RSA Digital Signature Standards

Burt Kaliski, RSA Laboratories

23rd National Information Systems Security Conference, October 16–19, 2000

RSA
LABORATORIES

Outline

I. Background

II. Forgery and provable security

III. Contemporary signature schemes

IV. Standards strategy

Part I: Background

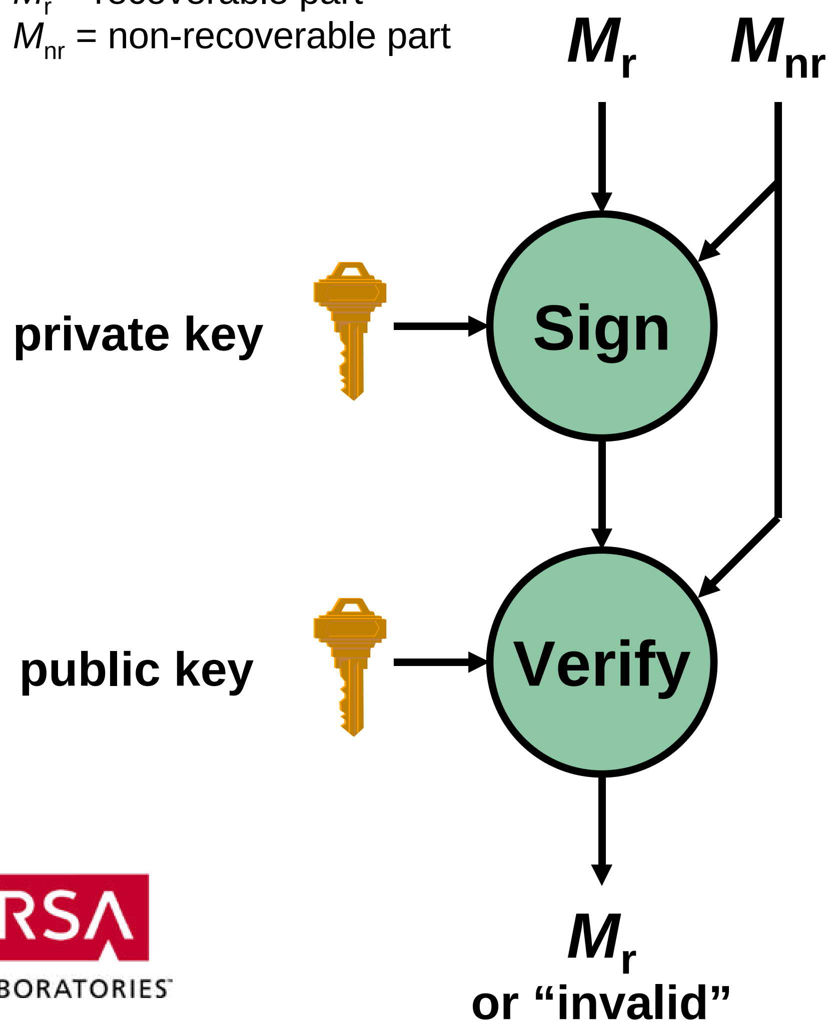
General Model

- A signature scheme consists of three (or more) related operations
- *Key pair generation* produces a public/private key pair
- *Signature operation* produces a signature for a message with a private key
- *Verification operation* checks a signature with a public key

RSA LABORATORIES

Types of Signature Scheme

M_r = recoverable part
M_{nr} = non-recoverable part



- ***Appendix:*** **message transmitted with signature**
- ***Total message recovery:*** **message recoverable from signature**
- ***Partial message recovery:*** **part of message recoverable from signature, part transmitted**

RSA LABORATORIES

Trapdoor One-Way Functions

- A *one-way function* $f(x)$ is easy to compute but hard to invert:
  - easy: $x \rightarrow f(x)$
  - hard: $f(x) \rightarrow x$
- A *trapdoor* one-way function has trapdoor information f^{-1} that makes it easy to invert:
  - easy: $f(x), f^{-1} \rightarrow x = f^{-1}(f(x))$
- Many but not all signature schemes are based on trapdoor OWFs

RSA LABORATORIES

RSA Trapdoor OWF

- The RSA function is

$$f(x) = x^e \bmod n$$

where $n = pq$, p and q are large random primes, and e is relatively prime to p-1 and q-1

