

Protocol Constraints on Post-Quantum Migration of Offline Authentication in Android HCE Payment System

Chaima EHBALI , Ali Ouacha , Ahmed El-Yahyaoui 

Faculty of Science, Mohammed V University in Rabat, Rabat, Morocco

Abstract—Mobile payment systems built on Host Card Emulation (HCE) inherit cryptographic foundations, such as RSA-based offline data authentication, that were designed long before quantum attacks were a concern. NIST has now finalized its first post-quantum cryptography (PQC) standards, and payment operators are starting to plan migrations, yet little work has looked at how PQC signatures fit into the protocol messages of wallet-based offline authentication. This study examines whether standardized post-quantum signatures can replace RSA in EMV offline data authentication without changing the data-object encoding, the short-form APDU exchange, or the terminal kernel. We evaluate a non-tokenized Android HCE card profile that supports Combined DDA/Application Cryptogram Generation (CDA), the offline authentication method required by the current Kernel 4 specification. A protocol state-machine model is reconstructed from public specifications, and the full set of offline-authentication objects, including the certificate chain, is budgeted against the protocol limits. The smallest standardized PQC signature, ML-DSA-44, produces a GENERATE AC response of at least 2438 bytes where the short-form limit is 256 bytes, and each ML-DSA-44 certificate needs at least 3732 bytes against a 254-byte record limit, since ML-DSA offers no message recovery. At the only bit rate EMV allows, sending this response alone takes about 184 ms of the 400 ms that EMV allocates to the card. Protocol framing, rather than computation, is therefore the main obstacle.

Keywords—Mobile payment security; post-quantum cryptography; NFC protocol constraints

I. INTRODUCTION

Card-based and mobile payment systems process a large share of global commerce on cryptographic foundations established in the 1990s and early 2000s. The EMVCo Integrated Circuit Card Specifications designated RSA for offline card authentication and Triple-DES (3DES) for online transaction cryptograms, while elliptic-curve Diffie–Hellman and elliptic-curve (EC) signatures secure the transport-layer channels linking acquirers, payment networks, and issuers [1]. Shor’s algorithm efficiently solves the factorization and discrete-logarithm problems underlying RSA and EC cryptography, breaking both completely on a sufficiently powerful quantum computer. Grover’s algorithm gives only a quadratic speed-up against symmetric primitives such as 3DES and AES, halving their effective security without breaking them outright [2], [3]. NIST finalized the first set of post-quantum standards in 2024 [4], [5], [6]. Hybrid post-quantum transport-layer security is already deployed at scale: Cloudflare supports

the hybrid X25519MLKEM768 key agreement on all TLS 1.3 connections to its network and has moved most of its internal connections to post-quantum cryptography [7]. Infrastructure-level PQC work, such as hardware security modules and hybrid-signature proposals, is comparatively mature [8], whereas software-level integration of PQC into wallet-centric payment protocols is rare, whether on-device or backend-mediated, and typically framed as generic embedded or IoT cryptography, and remains comparatively unexamined in the literature [9], [10], [11], [12], [13], [14].

Existing studies on post-quantum migration of payment systems primarily address cryptographic algorithm replacement, backend infrastructures, or high-level migration strategies [15], [16], [17], [18], [19], [20]. However, little attention has been paid to whether standardized PQC signatures can be integrated into the protocol constraints of Android HCE offline authentication. In particular, to the best of our knowledge, existing studies have not systematically analyzed the impact of protocol message-size limits on the direct substitution of RSA signatures within Static Data Authentication (SDA), Dynamic Data Authentication (DDA), and Combined DDA/Application Cryptogram Generation (CDA). This study addresses that gap through a protocol-level analysis of EMV offline authentication as implemented in Android HCE systems. Because the constraint concerns the message that carries the signature to the terminal, it does not depend on where the signature is generated.

The evaluated deployment is fixed as follows. The reference profile is a non-tokenized Android HCE application that emulates an EMV card and performs offline data authentication as specified in EMV Book 2 [1]. We take CDA as the primary case for two reasons. First, its Signed Dynamic Application Data (tag ‘9F4B’) has to be returned inside the response to the GENERATE AC command, which is the response most tightly bounded by the 256-byte short-form limit [1, Annex D1]. Second, the 2026 update of the Kernel 4 specification makes CDA the only offline authentication method that readers must support, performed as described in Sections V and VI of EMV Book 2 version 4.4 [21]. SDA and DDA are treated as secondary cases. Their certificates and signed static data are stored in records limited to 254 bytes, and the DDA signature is returned in the INTERNAL AUTHENTICATE response, which has the same 256-byte limit [1, Annex D1]. The analysis is defined at the level of EMV Book 2 and ISO/IEC 7816-4 [22] and does not model kernel-specific data placement. Network-tokenized wallets such as Google Pay and Samsung Pay are outside the

evaluated scope because their public documentation describes only symmetric cryptograms, derived from limited-use keys, at tap time [23], [24], [25]. If such a wallet kept offline authentication with signatures precomputed during provisioning, those signatures would still have to reach the terminal at tap time and would meet the same constraint (Section IV-C). This scoping agrees with EMVCo's view that, for EMV payments, asymmetric cryptography is mainly needed for offline operation [26], [27].

The study's structure is as follows: Section II presents the background on Android HCE and related research. Section III describes the methodology, and Section IV develops the constraint model. Section V presents the threat analysis and evaluates the mitigation strategies. Section VI discusses the findings, and Section VII concludes the study.

II. BACKGROUND

This section introduces the three elements the constraint analysis depends on. Subsection A explains how Android HCE delivers APDUs to a payment application and which EMV offline authentication mechanisms are involved. Subsection B describes the hardware-backed key storage available to such applications, and Subsection C positions this work against existing research on EMV security, HCE, and post-quantum migration.

A. Host Card Emulation and Offline Authentication

Host Card Emulation enables a standard application to emulate a contactless smart card without a carrier-controlled SIM or embedded hardware Secure Element [28]. The Android NFC stack routes Application Protocol Data Units (APDUs) addressed to a registered Application Identifier (AID) to the application's HostApuService instead of to an embedded Secure Element [29]. This architecture removed prior SIM-based and manufacturer-based gatekeeping, and it also became the foundation of Google Pay, Samsung Pay, and most bank-issued Android payment applications [23], [24]. To protect sensitive operations, HCE relies on hardware-backed key storage through the Android Keystore and, in many deployments, on network tokenization [30], [31]. Consequently, PQC migration in HCE cannot be analyzed by treating the smartphone as a monolithic hardware token. Instead, it must consider the interactions between the Android application, the hardware-backed keystore, and the back-end tokenization infrastructure [32].

