# پروژه‌ی درس مقدمه‌ای بر رمزنگاری

## رای‌گیری الکترونیکی

حجت آقاخانی ۹۰۱۰۵۶۵۶
زمستان ۱۳۹۳

## مقدمه

در پروژه‌ی شماره‌ی یک، سیستم رمز الجمال را پیاده‌سازی کردیم. هدف از این پروژه، پیاده‌سازی یک سیستم رای‌گیری الکترونیکی امن، با استفاده از سیستم رمز الجمال است. این پیاده‌سازی بر مبنای معماری کارگزار-مشتری می‌باشد و هریک از برنامه‌های کارگزار و مشتری چندریسه‌ای می‌باشند. در ادامه به توضیح این پیاده‌سازی می‌پردازیم.

## توضیح پروتکل
به‌طور کلی این پروتکل دو مرحله دارد.

## مرحله‌ی اول
در این مرحله، هر مشتری (رای‌دهنده)، کلید عمومی خود را به کارگزار می‌فرستد. هم‌چنین به همراه کلید عمومی خود، اثبات دانش لگاریتم گسسته را می‌فرستد. این کار بدین منظور است که ثابت کند، کلید خصوصی مربوط به این کلید عمومی را دارد. توجه کنید که این اثبات هیچ‌گونه اطلاعاتی در رابطه با کلید خصوصی در اختیار کارگزار قرار نمی‌دهد. این مرحله با اعلام کارگزار (وارد شدن دستور FirstStepFin) پایان می‌یابد و در پایان این مرحله، کلید عمومی هر مشتری به همراه اثبات آن، برای سایر مشتری‌ها فرستاده خواهدشد و این مرحله پایان می‌پذیرد.

## مرحله‌ی دوم
این مرحله پس از اتمام مرحله‌ی اول شروع می‌شود. پس پیش از شروع مرحله هر مشتری از کلیدهای عمومی (به‌همراه اثبات دانش آن‌ها) سایر مشتری‌ها باخبر است. در این مرحله با روشی که در صورت پروژه گفته‌شده است، هر مشتری رای خود را برای کارگزار ارسال می‌کند و کارگزار علاوه‌بر بررسی صحت رای، رای گرفته‌شده را برای سایر مشتری‌ها ارسال می‌کند. هر مشتری نیز پس از دریافت رای مشتری دیگر، وظیفه دارد صحت آن را بررسی کند. پس از اتمام این مرحله، هر مشتری می‌تواند از صحت رای‌گیری مطمئن شود.

## برنامه‌ی کارگزار

برنامه‌ی کارگزار به طور کلی چند ریسه دارد. یک ریسه وظیفه‌ی گوش‌کردن دارد، تا شاید مشتری‌ای به کارگزار وصل شود. توجه شود که کارگزار قبل از وارد شدن دستور FirstStepFin مشتری می‌پذیرد و بعد از آن، مشتری نمی‌پذیرد. بعد از پذیرفته‌شدن یک مشتری، یک ریسه‌ی دیگر اجرا می‌شود که وظیفه‌ی آن ریسه گرفتن اطلاعات از آن مشتری و پردازش آن‌ها می‌باشد. یک ریسه هم برای فرستادن اطلاعات به مشتری‌ها داریم. توجه شود که یک ریسه‌ی اصلی (در واقع برنامه‌ی اصلی کارگزار) وجود دارد که وظیفه‌ی آن گرفتن دستورها از کنسول و پردازش آن‌ها می‌باشد. در ادامه به توضیح کلاس‌های برنامه‌ی کارگزار می‌پردازیم:

## کلاس server

این کلاس مربوط به ریسه‌ی اصلی کارگزار می‌باشد که وظیفه‌ی آن گرفتن دستورها از کنسول می‌باشد و بر اساس آن دستورها کار لازم را انجام می‌دهد. در ادامه متدهای اصلی این کلاس را به‌همراه توضیح می‌آوریم.

```
private boolean handleCommand(String command)
      this method do the right thing according to the given command from
console!

private void printAllVotes()
      this method print all clients vote!

private void calculateResult()
      this method calculate result of poling

private void sendAllClientPacketsToOthers()
      this method send all receive packets from clients to others,
including DL and Ballot proof and public key

public void sendClientVoteToOthers(String id)
      this method send all receive packets from clients to others,
including DL and Ballot proof and public key

public void computeHIs()
      this method compute HI s that client need for voting

private void sendToAll(String message)
      this method send message to all clients!

private void verify()
      this method verify the proofs of voter!

private void getQuestion()
      this method read question, and add it to the all questions!

public void acceptClient(CommunicateToClient newClient)
      this method add newClient to clients of server
```

## کلاس AcceptClient

این کلاس در واقع همان ریسه‌ای است که وظیفه‌ی گوش کردن برای مشتری‌های جدید می‌باشد. اجرای این ریسه بعد از وارد شدن دستور FirstStepFin متوقف می‌شود.

