DD2448 Foundations of Cryptography
Lecture 10

Douglas Wikström
KTH Royal Institute of Technology
dog@kth.se

April 8, 2020

Last Lecture: Legendre Symbol (1/2)

Definition. Given an odd integer $b \geq 3$, an integer a is called a **quadratic residue** modulo b if there exists an integer x such that $a = x^2 \bmod b$.

Definition. The **Legendre Symbol** of an integer a modulo an **odd prime** p is defined by

$$\left(\frac{a}{p}\right) = \begin{cases} 0 & \text{if } a = 0 \\ 1 & \text{if } a \text{ is a quadratic residue modulo } p \\ -1 & \text{if } a \text{ is a quadratic non-residue modulo } p \end{cases} .$$

Last Lecture: Legendre Symbol (2/2)

Theorem. If p is an odd prime, then

$$\left(\frac{a}{p}\right) = a^{(p-1)/2} \bmod p \enspace .$$

Last Lecture: Legendre Symbol (2/2)

Theorem. If p is an odd prime, then

$$\left(\frac{a}{p}\right) = a^{(p-1)/2} \bmod p \ .$$

Proof.

- If $a = y^2 \bmod p$, then $a^{(p-1)/2} = y^{p-1} = 1 \bmod p$.

Last Lecture: Legendre Symbol (2/2)

Theorem. If p is an odd prime, then

$$\left(\frac{a}{p}\right) = a^{(p-1)/2} \bmod p \ .$$

Proof.

- If $a = y^2 \bmod p$, then $a^{(p-1)/2} = y^{p-1} = 1 \bmod p$.
- If $a^{(p-1)/2} = 1 \bmod p$ and b generates $\mathbb{Z}_p^*$, then $a^{(p-1)/2} = b^{x(p-1)/2} = 1 \bmod p$ for some x. Since b is a generator, $(p-1) \mid x(p-1)/2$ and x must be even.

Last Lecture: Legendre Symbol (2/2)

Theorem. If p is an odd prime, then

$$\left(\frac{a}{p}\right) = a^{(p-1)/2} \bmod p \ .$$

Proof.

- If $a = y^2 \bmod p$, then $a^{(p-1)/2} = y^{p-1} = 1 \bmod p$.
- If $a^{(p-1)/2} = 1 \bmod p$ and b generates $\mathbb{Z}_p^*$, then $a^{(p-1)/2} = b^{x(p-1)/2} = 1 \bmod p$ for some x. Since b is a generator, $(p-1) \mid x(p-1)/2$ and x must be even.
- If a is a non-residue, then $a^{(p-1)/2} \neq 1 \bmod p$, but $\left(a^{(p-1)/2}\right)^2 = 1 \bmod p$, so $a^{(p-1)/2} = -1 \bmod p$.

Last Lecture: Jacobi Symbol

Definition. The **Jacobi Symbol** of an integer a modulo an odd integer $b = \prod_i p_i^{e_i}$, with p_i prime, is defined by

$$\left(\frac{a}{b}\right) = \prod_i \left(\frac{a}{p_i}\right)^{e_i} \quad .$$

Note that we can have $\left(\frac{a}{b}\right) = 1$ even when a is a non-residue modulo b.

Properties of the Jacobi Symbol

Basic Properties.

$$\left(\frac{a}{b}\right) = \left(\frac{a \bmod b}{b}\right)$$

$$\left(\frac{ac}{b}\right) = \left(\frac{a}{b}\right)\left(\frac{c}{b}\right) \quad .$$

Law of Quadratic Reciprocity. If a and b are odd integers, then

$$\left(\frac{a}{b}\right) = (-1)^{\frac{(a-1)(b-1)}{4}}\left(\frac{b}{a}\right) \quad .$$

Supplementary Laws. If b is an odd integer, then

$$\left(\frac{-1}{b}\right) = (-1)^{\frac{b-1}{2}} \quad \text{and} \quad \left(\frac{2}{b}\right) = (-1)^{\frac{b^2-1}{8}} \quad .$$

Computing the Jacobi Symbol (1/2)

The following assumes that $a \geq 0$ and that $b \geq 3$ is odd.

$\textsc{Jacobi}(a, b)$
    if $a < 2$
        return a
    $s \leftarrow 1$
    while a is even
        $s \leftarrow s \cdot (-1)^{\frac{1}{8}(b^2-1)}$
        $a \leftarrow a/2$
    if $a < b$
        $\textsc{Swap}(\mathrm{a},\mathrm{b})$
        $s \leftarrow s \cdot (-1)^{\frac{1}{4}(a-1)(b-1)}$
    return $s \cdot \textsc{Jacobi}(a \bmod b, b)$

Solovay-Strassen Primality Test (1/2)

The following assumes that $n \geq 3$.

