

Multi-Scenario Security Evaluation of an RSA-PSS Anti-Replay QR Code System with a PRISMA 2020 Review of Indonesian Prior Work

Andika Agus Slameto ^{1*}, Yudi Sutanto ^{2*}, Hendra Kurniawan ^{3*}

^{*} Faculty of Computer Science, Universitas Amikom Yogyakarta

Jl. Ring Road Utara, Condongcatur, Depok, Sleman, Yogyakarta 55281, Indonesia

rmkt.andika@amikom.ac.id ¹, yudisuta@amikom.ac.id ², hendrakurniawan@amikom.ac.id ³

Article Info

Article history:

Received 2026-09-04

Revised 2026-09-25

Accepted 2026-10-06

Keyword:

Anti-replay,
Data tampering,
QR Code security,
RSA-PSS,
Prisma 2020.

ABSTRACT

QR Code tokens used for identity and document verification are exposed to replay, payload tampering, and signature forgery, yet published evaluations rarely separate what a digital signature guarantees from what application-level checks add. This study evaluates a Flask verification service that combines RSASSA-PSS (2048-bit modulus, 8-byte salt) with nonce-timestamp replay control. A threat model defines thirteen executable attack subtypes: seven tampering operations, replay, expired-token reuse, and four forgery operations. An empirical harness submitted 66,500 operations to the production verification path, including 5,549 validly signed negative controls, and a layer ablation repeated the identical campaign with signature-only and application-only verifiers. All attacks were rejected and no control was refused; one-sided 95% Clopper-Pearson lower bounds were 99.994% for tampering ($n = 50,000$), 99.925% for forgery ($n = 4,000$), and 99.800% for replay ($n = 1,500$ payloads). Signature verification alone rejected every tampering and forgery operation but accepted all replayed and expired tokens, whereas application checks alone caught replay and expiry but accepted signature substitution. Earlier simulated figures, including 78.14% tampering detection, were traced to hard-coded probabilities in the test harness and are not treated as system measurements. HTTP load testing from an external host at 100 to 1,500 concurrent users (25,607 requests) showed a throughput plateau near 105 requests per second with error rates of at most 0.09%. A PRISMA 2020 review of 26 Indonesian studies found none combining these components, a finding bounded by its search scope. The detection rates are empirical conformance evidence, not a proof of EUF-CMA security.



This is an open access article under the [CC-BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.

I. INTRODUCTION

QR Codes are now a routine carrier for identity credentials, access passes, certificates, and document verification links. The same convenience makes them attractive to attackers. In a study of 173 participants, Sharevski et al. found that 67% were willing to sign in with social-media credentials after scanning a malicious QR Code and only 14.5% declined [1], and Wahsheh and Al-Zahrani noted that QR Code generation has no standard authentication phase, leaving scanners exposed to phishing and malware links [2]. Yalda et al. cite a 433% rise in QR Code scanning between 2021 and 2023 alongside rapid growth in phishing [3]. A verification service that issues QR tokens therefore has to withstand three distinct

threats: replay of a genuine token outside its intended use [4], tampering with the encoded data, and forgery of the signature that is meant to bind that data to its issuer.

Prior work addresses these threats unevenly. International studies protect QR Codes through dynamic one-order-one-code payment tokens [5], physical anti-counterfeiting features [6]–[8], blockchain-backed verification [3], [9], or machine-learning classification of malicious links [10]; other work concentrates on the computational cost of RSA signing [11], [12]. Indonesian studies mostly embed an RSA, ECDSA, AES, or hash-based signature in a QR Code and validate it with a handful of functional test cases [13]–[22]; one scheme relies on MD5 [21], whose collision resistance is broken [23], and only one uses RSASSA-PSS, as an auxiliary layer above

a DSA signature [24]. Two gaps follow. First, adversarial evaluations in this literature seldom state a threat model or report negative controls, so a high detection rate cannot be distinguished from a verifier that rejects everything. Second, few studies distinguish the guarantee of the signature from the checks an application performs around it, even though a validly signed token can still be replayed or used after it has expired.

The evaluated system signs a canonical JSON payload with RSASSA-PSS as specified in PKCS #1 v2.2 [25] and FIPS 186-5 [26], and pairs the signature with a nonce ledger, a timestamp validity window, and structural checks. This study asks three questions. RQ1: within an explicit threat model, how reliably does the deployed verification path reject each defined attack subtype while accepting valid tokens, and with what statistical confidence? RQ2: which verification layer is responsible for each rejection, and what does the signature alone fail to cover? RQ3: how does the system compare with prior QR Code authentication studies, and what can a systematic review of Indonesian work establish about its novelty within the scope searched?

The paper makes four contributions. (i) It defines a threat model and thirteen executable attack subtypes mapped to the CAPEC catalogue [27]. (ii) It reports an empirical campaign of 66,500 operations with negative controls and exact binomial bounds, together with a layer ablation that attributes every rejection to the signature or to application-level checks. (iii) It corrects an earlier simulation-based account by showing from the harness source code that the simulated detection rates, including 78.14% for tampering, were generated from fixed constants. (iv) It reports a PRISMA

2020 review of 26 Indonesian studies with its search limits stated, and measures scalability over HTTP at four load levels. The system design and its performance efficiency are described in a companion paper; Section II.A summarises what this evaluation needs.

II. METHOD

A. System Overview and Threat Model

The prototype is a Flask service on Ubuntu with Python 3.12. An authenticated issuer creates a payload containing a name, identifier, ISO 8601 timestamp, 8-byte random nonce, and QR layout metadata; serialises it as JSON with sorted keys; and signs its SHA-256 digest with RSASSA-PSS using a 2048-bit key, MGF1 with SHA-256, and an 8-byte salt. RSA-2048 provides a 112-bit security strength [28]. Coron showed that a salt of roughly $\log_2 q_{\text{sig}}$ bits, where q_{sig} is the number of signatures an adversary may obtain, preserves a tight security reduction [29]; 64 salt bits exceed that requirement for any realistic issuance volume. The salt length does not change the 256-byte signature size. The issued record is stored server-side, and the QR Code carries either a short verification token or an encoded copy of the signed payload. Fig. 1 shows the verification pipeline. A token is valid only if it passes four layers: L1, RSASSA-PSS verification with the public key; L2, record binding, which compares the presented payload with the stored original; L3, structural validation of required fields, field types, and unknown fields; and L4, state and time checks consisting of a nonce ledger for replay, a 300-second clock-drift tolerance, per-nonce monotonic timestamps, and a seven-day validity window.

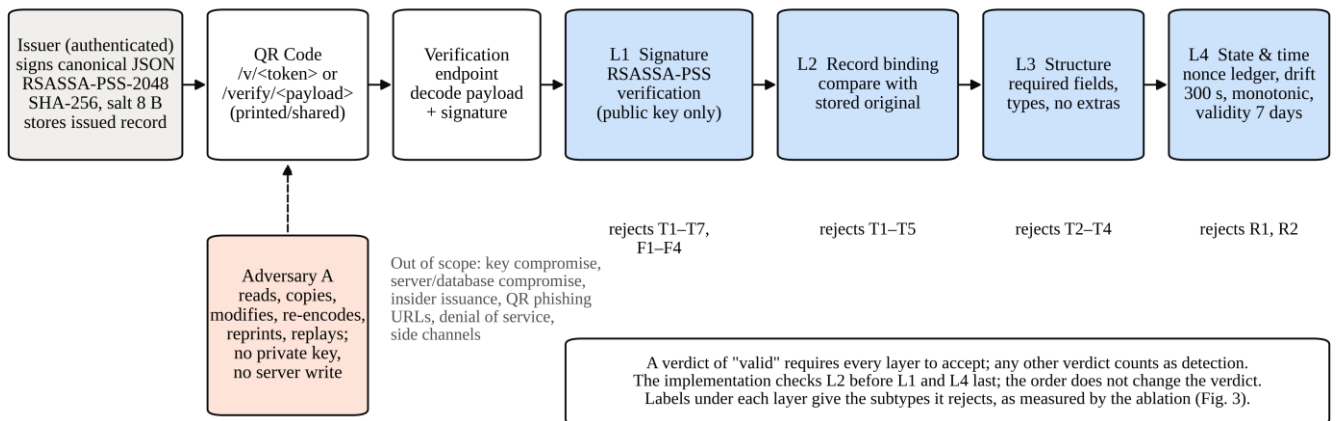


Fig. 1. Verification pipeline and threat surface. The adversary controls QR content presented to the verification endpoint; layers L1 to L4 must all accept for a verdict of valid.