Fig. 1 summarizes the Android HCE contactless transaction analyzed in this study as an eight-stage workflow, from application selection to transaction completion. Stages 1 to 5, 7, and 8 are steps of the transaction protocol, while State 6 stands for the acquirer-issuer communication channel; it is shown because it is the other part of the transaction that depends on Shor-vulnerable cryptography. Each state is labeled with its dominant cryptographic primitive and quantum-vulnerability class. One remark helps in reading the figure. In CDA, the dynamic signature is returned together with the application cryptogram, so for CDA the offline-authentication step spans State 3 (reading the certificates) and State 5 (GENERATE AC). As Fig. 1 shows, only two states depend on public-key cryptography that is vulnerable to Shor's algorithm: offline

authentication (State 3) and the issuer communication channel (State 6) [1], [33]. Application cryptogram generation and verification rely on symmetric cryptography and are affected only by Grover's algorithm. We focus on offline authentication because it is the only Shor-vulnerable operation whose message format is fixed by the device-terminal protocol. The acquirer-issuer channel uses network protocols whose move to hybrid post-quantum key exchange is already underway and is not bounded by APDU framing [7], [34], and EMVCo identifies offline operation as the setting where asymmetric cryptography matters for EMV payments [26].

The offline authentication mechanisms defined in EMV Book 2 consist of SDA, DDA, and CDA. The version analyzed in this work specifies these mechanisms exclusively using RSA digital signatures with message recovery [1]. With message recovery, part of the signed data is recovered from the signature itself, so every certificate and signature takes exactly the length of the RSA modulus, at most 248 bytes (1984 bits) [1, Annex B2]. Later EMV specifications introduced elliptic-curve cryptography [35], [36], but the analyzed protocol remains the RSA-based design. EMVCo currently recommends an anticipated lifetime of at least 31 December 2036 for 1984-bit RSA certification authority keys [37], so RSA-based offline authentication will stay deployed during the period in which migration has to be planned. Online authorization relies on application cryptograms generated using symmetric cryptography derived from issuer-specific master keys. Since these operations use symmetric primitives rather than public-key cryptography, they are affected by Grover's algorithm rather than Shor's algorithm. 3DES is also being retired for classical reasons: NIST disallowed three-key TDEA for encryption after 31 December 2023 [38], and EMVCo supports AES-128 as the entry-level symmetric algorithm [26]. As shown in Fig. 1, this distinction means that application cryptograms are a lower priority for post-quantum migration than offline authentication.

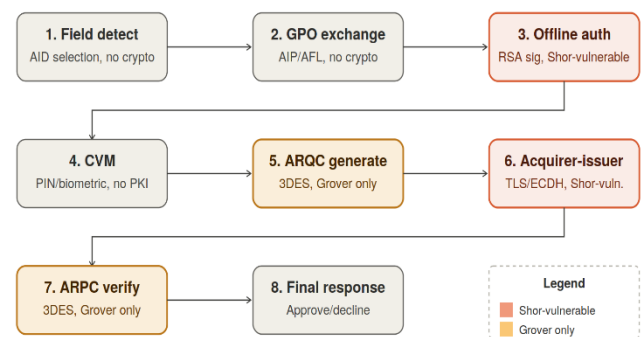


Fig. 1. Android HCE contactless transaction flow annotated with its dominant cryptographic primitive and quantum-vulnerability class.

B. Android Keystore and Hardware-Backed Keys

The Android Keystore is a system that facilitates the generation, storage, and execution of cryptographic operations in a secure manner, thereby ensuring that private keys remain protected from exposure to application memory [30]. The protection of keys is dependent on the capabilities of the device in question. Keys may be protected by a Trusted Execution Environment (TEE), usually built on ARM TrustZone as an

isolated part of the application processor, or by StrongBox, a separate secure element. The StrongBox KeyMint implementations studied in [39] support classical algorithms but not ML-KEM or ML-DSA. This is starting to change. Google has announced native ML-DSA support in Android Keystore from Android 17, executed in the TEE and limited to ML-DSA-65 and ML-DSA-87 [40]. An HCE application that keeps its card key in the Keystore would therefore use at least ML-DSA-65, whose CDA response needs at least 3327 bytes (Table I). On earlier Android versions, PQC signing would have to run in software inside the application process, without hardware key isolation, or in vendor-specific trusted applications.

C. Positioning Within Literature

The security of the EMV protocol has been widely studied. Previous research work has investigated relay attacks, pre-play attacks, authentication weaknesses, and implementation vulnerabilities in both contact and contactless payment [41]. These studies improve the understanding of classical EMV security, assuming the use of existing cryptographic algorithms. The practical implications of deploying standardized post-quantum signatures in this protocol are not yet well understood.

Android HCE has been extensively adopted as the software platform for contactless mobile payment. Existing HCE research has focused on system architecture, tokenization, secure storage, performance, and security [31], [32]. The impact of post-quantum cryptography on the protocol-level constraints of HCE offline authentication has received much less attention, even though transaction sequencing imposes strict implementation constraints [33].

Performance of standardized PQC algorithms has also been the subject of evaluation in the context of constrained embedded devices, with a particular focus on ARM Cortex-M microcontrollers [9], [10], [42]. Optimized implementations have also been reported for ARMv8-A application processors [43], and PQC libraries have been benchmarked on Android and iOS smartphones [44]. These results describe computation rather than protocol framing, and they cannot be applied directly to Android HCE, where the payment application runs within a fixed message structure. Post-quantum migration research has, in short, looked at algorithm performance far more than at the protocol-level requirements of mobile payment systems. This study analyzes the feasibility of implementing standardized post-quantum digital signatures in State 3; see Fig. 1. Instead of focusing on cryptographic algorithms or the system's performance, the analysis examines the impact of protocol message constraints on the practical deployment of PQC within the Android HCE transaction workflow.

III. METHODOLOGY

Our work is a theoretical characterization study. The objective is to analyze whether standardized post-quantum digital signatures can be integrated into the existing protocol structure of Android Host Card Emulation (HCE) offline authentication. Therefore, the study focuses on protocol behavior, implementation constraints, and deployment architecture rather than computational performance.

The methodology follows a protocol-analysis approach and has six consecutive stages between the research question and the

findings, as illustrated in Fig. 2. Stage 1 collects and cross-checks evidence from three categories of primary sources. Stage 2 reconstructs the Android HCE transaction as a protocol state machine. Stage 3 identifies the states that rely on quantum-vulnerable public-key cryptography. Stage 4 maps the standardized PQC algorithms onto these operations. Stage 5 analyzes the resulting protocol constraints along four dimensions: APDU size, timing, hardware support, and execution location. Stage 6 evaluates the threat model and the candidate mitigation strategies. The constraint model is thus derived from documented protocol behavior and standardized algorithm characteristics.

