Cryptography and Network Security HW1

b08201054 鄭承樺

Handwriting

0.1 CIA

Please explain three major security requirements: confidentiality, integrity, and availability. For each security requirement, please give an example in the real world.

solution

- **confidentiality**: It means the protection from unauthorized disclosure.
- **integrity**: It means the protection from unauthorized changes.
- **availability**: It means to ensure intended users can access the service.

0.2 Hash Function

Please explain three properties of a cryptographic hash function: one-wayness, weak collision resistance, and strong collision resistance. For each property, please give an example applied in the real world.

solution

- **one-wayness**: It means that for any given y, it is computationally infeasible to find a x s.t. $y = H(x)$.
- **weak collision resistance**: It means that for any given x, it is computationally infeasible to find a $x' \neq x$ s.t. $H(x) = H(x')$.
- **strong collision resistance**: It means that it is computationally infeasible to find x and x' s.t. $H(x) = H(x')$.

0.3 Multi-prime RSA

We all know how RSA works. First, we choose two prime numbers, and then magic happens. But what if we have more than two primes in the modulus? The idea of using multiple primes in RSA, often called multi-prime RSA or r-prime RSA, which is as old as regular (2-prime) RSA itself, is that the modulus has multiple **distinct** primes. In fact, we can say that regular RSA is a special case of multi-prime RSA. In this problem, we will look into some advantages and disadvantages of multi-prime RSA.

- Key generation:
  1. Choose r ($r \in \mathbb{N}, r \geq 2$) **distinct** large primes π such that $N = \prod_{i=1}^{r} p_i$.
  2. Calculate $\phi(N) = \prod_{i=1}^{r}(p_i - 1)$.
  3. Select e and compute $d \equiv e^{-1} \mod \phi(N)$.
  4. The public key is (e, N) and the private key is $(d, p_1, ..., p_r)$.
- Encryption: $c \leftarrow m^e$
- Decryption: $m \leftarrow c^d$

a) Prove the correctness of multi-prime RSA, i.e., decrypting an encrypted message would recover the message.

b) Explain briefly why RSA, whether 2-prime or multi-prime, must use distinct primes.

c) As shown in class, the Chinese Remainder Theorem (CRT) can be used to opti- mize the performance of 2-prime RSA decryption. Apply the CRT to multi-prime RSA decryption and prove its correctness.

d) What are the advantages and disadvantages of multi-prime RSA over regular RSA? Please give at least two of each. You may mention possible attacks without explaining or proving how they work, but you must explain why one type of RSA is more vulnerable to such an attack than the other. If you notice any advantage or disadvantage in other subproblems, you can mention it here.

e) Most implementations of RSA use the Miller-Rabin primality test with trial division to search for primes. The expected runtime is $O(n4/log(n))$ where n is the bit length of the prime1. Show that the multi-prime RSA key-generation is more efficient than the regular RSA key-generation when the moduli are of the same size.

solution

a) We need to show that we can decrypt using the private key $(d, p_1, ..., p_r)$. Say the encrypted message is m, then we can solve $c^d \equiv x_i \mod p_i$.

b) Since find a square root not hard, if number of primes is two, we can easily factorize n. For multi-prime RSA, and not all primes are distinct, it may happen that we the encrypted message $c|n$. ($n = \prod_{i=1}^{r} p_i$)

0.4 Fun With Semantic Security

During the course, we have learned the way of using security reduction and calculating the attacker' s advantage in a security game. Let' s have more fun with these adorable ciphers! Let $\mathcal{E} = E(Enc, Dec)$ be a cipher defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$. Assume that one can efficiently sample an element from a uniform distribution over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$. Assume that $\mathcal{K}, \mathcal{M}$ are both group with a binary operation $+$. Assume that E is semantically secure (using the definition for one-time key). You are requested to prove that the following ciphers $\mathcal{E}' = E(Enc', Dec')$, which are constructed from E, are semantically secure (also for one-time key) as well.

a) $Enc'(k, m)$ is defined by sampling $r \xleftarrow{R} \mathcal{K}$ and output $Enc(k, m)||r$.

a) $Enc'(k, m)$ is defined by sampling $r \xleftarrow{R} \mathcal{M}$ and output $Enc(k, m + r)||r$.

a) $Enc'(k, m)$ is defined by sampling $r \xleftarrow{R} \mathcal{K}$ and output $Enc(k + r, m)||r$.

