COMP 8920: Cryptography Winter 2025

Lecture 23 — April 1, 2025

Prof. Curtis Bright *Scribe: Rajat Yadav*

Recall the Euclidean algorithm run on a/b gives sequence of quotients $q_1, ..., q_m$

$$a/b = q_1 + \cfrac{1}{q_2 + \cfrac{1}{q_3 + \frac{1}{q_4 + \cdots}}}$$

which we denote $C([q_1, ..., q_m])$ and let C_j denote this truncated after j quotients, as j increases, C_j becomes closer to a/b. So, we can easily compute all convergents to a/b, and last class we showed that

$$|b/n - t/a| < 1/(3a^2)$$

which implies that t/a is a convergent of b/n. We just need to figure out which one. Since $\phi(n) = (ab-1)/t$ we can solve for $\phi(n)$ once t/a known (we just try all possibilities for t/a).

The Rabin Cryptosystem is a cryptosystem similar to RSA but is secure under the assumption n cannot be factored. Suppose p, q are primes $\equiv 3 \pmod 4$, let $pq = n$ and $\mathcal{P} = \mathbb{Z}_n^*$. Take

$$\mathcal{K} = \{(n, p, q) : p, q \equiv 3 \pmod 4\}$$

and define $e_k(x) = x^2 \bmod n$ and $d_k(y) = \sqrt{y} \bmod n$. n is the public key and (p, q) is the private key. Although Rabin is easier to describe than RSA, it has a drawback that there will be four square roots of $y \bmod n$. So, Bob cannot uniquely compute the plaintext from y, unless it has some form of redundancy. How can Bob decrypt y? By CRT, finding $x^2 \equiv y \pmod n$ is equivalent to solving

$$z^2 \equiv y \pmod p$$
$$z^2 \equiv y \pmod q$$

If Z is fixed, $\{\pm\sqrt{y} \equiv Z \pmod q, \pm\sqrt{y} \equiv Z \pmod p\}$ and all choices of $\pm$ give all square roots of y. How to solve $Z^2 \equiv y \pmod p$? When $p \equiv 3 \pmod 4$ there is a simple formula for Z. Note Euler's criterion says that $y^{(p-1)} \equiv 1 \pmod p$, since by construction y is a QR mod$p$. Multiply both sides by y:

$$y^{(p+1)/2} \equiv y \pmod p$$

and since $(p+1)/2$ is even, $(y^{(p+1)/4})^2 \equiv y \pmod p$, so $y^{(p+1)/4} \bmod p$ is a square root of $y \bmod p$. Similarly, $y^{(q+1)/4}$ is a square root of $y \bmod q$. If $p \equiv 1 \pmod 4$ there is no deterministic square root algorithm known, but a polynomial time Las Vegas algorithm is known.

We'll show Rabin is secure if n is hard to factor, i.e., if Rabin can be broken (i.e., square roots can be computed) then n can be factored. So we show that factoring n reduces to computing square roots mod$n$ or

$$\text{Factor}(n) \leq \sqrt{\text{in } \mathbb{Z}_n^*}$$

So suppose an oracle exists for computing square roots mod n. We'll give a Las Vegas algorithm to factor n with failure probability of at most $1/2$:

Choose a random $r \in \mathbb{Z}_n^*$

$$y := r^2 \bmod n$$

$$x := \sqrt{y} \bmod n$$

if $x \equiv \pm r \pmod{n}$ then return failure else return $\gcd(x + r, n)$ (a nontrivial factor).

Note $x^2 \equiv r^2 \pmod{n}$, but $x \not\equiv \pm r \pmod{n}$ so $n|(x - r)(x + r)$ but $n \nmid x - r$ and $n \nmid x + r$. Since $n = p \cdot q$, p, q prime, $pq|(x - r)(x + r)$ implies $pq|(x - r)$ or $pq|(x + r)$ or $(p|(x - r) \wedge q|(x + r))$ or vice versa. The first two cases contradict $n \nmid (x \pm r)$, so we have $p|(x - r)$ and $q \nmid (x - r)$ or vice versa. Thus, $\gcd(x - r, n)$ would be p or q. What's the success probability? Let w be a non-trivial square root of 1. Then $\{\pm r, \pm wr\}$ are the 4 roots of r^2. The oracle doesn't know the value of r, which was chosen randomly, and so it doesn't know which of the roots will lead to success. Half the roots it returns will lead to success (when it returns $\pm wr$), so with probability 1/2 the return value leads to a non-trivial gcd. This also shows Rabin is insecure against a chosen cipher-text attack, as with this algorithm would let them factor n, as the chosen cipher-text attack assumes a square root oracle exists.

Semantic Security: So far we've assumed Eve wants to break a cryptosystem by finding the secret/private key (called a total break). A partial break is when Eve can decrypt a previously unseen cipher-text (without the key) or learn some information about the plain-text given the cipher-text. Or, Eve might be able to distinguish between the encryption of different plain-texts or between a cipher-text and random string. The distinguishability problem is, given x_1, x_2, y with $e_K(x_i) = y$ for $i = 1$ or $i = 2$ to determine if $i = 1$ or $i = 2$. If encryption is done using a public key encryption, then randomness must be introduced in order to make this a difficult problem. If the problem cannot be solved with probability more than 1/2 the cryptosystem is said to be semantically secure. Achieving this is difficult, since it is a weak adversarial goal, and any bit of info leaked about the plain-text may make the distinguishability problem solvable. For example, RSA has the following partial break: Since b is coprime to $\phi(n) = (p - 1)(q - 1)$ which is even, b is odd. Then the Jacobi symbol $(y/n) = (x^b/n) = (x/n)^b = (x/n)$ since ± 1 raised to an odd number doesn't change. So, Eve can compute (x/n).

It is possible to show that RSA doesn't leak other information (assuming it is secure) like the value of $(x \bmod 2)$. To do this, you reduce the problem of decrypting RSA to the problem of computing $x \bmod 2$ from y, showing $\text{RSAdecryption}(y) \leq \text{parity}(y)$ where $\text{parity}(y) = d_k(y) \bmod 2$. It can be shown that if the distinguishability problem cannot be solved then no information of any kind is leaked about the plain-text, because any such info would allow solving the distinguishability problem. Since RSA is deterministic, Eve can solve this problem by simply computing $e_1(x), e_2(x)$, and checking which is $= y$. We introduce randomness to the cryptosystem with a random oracle $G : \mathbb{Z}_2^k \to \mathbb{Z}_2^m$ which in practice will be realized by a hash function.