Threat modelling follows the practice of stating assets, adversary capabilities, and goals explicitly [30]. The adversary is a network and physical attacker in the spirit of Dolev and Yao [31], restricted to the QR channel: it can read, copy, modify, re-encode, reprint, and re-present any QR Code, and it can collect any number of validly signed tokens

issued to others. It cannot obtain the private key, write to the server-side record store or nonce ledger, or issue tokens through the authenticated issuer. Table I lists the security goals and the attacks placed outside scope.

TABLE I
SECURITY GOALS AND SCOPE OF THE THREAT MODEL

Item	Definition
G1 Integrity	Any change to a signed field, or to the signature, leads to rejection.
G2 Authenticity	Only signatures produced with the issuer key are accepted.
G3 Freshness	A token is accepted at most once; later presentations are replays.
G4 Validity	Tokens older than seven days, or dated beyond the drift tolerance, are rejected.
G5 Conformance	Payloads with missing, mistyped, or unknown fields are rejected.
G6 Availability of valid use	Unmodified, unexpired, first-use tokens are accepted.
Out of scope	Private-key or server compromise, malicious insiders, phishing URLs in foreign QR Codes, denial of service, side channels, and scanner or print defects.

Within this model, thirteen attack subtypes were defined and implemented as executable operations on real payloads (Table II). Each is mapped to the closest entry in CAPEC version 3.9 [27], and each row states the goal it violates and the layer expected to reject it. Two subtypes are operationally identical and are kept for continuity with the scenario structure: T6 and F3 both substitute a valid signature taken from another payload. T7 was labelled encryption bypass in earlier project documents; it does not involve encryption and is renamed here to describe what it does.

TABLE II
DEFINITION OF THE THIRTEEN ATTACK SUBTYPES

ID	Operation applied to a validly issued payload	CAPEC entry	Goal violated	Expected layer	Operations (n)
T1	Replace the name field with a different name	153, 39	G1	L1, L2	20,030
T2	Add an unsigned field level_akses = administrator	477	G1, G5	L1, L2, L3	10,102
T3	Delete the identifier field	153	G1, G5	L1, L2, L3	7,467
T4	Change qr_version from integer to string	153	G1, G5	L1, L2, L3	4,993
T5	Move the timestamp forward by 1 to 400 days	153, 39	G1, G4	L1, L2	3,943
T6	Keep the data, attach the valid signature of another payload	473	G2	L1	2,491
T7	Keep data and signature, declare an unsupported algorithm (none)	475	G2	L1	974
R1	Present the same valid token a second and third time	60	G3	L4	3,000
R2	Present a valid token issued 8 to 400 days earlier	60, 21	G4	L4	2,451
F1	Replace the signature with 256 random bytes	473	G2	L1	1,016
F2	Truncate the signature to its first 128 bytes	475	G2	L1	963
F3	Replace the signature with the valid signature of another payload	473	G2	L1	1,032
F4	Flip one random bit of the valid signature	473	G2	L1	989

B. Evaluation Instruments

Three instruments were used, and their outputs are kept apart throughout the paper (Table III). The first, run in July 2026, was a calibrated discrete-event simulation of 350,000 operations over eight sessions: 100,000 normal operations, 110,000 replay attempts, 50,000 tampering attempts, 50,000 forgery attempts, and 40,000 load operations. Its operation timings were drawn from distributions calibrated on the prototype [32], but, as Section III.A shows, its detection outcomes were not produced by the verification code. It is reported only to explain figures that appeared in earlier drafts. The second instrument is the empirical campaign on which all security claims rest. Its harness signs every payload with the

production key, applies the operation defined in Table II, and passes the result to the same signature check and classification routine that the deployed service uses. The security state database is redirected to a temporary file so that the production nonce ledger is never touched. The third instrument is a layer ablation that re-runs the empirical campaign three times with the same seed and the same allocation of subtypes, replacing the verifier with (a) the full pipeline, (b) L1 alone, and (c) L2 to L4 with the signature treated as valid; a separate pass records L3 alone for the tampering subtypes. The verification functions used by the ablation are byte-identical to those in the release that produced the campaign, and the full-pipeline pass reproduced the archived subtype counts exactly.

TABLE III
EVALUATION INSTRUMENTS AND THEIR ROLE

Aspect	Simulation	Empirical campaign	Layer ablation
Date	23 Jul 2026	4 Sep 2026	13 Sep 2026
Detection decision	Random draw against fixed rates	Production verifier	Verifier variants
Payload modified	No	Yes	Yes
Negative controls	None	5,549	5,549 per variant
Operations	350,000	66,500	199,500 + 50,000
Seeded	No	Yes (20260824)	Yes (20260824)
Role in this paper	Explains superseded figures	Security claims	Layer attribution

C. Scenarios, Negative Controls, and Metrics

The empirical campaign comprised four scenarios. Tampering applied T1 to T7 to 50,000 fresh payloads, with the subtype drawn at random using weights 0.40, 0.20, 0.15, 0.10, 0.08, 0.05, and 0.02. Replay issued 1,500 payloads and verified each three times, giving 1,500 first uses that must be accepted and 3,000 replays that must be rejected. The validity scenario issued 5,000 payloads whose ages were assigned at random, producing 2,451 expired tokens (R2) and 2,549 unexpired controls. Forgery applied F1 to F4 to 4,000 payloads. Negative controls, which are validly signed and unmodified tokens that must be accepted, numbered 2,500 in the tampering scenario, 500 in the forgery scenario, and 2,549 in the validity scenario, for 5,549 in total. They were drawn from a separate random generator so that adding them did not shift the attack allocation. An operation counts as detected when the verifier returns any verdict other than valid. The detection rate and the false positive rate are defined in (1) and (2), where TP is the number of detected attacks, FN the number of attacks accepted as valid, FP the number of rejected controls, and TN the number of accepted controls.

$$DR = \frac{TP}{TP + FN} \times 100\% \quad (1)$$

$$FPR = \frac{FP}{FP + TN} \times 100\% \quad (2)$$

D. Statistical Analysis

Every observed proportion is reported with an exact Clopper-Pearson bound [33], which remains valid when all trials succeed, a case where Wald intervals collapse to zero width [34]. For x detections in n attacks, the one-sided lower confidence bound at level $1 - \alpha$ is given by (3), where B^{-1} is the quantile function of the beta distribution. When $x = n$, (3) reduces to $\alpha^{1/n}$, which is numerically close to the rule of three, $1 - 3/n$ [35]. The same construction gives upper bounds for false positive rates when no control is rejected.

$$p_L = B^{-1}(\alpha; x, n - x + 1) \quad (3)$$

Three assumptions qualify these bounds. First, each operation is treated as an independent trial. This holds for tampering and forgery, where every operation uses a fresh payload, but not for the 3,000 replay attempts, which come in pairs from 1,500 payloads; the replay bound is therefore computed at the payload level with $n = 1,500$. Second, the bounds refer to the attack distribution defined by Table II and say nothing about attack classes outside it. Third, when 13 subtype-level bounds are read together, a Bonferroni-adjusted level of $\alpha = 0.05/13$ is also reported. Detection here is largely deterministic: every field in the payload enters the signed digest, so a single modification is expected to fail L1 with certainty. The sample sizes serve to rule out implementation faults that would appear only for some inputs, such as encoding errors, type-dependent canonicalisation, or a fail-open exception path, and to bound their frequency.

These statistics do not measure cryptographic strength. EUF-CMA, existential unforgeability under adaptive chosen-message attack [36], is a property of the signature scheme proved in the random-oracle model under the RSA assumption [29], [37]. It bounds the success probability of any efficient forger by a negligible quantity, far below anything a finite experiment could resolve. The forgery subtypes F1 to F4 are non-adaptive and structural; rejecting them shows that the implementation verifies correctly and does not fail open, which is a necessary condition for inheriting the proof, not evidence of the proof itself.

