Introduction to Cryptography: Homework #7

Due on July 12, 2019 at 11:59pm

Professor Manuel

ShiHan Chan

Problem 1

1.1

There are three main algorithms: the key generator, the encryption algorithm, and the decryption algorithm. The encryption algorithm: Bob will send Alice plaintext m. And Bob maps m to an element of G. One random number r is chosen from 0.....p-1. Last we calculate $u_1 = g_1^r$, $u_2 = g_2^r$, $e = h^r m$, $a = H(u_1, u_2, e)$ and $v = e^k d^{aR}$. Then ciphertext c= (u_1, u_2, e, v) will be given to Alice.

1.2

The decryption algorithm: If $u_1^{x_1+ay_1} u_2^{x_2 y y_2 a} = v$, the plaintext m would be $\frac{e}{u_1^z}$. The decryption will fail in other way.

1.3

The generation of the key algorithm: Initially, a cyclic group G with order p is generated by Alice. Then she finds two key generators g1 and g2 for G. And Alice chooses x_1, x_2, y_1, y_2, z from 0.....p-1 and computes ciphertext$g_1^{x_1}$,$g_2^{x_2}$, d=$g_1^{y_1} g_2^{y_2}$, h=g_1^z. H is a one-way trapdoor function be generated. Lastly, private keys are x_1, x_2, y_1, y_2, z and c,d,h,G,p,g_1, g_2 are public keys.

2.

Chosen ciphertext attacks can be used since the attacker can conduct different ciphertexts with plaintext to attack the cipher. But decryption algorithm don't allow all ciphertexts created by the attacker because of one way collision-resistant hash function H. That explains why cipher is safe.

3.

The similarity is that calculation is in symmetric group is both hard to decrypt due to discrete logarithm problems.

The difference is that another protecting layer is added for Cramer algorithm (trpdoor one-way hash function), and it can restrict the input of cipher,and that explains why Cramer-Shoup system is safer.

Problem 2

1.
p is prime and p doesn't divide α means gcd(p,α)=1 and we can get $\alpha^{p-1} \equiv 1 \mod p$ by Euler's theorem. And the hash function isn't second pre-image resistant, given x, we can find $x' = x + (p-1)$ such that $h(x) = h(x')$. And it's not collision resistant too. For all x, we can find $x' = x + (p-1)$ such that $h(x) = h(x')$. So it's not a good cryptographic hash function.
2.
$2^{30} = 0x40000000$
$floor(2^{30} * \sqrt{2}) = 5A827999$
$floor(2^{30} * \sqrt{3}) = 6ED9EBA1$
$floor(2^{30} * \sqrt{5}) = 8F1BBCDC$
$floor(2^{30} * \sqrt{10}) = CA62C1D6$
And the result is same as $K_0||.....||K19, K20||.....||K39, K40||.....||K59 and K60||.....||K79$.

Problem 3

1.
g(x) = ln(1x)+x+x2
$g'(x) = -\frac{1}{1-x} + 1 + 2x$
When x=0 or 0.5, g'(x)=0
$g''(x) = -\frac{1}{x-1}^2 + 2$
g(0)=1 is local min, g(0.5)=-2 is local max.
When $x \in [0, 0.5]$, $g(x) \in [g(0), g(0.5)] >= 0$
In the same method we set h(x) = ln(1 x) + x
we can get $h(x) \in [g(0.5), g(0)] <= 0$
All in all, $xx^2 <= ln(1x) <= x$
2.

Due to $j \in [1, r-1]$ and $r \leq \frac{n}{2}$, we can show that $\frac{j}{n} \in \left[0, \frac{1}{2}\right]$
$-\frac{j}{n} - \left(\frac{j}{n}\right)^2 \leq \ln\left(1 - \frac{j}{n}\right) \leq -\frac{j}{n}$
$\sum_{j=1}^{r-1}\left[-\frac{j}{n} - \left(\frac{j}{n}\right)^2\right] \leq \sum_{j=1}^{r-1} \ln\left(1 - \frac{j}{n}\right) \leq \sum_{j=1}^{r-1} -\frac{j}{n}$
$-\frac{(r-1)r}{2n} - \frac{(r-1)r(2r-1)}{6n^2} \leq \sum_{j=1}^{r-1} \ln\left(1 - \frac{j}{n}\right) \leq -\frac{(r-1)r}{2n}$
When $r > 1$,
$\frac{(r-1)r(2r-1)}{6n^2} = \frac{r^3 - \frac{3}{2}r^2 + r}{3n^2} < \frac{r^3}{3n^2}$
So we know that
$-\frac{(r-1)r}{2n} - \frac{r^3}{3n^2} \leq \sum_{j=1}^{r-1} \ln\left(1 - \frac{j}{n}\right) \leq -\frac{(r-1)r}{2n}$
3.
Exponentiate the inequation in previous part, we can get:
$\exp\left(-\frac{(r-1)r}{2n} - \frac{r^3}{3n^2}\right) \leq \prod_{j=1}^{r-1}\left(1 - \frac{j}{n}\right) \leq \exp\left(-\frac{(r-1)r}{2n}\right)$

