizes additional notation. We refer to [6, Fig. 4] for a complete and detailed description of all phases of the RSP protocol.

Before summarizing the RSP protocol phases, we briefly describe the setup of its certificate-based authentication. The server uses three certificates throughout the protocol: the server's TLS certificate $Cert_{St}$ for its domain name, $Cert_{Sa}$ for authenticating itself to the eUICC and $Cert_{Sp}$ that attests the server's authorization to issue SIM profiles. Both $Cert_{Sa}$ and $Cert_{Sp}$ are for the server's object identifier (OID). The GSMA acts as the certificate issuer (CI), and all certificate chains originate from some GSMA CI root key. The eUICC has two certificates: the manufacturer certificate $Cert_{EUM}$ issued by the GSMA CI, and the eUICC certificate $Cert_U$, issued by the manufacturer for the eUICC identifier U.

Mutual Authentication. The eUICC and the SM-DP+ server perform a certificate-based mutual authentication. During this phase, the eUICC and the server exchange their respective identities U and Sa, together with the certificates for these identities. They then sign session identifiers using the private keys SK_U and SK_{Sp} belonging to the public keys in the certificates $Cert_U$ and $Cert_{Sp}$, respectively. Freshness of the mutual authentication is ensured through nonces and a fresh random session identifier I_t selected by the server.

Profile Binding. In this step, the previously authenticated SM-DP+ proves its authorization to issue SIM profiles through $Cert_{Sp}$. After the eUICC accepts $Cert_{Sp}$, SM-DP+ will prepare the bounded profile package (BPP) which contains

the encrypted metadata with the session key. The BPP itself is not sent to the eUICC in this step; it is done in a later stage.

Authenticated Key Exchange. In this phase, the eUICC and the SM-DP+ server perform an ECDH key exchange. First, both parties generate ephemeral key pairs. The eUICC generates their private key $d_U \in \mathbb{Z}$ and computes the public key $Q_U = d_U \cdot G$. The server computes $Q_S = d_S \cdot G$. They then exchange their public keys and both generate a shared secret Z_{US}. Note that this key exchange is authenticated through digital signatures, proving ownership of the private key belonging to the relevant certificates (messages 11 and 12). Both the eUICC and the SM-DP+ use their shared secret Z_{US} to derive two shared secrets k, k' through a key-derivation function (KDF). The KDF is the SHA-256-based `X9.63 Key Derivation Function` as described in BSI TR-03111 [13], and the RSP Technical Specification [25]. Using this shared secret, the SM-DP+ encrypts-and-MACs the subscriber profile P and sends it to the eUICC.

Profile Download. Upon receiving the encrypted profile, the eUICC verifies its integrity and then decrypts it using k, k'.

Notification. The eUICC notifies the SM-DP+ server of the success.

We note that there are more variants of the RSP protocol. In particular, we focus only on the *default server approach*, in which the server domain name S of the SM-DP+ server is pre-provisioned to the user's eUICC. Other approaches to discovering the SM-DP+ server, such as the *activation code approach*, eventually execute the same steps. All variants rely on an ECDH authenticated key exchange, and thus are equivalent to our exposition in terms of quantum security.

3.2 Quantum Threat Analysis

A quantum threat analysis must take into account several cryptographic building blocks of the RSP protocol. This includes the HTTPS tunnel between the SM-DP+ server and the LPA (see [6]), the certificates and digital signatures for authentication, and the key exchange. For the HTTPS tunnel, we rely on prior work for PQC instantiations (see Section 4). Hence, this section focuses on the key agreement and the authentication of the RSP protocol in Fig. 1. Furthermore, we take into account the subsequent communication using the subscriber's key K_i. We differentiate two cases of quantum attacks.

Harvest Now, Decrypt Later Attacks. The first case—usually termed *Harvest Now, Decrypt Later*—describes the attack scenario in which we currently find ourselves: cryptographically relevant quantum computers do not exist yet. However, attackers can intercept encrypted communication and store it until they have powerful enough quantum computers. Hence, this scenario has to be taken into account whenever data must still be kept secret at times when we assume that cryptographically relevant quantum computers exist. Since this scenario is about confidentiality only, it does not involve signature or authentication schemes, but key agreement and encryption schemes only.

Attacks based on Shor's algorithm. Concretely, imagine an attacker recording the communication between the SM-DP+ and the LPA. Once the attacker has access to a general-purpose quantum computer, they can use Shor's algorithm against the ECDH algorithm to obtain the subscriber's profile. This gives the attacker access to the subscriber's security key K_i. Using this key, the attacker can retroactively decrypt all subsequent communication between that user and the MNO, since it is encrypted using keys derived from the key K_i through a publicly known key derivation function.

The attacker can obtain a subscriber's profile from a recorded RSP communication as follows.

1. The attacker first uses Shor's algorithm to obtain the eUICC's private key d_U from the public key $Q_U = d_U \cdot G$ in message 9. The attacker then computes the shared secret $Z_{US} = d_U \cdot Q_S$, using the SM-DP+'s public key Q_S in message 10.
2. The attacker uses Z_{US} to compute k, k' through the public KDF, similar to the computation in event $S2$.
3. Using the ephemeral key k, the attacker can decrypt $E(P; k)$ and obtain the user's profile P.
4. This profile, in turn, contains the subscriber's key K_i, which can be used to decrypt all subsequent communication.

We highlight that RSP provides forward secrecy, since it uses ephemeral ECDH. As such, each profile that needs to be decrypted requires one invocation of Shor's algorithm.

Attacks based on Grover's algorithm. While Shor's quantum algorithm completely breaks the currently used public-key cryptography, including the ECDH key exchange in the RSP protocol, Grover's algorithm threatens the security of symmetric cryptography by significantly reducing the security of symmetric cryptographic algorithms, such as 3DES or AES [22]. These algorithms are used for the actual encryption of the communication with keys derived from K_i. A *Harvest Now, Decrypt Later* attack could therefore also target the symmetric scheme. Analyses of this threat can be considered independent of the RSP protocol and hence are not subject of this work. To mitigate this, it is generally recommended to double the key sizes for symmetric encryption. For the remainder of this work, we assume that all subsequent communication via symmetric encryption provides sufficient quantum security.

Active Quantum Attacks. The second case describes an attack scenario where cryptographically relevant quantum computers already exist. Then, they also pose a risk to cryptographic solutions which have to ensure certain security properties only for a short time. In the case of RSP, an active attacker could forge signatures to create seemingly valid certificates. This would allow the attacker to act as a party involved in the RSP protocol. As a result, the attacker could act as the authorized eUICC to the server, or act as an authorized SM-DP+ server

to the eUICC. In the former case, the attacker could retrieve another eUICC's SIM profile—even if the key material that is used to encrypt the victim's profile is encrypted using post-quantum secure cryptographic primitives.

In summary, active attacks only apply if large-scale quantum computers already exist during an execution of the RSP protocol. Although mitigating such attacks seems less urgent than *Harvest Now, Decrypt Later* attacks, a timely transition to fully quantum-resistant protocols is generally recommended. Thus, we will also consider post-quantum signatures (and certificates) to provide security against active quantum attacks in the remainder of this work.

4 A Post-Quantum Secure eSIM Provisioning Protocol

In this section, we describe the required adaptations to the RSP protocol that allow for the integration of post-quantum KEMs as well as a hybrid key agreement. We will prove the security of `PQC-RSP`, and briefly discuss the practicality of potential instantiations.

