CPSC 420 Lecture 30: Today's announcements:

- Examlet 4 is OPTIONAL.
  I will use the best 3 examlet marks for grading.
- Examlet 4 on April 5 in class. Closed book & no notes
- Reading: Cuckoo Hashing for Undergraduates [by Pagh]
- Reading: RSA public-key cryptosystem [Intro to Algs 4th Ed. by Cormen, Leiserson, Rivest, Stein Ch.31.7]

Today's Plan

- Cuckoo Hashing
- RSA cryptosystem

Cuckoo Rehash

insert(x)
1. if $T[h_1(x)] = x$ or $T[h_2(x)] = x$ return
2. $i \leftarrow h_1(x)$
3. repeat n times
4. $\quad y \leftarrow T[i]$
5. $\quad T[i] \leftarrow x$
6. $\quad$ if $y =$ NULL return
7. $\quad$ if $i = h_1(y)$ then $i \leftarrow h_2(y)$ else $i \leftarrow h_1(y)$
8. $\quad x \leftarrow y$
9. rehash; insert(x)

Lemma 3: If $m \geq 2cn$ then the probability of a cycle in the cuckoo graph after n insertions is at most $\frac{1}{c-1}$.

Proof: Slot i is involved in a cycle iff there is a path from i to itself of length $\ell \geq 1$. By Lemma 1, this happens with probability $\leq \sum_{\ell=1}^{\infty} \frac{1}{c^{\ell} m} = \frac{1}{(c-1)m}$. Summing over all m slots, gives probability $\leq \frac{1}{c-1}$ for a cycle. $\square$

Cuckoo Rehash

Lemma 3: If $m \geq 2cn$ then the probability of a cycle in the cuckoo graph after n insertions is at most $\frac{1}{c-1}$.

Proof: Slot i is involved in a cycle iff there is a path from i to itself of length $\ell \geq 1$. By Lemma 1, this happens with probability $\leq \sum_{\ell=1}^{\infty} \frac{1}{c^{\ell} m} = \frac{1}{(c-1)m}$. Summing over all m slots, gives probability $\leq \frac{1}{c-1}$ for a cycle. $\square$

Each rehash takes $O(n)$ time.

By Lemma 3, for $c > 3$, the prob. that one rehash occurs after n insertions is $\leq 1/2$, that two rehashes occur $\leq 1/4$, etc. So expected amortized cost of rehash is $O(1)$.

Note: A rehash triggers $k > 0$ consecutive rehashes with prob. $\leq 1/2^k$. So the expected cost is still $O(n) \cdot \sum_{k=1}^{\infty} 1/2^k = O(n)$.

Cryptography

Cryptography

Alice **encrypts** her message M and sends encrypted version to Bob.
Bob **decrypts** to get original message.

Possible cryptosystems

One-time pad Alice and Bob agree beforehand on a *random* n-bit string P (the pad).

Alice sends $M \oplus P$ (bitwise exclusive or) to Bob.
Bob decrypts $(M \oplus P) \oplus P = M$

Alice	Bob
$M = 1011011$	$M \oplus P = 1100001$
$P = 01110101101\ldots$	$P = 01110101101\ldots$
$M \oplus P = 1100001$	$(M \oplus P) \oplus P = 1011011$

Cryptography

Alice **encrypts** her message M and sends encrypted version to Bob. Bob **decrypts** to get original message.

Possible cryptosystems

One-time pad Alice and Bob agree beforehand on a *random* n-bit string P (the pad).

Alice sends $M \oplus P$ (bitwise exclusive or) to Bob.

Bob decrypts $(M \oplus P) \oplus P = M$

Good: Information theoretically secure. Eve gets no information about M. Given $M \oplus P$, any message M is equally likely.

Bad: Can use just once. $(M_1 \oplus P) \oplus (M_2 \oplus P) = M_1 \oplus M_2$ leaking information

RSA public/private key cryptosystem [Rivest,Shamir,Adleman '77]

Bob has two functions: secret $S_B()$ and public $P_B()$

Properties:

1. $S_B(P_B(M)) = M$ and $P_B(S_B(M)) = M$
2. Hard to find M given $P_B(M)$ without $S_B()$

$\boxed{\begin{array}{l}\text{Alice sends } P_B(M) \text{ to Bob.} \\ \text{Bob decrypts: } S_B(P_B(M)) = M\end{array}}$

Good: Use again and again
Bad: No one knows if it's secure.

factoring easy $\Rightarrow$ RSA breakable.
factoring hard $\Rightarrow$ RSA secure? (unknown)

