# Tutorials and Exercises

- Number Theory and Cryptography -

Nicolas T. Courtois, December 2007

**Exercise 1.** Recall and prove the Bezout theorem.
Show that if $d|a$ and $d|b$, then $d|GCD(a,b)$.
Show that if $p$ is prime and $p|ab$, then $p|a$ or $p|b$. (the fact that each integer has a unique decomposition as a product of prime numbers is a direct consequence of that and cannot be used).

**Exercise 2.** Recall the Chinese remainder theorem (CRT).
A small Chinese army unit has been ordered in 3 ranks. Two soldiers remained. Then it has been ordered in 7 ranks. 6 soldiers remained.
Finally in 10 ranks. Again two remained. How many soldiers (at least) were there ?

**Exercise 3.**
Compute the $7-th$ root of 23 in $Z_{77}^*$.

**Exercise 4.**

Show that, in order to prove that an integer $n$ is prime, it is sufficient to exhibit **one** element of (multiplicative) order $n-1$ mod $n$. Prove that 257 is prime (for this we use the Lagrange Theorem, show that $3^{256}=1 \bmod 257$ and that $3^{128} \neq 1 \bmod 257$). Factor $2^{16}-1$.

**Exercise 5.**
Let $p$ be an odd prime. We assume that the factorisation of $p-1$ is known.

$$p-1 = p_1^{\alpha_1} \cdot \ldots \cdot p_r^{\alpha_r}$$

1. Devise an algorithm which checks that an element $g \in Z_p^*$ is a generator. What is its complexity ?
2. What is the probability that a random $g$ is a generator ? Hint: $\phi(p-1)/(p-1)$.
3. Propose an algorithm to find a generator. What is the complexity ?

**Exercise 6.**

Let $n = p_1 \times \ldots \times p_r$ with **distinct** odd primes.
Let $\mathcal{C}$ be the ring isomorphism used in the CRT.

1. By checking all possible cases, find the QR and QNR for $Z_{35}^*$.
2. Show that $a$ is a $QR$ modulo $n$, if and only if **all** the $a$ mod $p_i$ are $QR$.
3. Show that for any QR $x$, there exists $k$ such that there are **exactly** $2^k$ square roots of $x$.
4. Show that $QR$ is a subgroup of $Z_{35}^*$. Compute the order of this subgroup.
5. Show that we always have QR $\times$ QNR= QNR.
6. Find one example when QNR $\times$ QNR is a $QR$. Show that QNR is NOT a subgroup for $n = 35$.

**Exercise 7.**

We assume that two banks use RSA with two different public exponents $e_1 = 3$ and $e_2 = 5$ but with the same 1024-bit RSA modulus $n$ that has been generated by some central authority. The authority has delivered to the banks the private exponents $d_1$ and $d_2$.
1. Show that each bank can forge signatures on behalf of the other bank.
2. Assume that an important message $m$ on 1024 bits is being sent to both banks. Show that everybody can decrypt this message.

**Exercise 8.** A Provably Collision-Resistant Hash Function Based on Number Theory.

Let $p = 2p' + 1$ and $q = 2q' + 1$ are two safe primes, which means that $p'$ and $q'$ are also primes. Let $n = pq$ and let $g$ be an element of order $p'q'$ in $Z_n^*$.

1. Recall what is the probability that a random number $n$ is a prime. Recall what is the complexity of Miller-Rabin primality test (it requires a constant number of modular exponentiations to achieve error probability as small as necessary, e.g. $2^{-80}$). Give an algorithm to generate a safe prime $p$ (the same is used for $q$) and evaluate the complexity for the final result being on $k$ bits.
2. Explain how to generate $g$.
   Hint 1: Note that $g$ is of order $p'$ mod $p$ and of order $q'$ mod $q$. Hint 2: Show that the subgroup of QR in $Z_p$ are exactly the residues such that $x^{(p-1)/2} = 1$ and that the subgroup QR is cyclic of order $p'$.
   Hint 3: Show that in a cyclic group of prime order, all elements except 1 are generators.
   Hint 4: Show that it is sufficient to pick a random square in $Z_n^*$ such that $g$ mod $p \neq 1$ and $g$ mod $q \neq 1$.

