Public-Private Cryptography

Hsiang Yun Lu

February 27, 2023

1 Applications of Public-Private Cryptography

The most commonly seen applications of public-private key cryptography are digital signature and data encryption. The encryption process provides confidentiality and integrity for the data to be encrypted. The confidentiality results from the fact that only the pre-determined user's private key can be used to decrypt the encrypted message. Data integrity is achieved by securing information against unauthorized changes which are not distinguishable to authorized users. Public key cryptography also allows authentication and non-repudiation during data communication, both enforcing the security of exchanging data.

As information becomes so important nowadays, protecting information, especially personal information, becomes critical. As described above, the information and our privacy can be protected via public-private key cryptography. For instance, the Bitcoin blockchain has billions of USD secured by asymmetric key algorithms. In addition, the HTTPS protocol is the encrypted form of HTTP. It allows sessions to be protected from eavesdropping and tampering. Another real-life example of public-private key cryptography is SSH keys used to communicate with GitLab, which I use daily for assignments and work. The SSH protocol allows us to communicate securely with Git. After SSH keys authentication, we don't need to supply a username and password each time.

2 Schmidt-Samoa (SS) Algorithm

This assignment gave me a chance to dig deeper into the Schmidt-Samoa algorithm. I re-read the description more than ten times but was still a bit confused. I don't think I understood it even if I had almost finished my implementation. I didn't realize I had misunderstood part of the logic until I got stuck decrypting the input message. My program couldn't decrypt the message using the private key. In the end, I found that I mixed up pq with $lcm(p-1,\ q-1)$ so that my private key was incorrect.

Also, before this assignment, I was unfamiliar with modulo operation and mathematic symbols (e.g., "≡"). Therefore, it took me some time to figure them out, which gave me a clearer understanding of the Schmidt-Samoa algorithm.

3 GNU Multiple Precision Arithmetic Library

I learned how to use the GNU multiple precision arithmetic library, which was very interesting. It is a totally new thing for me, so I also got a chance to practice reading manuals and instructions. The challenge I encountered during implementation was bit fiddling and logarithm calculation. At first, they are intimidating because there aren't functions written for shifting bits or taking logarithms. But after thinking more about the fundamental math and bit operations, these problems were easy to solve.

4 Random Number Generation

I spent some time on figuring out how to generate a random integer in a certain range using GMP library. In `is_prime`, we have to generate a random number in the range [2, n-2] but the corresponding function in GMP library can only generate a random number in the range [0, n-1]. Thanks for Omar's help so that I could understand the concept of shifting the range by adding the lower bound to the generated random number.

5 Making Prime

The most interesting takeaway from the lecture on cryptography is the method of checking whether a number is prime and generating a prime. The concept of primality tests and probabilistic tests is genius. I spent the most time modifying and checking the functions of the `is_prime` and `make_prime`.

6 Reference

- van Oorschot, Paul C. "Public Key Cryptography's Impact on Society: How Diffie and Hellman Changed the World." Democratizing Cryptography: The Work of Whitfield Diffie and Martin Hellman. 2022. 19-56.
- How Public Key Cryptography will continue to liberate a global society