During the authenticated key exchange and profile download phase the key exchange protocol needs to be modified to support KEMs instead of ECDH and mutliple key exchanges to support a hybrid setting. In contrast, to make the signatures used in the mutual authentication and the certificates quantum resistant, the structure of the RSP protocol does not have to be modified. The corresponding signature schemes and certificates can simply be replaced by corresponding PQC variants. However, at the beginning of the protocol, a HTTPS tunnel between the SM-DP+ and LPA is established using TLS 1.3. In order to prevent quantum attacks, we assume the usage of a PQC variant of TLS 1.3, such as, e.g., [39].

4.1 Required Adaptations of the Protocol

As pointed out in Section 3.2, the original RSP is vulnerable against quantum attacks that target the authenticated key exchange performed with ECDH in messages 10 - 13.

Integration of a PQC KEM. In order to replace the ECDH key exchange with a PQC KEM, we modify the RSP protocol as follows. In Figure 2, we provide an overview of the adapted messages of the obtained `KEM-PQC-RSP` protocol. The respective message numbers are the same for the original RSP protocol in Figure 1 and the proposed PQC and hybrid variants.

1. First, in event $U2$ the eUICC generates a public-private key pair $(k_{p,U}, k_{s,U}) = KEM.KeyGen()$.
2. The public key $k_{p,U}$ is signed and sent to the LPA and the SM-DP+ server via messages 10 and 11 of the form $\text{Sign}(I_t, k_{p,U}; SK_U)$.
3. The SM-DP+ server uses the public key $k_{p,U}$ to compute the shared secret Z_{US} and the ciphertext c as $(Z_{US}, c) = KEM.Encapsulate(k_{p,U})$ (event $S2$). The keys k, k' are then generated using the KDF as in the original RSP protocol, i.e., $(k, k') = KDF(Z_{US}, serverOID, U)$.

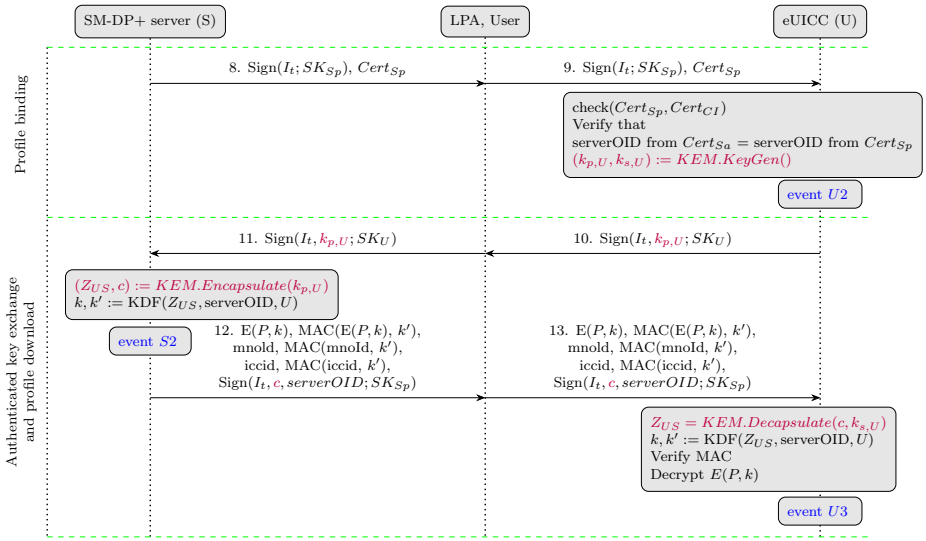

Fig. 2. The proposed `KEM-PQC-RSP` variant of the RSP protocol. Adaptations w.r.t the original RSP protocol are highlighted.

4. The signed ciphertext c is then sent to the LPA and the eUICC along with messages 12 and 13 as $\text{Sign}(I_t, c, serverOID; SK_{Sp})$.
5. In event $U3$, the eUICC decapsulates the ciphertext c using the secret key $k_{s,U}$ to obtain the shared secret $Z_{US} = KEM.Decapsulate(c, k_{s,U})$. The keys k, k' are again generated using the KDF as in the original RSP protocol, i.e. $(k, k') = KDF(Z_{US}, serverOID, U)$.

Hybrid key exchange. During the migration phase towards PQC, it is highly recommended by governmental institutions such as the German BSI that a hybrid key exchange, which consists of a combination of a classical key exchange and a post-quantum key exchange, is performed (see, e.g., [14]). In order to obtain a hybrid variant of the RSP protocol, we modify the protocol in the spirit of hybrid key exchange in TLS 1.3 [39] to allow for additional key exchanges. In Figure 3, we provide an overview of the messages of the proposed `hybrid-PQC-RSP` protocol.

1. In event $U2$, the eUICC generates a public-private key pair $(k_{p,U}, k_{s,U}) = KEM.KeyGen()$ in addition to the ECDH public key Q_U.
2. Messages 10 and 11 are then modified to carry both public keys Q_U and $k_{p,U}$, i.e., we send $\text{Sign}(I_t, Q_U, k_{p,U}; SK_U)$.
3. In event $S2$ the SM-DP+ server computes the ECDH shared secret $Z_{US,1} = d_S \cdot Q_U$ along with the KEM shared secret $Z_{US,2}$ and ciphertext c as $(Z_{US,2}, c) = KEM.Encapsulate(k_{p,U})$. Similarly as in the hybrid key exchange for TLS 1.3 [39] we use a simple concatenation approach to obtain the shared secret $(Z_{US,1} \,||\, Z_{US,2})$ and obtain the keys $(k, k') = KDF(Z_{US,1} \,||\, Z_{US,2}, serverOID, U)$.

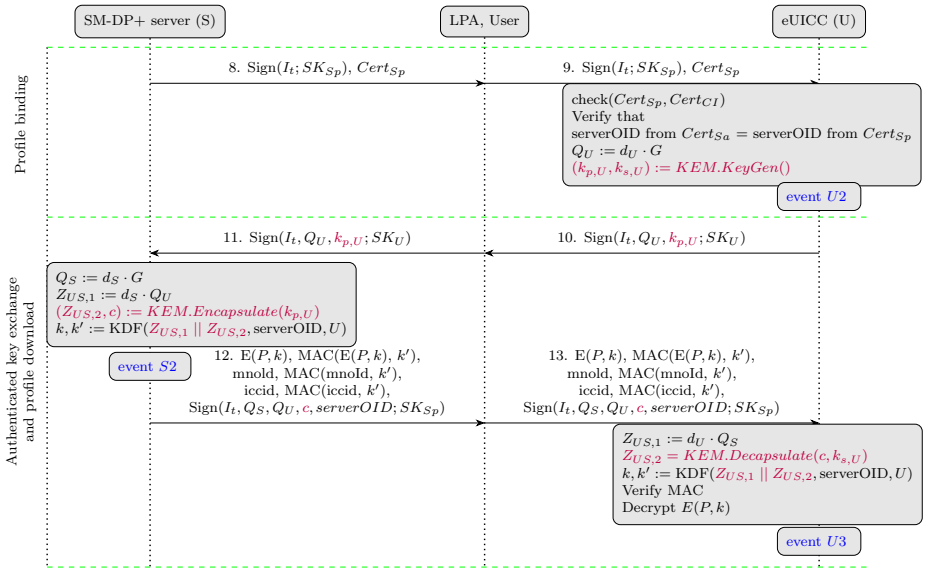

Fig. 3. The proposed `hybrid-PQC-RSP` variant of the RSP protocol. Adaptations w.r.t the original RSP protocol are highlighted.