E. Load Testing

Scalability was measured over HTTP rather than simulated, because in-process simulation measures the simulator [38]. The k6 load generator ran on a laptop outside the server, since three earlier attempts with the generator on the same two-core host measured contention between generator and application rather than service capacity. The target was a staging instance configured as production, with one worker process and twelve threads behind a TLS reverse proxy, on 2 vCPU and 9.7 GB of RAM. Each of four levels, 100, 500, 1,000, and 1,500 concurrent virtual users, ran for 60 seconds against a pool of 120,000 single-use tokens, so that every request traversed the full verification path of an unused token. Error rates are reported with two-sided 95% Clopper-Pearson intervals, and mean response time is checked against Little's law in (4) [39], [40], where N is the number of concurrent users, X the throughput, R the mean response time, and Z the think time between requests. The load generator issued requests without think time, so $Z = 0$ and $R = N/X$.

$$N = X(R + Z) \quad (4)$$

F. Systematic Review Protocol

The literature review followed the PRISMA 2020 reporting guideline [41] and, for the search, the items of PRISMA-S [42]; the procedure follows the systematic review practice of software engineering [43]. Table IV summarises the protocol. PRISMA governs how a review is reported; compliance

makes the search auditable but does not by itself establish novelty. The review question was deliberately narrow: whether any Indonesian study of QR Code-based signature or document authentication reports all four components C1 to C4, which together define a direct predecessor (C5).

TABLE IV
PROTOCOL OF THE SYSTEMATIC REVIEW

Element	Specification
Question	Does any Indonesian QR Code authentication study combine C1 to C4?
Sources	Track A: 10 expert-seeded full texts; Track B: Garuda, SINTA, Google Scholar, and publisher portals, plus reference and author chaining; Track C: re-run of the strings, January to July 2026, with Semantic Scholar, Scopus, and PubMed indexes; Track D: chaining from Track C.
Search strings	Nine bilingual English and Indonesian strings, e.g. "QR Code" AND "digital signature" AND Indonesia; ("RSA-PSS" OR "RSASSA-PSS") AND "QR Code"; "replay attack" AND "QR Code".
Period, last search	2016 to 2026; 30 July 2026 (Tracks A to C), 2 September 2026 (Track D).
Eligibility	QR Code or signature-based authentication; a described cryptographic mechanism; Indonesian authors, use case, or venue; accessible full text.
Selection	One reviewer screened titles and abstracts, then full texts, and re-checked all codings; no automation tools; no registration.
Coding	C1 RSA-PSS present (primary or auxiliary); C2 PSS salt length adapted; C3 nonce-timestamp anti-replay implemented and tested; C4 explicit ISO/IEC 20248 alignment [44]; C5 all of C1 to C4.

III. RESULTS AND DISCUSSION

A. Why the Simulated Detection Rates Are Not Measurements

Earlier drafts reported, from the simulation, 92.21% replay detection, 78.14% tampering detection, 54.17% detection of the two critical tampering subtypes, and 97.08% forgery rejection, and interpreted the tampering shortfall as a weakness of semantic validation. Inspection of the harness source (modules/testing_controller.py in the archived release) shows that none of these figures involved the verification code. The tampering routine selects a subtype label k , draws U from a uniform distribution on $[0.95, 1.05]$, and records a detection if a pseudo-random number falls below $0.92 \times d_k \times U$, where d_k is a hard-coded difficulty constant. No payload is built, modified, or verified, and the generator was not seeded. The expected session rate therefore follows from the constants alone, as (5) shows, where w_k is the observed share of subtype k ; with the recorded shares, (5) gives $0.92 \times 0.8427 = 77.53\%$.

$$E[DR_{sim}] = 0.92 \sum_k w_k d_k \quad (5)$$

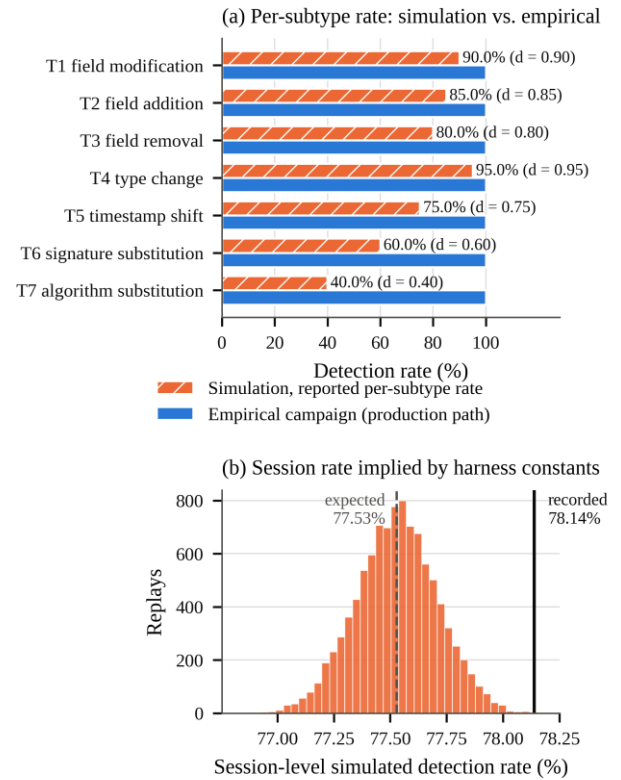


Fig. 2. (a) Simulated per-subtype rates coincide with the hard-coded constants d_k , whereas the empirical campaign rejected every operation. (b) Session rates generated by the harness constants alone, compared with the recorded 78.14%.

Four observations in Table V and Fig. 2 support this account. First, the per-subtype rates of 90.0%, 85.0%, 80.0%, 95.0%, 75.0%, 60.0%, and 40.0% equal the constants d_k exactly, because the routine that tabulates them computes $\lfloor n_k d_k \rfloor / n_k$ instead of counting outcomes. Second, the critical-category figure of 54.17% is $(\lfloor 0.60 \times 2,477 \rfloor + \lfloor 0.40 \times 1,016 \rfloor) / 3,493 = 1,892 / 3,493$, again a function of constants. Third, 10,000 replays of the decision rule with the recorded subtype counts give a session rate of 77.53% (95% range 77.17% to 77.89%). The recorded 78.14% lies 0.61 percentage points above that range; because the generator was unseeded and individual draws were not logged, this residual cannot be traced, but it cannot reflect system behaviour either, since no system code was called. Fourth, the replay and forgery routines follow the same pattern with base rates of 0.95 and 0.985 multiplied by pattern or type factors; they predict 92.17% and 96.94%, against recorded values of 92.21% and 97.08%.

TABLE V
SIMULATED TAMPERING FIGURES TRACED TO HARNESS CONSTANTS

Subtype	<i>n</i> (sim.)	<i>d</i>	Simulated	Empirical
T1 field modification	19,781	0.90	90.0%	100%
T2 field addition	9,934	0.85	85.0%	100%
T3 field removal	7,654	0.80	80.0%	100%
T4 type change	5,088	0.95	95.0%	100%
T5 timestamp shift	4,050	0.75	75.0%	100%
T6 signature substitution	2,477	0.60	60.0%	100%
T7 algorithm substitution	1,016	0.40	40.0%	100%
Session rate	50,000	Eq. (5)	78.14%	100%

The layer ablation in Section III.C gives an independent argument. Each verification layer decides every subtype deterministically, either always or never, so no configuration of the deployed verifier can produce graded rates such as 40% or 60% for a subtype. The 78.14% figure is therefore a property of the test harness and not a measured weakness of the system. The simulation is not used for any claim below, and the target comparisons against it that appeared in earlier drafts, including the z-test, are withdrawn.

B. Empirical Detection Results (RQ1)

Table VI reports the empirical campaign. All 50,000 tampering operations, 3,000 replay attempts, 2,451 expired tokens, and 4,000 forgery attempts were rejected, and none of the 5,549 negative controls or the 1,500 first uses was refused. Negative controls matter here: a verifier that rejected every input would also score 100% detection, but it would have refused all 7,049 legitimate presentations.