Digital Signatures:

RSA public/private key cryptosystem [Rivest,Shamir,Adleman '77]

Bob has two functions: secret $S_B()$ and public $P_B()$

Properties:

1. $S_B(P_B(M)) = M$ and $P_B(S_B(M)) = M$
2. Hard to find M given $P_B(M)$ without $S_B()$

Alice sends $P_B(M)$ to Bob.
Bob decrypts: $S_B(P_B(M)) = M$

Good: Use again and again
Bad: No one knows if it's secure.

factoring easy $\Rightarrow$ RSA breakable.
factoring hard $\Rightarrow$ RSA secure? (unknown)

Digital Signatures: Alice sends $(M, \sigma = S_A(M))$ to Bob
Bob can check that $P_A(\sigma) = M$.

Constructing public/private keys

1. Select two large (> 2048 bits) prime numbers p and q.

   $p = 31 \;\; q = 17$

2. Compute $n = p \cdot q$

   $n = 527$

3. Select a small odd integer e **relatively prime** to $\phi(n) \triangleq (p-1)(q-1)$ i.e. $\gcd(e, \phi(n)) = 1$.

   $\phi(n) = 30 \cdot 1\overset{6}{5} = 480$

   480

   $e = 7 \; (\gcd(7, 480) = 1)$

4. Compute $d = e^{-1} \pmod{\phi(n)}$ i.e. $ed = 1 \pmod{\phi(n)}$

   solve $7d = 1 \pmod{480}$

   $7d + 480c = 1$ (and $0 \le d < 480$)

   extended gcd given a,b finds x,y with

   $ax + by = \gcd(a, b)$

   $7 \cdot \underset{d}{343} + 480 \cdot \underset{c}{(-5)} = 1$

5. Public key $P = (e, n)$ Private key $S = (d, n)$

   $P = (7, 527) \;\; S = (343, 527)$

6. $\boxed{P(M) = M^e \pmod{n}}$ $\boxed{S(C) = C^d \pmod{n}}$

How does this work?

Theorem

For all $M < n$, $P(S(M)) = S(P(M)) = M$

Proof.

$P(S(M)) = S(P(M)) = M^{ed} \pmod{n}$. Since $ed = 1 \pmod{\phi(n)}$,
$e \cdot d = 1 + k(p-1)(q-1)$ for integer k.
If $M \neq 0 \pmod{\phi(n)}$ then

$(p-1)(q-1)$

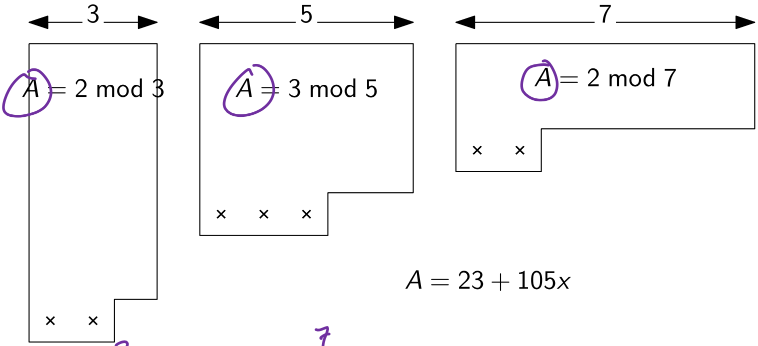
$$
\begin{aligned}
M^{ed} &= M(\boxed{M^{p-1}})^{k(q-1)} \pmod{p} \\
&= M(1)^{k(q-1)} \pmod{p} && \text{Fermat's little thm} \\
&= M \pmod{p}
\end{aligned}
$$

1

If $M = 0 \pmod{\phi(n)}$ then $M^{ed} = M \pmod{p}$ as well.
Similarly, $\boxed{M^{ed}} = M \pmod{q}$.
By Chinese Remainder Thm, $M^{ed} = M \pmod{n}$ for all $M < n$. ☐

Chinese Remainder Theorem [Sun-Tzu 300AD]

"Looks like the army has between 400 and 500 soldiers."



$$A = 23 + 105x$$

Let $n = \overset{3}{n_1} \times \overset{5}{n_2} \times \cdots \times \overset{7}{n_r}$ where n_i are pairwise relatively prime.
Then $\underbrace{(a_1, a_2, \ldots, a_r)}_{(2,3,2)}$ uniquely determines a mod n where
$a_i = a \bmod n_i$

Please fill out course evaluations

I read them.

I change.

Future students thank you.