- This function is conjectured to be a trapdoor OWF
- Trapdoor is

$$f^{-1}(x) = x^d \bmod n$$

where $d = e^{-1} \bmod \mathrm{lcm}(p-1, q-1)$

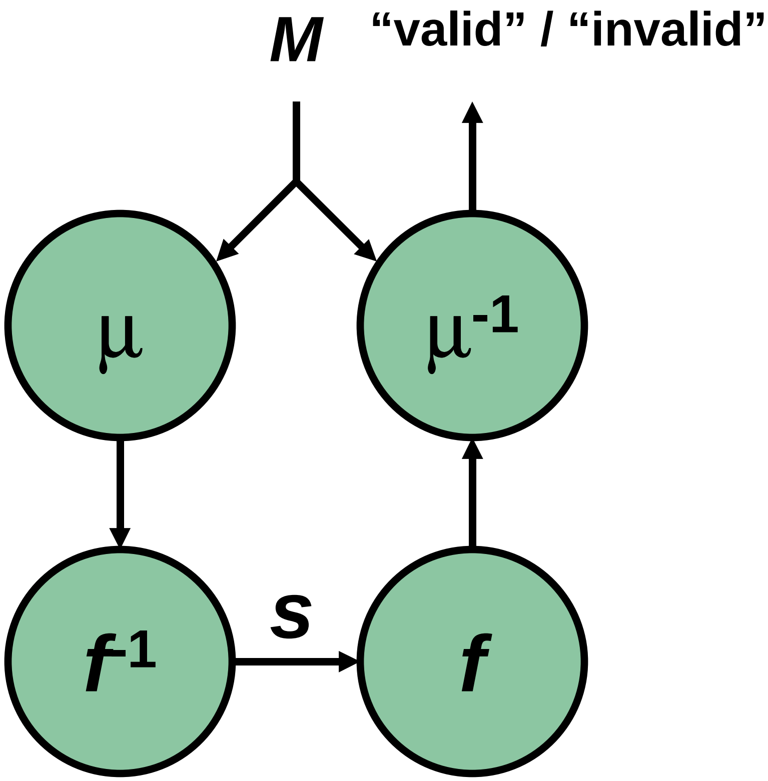
RSA LABORATORIES

Embedding Operations

- An embedding operation $\mu(M)$ maps from message strings to "message representatives," which can be input to f^{-1}
  - e.g., a hash function with padding
  - may be randomized
- Inverse operation checks whether a message representative is correct
  - in scheme with message recovery, also recovers message part
- Current RSA signature schemes differ primarily in terms of the embedding operation

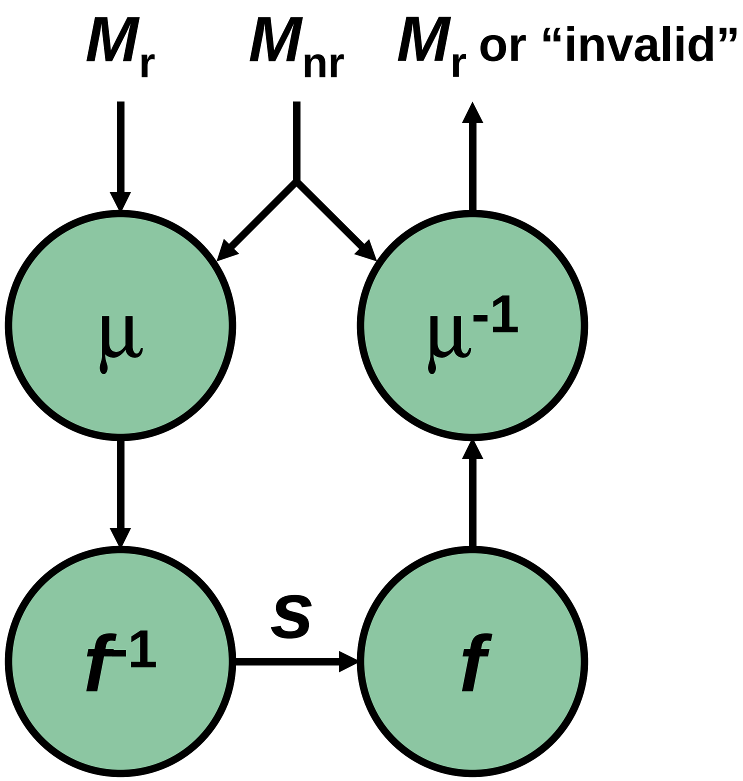
RSA LABORATORIES

Scheme with Appendix



- Signature generation embeds message, applies trapdoor:
  - $s = f^{-1}(\mu(M))$
- Signature verification applies OWF, checks against message:
  - $\mu^{-1}(f(s), M)$ valid?