Capture The Flag

0.5 Simple Crypto

flag: CNS{5*upeR*_3*asy*_*c*1*a*55*ic*@*l*_*cryp*70!}
write up :

R1: The cipher text is Ceaser cipher, try 30 keys of shift and choose which decoded text has most words to be plain text.

R2: The cipher text is fence cipher, as R1, try 30 rails and choose the mode reasonable text to be plain text.

R3: Denoted the cipher text and plain text we get to be $c1$, $p1$, respectively, and the text we want to decrypt and the key to be $c2$, k. So, the plain text

$$p2 = p1 \oplus c1 \oplus c2$$

Since we have

$$p1 \oplus k = c1, \text{ and } \quad p2 \oplus k = c2$$

R4: Ignore all space and punctuation marks, and replace the upper case, lower case in the cipher text to b and a respectively. Then decrypt it with bacon cipher. Use functions in R2 to decrypt the text we got in the previous step as fence cipher. Finally, decode the flag in base64 to get the correct flag format.

0.6 ElGamal Cryptosystem

a) **flag:** CNS{*n*0_*r*3*us*3*d*_3*ph*3*m*3*ra*1_*K*3*Y*!}
writeup: Send an arbitrary message m to get

$$E(m) = (g^{sk})^y m \mod P$$

Since we konw m, we can calculate $m^{-1} \mod P$, and $(g^{sk})^y = E(m)m^{-1} \mod P$. Let the cipher text be $C = (g^{sk})^y p \mod P$, the plain text

$$p = C(g^{-sk})^y \mod P$$

b) **flag:** CNS{$d3CrYp7\_MY\_m355493\_w17H0u7\_K3y!?$}
writeup: Choose the lower bound 1, and the upper bound $\frac{2^{1024}}{\text{cipher text}}$. Apply binary search in this range, if server return the message is too long, search the left part of the range is saw in this iteration. Finally, search linealy to get the correxct plain text.

c) **flag:** CNS{$l4gr4ng3\_P0lyn0m14L\_12\_s0\_34SY$}
writeup:Notice that if we can obtain $(g^{sk})^y$, then the plain text $p = cg^{-sky}$ for cipher text c. In the source code we have $f(0) = sk$, hence $g^{sky} = (g^y)^{f(0)}$. In the server, we can get $(g^y)^{f(i)}$ for $i \in \{1, 2, 3, 4, 5\}$. For simplicity, denote $g^y = x$. So

$$x^{sk} = x^{f(0)} = x^{\sum_{i=1}^{5} \prod_{j=1, j\neq i}^{5} \frac{(0-i)}{(j-i)}}$$

By using the arithmetic relation of product and sum of exponent terms of x, we can calculate $x^{f(0)}$ to obtain the plain text.

0.7 Bank

a) **flag:** CNS{$ha\$h\_i5\_m15used$}
writeup: Found the two pdf files for $SHA1$ collision and use them as the name to register the account. So when I registered the second account, 10 dollars will be added to the same element in the list, hence has 20 dollars in that index.

b) **flag:** CNS{$\$ha1\_15\_n0t\_c0ll1510n\_r3s1st@nt$}
writeup: Just append a string "I love CNS" after the above collision. Since it use Merkle Damgard construction, the string before "I love CNS" has collision and the postfix is the same. Two collisioned pdf files after append "I love CNS" also has collision, and the appended string will gain an extra 5 dollars for each user. So we have 30 dollars to get flag2.

0.8 Clandestine Operation

a) **flag:**CNS{$Aka\_BIT\_f1ipp1N9\_atTaCk!$}
writeup: Use padding oracle attack against ID received from the server. The string after *secret* : is flag1.

b) **flag:** CNS{$W15h\_y0U\_hav3\_a\_n1c3\_d@y!$}
writeup: Modify the cipher text in the previous block to get the correct name field, and guess the other bytes in that block to make the block decodable.