TABLE VI
EMPIRICAL DETECTION RESULTS WITH EXACT 95% BOUNDS

Scenario or control	<i>n</i>	Rejected	Lower bound	Bonferroni
Tampering, T1 to T7	50,000	50,000	$\geq 99.994\%$	$\geq 99.989\%$
T7, smallest subtype	974	974	$\geq 99.693\%$	$\geq 99.431\%$
Replay R1, payloads	1,500	1,500	$\geq 99.800\%$	$\geq 99.630\%$
Expired tokens R2	2,451	2,451	$\geq 99.878\%$	$\geq 99.773\%$
Forgery F1 to F4	4,000	4,000	$\geq 99.925\%$	$\geq 99.861\%$
F2, smallest type	963	963	$\geq 99.689\%$	$\geq 99.424\%$
Controls, tampering	2,500	0	$\text{FPR} \leq 0.120\%$	
Controls, forgery	500	0	$\text{FPR} \leq 0.597\%$	
Controls, validity	2,549	0	$\text{FPR} \leq 0.117\%$	
First uses (replay)	1,500	0	$\text{FPR} \leq 0.200\%$	
All legitimate inputs	7,049	0	$\text{FPR} \leq 0.042\%$	

Every one-sided lower bound exceeds 99.6%, including those of the smallest subtypes, and the pooled false positive rate is below 0.042% with 95% confidence. Adjusting for thirteen simultaneous subtype bounds lowers them only slightly. Mean decision time was 0.879 ms for tampering (median 0.862 ms, 95th percentile 1.051 ms), 0.671 ms for forgery, and 1.575 ms and 1.924 ms for replay and validity, which also write to the nonce ledger. Table VII gives the subtype breakdown. These results answer RQ1 within the stated threat model. They do not show that attacks outside Table II would fail, and the complete detection is expected rather than surprising, since every modification of a signed field changes the digest. What the campaign adds is evidence that the implementation realises this expectation for all tested inputs, without fail-open paths or false rejections.

TABLE VII
EMPIRICAL DETECTION RATE BY DATA TAMPERING SUBTYPE

Tampering subtype	Category	Attempts	Detected	Rate	Lower bound
<i>T1 field_modification</i>	Regular	20,030	20,030	100%	$\geq 99.985\%$
<i>T2 field_addition</i>	Regular	10,102	10,102	100%	$\geq 99.970\%$
<i>T3 field_removal</i>	Regular	7,467	7,467	100%	$\geq 99.960\%$
<i>T4 data_type_change</i>	Regular	4,993	4,993	100%	$\geq 99.940\%$
<i>T5 timestamp_tampering</i>	Regular	3,943	3,943	100%	$\geq 99.924\%$
<i>T6 signature_injection</i>	Critical	2,491	2,491	100%	$\geq 99.880\%$
<i>T7 encryption_bypass</i>	Critical	974	974	100%	$\geq 99.693\%$

Note. Harness labels are shown in italics; Table II gives the operation behind each label. Lower bounds are one-sided 95% Clopper-Pearson bounds.

C. Cryptographic Integrity versus Application-Level Validation (RQ2)

Fig. 3 shows the ablation. Signature verification alone (L1) rejected all seven tampering subtypes and all four forgery types, so cryptographic integrity by itself satisfies goals G1 and G2. It accepted all 3,000 replays and all 2,451 expired tokens, because those tokens carry genuine signatures over unmodified data. The application layers alone (L2 to L4, with the signature assumed valid) show the mirror image. They rejected replayed and expired tokens and tampering subtypes T1 to T5, which change the data, but accepted every operation that leaves the data untouched and attacks only the signature: T6, T7, and F1 to F4, 7,465 operations in total. Structural validation alone (L3) rejected only T2, T3, and T4, the subtypes that break the payload schema; a well-formed change of name or date passed it.

	Signature only (L1)	Structure only (L3)	Application only (L2–L4)	Full pipeline
T1 field modification	100%	0%	100%	100%
T2 field addition	100%	100%	100%	100%
T3 field removal	100%	100%	100%	100%
T4 type change	100%	100%	100%	100%
T5 timestamp shift	100%	0%	100%	100%
T6 signature substitution	100%	0%	0%	100%
T7 algorithm substitution	100%	0%	0%	100%
R1 replay (2nd/3rd use)	0%	n/a	100%	100%
R2 expired-token reuse	0%	n/a	100%	100%
F1 random signature	100%	n/a	0%	100%
F2 truncated signature	100%	n/a	0%	100%
F3 swapped signature	100%	n/a	0%	100%
F4 single-bit flip	100%	n/a	0%	100%

Fig. 3. Layer ablation over the identical campaign. Cells give the share of each subtype rejected by each verifier variant; n/a marks subtypes to which structural validation does not apply.

Two conclusions follow. First, the two kinds of assurance are complementary rather than redundant. Freshness (G3) and validity (G4) are not properties a signature can provide, so replay and expiry defences must be judged as application logic with their own state, and a verifier holding only the public key, such as an offline verifier, cannot enforce them. Second, the earlier claim that a semantic layer failed while the cryptographic core succeeded is reversed by measurement. Neither layer failed within its own responsibility; T1 to T5 are rejected by two or three independent layers, whereas signature substitution and forgery depend on L1 alone, and replay and expiry on L4 alone. Those single points of dependence are

where future hardening, such as key rotation and a replicated nonce ledger, should be concentrated.

D. Scalability over HTTP

Table VIII and Fig. 4 report the external load test: 25,607 requests over four levels. Throughput stayed between 100.4 and 111.5 requests per second across a fifteen-fold increase in concurrency, the signature of a saturated service: extra users lengthen the queue without adding completed work. Mean response time followed Little's law closely; N/X predicts 0.94, 4.98, 9.23, and 13.45 s against measured means of 0.93, 4.63, 9.09, and 13.06 s, a deviation of 1.4% to 7.0%. Error rates were 0% at 100 and 500 users and 0.09% and 0.06% at 1,000 and 1,500 users, 10 failures in all, with every upper confidence limit at or below 0.21%. Both the proposal target of 0.79% at 500 users and the 2% design threshold were met at every level.

TABLE VIII
HTTP LOAD MEASUREMENT FROM AN EXTERNAL HOST

Users	Requests	Req/s	Mean (s)	p95 (s)	Error rate, % (95% CI)
100	6,389	106.5	0.93	1.11	0.00 (0 to 0.06)
500	6,025	100.4	4.63	5.65	0.00 (0 to 0.06)
1,000	6,501	108.3	9.09	11.88	0.09 (0.03 to 0.20)
1,500	6,692	111.5	13.06	19.16	0.06 (0.02 to 0.15)

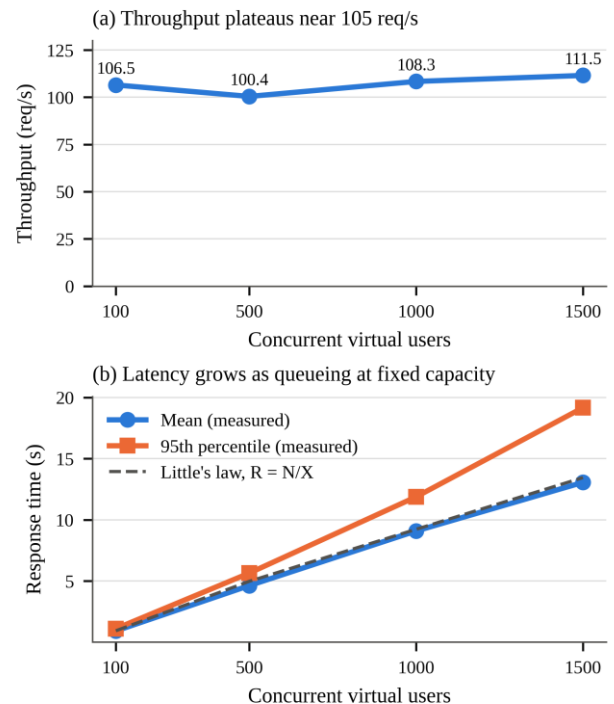


Fig. 4. External HTTP load test: (a) throughput and (b) mean and 95th-percentile response time, with the Little's law prediction $R = N/X$.