## کلاس CommunicateToClient

وظیفه‌ی اصلی این کلاس، ریسه‌ای است که اطلاعات را از مشتری می‌گیرد و براساس آن کار لازمه را انجام می‌دهد. در ادامه مهم‌ترین متدهای این کلاس را به‌همراه توضیح می‌آوریم.

```
public void sendToClient(String data)
      this method send data from server to client!

private void handleMessage(String command)
this method do the right thing according to the received message!!!

private boolean checkBallotProof()
      this method check that ballot proof is valid or not!

private void sendQuestionToClient()
      this method send current question to client!

private boolean checkDiscreteLogProof()
      this method check that discrete proof logic is valid or not!
```

## کلاس CycleGroup

این کلاس، به طور کامل در مستندات پروژه‌ی شماره‌ی یک توضیح داده شده‌است. به طور خلاصه یک گروه مرتبه‌ی p با مولد g می‌سازد.

## کلاس RunServer

در واقع با اجرای متد main این کلاس، برنامه‌ی کارگزار اجرا می‌شود.

## برنامه‌ی مشتری

برنامه‌ی مشتری کلا دو ریسه دارد. یک ریسه برای خواندن اطلاعات از کارگزار می‌باشد و ریسه‌ی دیگر برای فرستادن اطلاعات به کارگزار می‌باشد. در ادامه به توضیح کلاس‌های برنامه‌ی مشتری می‌پردازیم:

## کلاس Client

با اجرای این کلاس، ریسه‌ای اجرا می‌شود که وظیفه‌ی آن، خواندن دستورات از کنسول و اجرای کارهای لازم می‌باشد. در ادامه متدهای اصلی این کلاس به همراه توضیح آن‌ها آمده است:

```
private boolean handleCommand(String command)
this method handle command that client enter it!

public void computeHIs()
this method compute HI s that client need for voting

private void sendMyQuestionToServer()
this method send current question to client!

private void verify()
this method verify the proofs of voter!

private void calculateResult()
this method calculate result of poling

private void printAllVotes()
this method print all clients vote!

private boolean checkBallotProof(String ballotProof, int clientNum)
this method check that ballot proof is valid or not!

private void getVoteAndSendToServer()
this method get vote from client and send them to server!
```

## کلاس CommunicationToServer

با اجرای این کلاس، ریسه‌ای اجرا می‌شود که وظیفه‌ی آن خواندن پیام‌ها از مشتری می‌باشد و در ادامه اجرای کار لازم. در ادامه متدهای اصلی این کلاس به همراه توضیح آن‌ها آمده است:

```
private void handleMessage(String command)
this method do the right thing after receiving command from client

private void saveVote(int clientNum)
this method save vote of client!!!

private void getClientsInfo(int numOfClients)
this method get clients public key and DL proofs from server!!!

private void generateKeysAndSendThem()
this method generate private and public keys and send them to server!!!
```

```
private void discreteLogProof(BigInteger x)
this method send requirements data for discrete log proof of knowing x
from g^x

public void ballotProof(BigInteger b, int choiceNumber)
this method send requirements data for ballot proof
```

## کلاس AbstractClient

برای ذخیره کردن اطلاعات مربوط به مشتری‌ها، برای هر مشتری نیاز دیدیم این کلاس را پیاده‌سازی کنیم. این کلاس اطلاعات مربوط به کلیدهای عمومی، اثبات‌های لازم و ... را مدل می‌کند. در ادامه متدهای اصلی این کلاس به همراه توضیح آن‌ها آمده است:

```
public boolean checkDiscreteLogProof(BigInteger p, BigInteger g)
this method check that discrete proof logic is valid or not!
```

## کلاس RunClient

در واقع با اجرای متد main این کلاس، برنامه‌ی مشتری اجرا می‌شود.

## کلاس‌های عمومی

در این‌جا به توضیح کلاس‌های عمومی که هم در برنامه‌ی مشتری و هم در برنامه‌ی کارگزار استفاده می‌شود، می‌پردازیم.

### کلاس Question

این کلاس، مفهوم سوال را مدل می‌کند، و در آن اطلاعاتی نظیر گزینه‌ها، صورت سوال و ... نگه‌داری می‌شود.

### کلاس Choice

این کلاس، مفهوم گزینه را مدل می‌کند.