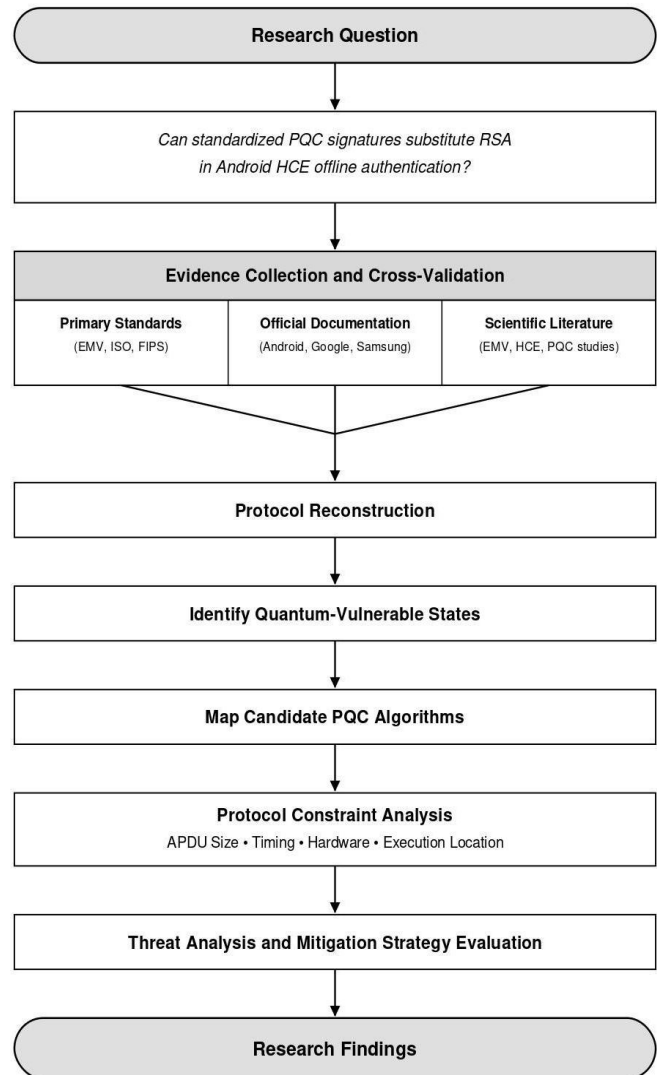


Fig. 2. Research methodology for evaluating post-quantum signature migration in Android HCE offline authentication.

In Stage 1, we used three complementary categories of primary sources. The first consists of official protocol specifications, including the EMV Integrated Circuit Card Specifications [1], the EMV contactless specifications [45], [46], and the current Kernel 4 specification bulletin [21], which define offline authentication and the contactless interface; ISO/IEC 7816-4, which defines APDU message exchanges [22]; and the official standards for the standardized post-quantum

algorithms, namely FIPS 203, FIPS 204, and FIPS 205 [4], [5], [6]. The second category comprises official implementation documentation published by Android, Google, and Samsung describing Android HCE architecture, Android Keystore, StrongBox, and tokenization payment deployment. The third category includes peer-reviewed literature and publicly available scientific studies to position the work within existing research, compare implementation assumptions, and identify measurements unavailable in the primary specifications, such as current benchmark results for standardized PQC algorithms. The information was verified for consistency whenever multiple documents described the same protocol component. Informal sources, such as blogs, forums, and news articles, were excluded; official vendor documentation pages were kept as primary sources for implementation behavior.

In Stage 2, the Android HCE payment workflow was reconstructed from the collected sources as a protocol state machine, from application selection to transaction completion. Each protocol state was annotated with its dominant cryptographic primitive, execution environment, quantum-vulnerability class, and the protocol message that carries the cryptographic operation. In Stage 3, this reconstruction identifies the protocol stage where public-key cryptography directly influences transaction acceptance, which sets the scope of the rest of the analysis.

Stages 4 and 5 evaluate whether standardized PQC algorithms satisfy the requirements imposed by the reconstructed protocol. Four dimensions are analyzed: protocol message-size limits, transaction timing, hardware-backed key support, and the execution location of the cryptographic operation, which was examined as a possible confound and found not to change the applicability of the size constraint. Rather than evaluating algorithmic security in isolation, we look at the interaction between standardized PQC signatures and the surrounding protocol architecture. The resulting constraint model is the basis for Stage 6, the threat analysis and comparison of candidate migration strategies in Section V.

IV. PROTOCOL CONSTRAINT MODEL FOR POST-QUANTUM OFFLINE AUTHENTICATION

Our analysis identifies complementary constraints that determine the feasibility of introducing standardized PQC signatures into Android HCE offline authentication. The first is a quantified mismatch between the size of standardized PQC signatures, certificates, and public keys and the message sizes allocated to offline authentication. The second comprises timing constraints, whose practical impact depends on the amount of data transferred and on the number of protocol exchanges. Successful migration depends on how protocol design, implementation architecture, and platform capabilities interact.

A. Candidate PQC Primitives

The first generation of NIST-standardized post-quantum cryptographic algorithms comprises the key-encapsulation mechanism (FIPS 203), the lattice-based digital signature scheme (FIPS 204), and the stateless hash-based digital signature scheme (FIPS 205) [4], [5], [6]. Each algorithm provides different cryptographic functionality, and only some of them are suitable candidates for replacing the RSA-based offline

authentication mechanisms defined by EMV. ML-KEM is for key establishment and therefore applies to applications such as transport-layer security and token provisioning. Consequently, it is not considered further in the offline-authentication analysis. ML-DSA is a lattice-based digital signature scheme with three security parameter sets (ML-DSA-44, ML-DSA-65, ML-DSA-87). Its public keys range from 1312 to 2592 bytes and its signatures from 2420 to 4627 bytes [5]. Because RSA is currently used to generate offline authentication signatures, ML-DSA represents the most direct standardized replacement and therefore the primary focus of this study. SLH-DSA is a stateless hash-based digital signature scheme that provides a conservative security assumption at the expense of considerably larger signatures. Depending on the parameter set, SLH-DSA signatures range from 7856 bytes (SLH-DSA-SHA2-128s) to 49856 bytes (SLH-DSA-SHA2-256f) [6]. For this reason, SLH-DSA is even less compatible with protocols constrained by fixed message sizes [6], [47].

Two further families of signature schemes are relevant, although the finalized standards do not cover them. FN-DSA, the upcoming standard based on Falcon, has 666-byte signatures and 897-byte public keys at its lowest security level [48], but its standard had not been finalized at the time of writing. Among the third-round candidates of NIST's additional-signature process, HAWK reaches 555-byte signatures, SQIsign 148-byte signatures, and UOV 96-byte signatures at security category 1 [49]. FN-DSA and HAWK would still exceed the 240-byte signature budget of the CDA response, whereas SQIsign and UOV signatures would fit. These two candidates have other costs. UOV public keys exceed 200 kilobytes in expanded form, which rules out certificates of the EMV kind. SQIsign has the smallest combined public-key and signature size of the second-round candidates, but its signing latency is higher than that of the other candidates [49]. Since neither scheme is standardized and both are still under evaluation, the rest of this study focuses on FIPS 204 and FIPS 205; Section V-B comes back to smaller-signature schemes as a mitigation path.