The capacity ceiling is not set by the cryptography. RSASSA-PSS verification took 0.865 ms on average in a calibration of 100,000 operations, about 9% of the 9.5 ms

interval between completions at 105 requests per second. Reaching the final result required three infrastructure corrections that each first appeared as application failures: a reverse-proxy limit of 768 worker connections, which caused 45% failures at 1,000 users; an open-file limit, which caused a further 7.6%; and a record index that was not marked authoritative, which made each lookup scan 120,000 files in 148.6 ms instead of 0.074 ms. The simulation had reported error rates of 1.78% to 4.28% and attributed saturation to nonce tracking; direct profiling put nonce recording at 0.8 ms, so that attribution is also withdrawn. The evaluation remains limited to one deployment configuration, one run of 60 seconds per level, and a single service instance; latencies above one second at 500 users mean that horizontal scaling, rather than error rate, is the binding constraint.

E. Comparative Security Evaluation and PRISMA 2020 Novelty Analysis (RQ3)

Table IX compares the proposed system against thirteen individually itemised comparators — the ten Indonesian Track A studies [13]–[22], one international study [10], and the two most recent Indonesian studies [24], [45] — plus one group summary row standing for fourteen further PRISMA-validated Indonesian studies. All values are taken directly from the cited sources. NR marks what a cited study did not report; NA marks a criterion that does not structurally apply. Three differential advantages hold up under this comparison: integrated anti-replay protection, which no comparator provides; the broadest evaluation coverage in the Indonesian corpus, with 66,500 empirically measured operations across thirteen defined attack subtypes and 5,549 negative controls; and transparent multi-metric security reporting.

TABLE IX
COMPARATIVE SECURITY EVALUATION AGAINST PRIOR QR CODE RESEARCH

Study	Method	Replay detection	Tampering accuracy	Forgery rejection	ISO 20248	Test operations
This study	RSA-PSS + nonce + ISO/IEC 20248 (partial)	100% (n = 3,000)	100% (n = 50,000)	100% (n = 4,000)	Partial (self-assessed)	66,500 empirical
Lorien & Wellem [13]	SHA-256 + RSA + QR	None	NR (a)	100%, 15/15 (a)	No	30 (15V + 15F)
Nuraeni et al. [14]	MD5 + RSA + AES-128	None	NR (b)	NR	No	27
Az-Zahra et al. [15]	SHA-256 + 3-prime RSA	None	100% (1/1)	100% (1/1)	No	3 scenarios
Almoussa et al. [10]	Dual ML (QR Shield)	None	NA (g)	NA	No	NR
Saepulrohman & Negara [18]	ECDSA + ECDH + QR	NR	NR	NR	No	—
Abdurrachman & Suteja [16]	SHA-256 + AES-256	NR	NR (c)	NR	No	10
Sholihah et al. [19]	SHA-256 + DB + PDF	NR	NR	NR	No	~17 scenarios
Nadzifarin & Asmunin [22]	ECDSA + BLAKE2b + PDF	NR	100% (1/1)	100% (2/2)	No	25
Prabowo & Afrianto [17]	SHA-256 + RSA + QR	NR	100% (1/1)	NR	No	1 test
Ardhianto & Wakhidah [20]	QR imaging + PHP	NR	NR	NR	Ref. 18004	—
Kurniawan et al. [21]	MD5 + QR + CodeIgniter	NR	NR	NR	No	7 (valid)
Kharisma et al. [24]	DSA-2048 + RSA-PSS-2048 + SHA-256 + QR	None	100% (1/1)	NR	No	4 security scenarios (e)
Paula et al. [45]	ISO/IEC 18004 QR + server-side token	None (token expiry/revoke)	NR (d)	NA	Ref. 18004	14 black-box scenarios (f)
Tracks B–D (n = 14)*	RSA-PKCS#1-v1.5 / raw RSA / AES / hash-only	None (0/14)	NR	NR	No (0/14)	NR

Note. NR = not reported; NA = not applicable. (a) No independent forgery test; 100% of 15 randomly fabricated QR Codes. (b) Tampering accuracy not quantified numerically. (c) An avalanche effect of 40.97% was reported, which is not equivalent to tampering detection accuracy. (d) Validation is a token-to-database lookup; document content is not verified cryptographically. (e) Four security scenarios, one trial each; the accompanying 5–20-page runs are computational-performance profiling, not adversarial testing. (f) Functional black-box testing; four of the fourteen cases exercise token validity states (valid, invalid, expired, revoked). (g) QR Shield classifies malicious URLs embedded in QR Codes; its accuracy is not a tampering-detection rate. * Tracks B–D comprise fourteen further Indonesian studies; full citations are provided in the Supplementary Material (PRISMA 2020 Novelty Validation Report). Figures for this study are taken from the empirical campaign (Section III.B), because the comparators likewise report measurements of their own deployed systems; the simulation figures are not used (Section III.A). Two cautions apply when reading the rate columns. First, complete detection in this study is structural within the stated threat model, as Section III.B explains, so the columns should not be read as a ranking of detector quality. Second, several comparator figures rest on a single trial (reported as 1/1), so the more informative contrast is the evaluation scale in the final column rather than the headline rates.

To examine the novelty claim systematically, an independent systematic review following PRISMA 2020 [41], reported in full in the Supplementary Material, widened the Indonesian screening corpus to twenty-six QR Code-based digital signature and document authentication studies published between 2016 and 2026. Ten came from the initial comparator set (Track A, [13]–[22]); nine (Track B, cited in full in the Supplementary Material) came from systematic bilingual keyword searches of Garuda, SINTA, Google Scholar, and Indonesian publisher portals; seven more (Track C) came from a confirmatory update search covering January–July 2026; and two (Track D) came from reference chaining over the Track C records. Twenty-eight records were identified in total, of which two were excluded at screening — one non-cryptographic legal analysis and one non-

Indonesian protocol (Fig. 5). Exactly one of the twenty-six satisfies C1: Kharisma et al. [24] employ RSA-PSS-2048, but as an auxiliary signature over a DSA-2048 signature package rather than as the signature scheme of the system, and they report neither an adapted PSS salt length nor any replay-detection mechanism; their 5–20-page timing profile measures computational overhead rather than adversarial resistance. No study in the corpus satisfies C2, C3, or C4, which leaves C5 at 0/26: no direct predecessor for this integrated architecture was identified within the screened corpus. RSASSA-PSS as the primary signature scheme, therefore remains unreported in that corpus, and within this scope the novelty lies in the integrated architecture rather than in any single component. Table X and Table XI report the novelty matrix and the PRISMA 2020 stage accounting.

TABLE X
 PRISMA 2020 NOVELTY VALIDATION MATRIX: INDONESIAN CORPUS (N = 26, 2016–2026)

Technology family (n, %)	Studies (ref.)	C1 RSA-PSS	C2 Salt	C3 Anti-replay	C4 ISO 20248	C5 Predecessor
A — RSA-based (n = 13, 50.0%)	[13]–[15], [17] + 9*	No	No	No	No	No
B — Hash/metadata/token (n = 8, 30.8%)	[19]–[21], [45] + 4*	No	No	No	No	No
C — ECDSA (n = 2, 7.7%)	[18], [22]	No	No	No	No	No
D — AES (n = 2, 7.7%)	[16] + 1*	No	No	No	No	No
E — Hybrid DSA + RSA-PSS (n = 1, 3.8%)	[24]	Yes (auxiliary)	No	No	No	No
Combined (N = 26, 100%)	—	1/26	0/26	0/26	0/26	0/26

Note. C1 = RSA-PSS/RSASSA-PSS present as a signature scheme, whether primary or auxiliary; C2 = adapted PSS salt length; C3 = nonce-timestamp anti-replay; C4 = ISO/IEC 20248:2022 alignment; C5 = simultaneous satisfaction of C1–C4. Of the 28 records identified through the PRISMA 2020 protocol [41], two were excluded at screening, and 26 were retained. The single C1 = Yes entry is an auxiliary RSA-PSS layer in one 2026 study [24]; RSA-PSS as the primary signature scheme was not identified in the screened Indonesian QR Code authentication literature. * Full citations for the Track B–D records are provided in the Supplementary Material.

