YAŞAR UNIVERSITY

GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

MASTER THESIS

MOBILE BASED ELECTRONIC VOTING SYSTEM

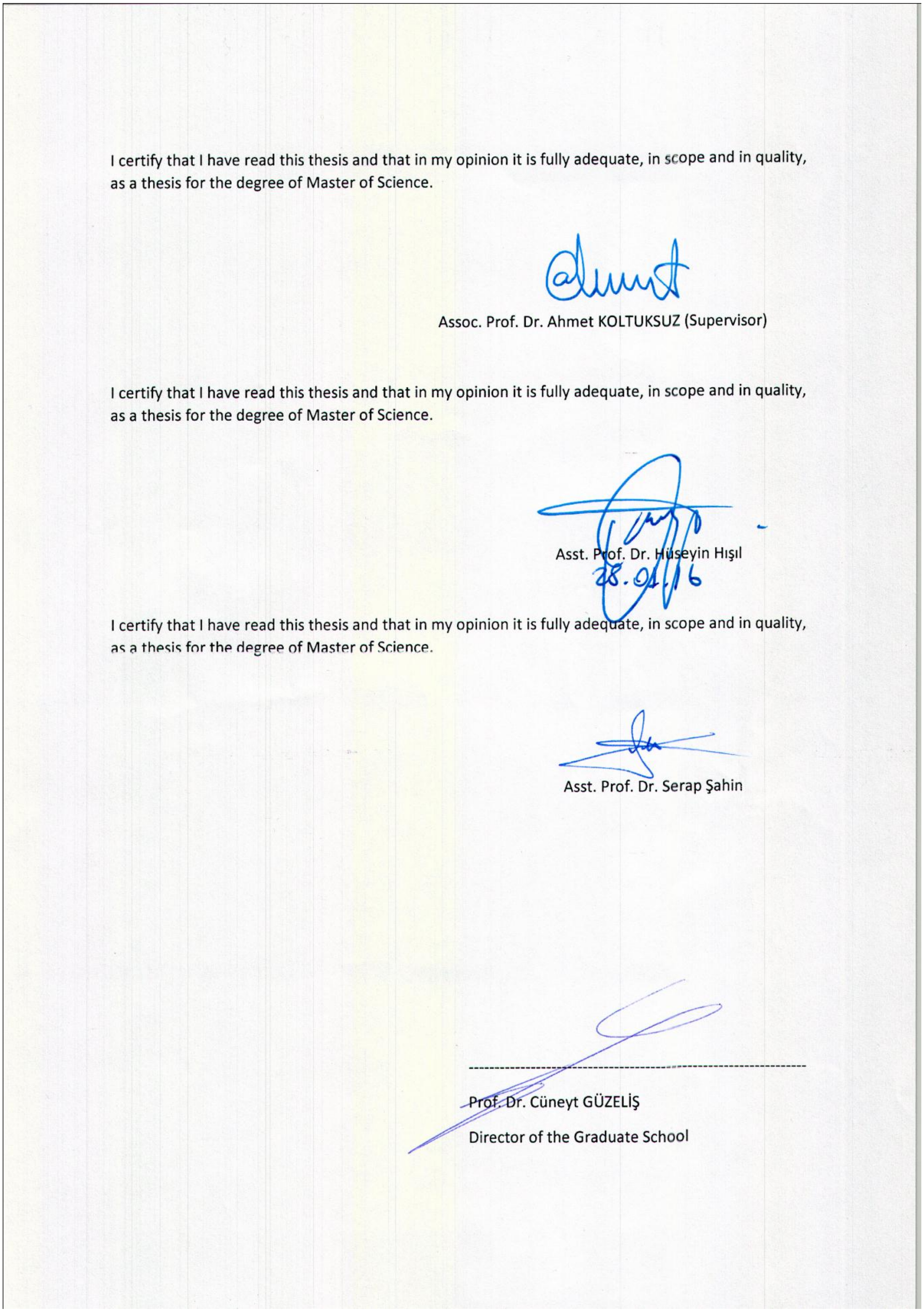
Murat Ödemiş

Thesis Advisor: Assoc. Prof. Ahmet Koltuksuz, Ph. D.

Department of Computer Engineering

Presentation Date: 22.01.2016

Bornova-İZMİR
2016



I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Ahmet KOLTUKSUZ (Supervisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Asst. Prof. Dr. Hüseyin Hışıl
28.01.16

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Asst. Prof. Dr. Serap Şahin

Prof. Dr. Cüneyt GÜZELİŞ
Director of the Graduate School

ABSTRACT

MOBILE BASED ELECTRONIC VOTING SYSTEM

Murat Ödemiş
MSc in Computer Engineering
Supervisor: Ahmet Koltuksuz, Ph. D.

January 2016, 104 pages

The main objective of this thesis is to develop a secure online mobile voting system. This voting system is intended not only to be used for governmental elections but also for public and private institutions and also in the meetings that need an instantaneous, reliable and authorization-based mobile voting.

The application was made for the IOS platform, it is also compatible with other mobile systems. The mobile application was coded using CORDOVA, and server-side were coded by .NET MVC5. Also, PHP is used for certification processes. MSSQL is used to authorize the database. The data is encrypted and decrypted via AES-256 to ensure its security. To complete a symmetric encryption, a common session key is used by the client and the server, and it was computed by an Elliptic Curve Diffie-Hellman schema, called Curve25519 (Bernstein, 2006). In addition to this, the integrity of the data is checked by HMAC. Biometric fingerprint-scanning technology is used in collaboration with an Apple Touch ID. One of the most important features of the system is that, when under pressure, the user can enter the system with a fake password and cast their vote. After user enters to the system with his institutive credential, the user sets a character-based password, a fake password, a geometric pattern password and a fake geometric pattern password in activation period. Before the election page is displayed, they can enter with either fake password, but their votes won't be valid. The application consists of a login, certification, activation, listing election, and a voting screen. The network is protected with SSL. During the final stage, a one-time password is sent to the users via SMS, then, the vote will be casted to the server.

Keywords: Mobile Voting, Electronic Voting, Curve25519, Key Exchange, Cordova, Mobile Application, iOS

ÖZET

MOBİL BAZLI ELEKTRONİK OYLAMA SİSTEMİ

Murat Ödemiş
Yüksek Lisans Tezi, Bilgisayar Mühendisliği Bölümü
Tez Danışmanı: Doç. Dr. Ahmet Koltuksuz

Ocak 2016, 104 sayfa

Bu tezin temel amacı, devletin yaptığı seçimlerde, kamu kurumlarının ve özel kurumların kullanabileceği, toplantılar esnasında da anlık olarak kullanılabilecek güvenli ve yetkilendirme sistemi içeren bir seçim sistemi oluşturmaktır.