B. Protocol Size Constraint

Cryptographic equivalence alone is not enough to replace RSA with a standardized post-quantum signature. The new signature must also respect the message format defined by the EMV specifications [1], [22]. The first question of this study is therefore whether standardized PQC signatures fit within the response structure allocated to offline authentication. The relevant limits are given in EMV Book 2, Annex D1: a response may carry at most 256 data bytes, and records are limited to 254 bytes [1, Annex D1]. The 256-byte figure is the short-form limit of ISO/IEC 7816-4, in which an expected length field $Le = '00'$ requests up to 256 data bytes [22]; the two-byte status word (SW1 SW2) is sent in addition to these data bytes. This limit is not a choice made by this study. It is the limit under which the unmodified protocol operates, and departures from it, such as extended-length APDUs or chaining, are evaluated as protocol changes in Section V-B. In the CDA response to GENERATE AC (format 2, template '77'), 16 bytes are taken by the response template, the Cryptogram Information Data (CID), the Application Transaction Counter (ATC), and the tag and length of the Signed Dynamic Application Data. This leaves at most 240 bytes for the signature, or 205 bytes when the maximum 32-

byte Issuer Application Data (IAD) is included [1, Annex D1.2.1]. Table I compares these limits with the size of standardized PQC signatures [1], [5], [6].

For each scheme, Table I gives the FIPS parameter set, the signature length, and the minimum length of the encoded CDA response to GENERATE AC, computed as:

$$L_{resp} = |sig| + 18 \quad (1)$$

where, L_{resp} is the encoded response length in bytes, $|sig|$ is the signature length defined in FIPS 204 or FIPS 205 for the parameter set, and 18 is the number of bytes taken by the fixed part of the response when no IAD is present. The fixed part grows from 16 to 18 bytes because lengths above 255 bytes need a three-byte length field in BER-TLV (Basic Encoding Rules, tag-length-value) encoding, both for the response template and for the Signed Dynamic Application Data [22]. The overhead factor is then

$$r = \frac{L_{resp}}{L_{APDU}} \quad (2)$$

where, r is the overhead factor and $L_{APDU} = 256$ bytes is the short-form response limit [1], [22]. All factors are rounded to one decimal. With the maximum IAD, a further 35 bytes would be added to L_{resp} .

TABLE I. CDA SIGNATURE (TAG '9F4B') IN THE GENERATE AC RESPONSE AGAINST THE 256-BYTE SHORT-FORM LIMIT; $L_{resp} = |sig| + 18$ BYTES, WITHOUT ISSUER APPLICATION DATA (IAD)

Scheme (FIPS parameter set)	$ sig / L_{resp}$ (bytes)	Fits 256-byte response ($L_{resp} \div 256$)
RSA, modulus ≤ 248 bytes [1]	$\leq 240 / \leq 256$ (≤ 205 with max. IAD)	Yes
ML-DSA-44 (FIPS 204)	2420 / 2438	No (9.5 $\times$)
ML-DSA-65 (FIPS 204)	3309 / 3327	No (13.0 $\times$)
ML-DSA-87 (FIPS 204)	4627 / 4645	No (18.1 $\times$)
SLH-DSA-SHA2/SHAKE-128s (FIPS 205)	7856 / 7874	No (30.8 $\times$)
SLH-DSA-SHA2/SHAKE-128f (FIPS 205)	17088 / 17106	No (66.8 $\times$)

The comparison shows that the smallest standardized PQC signature, ML-DSA-44 (2420 bytes), produces an encoded response of at least 2438 bytes, 9.5 times the available response space. Larger ML-DSA parameter sets and all standardized SLH-DSA variants exceed it by much larger margins. The signature is, however, only one of the objects that offline authentication has to carry, and Table II extends the budget to the complete object set of SDA, DDA, and CDA. ML-DSA and SLH-DSA do not provide the ISO/IEC 9796-2 message recovery used by EMV RSA signatures, so every signed data element that RSA recovers from the signature must be sent in addition; in CDA, this includes part of the 32 to 38 leftmost bytes of the ICC Dynamic Data, such as the application cryptogram [1, Table 18]. Certificates are hit hardest, because the certified public key has to travel alongside the signature. An ML-DSA-44 Issuer or ICC Public Key Certificate therefore needs at least $2420 + 1312 = 3732$ bytes, almost 15 times the 254-byte record limit that already restricts RSA certificates to 247 bytes [1, Annex D1.1], [5]. Terminals would also have to

store ML-DSA certification authority public keys of at least 1312 bytes each, instead of RSA moduli of at most 248 bytes.

TABLE II. OFFLINE-AUTHENTICATION OBJECTS UNDER RSA AND ML-DSA-44 (LOWER BOUNDS; DATA RECOVERED BY RSA MESSAGE RECOVERY MUST BE ADDED FOR ML-DSA)

EMV object (tag)	Carried in	Limit (bytes)	RSA (bytes)	ML-DSA-44, lower bound (bytes)
Issuer Certificate ('90') PK	READ RECORD	254 per record	≤ 248	3732 (2420 + 1312)
ICC Certificate ('9F46') PK	READ RECORD	254 per record	≤ 247	3732 (2420 + 1312)
SSAD ('93'), SDA	READ RECORD	254 per record	≤ 248	2420
SDAD ('9F4B'), DDA	INTERNAL AUTHENTICATION response	256	≤ 248	2420 + ICC Dynamic Data
SDAD ('9F4B'), CDA	GENERATE AC response	256	≤ 240 (≤ 205 with IAD)	2438 + ICC Dynamic Data
CA public key	Terminal storage	—	≤ 248	1312 per key

ISO/IEC 7816-4 [22] defines extended-length APDUs, in which a response may carry up to 65536 data bytes, and ISO/IEC 14443-4 [33] carries such APDUs through block chaining at the transmission-protocol level. This would fit standardized PQC signatures into a single command-response pair. However, EMV Book 2 assumes the short-form limit [1, Annex D1], and the public sources we reviewed do not document support for extended-length APDUs in deployed contactless terminals and kernels. On Android-based readers, such support can be queried with `IsoDep.isExtendedLengthAduSupported()` and `IsoDep.getMaxTransceiveLength()` [50], but we did not measure these values in this study. Terminal compatibility is therefore treated as an open deployment question.

