Computer Algebra
Spring 2011
Assignment Sheet 3

Exercises marked with a ★ can be handed in for bonus points. Due date is April 5.

**Exercise 1**
Determine the remainder that one gets when dividing $a = 2^{37500120314007842499}$ by 101. We know that $a = q \cdot 101 + r$, where $q \in \mathbb{Z}$ is an enormously large number, and $0 \leq r < 101$. We are only interested in $r$.

By Fermat’s theorem, $x^{100} \equiv 1 \pmod{101}$ for all $x \in \mathbb{Z}_{101}^\times$. Therefore, $a = 2^{37500120314007842499} \equiv 2^{99} \equiv 2^{-1} \pmod{101}$. That is, we need to find the inverse of 2. Using the extended Euclidean algorithm or intuition, we find that $2 \cdot 51 \equiv 1 \pmod{101}$.

Now we know $a - 51 \equiv 0 \pmod{101}$, which implies $a - 51 = q \cdot 101$ for some $q \in \mathbb{Z}$. Since $0 \leq 51 < 101$ and division with remainder has a unique result, we know that $r = 51$.

**Exercise 2**
Let $N = pq$, where $p \neq q$ are primes. Show that given only $N$ and $\varphi(N)$, one can compute the prime factors $p$ and $q$ efficiently.

We know that $\varphi(N) = (p - 1)(q - 1)$, allowing us to compute:

$$\varphi(N) = pq - (p + q) + 1 = N - p - N \frac{1}{p} + 1$$

Since we are given $N$ and $\varphi(N)$, this is a quadratic equation in $p$:

$$p^2 + (\varphi(N) - N - 1) \cdot p + N = 0$$

Since we can find the roots of a polynomial efficiently, this allows us to find $p$ and $q$.

A different way to arrive at the same result is to explicitly define the polynomial which has roots $p$ and $q$:

$$f(x) = (x - p)(x - q) = x^2 - (p + q) \cdot x + pq = x^2 + (\varphi(N) - N - 1) \cdot x + N$$

**Exercise 3**
Let us first compute $x^{N-1} \pmod{p}$. Since $p$ is prime, $x^{p-1} \equiv 1 \pmod{p}$ for all $x$.

$$x^{N-1} \equiv x^{p(2p-1)-1} \equiv x^{p(2p-2)+(p-1)} \equiv 1 \pmod{p}$$
In other words, \( x^{N-1} \equiv 1 \pmod{p} \) for all \( x \in \mathbb{Z}_N^* \). So it follows that \( x \) is a Fermat liar if and only if \( x^{N-1} \equiv 1 \pmod{2p-1} \). If \( x \equiv a^2 \pmod{2p-1} \), then
\[
x^{N-1} \equiv a^{2(N-2)} \equiv a^{2p(2p-1)-2} \equiv a^{2p(2p-2)+2p-2} \equiv 1 \pmod{2p-1},
\]
so \( x \) is a Fermat liar.

Conversely, if \( x \) is a Fermat liar, then
\[
1 \equiv x^{N-1} \equiv x^{p(2p-1)-1} \equiv x^{p(2p-2)+p-1} \equiv x^{p-1} \pmod{2p-1},
\]
so the order of \( x \in \mathbb{Z}_{2p-1}^* \) is a factor of \( p-1 \). Since \( \mathbb{Z}_{2p-1}^* \) is cyclic, this implies that \( x \) is a square, i.e. \( x \equiv a^2 \pmod{2p-1} \) for some \( a \in \mathbb{Z}_{2p-1}^* \).

Furthermore, since \( \mathbb{Z}_{2p-1}^* \) is cyclic, exactly half its elements are squares. Since \( \mathbb{Z}_N^* \equiv \mathbb{Z}_p^* \times \mathbb{Z}_{2p-1}^* \), and an element \((x, y) \in \mathbb{Z}_p^* \times \mathbb{Z}_{2p-1}^* \) corresponds to a Fermat liar if and only if \( y \) is a square, it follows that exactly half of such pairs are Fermat liars.

**Exercise 4**

Let \( N = p^k \) where \( p \) is prime and \( k \geq 2 \). Show that \( N \) is not a Carmichael number.

We know that \( |\mathbb{Z}_N^*| = \phi(N) = (p-1) \cdot p^{k-1} \). In other words, the group order is a multiple of \( p \), since \( k \geq 2 \). It follows that there exists a group element \( x \in \mathbb{Z}_N^* \) of order \( p \). Let us compute:
\[
x^{N-1} \equiv x^{p^{k-1}} \equiv x^{-1} \pmod{N},
\]
because \( x^p \equiv 1 \pmod{N} \) by the order of the element. Since \( x \not\equiv 1 \), we also have \( x^{-1} \not\equiv 1 \), and therefore \( x \) is not a Fermat liar. Consequently, \( N \) is not Carmichael.

