$\mathbb{Z}_n^*$; hardness of squaring modulo a composite; and RSA

Leonid Reyzin

Notes for October 1, 2019

1 Multiplicative Inverses and $\mathbb{Z}_n^*$

We will denote by $\mathbb{Z}_n^*$ the set of values in $\mathbb{Z}_n$ that are relatively prime to n (that is, not 0 modulo p and not 0 modulo q). Note that the "coordinates" of $\mathbb{Z}_n^*$ are in $\mathbb{Z}_p^*$ and $\mathbb{Z}_q^*$, and that $\mathbb{Z}_n^*$ has $(p-1)(q-1)$ elements. Every element of $\mathbb{Z}_n^*$ has a multiplicative inverse modulo n (because it has a multiplicative inverse modulo p and modulo q, and thus modulo n by CRT) — in fact, $\mathbb{Z}_n^*$ contains all values in $\mathbb{Z}_n$ that have multiplicative inverses. $\mathbb{Z}_n^*$ is thus a group with the multiplication operation.

The size of $\mathbb{Z}_n^*$ is denoted by a special function ϕ called Euler's totient function: $|Z_n^*| = (p-1)(q-1) = \phi(n)$.

The size of $\mathbb{Z}_n^*$ is unknown to anyone who doesn't know the factorization of n. In fact, finding the factorization of n is computationally about as hard as finding $\phi(n)$ (because if you know p, q you can trivially compute $(p-1)(q-1)$, and if you know $n = pq$, and $\phi(n) = (p-1)(q-1)$, then you can find p and q by solving the quadratic equation $\phi(n) = (p-1)(n/p-1)$, i.e., $p\phi(n) = (p-1)(n-p)$).

Unlike the groups we had in this class until now, which had known size, $\mathbb{Z}_n^*$ a *group of unknown size* (or, more commonly, *group of unknown order*, because people use the word "order" instead of "size" when they talk about groups). Of course, the size is known to those who know the factorization of n. But even those who do not know the size of $\mathbb{Z}_n^*$ can still operate (i.e., multiply and divide) in $\mathbb{Z}_n^*$. So it is possible to operate in groups of unknown order.

2 Square Roots Modulo a Composite are as Hard as Factoring

We want to justify why we believe it's hard to compute x from x^2 modulo n. Indeed, let $s = r^2 \bmod n$. Then s has four square roots, as discussed above $\text{crt}(r_1, r_2), \text{crt}(-r_1, -r_2), \text{crt}(r_1, -r_2), \text{crt}(-r_1, r_2)$. Take two of these that are not negatives of each other, e.g., $r = \text{crt}(r_1, r_2)$ and $r' = \text{crt}(r_1, -r_2)$. Add them to get $r + r' = \text{crt}(2r_1, 0)$. Thus, $r + r' \equiv 0 \pmod q$, so $q|(r + r')$. Note also that $r + r' \not\equiv 0 \pmod p$, so $p \not| (r + r')$. Hence, $\gcd(r + r', n) = q$. Thus, if you know two such roots, you can factor n, by simply computing the greatest common divisor (this can be done quickly with Euclid's algorithm).

Now suppose we have an algorithm A that computes square roots modulo n. We will use it to factor n as follows: take a random $r \in \mathbb{Z}_n^*$, compute $s = r^2 \bmod n$, and give s to A. A will return some root r' of s. Because s has four roots and r was chosen at random (and not given to A), no matter how A works, $\Pr[r = \pm r'] = 1/2$. Hence, in half the cases, $\gcd(r + r', n)$ will give you a factor p or q of n.

3 RSA

3.1 The RSA function

RSA function [RSA78] is similar to modular squaring, but replaces exponent 2 with another power $e \neq 2$. The reason for doing so is to make sure the function is a bijection—i.e., the inverse is well-defined.

Before considering raising to the power e modulo a composite number $n = pq$, let us consider first raising to the power e modulo a prime p. Suppose $d = e^{-1} \bmod (p-1)$—i.e., $ed \equiv_{p-1} 1$. Such d exists whenever $\gcd(e, p-1) = 1$ (and can computed efficiently by extended Euclid's gcd algorithm).On HW2 we proved exponents work modulo $p-1$ when you operate modulo p, and therefore for any a, $a^{ed} \equiv_p a^1 \equiv_p a$.

Now doing this modulo n, suppose $ed \equiv 1 \pmod{p-1}$ and $ed \equiv 1 \pmod{q-1}$. Then if we let $a \in \mathbb{Z}_n$, a^{ed} is a both modulo p and modulo q, and hence is a modulo n, by Chinese Remainder Theorem.

So pick two primes $p \neq q$, let $n = pq$, and let e, d be such that $ed \equiv 1 \pmod{p-1}$ and $\pmod{q-1}$. Note that because e has an inverse modulo $p-1$ and $q-1$, it must be relatively prime with $p-1$ and $q-1$; in particular, e must be odd.

Let (n, e) be the public key and (n, d) be the corresponding secret key. The "easy" ("forward") direction of the RSA function is to take $x \in \mathbb{Z}_n$, and compute $y = x^e \,\%n$. The "hard" ("inverse") direction of the RSA function is to take $y \in Z_n$ and compute $x = y^d \,\%n$. RSA is an example of a "trapdoor permutation": it is a permutation (bijection) of Z_n such that the forward direction is easy given the public key, but the inverse direction is conjectured to be hard without the secret key.

By using an exponent e that is relatively prime with $p-1$ and $q-1$, we obtained a permutation (instead of, for example, exponent $e = 2$, which gives a 4-to-1 mapping). However, we gave up the equivalence to factoring: it is not known whether taking e-th roots modulo n is as hard as factoring for odd e. To be precise, we know that if taking e-th roots is hard, then factoring is hard (because if factoring were easy, then we could take e-th roots by taking them modulo p and q and combining them using CRT). The other direction is not known. We do know, however, that finding d from e is as hard as factoring [Ros17, Theorem 12.4]. So, if there is a way to find e-th roots without factoring, it must not find d.

References

[Ros17] Mike Rosulek. *The Joy of Cryptography*. 2017. `http://web.engr.oregonstate.edu/~rosulekm/crypto/`.

[RSA78] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, February 1978.