C. Execution Location of the Constraint Operation

The protocol constraint applies wherever an offline-verifiable signature is delivered to the terminal in an EMV-formatted response, whether the signature was generated during the contactless transaction itself or during backend provisioning. To understand the practical impact of the identified constraint, it is essential to determine where this cryptographic operation is actually performed. Google's device tokenization documentation states that Android devices store the primary key used to generate transaction cryptograms locally, while sensitive card information remains protected through Android Keystore and network tokenization [23]. Similarly, Samsung Pay documentation states that payment credentials and transaction keys are protected within the device Trusted Execution Environment (TEE), where the transaction cryptogram is generated during payment [24]. Neither source describes the generation of RSA signatures corresponding to the EMV DDA or CDA mechanisms. Instead, both describe symmetric cryptographic operations consistent with the Application Cryptogram process discussed in Section II.

This suggests that, in current tokenized HCE deployments, public-key operations happen mainly during backend token provisioning rather than during the tap-time transaction. Under this approach, the smartphone performs only lightweight symmetric operations during payment. This narrows the scope of the constraint model rather than weakening it. The APDU response-size limit applies to the exchange that delivers a signature to an offline-verifying terminal, not to the place where the signature is generated. Provisioning itself runs over network channels and is not subject to the APDU limit. In non-tokenized HCE implementations, the exchange that carries the signature is the tap-time transaction. In a tokenized deployment, a signature precomputed during provisioning would still have to be returned to the terminal at tap time if offline verification were kept; if the deployment relies instead on online verification of the symmetric cryptogram, offline authentication, and with it the size constraint, is not exercised. Because the public documentation does not show which of these configurations production wallets use, tokenized deployments are left outside the evaluated scope.

D. Timing Constraints

We identify the protocol-size limit as the primary constraint. Timing matters for any strategy that transmits a larger signature, because every extra byte has to cross the contactless link while the phone is held to the reader. A lower bound on this transmission time follows from the bit rate of the link:

$$T_{tx} = \frac{8 \cdot L_{resp}}{R} \quad (3)$$

where, T_{tx} is the transmission time in seconds, L_{resp} is the response length in bytes, 8 is the number of bits per byte, and R is the bit rate in bit/s. EMV fixes R at $fc/128 \approx 106$ kbit/s in both directions, where $fc = 13.56$ MHz is the carrier frequency of the reader field [51], [46, Req. 4.2.1.1]. These values are therefore not a pessimistic assumption but the operating point of EMV readers. Frame headers, parity bits, error-detection codes, and processing time are ignored, so the real time is higher. For RSA (256 bytes), Eq. (3) gives about 19 ms. For ML-DSA-44 (2438 bytes), it gives about 184 ms; for ML-DSA-65 (3327 bytes), about 251 ms; and for SLH-DSA-128s (7874 bytes), about 595 ms. EMV Contactless Book A limits the time a card may stay in the reader field to 500 ms, of which at most 100 ms goes to the reader; this leaves a card tariff of 400 ms that includes the card's own transmission time [45, Sect. 10.1, 10.2]. The ML-DSA-44 response would use about 46% of this card tariff and ML-DSA-65 about 63%, while SLH-DSA-128s would not fit at all. If the signature is instead split into short-form exchanges, the number of exchanges is

$$n = \left\lceil \frac{L_{resp}}{256} \right\rceil \quad (4)$$

where, n is the number of command-response pairs and $\lceil \cdot \rceil$ denotes rounding up to the next integer. The transmission time stays the same, and each of the $n - 1$ extra exchanges adds at least a 5-byte GET RESPONSE command and a 2-byte status word (about 0.53 ms at 106 kbit/s), plus reader and device processing. The card-side processing per exchange is bounded by the frame waiting time

$$FWT = \frac{256 \cdot 16}{f_c} \cdot 2^{FWI} \quad (5)$$

where, FWT is the frame waiting time, f_c is the carrier frequency, and FWI is the frame waiting time integer announced by the card. EMV limits FWI to 7 for cards, which gives $FWT \approx 38.7$ ms per exchange [46, Annex A.4]. Table III reports the resulting lower bounds for each strategy. These expressions only cover communication. For computation, optimized ML-DSA code has been benchmarked on ARMv8-A application processors outside a TEE [43] and on Android smartphones through liboqs and Bouncy Castle at security levels 3 and 5 [44]. ML-DSA-65 signing on StrongBox hardware has been reported at 1229 to 1826 ms for a 2 KB message on a Samsung Galaxy S23 FE [39]. This study does not report on-device measurements; other evaluations focus on ARM Cortex-M microcontrollers [9], [10], [42]. We leave the computation part to future work (Section VII).

V. THREAT ANALYSIS AND EVALUATION OF MITIGATION STRATEGIES

A. Threat Model

In light of the constraint model, which identifies offline authentication as the protocol stage where standardized PQC signatures meet their main deployment obstacle, this section defines the threat model in five steps: assets, trust boundaries, adversaries, adversary capabilities, and attack goals with their consequences. The protected assets are 1) the certification-authority and issuer RSA private keys that sign the certificate chain and the static application data, 2) the per-card ICC private key used for DDA and CDA, 3) the symmetric issuer master keys and derived session keys used for application cryptograms, and 4) the confidentiality of provisioning and transaction data in transit. The trust boundaries lie between the payment application and the hardware-backed keystore on the device, on the contactless interface between the device and the terminal, on the network linking the terminal, acquirer, payment network, and issuer, and on the provisioning channel between the tokenization service and the device. Three adversaries are considered:

- A passive network adversary, who records provisioning and transaction traffic for later cryptanalysis but has no access to device or terminal internals.
- A malicious or compromised terminal, which controls the contactless interface and can relay, replay, or alter APDUs within a transaction.
- A compromised issuer or tokenization infrastructure, with insider access to key-management or provisioning systems.

A cryptographically relevant quantum computer (CRQC) is modeled not as a separate actor but as a capability that any of these adversaries may acquire. The certification-authority and issuer public keys are distributed to every terminal and card, so an adversary with a CRQC does not need recorded transactions to attack offline authentication: running Shor's algorithm on a public key gives the corresponding private key, which can then sign counterfeit data. Forgery therefore requires a CRQC while the attacked key is still trusted, not at the time a transaction was recorded. Recorded traffic matters only for confidentiality, through harvest-now, decrypt-later attacks on classically

protected provisioning and network channels, including the acquirer–issuer channel (State 6).

The attack goals and their consequences follow from these assets. Recovering a certification-authority or issuer private key would let an adversary produce valid certificates and signed data for counterfeit cards; EMVCo notes that breaking a payment-system or issuer key would allow the production of fake cards used for fraudulent offline payments [26]. Such counterfeits would pass offline authentication at terminals that approve transactions offline, whereas online-authorized transactions stay protected by the issuer’s verification of the symmetric application cryptogram (ARQC/ARPC). Symmetric cryptography is affected only by Grover’s algorithm, and EMVCo does not consider AES-128 threatened by it in practice [26], which makes application cryptograms a lower migration priority. Recovering session keys or provisioning secrets from recorded traffic would compromise confidentiality and could expose tokenized credentials. Finally, the transport-layer channels between payment infrastructure components still rely on classical public-key cryptography and are further candidates for migration, although this transition has already started in production Internet infrastructure through hybrid key-exchange mechanisms [7], [52], [34].