Uygulamanın yazılımı için Apple iOS platformu baz alınmıştır. Bunun yanında da Android ve mobil tarayıcılarla da yüksek ölçüde uyumludur. Mobil uygulama tarafında CORDOVA platformu, sunucu tarafındaki yönetim sistemi ve web servisler ASP.NET MVC5, sertifikasyon sürecinde de bunlara ek olarak PHP kullanılmıştır. Veri tabanı ve sertifika otoritesi için ise MSSQL kurulmuştur. Uygulama güvenliği için veriler AES-256 ile şifrelenip, çözülmektedir. Bu simetrik şifreleme için Kullanıcı ve sunucu tarafında kullanılacak ortak anahtar, Curve25519 isimli Eliptik Eğri Diffie-Hellman şemasıyla belirlenmektedir. Bunun yanında HMAC ile veri bütünlüğü kontrol edilmektedir. Apple Touch ID yardımıyla parmak izi teknolojisi kullanılmıştır. Sistemin önemli özelliklerinden biri de baskı altındayken aldatıcı şifre ile giriş yapıp oy verebilmektir. Kullanıcı kurumunun bilgileriyle giriş yaptıktan sonra, aktivasyon aşamasında bir şifre ve aldatıcı şifre, bir çizim şifresi ve aldatıcı çizim şifresi belirler. Seçim ekranını görmeden önce, bu aldatıcı şifrelerden birini girerek oy verebilir fakat oyu sayılmayacaktır.

Uygulama; giriş, sertifikasyon, aktivasyon, seçim listeleme, seçim görüntüleme, oy verme ekranlarından oluşmaktadır. Tüm veriler şifreli gelip gider ve ağ SSL ile korunur. Oy gönderim aşamasında kullanıcı doğrulamasını seviyesini arttırmak için SMS ile Tek Kullanımlık Şifre girişi yapılması istenir ve oy sunucuya iletilmiş olur.

Anahtar sözcükler: Mobil Oylama, Elektronik Oylama, Curve25519, Anahtar Değişimi, Cordova, Mobil Uygulama, iOS

ACKNOWLEDGEMENTS

I would like to thank my advisor, Ahmet H. Koltuksuz, Ph.D. Without his quick understanding and practical advice, I could not have proceeded in this study. He always helped me to make my decisions with his perspicacity and prognoses. He supported me while I was forming my sector and academic career through his well-appreciated guidance.

Secondly, I would like to thank Hüseyin Hışıl, Ph.D. for his constructive, collimating comments and support. His valuable contributions to critical issues of this thesis geared me to think systematically and develop an outline for my study.

I would like to express my sincere thanks to my colleague Cağatay Yücel for his constructive comments and helps about cryptologic background. He always answered my questions patiently in every situation. Also I would like to thank Mutlu Beyazıt, Ph.D. for his contributions about future work of this thesis and my sincere thanks goes to Serap Şahin, Ph. D. for her constructive and precious comments at the defense.

I gratefully thank Caner Hekimoğlu and all of research assistants in Yaşar University's computer engineering department. Their ameliorations of administrative issues I encountered while constructing this thesis rapidly sped up the process.

Murat Ödemiş
İzmir, 2016

TEXT OF OATH

I declare and honestly confirm that my study, titled "MOBILE BASED ELECTRONIC VOTING SYSTEM" and presented as a Master's Thesis, has been written without applying to any assistance inconsistent with scientific ethics and traditions, that all sources from which I have benefited are listed in the bibliography, and that I have benefited from these sources by means of making references.

TABLE OF CONTENTS

Page

INDEX OF FIGURES

INDEX OF TABLES

INDEX OF SYMBOLS AND ABBREVIATIONS

Abbreviations

SIM	Subscriber Identity Module
JSON	JavaScript Object Notation
MVC	Model View Controller
SMS	Short Message Service
GSM	Global System for Mobile Communications
NIC	Network Interface Controller
API	Application Programming Interface
WPKI	Wireless Public Key Infrastructure
SDK	Software Development Kit
GPS	Global Positioning System
NFC	Near Field Communication
PDA	Personal Digital Assistant
CS	Counting Server
VS	Verification Server
XML	Extensible Markup Language

1 INTRODUCTION

Recently, with technological improvements, printed material has shifted to exist on computers instead, and nowadays, computers are being replaced by mobile devices. The main objectives of this thesis are to develop a reliable and rapid mobile voting system that depends on the most recent technologies and to design applicable security protocol for the mobile application. This thesis strives to be of use not only in government elections, but also for firms and other institutions such as universities, factories etc. The system's goals are to generate a quick solution for decisions and to collect the opinions of employees or citizens in the case of an election.

Consumers have grown to depend on mobility as a primary deciding feature in all of their chosen technologies. The general expectations for mobility rises daily, as most consumers now require at least one small, portable device to satisfy all of their needs. It is predicted that the number of mobile devices in use, including both phones and tablets, will grow from over 7.7 billion in 2014 to over 12.1 billion by 2018 (Radicati, 2014).

To use phones outside of the home, car phones started to be produced in the early 80s, but they were not enough for their users. A couple of decade’s thereafter mobile phones became popular to serve the same purpose. Moreover, there are several examples in everyday life that explain the needs of mobility. In order to satisfy the craving for music, Walkman appeared. In another representation, desktop computers were replaced by laptops, then laptops were switched to tablets. The number of active cell phones reached 7.3 billion in 2014, and the cell phones outnumbered the global population that year. At the end of 2013, the global mobile data traffic was at a 1.5 Exabyte per month, then it increased by 69 percent in 2014 to reach a rate of 2.5 Exabyte per month by the end of the year (Cisco, 2014).

At the end of the 2014, it was recorded that the number of individual mobile users reached 3.6 billion, which meant that the mobile industry constantly expanded. About ten years ago, the total number of mobile users was one-fifth of the world’s population. Now, however, half of the population is composed of this group. Also, it is estimated that one billion more users will filter in by 2020, considering that the population growth rate is currently at 60%. According to the GSMA (2015) the total number of global SIM connections was 7.1 billion, while the number of machine-to-machine (M2M) connections was 243 million at the end of 2014.

