Admin:

Outline: RSA security; digital signatures

- ☐ RSA security; factoring
- ☐ RSA-OAEP for ACCA security
- ☐ Digital Signatures – definitions & security
- ☐ Signing with RSA
- ☐ El Gamal signatures
- ☐ Digital signature standard

RSA is homomorphic (like El Gamal):

$$E(m_1) \cdot E(m_2) = E(m_1 m_2) = (m_1 m_2)^e \pmod n$$

RSA is not even semantically secure (since it is deterministic)!

Adversary can easily tell $E(m_0)$ from $E(m_1)$ since these values are fixed.

Need to repair this...

Notes on factoring:

- knowing e doesn't help in factoring n (e is random)
- computing d is as hard as factoring n

  knowing $e, d \Rightarrow$ know multiple of $\varphi(n) \Rightarrow$ know p, q

Best factoring algorithms (Number field sieve)

take time
$$\exp\left\{ \text{const} \cdot (\ln(n))^{1/3} \cdot (\ln \ln n)^{2/3} \right\}$$
$$\approx \exp\left\{ k^{1/3} \right\} \quad \text{for } k\text{-bit } n$$

Now: 512-bit #'s can be factored, 1024-bit seem a bit out of reach..

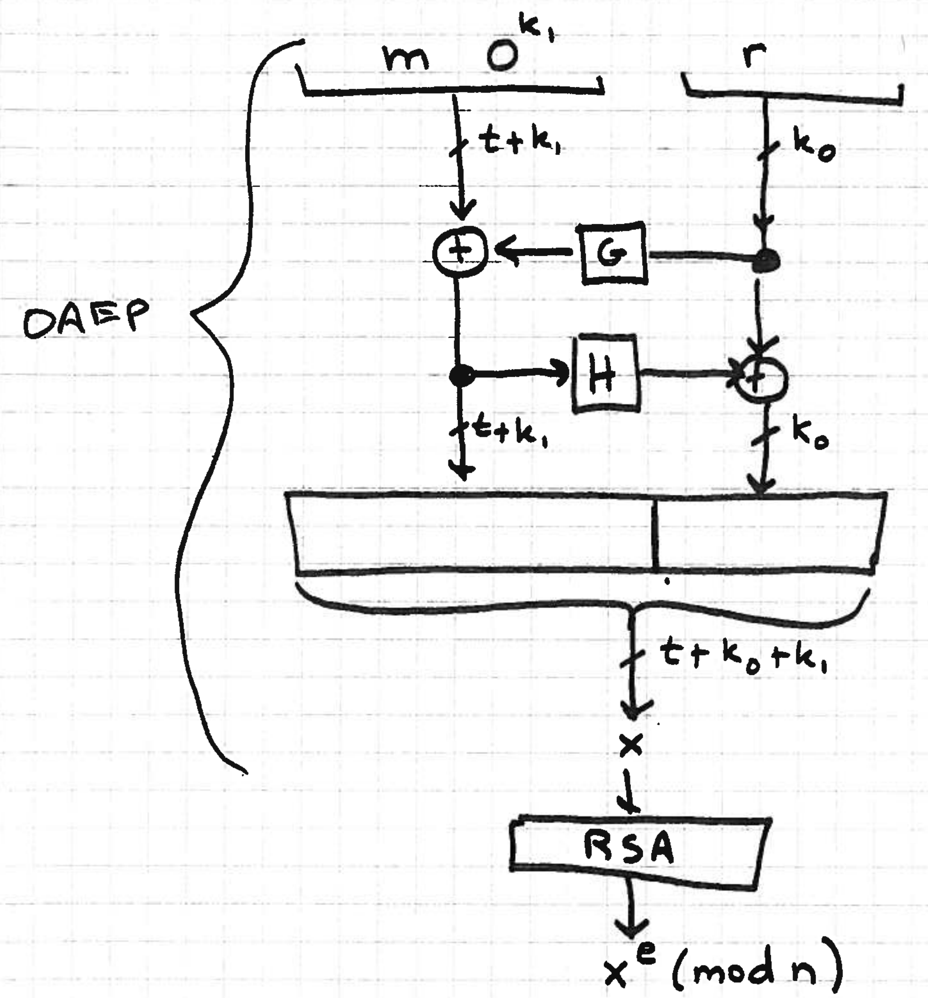
Using n in range 1024... 4096 seems fine... (for now...)