The timing of these threats can be framed with Mosca’s inequality [53]:

$$x + y > z \quad (6)$$

where, x is the time for which protection must hold, y is the time needed to migrate, and z is the time until a CRQC exists. When the inequality holds, migration is already overdue. For offline authentication, x is the remaining period during which the attacked keys are trusted by terminals; EMVCo expects 1984-bit RSA certification-authority keys to stay in use until at least 31 December 2036 [37]. For y , EMVCo states that its warning indicators are chosen to give at least ten years of notice, because ten years is the minimum time EMV terminal infrastructure needs to deploy mitigations [27]. For z , EMVCo now writes that a CRQC may be decades away, while noting that the German BSI considers 2040 a reasonable estimate [27]. If the BSI estimate holds, a ten-year terminal migration would have to start around 2030 to finish in time. The protocol constraints identified in this study make that migration harder, since PQC certificates cannot be introduced until terminals, kernels, and card applications can carry the larger objects of Section IV. Confidentiality is a different case: there, x is the

lifetime of data recorded today. EMVCo notes that harvest-now, decrypt-later does not apply to chip authentication itself [27], but it does apply to classically protected provisioning and network channels.

B. Mitigation Strategies

Three mitigation strategies are evaluated against the constraint model. The first extends the APDU response length so that larger post-quantum signatures fit in a single command–response pair. This removes the immediate size limit, but support for extended-length APDUs across deployed contactless terminals is not established in public documentation, and the full signature still has to be transmitted (Section IV-D). The second splits a large signature across several APDU exchanges, for example with the response-chaining mechanism of ISO/IEC 7816-4, in which status word ‘61XX’ indicates that more data can be fetched with GET RESPONSE [22]. This keeps the short-form message structure but adds command–response turnarounds and depends on kernel acceptance of chained responses, which the sources we reviewed do not document; Book A even states that reference cards used for reader testing should not need chaining in either direction [45, Sect. 10.4]. The third moves signature generation from the mobile device to backend infrastructure. On its own, this does not remove the constraint: if the terminal still verifies a post-quantum signature offline, that signature must still reach the terminal through the same APDU interface, as shown in Section IV-C. The size constraint is avoided only if verification also moves online, so that the terminal relies on the issuer’s real-time validation of the symmetric application cryptogram, or if the authentication construction itself changes. Both options alter availability and trust assumptions. The first removes offline acceptance, which EMVCo describes as the ultimate protection against compromised asymmetric keys [26]; the second replaces proof of locally held private-key possession with a different mechanism. Dropping offline authentication for tokenized flows is the online-verification variant of this third strategy. Other options offer limited relief. Hybrid RSA and post-quantum signatures add to, rather than reduce, the transmitted size. The smaller-signature schemes of Section IV-A could fit the signature field, but they are not yet standardized and, for UOV, would not fit certificates. Table III summarizes these trade-offs [15], [19]. Its entries are derived from the analysis in Sections IV-B to IV-D and from the cited specifications, and compatibility is marked as not established wherever public documentation does not confirm support.

TABLE III. COMPARATIVE EVALUATION OF MITIGATION STRATEGIES

Strategy	Size constraint	Timing cost (lower bound at 106 kbit/s; ML-DSA-44 / ML-DSA-65 / SLH-DSA-128s)	Terminal compatibility	Trust-model change
Extended-length APDU	Resolved (single exchange)	184 / 251 / 595 ms (single transfer; RSA: 19 ms)	Not established: requires extended-length support in terminal and kernel	None
Multi-round APDU chaining	Resolved (split)	189 / 258 / 610 ms ($n = 10 / 13 / 31$ exchanges)	Not established: ISO/IEC 7816-4 chaining; Book A reference cards should not need chaining [45]	None
Backend signing with online verification	Avoided only if verification moves online	No added NFC transfer; online authorization round trip (not quantified)	Requires online-capable terminal; no change to APDU framing	Significant (loss of offline acceptance)

VI. DISCUSSION

The constraint model developed in this study applies to deployments that perform offline authentication. In non-tokenized implementations, where RSA signing happens during the contactless transaction, the APDU size mismatch directly affects tap-time feasibility. In tokenized deployments, the constraint would apply only if a signature still had to be delivered to an offline-verifying terminal; if such deployments rely only on online cryptogram verification, offline authentication is not exercised, and the constraint does not arise. In no configuration that keeps offline verification does direct substitution of standardized PQC signatures avoid the size mismatch. The existing literature on PQC migration for payment systems does not distinguish between these configurations.

This study examined whether standardized post-quantum digital signatures can directly replace RSA in offline authentication. The protocol-level analysis indicates that the main obstacle is not the computational cost of post-quantum algorithms but the message-size limits of the EMV protocol. RSA signatures and certificates were designed to fit the response and record structure; the smallest standardized ML-DSA signature exceeds the response allocation by a factor of 9.5, and its certificates exceed the record limit by a factor of almost 15. Direct substitution is therefore not possible without modifying the protocol. The analysis also shows that the protocol constraints have to be considered together rather than one at a time. The study has several limitations. First, the protocol model and the eight-stage workflow are reconstructed from specifications and documentation and have not been validated with APDU traces, a HostApuService prototype, or terminal tests. Second, the analysis uses EMV Book 2 version 4.1; current kernel specifications reference version 4.4 [21], and kernel-specific data placement was not modeled. Third, the timing analysis bounds the communication part analytically and includes no on-device measurements. Fourth, the terminal-compatibility entries in Table III have not been verified against deployed terminals. Finally, EMVCo reports that offline transactions make up less than one percent of transactions in most countries [27]. The constraint studied here concerns this small share, but it is exactly the part of EMV whose security rests on public-key signatures.

VII. CONCLUSION

This study characterized the protocol constraints on post-quantum migration of EMV offline authentication in Android HCE systems, for deployments that deliver an offline-verifiable signature to the terminal. Instead of evaluating cryptographic performance alone, it reconstructed the protocol workflow and examined the constraints imposed by message size, transaction timing, hardware support, and execution architecture. Direct substitution of even the smallest ML-DSA parameter set is prevented by the short-form protocol limits: the encoded CDA response grows from at most 256 bytes to at least 2438 bytes, each certificate grows from at most 248 bytes to at least 3732 bytes, and the transfer alone takes almost half of the 400 ms that EMV allocates to the card. Deployment architecture, in turn, decides whether offline authentication, and with it this constraint, is exercised in practice. Post-quantum migration of

Android HCE payment systems cannot be evaluated through cryptographic algorithms alone.