   Now we encode a message as an integer of an arbitrary size, and we define the hash function $H(m) = g^m$ mod $n$.

3. We call the exponent of $Z_n^*$, an integer $\lambda(n)$ being the lcm of orders of all elements of $Z_n^*$. Compute $\lambda(n)$. We assume that the following fact is known: given $\lambda(n)$ and $n$, one can compute factors of $n$.
Show that finding collision on $H$ is at least as hard as factorizing $n$.

4. Show that finding a pre-image on $H$ is at least as hard as solving the DL problem w.r.t. to base $g$ in both $Z_p^*$ and $Z_q^*$.

**Exercise 9.** Let $GF(2)$ be a field with two elements.

1. Look at these two polynomials $X^3+1$, $X^3+X+1$, which one is irreducible in $GF(2)[X]$? Let $P(x)$ be this polynomial and $P'(X)$ the other. Give a complete proof that $P(X)$ is indeed irreducible.

2. Factor $P(X)$ and $P'(X)$ in $\mathbb{F}_2[X]$.

3. We define a field $F$ as the set of all monomials modulo $P(X)$, or in other terms $F = GF(2)[X]/P(X)$, with the addition and the multiplication of polynomials modulo $P(X)$. How many elements has this field?

4. How many solutions in $F$ has the equation $x^2 = x$? Write all of them and prove that there is no more.

5. Compute $1, X, X^2, \ldots$ modulo $P(X)$.

6. Is $P$ a primitive polynomial?

7. How many solutions in $GF(2)$ has the equation $x^2 = 1$?

8. How many solutions in $F$? Prove that there is no more.

9. And in $GF(3)$?

10. And in $Z_n$ with $n = pq$ a product of two large primes? (Hint: Chinese Remainder Theorem).

11. And in $Z_6$? (Hint: under what condition on $p$ and $q$ one can apply CRT?).

12. And in $Z_4$?

13. Let $Sq : F \to F$ be defined as $Sq(x) = x^2$ in $GF(2)[X]/P(X)$. Show that $Sq(x+y) = Sq(x) + Sq(y)$.

14. Show that $Sq$ is one-to-one.

15. We define a **one-time** encryption scheme on 3-bits with 3-bit keys as $c = E_k(m) = k+Sq(m)$. Show that this scheme achieves perfect unconditional confidentiality (Shannon).

**Exercise 10.** Let $\alpha \in GF(p^2)$ such that $\alpha$ satisfies the polynomial equation $X^2 + aX + b = 0$ with $a, b \in GF(p) \subset GF(p^2)$ and such that $\alpha \notin GF(p)$.
1. Prove that if some $\beta \neq \alpha$ also satisfies this equation then $\beta = \alpha^p$.
2. Prove that $a = -\alpha - \alpha^p$ and $b = \alpha^{p+1}$.

**Exercise 11.** Prove that when ...(complete the statement)... we have $\phi(mn) = \phi(m)\phi(n)$. Hint: Use CRT.

**Exercise 12.** Prove that

$$\sum_{i|n} \phi(i) = n$$

Check what happens when $n = pq$.

**Exercise 13.** Let $n = pqr$ a product of 3 different prime numbers. How many solutions has the equation $x^2 = 81 \bmod n$ ?

**Exercise 14.** Basic Attacks on DSS.
We recall very briefly the DSS signature scheme. Let $p$ be an integer on 1024 bits and let $g$ be an element of order $q|p-1$ with $q$ being a prime that has 160 bits. The private key of Alice is an integer $0 < a < q$ and her public key is $g^a \bmod p$.
To sign a message $m$ Alice hashes the message to obtain an integer $h = H(m)$ with $0 < h < q$. Then she chooses a random integer $k$ on 160 bits such that $0 < k < q$. Let $r$ be $r = (g^k \bmod p) \bmod q$. Then she computes an integer $s$ such that $sk \equiv h + ar \bmod q$. The signature is a couple $(r, s)$.