How to make RSA IND-CCA2 secure?

"OAEP" = Optimal asymmetric encryption padding [BR 94]

Given m, $|m| = t$ bits

Pick r at random, $|r| = k_0$



$G: \{0,1\}^{k_0} \rightarrow \{0,1\}^{t+k_1}$

$H: \{0,1\}^{t+k_1} \longrightarrow \{0,1\}^{k_0}$

G, H: "random oracles"

like UFE (!) of Desai

On decryption: invert RSA
invert OAEP
reject if 0^{k_1} not present

Thm: RSA with OAEP secure against ACCA, assuming RO model & that RSA hard to invert on random inputs.

bug in proof, but ok with slightly different assumptions... (or OAEP$^+$)

OAEP: used in practice

theory: (we don't have random oracles...)

Digital Signatures

- Invented by Diffie/Hellman in 1976 (New Directions)
- First implementation: RSA (1977) [key motivation for me for PK... !]
- Initial idea: switch PK/SK – enc with secret key ⇒ sig
  – if PK decrypts it – then sig ok

- Current way of describing digital signatures
  - (Note: law is confused (includes hashes, MACs, etc...) – ignore it
  - **Keygen** $(1^\lambda) \rightarrow (PK, SK)$

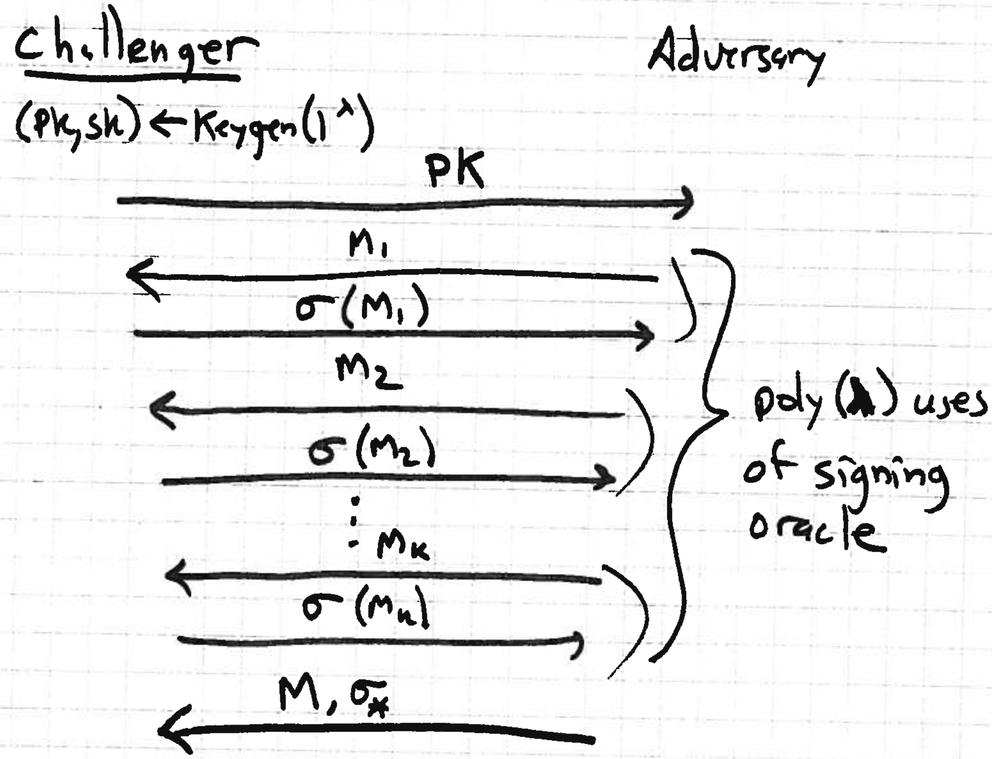
    PK: verification key; SK: signing key

