RSA Digital Signatures

Lecture 24a: From trapdoors to public verification

Lecture overview

RSA gives a secret way to take roots and a public way to check them. We use this asymmetry to sign messages, break the first construction, and examine why the message encoding matters.

Based on the instructor's Lecture 24a, pp. 1–12 [Maj26].

Abstract and lecture roadmap

We follow the source lecture: recall RSA, reverse its roles for signing, construct an existential forgery, and introduce a fixed public prefix.

1. Define what a signature guarantees after an attacker obtains signatures.
2. Prove correctness of raw RSA signatures and then break their security.
3. Analyze the source's prefix encoding, including the limit of the argument.
4. Preview full-domain hashing as a construction with a security theorem.

The question that drives the lecture

If taking an RSA root is hard, how can an attacker produce a valid message–signature pair without taking any roots?

Source sequence: [Maj26, pp. 2–12]; definitions: [BS23, §13.1].

Three conclusions to keep in view

Correctness follows from the RSA inverse

If $\sigma = m^d \bmod N$, then $\sigma^e \equiv m \pmod{N}$.

Raw RSA signatures admit an existential forgery

Choose σ first and set $m = \sigma^e \bmod N$. This gives a valid pair when m belongs to the message space. Even the restricted source domain admits $(m, \sigma) = (1, 1)$.

Encoding needs its own security argument

A fixed prefix restricts the accepted RSA images. We prove correctness and a bound for uniform guessing; these do not establish unforgeability.

[HPS14, Table 4.1, p. 197]; [BS23, §13.3, p. 536]; source pp. 9–12.

Recall: one public permutation, one secret inverse

Choose distinct odd n-bit primes p, q and set

$$N = pq, \qquad \varphi(N) = (p-1)(q-1), \qquad ed \equiv 1 \pmod{\varphi(N)}.$$

Public key: (N, e)

For $x \in \mathbb{Z}_N^*$,

$$f_e(x) = x^e \bmod N.$$

Repeated squaring evaluates f_e efficiently.

Secret key: d

For $y \in \mathbb{Z}_N^*$,

$$f_e^{-1}(y) = y^d \bmod N.$$

The signer can take RSA roots efficiently.

The RSA assumption concerns finding the preimage of a *uniform random* image without the trapdoor. The last section states it precisely.

[HPS14, §3.2, pp. 119–125]; [BS23, §10.3, pp. 403–406]; [RSA78].

Recall: correct RSA decryption is only a first step

The source recalls the randomized encoding

$$x = s \,\|\, m = 2^{n/2}s + m, \qquad c = x^e \bmod N,$$

where $s, m \in \{0,1\}^{n/2}$ and $n \geq 4$ is even.

Correctness

The secret exponent recovers $c^d \equiv x \pmod{N}$. Since $0 \leq x < 2^n < N$, the receiver recovers the encoding and reads its suffix m.

Security requires an additional argument

Fresh pads and a birthday bound control repeated encodings. They do not by themselves prove privacy under the RSA assumption. As in Lecture 23, we retain this as a candidate encoding.

Source recall: [Maj26, pp. 2–3]; analyzed encryption: [BS23, §11.4].

Encryption and signatures use the trapdoor differently

	RSA encryption idea	**RSA signature idea**
Public operation	Encrypt an encoded message with $x \mapsto x^e$	Check a signature with $\sigma \mapsto \sigma^e$
Secret operation	Decrypt with $c \mapsto c^d$	Sign an encoded message with $x \mapsto x^d$
Security goal	Hide the message	Prevent a fresh valid forgery

Efficiency is relative to the information available

Signing is efficient *with* the secret key. The goal is to make forgery difficult for an attacker who has the public key and access to signatures.

This is an intuition for designing RSA schemes. The raw operations alone do not supply either security guarantee.

Source abstraction: [Maj26, pp. 4,6]; [HPS14, §4.1, pp. 193–196].

Three algorithms and one correctness requirement

A digital signature scheme

$$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^n), \quad \sigma \leftarrow \mathsf{Sign}_{\mathsf{sk}}(m), \quad \mathsf{Ver}_{\mathsf{pk}}(m, \sigma) \in \{\mathsf{accept}, \mathsf{reject}\}.$$

The public key determines the allowed message space $\mathcal{M}_{\mathsf{pk}}$.