Smart phones are capable of managing multiple processes, such as banking, playing games, listening to music, reading, writing and more, all in a single device, and they have become extremely popular because of those features. Banks keep up with mobility by making many operations executable by mobile phones, such as remittance. Similarly, governments are trying to match the speed of technological advancement. Payment operations, information updates and other services can be performed in electronic applications. Most countries have developed e-Government portals for their citizens, and these portals also have mobile applications. Through their information technologies, they aim to provide public services to citizens, businesses, and government agencies in efficient and effective manners.

Governments try to keep pace with the constant innovations, like smart phones. Their electronic applications offer access to all kinds of public services. E-government solutions are continuously being improved as necessary, and they are highly adaptable. When we prefer a mentioned governmental system or common authority, we think about voting. While the governments are administering their existing duties through an electronic platform, they will also transform the standard paper-based voting system into an electronic method.

In most countries nowadays, manipulated, stolen, and forced (etc.) votes are reported to exist. It is necessary that the information technologies also help to manage these kinds of problems with their generated solutions. Properly introducing such technology would provide the highest understanding of the freedom and mobility that it can offer. In order to solve the problems in the paper-ballot voting system, studies of mobile voting have started in parallel with the e-voting researches.

In any democratic society, all of its members are supposed to have equal suffrage without any differentiation because of their education or income levels. The objective of democracy was initially written to be a permittance of unrestricted public voting so anyone could simply cast votes according to their wishes.

To establish a coalition between technology and ballots, e-voting applications started to be implemented in the '90s in various countries, such as the United

Kingdom, Estonia, Switzerland, Canada, the United States and France. In the 2000s, nearly 40 countries tried electronic voting systems.

In Estonia, e-voting was started in 2005, and nearly a quarter of its votes were cast online in 2009. It was a real success for a country to use an e-voting method that much. Now, voters in that location use a card or a mobile phone ID in order to cast their votes within a predetermined time period (Scammell, 2013). Analyses found that 175,000 Estonians voted online in the 2015 elections, which set the latest record. This statistic indicated that there has been a 25% increase in online voting there since the previous elections in 2011. It seems that in Estonia, e-voting became the ordinary way of voting. The main objective behind the newer method is to increase people's ease of accessing a ballot system, not to totally eliminate the paper-based one. According to Estonian records, e-voting became successful in pursuit of this same purpose. The number of its participants has increased from 61.9% in 2007 to 64.2% in 2015. (Roonemaa and Lõugas, 2015). In Table 1.1 it can be seen the statics about electronic voting in Estonia. As seen on table voting with mobile id started in 2011.

Table 1.1 General statics of Electronic Voting in Estonia (Vabariigi Valimiskomisjon, 2015)

	Local Elections 2005	**Parliamentary Elections 2007**	**European Parliament Elections 2009**	**Local Elections 2009**	**Parliamentary Elections 2011**	**Local Elections 2013**	**European Parliament Elections 2014**	**Parliamentary Elections 2015**
Eligible voters	1 059 292	897 243	909 628	1 094 317	913 346	1 086 935	902 873	899 793
Participating voters	502 504	555 463	399 181	662 813	580 264	630 050	329 766	577 910
Voter turnout	47,4%	61,9%	43,9%	60,6%	63,5%	58,0%	36,5%	64,2%
voters	9 317	30 275	58 669	104 413	140 846	133 808	103 151	176 491
I-votes counted	**9 287**	**30 243**	**58 614**	**104 313**	**140 764**	**133 662**	**103 105**	**176 329**
votes cancelled	30	32	55	100	82	146	46	162
voters using mobile-ID	n.a.	n.a.	n.a.	n.a.	2 690	11 753	11 609	22 084

In the last general elections of Turkey, 2015, while the number of local registered voters was 56.949.009 the number of votes was 48.537.695 According to the data from Supreme Electoral Council of Turkey the number of invalid votes was 697.464 so the valid ones were 47.840.231. (Supreme Electoral Council of Turkey, 2015) Thus %1.43 of the votes were invalid. When this number is compared to the last elections in Estonia, 2015, although in Estonia e-voting system was performed, for a total of 176,329 votes, just 162 of them were invalid.(Vabariigi Valimiskomisjon, 2015) The detailed numbers are shown in the Table 1.1. These numbers correspond to 0.09% of the votes were invalid. With respect to this information, in Turkey the percentage of invalid votes are nearly 15 times higher than Estonia. It shows the power of electronic voting.

Electronic voting could speed up the process of counting votes, offer accessibility for disabled voters and voters in remote locations, and it could increase the security and reliability of elections.

Fully manual, paper-based voting processes can be overwhelming, time-consuming and prone to security breaches, as well as fraud. Also, they also are prone to having more issues, such as lost, stolen, or miscounted ballots. Furthermore, some votes could be lost because they are marked unclearly or invalidly. For disabled people, it is hard to vote in limited accommodations. There are plenty of other problems with the paper method – they can be inconvenient and unfair, and they can lack mobility with anonymity. To solve these issues, the concept of electronic voting was introduced.

Recent advancements in mobile-based communication networks and cryptographic techniques have made it possible to consider mobile voting as a feasible alternative for conventional elections. Mobile voting has the flexibility of allowing citizens to participate in an election no matter where they physically are during the voting process.

Benefits of this alternative may include a reduced cost and increased participation, speed, flexibility and accuracy, as well as improved accessibility for disabled people. According to Chung and Wu (2012), mobile voting schemes should all have anonymity, eligibility, fairness, mobility, uniqueness, verifiability, uncoercibility, limited transparency and appropriate location freedom.

Some standards have been published by official institutions regarding electronic systems. Such as The Council of Europe's Standards (2004), NIST Standards (2009) etc. Although, there are none about mobile voting, the standards for e-voting are applied and discussed for mobile voting literature, too. These rules include:

Democracy - The system enables the voting procedure to be executed in a democratic way by providing equity among the voters. To follow this requirement, only registered and authenticated voters can vote through the system. Suitable conditions should be provided for voters to cast their votes, such as different language choices for foreigners and special aid for disabled folk. Also, there should be a convenient way of assessing missing and early votes. All voters are required to submit their ballot once, and if someone does not want to vote, it should be their choice. Thus, they should be free to choose to vote, or not.

Privacy – One of the most important aspects is the privacy of the votes. During the voting process, all polls should remain secret. Neither the votes nor the voters should need to reveal the other, and each individual's submission should never be disclosed.

Accuracy – The ballots should be detected accurately to reflect the choices of the voters objectively. Votes should be recorded by the election commission server without being eliminated, duplicated or altered.

Fairness – Neither the number of votes for each candidate nor the partial tally results should be announced or accessible from anywhere, unless the predetermined official voting time has expired.

