RSA

Faculdade de Informática- PUCRS
Prof. Avelino Francisco Zorzo

RSA

- Rivest, Shamir & Adleman - MIT - 1977
- Mais conhecido e usado esquema de chave pública
- Baseado em exponenciação em um corpo finito sobre inteiros módulo um primo
- Usa grandes inteiros (e.g. 1024 bits)
- Segurança devido ao custo de fatorar grandes números

RSA: Estabelecendo chave

- Gerar chaves pública e privada:
  - Selecione 2 primos grandes - p, q
  - Computar o módulo do sistema - $N=p.q$
  - Lembre-se: $\varphi(N)=(p-1)(q-1)$
  - Selecione a chave de cifrar e tal que $1< e < \varphi(N)$ e $\gcd(e, \varphi(N))=1$
  - Encontre a chave de decifrar d tal que $e.d=1 \mod \varphi(N)$ e $0 \leq d \leq N$
- Publique a chave de cifrar pública: $PK=\{e,N\}$
- Mantenha a chave secreta: $SK=\{d,p,q\}$

RSA: Uso

- Para cifrar uma mensagem M :
  - Obtenha a **chave pública** $PK=\{e,N\}$
  - Computar: $C=M^e \mod N$, onde $0 \leq M < N$
- Para decifrar o texto cifrado C:
  - Use a chave privada $SK=\{d,p,q\}$
  - Compute: $M=C^d \mod N$
- Mensagem M deve ser menor que o módulo N
  - Dividir a mensagem em blocos

Por que RSA funciona?

- Lembre-se do Teorema de Euler :
  - $a^{\varphi(N)}=1 \mod N$ onde $\gcd(a,N)=1$
- RSA:
  - $N=p.q$
  - $\varphi(N)=(p-1)(q-1)$
  - e e d inversos em $Z_{\varphi(N)}$
  - $e.d=1+k.\ \varphi(N)$ em $Z_{\varphi(N)}$ para algum k
- Assim:

$$C^d = (M^e)^d = M^{1+k.\varphi(N)} = M^1.(M^{\varphi(N)})^k = M^1.(1)^k = M^1 = M \mod N$$

RSA: Exemplo

1. Selecione primos: $p=5$ e $q=11$ e $N = p.q = 5 \times 11 = 55$
2. Compute $\varphi(N)=(p-1).(q-1) = 4 \times 10 = 40$
3. Selecione e : $\gcd(e,40)=1$; $e=7$
4. Compute d: $d.e=1$ in Z_{40} e $d<40$, $d=23$ pois $23 \times 7 = 161 = 4 \times 40+1$
5. Publique a chave pública $PK=\{7,55\}$
6. Mantenha secreta a chave privada $SK=\{23,5,11\}$

RSA: Exemplo

- Dada a mensagem `M = 8`
- Cifrar:

  $C=8^7 \bmod 55 = 2,097,152 \bmod 55=2$
- Decifrar:

  $M=2^{23} \bmod 55=8,388,608 \bmod 55=8$

Fim.

Attacks on textbook RSA

- Fact 1
  - *Let <N,e> be an RSA public key. Given the private key d, one can efficiently factor the modulus N = pq. Conversely, given the factorization of N, one can efficiently recover d.*
  - Proof of the last part is simple
    - » If you factor N, then you can calculate φ (N) and e is given than you have to calculate the inverse of e mod φ (N) and you have found d.

Attacks on textbook RSA

- Proof of first part
  - N = 55, e = 7, d = 23, compute k = de -1, k = 7.23-1
  - k is a multiple of φ(N) (in our case, φ(N)=40, k = 160).
  - φ(N) is even ((p-1) is even (q-1) is even) therefore φ(N) is even
  - k = 2^5.5, r = 5 (odd), t = 5, t >=1
  - g^k = 1 (from Eulers theorem x^φ(N) = 1 mod N)
  - square root unity module N is x such that x^2 congruent (=) to 1 mod N
  - x^2 = 21^2 = 441 mod 55 = 1 mod 5
  - Now calculate gcd(x-1, N), gcd(20,55) = 5 or gcd(-x-1,N), gcd(-22,55) = gcd(33,55) = 11

Attack: Common modulus

- To avoid generating different modulus for each user, fix N=p.q to several users
- Provide a unique pair $<e_i,d_i>$ for each $user_i$
- Alice, e.g., has PK=$\{e_a,N\}$ and SK=$\{d_a,N\}$
- Bob, e.g., has PK=$\{e_b,N\}$ and SK=$\{d_b,N\}$
- Both Alice and Bob can factor N (Fact 1), and either knows the PK from the other, therefore, either can find the SK from the other.