Correctness

For every generated key pair and every $m \in \mathcal{M}_{\mathsf{pk}}$,

$$\Pr[\mathsf{Ver}_{\mathsf{pk}}(m, \mathsf{Sign}_{\mathsf{sk}}(m)) = \mathsf{accept}] = 1.$$

Anyone can check a signature. A verifier must also know whose public key it has, for example through an authenticated key distribution mechanism.

[BS23, Def. 13.1, pp. 528–529]; [HPS14, §4.1]; source p. 5.

Security: a fresh message after chosen-message queries

1. Generate $(\mathsf{pk}, \mathsf{sk})$ and give pk to the attacker.
2. The attacker adaptively requests signatures on $m_1, \ldots, m_q$.
3. It outputs (m^*, σ^*) and wins if

$$m^* \in \mathcal{M}_{\mathsf{pk}}, \quad m^* \notin \{m_1, \ldots, m_q\}, \quad \mathsf{Ver}_{\mathsf{pk}}(m^*, \sigma^*) = \text{accept}.$$

EUF-CMA security

Every efficient attacker has negligible winning probability. The probability includes key generation and all random choices.

Existential: the attacker may choose the forged message.
Chosen-message: the attacker may choose which signatures it obtains.

Zero signing queries are allowed. Replaying a signature on a queried message does not win this game.

[BS23, Attack Game 13.1, pp. 529–530; Def. 13.2, p. 530].

The first attempt: textbook RSA signatures

Use the RSA key pair $\mathsf{pk} = (N, e)$ and $\mathsf{sk} = d$. First take $\mathcal{M}_{\mathsf{pk}} = \mathbb{Z}_N^*$.

Signing

Input: $m \in \mathbb{Z}_N^*$.

$$\boxed{\mathsf{Sign}_d(m) = m^d \bmod N.}$$

The signer computes an RSA root of the message.

Verification

Reject inputs outside the specified domains. Otherwise accept exactly when

$$\boxed{\sigma^e \equiv m \pmod{N}.}$$

Verification uses only the public key.

The source uses n-bit messages instead. We analyze that domain explicitly after the full-domain attack.

[HPS14, Table 4.1, p. 197]; [BS23, §13.3.1, p. 537]; source p. 7.

Correctness: public verification undoes signing

Proof idea. Write $ed = 1 + k\varphi(N)$ and apply Euler's theorem.

Proof for unit messages

For every $m \in \mathbb{Z}_N^*$,

$$\begin{aligned}\big(\mathrm{Sign}_d(m)\big)^e &\equiv (m^d)^e = m^{ed} \\ &= m\big(m^{\varphi(N)}\big)^k \equiv m \pmod{N}.\end{aligned}$$

Thus every honestly generated signature passes verification.

Correctness analyzes the signer's output. Security must also analyze an attacker's output.

The appendix gives the CRT proof for zero and nonunits as well.

[HPS14, Prop. 3.5, pp. 120–121; Table 4.1, p. 197].

A worked key and an honest signature

Take two 4-bit primes $p = 11$, $q = 13$. Then

$$N = 143, \qquad \varphi(N) = 120, \qquad e = 7, \qquad d = 103,$$

since $7 \cdot 103 = 721 = 1 + 6 \cdot 120$.

Alice signs $m = 2$

$$\sigma = 2^{103} \bmod 143 = 63.$$

She publishes $(m, \sigma) = (2, 63)$.

Bob verifies

$$63^7 \bmod 143 = 2.$$

The public check accepts.

These small parameters illustrate the arithmetic only. They have no cryptographic security.

Worked instance of [HPS14, Table 4.1 and Example 4.5, pp. 197–198].

The forgery: choose the signature first

The attacker can choose the message after choosing the signature.

Proof idea. Evaluate the public RSA map and use its output as the message.

A no-query existential forgery on $\mathcal{M}_{\mathsf{pk}} = \mathbb{Z}_N^*$

Choose any $\sigma^* \in \mathbb{Z}_N^*$ and compute

$$m^* := (\sigma^*)^e \bmod N.$$

Output (m^*, σ^*). Verification accepts by construction.

There were no signing queries, so the message is fresh. The attack succeeds with probability 1 and uses no secret information.

The attacker has limited control over m^*. EUF-CMA allows it to output any fresh legal message.

Source p. 9 [Maj26]; [BS23, §13.3, p. 536].