4. Messages 12 and 13 are modified so that in addition to Q_S and Q_U they also contain the ciphertext c. In particular, we send $\text{Sign}(I_t, Q_S, Q_U, c; SK_U)$ instead of $\text{Sign}(I_t, Q_S, Q_U; SK_U)$ as in Figure 1.
5. In event $U3$ the eUICC computes the ECDH shared secret $Z_{US,1} = d_U \cdot Q_S$ and the KEM shared secret $Z_{US,2} = KEM.Decapsulate(c, k_{s,U})$ to obtain the keys $(k, k') = KDF(Z_{US,1} \,||\, Z_{US,2}, serverOID, U)$.

The role of the eUICC and the SM-DP+ server in the PQC-RSP protocol. In both variants of the `PQC-RSP` protocol, the key generation as well as the decapsulation is performed at the eUICC, while the encapsulation is performed at the SM-DP+ server. This approach reduces the computational load on the SM-DP+ server, as the server has to handle provisioning requests constantly, whereas the eUICC has to perform profile provisioning only rarely. It is also possible to switch the roles of the eUICC and SM-DP+ server, in order to reduce the computational load for the less powerful eUICC. However, we note that this requires some changes to the flow of the protocol in the authenticated key exchange phase of the RSP protocol.

4.2 Security Proof of `PQC-RSP`

The design of security protocols is known to be error-prone, with flaws often discovered years after deployment, which necessitates careful security analysis with rigorous proofs [31].

Two main methods have been developed to analyze protocol security [12,31]: the cryptographic approach and formal methods.

The current RSP protocol security has been analyzed using formal methods, specifically with ProVerif [6]. While this provides useful assurance through automated verification in the symbolic model, we seek to strengthen the security guarantees for our `PQC-RSP` protocol through computational proofs that provide concrete security reductions to underlying primitives.

For `PQC-RSP`, we adopt the *game-based approach* within the cryptographic model [34,38] to establish provable security. For brevity, we will only present a proof for `KEM-PQC-RSP`. The confidentiality and integrity of `hybrid-PQC-RSP` follow from the same reduction structure, and the proof can be derived in an analogous manner.

Adversarial Model. This work adheres to the standard security notions established in modern cryptography. We rely on the conventional definitions of existential unforgeability under chosen-message attacks (EUF-CMA) for digital signatures and MACs, indistinguishability under chosen-ciphertext attacks (IND-CCA) for key encapsulation mechanisms, indistinguishability under chosen-plaintext attacks (IND-CPA) for symmetric encryption, and pseudorandom function (PRF) security for key-derivation functions. The formal definitions of these security experiments and their corresponding advantage measures are well-established and can be found in references [8,28,38] and related work therein.

Our goal is to formally establish that any adversary capable of violating the security guarantees of our protocol must also break the security of at least one of the underlying cryptographic primitives. We consider an adversary $\mathcal{A}$ operating within a standard cryptographic framework, and we analyze the protocol with respect to mutual authentication, confidentiality and integrity.

We consider a quantum polynomial time (QPT) adversary $\mathcal{A}$ attempting to compromise the security of the protocol by interacting with honest parties. The protocol is analyzed in the classical random oracle model. Although $\mathcal{A}$ is quantum-capable, it is restricted to classical queries to these oracles.

The adversary $\mathcal{A}$ is permitted to operate in both passive and active capacities. In passive mode, $\mathcal{A}$ may eavesdrop on protocol executions between honest eUICC and SM-DP+ servers, observing all message exchanges. In active mode, $\mathcal{A}$ may initiate and control protocol sessions while impersonating either communication party (server or eUICC) and may conduct man-in-the-middle attacks.

We now proceed to formally state and prove each of the security properties under the adversarial model described above.

Mutual Authentication. The goal of mutual authentication is to ensure that both the eUICC and SM-DP+ server accept a session only if they are communicating with their legitimate counterpart. In particular, we aim to prevent any adversary from successfully impersonating either party to the other.

This authentication is achieved via a combination of certificate verification and digital signatures. A successful impersonation must either forge valid certificates or signatures to authenticate as an unauthorized party. Mutual authentication in the `PQC-RSP` protocol is achieved during the initial authentication

phase (messages 1–7). In this phase, the server authenticates itself to the user using the certificate Cert_{Sa} and the signature contained in message M_4, while the client authenticates itself to the server using the certificates Cert_U, Cert_{EUM}, and the signature in message M_6.

This is analogous to the original RSP protocol; however, for the sake of completeness we state the following theorem and provide the proof in Appendix B.

Theorem 1. *Let $\mathcal{A}$ be any QPT adversary attacking the mutual authentication between eUICC and SM-DP+ server in the PQC-RSP protocol. If the signature schemes for the protocol and the certificate authority signatures are EUF-CMA secure, then $\mathcal{A}$'s advantage is negligible. Specifically, there exist adversaries $\mathcal{B}_1$, $\mathcal{B}_2$, $\mathcal{B}_3$, and $\mathcal{B}_4$ against EUF-CMA such that*

$$\mathsf{Adv}^{\mathsf{auth}}_{\mathcal{A}}(\lambda) = \sum_{i=1}^{4} \mathsf{Adv}^{\mathsf{EUF\text{-}CMA}}_{\mathcal{B}_i}(\lambda).$$

Confidentiality of the Profile. Having established authentication guarantees, we now turn to confidentiality, the protection of the SIM profile P that contains the subscriber key K_i.

Theorem 2 (Confidentiality of PQC-RSP Protocol). *Let $\mathcal{A}$ be a QPT adversary against confidentiality. Assume the key-exchange signature scheme (messages 10-13) is EUF-CMA secure, $\mathcal{KEM}$ is IND-CCA2 secure, and* Enc *is IND-CPA secure. Let* KDF *be modeled as a random oracle. Then there exist adversaries $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_4$ such that*

$$\mathsf{Adv}^{\mathsf{conf}}_{\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathsf{EUF-CMA}}_{\mathcal{B}_1}(\lambda) + \mathsf{Adv}^{\mathsf{IND-CCA2}}_{\mathcal{B}_2}(\lambda) + \frac{q_H}{|\mathcal{K}|^2} + \mathsf{Adv}^{\mathsf{IND-CPA}}_{\mathcal{B}_4}(\lambda),$$

where q_H is the number of random oracle queries made by $\mathcal{A}$ and $|\mathcal{K}|$ is the size of the key space.

Proof. We prove confidentiality via a sequence of games, defining events S_i where S_i denotes that $\mathcal{A}$ wins in Game i.

Game 0: This is the confidentiality experiment for `PQC-RSP`. The challenger generates all long-term keys and certificates honestly. $\mathcal{A}$ selects two distinct profiles P_0, P_1. The challenger samples $b \leftarrow \{0,1\}$ and executes: $\mathsf{Sign}(I_t, k_{p,U}; SK_U)$, $\mathsf{Sign}(I_t, c, \mathsf{serverOID}; SK_{Sp})$, $(Z_{US}, c) \leftarrow \mathsf{KEM.Encapsulate}(k_{p,U})$,
$(k, k') \leftarrow \mathsf{KDF}(Z_{US}, \mathsf{serverOID}, U)$, $\psi = (E(P_b, k), \mathsf{MAC}(E(P_b, k), k'))$

$\mathcal{A}$ receives the challenge transcript and outputs guess b'. $\mathcal{A}$ wins (event S_0) if $b' = b$. By definition, $\mathsf{Adv}^{\mathsf{conf}}_{\mathcal{A}}(\lambda) = |\Pr[S_0] - \frac{1}{2}|$.

Game 1: We change Game 0 to abort if any key-exchange signature (message 10) is a fresh forgery. Let F_1 be that event.