Scheme with Message Recovery



- Signature generation embeds message, applies trapdoor:
  - $s = f^{-1}(\mu(M_r, M_{nr}))$
- Signature verification applies OWF, checks against M_{nr}, recovers M_r:
  - $M_r = \mu^{-1}(f(s), M_{nr})$

RSA LABORATORIES

Embedding Properties

- **Embedding operation should have similar properties to a hash function:**
  - one-way: for random x, hard to find M s.t. $\mu(M) = x$
  - collision-resistant: hard to find M_1, M_2 s.t. $\mu(M_1) = \mu(M_2)$
- **May also identify underlying algorithms**
  - but if so, must be done with care
- **Should also interact well with trapdoor function**
  - ideally, mapping should appear "random"

RSA LABORATORIES

Multiplicative Properties of RSA

- **RSA function is a *multiplicative homomorphism:* for all x, y,**

$$f\,(xy \bmod n) = f(x)\,f(y) \bmod n$$

$$f^{-1}(xy \bmod n) = f^{-1}(x)\,f^{-1}(y) \bmod n$$

- **More generally:**

$$f^{-1}\left(\prod x_i \bmod n\right) = \prod \left(f^{-1}(x_i)\right) \bmod n$$

- **Property is exploited in most forgery attacks on RSA signatures, but also enhances recent security proofs**

RSA LABORATORIES

Part II: Forgery and Provable Security

RSA LABORATORIES

Signature Forgery

- **A *forgery* is a signature computed without the signer's private key**
- **Forgery attacks may involve interaction with the signer: a *chosen-message* attack**
- **Forgery may produce a signature for a specified message, or the message may be output with its signature (*existential forgery*)**

RSA LABORATORIES™

Multiplicative Forgery

- **Based on the multiplicative properties of the RSA function, if**

$$\mu(M) = \prod \mu(M_i)^\wedge\alpha_i \bmod n$$

then

$$\sigma(M) = \prod \sigma(M_i)^\wedge\alpha_i \bmod n$$

- **Signature for *M* can thus be forged given the signatures for $M_1, \ldots, M_l$ under a chosen-message attack**

Small Primes Method

- Suppose $\mu(M)$ and $\mu(M_1)$, …, $\mu(M_l)$ can be factored into small primes
  - Desmedt-Odlyzko (1986); Rivest (1991 in PKCS #1)
- Then the exponents α_i can be determined by relationships among the prime factorizations
- Requires many messages if μ maps to large integers, but effective if μ maps to small integers
- Limited applicability to current schemes

Recent Generalization

- **Consider $\mu(M)$, $\mu(M_1)$, …, $\mu(M_l)$ mod n, and also allow a fixed factor**
  - Coron-Naccache-Stern (1999)
- **Effective if μ maps to small integers mod n times a fixed factor**
- **Broader applicability to current schemes:**
  - ISO 9796-2 [CNS99]
  - ISO 9796-1 [Coppersmith-Halevi-Jutla (1999)]
  - recovery of private key for Rabin-Williams variants [Joye-Quisquater (1999)]

RSA LABORATORIES

Integer Relations Method

- **What if the equation**

$$\mu(M) = f(t) \prod \mu(M_i)\text{^}\alpha_i$$

  could be solved without factoring?

- **Effective for *weak* μ**
- **ISO 9796-1 broken with *three* chosen messages [Grieu (1999)]**

RSA LABORATORIES

Reduction Proofs

- A *reduction proof* shows that inverting the function *f* "reduces" to signature forgery: given a forgery algorithm *F*, one can construct an inversion algorithm *I*
- *Provable security*: inversion hard $\rightarrow$ forgery hard
- "Tight" proof closely relates hardness of problems

RSA LABORATORIES

Random Oracle Model

- **In the *random oracle* model, certain functions are considered “black boxes”: forgery algorithm cannot look inside**
  - **e.g., hash functions**
- **Model enables reduction proofs for generic forgery algorithms — inversion algorithm hides value to be inverted in oracle outputs**
- **Multiplicative properties of RSA can enhance the proof**

RSA LABORATORIES

Part III: Contemporary Signature Schemes

RSA LABORATORIES

Overview

- Several popular approaches to RSA signatures
- Approaches differ primarily in the mapping μ
- Some differences also in key generation
- Some also support Rabin-Williams (even exponent) signatures

