CS1340: Course Project
Submission Date: December 06
Marks: 50
Weightage: 20%

---

# Course Project Evaluation: Instructions
- Use `https://forms.gle/ximmLynGqcSPa46F6` to submit your team members and project topic Respond with your team information by November 24
- Each team consists of $\leq 3$ students

---

## A Few Project Ideas

1. Wireshark Lab (Kurose and Ross) (For a very good introduction see `http://www-net.cs.umass.edu/wireshark-labs/Wireshark_Intro_v8.0.pdf`). As part of the project, your team can select any one of the following topic. For evaluation, your team will make a one-hour presentation wherein you demonstrate the key concepts of the selected topic (as per the instructions set in the selected lab) by capturing packets in realtime and analysing on them.

   (a) IP (`http://www-net.cs.umass.edu/wireshark-labs/Wireshark_IP_v8.0.pdf`)

   (b) TCP (`http://www-net.cs.umass.edu/wireshark-labs/Wireshark_TCP_v8.0.pdf`)

   (c) NAT (`http://www-net.cs.umass.edu/wireshark-labs/Wireshark_NAT_v8.0.pdf`)

   (d) DNS (`http://www-net.cs.umass.edu/wireshark-labs/Wireshark_DNS_v8.0.pdf`)

2. Interactive authentication (log-in) systems. In a password-based authentication system, a user $\mathtt{U}$ creates its user credential record — a user identity $\mathtt{id_U}$ , and a hash of a secret password $\mathtt{pwd}$, i.e. $\mathsf{hash}(\mathtt{pwd})$, with the application server through a one-time registration process. Later, in order to gain access (log in) to the application server, $\mathtt{U}$ provides $(\mathtt{id_U}, \mathtt{pwd})$ to its client which then computes $\mathsf{hash}(\mathtt{pwd})$, and sends $(\mathtt{id_U}, \mathsf{hash}(\mathtt{pwd}))$ to the application server. The server compares this to the stored record for the stated $\mathtt{id_U}$ and gives access if it matches.

   There exists new proposals that do not require the server to store hashed passwords of users. These systems are interactive and provides strong password safety. (attend office hours to discuss more on this)

   **Protocol A**: In the following, an interactive authentication system is described. The security of this system is based on the fact that it is very hard to factor a special type of large composite number. An attacker can not impersonates a valid user unless it knows how to factor these numbers. The details are as follows.

   - **User Registration (User credential generation)**
     - A user $\mathsf{U}$ with identity $\mathsf{id_U}$ generates a registration information as follows
     - It generates two 50-bit primes $p, q$ such that $p \neq q$.
     - Computes $N = p \times q$.
     - Picks a $w \in \{1, 2, \ldots, n\}$ uniformly at random

– Computes $v = w^2 \bmod N$

– Submit to the server the following login credentials:

$$\begin{cases} \mathsf{login\ id} = & \mathsf{id_U} \\ \mathsf{Record} = & (\mathrm{v,N}) \end{cases}$$

– Finally, it discards $p, q$ and only keep $w$ as secret password.

- **Log-in Process:** In order to gain access to the server, a user must prove to the server that it knows a square root $w$ of $v$ (modulo $N$), i.e. $w^2 \equiv v \bmod N$. An interactive proof between client and server for the same is carried out as follows. The proof satisfies a so called zero-knowledge property: at the end of the proof the server is convinced that the client knows a square root $w$ of $v$, but without the server ever getting access to the $w$.

| Client (U)($w$, $N$) | | Server ($v$, $N$) |
|---|---|---|
| 1. U picks an $x \in \{1, 2, \ldots, n\}$ such that $\mathsf{gcd}(x, N) = 1$<br>2. It computes $y = x^2 \bmod N$<br>3. It sends $y$ to the server | $\xrightarrow{y}$ | |
| | $\xleftarrow{b}$ | 1. Server picks a random bit $b \in \{0, 1\}$<br>2. Sends back $b$ to the client |
| 1. U sets $z = \begin{cases} x & \text{if } b = 0 \\ wx & \text{if } b = 1 \end{cases}$<br>2. It sends $z$ to the server | $\xrightarrow{z}$ | |
| | | $\begin{cases} \text{Access granted,} & \text{if } (b = 0) \wedge (z^2 = y \bmod N) \\ \text{Access granted,} & \text{if } (b = 1) \wedge (z^2 = v \cdot y \bmod N) \\ \text{Access denied,} & \text{otherwise} \end{cases}$ |

3. Innovative socket-programming based networking applications (attend office hours to discuss this).