Claim. Games 0 and 1 are identical unless F_1 occurs. Moreover,
$|\Pr[S_0] - \Pr[S_1]| \leq \Pr[F_1] = \mathsf{Adv}^{\mathsf{EUF-CMA}}_{\mathcal{B}_1}(\lambda)$.

Proof. We construct adversary $\mathcal{B}_1$ against the key-exchange signature scheme. $\mathcal{B}_1$ receives a public key and a signing oracle. It simulates Game 1 for $\mathcal{A}$: it generates all keys honestly and uses its oracle to produce all key-exchange signatures. It records all signatures generated by the oracle. If $\mathcal{A}$ ever causes an abort due to a fresh valid signature, $\mathcal{B}_1$ outputs that signature as its forgery. The simulation is perfect, so $\Pr[F_1] = \mathsf{Adv}^{\mathsf{EUF-CMA}}_{\mathcal{B}_1}(\lambda)$.

Game 2: We replace the real KEM shared secret with a uniformly random value, i.e. instead of $(Z_{US}, c) \leftarrow\!\$\ \mathsf{KEM.Encapsulate}(k_{p,U})$, we set $Z_{US} \leftarrow\!\$\ \mathcal{K}$ and $c \leftarrow \mathsf{KEM.Encapsulate}(k_{p,U})$.

Claim. Any adversary distinguishing Games 1 and 2 can be turned into an IND-CCA2 adversary against $\mathcal{KEM}$. Hence $|\Pr[S_1] - \Pr[S_2]| \leq \mathsf{Adv}^{\mathsf{IND-CCA2}}_{\mathcal{B}_2}(\lambda)$.

Proof. We construct adversary $\mathcal{B}_2$ against the KEM. $\mathcal{B}_2$ receives a public key k_p, U^* and a challenge (c^*, k_b) from its IND-CCA2 challenger. It sets $k_p, U = k_p, U^*$ and uses c^* as the challenge ciphertext. It simulates Game 1 for $\mathcal{A}$, using k_b as Z_{US}. If $b = 0$ the simulation matches Game 1; if $b = 1$ it matches Game 2. Hence $\mathcal{B}_2$'s advantage is exactly $|\Pr[S_1] - \Pr[S_2]|$.

Game 3: We replace the KDF output with truly random keys. In Game 2, $(k, k') = \mathsf{KDF}(Z_{US}, \mathsf{serverOID}, \mathsf{U})$. In Game 3, we sample $(k, k') \leftarrow\!\$\ \mathcal{K} \times \mathcal{K}$ independently. Let F_3 be the event that $\mathcal{A}$ queries the random oracle on the input $(Z_{US}, \mathsf{serverOID}, \mathsf{U})$.

Claim. Games 2 and 3 are indistinguishable unless F_3 occurs. Consequently, $|\Pr[S_2] - \Pr[S_3]| \leq \Pr[F_3] = \frac{q_H}{|\mathcal{K}|^2}$.

Proof. In Game 2, Z_{US} is uniformly random over $\mathcal{K}$ and independent of $\mathcal{A}$'s view except through the random oracle. The probability that any of the q_H queries hits the specific input $(Z_{US}, \mathsf{serverOID}, \mathsf{U})$ is at most $\frac{q_H}{|\mathcal{K}|^2}$. Until that query, the games are identical, so the difference in success probability is bounded by this probability.

Game 4: We replace the encryption of the profile with an encryption of a zero string of the same length. Since k is now independent of the challenge bit, any distinguishing advantage would break IND-CPA.

Claim. $|\Pr[S_3] - \Pr[S_4]| \leq \mathsf{Adv}^{\mathsf{IND-CPA}}_{\mathcal{B}_4}(\lambda)$.

Proof. We construct adversary $\mathcal{B}_4$ against the IND-CPA security of encryption, E. $\mathcal{B}_4$ simulates Game 3 for $\mathcal{A}$. When it needs to produce the challenge ciphertext, it submits (P_0, P_1) to its IND-CPA challenger and receives a ciphertext c^* that encrypts either P_0 or P_1. $\mathcal{B}_4$ uses c^* as ψ. The IND-CPA challenger uses a random key independent of the simulation; therefore $\mathcal{B}_4$'s advantage equals $|\Pr[S_3] - \Pr[S_4]|$.

Final analysis: In Game 4, ψ contains no information about b, so $\Pr[S_4] = \frac{1}{2}$. Combining bounds yields the theorem. □

Integrity of Profile and Messages. While confidentiality protects the secrecy of the profile, integrity ensures it cannot be tampered with during transmission. This property prevents adversaries from modifying protocol messages or injecting malicious profiles.

Theorem 3 (Integrity of the PQC-RSP Protocol). *Let $\mathcal{A}$ be any QPT adversary against the integrity of the PQC-RSP protocol. If the digital signature scheme is EUF-CMA secure, the KEM is IND-CCA2 secure, the* KDF *is modeled as a random oracle, and the MAC scheme is SUF-CMA secure, then no such adversary can break the integrity of $\mathcal{P}$ with non-negligible advantage. Specifically, there exist efficient adversaries $\mathcal{B}_1, \mathcal{B}_2, \mathcal{B}_3$, and $\mathcal{B}_4$ such that*

$$\mathsf{Adv}^{\mathsf{int}}_{\mathcal{A}}(\lambda) \leq \mathsf{Adv}^{\mathsf{EUF-CMA}}_{\mathcal{B}_1}(\lambda) + \mathsf{Adv}^{\mathsf{EUF-CMA}}_{\mathcal{B}_2}(\lambda) + \mathsf{Adv}^{\mathsf{IND-CCA2}}_{\mathcal{B}_3}(\lambda) + \frac{q_H}{|\mathcal{K}|^2} + \mathsf{Adv}^{\mathsf{SUF-CMA}}_{\mathcal{B}_4}(\lambda),$$

where q_H is the number of random oracle queries made by $\mathcal{A}$ and $|\mathcal{K}|$ is the size of the key space.

Proof. We prove integrity via a sequence of games, defining events S_i where S_i denotes that $\mathcal{A}$ wins in Game i.

Game 0: This is the real integrity experiment. The challenger generates all long-term keys and certificates as in the authentication proof. It then runs polynomially many honest sessions of the protocol. During each session: the client sends $\mathsf{Sign}((I_t, k_p, U; SK_U))$, the server sends $\mathsf{Sign}(I_t, c, serverOID; SK_{Sp})$. After key exchange, the server computes $(k, k') = \mathsf{KDF}(Z_{US}, \mathsf{serverOID}, \mathsf{U})$ and sends $E(P, k)$ with $\mathsf{MAC}(E(P, k), k')$, MNOID with $\mathsf{MAC}(\mathsf{MNOID}, k')$, ICCID with $\mathsf{MAC}(\mathsf{ICCID}, k')$. $\mathcal{A}$ outputs (m^*, τ^*). Return 1 if (m^*, τ^*) is a valid signature or MAC not produced by honest sender; else 0. Hence $\mathsf{Adv}^{\mathsf{int}}_{\mathcal{A}}(\lambda) = \Pr[S_0]$.

Game 1: We change Game 0 to abort if the adversary causes acceptance of a client signature with $\mathsf{Verify}(\sigma^*, \mathsf{pk_U}) = 1$ and $\sigma^* \notin \mathcal{Q}_{\text{client}}$, where $\mathcal{Q}_{\text{client}}$ is the set of client signatures produced. Let F_1 be that event.

Claim. Games 0 and 1 are identical unless F_1 occurs. Moreover,
$|\Pr[S_0] - \Pr[S_1]| \leq \Pr[F_1] = \mathsf{Adv}^{\mathsf{EUF-CMA}}_{\mathcal{B}_1}(\lambda)$.