The stated message domain must also be checked

The source's first scheme signs $m \in \{0,1\}^n$, identified with $0 \le m < 2^n$.

An arbitrary RSA image may be too large

With $N = 143$ and $e = 7$, choosing $\sigma = 2$ gives

$$\sigma^e \bmod N = 128.$$

This is not a 4-bit message. The full-domain attack needs a domain check here.

A valid no-query forgery still exists

Choose $(m^*, \sigma^*) = (1, 1)$. Then m^* is a legal n-bit encoding and

$$(\sigma^*)^e = 1 = m^*.$$

The source's restricted-message scheme is also insecure.

Domain in source p. 7 and attack in p. 9 [Maj26]; game: [BS23, Def. 13.2].

A chosen-message attack from multiplicativity

Proof idea. Multiply the two verification equations.

If $\sigma_1^e \equiv m_1$ and $\sigma_2^e \equiv m_2 \pmod N$, then

$$(\sigma_1\sigma_2)^e \equiv m_1 m_2 \pmod N.$$

An explicit fresh-message forgery

Request signatures on $m_1 = 2$ and $m_2 = 3$. Output

$$m^* = 6, \qquad \sigma^* = \sigma_1\sigma_2 \bmod N.$$

For $n \geq 4$, all three messages belong to both the source's n-bit domain and $\mathbb{Z}_N^*$, and 6 is fresh.

In our example, $\sigma_1 = 63$, $\sigma_2 = 16$, so

$$\sigma^* = 63 \cdot 16 \bmod 143 = 7, \qquad 7^7 \bmod 143 = 6.$$

RSA multiplicativity: [BS23, §13.3.1, p. 537]; the displayed attack is a direct calculation.

Why these forgeries do not invert random RSA images

What the RSA assumption asks

Given a generated public key and a uniform random image y, find x with $x^e \equiv y \pmod{N}$.

What our attacks do

No-query forgery: choose σ first and compute its image.

Multiplicative forgery: use roots supplied by the signing oracle to derive a root of a related message.

A one-way permutation can be hard to invert on random inputs and still yield an insecure signature scheme. Signature security must account for the attacker's choice of message and access to signatures.

[BS23, §13.3, pp. 534–537]; [HPS14, §8.6, pp. 482–485].

The source's next idea: constrain the accepted image

Instead of signing the bare integer m, sign an encoding $E(m)$ and check

$$\sigma^e \equiv E(m) \pmod{N}.$$

Why encoding changes the first attack

Choosing σ still gives $y = \sigma^e \bmod N$. To turn this into a forgery, the attacker must find an allowed m with $E(m) = y$.

The source's proposal

Choose a random prefix r once during key generation, publish it, and accept only encodings whose prefix equals r.

The prefix is public and fixed across signatures. It is not fresh signing randomness.

Source motivation: [Maj26, pp. 10–12].

Fixed-prefix encoding: the bits and the integer

Let $n \geq 4$ be even and put $t = n/2$. Choose $r \in \{0,1\}^t$ and define

$$E_r : \{0,1\}^t \longrightarrow \mathbb{Z}_N, \qquad E_r(m) = r \parallel m = 2^t r + m.$$



The valid images form the interval

$$I_r = \{r2^t, \ldots, r2^t + 2^t - 1\}, \qquad |I_r| = 2^t.$$

Since $E_r(m) < 2^n < N$, encoding has no modular ambiguity.

Integer form of the source encoding, [Maj26, p. 10].

Fixed-prefix candidate: key generation

$\mathsf{Gen}(1^n)$, for even $n \geq 4$

1. Sample distinct odd n-bit primes p, q.
2. Set $N = pq$ and $\varphi(N) = (p-1)(q-1)$.
3. Sample $e \xleftarrow{\$} \mathbb{Z}^*_{\varphi(N)}$ and compute $d = e^{-1} \bmod \varphi(N)$.
4. Sample $r \xleftarrow{\$} \{0,1\}^{n/2}$.
5. Return $\mathsf{pk} = (n, N, e, r)$ and $\mathsf{sk} = d$.

What is public and what is secret?

Publish n, N, e, r. Keep $d, p, q, \varphi(N)$ secret.

The prefix r is sampled once per key.

Source algorithm: [Maj26, p. 11]; RSA setup: [HPS14, §3.2].

Fixed-prefix candidate: signing and verification

$\mathrm{Sign}_d(m)$

Require $m \in \{0,1\}^t$, where $t = n/2$.

