Introduction to Cryptography

Midterm 1

November 8, 2006

1. Solve the following cryptogram:

   ```
   NOVSG XJ NOQ QMPNO MD M WVCVSI XPIMSVDU NOMN VD LQVSI

   MNNMKGQT LZ LVWWVXSD XJ LMKNQPVM BOXDQ SRULQPD TXRLWQ

   QCQPZ JXPNZ ZQMPD.

   IXPQ CVTMW
   ```

   Explain your first steps.

   The single letter M stands for A or I; the three-letter word NOQ could stand for THE; in fact, the word NOMN would then mean THAT, and this tells us we're on the right track. QMPNO then must be EARTH. MD is then AS (it can't be AT since T is encrypted by N); similarly, ZQMPD must be YEARS or BEARS, not TEARS.

2. Explain how RSA works. How do you pick the parameters p, q, N, d, and e? What is the private and public key? How are messages encrypted and decrypted? Why does it work?

   See the notes. Note that p and q must be chosen to be *large* primes, and that $p-1$ and $q-1$ should have at least one large prime factor to prevent the factorization of N with Pollard's $p-1$-method.

3. In the proof that RSA works, you have used the Theorem of Euler-Fermat, which is only valid if $\gcd(m, N) = 1$. Show that the RSA-protocol even works if $\gcd(m, N)$ is nontrivial.

   Since $N = pq$, there are only three cases: $d = \gcd$ must be one of p, q, or N. In the last case we have $m = 0$ since $m < N$, and then clearly $c = 0$; decryption gives back $m = 0$.

   By symmetry, we may now assume that $d = \gcd(m, N) = p$. Then $m = p^t r$ for some r coprime to N and some integer $t \geq 1$. Encryption gives us

$c \equiv m^e = p^{te} r^e \bmod N$, and decryption $M :\equiv c^d \equiv p^{tde} r^{de} \equiv p^{tde} r \bmod N$ (here we have used Euler-Fermat: $r^{\phi(N)} \equiv 1 \bmod N$). It remains to show that $p^{tde} r \equiv pr \bmod N$. Since $\gcd(r, N) = 1$, this is equivalent to $p^{tde} \equiv p \bmod N$. By the Chinese Remainder Theorem it suffices to prove this congruence mod$p$ and mod$q$. But clearly $p^{tde} \equiv p^t \bmod q$ by Euler-Fermat, and $p^{tde} \equiv 0 \equiv p^t \bmod p$ trivially; thus $p^{tde} \equiv p^t \bmod N$ as desired.

4. We have shown that the complexity of (naive) addition, multiplication and division in terms of bit operations is as follows:

operation	complexity
$a + b$	$O(\max\{\log a, \log b\})$
ab	$O(\log a \cdot \log b)$
$a = bq + r$	$O(\log b \cdot \log q)$

Estimate the number of bit operations to compute the left side and the right side of the equation

$$\sum_{j=1}^{n} j^2 = \frac{n(n+1)(2n+1)}{6}.$$

Left hand side: computing j^2 requires $O((\log j)^2)$ bit operations, hence computing all the squares costs us $O(\sum_{j=1}^{n} (\log j)^2)$ bit operations; using the crude estimate $j \leq n$ shows that this can be bounded by $O(n(\log n)^2)$ [A more careful estimate uses the comparison with $\int_2^{n+1} \log x dx$ and gives a bound of the form $O(n \log n)$].

Now we have to add these numbers; since the sum is bounded by n^3, adding two terms (of size $< n^3$ and $< n^2$, respectively) costs at most $\log n^3$, i.e., $O(\log n)$, bit operations, and all of the additions can be performed using less than $O(n \log n)$ bit operations.

Both calculations therefore require less than $O(n(\log n)^2)$ bit operations.

Right hand side: since the product $n(n+1)$ costs $O(\log n \log(n+1)) = O((\log n)^2)$ bit operations, we can neglect everything that is cheaper, like adding 1, or multiplying or dividing by 2 or 6. Computing the product $(n^2 + n)(2n + 1)$ takes $O(\log(n^2 + n) \log(2n + 1)) = O((\log n)^2)$ bit operations.

Thus the complexity of computing the right hand side is $O((\log n)^2)$; as expected, this is a lot faster than computing the left hand side directly.

5. The Paillier cryptosystem works as follows: Alice picks distinct primes p and q, computes $n = pq$, and determines an integer a with $an \equiv 1 \bmod \phi(n)$. The public key is n, the private key is a.

   To encrypt a message $m < n$, Bob picks a random integer $r < n$ coprime to n, computes $c \equiv (1+n)^m r^n \bmod n^2$, and sends c to Alice.

   For decrypting c, Alice computes the smallest positive $R < n$ with $R \equiv c^a \bmod n$, $z \equiv cR^{-n} \bmod n^2$, and finally $M = \frac{z-1}{n}$.

   (a) Show that $c \equiv r^n \bmod n$.

   This is trivial since $1 + n \equiv 1 \bmod n$.

   (b) Show that $R = r$.

   We have $R \equiv c^a \equiv r^{an} \equiv r \bmod n$ since $an \equiv 1 \bmod \phi(n)$. Since $0 < R, r < n$, the claim follows.

   (c) Show that $(1+n)^m \equiv 1 + mn \bmod n^2$.

   This is a trivial consequence of the binomial theorem $(1+n)^m = 1 + \binom{m}{1} n + \ldots$.

   (d) Show that $M = m$ (in particular, anyone knowing a can decrypt the cipher texts c).

   We have $z \equiv (1+n)^m r^n r^{-n} \equiv 1 + mn \bmod n^2$, hence $M = \frac{z-1}{n} = m$ as desired.

   (e) Show that Eve can break the Paillier system if she can factor n.

   If Eve can factor n, she can compute $\phi(n)$ and then solve the congruence $an \equiv 1 \bmod \phi(n)$. But this gives her Alice's private key a.

   (f) Show that the Paillier cryptosystem is *additively homomorphic*, i.e. that if c_1, c_2 are encryptions of m_1 and m_2, then $c_3 \equiv c_1 c_2 \bmod n^2$ is an encryption of the message $m_3 \equiv m_1 + m_2 \bmod n$.

   We have $c_1 \equiv (1+n)^{m_1} r_1^n \bmod n^2$, $c_2 \equiv (1+n)^{m_2} r_2^n \bmod n^2$, hence $c_1 c_2 \equiv (1+n)^{m_1+m_2} (r_1 r_2)^n \bmod n^2$ is a possibly encryption of $m_1 + m_2$.

6. Bob computes $2^{n-1} \bmod n$. What conclusions can Bob draw if he finds that $2^{n-1} \not\equiv 1 \bmod n$? What if $2^{n-1} \equiv 1 \bmod n$?

   If $2^{n-1} \not\equiv 1 \bmod n$, then Fermat's Little Theorem tells Bob that n is not a prime. If $2^{n-1} \equiv 1 \bmod n$, then n may be prime or composite.

7. Describe Pollard's rho method for finding prime factors p of composite integers. What is the connection with the birthday paradox?

   See the notes.