Several research directions follow from this work. First, a HostApuService prototype exchanging ML-DSA-sized responses with Android-based and commercial readers should be used to collect APDU traces, check support for extended-length APDUs and response chaining, and measure transfer time against the bounds derived here. Second, standardized PQC signatures should be implemented in Android HCE applications to measure execution latency, memory use, and energy consumption on current smartphones, including the ML-DSA support announced for Android Keystore [40]. Third, protocol modifications that keep interoperability while carrying larger post-quantum objects should be investigated, including authentication mechanisms designed for post-quantum environments and smaller-signature schemes once they are standardized. Finally, confirming whether production tokenized Android HCE wallets perform offline authentication at all, for instance through controlled reverse engineering or vendor disclosure, would settle the architectural question left open in this study.

FUNDING

The research was supported by the PhD-Associated Scholarship-PASS, funded by Centre National pour la recherche Scientifique et Technique (CNRST), Morocco.

REFERENCES

- [1] EMVCo, LLC, "EMV Integrated Circuit Card Specifications for Payment Systems, Book 2: Security and Key Management," Version 4.1, EMVCo, LLC, May 2004.
- [2] D. Dam, T. Tran, V. Hoang, C. Pham, and T. Nguyen, "A Survey of Post-Quantum Cryptography: Start of a New Race," *Cryptography*, vol. 7, no. 3, p. 40, Aug. 2023, doi: 10.3390/cryptography7030040.
- [3] G. Alagic, D. Apon, D. Cooper, Q. Dang, T. Dang, J. Kelsey, J. Lichtinger, Y. K. Liu, C. Miller, D. Moody, et al., "Status Report on the Third Round of the NIST Post-Quantum Cryptography Standardization Process," NIST IR 8413-upd1, National Institute of Standards and Technology, 2022.
- [4] National Institute of Standards and Technology, "Module-Lattice-Based Key-Encapsulation Mechanism Standard," FIPS Publication 203, U.S. Department of Commerce, 2024.
- [5] National Institute of Standards and Technology, "Module-Lattice-Based Digital Signature Standard," FIPS Publication 204, U.S. Department of Commerce, 2024.
- [6] National Institute of Standards and Technology, "Stateless Hash-Based Digital Signature Standard," FIPS Publication 205, U.S. Department of Commerce, 2024.
- [7] Cloudflare, Inc., "Post-Quantum Cryptography (PQC)," 2026 (accessed Oct. 2026). [Online]. Available: <https://developers.cloudflare.com/ssl/post-quantum-cryptography/>
- [8] S. Hoque, A. Aydeger, J. Baranda, E. Zeydan, and F. Wilhelmi, "Post-Quantum Cryptography Integration into the Future Mobile Core: A Service-Based Architecture Perspective," *IEEE Network*, 2026. [Online]. Available: <https://ieeexplore.ieee.org/document/11454697>
- [9] M. J. Kannwischer, J. Rijneveld, P. Schwabe, and K. Stoffelen, "pqm4: Testing and Benchmarking NIST PQC on ARM Cortex-M4," *Cryptology ePrint Archive*, Jul. 2019. [Online]. Available: <https://eprint.iacr.org/2019/844>
- [10] R. Chhetri, "Benchmarking NIST-Standardised ML-KEM and ML-DSA on ARM Cortex-M0+: Performance, Memory, and Energy on the RP2040," arXiv preprint arXiv:2603.19340, 2026.