Security - Complex mechanisms must be devised to prevent man-in-the-middle attacks, and they should also provide security to the server, as well as security between the client and the server. Furthermore, the mechanisms should make data transactions secure by encrypting them without altering the data.

Integrity - Data corruption should be avoided in order to protect valid votes that are not modified, replaced or deleted.

Consistency - Every component should be consistent with each other. In other words, components should all be reliable within certain parameters.

Authenticity - Each user should be able to verify the system to ensure its reliability. On the other hand, the system should also verify the identity of every user before collecting their votes.

Eligibility - Only the authorized voters are allowed to cast their votes.

Mobility – Any vote can be cast from anywhere. Voters should not be restricted by requirements involving their location.

Uniqueness - Each voter should only be able to cast only one vote in each election.

Verifiability - There should be a convenient way for voters to verify that their votes are collected for the final election result.

Uncoercibility - The voting process should not be done under any pressure implied by another individual or mechanism. Any incident of bribery should be handled so that only legitimate votes are collected.

Cost-effectiveness - In order to collect the votes from every single user, the election system should be carried out in an efficient way, and it should be affordable for each user. Cost effectiveness is a very important issue because there are fewer requirements for mobile devices than immobile devices.

With the help of the study behind this thesis the goal of turning the aforementioned system into a product could be achieved. The long-term objective is to design a system that will be used by the government. This thesis also has been designed to be reliable and efficient enough to be used by the government. Another crucial motive of this thesis is becoming applicable for the elections of other institutes and universities besides the government. Its design, speed and flexibility support its capacity to be used by other institutes. For instance, a company can constitute its own voting system through an application so the company's annual growth rate can be submitted by its employees. Also, while selecting a suitable manager for any department, the company can again use this system. The application can be used by the students in the universities while selecting the students' president in a secure way. Behind the system that is constructed within this thesis, a separate plan was developed for an institution, department, and a group. To gain a membership in any of these parts, an authorization system was constituted. To avoid wasting long hours in a large-scale voting setup, quick and real time voting can be integrated in ways such as making decisions during the meeting or asking multiple questions in one survey. In the latter situation, only one electing process would be needed, so, overall, less time and effort would be consumed.

With the developed systems in this thesis, the following electoral examples can be implemented:

a) Parliamentary election in the Republic of Turkey

Authorized users: the citizens of the Republic of Turkey

b) Selecting the rector in “X University”

Authorized users: All of its faculty members, except the research and teaching assistants in “X University”

c) Selecting a job security representative in “X University”

Authorized users: All of the faculty members, all personnel from its general secretary except for one person, and one person from the student affairs department.

d) Selecting a representative for the research assistant of the computer engineering department from “X University's” engineering faculty

Authorized users: Only the research assistant of “X University's” computer engineering department

e) Voting for workers' annual rate of salary in “Business Y”

Authorized users: The board of directors in “Business Y,” its human resources and financial affairs departments’ one person from its technical staff.

f) Voting via answering yes/no questions and other multiple choice questions

The moderator of the meeting would start a ballot instantaneously and authorize the participants. After all of the participants have cast their votes through the application, the results would be displayed.

The system's background and its management panel were created in the .NET MVC5 framework. In Appendix B it can be seen the entity relationship database diagram. The software was developed for users (client-side), using the Apple iOS platform. The application is named as “M-VOTR”. The interaction between the client and server sides was provided by the output of the .NET web services, AJAX and JSON. In the certification phase, the OpenSSL PHP functions are utilized. On the client side, the hybrid technique for mobile applications was used with the JavaScript jQuery HTML5 coding language in the Cordava platform.

The network security between the client and server was administered through the integration of SSL to the network. The SSL service RapidSSL is included the whole system and all data transmits over https:// link. The network then started recognizing its users through certificate authorization, and in this step, OpenSSL functions were used. The Elliptic Curve Curve25519 key exchange mechanism was used to share the key that enabled the security of the transmitted data in questions and answers. The data were first encrypted by AES with using 256 bit session key which is derived from Curve25519, then data started to be transferred, and was compared to that of HMAC. In order have biometric authorization, fingerprints were used. For this type of authentication, the iOS Touch ID feature was used. In it, a unique device ID was activated in multiple encrypting steps, and all iOS devices had this feature. Similarly, a fake password (trapdoor) system was developed to extract votes which are given under pressure. Users predefined legitimate and fake character passwords, and genuine and fake geometric pattern passwords, in that system. If it the user felt pressured into submitting an alternative opinion, then they would enter their fake identification information. The system would not warn the user that they have entered a fake password, but their vote would be invalidated. The system's structure will be explained in detail in the upcoming chapters.

In this work system sequence diagrams drawn with Visual Paradigm Tool to illustrate client-server interactions; self-returning arcs indicates that the process is implemented at the returned side, arcs which goes to the opposite side shows that data is transferred to the directed side, dotted self-returning arcs indicates that data returns after their process is finished on the opposite side.

The rest of the thesis is organized as follows: Previous studies about the mobile voting application is explained in section 2. General information and background of the materials and algorithms that are used in this thesis are given in section 3. In section 4, system architecture of the algorithms which are developed for the thesis is explained. The specifications of the methods and the algorithms that are used in the thesis are clarified in section 5. Section 5 Computational results are illustrated. In the last section, general evaluation of the system and future works are summarized.

2 LITERATURE REVIEW

The initial concept of an electronic election scheme, and a real one, were suggested for the first time by Chaum in 1981. Even though the first actual studies of electronic voting started in the 1990s and didn't become popular until the 2000s, Chaum was the leader who discovered the first steps of creating an electronic voting system back in 1981. The general purpose behind his idea was to solve the problems in traffic analysis using a public key infrastructure. With that intention, the author proposed that each participant should trust a common authority and that each participant should be an authority. During that study, the concepts of electronic voting and the privacy of votes have become significant. The author also proposed that anonymously mailed ballots should be signed from a roster of registered votes.

2.1 Researches based on Cryptosystems

(Xun et al., 2006) studied a method that used a modular square root and blind signature schema for obtaining the administrator's signature on its users' votes. The administrator could not see any details of the ballot they were signing. Their plan needed low computation complexity in mobile devices. It is assumed that the administrator and counters never would collude with the base station when using that strategy, although, messages can be altered or halted. The electronic voting tactic in that paper was composed of six phases, which were setup, registration, ballot application, ballot casting, tallying, and confirmation.