**Exercise 5**

Suppose you are given \( N = pq \), where \( p \neq q \) are primes, and \( x \in \mathbb{Z}_N^* \) such that \( x \equiv 1 \pmod{p} \) and \( x \not\equiv 1 \pmod{q} \). Show how to compute \( p \) and \( q \) efficiently.

We have \( x-1 \equiv 0 \pmod{p} \) and \( x-1 \not\equiv 0 \pmod{q} \). This means that \( x-1 \) is a multiple of \( p \), but not of \( q \). Therefore, \( \gcd(x-1, N) = p \).

**Exercise 6**

Let \( N = pq \), where \( p \neq q \) are primes, and let \( e \neq d \) be natural numbers such that \( ed \equiv 1 \pmod{\phi(N)} \). Show that given only \( N \), \( e \), and \( d \), one can efficiently compute the prime factorization of \( N \).

We can easily check whether \( N \) is even, so for the remainder, we will assume that \( p > 2 \) and \( q > 2 \). We know that
\[
ed = k \cdot \phi(N) + 1
\]
for some \( k \in \mathbb{Z} \). So for any \( x \in \mathbb{Z}_N^* \), one has (in the ring \( \mathbb{Z}_N \)):
\[
x^{ed-1} = x^{k \cdot \phi(N)} = (x^{\phi(N)})^k = 1^k = 1
\]
Write \( ed - 1 \) as the product of an odd number and a power of two:
\[
ed - 1 = M \cdot 2^m
\]
Since we are given $e$ and $d$, both $m$ and $M$ can be computed efficiently. Consider the following fragment of an algorithm:

1. $x \leftarrow_R \{1, \ldots, N-1\}$
2. If $\gcd(x, N) \neq 1$, return that factor.
3. $y_0 \leftarrow x^M$
4. for $j \leftarrow 1 \ldots m$
   5. $y_j \leftarrow y_{j-1}^2$
   6. if $y_j = 1$
      7. then return $\gcd(y_{j-1} - 1, N)$

If the random choice of $x$ happens to be a non-invertible element of $\mathbb{Z}_N$, then the initial computation of the greatest common divisor yields a factor of $N$ (why?). Of course, the probability that this happens is rather small.¹

Otherwise, conditioning on this not happening, $x$ is uniformly distributed in $\mathbb{Z}_N^\ast$. Note that $y_j = x^{M\cdot 2^j}$ (by induction!), and so one always has $y_m = 1$, independently of the random choice of $x$ in the beginning, so the algorithm will always eventually return from the last line.

The goal is now to show that when it does, it will return $p$ or $q$ with a high probability. The previous exercise is essentially a hint on how to show this. If $y_{a-1}(x) \equiv 1 \pmod{p}$ and $y_{a-1}(x) \equiv 1 \pmod{q}$, then the algorithm will return $p$. What is the probability that this happens? We will follow a strategy similar to the proof of the Miller-Rabin primality test.

Let $a$ be the smallest number such that $y_a(x) \equiv 1 \pmod{p}$ for all possible choices of $x \in \mathbb{Z}_N^\ast$. To make the notation less confusing, we will write $y_j(x) = x^{M\cdot 2^j}$, i.e. $y_j(x)$ is the value that $y_j$ takes given a fixed initial choice of $x$. So the formal definition of $a$ is

$$a := \min\{j \mid y_j(x) \equiv 1 \pmod{p} \text{ for all } x \in \mathbb{Z}_N^\ast\}$$

Similarly, we define

$$b := \min\{j \mid y_j(x) \equiv 1 \pmod{q} \text{ for all } x \in \mathbb{Z}_N^\ast\}$$

We know that $a, b \leq m$ by the observation above. We also know that $a, b \geq 1$, because $y_0(-1) \equiv (-1)^M = -1 \pmod{p, q}$.² Assume without loss of generality that $a \leq b$ (otherwise exchange the role of $p$ and $q$). Let us define two useful subgroups of $\mathbb{Z}_N^\ast$:

$$G := \{x \in \mathbb{Z}_N^\ast \mid y_{a-1}(x) \equiv 1 \pmod{p}\}$$

$$H := \{x \in \mathbb{Z}_N^\ast \mid y_{a-1}(x) \equiv 1 \pmod{p} \text{ and } y_{a-1}(x) \equiv 1 \pmod{q}\}$$

Note that $H$ can be equivalently defined as those $x$ for which $y_{a-1}(x) = 1$ in $\mathbb{Z}_N^\ast$. Convince yourself that these really are subgroups and that $H \leq G \leq \mathbb{Z}_N^\ast$.

**Claim:** $|G| = \frac{|\mathbb{Z}_N^\ast|}{2^a}$, and $H$ is a strict subgroup of $G$.

¹You can compute exactly how small it is, try it!

