AARHUS UNIVERSITY - COMPUTER SCIENCE DEPARTMENT

Crypthology - Handin 7

Peter Burgaard - 201209175

November 10, 2016

Let A be an algorithm that gets as input an RSA public key (n, e) and a ciphertext y. A will either return the correct plaintext x, or will return "no answer". Suppose A is able to decrypt if and, and only if y is in some subset S of $Zn*$. Assume also that the size of S is $\epsilon \cdot (p-1)(q-1)$, for $0 < \epsilon < 1$.

Your task: construct a probabilistic algorithm B that uses A as a subrutine. B gets input public key (n,e) and ciphertext z, where z can be any number in Zn. We will assume that z is not 0, as 0 is easy to decrypt anyway.

You must construct B such that for ANY fixed z, B returns the correct plaintext for z with probability at least ϵ.

FIRST PART

We are given as a hint to: "first show that if z is non-zero, but not in Zn*, you can decrypt it easily without using A, by first computing the secret key. If z is in Zn*, you do need to use A."

When $z \in Zn$ but $z \notin Zn^*$ this means z has no inverse in Zn which implies

$$gcd(z, n) = x \qquad \text{where } x \neq 1$$

This means when we know one of the dividers of $n = p \cdot q$(where p and q are primes) we are done. By the above we know that x must be either p or q. So if we let $p = x$ we can easily compute q. Now that we know both primes which divide n we can compute d as

$$e^{-1} \equiv d \mod \phi(n) \equiv d \mod (p-1)(q-1)$$

With the d we can now decode z as $c = z^d$

Second part

We are given another hint for this part: "You will need to use the multiplicative property of RSA stated in Stinson exercise 5.14. Note that you cannot simply run A on input (n,e) and z. If z is not in S, A would always return "no answer" so for such z the success probability would be 0 and not ϵ as required."

If we first assume $z \in S$, then we can just use A and we're done. If $z \notin S$ then A returns no answer, and we will have to use the hint. Firstly we will review multiplicative property of RSA from Stinson:

$$e_K(x_1)e_K(x_2) \mod n = e_K(x_1 \cdot x_2 \mod n)$$

This will be used later.

We will generate some new plain text x, and encrypt it with the public key e

$$e_K(x) = x^e \mod n = y$$

By choosing a random string x we will have probability ϵ of choosing from subset S since we are choosing from $Zn \supset Zn^*$ which is $\frac{\epsilon \cdot (p-1)(q-1)}{(p-1)(q-1)} = \epsilon$

Now we will use the multiplicative property of RSA. Let $e_K(z') = z$, then

$$e_K(z')e_K(x) \mod n = e_K(z' \cdot x \mod n) = y'$$

We can feed y' to A and if it returns an answer, we will get $x \cdot z' \mod n$, where it would be trivial to extract the original z'.

Probabilities

In the above we see that out algorithm has three cases when for input z.

- if $z \in Zn$ and $z \notin Zn^*$. The probability of z being in this case is $Pr(\frac{|Zn| - |Zn^*|}{|Zn|})$, and probability of decoding the ciphertext z in this case is 1.
- if $z \in S$. The probability of z being in this case is $Pr(\frac{|S|}{|Zn|})$, and probability of decoding the ciphertext z in this case is 1.
- if $z \notin S$. The probability of z being in this case is $Pr(\frac{|Zn| - |S|}{|Zn*|})$, and probability of decoding the ciphertext z in this case is ϵ.

As we can see, our algorithm fits the given criterias and we are done.