In 2013, (Ullah et al) used a hybrid cryptosystem to combine the benefits and dynamics of symmetric and asymmetric encryption. Their work, which was based on the phases of online registration, vote casting, vote collecting and result evaluating, was also designed to work together with a paper ballot voting system. One of the critical points made in that paper was that it may be necessary to vote offline in an electronic setup instead of using the traditional online voting method, when the latter fails. In such a scenario, one could vote offline via an SMS (Short Message Service), which is a well-known and very common technology. That paper proposed to design a mobile phone for voting. Such a phone would use SIM and GSM, which would be secure and globally used, and it would ensure that the voters' identities remain private. Users would register with their NIC and SIM Card, which would be verified from a mobile phone operator. Their information would be stored in two updated databases of the election commission server; the first database would keep NIC information, and the second one would keep SIM data.

According to (Khelifi et al., 2013), users' mobile phones and the internet together are enough to study what makes a suitable application. All of the application's requirements were defined through a survey given to people who participate or have participated in elections, and to the election officials. The election officers, who manage the voting process, were thought to be qualified enough to elicit adept requirements for the system. The server side connected to the government's identity authorization server, which checks the voters' identity. In the study, a modified AES algorithm was suggested for a security mechanism. The algorithm's substitute byte and shift row would remain as they were in the original AEX, and the mix column would be changed with a 128 permutation operation, then an "add round key" operation would be implemented. The 128-bit AES would be used by both the client and the server. In the algorithm, sets of voting periods exist, and voting would have to be done in those sets only. The voters suggested that fingerprints be used to enter the voting control panel. The system would then just accept one vote per individual, and it would not allow a re-vote. The publication mentioned a plethora of cryptosystems which were tested in its encompassed studies, and Pailler's Encryptions (Paillier, 1999) was one of them.

(Ying and Zhu, 2009) created a system which used Pailler's encryptions and the standard cut-and-choose technique was applied to eliminate the computational zero-knowledge proofs. The authors searched the previous electronic voting systems, like the blind signature one, and encountered the Paillier's cryptosystem and zero-knowledge proof techniques, which were proposed by Baudron et al. in 2001. However, they found that zero-knowledge proof techniques were not very efficient. Thus, they searched for a way to dually ensure efficiency and security in a system. Their study considered the sufficiency of mobile devices with e-voting systems, which were explicitly executed by the larger-scale CalTech/MIT voting system**.** Their design criteria tried to minimize the cost of otherwise expensive computations as much as possible, so, in order to do so, they utilized cut-and-choose techniques.

Most people do not want to vote when they are on a holiday or a trip. So, (Biswas and Sujit, 2015) sought to create a mobile voting system which would allow users to vote independent of their location. GSM was used to access the user's location. For security concerns, they suggest a two-step security protocol. They used the blind signature method in message transmission and assumed that SMS security would be ensured by the GSM operator. SIM cards and national identity cards were encrypted through a symmetric key cryptosystem, and their data was sent to the server. After the server received it, the user got a pin code. On the voting date, the user would decrypt the pin code and give their vote. Pailler also proposed for his encryption technique to be used in all of the encryptions in the given protocol.

(Chang and Lee, 2006) presented a voting mechanism that used a blind signature scheme and the Diffie-Hellman key exchange protocol with a public proxy server. The mechanism ensured that the public proxy server could replace the network address of any ballot with another address.

(Li and Hwang, 2014) decided to use the Diffie-Hellman key exchange method to ensure voter anonymity and performance efficiency. They examined Chang-Lee's e-voting scheme, then they presended that Lee's scheme opened or hacked the votes, so they created an e-voting scheme by improving that system.

(Ahmad et al., 2009) stated that instead of using hybrid symmetric methods or RSA, an elliptic curve cryptography algorithm could be used to secure votes. They said *"The primary advantage that elliptic curve systems have over systems based on the multiplicative group of a finite field (and also over systems based on the intractability of integer factorization) is the absence of a sub exponential-time algorithm (such as those of "index-calculus" type) that could find discrete logs in these groups. Consequently, one can use an elliptic curve group that is smaller in size while maintaining the same level of security. The result is smaller key sizes, bandwidth savings, and faster implementations, features which are especially attractive for security applications where computational power and integrated circuit space is limited, such as smart cards, PC (personal computer) cards, and wireless devices (*Koblitz, 2000).*"* The most important reason to choose the elliptic curve cryptography for this research is in having smaller keys than in public key cryptography. With respect to their studies, the authors tried to prove that the ECC-based scheme performed better than traditional hybrid schemes of symmetrical and asymmetrical cryptography. Elliptic curve cryptographs, with their smaller key size, may have enough to provide ecc256 medium-term security. This study differs from the others by using two counters to strengthen the data security. Each counter has different key pairs which are used to encrypt/decrypt users’ messages later. These counters work on the votes without looking at their submitters' identity. Furthermore, the counters do not keep the users’ identity data, because it is only accessible to the administrator. As a result, this paper states that having 15-second encryptions for ECDH-256 and AES-128 is not acceptable, but having a 3-second encryption for ecc-256, which is also mentioned in this paper, is acceptable and improvable. (Monaly Shetty et al., 2015) released a written piece which was similar to this one, and it had the same name. It represented one of the mobile voting schemes that used elliptic curve cryptography, and it compared the RSA vs ECC voting schemes.

2.2 Researches of Biometric Methods on Mobile Voting

Biometric methods in voting systems became popular as biometric technologies, such as fingerprint analysis, have started to be used in mobile phones. In one of these studies (Donovan and Suresh, 2011) suggested using finger prints, and they are also mainly used in health and government sectors, as well as many others. The security part of the system covered public key encryption by using finger prints in place of signatures and passwords for identifying unique user information. Similar work to theirs was done in 2014, which especially concentrated on preventing bribery and coercion by using trapdoor authentication and virtual receipts. (Alrodhan et al., 2014) Trapdoor authentication gives a password or key of two types; one was the real password and the other was the fake password. Therefore, if any individual would be forced to vote by someone else or would have to vote under someone else's influence, they can use the (pre-declared) fake password or key so their vote would be nullified. In their method, the RSA blind signature and GPS methods secured the signatures. The layout of all voting systems includes an authentication center and a ballot center. Authentication centers register the users in the system and provide the first physical control. Ballot centers collect the votes of legitimate voters after they are authenticated. With the aforementioned authors' recommendations, VeriFinger Embedded SDK would be used to scan finger prints, and an android platform would be used to implement them. Fingerprint verification is not the only biometric method that is used; there are many different ones that exist.

