Massachusetts Institute of Technology Handout R1
6.857: Network and Computer Security February 19, 2020
Professors Ronald L. Rivest and Yael Tauman Kalai Deep Gupta

Recitation 1: Modular Arithmetic & Security Policies

1 Modular Arithmetic

Modular Arithmetic is a system of arithmetic within a finite set of integers, where numbers "wrap around" when reaching a certain number. For example,

$$2 \equiv 12 \equiv 22 \mod 10$$

Generally, for $n > 0$ and integers a and b, we can say that $a \equiv b \mod n$ if n divides $a - b$, also denoted as $n | a - b$.

2 Basic Operations

For addition, subtraction and multiplication, modular arithmetic operations are consistent with normal math. So for a, b, a', b' where $a \equiv a' \mod n, b \equiv b' \mod n$, we have the following:

- $a + b \equiv a' + b' \mod n$
- $a - b \equiv a' - b' \mod n$
- $a \cdot b \equiv a' \cdot b' \mod n$

Division is trickier and relies on the existence of multiplicative inverses. Given an integer a, a^{-1} is the multiplicative inverse of a modulo n if $a \cdot a^{-1} \equiv 1 \mod n$.

Note that a only has a multiplicative inverse modulo n if and only if $\gcd(a, n) = 1$. An outline of the proof is as follows:

1. If $\gcd(a, n) = 1$, then using the extended Euclidean algorithm we can find x, y such that $ax + ny = \gcd(a, n) = 1$. Since $n|(ax - 1)$ then $ax \equiv 1 \mod n$, so x is the multiplicative inverse of a.
(The algorithm, which also lets us calculate the explicit multiplicative inverse, is described here: `https://en.wikipedia.org/wiki/Extended_Euclidean_algorithm`.)

2. If $\gcd(a, n) \neq 1$, then letting $d = \gcd(a, n)$, we can see that for any integers b, c, $d|ab$ and $d|n$ so regardless of choice of b, $d|(ab - cn)$. But if a had a multiplicative inverse then $d|1 - cn$ by setting b to be a^{-1} but this implies $d|1$ which is impossible for $d \neq 1$.

On the other hand because for any non-zero element in modulo p where p is prime, there exists a multiplicative inverse so division works all the time. Other than the method of computing the multiplicative inverse, division in modular arithmetic works the same as normal division.

For example, $\frac{1}{2} + \frac{1}{3} = \frac{5}{6}$ over rationals. Considering modulo 7, we get $2^{-1} = 4, 3^{-1} = 5, 6^{-1} = 6$ so applying this to the fractional equation $1 \cdot 4 + 1 \cdot 5 \equiv 9 \equiv 5 \cdot 6 \mod 7$.

3 Modular Exponentiation

Exponentiation works the same way as it does in normal math, i.e. repeated multiplication. The interesting part about exponentiation is how the value cycle as we go through the exponents.

For example, the powers of 2 appear as follows when simplified modulo 7.

2^1	2^2	2^3	2^4	2^5	2^6
2	4	1	2	4	1

In this case, it cycles every 3 elements. For any a and prime p, $a^1, a^2, \ldots, a^{p-1}$ must cycle through some subset of the $p-1$ possible residues modulo p. The length of the cycle is also called the order, i.e. $ord_7(2) = 3$.

For any prime p, there exists some integer g such that $ord_p(g) = p - 1$, in other words, the cycle loops through all possible residues. g is often referred to as a primitive root or generator of p.

4 Cryptographic Applications

Several classes of cryptographic schemes rely on g^k being easy to compute ($\log(k)$ operations by using repeated squaring) and finding k from g^k being hard (known as the "discrete logarithm" assumption).

For example, the Diffie-Hellman key exchange described below protocol relies on this assumption to securely create a shared public key between two parties over a public channel.

1. Alice and Bob agree on a large prime p and a generator g for p.
2. Alice randomly generates a number a, Bob randomly generates a number b.
3. Alice sends g^a to Bob, Bob sends g^b to Alice.
4. Alice and Bob independently calculate g^{ab} and use this is a shared key.

5 Writing Security Policies: A Guided Example

Writing security policies is a powerful skill that not only helps engineers, developers, and scientists reinforce their understanding of a system but also helps them establish protocols to address the system's weaknesses and vulnerabilities. Let's take a look at the following problem set question from last year and walk through writing a security policy.

