Each group (two students at most) may choose one of the projects. Each of them has equal point and any extra work will be considered as bonus point.

**Project Goal: Implementing RBAC**

**Project Description**

In this project, you are supposed to define a role-based access control (RBAC) module that implements PA, UA and RH. For a thorough description of RBAC model, you can visit [this link](#) and [this entry](#).

You must develop a module or DLL, which can be used as a general RBAC core. By “general”, we mean it can be used effectively for controlling access of users to some resources and it must be dynamic so that it can be used in different systems without any great change.

This module must benefit from a very easy-to-use and efficient management console that provides administrator with ordinary requirements of the model and log of any illegal attempts made by any one regarding his/her authentication code, time and any required field in order to precisely identify the intruder.

Presenting administrator with a good GUI and showing hierarchies using graphs is preferable. Please note that use of any open source software is prohibited.
Project Goal: Implementing Mailing system with proper encryption over public key infrastructure

Project Description
In this project, you must create a mailing system (not necessarily over SMTP, but a simple socket based system suffices) which is supported by encryption and decryption of messages. The program must have a key ring manager, which is updated by a server that is developed by you. A good example of such system is GPG4win. Everyone must use the composer to create a new mail. Each email must be automatically and transparently signed by your private key and encrypted with recipient public key. The program must ask for public key of the receiver and also before encrypting the message with that public key, it must check the validity of that key using CRL service. There is no need to implement CRL with real certificates but issuing a list with expired public keys suffices. For each recipient in the list this action must reoccur. Needless to say that checking the validity of sender public key is implied.

Receiver of the mail must decrypt the message automatically and transparently.

To cut the matter short, you need to have a key manager, a CRL issuer service, an authority to hold public keys, ability to expire your public key, ability to introduce new public key, composing and receiving emails and any other necessary options that is required for managing service, mailing system and key ring manager.

Everyone must have one active public key at a time and producing new public key can be handled manually, but the program must accept this public key and send it to the corresponding authority.

Use of any open source software is permitted in this project but open source components must not constitute more than 30% of the project.
**Project Goal: Implementing an e-voting protocol using X.509 v.1 certificates**

**Project Description**

In this project, a security protocol for electronic voting is simulated. Everyone can cast a vote only once and voting manager must hold these votes until the end of voting process. Then it must disclose the results. The following descriptions can be clearly comprehended using diagram depicted below.

In order to authorize users to the authority, everyone must get his/her certificate from the root CA. The public key required for anyone to authenticate him/herself to the CA is implied and need not be generated but to be manually assigned to each client and need not to be verified by the CA.

After getting X.509 certificate from the CA, each client sends his/her certificate to the authority. The authority validates this certificate and issues a session key for...
the certificate while saving the pair in a table. Then every client chooses a candidate, signs it with the session key and sends the signed vote to the collector. Collector returns an index. Clients save this index and relay it to the authority. The authority adds this index to the table using the information sent by the client.

When vote collection is finished, collector requests (session key, index) pairs from the authority. After receiving this table, the collector can open votes and count them. By the way, the collector must output table of votes and indices to enable everyone to verify his/her vote. Finally, the collector must introduce the winner.

Please note that any kind of transmission must be secure and integrity of all messages must be preserved using public key infrastructure, but the process of signing, encryption and decryption must be transparent.

Please note that except for generating certificates, use of any open source software is prohibited.

**Project submission**

You may submit your project via e-mail and defend it in a to-be-decided session face-to-face. Every student must be able to defend his/her submitted project. Project deadline will be announced in the course website.

You may send your solutions to either Sobhdel@nsc.sharif.edu or Sobhdel@ce.sharif.edu. You may compress the project and name it according to the following formula:

<Your_Team_Name> + “_” +<Your_Student_Number(s)> + “_” + DNS_Project + <ProjectNo>

Project No : RBAC System = 1
Mailing System = 2
E-voting System = 3