Compute $x = E_r(m) = 2^t r + m$.

$$\boxed{\sigma = x^d \bmod N.}$$

Return σ as an integer in $[0, N-1]$.

$\mathrm{Ver}_{\mathrm{pk}}(m, \sigma)$

Require $m \in \{0,1\}^t$ and integer $0 \le \sigma < N$; otherwise reject.

Compute $y = \sigma^e \bmod N$.

$$\boxed{\text{Accept iff } y = 2^t r + m.}$$

Compare the *entire* encoded integer.

This scheme is deterministic after key generation. Repeated signing of the same message produces the same signature.

Source algorithms: [Maj26, p. 12].

Correctness of the fixed-prefix candidate

Proof idea. Apply RSA inversion to the encoded residue, then use the fact that the encoding is smaller than N.

Correctness proof

Let $x = E_r(m)$. RSA correctness on all of $\mathbb{Z}_N$ gives

$$\sigma^e \equiv (x^d)^e \equiv x \pmod{N}.$$

Since both the canonical value $\sigma^e \bmod N$ and x lie in $\{0, \ldots, N-1\}$, they are equal as integers. The verifier accepts.

Why we use the full-residue statement

An encoding can equal 0, p, or q. Euler's theorem applies only to units; the CRT proof in the appendix also covers these encodings.

[HPS14, Prop. 3.5, pp. 120–121; CRT, Thm. 2.24, pp. 84–85].

The same key, now with an encoded message

Continue with $N = 143$, $e = 7$, $d = 103$, $n = 4$, and $t = 2$. Choose the public prefix $r = 10_2 = 2$ and sign $m = 01_2 = 1$.

$$E_r(m) = 10 \,\|\, 01 = 1001_2 = 9.$$

An honest signature

$$\sigma = 9^{103} \bmod 143 = 113.$$

Verification recovers

$$113^7 \bmod 143 = 9 = 1001_2.$$

The prefix and the message both match.

A guessed signature

Choose $\sigma^* = 7$. Then

$$7^7 \bmod 143 = 6 = 0110_2.$$

The required prefix is 10_2, so no allowed message makes this guess valid.

Worked instance of the source candidate [Maj26, pp. 10–12]; arithmetic checked independently.

Exactly what uniform guessing tells us

Fix any generated key and its prefix r. Sample $\sigma \xleftarrow{\$} \mathbb{Z}_N$ and set $y = \sigma^e \bmod N$.

Proof idea. RSA permutes all N residues. Count the 2^t images with the required prefix.

A bound for this specific no-query attack

Because y is uniform on $\mathbb{Z}_N$,

$$\Pr[y \in I_r] = \frac{|I_r|}{N} = \boxed{\frac{2^t}{N}}.$$

If $y \in I_r$, output $m = y - r2^t$ and the guessed signature σ.

Since $p, q \geq 2^{n-1}$ and $t = n/2$, this is at most $2^{2-3n/2}$.

For the toy key it is $4/143$. The union bound gives success at most $Q\,2^t/N$ for Q independent uniform guesses.

Counting argument for the source candidate; RSA permutation: [HPS14, Prop. 3.5].

Uniform guessing is only one possible attack

What we established for the prefix candidate

Honest signatures verify. A uniformly guessed signature rarely lands in the required interval for large parameters.

What a security theorem would still have to show

Every efficient attacker has negligible success, including attackers that choose signatures nonuniformly, exploit the arithmetic of I_r, or request signatures adaptively.

The simple multiplication attack also needs a new analysis: $E_r(m_1)E_r(m_2) \bmod N$ need not be an allowed encoding.

Status. We retain the source's fixed-prefix construction as a candidate. Neither the guessing calculation nor the RSA assumption alone supplies a proof of its EUF-CMA security here.

Source proposal: [Maj26, pp. 10–12]; proof standard: [BS23, §13.3; §13.6].

Looking ahead: hash the message into the RSA domain

For arbitrary-length messages, consider an ideal hash function

$$H : \{0,1\}^* \longrightarrow \mathbb{Z}_N^*.$$

RSA full-domain hash: signing

$$\boxed{\sigma = H(m)^d \bmod N.}$$

Hash the entire message and take the RSA root of its image.

