EE 720: An Introduction to Number Theory and Cryptography (Spring 2019)

Lecture 20 — April 1, 2019

Instructor: Saravanan Vijayakumaran *Scribe: Saravanan Vijayakumaran*

1 Lecture Plan

- Miller-Rabin Primality Test
- Relating the RSA and Factoring Assumptions

2 Recap

- **Fermat's little theorem:** If p is a prime and a is any integer not divisible by p, then $a^{p-1} = 1 \bmod p$.
- For $a \in \{1, 2, \dots, N-1\}$, if $a \notin \mathbb{Z}_N^*$ then $a^{N-1} \neq 1 \bmod N$, i.e. such an a is a witness for the compositeness of N.
- But integers in the range $1, 2, \dots, N-1$ **not** belonging to $\mathbb{Z}_N^*$ are rare.
- For an integer N, we say that the integer $a \in \mathbb{Z}_N^*$ is a *witness for compositeness of* N if $a^{N-1} \neq 1 \bmod N$.
- **Theorem:** If there exists a witness (in $\mathbb{Z}_N^*$) that N is composite, then at least half the elements of $\mathbb{Z}_N^*$ are witnesses that N is composite.
- But there exist composite numbers for which $a^{N-1} = 1 \bmod N$ for all integers $a \in \mathbb{Z}_N^*$. These are called *Carmichael numbers.* The number $561 = 3 \cdot 11 \cdot 17$ is one such number.

2.1 Miller-Rabin Primality Test

- The Miller-Rabin algorithm takes two inputs: an integer p and a parameter t (in unary format) that determines the error probability. It runs in time polynomial in $\|p\|$ and t.
- **Theorem:** If p is prime, then the Miller-Rabin test always outputs "prime". If p is composite, then the algorithm outputs "composite" except with probability at most 2^{-t}.
- The algorithm for generating a random n-bit prime using the Miller-Rabin test is shown in Algorithm 1.
- **Lemma:** We say that $x \in \mathbb{Z}_N^*$ is a **square root of 1 modulo** N if $x^2 = 1 \bmod N$. If N is an odd prime, then the only square roots of 1 modulo N are $\pm 1 \bmod N$.[1]

[1] Note that $-1 \bmod N = N - 1 \in \mathbb{Z}_N^*$

Algorithm 1 Generating a random n-bit prime

Input: Length n

Output: A uniform n-bit prime

for $i = 1$ to $3n^2$ **do**
  $p' \leftarrow \{0,1\}^{n-2}$
  $p := 1\|p'\|1$
  Run the Miller-Rabin test on p
  if the output is "prime," **then**
    return p
return fail

- By Fermat's little theorem, if N is an odd prime $a^{N-1} = 1 \bmod N$ for all $a \in \{1, 2, \ldots, N-1\}$. Suppose $N - 1 = 2^r u$ where $r \geq 1$ is an integer and u is an odd integer. Then

$$a^u \bmod N,\ a^{2u} \bmod N,\ a^{2^2 u} \bmod N,\ a^{2^3 u} \bmod N,\ \ldots,\ a^{2^r u} \bmod N$$

  is a sequence where each element is the square of the previous element. In other words, each element is the square root modulo N of the next element. Since the last element in the sequence is a 1, by the above lemma the previous elements can only be ± 1. So one of two things can happen:

  - Either $a^u = 1 \bmod N$. In this case, the whole sequence has only ones.
  - Or one of $a^u \bmod N,\ a^{2u} \bmod N,\ a^{2^2 u} \bmod N,\ a^{2^3 u} \bmod N,\ \ldots,\ a^{2^{r-1} u} \bmod N$ is equal to -1.

- We say that $a \in \mathbb{Z}_N^*$ is a **strong witness that** N **is composite** if both the above conditions do not hold. If we can find even one strong witness, we can conclude that N is composite.

3 Miller-Rabin Primality Test (contd)

- We say that a integer N is a **prime power** if $N = p^r$ where $r \geq 1$.
- **Theorem:** Let $N > 1$ be an odd number that is not a prime power. Then at least half the elements of $\mathbb{Z}_N^*$ are strong witnesses that N is composite.
- **Proof outline:**
  - Let Bad $\subseteq \mathbb{Z}_N^*$ be the set of elements that are **not** strong witnesses.
  - We define a set Bad$'$ such that:
    1. Bad is a subset of Bad$'$.
    2. Bad$'$ is a strict subgroup of $\mathbb{Z}_N^*$. This implies that Bad$' \leq |\mathbb{Z}_N^*|/2$.

