CS 359C – Classics of Cryptography April 19, 2017

Lecture 3: Number-Theoretic Cryptography

Instructor: Henry Corrigan-Gibbs, David Wu *Scribe: Mark Matthew Anderson*

Review from Last Week

– Two ways to build crypto schemes:
  1) Use assumptions (e.g. factoring is hard)
  2) Change the model

– Even-Mansour Cipher
  - Uses random permutation model
  - All parties have access to Π/Π^{-1} random permutations
  - In practice, $\hat{\Pi}$ is coded into standard

  Even-Mansour security proof (used hybrid argument)
    Game 0: Real attack game (adversary talks to EM cipher)
    Game 1: Rephrasing
    Game 2: Ideal World (adversary talks to random/ideal cipher)

– Time/space Tradeoffs (Hellman Tables)
  "Inverting a function with advice"
  $[N] : \{1, \ldots, N\} : 2^n$
  Given: $f : [N] \to [N]$,
    $y \in [N]$,
    s bits of "advice" $\to$ precomputation
  Task: find $x \in [N]$ such that $y = f(x)$
  Theorem (Hellman): With $s \in \mathcal{O}(N^{2/3})$ bits of advice, can invert f in time $\mathcal{O}(N^{2/3})$
    $\Rightarrow$ Inverting DES takes $\approx 2^{40}$ time (keys: 2^{56})

– Collision Finding

• Meet in the Middle	space: $\mathcal{O}(\sqrt{N})$	time: $\mathcal{O}(\sqrt{N})$
• Rho Method	space: $\mathcal{O}(1)$	time: $\mathcal{O}(\sqrt{N})$
• Parallel Rho (P processors)	space: $\mathcal{O}(1)$ (per processor)	time: $\mathcal{O}(\sqrt{N}/P)$

RSA

– First public key encryption and digital signatures
– RSA assumptions have more structure than other assumptions
– Going out of style
  - Quantum algorithms can break all assumptions
  - Large keys (λ^3-bit keys $\approx$ 4096 bits)

A Survey of Hard Problems (Related to RSA)

Factoring

Sample $p, q \xleftarrow{R} \{\lambda\text{-bit primes}\}$
$N \leftarrow p \cdot q$
Given N, produce (p, q)
Best attack: $e^{\mathcal{O}(\lambda^{1/3} \cdot (\log \lambda)^{2/3})} \notin$ polynomial time
$\ll 2^{\lambda}$
General Number Field Sieve (Pollard 1988)

RSA-e (e is an Odd Prime)

Sample $p, q \xleftarrow{R} \{\lambda\text{-bit primes}\}$
$gcd(e, p-1) = 1$, $gcd(e, q-1) = 1$
$N \leftarrow p \cdot q$
$x \xleftarrow{R} \mathbb{Z}_N$
$a \leftarrow x^e \in \mathbb{Z}_N$
Given (N, a) produce x
often: $e = 3, e = 65537$
"Taking e^{th} roots mod N is hard without the factors of N"

Strong RSA Problem

Sample $p, q \xleftarrow{R} \{\lambda\text{-bit primes}\}$
$gcd(e, p-1) = 1$, $gcd(e, q-1) = 1$
$N \leftarrow p \cdot q$
$a \xleftarrow{R} \mathbb{Z}_N$
Given (N, a) produce (x, e) such that
$a = x^e \in \mathbb{Z}_N$ and
$e \neq \pm 1$

Hardness

Factoring $\geq$ RSA-e $\geq$ Strong-RSA
RSA-e has unique answer
Strong-RSA has exponential answers

Random Self Reduction

For a given modulus N, we'd like that computing $a^{1/e}$ mod N is hard for "almost all" $a \in \mathbb{Z}_N$.
"Hard on average"

We know that for some $a \in \mathbb{Z}_N$, computing $a^{1/e}$ mod N is easy!
$\rightarrow a = 1$, numbers with cube roots over the integers

We can show that either:
a) finding $a^{1/e}$ mod N is hard for "almost all" $a \in \mathbb{Z}_N$ or
b) finding $a^{1/e}$ mod N is easy everywhere

Claim:
Say there exists an efficient algorithm $\mathcal{A}_N$ such that

$$\Pr_{a \xleftarrow{R} \mathbb{Z}_N} [\mathcal{A}_N(a) = a^{1/e} \in \mathbb{Z}_N] = \epsilon$$

then there exists an efficient algorithm $\mathcal{B}_N$ such that for all $x \in \mathbb{Z}_N$

$$\Pr_{random\, coins\, of\, \mathcal{B}_N} [\mathcal{B}_N(x) = x^{1/e} \in \mathbb{Z}_N] = \epsilon$$

Proof.