Proof. We construct adversary $\mathcal{B}_1$ against the client's signature scheme. $\mathcal{B}_1$ receives $\mathsf{pk_U}$ and a signing oracle. It simulates Game 1 for $\mathcal{A}$: it generates all other keys honestly and uses its oracle to produce client signatures when needed. It records all signatures generated by the oracle in $\mathcal{Q}_{\text{client}}$. If $\mathcal{A}$ ever causes an abort due to a fresh valid client signature, $\mathcal{B}_1$ outputs that signature as its forgery. The simulation is perfect, so $\Pr[F_1] = \mathsf{Adv}^{\mathsf{EUF-CMA}}_{\mathcal{B}_1}(\lambda)$.

Game 2: We change Game 1 to abort on fresh valid server signature, i.e., σ^* with $\mathsf{Verify}(\sigma^*, \mathsf{pk}_{\mathsf{Sp}}) = 1$ and $\sigma^* \notin \mathcal{Q}_{\text{server}}$, where $\mathcal{Q}_{\text{server}}$ be the set of server signatures produced. Let F_2 be that event.

Claim. Games 1 and 2 are identical unless F_2 occurs. Moreover, $|\Pr[S_1] - \Pr[S_2]| \leq \Pr[F_2] = \mathsf{Adv}^{\mathsf{EUF-CMA}}_{\mathcal{B}_2}(\lambda)$.

Proof. We construct adversary $\mathcal{B}_2$ against the server's signature scheme. $\mathcal{B}_2$ receives $\mathsf{pk}_{\mathsf{Sp}}$ and a signing oracle. It simulates Game 2 for $\mathcal{A}$: it generates all other keys honestly and uses its oracle to produce server signatures when needed. It records all signatures generated by the oracle in $\mathcal{Q}_{\text{server}}$. If $\mathcal{A}$ ever causes an abort due to a fresh valid server signature, $\mathcal{B}_2$ outputs that signature as its forgery. The simulation is perfect, so $\Pr[F_2] = \mathsf{Adv}^{\mathsf{EUF-CMA}}_{\mathcal{B}_2}(\lambda)$.

Game 3: We now make the KEM shared secret uniformly random, independent of the ciphertext. In Game 2, Z_{US} is obtained from $\mathsf{KEM.Encapsulate}(k_{p,U})$. In Game 3, we set $Z_{US} \leftarrow\!\$\, \mathcal{K}$.

Claim. Any adversary distinguishing Games 2 and 3 yields an IND-CCA2 adversary against $\mathcal{KEM}$. Hence $|\Pr[S_2] - \Pr[S_3]| \leq \mathsf{Adv}^{\mathsf{IND-CCA2}}_{\mathcal{B}_3}(\lambda)$.

Proof. We construct adversary $\mathcal{B}_3$ against the KEM. $\mathcal{B}_3$ receives a public key pk^* and a challenge (c^*, k_b) from its IND-CCA2 challenger. It sets $\mathsf{pk}_{\mathsf{U}} = \mathsf{pk}^*$ and uses c^* as the challenge ciphertext. It simulates Game 2 for $\mathcal{A}$, using k_b as Z_{US}. If $b = 0$ (real key) the simulation matches Game 2; if $b = 1$ (random) it matches Game 3. Hence $\mathcal{B}_3$'s advantage equals $|\Pr[S_2] - \Pr[S_3]|$.

Game 4: We replace the KDF output with truly random keys. In Game 3, $(k, k') = \mathsf{KDF}(Z_{US}, \mathsf{serverOID}, \mathsf{U})$. In Game 4, we sample $(k, k') \leftarrow\!\$\, \mathcal{K} \times \mathcal{K}$ independently. Let F_4 be the event that $\mathcal{A}$ queries the random oracle on the input $(Z_{US}, \mathsf{serverOID}, \mathsf{U})$.

Claim. Games 3 and 4 are indistinguishable unless F_4 occurs. Hence $|\Pr[S_3] - \Pr[S_4]| \leq \Pr[F_4] = \frac{q_H}{|\mathcal{K}|^2}$.

Proof. In Game 3, Z_{US} is uniformly random over $\mathcal{K}$ and independent of $\mathcal{A}$'s view except through the random oracle. The probability that any of the q_H queries hits the specific input $(Z_{US}, \mathsf{serverOID}, \mathsf{U})$ is at most $\frac{q_H}{|\mathcal{K}|^2}$. Until that query, the games are identical, so the difference in success probability is bounded by this probability.

Game 5: We change Game 4 to abort if the adversary causes acceptance of any payload item (encrypted profile, MNOID, ICCID) with a fresh valid MAC tag under k', i.e., τ^* with $\mathsf{MAC}(\cdot, k')$ valid and $\tau^* \notin \mathcal{Q}_{\text{MAC}}$, where $\mathcal{Q}_{\text{MAC}}$ is the set of MAC tags produced. Let F_5 be that event.

Claim. Games 4 and 5 are identical unless F_5 occurs.
Moreover, $|\Pr[S_4] - \Pr[S_5]| \leq \Pr[F_5] = \mathsf{Adv}^{\mathsf{SUF-CMA}}_{\mathcal{B}_4}(\lambda)$.

Proof. We construct adversary $\mathcal{B}_4$ against the MAC scheme. $\mathcal{B}_4$ receives a MAC oracle. It simulates Game 5 for $\mathcal{A}$: it generates all other keys honestly and uses its oracle to produce MAC tags when needed. It records all tags generated by the oracle in $\mathcal{Q}_{\text{MAC}}$. If $\mathcal{A}$ ever causes an abort due to a fresh valid MAC tag, $\mathcal{B}_4$ outputs that tag as its forgery. The simulation is perfect, so $\Pr[F_5] = \mathsf{Adv}^{\mathsf{SUF-CMA}}_{\mathcal{B}_4}(\lambda)$.

Final analysis: In Game 5, all keys are independent and uniformly random, and the abort conditions prevent any acceptance of forged signatures or MACs. Therefore $\Pr[S_5] = 0$.
Collecting all the bounds yields the desired inequality, which completes the proof. □

4.3 Instantiations

We briefly estimate the expected performance and bandwidth implications when instantiating our `PQC-RSP` variants with PQC schemes. More details are presented in Appendix A. An evaluation of concrete implementations and their efficiency analyses are left for future work.

To estimate the performance of the considered schemes on the eUICC, we use benchmark results for Cortex M4 microcontrollers. For the SM-DP+ server, we use benchmark results for modern Intel processors. Table 2 provides the relevant data for KEMs and Table 3 for signature schemes. Other metrics, such as memory usage and power consumption on the eUICC or resistance against physical attacks, are beyond the scope of our rough estimates.

Integration of PQC KEMs. Integrating a PQC KEM would replace the currently used ECDH in `PQC-RSP`, or use both schemes in a hybrid way, respectively. The best PQC candidate for this is ML-KEM (NIST standardized). It improves upon ECDH in terms of performance, but features more than 20× increased public key and cipertext sizes. In spite of the performance improvements, the increased latency of the protocol due to more data packages being sent might impact the overall performance of the protocol. We refer to Appendix A for more details.

Integration of PQC Signatures. Integrating PQC signatures would replace the currently used ECDSA in `PQC-RSP`. The most promising PQC candidates for this are ML-DSA (standardized; based on Dilithium [30]), FALCON [35] (to be standardized as FN-DSA), and MAYO [11]. While all three schemes have a relatively good overall performance (although in most cases slower than ECDSA), their signature and public key sizes are much larger than in ECDSA. Given that multiple rounds of signing and verification are executed during the protocol, this has a bigger impact on the overall performance than when integrating KEMs. We describe some more details in Appendix A, but leave a detailed practical analysis for future work to decide which scheme is best suited for this application.

