Reductions and Asymmetric cryptography

Exercice 1. RSA reduction to factorization: small exponent case

Let $n = pq$ with $p, q > 3$, two prime numbers, and $e < n$ be an integer coprime with $\phi(n) = (p-1)(q-1)$. Let $d = e^{-1} \mod \phi(n)$.

RSA is a public key cryptosystem where the public key is a the pair (n, e) and the private key is the pair (n, d).

- The encryption function E is defined by $E(m) = m^e \mod n$
- The decryption function D is defined by $D(c) = c^d \mod n$

The computational problem BREAK_RSA(n, e) is to find the secret exponent d from the public key (n, e).

a. What is the complexity of the key generation and the encryption function ?

b. Show that BREAK_RSA $\leq_P$ FACTORIZATION

We will now focus on the converse: showing that FACTORIZATION $\leq_P$ BREAK_RSA. We will first consider a weaker problem, where the secret exponent e is small: $e = \Theta(\log n)$.

c. Show that $p, q \leq n/4$ and deduce that $\phi(n) \geq n/2$.

d. Show that $\exists k \leq 2e$ such that $ed - 1 = k\phi(n)$.

e. Express $S_k = p + q$ as a function of k, n, e, d

f. How to recover p and q from S_k and n?

g. Write down an algorithm and its complexity analysis.

Exercice 2. RSA reduction to factorization: arbitrary case

We now consider the general case where e can be arbitrarily large. Define s and t such that $ed - 1 = t2^s$, where t is odd. Consider an integer $a \leq n$ coprime with n, chosen at random.

a. Show that $\exists i < s,\ u = a^{t2^i}$ and $\begin{cases} u^2 = 1 \mod n \\ u \neq 1 \mod n \end{cases}$

For the moment, assume that a verifies $\exists i < s,\ u = a^{t2^i}$ and

$$\begin{cases} u^2 = 1 \quad \mod n \\ u \notin \{1, -1\} \quad \mod n \end{cases} \tag{1}$$

b. Show that $gcd(u-1, n) \neq 1$ and deduce an algorithm factoring n.

We will now show that half of the choices for $a \in (\mathbb{Z}/n\mathbb{Z})^*$ satisfy (1).
Write

$$ed - 1 = \underbrace{k}_{\ell 2^\sigma} \underbrace{(p-1)}_{t_1 2^{s_1}} \underbrace{(q-1)}_{t_2 2^{s_2}} = \ell t_1 t_2 2^{\sigma s_1 s_2}$$

where ℓ, t_1, t_2 are odd. Suppose without loss of generality that $s_1 \leq s_2$ and define

$$r = \frac{ed-1}{2^{\sigma+s_1+1}} = \ell t_1 t_2 2^{s_2-1}$$

c. Show that half of the $a \in (\mathbb{Z}/q\mathbb{Z})^*$ verifiy $a^{\frac{q-1}{2}} = 1$ and the other half verify $a^{\frac{q-1}{2}} = -1$

d. Deduce from the above question that half of the $a \in (\mathbb{Z}/q\mathbb{Z})^*$ verify $a^r = -1$ and the other half $a^r = 1$.

e. When $s_1 < s_2$, conclude that half of the $a \in (\mathbb{Z}/n\mathbb{Z})^*$ verify $a^r \notin \{1, -1\}$

f. When $s_1 = s_2$, conclude that half of the $a \in (\mathbb{Z}/n\mathbb{Z})^*$ verify $a^r \notin \{1, -1\}$

Exercice 3. Reduction and RSA

Suppose $n = pq$ is an RSA integer with p, q two large primes, such that it is computationnaly intractable to factor n.

a. Is it possible to compute $\phi(n)$?

b. Which reduction scheme would you use, between the two given below:

1.
```
AlgoReduction1-Fatctorize (n)
{
...
phi = OracleComputePhi (n)
...
return p,q
}
```

2.
```
AlgoReduction1-ComputePhi (n)
{
...
(p,q) = OracleFactorize (n)
...
return phi
}
```

Exercice 4. Merkle-Hellman

Consider Merkle-Hellman protocol (MH). Bob chooses a super-increasing secret sequence of $n = 1000$ integers a_i for $0 \leq i < n$. Alice signs a binary plain text P (a block), computes $C = E_{Bob}(P)$ and sends C to Bob.

a. What is the size of a P?

b. Give an algorithm that Bob uses to build its secret integers a_i.

c. Deduce that, if $a_0 = c$, we may consider $a_i \leq 4^i.c$.

d. What is the order of the size of the cipher text C?

e. Write the algorithms for encoding and decoding and analyze their costs.

f. Conclude on the provable security of MH$(b_0, \ldots, b_{n-1}, m)$.