    λ = "security parameter"
    all lengths are polynomial in λ
    security may be negligible fn of λ.
  - ignore for now "PKI" issue:
    knowing that you have "right" PK
  - **Sign** $(SK, M) \longrightarrow \sigma_{SK}(M)$ (may be randomized)
    $M \in \{0,1\}^*$
  - **Verify** (PK, M, σ) = True/False

Correctness: $(\forall M)$ $\underline{\text{Verify}}(PK, M, \underline{\text{Sign}}(SK, M)) = \text{True}$

Security: (Weak) existential unforgeability under adaptive chosen message attack:

Game:



Adv **wins** if $\text{Verify}(PK, M, \sigma_*) = \text{True}$

$$\& \quad M \notin \{M_1, \ldots, M_k\}$$

Scheme is secure (i.e. weakly existentially unforgeable against adaptive chosen message attack)

iff Prob[Adv wins] is negligible (i.e. $\le 1/\lambda^c$ for all c & all suff. large λ)

Scheme is **strongly** secure if adversary can't even produce new sig for ~~previous~~ message previously signed

i.e. Adv wins if $\text{Verify}(PK, M, \sigma_*) = \text{True}$

$$\& \quad (M, \sigma_*) \notin \{(M_1, \sigma_1), (M_2, \sigma_2), \ldots, (M_k, \sigma_k)\}$$

Signing with RSA

① Hash & sign with PKCS

Let $H(M) = SHA256(M)$

Let $H'(M) =$ 0x 00 01 FF FF... FF 00 $\|$ ASN.1 (name of hash) $\|$ $H(M)$

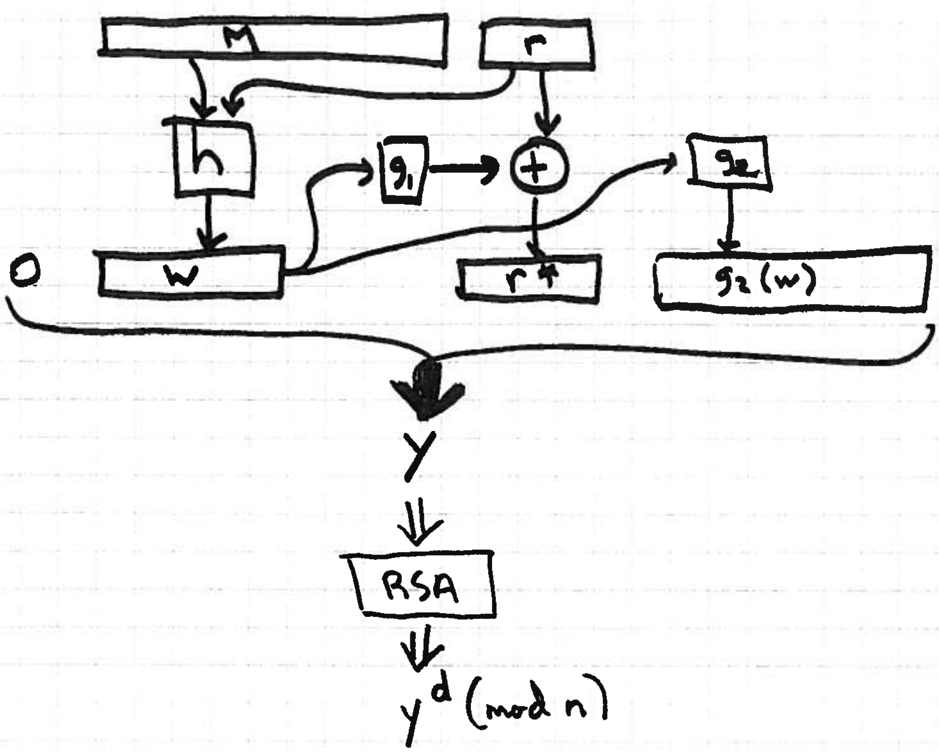
$\sigma(M) = (H'(M))^d \bmod n$

Some problems with $e=3$ (bad implementations
can find 0
parse ASN.1
take $H(M)$
miss other stuff after $H(M)$)