$\textsc{SolovayStrassen}(n, r)$

(1) **for** $i = 1$ **to** r

(2) Choose $0 < a < n$ randomly.

(3) **if** $\left(\frac{a}{n}\right) = 0$ or $\left(\frac{a}{n}\right) \neq a^{(n-1)/2} \bmod n$

(4) **return** *composite*

(5) **return** *probably prime*

Solovay-Strassen Primality Test (2/2)

Analysis.

- If n is prime, then $0 \neq \left(\frac{a}{n}\right) = a^{(n-1)/2} \bmod n$ for all $0 < a < n$, so we never claim that a prime is composite.

Solovay-Strassen Primality Test (2/2)

Analysis.

- If n is prime, then $0 \neq \left(\frac{a}{n}\right) = a^{(n-1)/2} \bmod n$ for all $0 < a < n$, so we never claim that a prime is composite.
- If $\left(\frac{a}{n}\right) = 0$, then $\left(\frac{a}{p}\right) = 0$ for some prime factor p of n. Thus, $p \mid a$ and n is composite, so we never wrongly return from within the loop.

Solovay-Strassen Primality Test (2/2)

Analysis.

- If n is prime, then $0 \neq \left(\frac{a}{n}\right) = a^{(n-1)/2} \bmod n$ for all $0 < a < n$, so we never claim that a prime is composite.
- If $\left(\frac{a}{n}\right) = 0$, then $\left(\frac{a}{p}\right) = 0$ for some prime factor p of n. Thus, $p \mid a$ and n is composite, so we never wrongly return from within the loop.
- At most half of all elements a in $\mathbb{Z}_n^*$ have the property that

$$\left(\frac{a}{n}\right) = a^{(n-1)/2} \bmod n \ .$$

More On Primality Tests

- The Miller-Rabin test is faster.
- Testing many primes can be done faster than testing each separately
- Those are *probabilistic* primality tests, but there is a *deterministic* test, so Primes are in P!

Security of RSA

Factoring

The obvious way to break RSA is to factor the public modulus N and recover the prime factors p and q.

- The number field sieve factors N in time

$$O\left(e^{(1.92+o(1))\left((\ln N)^{1/3}+(\ln\ln N)^{2/3}\right)}\right) \quad .$$

- The elliptic curve method factors N in time

$$O\left(e^{(1+o(1))\sqrt{2\ln p\ln\ln p}}\right) \quad .$$

Factoring

The obvious way to break RSA is to factor the public modulus N and recover the prime factors p and q.

- The number field sieve factors N in time

$$O\left(e^{(1.92+o(1))\left((\ln N)^{1/3}+(\ln\ln N)^{2/3}\right)}\right) .$$

- The elliptic curve method factors N in time

$$O\left(e^{(1+o(1))\sqrt{2\ln p\ln\ln p}}\right) .$$

Note that the latter only depends on the size of p!

Small Encryption Exponents

Suppose that $e = 3$ is used by all parties as encryption exponent.

- **Small Message.** If m is small, then $m^e < N$. Thus, **no reduction takes place**, and m can be computed in $\mathbb{Z}$ by taking the eth root.

Small Encryption Exponents

Suppose that $e = 3$ is used by all parties as encryption exponent.

- **Small Message.** If m is small, then $m^e < N$. Thus, **no reduction takes place**, and m can be computed in $\mathbb{Z}$ by taking the eth root.
- **Identical Plaintexts.** If a message m is encrypted under moduli N_1, N_2, N_3, and N_4 as c_1, c_2, c_3, and c_4, then CRT implies a $c \in \mathbb{Z}^*_{N_1 N_2 N_3 N_4}$ such that $c = c_i \bmod N_i$ and $c = m^e \bmod N_1 N_2 N_3 N_4$ with $m < N_i$.

Additional Caveats

- **Identical Moduli.** If a message m is encrypted as c_1 and c_2 using distinct encryption exponents e_1 and e_2 with $\gcd(e_1, e_2) = 1$, and a modulus N, then we can find a, b such that $ae_1 + be_2 = 1$ and $m = c_1^a c_2^b \bmod N$.

Additional Caveats

- **Identical Moduli.** If a message m is encrypted as c_1 and c_2 using distinct encryption exponents e_1 and e_2 with $\gcd(e_1, e_2) = 1$, and a modulus N, then we can find a, b such that $ae_1 + be_2 = 1$ and $m = c_1^a c_2^b \bmod N$.
- **Reiter-Franklin Attack.** If e is small then encryptions of m and $f(m)$ for a polynomial $f \in \mathbb{Z}_N[x]$ allows efficient computation of m.

