COMP6453: Week 7 Answers

1 OpenSSL

Use OpenSSL to view the Moodle (moodle.telt.unsw.edu.au) certificate. Answer the following questions:

1. What is the RSA public key algorithm? Analyse its components.
2. What is the signature algorithm? Analyse its components.
3. Try viewing other website certificates and answer questions 1 and 2.

Answer: See Figure 1 at the bottom of the page. Use | grep to view the public key algorithm and signing algorithm.

2 Random Self Reducibility of Discrete Log

We would like the security of cryptographic algorithms to be based on problems which are hard on *average*. Many problems are hard only in the worst case, such as subset sum, which is not optimal for cryptographic purposes. We will show if the discrete log problem is hard in the worst case (which we believe is true), then it must be hard on average.

Let G be a fixed cyclic group of order q and g be a fixed generator of G. Suppose $\mathcal{A}$ is an efficient adversary with the following property: if $\mu \in G$ is chosen at random, then $Pr[A(\mu) = Dlog_g(\mu)] = \epsilon$. Show there is an efficient algorithm $\mathcal{B}$ with the following property: for all $\mu \in G$, algorithm $\mathcal{B}$ outputs $dlog_g(\mu)$ with probability ϵ.

Answer: Algorithm $\mathcal{B}$ takes some fixed $\mu \in G$ as input. It starts by sampling some random $\sigma \leftarrow \mathbb{Z}_q$. Then we compute $\mu_1 = \mu^\sigma$. $\mathcal{B}$ calls algorithm $\mathcal{A}$ on μ_1. Note that μ_1 has a uniform distribution over $\mathbb{Z}_q$. Let α be what $\mathcal{A}$ outputs after being summoned by $\mathcal{B}$. Now if $g^\alpha \neq \mu_1$, output `fail`. Otherwise output $\alpha \cdot \sigma^{-1}$. Since σ is uniform over $\mathbb{Z}_q$, it is clear that $\mathcal{B}$ outputs the correct discrete log value with the same probability as $\mathcal{A}$.

3 Wiener's Attack

Here is an example of the attack carried out on an RSA instance. We are given the public modulus and exponent $pk = (N = 64741, e = 42677)$. We will use Wiener's attack to recover the decryption exponent d.

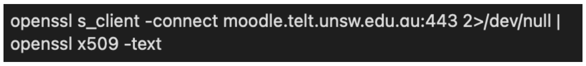
The continued fraction expansion of e/N is $[0, 1, 1, 1, 13, 1, 9, 1, 1, 70]$. The first convergent $0/1$ clearly does not work as d cannot be equal to 1.

Similarly the second convergent $1/1$ does not work.

The third convergent is $1/2$. This does not work, since we discussed in class that d must be odd.

The next convergent is $2/3$. Our value for d is 3, so it is odd. The value for k is 2 and we have that 2 divides $N \cdot d - 1$. So the second check passes. Since we have a candidate value for d, we can derive a candidate $\phi(N)$ value which is $\frac{N \cdot d - 1}{d} = \frac{42667 \cdot 3 - 1}{2} = 64000$.

Next, the attack algorithm tells us to form the equation $x^2 - (N - \phi(N) + 1)x + N$ and check for integer roots. Substituting out values, this equation is $x^2 - 742x + 64741$. Using the quadratic formula, we see that this equation has integer roots. This means we are done and $d = 3$ is the decryption exponent.



```
openssl s_client -connect moodle.telt.unsw.edu.au:443 2>/dev/null |
openssl x509 -text
```

Figure 1: Caption