Agenda

PROBLEM DOMAIN: NUMBER THEORY -APPLICATION DOMAIN: CRYPTOGRAPHY - PUBLIC KEY ENCRYPTION : RSA SECURITY OF RSA

RSA: Security

- Protocol RSA: (Public Key Encryption)
  - **Offline Steps** (i.e. pre-processing) by B or a third party trusted by B:
    - Choose $\mathbf{n = p * q}$ for large primes $\mathbf{p}$ and $\mathbf{q}$
    - Choose $\mathbf{k}$ in $\mathbf{Z^*_{\phi(n)}}$ and publish $\mathbf{(n, k)}$ as public key for B
    - Compute B's private key k', the inverse of k in $(Z^*_{\phi(n)}, *\boldsymbol{\phi}\mathbf{(n)})$
  - **Definitions:**
    - Let $\mathbf{E(M) = M^k \ (mod\ n)}$ and $\mathbf{E^{-1}(M') = (M')^{k'} \ (mod\ n)}$
  - **Online Steps** (i.e. at communication time):
    - A sends $\mathbf{M' = E(M)}$ to B
    - B receives $\mathbf{M'}$
    - B computes $\mathbf{E^{-1}(M')}$
- Security Requirement:
  - Given **n** (but not **p** nor **q**), and k
    - *an attacker cannot get **M** in polynomial time*

RSA: Security

- (Provable) Security Correctness Requirement:
  - Given n (but not p nor q), and k
    - an attacker cannot get any of these:
      - p or q
        - because factoring is "hard"
      - $\phi(n)$
        - because computing $\phi(n)$ is as "hard" as factoring
      - k', the inverse of k in $(Z^*_{\phi(n)}, *n)$
        - because if k' is known, then the attacker knows $\phi(n) \mid (k*k' - 1)$
        - i.e. attacker knows $j*\phi(n)$ for some +ve integer j
          - ***Then n can be factorized efficiently.***
          - ***(claim w/o proof).***

RSA: Security - Pragmatics

- (Provable) Security Property for RSA:
  - Given n (but not p nor q), and k
    - an attacker cannot get any of these:
      - p or q
      - $\phi(n)$
      - k′ , the inverse of k in $(Z^*_{\phi(n)}, *n)$
- The above statement states the hardness of breaking RSA scheme completely - by computing the private key
  - Alternatively, is there a way to infer (i.e. decrypt) messages without the decryption key?
  - i.e. we want a guarantee of the form:
    - ***It is not possible to decode more than a small fraction of encrypted messages***

RSA: Security - Pragmatics

- We want a guarantee of the form:
  - *It is not possible to decode more than a small fraction of encrypted messages*
- Given an attacker's algorithm A that knows only n and k, define
  - $C(A) = \{ x \text{ in } Z^*_n \mid A \text{ can compute } x^{k'} \pmod{n} \text{ given } x \}$
    - where k' is the inverse of k in $(Z^*_{\phi(n)}, *n)$
  - i.e. C(A) is the set of messages in Z^*_n that can be recovered using A.

RSA: Security - Pragmatics

- Theorem:
  - Suppose there exists a (possibly randomized) polynomial time algorithm A_1 for which $|C(A_1)| >= \epsilon * |Z^*_n|$ for some $\epsilon > 0$.
  - Then there exists a Las Vegas algorithm A_2 for which
    - $|C(A_2)| = |Z^*_n|$ and
    - the expected running time of A_2 *__is polynomial in log(n) and 1/ ϵ.__*
- Implications:
  - If RSA can be broken (i.e. a more than a small number of messages decrypted) then it can be broken almost completely:
    - Note that $|Z^*_n| = p * q + 1 – (p + q) = \Theta(|Z_n|)$
      - for $n = p * q$ where p and q are large primes.
  - If ϵ is vanishingly small, say for instance $o(1/n)$, then the expected time complexity is exponential in size of n
    - i.e. the attack – using A_2 – may not be practical.

Monte-Carlo Algorithms

- Monte Carlo Algorithms:
  - 1-sided error
    - An algorithm A for a decision problem Π is said to be a Monte-Carlo algorithm with one sided-error if:
      - $A(x) = 1$ implies $\Pi(x) = 1$ (always) and
      - $A(x) = 0$ implies $\Pi(x) = 0$ with a probability p.
    - Then A is said to have an error probability 1-p
  - 2-sided error
    - An algorithm A for a decision problem Π is said to be a Monte-Carlo algorithm with two sided-error if:
      - $A(x) = 1$ implies $\Pi(x) = 1$ with a probability p and
      - $A(x) = 0$ implies $\Pi(x) = 0$ with a probability q.
    - Then A is said to have an error probability 1-p and 1-q
- Appropriate modifications can be made in the definition for function or optimization problems.

Monte-Carlo and Las Vegas Algorithms

- Can a Monte Carlo Algorithm with 1-sided error be used to construct a Las-Vegas algorithm?
  - Consider a (biased) coin that on a toss has the probability p of coming up HEADS and probability 1-p of coming up TAILS
    - What is the expected number of (independent) tosses up to and including the first head?
      - The random variable X denoting the total number of such coin tosses
        - has the geometric distribution with $E[X] = 1/p$

Monte-Carlo and Las Vegas Algorithms

- Lemma Conversion-MC-LV:
  - Consider a Monte Carlo algorithm A for a problem Π

    with expected running time of at most $T_{mc}(n)$

    and an error probability of $U(n)$

    on any instance of size n
  - Suppose further given a solution to Π we can verify its correctness in time $T_v(n)$
  - Then a Las Vegas algorithm to Π can be constructed with expected running time at most
    - $(T_{mc}(n) + T_v(n)) / U(n)$
- Proof:
  - Construct algorithm A':

    repeat cert=A(n); until (test(cert));