Additional Caveats

- **Identical Moduli.** If a message m is encrypted as c_1 and c_2 using distinct encryption exponents e_1 and e_2 with $\gcd(e_1, e_2) = 1$, and a modulus N, then we can find a, b such that $ae_1 + be_2 = 1$ and $m = c_1^a c_2^b \bmod N$.
- **Reiter-Franklin Attack.** If e is small then encryptions of m and $f(m)$ for a polynomial $f \in \mathbb{Z}_N[x]$ allows efficient computation of m.
- **Wiener's Attack.** If $3d < N^{1/4}$ and $q < p < 2q$, then N can be factored in polynomial time with good probability.

Factoring From Order of Multiplicative Group

Given N and $\phi(N)$, we can find p and q by solving

$$\begin{aligned} N &= pq \\ \phi(N) &= (p-1)(q-1) \end{aligned}$$

Factoring From Encryption & Decryption Exponents (1/3)

- If $N = pq$ with p and q prime, then the CRT implies that

  $$x^2 = 1 \bmod N$$

  has **four distinct solutions** in $\mathbb{Z}_N^*$, and **two** of these are **non-trivial**, i.e., distinct from ± 1.

Factoring From Encryption & Decryption Exponents (1/3)

- If $N = pq$ with p and q prime, then the CRT implies that

$$x^2 = 1 \bmod N$$

  has **four distinct solutions** in $\mathbb{Z}_N^*$, and **two** of these are **non-trivial**, i.e., distinct from ± 1.

- If x is a non-trivial root, then

$$(x-1)(x+1) = tN$$

  but $N \nmid (x-1), (x+1)$, so

$$\gcd(x-1, N) > 1 \quad \text{and} \quad \gcd(x+1, N) > 1 \enspace .$$

Factoring From Encryption & Decryption Exponents (2/3)

- The encryption & decryption exponents satisfy

$$ed = 1 \bmod \phi(N) \ ,$$

so if we have $ed - 1 = 2^s r$ with r odd, then

$$\begin{aligned}(p-1) &= 2^{s_p} r_p \text{ which divides } 2^s r \quad \text{and}\\(q-1) &= 2^{s_q} r_q \text{ which divides } 2^s r \ .\end{aligned}$$

- If $v \in \mathbb{Z}_N^*$ is random, then $w = v^r$ is random in the subgroup of elements with order 2^i for some $0 \leq i \leq \max\{s_p, s_q\}$.

Suppose $s_p \geq s_q$. Then for some $0 < i < s_p$,

$$w^{2^i} = \pm 1 \bmod q$$

and

$$w^{2^i} \bmod p$$

is uniformly distributed in $\{1, -1\}$.

Conclusion.
$w^{2^i} \pmod{N}$ is a non-trivial root of 1 with probability $1/2$, which allows us to factor N.

CPA Security

CPA Security (1/3)

- RSA clearly provides some kind of "security", but it is clear that we need to be more careful with what we ask for.

CPA Security (1/3)

- RSA clearly provides some kind of "security", but it is clear that we need to be more careful with what we ask for.
- Intuitively, we want to leak no information of the encrypted plaintext.

CPA Security (1/3)

- RSA clearly provides some kind of "security", but it is clear that we need to be more careful with what we ask for.
- Intuitively, we want to leak no **knowledge** of the encrypted plaintext.

CPA Security (1/3)

- RSA clearly provides some kind of "security", but it is clear that we need to be more careful with what we ask for.
- Intuitively, we want to leak no **knowledge** of the encrypted plaintext.
- In other words, no function of the plaintext can efficiently be guessed notably better from its ciphertext than without it.

CPA Security (2/3)

$\mathbf{Exp}^{b}_{\mathcal{CS},A}$ **(CPA Security Experiment).**

1. **Generate Public Key.** $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^n)$.
2. **Adversarial Choice of Messages.** $(m_0, m_1, s) \leftarrow A(\mathsf{pk})$.
3. **Guess Message.** Return the first output of $A(\mathsf{E}_{\mathsf{pk}}(m_b), s)$.

CPA Security (2/3)

$\mathrm{Exp}^b_{\mathcal{CS},A}$ **(CPA Security Experiment).**

1. **Generate Public Key.** $(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{Gen}(1^n)$.
2. **Adversarial Choice of Messages.** $(m_0, m_1, s) \leftarrow A(\mathsf{pk})$.
3. **Guess Message.** Return the first output of $A(\mathsf{E}_{\mathsf{pk}}(m_b), s)$.

Definition. A cryptosystem $\mathcal{CS} = (\mathsf{Gen}, \mathsf{E}, \mathsf{D})$ is said to be **CPA secure** if for every polynomial time algorithm A

$$|\Pr[\mathrm{Exp}^0_{\mathcal{CS},A} = 1] - \Pr[\mathrm{Exp}^1_{\mathcal{CS},A} = 1]|$$

is negligible.