1. Show that when a smart card does not have a physical random number generator and no non-volatile memory the the scheme may be very insecure. More precisely, show that if the card can be reset to a previous state and if the same random $k$ is used twice to sign with DSS, then one can recover the private key.

2. Show that when a smart card does have a very poor random number generator and the entropy of $k$ is only 30 bits, the scheme is insecure.

3. What will be the complexity of the previous attack when the random number generator is good ?

4. Show that if the DL is easy in $< g > \subset Z_p^*$, then anyone can forge signatures.

5. What if the DL is easy in $Z_p^*$ ?

6. Otherwise, what is the complexity of this attack with Shanks baby-step giant-step algorithm ?

7. We assume that $H$ is SHA-1 with the output reduced modulo $q$ if $\geq q$.
   Show that if $H$ is not one-way, then there is an existential forgery.

8. What will be the complexity of the previous attack when $H$ is SHA-1 and with brute force inversion ?

9. IF $H$ is OW but not CR, show that we can forge a signature

10. What will be the complexity of this attack if $H$ is SHA-1 and with birthday paradox ?

**Solution to Exercise 4.**

Let $g$ be such that $g^{n-1} \equiv 1 \bmod n$ and $g^i \neq 1 \bmod n$ for $0 < i < n-1$. We show that these numbers $g^i \neq 1 \bmod n$ for $0 < i < n-1$ are distinct non-zero residues mod $n$. They are distinct because if $g^i \equiv g^j \bmod n$, then $g^{i-j}$ would be $\equiv 1 \bmod n$. They are non-zero because $g$ is invertible and therefore any power of $g$ is.
So these $g^i \neq 1 \bmod n$ for $0 < i < n-1$ are exactly all residues $1 \dots n-1$, each is taken once. So all residues $1 \dots n-1$ are invertible. Now if $p|n$ then $p$ is not invertible mod $n$, contradiction.
Now, to prove that 257 is prime we need to prove that the order of 3 mod 257 is 256 and not less. First we check that $3^{256} = 1 \bmod 257$. Then, by Lagrange theorem, $ord(3)|256$. So we verify that $3^{128} \neq 1 \bmod 257$.
Finally, $2^{16} - 1 = (2^8+1)(2^8-1) = 257 \cdot (2^4+1)(2^4-1) = 257 \cdot 17 \cdot 5 \cdot 3$.

**Solution to Exercise 8.** A Provably Collision-Resistant Hash Function Based on Number Theory.

We have $p = 2p' + 1$ and $q = 2q' + 1$ and $n = pq$.

1. Let $k = \lceil \log_2(n) \rceil$. The probability that $n$ is a prime is about $\frac{1}{\ln n}$. We expect that the probability that $n$ and $2n+1$ are both primes will be about $\frac{1}{\ln^2 n} = \frac{1}{k^2/log_2^2(e)}$.

   Assuming that a modular multiplication takes $\mathcal{O}(k^2)$ operations, one modular exponentiation requires $\mathcal{O}(k^3)$ operations. since Miller-Rabin primality test requires a constant number of modular exponentiations to achieve error probability as small as necessary, e.g. $2^{-80}$, the complexity of the test is $\mathcal{O}(k^3)$. The test has to be repeated $k^2/log_2^2(e)$ times on average. Thus the total complexity is $\mathcal{O}(k^5)$.

2. Let $g$ be a randomly chosen square in $Z_n^*$ such that $g \bmod p \neq 1$ and $g \bmod q \neq 1$. Since $g$ is a square by Fermat's little theorem $g^{(p-1)/2} = 1 \bmod p$. Then $(p-1)/2 = p'$ is a prime by Lagrange theorem, the order of $g \bmod p$ must divide $p'$, and thus it must be $p'$. Similarly, the order of $g \bmod q$ is $q'$. Now what is the order of $g$ modulo $n = pq$ ? Let $\mathcal{C}$ be the CRT ring isomorphism: $\mathcal{C}(x) = (x \bmod p, x \bmod q)$