(Kao et al., 2011) designed a voting scheme based on a simple eye gaze calibration procedure, which reduced computational complexity. The proposed method doesn’t need multiple spatial coordinates to estimate the parameters of the camera. The authors stated that an iris's location can be determined easily by using the voting weight. The proposed method didn’t use facial recognition techniques; instead users can choose verification options with eye tracking. The proposed method can be useful for disabled people who cannot use their hands.

Ready-to-work systems and common and large packet systems can be defined as common off-the-shelf (COTS) systems. (Thakur et al., 2014) proposed a mobile voting model which used a COTS system with Near Field Communication (NFC) and pragmatic biometric verification schemes. NFC schemes enable devices to communication with each other without physical contact. This technology is used by smartphones and tablets (etc.). Contactless communication sends information via waves between compatible devices, and it does not require the devices to touch, or multiple steps to connect. NFC programs use radio-frequency identification (RFID) technology so that devices can be held a short distance (a few centimeters) away from each other and small amounts of data can be transferred wirelessly with just a little power. The novelty of the research findings was in how the unique storage capability and auto-coupling NFC features can be exploited. Auto-coupling is an original feature of NFC that removes all navigation links. In order to start a voting application, the NFC device tags onto document. To vote within a country's borders, GPS is used in their proposed system. In their system, voters also enrolled by capturing their biometric data onto an NFC tag for baseline verification, like in the previously discussed methods, which included voice, fingerprint, and facial recognition. (Ok et al., 2010) also used NFC for a mobile voting scheme, and they developed a remote-style, electronic mobile voting system named NFC Voting. In their system, a voter touched his/her NFC-enabled device to a NFC tag, and the information in the tag was transferred to their mobile device. Then, the user entered their unique key (an access code), and the vote was transferred to a validating server in the internet where it was counted. In their scheme, a NFC tag could be in a voting room or on a poster and pictures. NFC technology can be used efficiently up to a distance of 4 centimeters, though the creators of that system thought that mobile devices should be brought closer to the tags. At the end of the recollection, the experimental survey results were presented. The aim of the study was to vote by approaching a NFC tag.

(Mohit et al., 2014) conducted a similar study that used a QR code instead of the NFC tags. Again, the voting process was done by approaching a pre-determined area or location, but this time with more simple technology. The QR codes are a common technology used for many things. They are matrix barcodes, and they can be decoded easily. Toyota developed QR codes in Japan in 1994. In that study of QR codes in the voting system, the codes matched the users and the voting server, and they encoded the voters' information. QR codes captured the verified information, decoded it, and sent it to the server for authentication.

Sivagami (2011) focused on the performance of the system. In Sivigami's paper, the authors expressed their goals to advance the performance of all voting systems and to develop a system based on grid management, which they named Vote Grid. The proposed system could work with mobiles, table PDAs and laptops. The term "mobiles" did not solely apply to mobile phones; laptops were also covered in mobility. It was stated that this system was not only for government polling, but it also was intended for any other multi-discipline polling. The paper proposed that in generating a widely-distributed voting application that allows users to vote through a mobile network, the application could be used to study issues in human judgment and decision-making within varying decision contexts. The Vote Grid structure included servers to submit ballots and authenticate, as well as having voting booths and grid sites, which were defined as the hearth of the Vote Grid system. All of the filled out ballots were kept in a grid management server, and the results were tallied. The information would then go to another suitable program in case the node/resource task failed, therefore, it had a degree of fault tolerance. According to the authors, the proposed mobile voting mechanism fulfilled all essential requirements, including guaranteed eligibility, unique authentication, privacy, accuracy, availability and verifiability, and it also has better performance than other related networks. As a result, they believed that the proposed voting mechanism could be practically applied over the internet.

(Kumar et al., 2011) created a GSM based structure which was used to connect mobile devices and the GSM network. Also, a subscriber identity authentication feature was provided by standard protocol. The generated keys, which provided security between the server and the voter, were stored in SIM cards by using the (HLR) Home Location Register method. The HLR was the main database of consistent subscriber information for a mobile network. A digital signature, blind signature and bit-commitment mechanisms were among the used schemes. There are some assumptions about their plan, such as that voters would have been capable of using different methods while they are voting, and that the mobile operator was trusted to authenticate for the mobile users, in the purpose of voting and sending the correct information to Voting Server and Counting Server. Only authorized voters can submit their opinion, and all of the ballots remain secret during their submission, while each individual's vote cannot be linked back to who cast it. The plan enhanced security and was more mobile and convenient for voters, and the voters' privacy was protected through a blind signature scheme.

(Campanelli et al, 2008) created their own mobile voting protocol, named M-Seas, which was based on Sensus protocol (Cranor and Cytron., 1997). The protocol was indicated to be a series of complex systems, and by using M-seas, the authors tried to avoid Sensus's vulnerability – administrators of the electoral process could cast votes as the eligible citizens who have before claimed that they would abstain from voting. The authors also overcame another one of its vulnerabilities in which the validator could do the same thing by using M-seas. After their modification, ballots were to be signed with private keys. In order to have an accepted vote, the only required procedure was to be a registered user, and the user could not vote twice. In that proposal, Java was used for the application, and Signet Mobile was used for its cryptologic operations. A library of cryptography was developed to use some advanced mechanisms, such as the blind signatures. Also, they proposed a user interface which included the XML parser.

In this thesis' study, the client's application could have been made for the IOS operating system, and the Android platform was also a possibility. In order to get full performance from the application, the IOS operating system was chosen. The application will be tested on iPhone devices especially iPhone 6. Also, in different studies, iPhone applications have been developed. For an example, (Campbell et al., 2011) developed an iPhone application for mobile voting and compared its usability with traditional voting platforms. Then they tried to assess the usability of a voting system. The results of the paper indicated that the proposed system was not as efficient as other voting methods - the authors measured the total interaction time using mathematical equations, and the mobile voting system was 90 seconds slower than non-mobile voting systems. However, the Smartphone owners committed fewer errors on this system than on the traditional voting systems.

There are too many mobile voting applications which don't succeed in securing mobile voting processes. Most of these applications use wireless networks to obtain mobility. WPKI security and the potential complications of mobile voting were analyzed report written by Jaak (2010.). The researchers mentioned problems which can occur, such as man-in-the-middle attacks and those in having a one-time password. After analyzing these problems, they set requirements and gave recommendations to take care of these issues.