> **Problem Statement:** Video and audio conferencing platforms such as Zoom, Skype, and WebEx are extremely common among businesses in 2020.
>
> Write a set of basic functionalities that these platforms should have, and then describe an ideal security policy for these functionalities. Note that these are large pieces of software with many diverse functionalities - focus mainly on the following 3: video chats, screen sharing, and recordings.
>
> The policies you come up with should address each of the security goals discussed in class, though focus on the one(s) that are most relevant for video and audio. Given the time constraints and the complexity of the problem, we expect your solutions to be less than comprehensive. That being said, keep in mind that an adversarial party can be either inside a private call or outside a private call and try to cause some undesirable result.

5.1 Describe the System: Functions and Actors

A good place to start is to understand the *functionality* of the system in question, or how the system should behave in the absence of adversaries. Some questions to help think about functionality are:

1. What roles should the system fulfill?
2. Who are the actors or players in the system? What permissions, restrictions, and responsibilities do they have?
3. When do two or more actors or entities interact and exchange information? Where is information stored? What properties do we want for the exchange or storage (think CIA triad)?

We can answer all of these questions for a video-calling system! The system's high-level functions have already been given in the problem statement: video chats, screen sharing, and recording. Next, let's consider possible actors and their roles and permissions in the context of the aforementioned functions.

- Customers. These are the most common users of the system. They have various permissions and responsibilities, such as the following:
  - Login - Customers have access to their account and are required to create a (strong) password to secure it. Users are responsible for keeping their passwords and personal account information private. The system will have some form of two-factor authentication, so customers have the ability to decide whether to allow or deny access to their account from a new device.
  - Video call responsibilities - Users are allowed to reach out to other users to initiate video calls. They may invite as many users as they would like to join the call, up to some upper limit, by sending them a link to the call. The system will encrypt the calls, but users are responsible for ensuring that they invite users they trust and intend to invite.
  - Screen Sharing - Users have the option of sharing their whole screen or a specific application window. Users are responsible and liable for any information they disclose during this process.
  - Recordings - The system will encrypt recordings. Users can access these recordings for a limited time after their call and download them. The user is responsible for storing any recordings in a safe location.
- Workers. They maintain the system and log and store call data, including when calls were made, users involved in the call, duration, etc.
- Bystanders. Any entity with no access to the system or its parts.

5.2 Identify Adversaries and Attacks

After establishing how a system should work, the next step is to identify the system's weaknesses and vulnerabilities. One approach to do so is to understand the *adversaries* of the system and how they would challenge it. Can you have adversaries among your system's players or actors? Can some third party with no intended relationship with the system be an adversary? How powerful are the adversaries? Once you identify potential adversaries and their capabilities, we can anticipate possible attacks.

In a video call, we could have adversaries among other users within the video call and among users or bystanders outside of the video call. If the adversary is a user already in the video call, the adversary could attempt to extract personal information during screen sharing or through a social engineering attack. If the adversary is outside the video call, the adversary may attack in the following ways:

- eavesdropping on the private video call
- disrupting communication channels between calling parties so they are unable to call
- acquire call recordings they shouldn't have access to

5.3 Provide the Security Policy

Now that we have identified potential attacks, we are in a good position to write policies and mechanisms to defend against them. Here are security policies for each of the three high-level functions the question asked us to consider:

1. Video chats - The invitations to the call will be distributed only through the users' accounts which are password protected. This helps prevent unwanted parties from getting access to the call. Additionally, users who create a meeting can protect it with an additional password. Expected parties will be given this password and held responsible for keeping it secure while unexpected parties will not be able to enter the meeting. Another feature is that the meeting host or admin can lock the call, preventing additional users from joining even if they have the password.

2. Screen Sharing - One security measure is that users can enable watermarks during screen sharing. These watermarks will make it difficult for images to be copied via screenshot and used for malicious purposes. Another measure is that users can choose which window to share rather than share their whole screen to prevent accidental information leaks. Third, if the user chooses to share their whole screen, the system will throw a warning to the user as they are about to open a new window to ensure that they don't accidentally leak information.

3. Recordings - All recordings will be encrypted before being stored. Additionally, users (meeting hosts or admins) have the option to disable recording altogether for their meeting. Note, however, that this does not prohibit users from recording in ways outside of the system.

As intended, we were able to devise strong security measures after considering questions such as who our adversaries may be and how they may attack our system. Notice that the security measures list for recordings includes a clear statement about what is anticipated but not included. Making clear what aspects of privacy are guaranteed and what aspects are not guaranteed is the hallmark of a strong security policy.