²Here we use that $M$ is odd, $p \neq 2$ and $q \neq 2$. 
Let us first convince ourselves that given the claim, the result follows. The idea is that when the random choice of the algorithm happens to yield an \( x \in G \setminus H \), then it will return \( p \). Here's why: In this case, we have

\[
y_{a-1}(x) \equiv 1 \quad (\text{mod } p)
\]
\[
y_{a-1}(x) \not\equiv 1 \quad (\text{mod } q)
\]

Suppose \( c \) is the smallest number such that \( y_c(x) = 1 \). Then clearly

\[
y_{c-1}(x) \equiv 1 \quad (\text{mod } p)
\]
\[
y_{c-1}(x) \not\equiv 1 \quad (\text{mod } q)
\]

In fact, we even know that \( y_{c-1}(x) \equiv -1 \quad (\text{mod } q) \), because \( q \) is a prime and \( y_{c-1}(x) \) is a square root of 1 in \( \mathbb{Z}_q \), but this particular detail is not needed. The point is that the algorithm will return the greatest common divisor when \( j = c \), and it will return \( \gcd(y_{c-1}(x) - 1, N) \). By the previous exercise, this is equal to \( p \).

So what is the probability that this happens? Again, assume that the claim above is true. If \( H \) is a strict subgroup of \( G \), then \( |H| \leq \frac{|G|}{2} \), since the size of a subgroup is a factor of the size of the group it is contained in. So then (implicitly conditioning on \( x \in \mathbb{Z}_N^* \)):

\[
\Pr[x \in G \setminus H] = \frac{|G \setminus H|}{|\mathbb{Z}_N^*|} = \frac{|G| - |H|}{|\mathbb{Z}_N^*|} \geq \frac{1}{2} \cdot \frac{|G|}{|\mathbb{Z}_N^*|} = \frac{1}{4}
\]

We can conclude that the algorithm returns \( p \) with probability at least \( \frac{1}{4} \). Repeating the algorithm often enough with independent random choices of \( x \), we expect to obtain \( p \) after at most 4 runs. If we run the algorithm \( n \) times, the probability of \emph{never} obtaining \( p \) drops exponentially with \( n \). It only remains to prove the claim above.

**Proof of the Claim:** By the Chinese Remainder Theorem, \( \mathbb{Z}_N^* \cong \mathbb{Z}_p^* \times \mathbb{Z}_q^* \). It is easy to see that this isomorphisms restricts to \( G \cong G' \times \mathbb{Z}_q^* \), where

\[
G' = \{ x \in \mathbb{Z}_p^* \mid x^{M \cdot 2^{a-1}} \equiv 1 \text{(mod } p) \}
\]

Consider the group homomorphism \( f : \mathbb{Z}_p^* \to \mathbb{Z}_q^* \) defined by \( f(x) = x^{M \cdot 2^{a-1}} \). We know that \( f(x)^2 \equiv 1 \) for all \( x \in \mathbb{Z}_p^* \) because of how \( a \) was defined. Another way to put this is to say that \( f(x) \) is a square root of 1. Since \( p \) is prime, \( \mathbb{Z}_p^* \) is the multiplicative group of a field, where \( 1 \) and \( -1 \) are the only square roots of 1. Therefore, the image of \( f \) is the set \( \{ \pm 1 \} \). It is easy to see that \( G' \) is the kernel of \( f \). Since \( f \) is a group homomorphism between finite groups, we have

\[
|\text{domain}(f)| = |\ker f| \cdot |\text{im} f|
\]

This implies \( |\mathbb{Z}_p^*| = |G'| \cdot 2 \), and via the isomorphism above \( |\mathbb{Z}_N^*| = |G| \cdot 2 \), which establishes the first part of the claim.

---

\(^3\) \( c \) depends on \( x \). We have \( a \leq c \leq b \).

\(^4\) Of course, there is also a chance that the algorithm returns \( q \). For the purpose of factoring \( N \), it doesn't matter which factor is returned, and so this can only increase the algorithm's chance of success. Note that I have made no attempt to estimate the probability that \( q \) is returned, and in fact, the probabilities are \emph{not} equal, and the proof is \emph{not} symmetric, because it does use \( a \leq b \).
For the second part of the claim, we simply have to show that $G \setminus H$ is non-empty, and to do that, it is sufficient to construct an element $x \in G \setminus H$. By the definition of $b$, there is an element $y \in \mathbb{Z}_q^*$ such that

$$y^{M \cdot 2^{b-1}} \not\equiv 1 \pmod{q}$$

Since $b \geq a$, this implies that

$$y^{M \cdot 2^{a-1}} \not\equiv 1 \pmod{q}$$

By the Chinese Remainder Theorem, let $x \in \mathbb{Z}_N^*$ such that

\[
\begin{align*}
    x &\equiv 1 \pmod{p} \\
    x &\equiv y \pmod{q}
\end{align*}
\]

We can compute that

\[
\begin{align*}
    x^{M \cdot 2^{a-1}} &\equiv 1 \pmod{p} \\
    x^{M \cdot 2^{a-1}} &\equiv y^{M \cdot 2^{a-1}} \not\equiv 1 \pmod{q},
\end{align*}
\]

from which we see that $x \in G \setminus H$, and so $G \setminus H$ is not empty. This means that $H$ is a proper subgroup of $G$, which establishes the second part of the claim and completes the proof.