TABLE XI
 PRISMA 2020 STAGE ACCOUNTING — COMBINED INDONESIAN CORPUS (N = 26)

PRISMA 2020 stage	n
Records identified — Indonesian full-text corpus (Track A)	10
Records identified — supplementary database search/reference chaining (Track B)	9
Records identified — confirmatory update search, January–July 2026 (Track C)	7
Records identified — reference chaining from the update-search records (Track D)	2
Total records identified	28
Duplicates removed	0
Records excluded at screening (title/abstract)	2
Records excluded at eligibility (full text/metadata)	0
Records included in the combined novelty matrix	26
Studies satisfying the direct-predecessor criterion C5	0

Note. Two records were excluded at screening — a non-cryptographic legal analysis and a non-Indonesian protocol study — and none at the eligibility stage, reflecting the deliberately broad and sensitive eligibility criteria

applied to ensure comprehensive coverage of the Indonesian QR Code authentication landscape (2016–2026).

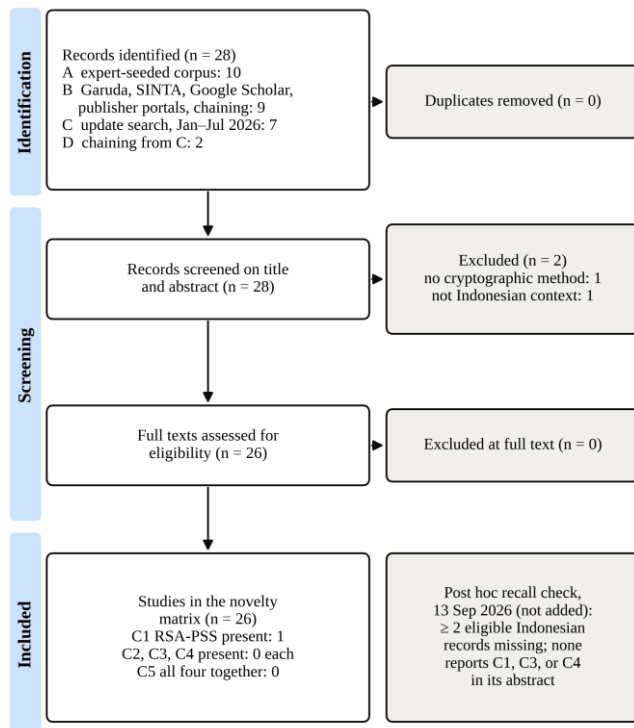


Fig. 5. PRISMA 2020 flow of record identification, screening, and inclusion, with the post hoc recall check.

This result supports a bounded statement: no direct predecessor was identified within the 26 Indonesian records screened. It does not establish global novelty or completeness, for five reasons. The corpus excludes non-Indonesian work by design, and outside it a Schnorr-based QR identification protocol with nonce and timestamp replay protection was encountered and excluded on geography alone. Track A was seeded by expert selection, and Google Scholar searches cannot be replicated exactly. Raw hit counts per source were not recorded, so recall cannot be estimated. Screening and coding were done by one reviewer without an agreement statistic. Finally, a post hoc check of the Crossref index on 13 September 2026 found at least two eligible Indonesian records absent from the corpus [46], [47]; their abstracts report ECDSA and an unspecified signature scheme with distributed hash storage, and neither mentions RSA-PSS, nonce-timestamp replay control, or ISO/IEC 20248, but they show that recall was incomplete. The novelty statement is therefore bounded to the screened corpus, and the principal

contribution rests on the evaluation evidence of Sections III.B to III.D rather than on the review. The system itself follows ISO/IEC 20248 only in part: it signs structured data, but it stores payloads server-side and uses a 256-byte signature, so it does not provide the offline verification and compact signatures that the standard anticipates.

F. Threats to Validity and Consolidated Gap Analysis

Five threats to validity follow the Campbell-Stanley framework [48]. Internal validity is limited by testing on a single deployment host (Ubuntu, two vCPUs), although the empirical harness is seeded, runs against a fixed software release, and isolates its security state from production. External validity is limited by the attack generator of Table II, which may not mirror every adversarial behaviour, and by a staging configuration with a single service instance. Construct validity is shaped by two choices: detection is counted as any verdict other than valid, and the forgery subtypes are non-adaptive structural tests rather than attempts on the RSA assumption. Statistical conclusion validity rests on exact Clopper-Pearson bounds computed under the independence assumptions stated in Section II.D, with replay bounded conservatively at the payload level. Novelty validity is bounded by the scope of the PRISMA 2020 review: the C5 result of 0/26 means first identified within the reviewed corpus, not first in the world.

Grey literature, patent databases, and non-indexed Indonesian proceedings were not covered exhaustively, which leaves a residual risk for prior art discovery. All twenty-six records were checked against full text. Only one, Kharisma et al. [24], uses RSA-PSS, and there is an auxiliary layer whose salt length is not reported; no record uses an adapted PSS salt length or a nonce-timestamp anti-replay mechanism, and none aligns with ISO/IEC 20248:2022 — the closest, Paula et al. [45], invokes ISO/IEC 18004, which governs QR symbology rather than signature structure. The shape of the gap invites interpretation. RSA-based schemes dominate Family A at 50.0%, which reflects the ready availability of standard RSA implementations in common Indonesian development environments. The complete absence of ISO/IEC 20248:2022 alignment is unsurprising for a standard published in 2022. The absence of nonce-timestamp anti-replay across every family is harder to explain away and points to a systematic underestimation of the QR Code replay threat. Table XII consolidates the gap analysis with root causes and actionable remediation.

TABLE XII
GAP ANALYSIS: TARGETS VS. ACTUAL RESULTS WITH ROOT CAUSES AND REMEDIATION

Metric	Target	Simulation (withdrawn)	Empirical	Remaining gap and remediation
Replay detection	$\geq 98\%$	92.21%	100% ($\geq 99.800\%$)	Closed. The simulated shortfall came from constants in the harness, not from the verification path (Section III.A).
Data tampering accuracy	$\geq 85\%$	78.14%	100% ($\geq 99.994\%$)	Closed. The simulated figure reflects difficulty constants encoded in the harness (Section III.A).
Forgery rejection	$\geq 99.9\%$	97.08%	100% ($\geq 99.925\%$)	Closed as an implementation conformance test. EUF-CMA security is a property of the scheme and is not measured by this rate.
Error rate @1,000 users	$\leq 2\%$	3.07%	0.09%	Closed. Measured over HTTP from an external load generator (Section III.D). The simulated shortfall did not appear on the running service; the earlier attribution to file-based nonce tracking was incorrect.
Single-layer dependence	—	not assessed	L1 alone for T6, T7, F1–F4; L4 alone for R1, R2	Open. Key rotation and a replicated nonce ledger (Section III.C).
Printed-QR scan robustness	—	not tested	not tested	Open. Scheduled with borrowed scanning hardware.

Note. Metric = the evaluated parameter. Values in parentheses are one-sided 95% Clopper-Pearson lower bounds (Table VI). All remediation measures are specific and actionable.

IV. CONCLUSION

This study set out to measure how reliably an RSA-PSS QR Code verification service rejects defined attacks, which layer is responsible, and how the system relates to prior work. Within a threat model of thirteen executable attack subtypes, the production verification path rejected all 59,451 attack operations and accepted all 7,049 legitimate presentations, with one-sided 95% lower bounds of 99.994% for tampering, 99.925% for forgery, and 99.800% for replay. The layer ablation showed that the signature alone guarantees integrity and authenticity but not freshness or validity, while application-level checks alone cannot detect signature substitution or forgery; reliable verification needs both. The simulated rates reported in earlier drafts, including 78.14% tampering detection, were products of hard-coded constants and are withdrawn as evidence. Measured over HTTP, the service kept error rates at or below 0.09% up to 1,500 concurrent users at a throughput ceiling of about 105 requests per second that is unrelated to the signature scheme. Within the 26 Indonesian studies reviewed, no study combined RSA-PSS, adapted salt length, nonce-timestamp replay control, and ISO/IEC 20248 alignment. These detection rates are conformance evidence and do not replace the formal EUF-CMA guarantee of the scheme.