   If we look at the sequence $\mathcal{C}(1), \mathcal{C}(g), \mathcal{C}(g^2), \mathcal{C}(g^3), \dots$. The first coordinate has cycle $p'$ and the second has cycle $q'$. The whole has thus cycle of length $p'q'$. By CRT it has to be the same for the sequence $1, g, g^2, g^3, \dots$

3. (Difficult)

   If one has $g^m = g^{m'} \bmod pq$ then assuming $m > m'$ it means that $(m-m')$ is a multiple of $p'q'$. Let $(m-m') = Kp'q'$. We also have $n - 1 = (2p'+1)(2q'+1) - 1 = 4p'q' + 2(p'+q')$.

   If we compute $L = 4(m-m')/(n-1)$ we get a good approximation of $K$: $L(n-1) = 4Kp'q'$ and $L(n-1) - 4Lp'q' < 2\sqrt{n}$. Thus $K - L < 2/\sqrt{n}$. Since $K$ is an integer, this allows to compute $K$ as the closest integer to $L$. Finally, we get $p'q' = (m-m')/K$ and we compute $p' + q'$ from the

equation $n - 1 = 4p'q' + 2(p' + q')$. Now, given $p'q'$ and $p' + q'$ it is easy to compute $p'$ and $q'$.

For example, we can compute $(p' - q')$ by using the following quite remarquable identity:

$$(p' + q')^2 - (p' - q')^2 = 4p'q'$$

Then we compute $p'$ and $q'$.

4. Assume there is an oracle the can find a pre-image for $H$ that works with some probability $P$, i.e. given $y$ it computes $\alpha$ s.t. $y = g^\alpha \bmod n$. Then this oracle also computes a DL in $Z_p^*$ w.r.t the generator $g \bmod p$ which will be $\alpha \bmod p'$. It remains to use the fact that computing DL with respect to one basis is equivalent to computing it w.r.t. another basis: $\log_a b = \log_g b / \log_g a$.

**Solution to Exercise 9.**

1. $P(X) = X^3 + X + 1$.

   It is irreducible because of degree 3 and has no roots. (Beware, when a polynomial has no roots, it does NOT in general imply it is irreducible (!). It can simply be a product of two polynomials of degree 2 that have roots that are not in the base field. ) For polynomials of degree 3 the argument is valid, because if it has non-trivial polynomial factors, one must of degree exactly 1, (as 3=2+1). Every polynomial of degree exactly 1 and not 0 in a field has a root: $ax + b = 0$ has a root $-b/a$.

2. $P(X)$ has no factors other than itself, and $P'(X) = (X + 1)(X^2 + X + 1)$. (This factorisation can be found by finding a root of this polynomial and dividing it by $(X + 1)$.)

3. $2^3 = 8$.

4. 0 and $1 \in F = GF(2)[X]/P(X)$. Since we are in a field, a polynomial of degree 2 has at most two roots. 0 and 1 are roots, and there is no more.

5. $1, X, X^2, X + 1, X^2 + X, X^2 + X + 1, X^2 + 1, 1, \ldots$.

6. yes

7. 1

8. 1. There is at least 1 solution 1, and at most 2 solutions because the equation is of degree 2. All roots of this equation form a subgroup of $(F\backslash\{0\}, \times)$. The cardinal of this group must divide the order of $F\backslash\{0\}$ which is odd and equal to 7. It cannot be 2 and must be 1. This ends the proof that there is only one square root of 1 in $F$. (This proof works for any field $GF(2^k)$.)

9. 2, which are 1 and 2.

10. 4 solutions. Large prime numbers are $> 2$ and odd. We will use the Chinese Remainder Theorem. There are solutions 1 and $p-1$ mod $p$, there are solutions 1 and $q-1$ mod $q$, so there are 4 possible couples ($x$ mod $p$ and $x$ mod $q$), which by CRT are mapped to exactly 4 remainders modulo $n$.

11. only $2 = 2 \cdot 1$, because $p = 2$ is even there is 1 solution mod 2 (CRT still applies because $GCD(2,3) = 1$).