We note that the work of Bettale et al. [10] includes performance results for a different eSIM use case using PQC signatures. However, in their SCP11 protocol, the embedded chip only verifies signatures, while RSP also includes signing, leading to different considerations.

5 Conclusion & Outlook

This paper presents a quantum threat analysis of the RSP protocol for provisioning an eSIM profile to an eUICC. To alleviate the issue of *Harvest Now, Decrypt Later* and active quantum attacks, we present `KEM-PQC-RSP` and `hybrid-PQC-RSP`, which enable the use of post-quantum secure KEMs in the RSP protocol, and prove their security. While our work lays the foundation in terms of enabling quantum-resistant eSIM provisioning from the security point of view, more research on practical instantiations is required. This includes a fine-grained analysis of suitable PQC schemes for RSP and practical experiments to evaluate the impact on performance, bandwidth requirements, and power consumption of these instantiations.

References

1. eSIM Profile Database. https://osmocom.org/projects/sim-card-related/wiki/ESIM_profile_database.
2. eSIM Size Statistics. https://sim.obdo.cc/esim-size.
3. Marius A. Aardal, Gora Adj, Diego F. Aranha, et al. SQIsign. Technical report, National Institute of Standards and Technology, 2025.
4. Olivier Adjonyo, Arwa Alblooshi, Isaac Canales-Martinez, Jorge Chavez-Saab, Gilbert Dione, and Francisco Rodriguez-Henriquez. PQ-SORT: Post-quantum signatures on-ramp tests, 2025. https://pqsort.tii.ae/.
5. Carlos Aguilar-Melchor, Nicolas Aragon, Slim Bettaieb, et al. HQC. Technical report, National Institute of Standards and Technology, 2022. https://pqc-hqc.org/.
6. Abu Shohel Ahmed, Aleksi Peltonen, Mohit Sethi, and Tuomas Aura. Security analysis of the consumer remote sim provisioning protocol. *ACM Transactions on Privacy and Security*, 27(3):1–36, 2024.
7. Stephanie Alt, Pierre-Alain Fouque, Gilles Macario-Rat, Cristina Onete, and Benjamin Richard. A cryptographic analysis of umts/lte aka. In *ACNS 2016*, pages 18–35. Springer, 2016.
8. Mihir Bellare, Anand Desai, David Pointcheval, and Phillip Rogaway. Relations among notions of security for public-key encryption schemes. In *Annual International Cryptology Conference*, pages 26–45. Springer, 1998.
9. Daniel J. Bernstein and Tanja Lange. ebacs: Ecrypt benchmarking of cryptographic systems, 2025. https://bench.cr.yp.to/supercop.html.
10. Luk Bettale, Emmanuelle Dottax, and Laurent Grémy. Post-quantum secure channel protocols for esims. *Cryptology ePrint Archive*, 2024.
11. Ward Beullens, Fabio Campos, Sofía Celi, Basil Hess, and Matthias J. Kannwischer. MAYO. Technical report, National Institute of Standards and Technology, 2025. https://pqmayo.org/.
12. Bruno Blanchet. Mechanizing game-based proofs of security protocols. In *Software Safety and Security*, pages 1–25. IOS Press, 2012.
13. BSI. Technical Guideline BSI TR-03111: Elliptic Curve Cryptography. Technical Report TR-03111, Bundesamt für Sicherheit in der Informationstechnik (BSI), 2018. https://www.bsi.bund.de/dok/TR-03111.

14. BSI. Migration to post quantum cryptography, 2020. https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Crypto/Migration_to_Post_Quantum_Cryptography.pdf.
15. BSI. The status of quantum computer development, 2025. https://www.bsi.bund.de/dok/study_status_quantum_computer.
16. Fabio Campos, Jorge Chávez-Saab, Jesús-Javier Chi-Domínguez, et al. Optimizations and practicality of high-security CSIDH. *IACR Commun. Cryptol.*, 1(1):5, 2024.
17. Fabio Campos, Andreas Hellenbrand, Michael Meyer, and Krijn Reijnders. dc-tidh: Fast & deterministic CTIDH. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2025(3):516–541, 2025.
18. Wouter Castryck, Tanja Lange, Chloe Martindale, Lorenz Panny, and Joost Renes. CSIDH: an efficient post-quantum commutative group action. In *Advances in Cryptology - ASIACRYPT 2018*, pages 395–427. Springer, 2018.
19. T Charles Clancy, Robert W McGwier, and Lidong Chen. Tutorial: Post-quantum cryptography and 5g security. In *ACM WiSec'19*, 2019.
20. Hayato Fujii and Diego F Aranha. Curve25519 for the cortex-m4 and beyond. In *LATINCRYPT 2017*, pages 109–127. Springer, 2017.
21. Phillip Gajland, Bor de Kock, Miguel Quaresma, Giulio Malavolta, and Peter Schwabe. SWOOSH: efficient lattice-based non-interactive key exchange. In *USENIX Security 2024*. USENIX Association, 2024.
22. Lov K Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 212–219, 1996.
23. GSMA. Post Quantum Telco Network Impact Assessment Whitepaper. Technical Report PQ.01, GSM Association (GSMA), 2023. Version 1.0.
24. GSMA. RSP Architecture. Technical Report SGP.21, GSM Association (GSMA), 2023. Version 3.1.
25. GSMA. RSP Technical Specification. Technical Report SGP.22, GSM Association (GSMA), 2023. Version 3.1.
26. GSMA. Post Quantum Cryptography – Guidelines for Telecom Use Cases. Technical Report PQ.03, GSM Association (GSMA), 2024. Version 2.0.
27. Andreas Hülsing, Daniel J. Bernstein, Christoph Dobraunig, et al. SPHINCS$^+$. Technical report, National Institute of Standards and Technology, 2020. https://sphincs.org/.
28. Seong Oun Hwang, Intae Kim, and Wai Kong Lee. *Modern cryptography with proof techniques and implementations*. CRC Press, 2021.
29. Matthias J. Kannwischer, Richard Petri, Joost Rijneveld, Peter Schwabe, and Ko Stoffelen. PQM4: Post-quantum crypto library for the ARM Cortex-M4, 2025. https://github.com/mupq/pqm4.
30. Vadim Lyubashevsky, Léo Ducas, Eike Kiltz, et al. CRYSTALS-DILITHIUM. Technical report, National Institute of Standards and Technology, 2020. https://pq-crystals.org/.
31. Shin'ichiro Matsuo, Kunihiko Miyazaki, Akira Otsuka, and David Basin. How to evaluate the security of real-life cryptographic protocols? the cases of iso/iec 29128 and cryptrec. In *International Conference on Financial Cryptography and Data Security*, pages 182–194. Springer, 2010.
32. Chris J Mitchell. The impact of quantum computing on real-world security: A 5g case study. *Computers & Security*, 93:101825, 2020.
33. NIST. Post-quantum cryptography - csrc, nist, 2025.