Verification

Require $\sigma \in \mathbb{Z}_N^*$ and accept iff

$$\boxed{\sigma^e \equiv H(m) \pmod{N}.}$$

Choosing σ first now leaves the attacker with the task of finding a message whose hash equals σ^e. The random-oracle theorem handles chosen-message attacks as well.

This is a brief preview of the source's next topic: signing arbitrary-length messages.

[BR96, §3, preprint p. 8]; [BS23, §13.3, pp. 534–538].

The security theorem includes a model

Imported theorem: RSA full-domain hash

Under the RSA inversion assumption, full-domain hash signatures are EUF-CMA secure in the **random-oracle model**, where H is a uniform random function into $\mathbb{Z}_N^*$ available to all algorithms.

Proof roadmap. A reduction programs ordinary hash values as s^e for known s, so it can answer signing queries. It embeds an RSA challenge in one target hash value; a successful fresh forgery at that target gives the challenge's root.

A connection to standardized RSA signatures

RSA-PSS uses a specified hash-based, salted encoding. Its signing and verification procedures are in RFC 8017, §8.1 and §9.1. It is a different construction from a fixed public prefix.

The theorem's ideal-hash assumption is explicit. Merely inserting an arbitrary hash function is not a proof in the standard model.

[BR96, Thm. 3.1, preprint p. 9]; [MKJR16, §8.1, pp. 32–35; §9.1, pp. 40–45].

The lecture in three equations

The honest signer takes an RSA root

$$\sigma = m^d \pmod N \qquad \Longrightarrow \qquad \sigma^e = m \pmod N.$$

The attacker can choose a root before choosing its image

$$\sigma^* \text{ chosen first}, \qquad m^* = (\sigma^*)^e \bmod N.$$

An encoding changes what the verifier accepts

$$\sigma^e = E_r(m) \quad \text{or} \quad \sigma^e = H(m) \pmod N.$$

The fixed-prefix candidate needs further analysis. Full-domain hash has a random-oracle security theorem.

Source: [Maj26, pp. 7–12]; analyzed extension: [BR96, Thm. 3.1].

Check your understanding

1. Why does a forgery on an attacker-chosen message count as a security failure?
2. If an attacker obtains raw RSA signatures on 2 and 3, what fresh message can it sign?
3. In the prefix candidate, is r secret? Is it sampled again for each signature?
4. Does a small success probability for uniform guessing prove EUF-CMA security?
5. Does deterministic signing automatically imply insecurity?

Answers. (1) Existential security allows any fresh legal message. (2) 6, by multiplying signatures. (3) r is public and fixed with the key. (4) No; the game quantifies over all efficient chosen-message attackers. (5) No; full-domain hash is deterministic once the key and hash oracle are fixed.

[BS23, Attack Game 13.1; §13.3]; [BR96, Thm. 3.1].

Appendix: RSA correctness on every residue

Proof idea. Prove the identity modulo each prime separately, then apply the Chinese remainder theorem.

Let $ed = 1 + k(p-1)(q-1)$ and take any integer x.

- If $p \mid x$, then $x^{ed} \equiv 0 \equiv x \pmod{p}$.
- If $p \nmid x$, Fermat's theorem gives

$$x^{ed} = x\left(x^{p-1}\right)^{k(q-1)} \equiv x \pmod{p}.$$

The same argument applies modulo q. Since $p \neq q$, CRT gives

$$\boxed{x^{ed} \equiv x \pmod{pq} \quad \text{for every } x \in \mathbb{Z}_N.}$$

Thus $x \mapsto x^e$ is a permutation of *all* residues, with inverse $x \mapsto x^d$. This justifies both prefix correctness and the exact guessing count.

[HPS14, Prop. 3.5 and eq. (3.3), pp. 120–121; Thm. 2.24, pp. 84–85].

Reading guide

Hoffstein–Pipher–Silverman, second edition [HPS14]

- §3.2, pp. 119–125: RSA and its arithmetic correctness.
- §4.1, pp. 193–196: what a digital signature means.
- §4.2, pp. 196–198: RSA signatures, Table 4.1, Example 4.5.

Boneh–Shoup, version 0.6 [BS23]

- §13.1, pp. 528–530: algorithms, correctness, and the forgery game.
- §13.3, pp. 534–538: full-domain hashing and raw RSA attacks.
- §13.6, pp. 547–550: why small-range hashing needs more care.