12. Here (and only here) we CANNOT apply the CRT, as $p = 2$ and $q = 2$ are NOT relatively prime (and it would give a false result $1 \cdot 1$). By exhaustive enumeration we see that the equation has 2 solutions 1 and 3.

13. For all polynomials $x$ and $y$ in $GF(2)[X]$, we have $Sq(x+y) \equiv Sq(x) + Sq(y) + 2Sq(x)Sq(y) = Sq(x) + Sq(y)$. The same holds for their remainders modulo $P(X)$.

14. Since $F \backslash \{0\}$ is a cyclic multiplicative group, there is no element in $F \backslash \{0\}$ with $x^2 = 0$. Therefore, the equation $Sq(x) = 0$ has only one solution $x = 0$. From this we show that $Sq$ is injective, because if $Sq(x) = Sq(y)$ then $Sq(x-y) = Sq(x) - Sq(y) = 0$ which implies $x = y$. Finally, since it is injective function $F \to F$, it must be surjective (images of all elements of $F$ are all different, they must cover the whole set $F$).

15. Since $m \mapsto m^2$ is a publicly known bijective transformation, the security of $E_k$ is totally equivalent to the security of $E_k(m) = k + m$ which is the Vernam one-time pad on 3 bits that achieves perfect privacy.

**Solution to Exercise 12.** We want to prove that

$$\sum_{i|n} \phi(i) = n$$

We observe that each element $a \in Z_n$ generates some cyclic additive subgroup of it that will be called $< a >$. This including 0 that generates a group $\{0\}, +$ that is a trivial group. In each subgroup we will simply select the smallest non-zero element that must be a generator of this group, this is because if $g$ is not a multiple of $a$, then at some moment $ka > g$ while $(k-1)a < g$ and $ka - g$ would be a group element smaller than $a$, contradiction. Also, $a|n$ for the same reason. Each subgroup will be represented by this minimal generating element $a$ with $a|n$. For $\{0\}, +$ we will say that $a = n$ (as we want $a \neq 0$).
We define the following application: for any residue $x \in Z_n$ let $f(x) =$ this smallest generator $a$ of the group generated by $x$ with notably $f(0) = n$. Now all the integers $0, \ldots n-1$ can be grouped according to the same value of $a$. This is a partitioning.

$$\{0, \ldots n-1\} = \dot{\bigcup_{a|n}} f^{-1}(\{a\})$$

Now we need to look at the size of each preimage set $f^{-1}(a)$. Let $a \neq n$ and $a|n$. This subset, for some fixed $a$, contains elements $x$ that are multiples of $a$, but are not in any strict subgroup of $< a >$. So $x = ka$ with $0 \leq k < n/a$. We show that $GCD(k, n/a) = 1$. Indeed, otherwise we have a strict subgroup of

$< a >$. So there are $\phi(n/a)$ elements in this subset of $0 \dots n-1$. We note also when $a = n$ the size of the set $\{0\}$ is $1 = \phi(1) = \phi(n/a)$.
To summarize, for each divisor $a$ of $n$ we have exactly $\phi(n/a)$ elements in $0, \dots n-1$ such that $f(x) = a$ and these elements are defined as $\{x | x = ka$ with $k < n/a$ and $GCD(k, n/a) = 1\}$ (subtle point: this definition works also when $a = 1$ as $GCD(0, 1) = 1$). In particular when $n/a = 1$ the set is $\{0\}$ and when $n/a = n$ the set is $Z_n^*$.
This is a partition of the set $0, \dots n-1$.
Therefore $\sum_{i|n} \phi(n/i) = n$ and $\sum_{i|n} \phi(i) = n$.

When $n = pq$ we get $n = 1 + (p-1) + (q-1) + (p-1)(q-1)$.

**Solution to Exercise 14.**

1. For the fist signature we have $sk \equiv h + ar \bmod q$ and $r = (g^k \bmod p) \bmod q$.

   For the fist next signature we reset the card to the same state, assume that the "randomness" can be reproduced and we have $s'k \equiv h' + ar \bmod q$ and $r = (g^k \bmod p) \bmod q$.

