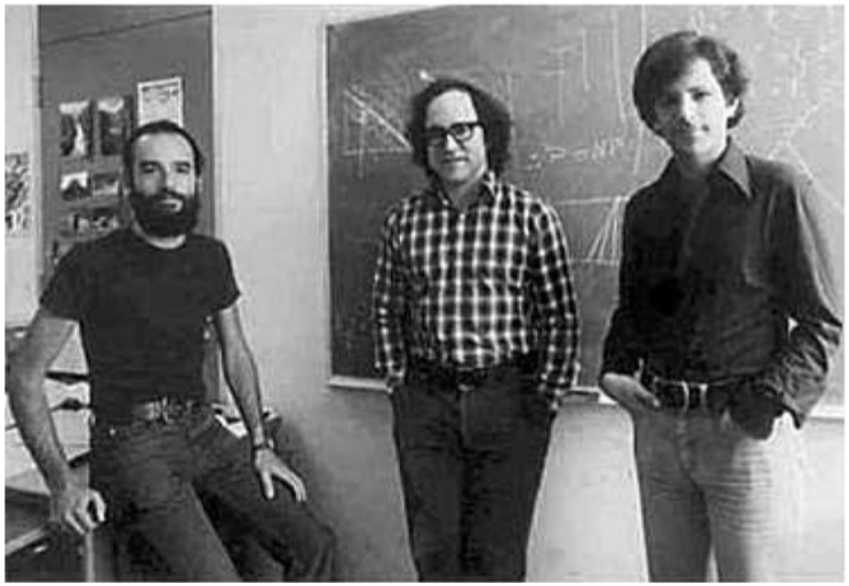
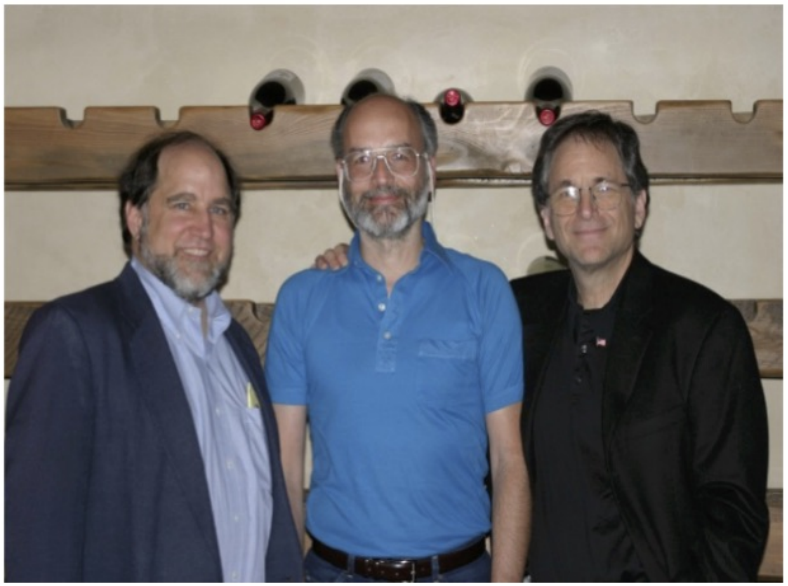
RSA - RECAP





RSA: Rivest, Shamir and Adleman (Turing Award 2003). First published: **Scientific American, Aug. 1977**.

Textbook RSA

Textbook RSA

KeyGen

- Pick two distinct prime numbers p and q
- Compute $\mathbf{N} = pq$ and $\phi(\mathbf{N}) = (p-1)(q-1)$
- Choose $\mathbf{e} \in \mathbb{Z}_{\phi(N)}$ such that $gcd(e, \phi(N)) = 1$
- Find $\mathbf{d} \in \mathbb{Z}_{\phi(N)}$ the modular inverse of e mod $\phi(N)$.
- Set $pk = (e, N)$ as public key, and $sk = (d, \phi(N))$ as secret key.

Enc(pk, m) take $m \in \mathbb{Z}_N^*$ and compute its cipher text $c = \mathbf{Enc}(pk, m) = m^e \mod N$

Dec(sk,c) take $c \in \mathbb{Z}_N^*$ and compute its plain text $m = \mathbf{Dec}(\mathbf{sk}, \mathbf{c}) = c^d \mod N$

Correctness Property:
$\mathbf{Dec(sk,c)} = (c)^d = (\mathbf{Enc(pk,m)})^d = (m^e)^d = m^{ed} = m^{k\phi(N)+1} = (m^{\phi(N)})^k \cdot m = m.$

Remarks on textbook RSA

Facts to remember:

- Textbook RSA encryption is **deterministic**, therefore textbook RSA is not IND-CPA/IND-CCA secure
- In order to find out the plaintext from a ciphertext the adversary should solve the discrete log problem, but if the message space is small (e.g. $N = pq$ is small) then the adversary could simply try out all possible pairs plaintext-ciphertext.
- How to fix these issues? RSA - OAEP.

RSA signatures

RSA signature

KeyGen

- Same as in textbook RSA.
- Set $pk = (e, N)$ as public key, and $sk = (d, \phi(N))$ as secret key.

Sign

- Alice uses her private key d to obtain the signature s on her message m by computing: $s = m^d \mod N$.

Verification

- In order to verify Alice's signature s, Bob uses Alice public key e and checks whether: $m = s^e \mod N$.
  If the equality hold, the Bob trusts that Alice has signed the message m, otherwise the signature s is considered invalid.

Problems with RSA signature

RSA signature is malleable:
Any adversary can construct a valid signature s^* for the message $m_1 * m_2$ by combining Alice's signatures s_1 and s_2 (s_i is a valid signature for message m_i, $i = 1, 2$).

How to avoid this?
Instead of verifying the message, verify the hash of a message!