Further sources. Bellare–Rogaway, §3 and Thm. 3.1 [BR96], for the unit-domain FDH theorem. RFC 8017, §8.1 and §9.1 [MKJR16], for RSA-PSS.

All textbook locators use printed page numbers. Bellare–Rogaway locators refer to the linked authors' preprint.

References I

[BR96] Mihir Bellare and Phillip Rogaway.
The exact security of digital signatures: How to sign with RSA and Rabin.
In Ueli M. Maurer, editor, *Advances in Cryptology – EUROCRYPT '96*, volume 1070 of *Lecture Notes in Computer Science*, pages 399–416, Berlin, Heidelberg, 1996. Springer.
Slide locators refer to the authors' preprint; construction in Section 3, p. 8; Theorem 3.1, p. 9.
URL: https://web.cs.ucdavis.edu/~rogaway/papers/exact.pdf, doi:10.1007/3-540-68339-9_34.

[BS23] Dan Boneh and Victor Shoup.
A graduate course in applied cryptography, 2023.
Version 0.6, January 2023.
URL: https://toc.cryptobook.us/book.pdf.

[HPS14] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman.
***An Introduction to Mathematical Cryptography*.**
Undergraduate Texts in Mathematics. Springer, New York, second edition, 2014.
doi:10.1007/978-1-4939-1711-2.

References II

[Maj26] **Hemanta K. Maji.**
CS355, Lecture 24a: Digital Signatures using RSA Assumption.
Instructor's lecture slides, Purdue University, 2026.
Supplied source file 24a.pdf, 12 slides.

[MKJR16] **Kathleen Moriarty, Burt Kaliski, Jakob Jonsson, and Andreas Rusch.**
PKCS #1: RSA Cryptography Specifications Version 2.2.
RFC 8017, RFC Editor, November 2016.
Kathleen Moriarty, editor; Sections 8.1 and 9.1 specify RSASSA-PSS and EMSA-PSS.
URL: https://www.rfc-editor.org/rfc/rfc8017, doi:10.17487/RFC8017.

[RSA78] **Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman.**
A method for obtaining digital signatures and public-key cryptosystems.
Communications of the ACM, 21(2):120–126, February 1978.
doi:10.1145/359340.359342.

Technical inputs: arithmetic and the RSA assumption

Arithmetic used in the proofs

Euler: $x^{\varphi(N)} \equiv 1 \pmod{N}$ for $x \in \mathbb{Z}_N^*$.
Fermat: $x^{p-1} \equiv 1 \pmod{p}$ for $p \nmid x$.
CRT: congruence modulo distinct primes p, q determines a unique residue modulo pq.

Precise sources: HPS, Prop. 3.5 and eq. (3.3), pp. 120–121, for their RSA application; Thm. 2.24, pp. 84–85, for CRT.

Assumed RSA inversion hardness for the stated Gen

Sample p, q, e as on the key-generation slide, and $x \xleftarrow{\$} \mathbb{Z}_N^*$. For every efficient A,

$$\Pr[A(n, N, e, x^e \bmod N) = x] \leq \mathrm{negl}(n).$$

The probability includes the generated key, x, and the attacker's coins. This is a computational assumption, not an arithmetic theorem.

[BS23, Attack Game 10.3, Def. 10.5, pp. 404–405]; [HPS14, §3.2].

Technical input: the theorem used in the FDH preview

Imported theorem: Bellare–Rogaway FDH security

Assume RSA inversion is hard for the key distribution in use. Let $H : \{0,1\}^* \to \mathbb{Z}_N^*$ be an independent ideal random oracle, conditional on the public key. Then

$$\mathrm{Sign}_d(m) = H(m)^d \bmod N, \qquad \mathrm{Ver}_{N,e}(m,\sigma) = [\sigma^e \equiv H(m) \pmod N]$$

with unit-domain validation is EUF-CMA secure for polynomially many signing and random-oracle queries.

Exact source. Bellare–Rogaway, *The Exact Security of Digital Signatures*, authors' preprint: construction in §3, p. 8; Theorem 3.1, p. 9. Their quantitative reduction loses a factor $q_{\mathrm{sig}} + q_{\mathrm{hash}} + 1$.

This theorem applies to the FDH preview. The preceding correctness proofs and raw-RSA forgeries require no hardness assumption.

[BR96, §3, pp. 8–9, Thm. 3.1]; compare [BS23, Thm. 13.3, p. 535].