34. David Nowak. A framework for game-based security proofs. In *International Conference on Information and Communications Security*, pages 319–333. Springer, 2007.
35. Thomas Prest, Pierre-Alain Fouque, Jeffrey Hoffstein, et al. FALCON. Technical report, National Institute of Standards and Technology, 2020. https://falcon-sign.info/.
36. Peter Schwabe, Roberto Avanzi, Joppe Bos, et al. CRYSTALS-KYBER. Technical report, National Institute of Standards and Technology, 2020. https://pq-crystals.org/.
37. Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, Oct 1997.
38. Victor Shoup. Sequences of games: a tool for taming complexity in security proofs. Cryptology ePrint Archive, Paper 2004/332, 2004.
39. Douglas Stebila, Scott Fluhrer, and Shay Gueron. Hybrid key exchange in TLS 1.3. Internet-Draft draft-ietf-tls-hybrid-design-12, Internet Engineering Task Force, January 2025.
40. Vincent Quentin Ulitzsch, Shinjo Park, Soundes Marzougui, and Jean-Pierre Seifert. A post-quantum secure subscription concealed identifier for 6g. In *ACM WiSec'22*, pages 157–168, 2022.
41. Jing Yang and Thomas Johansson. An overview of cryptographic primitives for possible use in 5g and beyond. *Science China Information Sciences*, 63(12):1–22, 2020.

A Practical Instantiations

This section gives more details on practical instantiations of `PQC-RSP` outlined in Section 4.3, based on benchmark results from the literature.

Table 2. Performance and data sizes for PQC KEMs and the ECDH key exchange (KEX). The numbers for the SM-DP+ server are taken from the SUPERCOP benchmarks on an Intel i7-13700H processor [9]. The numbers for the eUICC are taken from the PQM4 project [29] for the PQC KEMs and from [20] for ECDH.

Scheme	Public Key	Ciphertext	**SM-DP+** Encaps/KEX (Mcycles)	**eUICC** KeyGen (Mcycles)	**eUICC** Decaps/KEX (Mcycles)
ECDH (X25519)	32 B	32 B	0.11	0.89	0.89
ML-KEM (kyber-512)	800 B	768 B	0.03	0.39	0.43
HQC (hqc-128)	2,249 B	4,433 B	0.48	52.71	159.57

Choice of PQC Schemes. We follow the standardization results of NIST. For KEMs, we consider ML-KEM (standardized; based on Kyber [36]) and HQC [5] (to be standardized). For signature schemes, we consider ML-DSA (standardized; based on Dilithium [30]), SLH-DSA (standardized; based on SPHINCS+ [27]), and FALCON [35] (to be standardized as FN-DSA). Currently, an additional

Table 3. Performance and data sizes for PQC signature schemes and ECDSA. The numbers for the SM-DP+ server are taken from the PQ-SORT benchmarks on an Intel i7-13700K processor [4]. The numbers for the eUICC are taken from the PQM4 project [29], from the SQIsign specification [3], and from [20] for ECDSA.

Scheme	Public Key	Signature	SM-DP+		eUICC	
			Sign (Mcycles)	Verify (Mcycles)	Sign (Mcycles)	Verify (Mcycles)
ECDSA (Ed25519)	32 B	64 B	0.04	0.13	0.54	1.33
ML-DSA (ml-dsa-44)	1,312 B	2,420 B	0.25	0.09	12.13	3.24
SLH-DSA (shake-128f-s.)	32 B	17,088 B	56.34	4.02	1,182.42	70.50
FALCON (falcon-512)	897 B	752 B	0.67	0.16	22.47	0.40
MAYO (MAYO2)	4,912 B	186 B	0.33	0.07	11.98	5.13
SQIsign (sqisign_lvl1)	65 B	148 B	103.04	5.17	–	123.00

NIST standardization process for digital signatures is ongoing. Therefore, we also consider MAYO [11] and SQIsign [3], two promising schemes that are based on different security assumptions, to give a more complete overview. As a baseline, we also give numbers for the currently used ECDH and ECDSA schemes. For all schemes and protocols, we restrict to the NIST-I security level.

Integration of PQC Schemes in `PQC-RSP`. We briefly give performance and bandwidth estimates when integrating PQC KEMs and signature schemes in `PQC-RSP`. As in Section 4.1, we additionally assume the usage of a PQC variant of TLS 1.3 and that the subsequent communication via symmetric encryption offers sufficient quantum resistance.

Integrating a PQC KEM. Instead of running one instance of the ECDH key exchange, here the eUICC runs $KEM.KeyGen$ and $KEM.Decapsulate$, while the SM-DP+ server runs $KEM.Encapsulate$, with one execution, respectively.

Table 2 suggests that ML-KEM outperforms HQC in all metrics, and hence is the best choice for the usage in `KEM-PQC-RSP`, both on the SM-DP+ server and the eUICC. Compared to the RSP protocol using ECDH, `KEM-PQC-RSP` with ML-KEM provides better performance both on the SM-DP+ server and the eUICC. In terms of bandwidth, `KEM-PQC-RSP` needs to transmit a public key and ciphertext of 800 B and 768 B, respectively, compared to the 32 B public keys in ECDH. Note that communication to the eUICC is done through the Application Protocol Data Unit (APDU) protocol, which — in some versions — has a maximum package size of 256 bytes [26]. As such, more APDU packages will be required for `KEM-PQC-RSP`, which will lead to a moderate increase in latency. However, this increase is negligible compared to the 32 KB size of a typical SIM profile [26].

In `hybrid-PQC-RSP`, we run both the KEM and an ECDH key exchange. Due to the small key sizes and fast performance of ECDH, we expect this variant to be as practical as `KEM-PQC-RSP`, since ML-KEM's key and ciphertext size will be the bottleneck.

Assuming the long-term security of ML-KEM, its usage in `KEM-PQC-RSP` prevents *Harvest Now, Decrypt Later* attacks. Additionally, `hybrid-PQC-RSP` provides classical security in case practical classical attacks against the PQC KEM are found. We expect both variants to remain practical in terms of performance.

Integrating PQC signatures. To provide security also against active quantum attacks, we need to include PQC signatures and certificates in `KEM-PQC-RSP`. However, while key establishment through a KEM occurs only once per execution, the RSP protocol includes several certificates and signatures. In particular, the SM-DP+ server and the eUICC verify two certificates and generate and verify three or two signatures, respectively. Hence, overhead in performance or bandwidth for PQC signatures and certificates has more impact on the overall protocol than for PQC KEMs. Signing and verification use key pairs from the certificates, and therefore public keys do not have to be transmitted again. Hence, signatures and certificates must use the same PQC scheme, and we prioritize small signature sizes over small public keys.

It is evident from Table 3 that none of the schemes achieves the performance or bandwidth of ECDSA. Especially SLH-DSA seems not to be suitable for our purposes due to the large signature sizes and slow signing. ML-DSA, FALCON, and MAYO share a relatively similar performance and bandwidth profile, which appears to be most suitable due to the balance between performance and data sizes. All three schemes seem to lead to practical performance overall, yet it is difficult to estimate which scheme will perform best due to the different advantages. In contrast, SQIsign [3] offers smaller combined signature and public key sizes, and thus seems promising especially for certificates. However, it is too slow to be considered as an alternative in `PQC-RSP` in its current state.

It is evident that PQC signatures and certificates lead to a significant overhead compared to RSP using ECDSA: on the SM-DP+ server, the running time per round of signing and verifying increases by a factor of up to 5, and on the eUICC this factor reaches up to 12 for the schemes we consider (ML-DSA, FALCON, and MAYO). Furthermore, the increased signature and public key sizes may lead to additional latency, due to a higher number of APDU packages, bottlenecking communication. Taking this into account, using FALCON might be preferable, since its combined public key and signature size is less than half compared to ML-DSA. On the other hand, MAYO's signatures fit in a single APDU package for the MAYO2 parameter set, and since more signatures than public keys are transmitted, this might be preferable overall. In any case we estimate that for any of the three considered signature schemes, `KEM-PQC-RSP` would remain feasible, although the increase in APDU packages may impact the user experience in terms of a longer running time of the eSIM provisioning. We leave the practical analysis of the exact performance implications for future work.