- There are many other signature schemes based on factoring (e.g., Fiat-Shamir, GQ, Micali, GQ2); focus here is on those involving the RSA function

Schemes with Appendix

- Basic scheme
- ANSI X9.31
- PKCS #1 v1.5
- Bellare-Rogaway FDH
- Bellare-Rogaway PSS
- IEEE P1363a version of PSS

Basic Scheme

- $\mu(M) = \text{Hash}(M)$
- Pedagogical design
- Insecure against multiplicative forgery for typical hash sizes
- (Hopefully) not widely deployed

ANSI X9.31

(Digital Signatures Using Reversible Public-Key Cryptography for the Financial Services Industry, 1998)

- $\mu(M)$ = 6b bb … bb ba || Hash(M) || 3x cc

  where x = 3 for SHA-1, 1 for RIPEMD-160
- Ad hoc design
  - cc octet for RW support
- Resistant to multiplicative forgery
  - some moduli are more at risk, but still out of range
- Widely standardized
  - IEEE 1363, ISO/IEC 14888-3
  - US NIST FIPS 186-1
- ANSI X9.31 requires “strong primes”

RSA LABORATORIES

PKCS #1 v1.5

(RSA Encryption Standard, 1991)

- $\mu(M)$ = 00 01 ff … ff 00 || HashAlgID || Hash(M)
- Ad hoc design
- Resistant to multiplicative forgery
  - moduli near 2^k are more at risk, but still out of range
- Widely deployed
  - SSL certificates
  - S/MIME
- Included in IEEE P1363a; PKCS #1 v2.0 continues to support it

RSA LABORATORIES

ANSI X9.31 vs. PKCS #1 v1.5

- Both are deterministic
- Both include a hash function identifier
- Both are ad hoc designs
  - both resist [CNS99]/[CHJ99] attacks
- Both support RSA and RW primitives
  - see IEEE P1363a contribution on PKCS #1 signatures for discussion
- No patents have been reported to IEEE P1363 or ANSI X9.31 for these mappings

RSA LABORATORIES

Bellare-Rogaway FDH

(Full Domain Hashing, ACM CCCS ’93)

- $\mu(M)$ = Full-Length-Hash(m)
- Provably secure design
  - resists any attack where hash function is considered a black box, provided that RSA is hard to invert
- Variant included in IEEE P1363a, PKCS #1 v2.1 draft

RSA LABORATORIES

Bellare-Rogaway PSS

(Probabilistic Signature Scheme, Eurocrypt '96)

- $\mu(M) \approx H \,||\, G(H) \oplus salt$

  where H = Hash($salt$, M), $salt$ is random, and G is a mask generation function

- **Provably secure design**
- **Variant included in IEEE P1363a, PKCS #1 v2.1 draft**

RSA LABORATORIES

FDH vs. PSS

- FDH is deterministic, PSS is probabilistic
- Both are provably secure designs
  - same paradigm as Optimal Asymmetric Encryption Padding (OAEP)
- PSS has tighter security proof, is less dependent on security of hash function
- PSS-R variant supports message recovery, partial message recovery
- PSS is patent pending (but generously licensed)

RSA LABORATORIES™

IEEE P1363a Version of PSS

- $\mu(M) = G(H) \oplus [00 \ldots 01 \,||\, salt] \,||\, H \,||\, \mathrm{bc}$

  where $H \approx \mathrm{Hash}(salt, \mathrm{Hash}(M))$, *salt* is random, and G is a mask generation function

- Salt combined with Hash(*M*) rather than *M* for practical and security reasons:
  - "single-pass" processing
  - provable security if Hash(*M*) outside crypto module
  - protection against fault-analysis attacks

- Salt can be omitted for FDH-like scheme

Schemes with Message Recovery

- Basic scheme
- ISO/IEC 9796-1
- ISO/IEC 9796-2
- Bellare-Rogaway PSS-R
- IEEE P1363a version of PSS-R

Basic Scheme

- $\mu(M_r) = M_r$
- Another pedagogical design (“textbook RSA”)
- Insecure against various forgeries, including existential forgery
  - attacker can select signature s then “recover” $M_r = f(s)$
- Again, hopefully not widely deployed

RSA LABORATORIES

ISO/IEC 9796-1

(Digital Signature Scheme Giving Message Recovery, 1991)