This distinction carries a methodological lesson beyond the present system. Calibrated simulation is a legitimate instrument for exploring attack composition and load distribution at volumes impractical against a live service, but it cannot substitute for direct measurement when the claim at stake is the detection capability of a deployed verification path. Security claims reported in this paper rest on the empirical campaign; the simulation results are reported only

to explain superseded figures and are labelled as such throughout.

Future work runs along five lines: (1) integrating CRYSTALS-Dilithium (NIST FIPS 204) and benchmarking it against the RSA-PSS baseline; (2) replicating the nonce ledger and scaling the service horizontally, with repeated HTTP load runs beyond 1,500 concurrent users; (3) extending the PRISMA 2020 review to grey literature, patent databases, institutional repositories, and international databases with dual screening; (4) testing the scan robustness of printed QR Codes; and (5) field trials with governmental or private identity verification partners.

DATA AVAILABILITY

Every figure in this paper can be recomputed from archived material. The software, the measurement harnesses, and the complete datasets are deposited in a single Zenodo record under the MIT licence, citable through the concept DOI 10.5281/zenodo.21271011, which always resolves to the most recent version. The empirical campaign reported in this paper was produced with version 1.3.1, DOI 10.5281/zenodo.22292443 [49], and the layer ablation of Section III.C was added in version 1.4.2, DOI 10.5281/zenodo.22734707 [50]; readers reproducing these results should cite the version DOIs rather than the concept DOI, because the concept DOI will move as later versions are deposited. The deposit contains the calibrated simulation dataset of 350,000 operations exported from the test database and verified to be identical to its source across 461,950 values; the empirical measurement package of 66,500 operations reported in Section III.B, comprising the raw JSON output, three derived CSV summaries, and a SHA-256 manifest; the layer ablation output with its own summaries and manifest; and the external HTTP load measurements of

Section III.D. The harnesses are included as runnable scripts, and the empirical campaign is deterministic: re-executing it with the recorded seed (20260824) on the same revision reproduces the reported counts exactly. Only the timing statistics vary between runs, since they are measured against wall-clock time.

ACKNOWLEDGEMENT

This research was funded by the Ministry of Education, Science, and Technology of the Republic of Indonesia (Kemendiknasaintek) under the Penelitian Dosen Pemula (PDP) scheme, contract No. 284/C3/DT.05.00/PL-BARU/2026 and derivative contract No. 013/KONTRAK-LPPM/AMIKOM/V/2026. The authors thank LPPM Universitas Amikom Yogyakarta for administrative support, and the Programming Laboratory of Universitas Amikom Yogyakarta for the computing infrastructure used in prototype development and testing.

REFERENCES

- [1] F. Sharevski, A. Devine, E. Pieroni, and P. Jachim, "Phishing with malicious QR codes," in *Proc. 2022 European Symp. Usable Security (EuroUSEC '22)*, ACM, 2022, pp. 160–171, doi: 10.1145/3549015.3554172.
- [2] H. A. M. Wahsheh and M. S. Al-Zahrani, "Secure real-time computational intelligence system against malicious QR code links," *Int. J. Comput. Commun. Control*, vol. 16, no. 3, Art. no. 4186, 2021, doi: 10.15837/ijccc.2021.3.4186.
- [3] R. Yalda, A. Khan, and N. Nepal, "B-PhishQR—A blockchain-based framework for secure QR code verification against phishing attacks," *Telematics Informatics Rep.*, vol. 21, Art. no. 100289, 2026, doi: 10.1016/j.teler.2025.100289.
- [4] M. A. Al-Shareeda, S. Manickam, S. A. Laghari, and A. Jaisan, "Replay-attack detection and prevention mechanism in Industry 4.0 landscape for secure SECS/GEM communications," *Sustainability*, vol. 14, no. 23, Art. no. 15900, 2022, doi: 10.3390/su142315900.
- [5] Y. Zhou, B. Hu, Y. Zhang, and W. Cai, "Implementation of cryptographic algorithm in dynamic QR code payment system and its performance," *IEEE Access*, vol. 9, pp. 122362–122372, 2021, doi: 10.1109/ACCESS.2021.3108189.
- [6] Y. Yan, Z. Zou, H. Xie, Y. Gao, and L. Zheng, "An IoT-based anti-counterfeiting system using visual features on QR code," *IEEE Internet Things J.*, vol. 8, no. 8, pp. 6789–6799, 2021, doi: 10.1109/IIOT.2020.3035697.
- [7] T. Wang, H. Zheng, C. You, and J. Ju, "A texture-hidden anti-counterfeiting QR code and authentication method," *Sensors*, vol. 23, no. 2, Art. no. 795, 2023, doi: 10.3390/s23020795.
- [8] S. A. Alsubhany, "Innovative QR code system for tamper-proof generation and fraud-resistant verification," *Sensors*, vol. 25, no. 13, Art. no. 3855, 2025, doi: 10.3390/s25133855.
- [9] J. A. Berrios Moya, J. Ayoade, and M. A. Uddin, "A zero-knowledge proof-enabled blockchain-based academic record verification system," *Sensors*, vol. 25, no. 11, Art. no. 3450, 2025, doi: 10.3390/s25113450.
- [10] H. Almousa, A. Almarzqi, A. Alassaf, G. Alrasheed, and S. A. Alsubhany, "QR Shield: A dual machine learning approach towards securing QR codes," *Int. J. Comput. Digit. Syst.*, vol. 15, no. 1, pp. 887–898, 2024, doi: 10.12785/ijcds/160164.
- [11] K. Somsuk, "The development of signing and verification methods for high speed digital signatures on electronic official documents by using RSA cryptography," *Cogent Eng.*, vol. 11, no. 1, Art. no. 2432513, 2024, doi: 10.1080/23311916.2024.2432513.
- [12] K. Somsuk, "The special algorithm based on RSA cryptography for signing and verifying digital signature," *Heliyon*, vol. 11, no. 4, Art. no. e42481, 2025, doi: 10.1016/j.heliyon.2025.e42481.
- [13] A. Lorien and T. Wellem, "Implementasi sistem otentikasi dokumen berbasis Quick Response (QR) Code dan digital signature," *J. RESTI (Rekayasa Sist. Teknol. Inf.)*, vol. 5, no. 4, pp. 663–671, 2021, doi: 10.29207/resti.v5i4.3316.
- [14] F. Nuraeni, D. Kurniadi, and D. N. Rahayu, "Implementation of RSA and AES-128 super encryption on QR-code based digital signature schemes for document legalization," *J. Tek. Inform. (JUTIF)*, vol. 5, no. 3, pp. 675–684, 2024, doi: 10.52436/1.jutif.2024.5.3.1426.
- [15] F. Az-Zahra, R. Marwati, and R. Sispiyati, "Implementasi QR Code dengan algoritma SHA-256 dan RSA yang ditingkatkan untuk autentikasi dokumen digital," *J. EurekaMatika*, vol. 12, no. 1, pp. 11–22, 2024.
- [16] T. Abdurrachman and B. R. Suteja, "Pengembangan sistem informasi asosiasi jasa konstruksi dengan menerapkan tanda tangan digital," *J. Tek. Inform. Sist. Inf.*, vol. 7, no. 1, pp. 261–273, 2021, doi: 10.28932/jutisi.v7i1.3431.
- [17] E. C. Prabowo and I. Afrianto, "Penerapan digital signature dan kriptografi pada otentikasi sertifikat tanah digital," *Komputa: J. Ilm. Komput. Inform.*, vol. 6, no. 2, pp. 83–90, 2017, doi: 10.34010/komputa.v6i2.2481.
- [18] A. Saepulrohman and T. P. Negara, "Implementasi algoritma tanda tangan digital berbasis kriptografi kurva eliptik Diffie-Hellman," *Komputasi: J. Ilm. Ilmu Komput. Mat.*, vol. 18, no. 1, pp. 22–28, 2021, doi: 10.33751/komputasi.v18i1.2569.
- [19] W. Sholihah, S. Indriasari, I. Noviyanti, A. Mardiyono, and N. Aziezhah, "ESVISIGN: Tanda tangan digital Sekolah Vokasi IPB," *JTIM: J. Teknol. Inf. Multimedia*, vol. 3, no. 4, pp. 217–226, 2022, doi: 10.35746/jtim.v3i4.188.
- [20] E. Ardianto and N. Wakhidah, "Pengembangan metode otentikasi keaslian ijasah dengan memanfaatkan gambar QR Code," *J. Transformatika*, vol. 13, no. 2, pp. 35–41, 2016, doi: 10.26623/transformatika.v13i2.325.
- [21] R. Kurniawan R, R. F. Sari, and N. Azizah, "Sistem validasi keaslian dokumen digital berbasis QR-Code," *J. Teknol. Inf.*, vol. 4, no. 2, pp. 321–327, 2020, doi: 10.36294/jurti.v4i2.1722.
- [22] A. Nadzifarin and Asmunin, "Penerapan elliptic curve digital signature algorithm pada tanda tangan digital dengan studi kasus dokumen surat-menyurat," *J. Inform. Comput. Sci. (JINACS)*, vol. 4, no. 1, pp. 1–9, 2022, doi: 10.26740/jinacs.v4n01.p1-9.
- [23] X. Wang and H. Yu, "How to break MD5 and other hash functions," in *Advances in Cryptology – EUROCRYPT 2005*, LNCS, vol. 3494. Berlin, Germany: Springer, 2005, pp. 19–35, doi: 10.1007/11426639_2.
- [24] C. R. N. Kharisma, Suprpto, and A. Saputra, "Implementasi tanda tangan digital berbasis kombinasi algoritma Digital Signature Algorithm dan Rivest Shamir Adleman," *J. Inform.*, vol. 15, no. 1, pp. 15–28, 2026, doi: 10.55340/jiu.v15i1.2684.
- [25] K. Moriarty, B. Kaliski, J. Jonsson, and A. Rusch, "PKCS #1: RSA cryptography specifications version 2.2," IETF, RFC 8017, Nov. 2016, doi: 10.17487/RFC8017.
- [26] National Institute of Standards and Technology, "Digital signature standard (DSS)," FIPS PUB 186-5, Feb. 2023, doi: 10.6028/NIST.FIPS.186-5.
- [27] The MITRE Corporation, "Common Attack Pattern Enumeration and Classification (CAPEC)," version 3.9, Jan. 2023. [Online]. Available: <https://capec.mitre.org>
- [28] E. Barker, "Recommendation for key management: Part 1 – General," NIST SP 800-57 Part 1 Rev. 5, May 2020, doi: 10.6028/NIST.SP.800-57pt1r5.
- [29] J.-S. Coron, "Optimal security proofs for PSS and other signature schemes," in *Advances in Cryptology – EUROCRYPT 2002*, LNCS, vol. 2332. Berlin, Germany: Springer, 2002, pp. 272–287, doi: 10.1007/3-540-46035-7_18.
- [30] W. Xiong and R. Lagerström, "Threat modeling – A systematic literature review," *Comput. Secur.*, vol. 84, pp. 53–69, 2019, doi: 10.1016/j.cose.2019.03.010.
- [31] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Trans. Inf. Theory*, vol. 29, no. 2, pp. 198–208, 1983, doi: 10.1109/TIT.1983.1056650.