   Thus we get: $(s - s')k \equiv h - h' \bmod q$ and since $q$ is a prime, we can directly recover $k$ and compute $r = (g^k \bmod p) \bmod q$.

   Finally, now int he equation $sk \equiv h + ar \bmod q$, we know everything except $a$. Thus we can compute $a$ which allows to forge any signature.

2. Low entropy (assuming that the probability distribution is known) implies the possibility for the attacker to guess $k$ by trying some $2^{30}$ highly probable values of $k$. As above, given $k$ we compute $r$ and $a$. We do a Known-Message Attack. Thus, from one known message-signature pair we can compute a list of $2^{30}$ plausible candidates for $a$. Thus, given another message-signature pair we check if this $a$ is correct. Thus we can compute $a$ which allows to forge any signature. The complexity of the attack is about $2^{30} \cdot C$ where $C$ is less than the cost of one DSA signature...

3. Since in general the entropy of $k$ should be about 160 bits, as $0 < k < q$, it is about $2^{160}$.

4. If the DL is easy in $< g > \subset Z_p^*$, then from the public key of Alice $g^a \bmod p$ one immediately recovers the private key of Alice which is an integer $0 < a < q$.

5. If the DL is easy in $Z_p^*$ then the DL is also easy in a subgroup $< g > \subset Z_p^*$. We can hope that any algorithm that works in $Z_p^*$ just works in $< g >$.

6. The complexity is $2^{160/2}$. Memory is $2^{160/2}$ with baby-step giant step but this requirement can be removed with Pollard's rho method as will be explained here below.

   The most obvious method to compute DL of $g^a \bmod p$ is the exhaustive search to solve the equation $A = g^x$ with $A$ being the public key of Alice. the complexity will be about $q$ which is about $2^{160}$.

   The second method is called baby-step giant step. Instead of exhaustive search we use the Birthday-Paradox approach and proceed as follows. Let

$m = \lceil \sqrt{n} \rceil$. We can write $x = im + j$ with both $i$ and $j$ being 80-bit integers. then we need to solve the equation: $A(g^{-m})^i = g^j$. We use a hash table of size about $\mathcal{O}(\sqrt{q}) \approx 2^{80}$. The whole attack requires $\mathcal{O}(\sqrt{q}) \approx 2^{80}$ operations and as much memory. (this method is also described in Menezes-Oorshot-Vanstone book section 3.6.2. and Example 3.58.)

In order to remove the storage requirements we use Pollard's rho method with a specially prepared mapping. We use a method that allows to find cycles in $\mathcal{O}(\sqrt{q})$ with negligible storage. For more details we refer to Menezes-Oorshot-Vanstone book section 3.6.3. and Example 3.61.

7. If $H$ were not one-way, then in the Known-Message attack, given one pair message-signature $(m, (r, s))$, one would be able to compute another message $m'$ such that $H(m) = H(m')$ and therefore the same signature would also be valid for $m'$.

   ((Note that the attack might fail if we have only access to pairs message-signature $(m, r, s)$ with some very special $m$ for which SHA-1 is weak, would leak some information about $m$ and thus the attacker would systematically produce $m' = m$. This is very, very unlikely. In practice $m$ will be unrelated to any weakness of SHA-1 (and no such weakness was ever even suspected to exist). There are many other pre-images for $H(m)$, and the attacker cannot compute $m$ because he simply does NOT have any information on $m$ other than $H(m)$, so the probability that $m = m'$ is negligible. ))

8. Brute force inversion on SHA-1 will take $2^{160}$ computations of SHA-1.

9. This will be Chosen-Message attack. A secretary will produce a collision: $H(m) = H(m')$ and ask the boss to sign $m$ which gives $(r, s)$. The message $m'$ has also been signed and $(m, (r, s))$ is valid.

10. Brute force collision search with SHA-1 will take $2^{80}$ computations of SHA-1. It can implemented without using any memory with Shanks baby-step giant-step algorithm.