- $$\mu(M_r) = \pi^*(m_{l-1})\ \pi'(m_{l-2})\ m_{l-1}\ m_{l-2}\ \pi(m_{l-3})\ \pi(m_{l-4})\ m_{l-3}\ m_{l-4} \ldots \pi(m_3)\ \pi(m_2)\ m_3\ m_2\ \pi(m_1)\ \pi(m_0)\ m_0\ 6$$

  where m_i is the ith nibble of M_r and π^*, π' and π are permutations

- Ad hoc design with significant rationale
- *Not* resistant to multiplicative forgery [CHJ99] [Grieu 1999]
  - may still be appropriate if applied to a hash value
- Moderately standardized

RSA LABORATORIES

ISO/IEC 9796-2

(Digital Signature Scheme Giving Message Recovery — Mechanisms Using a Hash Function, 1997)

- $\mu(M_r, M_{nr}) \approx$ 6a || M_r || H || bc
- $\mu(M_r)$ = 4b bb … bb ba || M_r || H || bc

  where H = Hash (M_r, M_{nr}) or Hash (M_r)
  - (assumes modulus length is multiple of 8)
  - general format allows hash algorithm ID
- Ad hoc design
- *Not* resistant to multiplicative forgery if hash value is 64 bits or less [CNS99]
  - may still be appropriate for larger hash values
- Newly standardized

RSA LABORATORIES

Bellare-Rogaway PSS-R

(Probabilistic Signature Scheme with Recovery, 1996)

- $\mu(M_r, M_{nr}) \approx H \ || \ G(H) \oplus [salt \ || \ M_r]$

  where H = Hash($salt$, M_r, M_{nr}), $salt$ is random, and G is a mask generation function

- Provably secure design
- Variant included in IEEE P1363a, draft revision of ISO/IEC 9796-2

RSA LABORATORIES

IEEE P1363a Version of PSS-R

- $\mu(M_r, M_{nr}) = G(H) \oplus [00 \ldots 01 \,||\, M_r \,||\, salt] \,||\, H \,||\, \mathrm{bc}$

  where $H \approx \mathrm{Hash}(salt, M_r, \mathrm{Hash}(M_{nr}))$, *salt* is random, and G is a mask generation function
- Extension of PSS variant
  - PSS variant is special case where M_r is null

RSA LABORATORIES

Part IV: Standards Strategy

RSA LABORATORIES

Standards vs. Theory vs. Practice

- **ANSI X9.31 is widely standardized**
- **PSS is widely considered secure**
- **PKCS #1 v1.5 is widely deployed**

- **How to harmonize signature schemes?**
  - **(primary question for signature schemes with appendix; related question for message recovery)**

RSA LABORATORIES™

Challenges

- **Infrastructure changes take time**
  - **particularly on the user side**
- **ANSI X9.31 is more than just another encoding method, also specifies “strong primes”**
  - **a controversial topic**
- **Many communities involved**
  - **formal standards bodies, IETF, browser vendors, certificate authorities**

RSA LABORATORIES

Prudent Security

- What if a weakness were found in ANSI X9.31 or PKCS #1 v1.5 signatures?
  - no proof of security, though designs are well motivated, supported by analysis
  - would be surprising — but so were vulnerabilities in ISO/IEC 9796-1,-2
- PSS embodies “best practices,” prudent to improve over time

RSA LABORATORIES

Proposed Strategy

- **Short term (1-2 years): Support both PKCS #1 v1.5 and ANSI X9.31 signatures for interoperability**
  - e.g., in IETF profiles, FIPS validation
  - FIPS 186-2 schedule allows PKCS #1 v1.5 for an 18-month transition period, FPKI TWG is requesting a further extension
- **Long term (2-5 years): Move toward PSS**
  - upgrade in due course — e.g., with AES algorithm, new hash functions
  - separate *assurance* requirements from interoperability
    - e.g., key sizes, key protection, “strong primes”

RSA LABORATORIES™

Standards Work

- PSS, PSS-R standardization work in progress in various forums:
  - IEEE P1363a
  - PKCS #1 v2.1
  - ISO/IEC 9796-2 revision
- Coordination ongoing, ballot target Spring 2001
- Promotion in other forums planned
  - ANSI X9.31
  - FIPS
  - IETF

RSA LABORATORIES

Conclusions

- Several signature schemes based on RSA algorithm
  - varying attributes: standards, theory, practice
- Recent forgery results on certain schemes, security proofs on others
- PSS a prudent choice for long-term security, harmonization of standards