Otherwise seems OK, but no proofs. (even assuming collision resistance & RSA hard to invert...)

Commonly used, none the less...

② PSS [Bellare & Rogaway 1996]



Sign(M):

$r \xleftarrow{R} \{0,1\}^{k_0}$

$w \leftarrow h(M \| r)$ ⟸ note! (compare with El Gamal later) $|w| = k_1$

$r^* \leftarrow g_1(w) \oplus r$ $|r^*| = k_0$

$y \leftarrow 0 \| w \| r^* \| g_2(w)$ $|y| = k = |n|)$

return $y^d \pmod n$

Verify(M, x):

$y \leftarrow x^e \pmod n$

parse y as $b \| w \| r^* \| \gamma$

$r \leftarrow r^* \oplus g_1(w)$

if $h(M \| r) = w$ & $g_2(w) = \gamma$ & $b = 0$

return **True**

else return **False**

Theorem: PSS is (weakly) ~~secure~~ existentially unforgeable against chosen message attack in ROM if RSA is not invertible on random inputs. ($\neg\exists$ Adv who can produce x^d given x.)

El Gamal signatures

Public system parameters p prime

g generator

Keygen: $x \in_R \{0, 1, \ldots, p-2\}$ $SK = x$

$y = g^x$ $PK = y$

Sign(M): $m = h(M)$

$k \in_R Z^*_{p-1}$ $[\gcd(k, p-1) = 1]$

randomized signing

$r = g^k$ [hard work is indep of M]

$ks + rx = m$ $s = \frac{(m - rx)}{k} \pmod{p-1}$

$\sigma(M) = (r, s)$

Verify: check $0 < r < p$

" $y^r r^s = g^m \pmod p$ where $m = h(M)$

Return True if both checks pass else return False

Correctness: $g^{rx} g^{sk} = \underbrace{g^{rx+sk} \stackrel{?}{=} g^m}_{\equiv} \pmod p$

$rx + ks \stackrel{?}{=} m \pmod{p-1}$

$\equiv$

$s = \frac{(m - rx)}{k} \pmod{p-1}$

$(\text{if } \gcd(k, p-1) = 1)$

[El Gamal signatures, cont.d]

That was original version.

Theorem: El Gamal is existentially forgeable (without h fn or h = identity)
↑ note: CR!

Proof: Let $e \in_R Z_{p-1}$

$$r \leftarrow g^e y \pmod{p}$$

$$s \leftarrow -r \pmod{p-1}$$

(r,s) is sig for message $m = es \pmod{p-1}$

$$y^r r^s \stackrel{?}{=} g^m$$

$$g^{xr}(g^e y)^{-r} = g^{-er} = g^{es} = g^m \quad \text{for } m = es \pmod{p-1}$$

But: It is easy to fix.

Modified El Gamal (Pointcheval/Stern 1996)

sign(M): $k \in_R Z_p^*$

$$r = g^k \pmod{p}$$

$$m = h(M \| r) \quad \Longleftarrow ***$$

$$s = \frac{(m - rx)}{k} \bmod p-1$$

$$\sigma(M) = (r,s)$$

Verify: check $0 < r < p$
check $y^r r^s = g^m$ where $m = h(M \| r)$.

<u>Thm</u>: (Modified) El Gamal is existentially unforgeable against adaptive chosen message attack, in ROM, assuming DLP is hard.