(Velapure et al., 2015) published their plans for a mobile application that used an Android operating system. This study left a lot of uncertainty in minds about the security of its proposed Message Digest 5 algorithm. In the system, the voter could vote from any location, but security algorithms were used as external attacks on the system. More specifically, MD5 was used. The algorithm took in messages of arbitrary length and produced a 128-bit "fingerprint," or "message digest," as an output. It is conjectured that it is computationally unfeasible to produce two messages with the same message digest, or to produce any message with a pre-specified target message digest. The MD5 algorithm was intended for digital signature applications, where a large file must be "compressed" in a secure manner before being encrypted with a secret key under a public-key cryptosystem, such as Koblitz's RSA from 2000. A one-time password was part of its user authorization, along with facial recognition and national id numbers for security. Using Android may be a good opportunity for electronic voting, but the MD5 algorithm is not secure enough, especially in 2015.

2.3 Multi-disciplinary and Specific Areas

As mentioned before, mobile voting could be used not only for governmental operations, but also for universities, hospitals, and firms. In literature, some studies exist in specific and multi-disciplinary fields.

One of these studies is of Campus E-Voting, which was both an Android and a web-based application that was based on a mobile voting system specifically for campus elections (Pandit et al., 2014). With this system, students could give their vote from anywhere. The web application was developed with JSP, which is also compatible with an Android platform. Both applications used Java, so that updates could be carried out easily. For the system's security, the authors proposed to use steganography. Steganography hides binary data within an image while adding a few changes. The purpose of using steganography is to provide secrecy.

Least Significant Bit (LSB) stenography algorithms also have been used to secure vote transmission. Voting can be used in different fields, like the medical one. (Ooijen et al., 2014) designed an image-based mobile voting system that was used in radiology classes, but it was not directly related to electronic or mobile voting systems. The authors proposed a novel audience response system, and medical students' experiences of mobile voting were given. In the application, professors asked questions with pictures, and users gave answers instantly from their mobile phones. The results of the voting were kept in statistical structures. This methodology can be utilized whilst developing a mobile voting system, with regards to different experiential aspects that its users had noted before.

(Meida et al., 2013) suggested their idea of an android-based, mobile-terminal, voting system with the intention of solving some of the problems of traditional voting and e-voting systems. They combined static and dynamic password mechanisms together to create a double-verification scheme that was based on a symmetric algorithm. The proposed scheme defeated weaknesses that were found before in the process of acquiring a double identity authentication, and it tried to get rid of replays and personal attacks. The problems of existing voting systems were researched, and a new scheme was presented in order to avoid them. There were two innovations of the proposed system. One of them was in having three types of votes: those in a traditional election, a competitive election and a grading election. The competitive election is for race between voters and in grading election authorities grades the users. The other innovation shortened the voting time by sending different votes to the corresponding voters at the same time.

3 BACKGROUND

3.1 Security

3.1.1 Symmetric Cryptosystems

Symmetrical cryptosystems use the same key to encrypt and decrypt data. One side encrypts plain text with a key, and then generates a cipher text. The cipher text is sent to the other side. The other side must have the same key to decode the cipher text. The key must be shared with a secure channel. Symmetric encryption can be classified along three dimensions. Stream ciphers and block ciphers are the more common algorithms for symmetrical cryptography, and a block cipher operates on a plain text block of n-bits to produce a cipher text block of n-bits. Most symmetric block ciphers are based on the Feistel cipher structure which can be seen Figure 3.1.

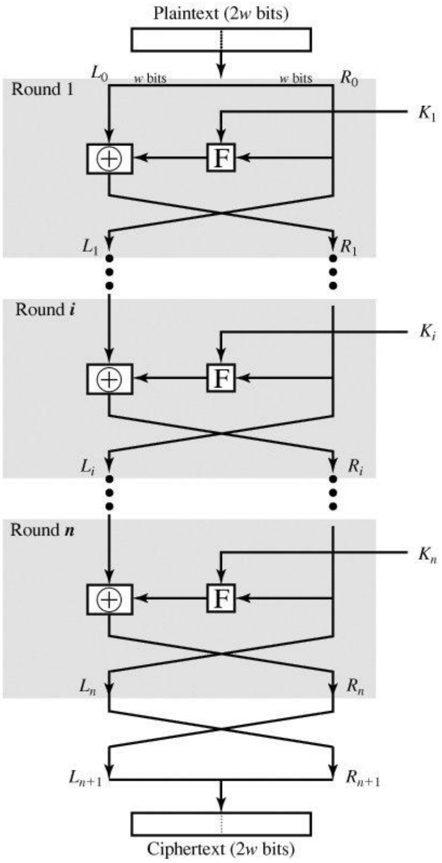


Figure 3.1 Feistel Structure (Stallings, 2005)

DES, 3DES, and AES are symmetric algorithms which are frequently used, like the AES that this thesis urges. Such applications seek help in concealing their users' private information, but they all still want their data to be conveniently converted and accessed within their system. They just want the job to be easy for them, and for their users, while outside forces are thoroughly flummoxed by their secret messages. Here, the inner-workings of these three codes are decoded.

Unsurprisingly, the symmetric algorithm trio was all birthed by the same institute, NIST (the National Institute of Standards and Technology), though not all at once. The first one to come was DES in 1977, which takes in 56-bit keys, then spits out 64-bit keys through a set of reversible phases. It would have been designed to reassemble the keys in the same size, but, the key size would have been highly contentious. (Koltuksuz, 2012) In fact, Diffie and Hellman wrote in 1979 about the controversies involving the code lengths of DES.

In 1999, DES got an upgrade and became known as 3DES. 3DES handled bigger keys than its predecessor, and they were 168 bits. It is 56 x 3, it can either use same key with 3 times or different keys. However, NIST soon wanted to replace DES altogether, and it released AES in 2001. AES was the one chosen to be behind the symmetric encryptions needed by the suggested program in this proposal. The Advanced Encryption Standard (AES) sets a block length of 128 bits and a key length of 128, 192, or 256 bits. Although, the most commonly used of the three key sizes is 128 bits, so, it will be used in the approaching example which will describe how AES acts