  As Bad $\subseteq$ Bad$'$, we get Bad $\leq$ Bad$' \leq |\mathbb{Z}_N^*|$. So at least half the elements of $\mathbb{Z}_N^*$ are strong witnesses.

- An integer N is a **perfect power** if $N = \hat{N}^e$ for integers $\hat{N}$ and $e \geq 2$. There exists a polynomial time algorithm to check that a given integer is a perfect power. If N is a perfect power, it is composite. If N is not a perfect power and it is not a prime, it cannot be a prime power. So the hypothesis of the above theorem will be satisfied.

Algorithm 2 The Miller-Rabin primality test

Input: Odd integer $N > 2$ and parameter 1^t
Output: A decision as to whether N is prime or composite

```
if N is a perfect power then
    return composite
Compute r ≥ 1 and odd u such that N − 1 = 2^r u
for j = 1 to t do
    a ← {0, ..., N − 1}
    if a^u ≠ ±1 mod N and a^(2^i u) ≠ −1 mod N for i ∈ {1, ..., r − 1} then
        return composite
return fail
```

- The Miller-Rabin test is given in Algorithm 2.

4 Revisiting RSA

4.1 The Factoring Assumption

- Let $\texttt{GenModulus}$ be a polynomial-time algorithm that, on input 1^n, outputs (N, p, q) where $N = pq$, and p and q are n-bit primes except with probability negligible in n.
- **The factoring experiment** $\texttt{Factor}_{\mathcal{A},\texttt{GenModulus}}(n)$:
    1. Run $\texttt{GenModulus}(1^n)$ to obtain (N, p, q).
    2. $\mathcal{A}$ is given N, and outputs $p', q' > 1$.
    3. The output of the experiment is 1 if $N = p'q'$, and 0 otherwise.
- We use p', q' in the above experiment because it is possible that $\texttt{GenModulus}$ returns composite integers p, q albeit with negligible probability. In this case, we could find factors of N other than p and q.
- **Definition: Factoring is hard relative to** $\texttt{GenModulus}$ if for all PPT algorithms $\mathcal{A}$ there exists a negligible function $\texttt{negl}$ such that $\Pr[\texttt{Factor}_{\mathcal{A},\texttt{GenModulus}}(n) = 1] \leq \texttt{negl}(n)$.
- The **factoring assumption** states that there exists a $\texttt{GenModulus}$ relative to which factoring is hard.

4.2 The RSA Assumption

- Let $\texttt{GenRSA}$ be a PPT algorithm that on input 1^n, outputs a modulus N that is the product of two n-bit primes, along with integers $e, d > 1$ satisfying $ed = 1 \bmod \phi(N)$.
- **The RSA experiment** $\texttt{RSA-inv}_{\mathcal{A},\texttt{GenRSA}}(n)$:
    1. Run $\texttt{GenRSA}(1^n)$ to obtain (N, e, d).
    2. Choose a uniform $y \in \mathbb{Z}_N^*$.

3. $\mathcal{A}$ is given N, e, y and outputs $x \in \mathbb{Z}_N^*$.
4. The output of the experiment is 1 if $x^e = y \bmod N$, and 0 otherwise.

- **Definition: The RSA problem is hard relative to** `GenRSA` if for all PPT algorithms $\mathcal{A}$ there exists a negligible function `negl` such that $\Pr[\texttt{RSA-inv}_{\mathcal{A},\texttt{GenRSA}}(n) = 1] \leq \texttt{negl}(n)$.

4.3 Relating the RSA and Factoring Assumptions

- For the RSA problem to be hard relative to `GenRSA`, the factoring problem must be hard relative to `GenModulus`.
- A PPT adversary who can factor N can win in the RSA experiment while remaining a PPT adversary.
- But it is not known whether an adversary who can win the RSA experiment can factor N.
- However, it is known that an adversary who can obtain d from N and e can factor N. See Theorem 8.50 for details.
- **Example:** Suppose a company wants to use the same modulus N for all its employees. To avoid one employee reading the messages meant for another, the company issues different (e_i, d_i) pairs to each employee but does not reveal the factorization of N to them. But this is insecure as knowledge of e_i, d_i can be used to factor N.

5 References and Additional Reading

- Sections 8.2.1, 8.2.2, 8.2.3, 8.2.4, 8.2.5 from Katz/Lindell