$\mathcal{B}_N(x)$ { $r \xleftarrow{R} \mathbb{Z}_N$
$y \leftarrow \mathbb{A}_N(x \cdot r^e)$
$z \leftarrow y \cdot r^{-1} \in \mathbb{Z}_N$
if $z^e \neq x$: output "$fail$"
else output z
}

$Pr[\overline{fail}] = \Pr_a[\mathcal{A}_N(a) = a^{1/e} \in \mathbb{Z}_N] = \epsilon$ ■
– caveat: only works for some N

Crypto from Factoring

Trapdoor One-Way Function

$(pk, sk) \leftarrow Gen(1^\lambda)$
$y \leftarrow F(pk, x) \quad x \in \mathcal{X}, y \in \mathcal{Y}$
$x \leftarrow F^{-1}(sk, y)$

Correctness: For all (pk, sk) from Gen,
for all $x \in \mathcal{X}$,
$F(pk, F^{-1}(sk, y)) = y$

Security: For all efficient adversaries $\mathcal{A}$
$TDFAdv[\mathcal{A}, F] := Pr[y = F(pk, x')]$
$TDFAdv[\mathcal{A}, F] \in negl(\lambda)$

Rabin (1979)

At a high level, this is just RSA with $e = 2$

RSA: $x^e \bmod N$

Rabin: $x^2 \bmod N$

$(N, p) \leftarrow Gen(1^\lambda)$

$y \leftarrow F(N, x \in \mathbb{Z}_N)$

returns $x^2 \bmod N$

$x \leftarrow F^{-1}(p, y)$

returns $\sqrt{y} \bmod N$

– collisions: $(-x)^2 = x^2$

Chinese Remainder Theorem (CRT)

Given primes p and q, $p \neq q$, and given x_p and x_q such that

$x_p = x \bmod p$

$x_q = x \bmod q$

there is an algorithm that outputs

$x \bmod N \to x \bmod pq$

Square Roots

Claim:

If $p \equiv 3 \bmod 4$, then

$p = 4p' + 3$

$x = y^{\frac{p+1}{4}} \bmod p$

is a square root of y in $\mathbb{Z}_p^*$

Proof.

$$
\begin{aligned}
x^2 = (y^{\frac{p+1}{4}})^2 = y^{\frac{p+1}{2}} &= y \cdot y^{\frac{p-1}{2}} \\
\text{(let } y = r^2\text{)} \qquad &= y \cdot (r^2)^{\frac{p-1}{2}} \\
&= y \cdot r^{p-1} \in \mathbb{Z}_p \\
&= y \in \mathbb{Z}_p \blacksquare
\end{aligned}
$$

Also easy (not as easy) if $p \equiv 1 \bmod 4$

If x is root of y, $(p - x)$ is also:

$$
\begin{aligned}
(p - x)^2 &= p^2 - 2px + x^2 \\
&= x^2 \bmod p
\end{aligned}
$$

$\longrightarrow$ There will be four square roots mod N if any square roots.

Rabin and Factoring

Claim:

Given an efficient algorithm $\mathcal{A}$ that inverts Rabin's function, there exists an efficient algorithm $\mathcal{B}$ that factors N.

We have x, x' such that

$$x^2 = (x')^2 \bmod N$$
$$x^2 - (x')^2 = 0 \bmod N$$
$$(x - x')(x + x') = 0 \bmod N$$

if $x = \pm x'$: $x - x' = 0 \in \mathbb{Z}$

$x + x' = 0 \in \mathbb{Z}$

else ($x \neq \pm x'$) then

$(x - x')(x + x') = k \cdot N$

$\rightarrow gcd(x - x', N)$ gives factor of N

Four cases:

$x = x' \bmod p$	$x = x' \bmod q$	$\rightarrow$ not useful
$x = x' \bmod p$	$x \neq x' \bmod q$	$\rightarrow$ useful
$x \neq x' \bmod p$	$x = x' \bmod q$	$\rightarrow$ useful
$x \neq x' \bmod p$	$x \neq x' \bmod q$	$\rightarrow$ not useful

Another View of RSA Problems

(Rabin)

$a \stackrel{R}{\leftarrow} \mathbb{Z}_N$

find a root of $f(x) = x^2 - a \in \mathbb{Z}_N$

(RSA)

$a \stackrel{R}{\leftarrow} \mathbb{Z}_N$

find a root of $f(x) = x^e - a \in \mathbb{Z}_N$

(Crazy RSA)

$a \stackrel{R}{\leftarrow} \mathbb{Z}_N$

find a root of $f(x) = x^7 + 4x^2 + 2x + a \in \mathbb{Z}_N$

Only (known) way to solve these without factors of N is to solve over the integers and reduce mod N