Set $\lambda = \frac{r^2}{2n}$, $c_1 = \sqrt{\frac{\lambda}{2}} - \frac{(2\lambda)^{3/2}}{3}$, $c_2 = \sqrt{\frac{\lambda}{2}}$.
$-\lambda + \frac{c_1}{\sqrt{n}} = -\frac{r^2}{2n} + \frac{r}{2n} - \frac{r^3}{n^2} = -\frac{(r-1)r}{2n} - \frac{r^3}{3n^2}$
$-\lambda + \frac{c_2}{\sqrt{n}} = -\frac{r^2}{2n} + \frac{r}{2n} = -\frac{(r-1)r}{2n}$

Finally we can get:
$e^{-\lambda}e^{c_1/\sqrt{n}} \leq \prod_{j=1}^{r-1}\left(1 - \frac{j}{n}\right) \leq e^{-\lambda}e^{c_2/\sqrt{n}}$ 4.
When n is large and $\lambda = \frac{r^2}{2n} < \frac{n}{8}$, $r < \frac{n}{2}$
Because λ is constant, c_1 and c_2 are all constants.
$\lim n- > \inf e^{\frac{c_1}{\sqrt{n}}} = 1$
$\lim n- > \inf e^{\frac{c_2}{\sqrt{n}}} = 1$
So we can get: $\sum_{j=1}^{j=r-1}(1 - \frac{j}{n}) = e^{-\lambda}$

Problem 4

1.
The probability is $1 - \sum_{i=1}^{i=39} \frac{1000-i}{1000} = 0.546$ by birthday paradox.
2.
$40 * (\frac{1}{1000}) * (\frac{999}{1000})^3 9 = 0.039$
3.
It's relatively easy to find a collision in a hash function, but it's comparatively hard to find a collision of a

specific massage. It implies that Alice can overcome the problem by changing a bit in the massage and Eve cannot find a collision of new massage easily.

Problem 5

1.
O(loga)*O(logb)=O(log(ab)) 2.
function fastmodularexponentiation(α, β, a, b, n)
sa = sizeofbase(a, 2)
sb = sizeofbase(b, 2)
k = max(sa, sb)
res = 1
for i = k 1 to 0:
res = res*res mod n
if bit(a, i) = 1:
res = res *α mod n
if bit(b, i) = 1:
res = res * β mod n
return res
3.
3l squaring and multiplications are necessary to compute $\alpha^a \beta^b \mod n$.
4.
programming problem, implement in C

```
#include <stdio.h>
int power(int x, int y,int p)
{
x=x%p;
int res = 1;
while (y > 0)
{
if (y %2== 1) {res = (res*x)%p;}
y = y/2;
x = (x*x)*p;
}
return res;
}
int main()
{
int alpha = 15;
int a = 7;
int p = 12;
printf("result is %u", power(alpha, a, p));
return 0;
}
```

Problem 6

hw6 problem 5
1.(a)
Suppose the map s is not injective, $\exists x \neq x'$ such that $y = y'$. Than we use this method to show: Set $y_0 = y$, if $y_{0,|y_0|-1}||y_{0,|y_0|} = 01$, then $x_{|x|} = x'_{|x'|} = 1$, and set $y_0 = y_{0,1}||\cdots||y_{0,|y_0|-2}$. In the other hand if $y_{0,|y_0|} = 0$, then $x_{|x|} = x'_{|x'|} = 0$, and set $y_0 = y_{0,1}||\cdots||y_{0,|y_0|-1}$. Then repeat the method until $|y_0| = 11$. Last we can find every bits of x and x' are the same. And we show $x = x'$, which is a contradiction. It means that map s is injective.
1.(b)
If z is empty, according to previous part, there is no strings $x \neq x'$ and z satisfy $s(x) = z||s(x')$.
If z isn't empty, set $a = z||s(x')$, we may find substring 11 in $a_1||a_2||\cdots||a_{|a|}$. But we can only find substring 11 in $s(x)_0||s(x)_1$, and it's a contradiction.
In conclusion, there is no strings $x \neq x'$ and z satisfy $s(x) = z||s(x')$.
2.
From the two conditions in previous part, we show that collisions can't be found by adding padding or changing bits of input, and it implies the map s is collision resistant.
3.
Assume there is a collision on h.For instance, $x \neq x'$ and $h(x) = h(x')$, we will like to show a collision on the compression function g can be found efficiently.

Because $x \neq x'$, x and x' are padded with two different values d and d'. In the same method, $k + 1$ and $k' + 1$ are the number of blocks for x and x'.
Because $t - 1 = 0$, we don't need to deal with $x \not\equiv x' \mod (t - 1)$. We only need to consider $k = k'$ and $k \neq k'$.
For $k = k'$, it means that $y_{k+1} = y_{k'+1}$, and we can get:
$g(z_{k-1}||y_k) = z_k = h(x) = h(x') = z_k = g(z'_{k-1}||y'_k)$

$z_{k-1} \neq z'_{k-1}$, we find a collision. Otherwise, repeating the process and get $g(z_{k-2}||y_{k-1}) = z_{k-1} = h(x) = h(x') = z_{k-1} = g(z'_{k-2}||y'_{k-1})$

So we will all find a collision. If none is found, it means that $z_1 = z'_1, \ldots, z_k = z'_k$, which makes a contradiction.

For $k \neq k'$ (assume$k' > k$) and do the same method as in the previous case. If no collision is found before $k = 1$ then we found $g(0^m||y_1) = z_1 = z'_{k'-k+1} = g(z'_{k'-k}||1||y'_{k'-k+1})$

m bits on the left is 0 and other bits on the right is 1. So we have found a collision.
The proof is complete since we consider all possible cases.