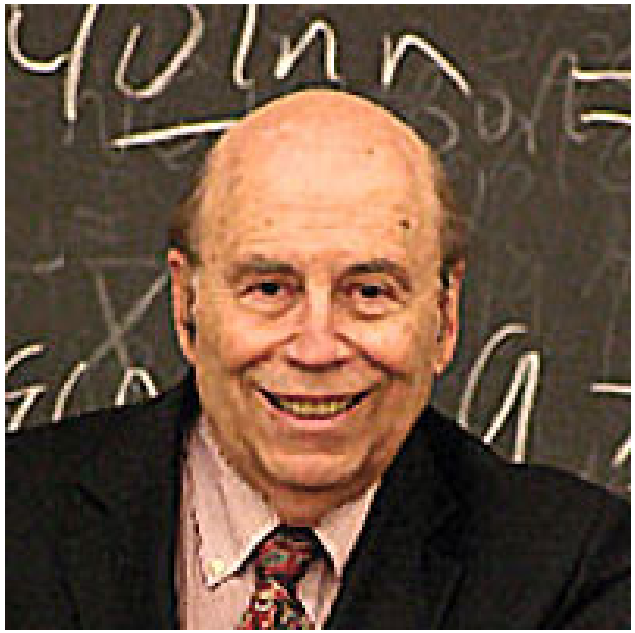
Rabin Algorithm



Well-Known One-Way Functions

- Discrete Logarithm:
  Given p, g, and x, computing y in $y = g^x \pmod{p}$ is EASY
  Given p, g, y, computing x in $y = g^x \pmod{p}$ is HARD
- **Factoring**:
  Given p and q, computing n in $n = p \cdot q$ is EASY
  Given n, computing p or q in $n = p \cdot q$ is HARD
- **Discrete Square Root:**
  Given x and y, computing y in $y = x^2 \pmod{n}$ is EASY
  Given y and n, computing x in $y = x^2 \pmod{n}$ is HARD
- Discrete eth Root:
  Given x, n and e, computing y in $y = x^e \pmod{n}$ is EASY
  Given y, n and e, computing x in $y = x^e \pmod{n}$ is HARD

Rabin Public-Key Encryption

- Michael Oser Rabin is an Israeli computer scientist, born in Wroclaw, Poland, and educated in Hebrew University and Princeton
- He made significant contributions to theory of computer science, particularly, finite automata and regular languages
- He invented a randomized polynomial-time primality testing algorithm together with Gary Miller while at MIT, which is now called Miller-Rabin test
- He also invented the Rabin algorithm, which is the first public-key encryption algorithm whose security was proved equivalent to the intractability of integer factorization

Rabin Public-Key Encryption

- The User generates two large, approximately same size random primes: p and q
- (For simplicity) the primes p and q have the property

$$p = q = 3 \pmod 4$$

- The modulus n is the product of these two primes: $n = pq$
- The public key: The modulus n
- Encryption: Another user obtains the public key n and forms the message m with $m < n$, and computes the ciphertext as

$$c = m^2 \pmod n$$

Rabin Public-Key Decryption

- The private key is the primes p and q, and integers a and b such that

$$a = p^{-1} \pmod{q} \quad \text{and} \quad b = q^{-1} \pmod{p}$$

  which are computed using the extended Euclidean algorithm:
  $a \cdot p + b \cdot q = 1$
- The User gets the ciphertext c, and computes the 4 square roots of c:

$$\begin{aligned} x &= c^{(p+1)/4} \pmod{p} \\ y &= c^{(q+1)/4} \pmod{q} \\ m_1 &= a \cdot p \cdot y + b \cdot q \cdot x \pmod{n} \\ m_2 &= a \cdot p \cdot y - b \cdot q \cdot x \pmod{n} \end{aligned}$$

  The plaintext is one of these 4 values: $m_1, m_2, -m_1, -m_2$:)

Rabin Example

- $p = 11$ and $q = 23$, such that $p = q = 3 \pmod 4$, and $n = 253$
- From $-2 \cdot 11 + 1 \cdot 23 = 1$, the EEA computes $a = -2$ and $b = 1$
- Encryption: $E(100) = 100^2 \pmod{253}$ gives $c = 133$
- Decryption: We compute the plaintext candidates as

$$\begin{aligned} x &= 133^{(11+1)/4} = 133^3 = 1 \pmod{11} \\ y &= 133^{(23+1)/4} = 133^6 = 8 \pmod{23} \\ m_1 &= (-2) \cdot 11 \cdot 8 + 1 \cdot 23 \cdot 1 = 100 \pmod{253} \\ m_2 &= (-2) \cdot 11 \cdot 8 - 1 \cdot 23 \cdot 1 = 54 \pmod{253} \end{aligned}$$

Plaintext candidates: 100, 54, $-100 = 153$, and $-54 = 199$

Security of Rabin Public-Key Encryption

- If we can factor n and find the primes p and q, we obtain the private key and therefore break Rabin public-key encryption:
  Factoring $n \Rightarrow$ **Breaking Rabin**
- On the other hand, if we can take discrete square roots modulo n, and find two square roots x and y of a random c such that

$$x^2 = c \pmod{n} \quad \text{and} \quad y^2 = c \pmod{n}$$

which implies

$$x^2 - y^2 = 0 = (x - y)(x + y) \pmod{n}$$

which gives the factors of n: $\gcd(n, x - y) = p$ or $\gcd(n, x - y) = q$
Breaking Rabin $\Rightarrow$ **Factoring** n

Provable Security

- Therefore, we determine that breaking Rabin algorithm is equivalent to factoring — this is sometimes called as "provable security"
- Provable security in cryptography generally means the security of an algorithm is equivalent to the difficulty of another well-known problem
- Provable security is a valuable theoretical concept: knowing that X is as difficult as Y has theoretical importance, but this may not mean much in practice
- Provable security should **not** be construed as **security is assured**
- RSA's security is not known to be equivalent to factoring:
  Factoring $n \Rightarrow$ Breaking RSA
  Breaking RSA $\overset{\text{probably}}{\not\Rightarrow}$ Factoring n