B Proof for Theorem 1

Proof. We prove mutual authentication via a sequence of games, defining a series of events S_i where S_i denotes the event that the adversary $\mathcal{A}$ wins in Game i.

Game 0: This is the real mutual authentication game for PQC-RSP.

- The challenger generates all keys and certificates:
  - Root CA key: (PK_{CI}, SK_{CI})
  - EUM key: (PK_{EUM}, SK_{EUM}) and certificate $Cert_{EUM} = \mathsf{Sign}(PK_{EUM}, \text{EUM-info}; SK_{CI})$
  - Server keys: (PK_{Sp}, SK_{Sp}) with certificates $Cert_{Sp}$ signed by SK_{CI}
  - Client keys: (PK_U, SK_U) and certificate $Cert_U = \mathsf{Sign}(PK_U, U; SK_{EUM})$
- $\mathcal{A}$ interacts with honest eUICC and SM-DP+ server instances in polynomially many sessions.
- $\mathcal{A}$ wins if it causes either server to accept without valid client authentication or client to accept without valid server authentication.

By definition, $\mathsf{Adv}^{\mathsf{auth}}_{\mathcal{A}}(\lambda) = \Pr[S_0]$.

Game 1: We change Game 0 to abort if $\mathsf{Verify}(\mathsf{Cert}^*_{\mathsf{EUM}}, PK_{CI}) = 1$ and $\mathsf{Cert}^*_{\mathsf{EUM}} \notin \mathcal{Q}_{\mathsf{CI}}$, where $\mathcal{Q}_{\mathsf{CI}} = \{\mathsf{Cert}_{\mathsf{EUM}}\}$ is the set of CI certificates issued. Let F_1 be that event.

Claim. Games 0 and 1 are identical unless F_1 occurs. Moreover, $|\Pr[S_0] - \Pr[S_1]| \leq \Pr[F_1] = \mathsf{Adv}^{\mathsf{EUF-CMA}}_{\mathcal{B}_1}(\lambda)$.

Proof. We construct adversary $\mathcal{B}_1$ against the CI signature scheme. $\mathcal{B}_1$ receives PK_{CI} and a signing oracle. It simulates Game 1 for $\mathcal{A}$: it generates all other keys honestly and uses its oracle to produce $\mathsf{Cert}_{\mathsf{EUM}}$. It records all certificates issued by the oracle in $\mathcal{Q}_{\mathsf{CI}}$. When $\mathcal{A}$ outputs its transcript, $\mathcal{B}_1$ checks whether $\mathsf{Verify}(\mathsf{Cert}^*_{\mathsf{EUM}}, \mathsf{pk}_{\mathsf{CI}}) = 1$ and $\mathsf{Cert}^*_{\mathsf{EUM}} \notin \mathcal{Q}_{\mathsf{CI}}$. If so, $\mathcal{B}_1$ outputs $(\mathsf{pk}^*_{\mathsf{EUM}}, \mathsf{Cert}^*_{\mathsf{EUM}})$ as its EUF-CMA forgery. The simulation is perfect, and $\mathcal{B}_1$ succeeds exactly when F_1 occurs. Hence $\Pr[F_1] = \mathsf{Adv}^{\mathsf{EUF-CMA}}_{\mathcal{B}_1}(\lambda)$.

Game 2: We change Game 1 to abort if $\mathsf{Verify}(\mathsf{Cert}^*_{\mathsf{U}}, PK_{EUM}) = 1$ and $\mathsf{Cert}^*_{\mathsf{U}} \notin \mathcal{Q}_{\mathsf{EUM}}$, where $\mathcal{Q}_{\mathsf{EUM}} = \{\mathsf{Cert}_{\mathsf{U}}\}$ is the set of EUM certificates issued. Let F_2 be that event.

Claim. Games 1 and 2 are identical unless F_2 occurs. Moreover, $|\Pr[S_1] - \Pr[S_2]| \leq \Pr[F_2] = \mathsf{Adv}^{\mathsf{EUF-CMA}}_{\mathcal{B}_2}(\lambda)$.

Proof. We construct adversary $\mathcal{B}_2$ against the EUM signature scheme. $\mathcal{B}_2$ receives PK_{EUM} and a signing oracle. It simulates Game 2 for $\mathcal{A}$: it generates all other keys honestly and uses its oracle to produce $\mathsf{Cert}_{\mathsf{U}}$. It records all certificates issued by the oracle in $\mathcal{Q}_{\mathsf{EUM}}$. When $\mathcal{A}$ outputs its transcript, $\mathcal{B}_2$ checks whether $\mathsf{Verify}(\mathsf{Cert}^*_{\mathsf{U}}, PK_{EUM}) = 1$ and $\mathsf{Cert}^*_{\mathsf{U}} \notin \mathcal{Q}_{\mathsf{EUM}}$. If so, $\mathcal{B}_2$ outputs $(PK^*_U, \mathsf{Cert}^*_{\mathsf{U}})$ as its forgery. The simulation is perfect, so $\Pr[F_2] = \mathsf{Adv}^{\mathsf{EUF-CMA}}_{\mathcal{B}_2}(\lambda)$.

Game 3: We change Game 2 to abort if an honest client accepts a transcript with $\mathsf{Verify}(\sigma^*_{Sp}, PK^*_{Sp}) = 1$ and $\sigma^*_{Sp} \notin \mathcal{Q}_{Sp}$, where $\mathcal{Q}_{Sp}$ is the set of server signatures produced during the game. Let F_3 be that event.

Claim. Games 2 and 3 are identical unless F_3 occurs. Moreover,
$|\Pr[S_2] - \Pr[S_3]| \leq \Pr[F_3] = \mathsf{Adv}^{\mathsf{EUF-CMA}}_{\mathcal{B}_3}(\lambda)$.

Proof. We construct adversary $\mathcal{B}_3$ against the server's signature scheme. $\mathcal{B}_3$ receives PK_{Sp} and a signing oracle. It simulates Game 3 for $\mathcal{A}$: it generates all other keys honestly and uses its oracle to produce server signatures when needed. It records all signatures generated by the oracle in $\mathcal{Q}_{Sp}$. When $\mathcal{A}$ outputs a transcript that would cause an abort, $\mathcal{B}_3$ checks whether σ^*_{Sp} is a fresh valid signature; if so, it outputs σ^*_{Sp} as its forgery. The simulation is perfect, so $\Pr[F_3] = \mathsf{Adv}^{\mathsf{EUF-CMA}}_{\mathcal{B}_3}(\lambda)$.

Game 4: We change Game 3 to abort if an honest server accepts a transcript with $\mathsf{Verify}(\sigma^*_{\mathsf{U}}, PK^*_U) = 1$ and $\sigma^*_{\mathsf{U}} \notin \mathcal{Q}_{\mathsf{U}}$, where $\mathcal{Q}_{\mathsf{U}}$ is the set of client signatures produced during the game. Let F_4 be that event.

Claim. Games 3 and 4 are identical unless F_4 occurs. Moreover,
$|\Pr[S_3] - \Pr[S_4]| \leq \Pr[F_4] = \mathsf{Adv}^{\mathsf{EUF-CMA}}_{\mathcal{B}_4}(\lambda)$.

Proof. Analogous, using a signing oracle for PK_U.

Final analysis: In Game 4, all accepted certificates and signatures are honest, so the adversary cannot win: $\Pr[S_4] = 0$.
Hence $\mathsf{Adv}^{\mathsf{auth}}_{\mathcal{A}}(\lambda) = \Pr[S_0] \leq \sum_{i=1}^{4} \mathsf{Adv}^{\mathsf{EUF-CMA}}_{\mathcal{B}_i}(\lambda)$. □