- [32] A. M. Law, *Simulation Modeling and Analysis*, 5th ed. New York, NY, USA: McGraw-Hill, 2014.
- [33] C. J. Clopper and E. S. Pearson, "The use of confidence or fiducial limits illustrated in the case of the binomial," *Biometrika*, vol. 26, no. 4, pp. 404–413, 1934, doi: 10.1093/biomet/26.4.404.
- [34] L. D. Brown, T. T. Cai, and A. DasGupta, "Interval estimation for a binomial proportion," *Stat. Sci.*, vol. 16, no. 2, pp. 101–133, 2001, doi: 10.1214/ss/1009213286.
- [35] J. A. Hanley and A. Lippman-Hand, "If nothing goes wrong, is everything all right? Interpreting zero numerators," *JAMA*, vol. 249, no. 13, pp. 1743–1745, 1983, doi: 10.1001/jama.1983.03330370053031.
- [36] S. Goldwasser, S. Micali, and R. L. Rivest, "A digital signature scheme secure against adaptive chosen-message attacks," *SIAM J. Comput.*, vol. 17, no. 2, pp. 281–308, 1988, doi: 10.1137/0217017.
- [37] M. Bellare and P. Rogaway, "The exact security of digital signatures – How to sign with RSA and Rabin," in *Advances in Cryptology – EUROCRYPT '96*, LNCS, vol. 1070. Berlin, Germany: Springer, 1996, pp. 399–416, doi: 10.1007/3-540-68339-9_34.
- [38] Z. M. Jiang and A. E. Hassan, "A survey on load testing of large-scale software systems," *IEEE Trans. Softw. Eng.*, vol. 41, no. 11, pp. 1091–1118, 2015, doi: 10.1109/TSE.2015.2445340.
- [39] J. D. C. Little, "A proof for the queuing formula: $L = \lambda W$," *Oper. Res.*, vol. 9, no. 3, pp. 383–387, 1961, doi: 10.1287/opre.9.3.383.
- [40] N. J. Gunther, *Guerrilla Capacity Planning*. Berlin, Germany: Springer, 2007.
- [41] M. J. Page *et al.*, "The PRISMA 2020 statement: An updated guideline for reporting systematic reviews," *BMJ*, vol. 372, Art. no. n71, 2021, doi: 10.1136/bmj.n71.
- [42] M. L. Rethlefsen *et al.*, "PRISMA-S: An extension to the PRISMA statement for reporting literature searches in systematic reviews," *Syst. Rev.*, vol. 10, Art. no. 39, 2021, doi: 10.1186/s13643-020-01542-z.
- [43] B. Kitchenham, O. P. Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, "Systematic literature reviews in software engineering – A systematic literature review," *Inf. Softw. Technol.*, vol. 51, no. 1, pp. 7–15, 2009, doi: 10.1016/j.infsof.2008.09.009.
- [44] ISO/IEC 20248:2022, *Information technology – Automatic identification and data capture techniques – Data structures – Digital signature meta structure*. Geneva, Switzerland: ISO, 2022.
- [45] E. W. Paula, A. I. Hidayat, and N. W. Asbara, "Implementasi autentikasi QR Code berbasis standar ISO/IEC 18004 menggunakan tanda tangan digital," *J. Sist. Komput. Inform. (JSON)*, vol. 7, no. 3, pp. 1130–1140, 2026, doi: 10.30865/json.v7i3.9641.
- [46] T. Wellem, Y. Nataliani, and A. Iriani, "Academic document authentication using elliptic curve digital signature algorithm and QR code," *JOIV: Int. J. Inform. Vis.*, vol. 6, no. 3, p. 667, 2022, doi: 10.30630/joiv.6.2.872.
- [47] W. Ayub, I. Winarno, and A. Sudarsono, "QR code-based smart document implementation using distributed database and digital signature," *Indones. J. Comput. Sci.*, vol. 13, no. 1, 2024, doi: 10.33022/ijcs.v13i1.3673.
- [48] D. T. Campbell and J. C. Stanley, *Experimental and Quasi-Experimental Designs for Research*. Boston, MA, USA: Houghton Mifflin, 1963.
- [49] A. A. Slameto, Y. Sutanto, and H. Kurniawan, "QR Code Security System RSA-PSS: Software, measurement harnesses and evaluation datasets," version 1.3.1, Zenodo, 4 Sep. 2026, doi: 10.5281/zenodo.22292443.
- [50] A. A. Slameto, Y. Sutanto, and H. Kurniawan, "QR Code Security System RSA-PSS: Software, measurement harnesses and evaluation datasets," version 1.4.2, Zenodo, 13 Sep. 2026, doi: 10.5281/zenodo.22734707.