- [11] D. Kim, J. Choi, S. Yoon, and S. C. Seo, "Optimized Implementation of HQC on Cortex-M4," *ICT Express*, vol. 11, no. 5, pp. 939-944, Sep. 2025, doi: 10.1016/j.ict.2025.07.001.
- [12] S. Hoque, A. Aydeger, and E. Zeydan, "Exploring Post-Quantum Cryptography with Quantum Key Distribution for Sustainable Mobile Network Architecture Design," in *Proc. 4th Workshop on Performance and Energy Efficiency in Concurrent and Distributed Systems*, Jan. 2024, doi: 10.1145/3659997.3660033.
- [13] A. Scrivano, "A Comparative Study of Classical and Post-Quantum Cryptographic Algorithms in the Era of Quantum Computing," *arXiv preprint arXiv:2508.00832*, Aug. 2025. [Online]. Available: <https://arxiv.org/abs/2508.00832>
- [14] R. Zhou, H. Guo, and F. E. C. Teo, "Investigating Post-Quantum Cryptography to Secure Transmitted Data via Mobile Communication," *Electronics*, vol. 15, no. 6, art. 1275, Mar. 2026, doi: 10.3390/electronics15061275.
- [15] Bank for International Settlements, "Project Leap Phase 2: Quantum-Proofing Payment Systems," Technical Report, BIS Innovation Hub, 2025.
- [16] Md. A. Mansur, "A Quantum-Safe, Interoperable, and Decentralized Payment Infrastructure for the Post-Classical Era as a Strategic Framework for Secure Global Transactions," *European Scientific Journal*, ESJ, vol. 21, no. 19, p. 17, Jul. 2025, doi: 10.19044/esj.2025.v21n19p17.
- [17] D. Nijimbere and V. Mbonigaba, "Securing Mobile Payments in IoT Networks: Post-Quantum Challenges and Solutions," *Journal of Computer and Communications*, vol. 14, no. 1, pp. 33-45, Jan. 2026, doi: 10.4236/jcc.2026.141003.
- [18] Nacha Payments Innovation Alliance, "Protecting Payments in the Quantum Era: Setting a Course for Action," Nacha, 2025. [Online]. Available: <https://www.nacha.org/system/files/2025-09/725-Alliance-Quantum-paper.pdf>
- [19] C. Bush, "Quantum Computing and Digital Payments: Opportunities, Threats, and the Path Toward Regulation," *International Review of Law, Computers & Technology*, 2026, doi: 10.1080/13600869.2026.2623674.
- [20] Y. Kethepalli, R. Joseph, S. R. Vajralla, J. Vemula, and N. S. Naik, "Reinforcing Security and Usability of Crypto-Wallet with Post-Quantum Cryptography and Zero-Knowledge Proof," *arXiv preprint arXiv:2308.07309*, Aug. 2023, doi: 10.48550/arXiv.2308.07309.
- [21] EMVCo, LLC, "Specification Bulletin No. 324: Updates to EMV Contactless Book C-4 – Kernel 4 Specification," First Edition, Jul. 2026.
- [22] ISO/IEC, "Identification Cards — Integrated Circuit Cards — Part 4: Organization, Security and Commands for Interchange," ISO/IEC 7816-4:2020, 2020.
- [23] Google LLC, "Security — Device Tokenization Developer Site," 2026 (accessed Oct. 2026). [Online]. Available: <https://developers.google.com/pay/issuers/overview/security>
- [24] Samsung Developers, "Security," 2026 (accessed Oct. 2026). [Online]. Available: <https://developer.samsung.com/pay/security>
- [25] S. Pandey, M. Crowe, and B. Russell, "Understanding the Role of Host Card Emulation in Mobile Wallets," *Payment Strategies*, Federal Reserve Bank of Boston, May 2016.
- [26] EMVCo, LLC, "EMVCo Position Statement: Quantum Computing and EMV Chip Cryptography," EMV-SWG-NM07r17c-public, Mar. 2024 (published Sep. 2024).
- [27] EMVCo, LLC, "EMVCo SWG Position Paper: Update on Quantum Computing and EMV Chip Cryptography," EMV-SWG-NP36r11, Aug. 2026.
- [28] Secure Technology Alliance, "Host Card Emulation (HCE) 101," Technical Report, Secure Technology Alliance, 2014.
- [29] Android Open Source Project, "Host-Based Card Emulation Overview," 2026 (accessed Oct. 2026). [Online]. Available: <https://developer.android.com/develop/connectivity/nfc/hce>
- [30] Android Open Source Project, "Android Keystore System," 2026 (accessed Oct. 2026). [Online]. Available: <https://developer.android.com/privacy-and-security/keystore>
- [31] U.S. Payments Forum, "EMV Payment Tokenization Primer: Lessons Learned," Technical Report, U.S. Payments Forum, Jun. 2019.
- [32] K. Fan, W. Jiang, H. Li, and Y. Yang, "NFC Secure Payment and Verification Scheme with CS E-Ticket," *Security and Communication Networks*, vol. 2017, art. 4796373, 2017, doi: 10.1155/2017/4796373.
- [33] ISO/IEC, "Identification Cards — Contactless Integrated Circuit Cards — Proximity Cards — Part 4: Transmission Protocol," ISO/IEC 14443-4:2018, 2018.
- [34] T. Reddy.K, "Post-Quantum Cryptography Recommendations for TLS-Based Applications," IETF Internet-Draft draft-reddy-uta-pqc-app-08, Jan. 2025.
- [35] EMVCo, LLC, "EMV Chip Specifications to Support Elliptic Curve Cryptography," Oct. 2021. [Online]. Available: <https://www.emvco.com/knowledge-hub/emv-chip-specifications-tosupport-elliptic-curve-cryptography/>
- [36] EMVCo, LLC, "EMV Contactless Specifications for Payment Systems, Book E: Security and Key Management," Version 1.1, Feb. 2025.
- [37] EMVCo, LLC, "EMVCo Annual Public Key Lengths Assessment," Notice Bulletin No. 30, EMV SWG NQ47r2, Jul. 2026.
- [38] E. Barker and A. Roginsky, "Transitioning the Use of Cryptographic Algorithms and Key Lengths," NIST Special Publication 800-131A Rev. 2, Mar. 2019, doi: 10.6028/NIST.SP.800-131Ar2.
- [39] S. Lee, H. Jung, N. Heo, and T. T. Kwon, "Case Study: PQC Migration for Android Strongbox Keymaster on Samsung Galaxy Mobile Devices," *IEEE Access*, vol. 14, pp. 25770-25782, 2026, doi: 10.1109/ACCESS.2026.3660949.
- [40] E. Lynch and D. Elliot, "Security for the Quantum Era: Implementing Post-Quantum Cryptography in Android," *Google Security Blog*, Mar. 2026 (accessed Oct. 2026). [Online]. Available: <https://security.googleblog.com/2026/03/post-quantum-cryptography-in-android.html>
- [41] D. Basin, R. Sasse, and J. Toro-Pozo, "SoK: Security of EMV Contactless Payment Systems," *arXiv preprint arXiv:2504.12812*, 2025.
- [42] J. Huang, A. Adomnici, J. Zhang, W. Dai, Y. Liu, R. C. C. Cheung, Ç. K. Koc, and D. Chen, "Revisiting Keccak and Dilithium Implementations on ARMv7-M," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2024, no. 2, pp. 1-24, Mar. 2024, doi: 10.46586/tches.v2024.i2.1-24.
- [43] H. Becker, V. Hwang, M. J. Kannwischer, B.-Y. Yang, and S.-Y. Yang, "Neon NTT: Faster Dilithium, Kyber, and Saber on Cortex-A72 and Apple M1," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2022, no. 1, pp. 221-244, 2021, doi: 10.46586/tches.v2022.i1.221-244.
- [44] O. Bajus, "Analysis of Transition to Post-Quantum Cryptography on Mobile Platforms," Master's thesis, Faculty of Informatics, Masaryk University, Brno, 2026.
- [45] EMVCo, LLC, "EMV Contactless Specifications for Payment Systems, Book A: Architecture and General Requirements," Version 2.11, Jun. 2023.
- [46] EMVCo, LLC, "EMV Level 1 Specifications for Payment Systems: EMV Contactless Interface Specification," Version 4.0, Jan. 2026.
- [47] M.-J. O. Saarinen, "Accelerating SLH-DSA by Two Orders of Magnitude with a Single Hash Unit," in *Advances in Cryptology - CRYPTO 2024*, pp. 276-304, 2024, doi: 10.1007/978-3-031-68376-3_9.
- [48] P.-A. Fouque, J. Hoffstein, P. Kirchner, V. Lyubashevsky, T. Pornin, T. Prest, T. Ricosset, G. Seiler, W. Whyte, and Z. Zhang, "Falcon: Fast-Fourier Lattice-Based Compact Signatures over NTRU, Specification v1.2," NIST PQC Round 3 submission, Oct. 2020.
- [49] G. Alagic et al., "Status Report on the Second Round of the Additional Digital Signature Schemes for the NIST Post-Quantum Cryptography Standardization Process," NIST IR 8610, National Institute of Standards and Technology, May 2026, doi: 10.6028/NIST.IR.8610.
- [50] Android Open Source Project, "IsoDep," Android API Reference, 2026 (accessed Oct. 2026). [Online]. Available: <https://developer.android.com/reference/android/nfc/tech/IsoDep>

- [51] ISO/IEC, "Cards and Security Devices for Personal Identification — Contactless Proximity Objects — Part 2: Radio Frequency Power and Signal Interface," ISO/IEC 14443-2:2020, 2020.
- [52] National Security Agency, "Commercial National Security Algorithm Suite 2.0 (CNSA 2.0)," NSA Technical Report, 2022.
- [53] M. Mosca, "Cybersecurity in an Era with Quantum Computers: Will We Be Ready?," IEEE Security & Privacy, vol. 16, no. 5, pp. 38-41, 2018, doi: 10.1109/MSP.2018.3761723.