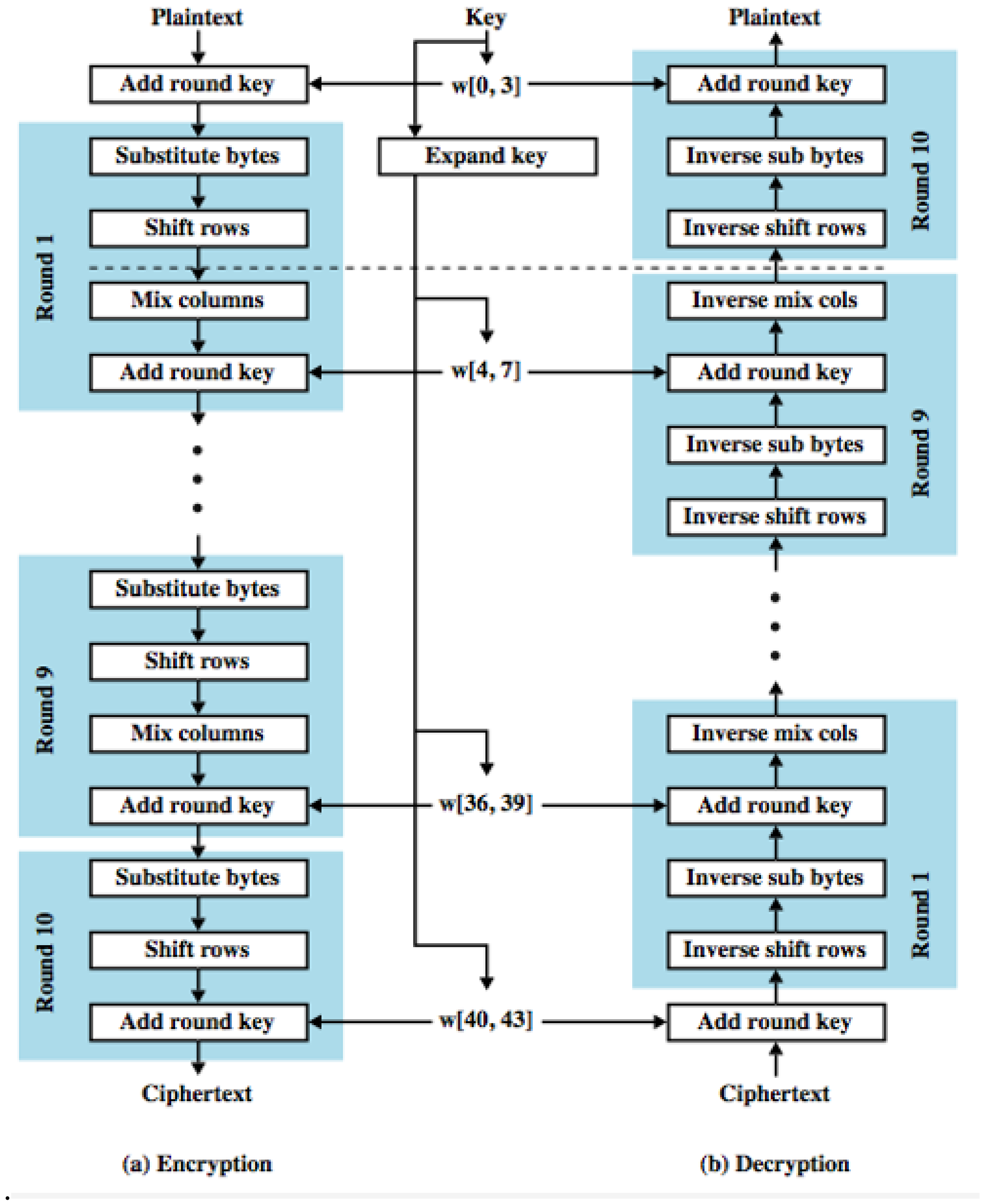


Figure 3.2 AES Structure (Firat, 2007)

FIPS PUB 197 (NIST, 2001) portrays a byte mold in the form of a 128-bit rectangle. It is a representative shape of the data that AES handles, coming in and going out. With each spurt of alteration, AES jumbles its code while keeping it crammed into its block. Zooming in on the action, the entire code-string gets stretched out, portioned, and sorted. First, the string is broken down into chunks, which are 32-bit Plain Text words. Next, each word is deconstructed into 4-bit fragments (A.K.A. - XORs), which are respectively and singly dealt out to each column of the rectangle shape. Then each XOR is scrambled 44 times, then 52 times, then 60, and sometimes AES may replace a byte with its own. If every 4-bit piece was reset to its original form, then a series of words could be read if the columns were horizontally observed while switching across each every four characters. After all of that reassembling, there is an “add round key” point, and then the procedure is carried through twice more. The altercation formula next has 9, then 11, then 13 rounds to go through, and in the last sets, it will transfigure them in pulses of 10, 12 and 14. In each warping pulse, two operations are committed, byte substitutions and color mixing, and an “add round key” operation tags on the end of every third stage.

3.1.2 Asymmetric Cryptography

A pair of keys are created and used together in an asymmetric algorithm, and each pair is made just for one strand of information. One key is labeled to be “private,” used to decrypt information, and cannot be shared, and the other is dubbed the “public key,” which is free to be exposed as its owner pleases. The latter kind is more commonly used for encrypting, though both are suitable for the job. Asymmetric Cryptography scheme is illustrated in Figure 3.3.

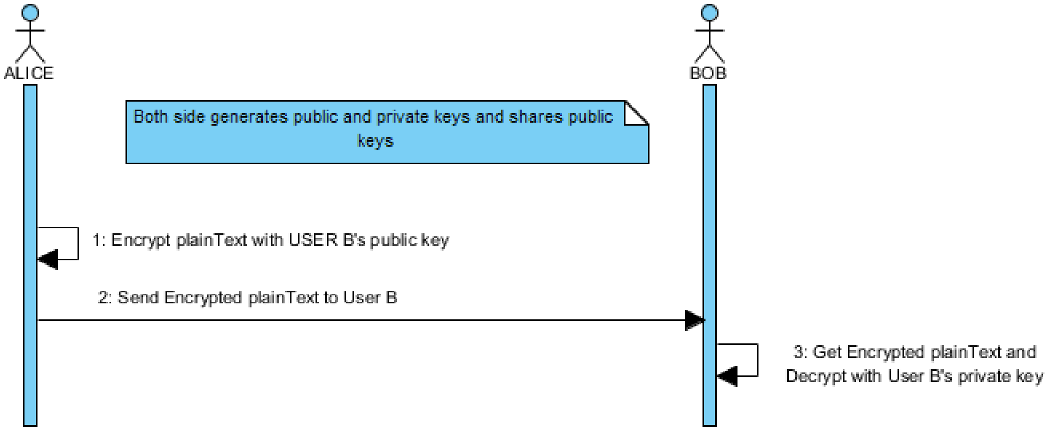

ALICE
BOB
Both side generates public and private keys and shares public keys
1: Encrypt plainText with USER B's public key
2: Send Encrypted plainText to User B
3: Get Encrypted plainText and Decrypt with User B's private key

Figure 3.3 Asymmetric Cryptography Scheme

The client and server obviously can generate and share both of their public keys, but those who get a public key can also generate and share keys created from their received key. However, when two parties exchange a coded message, both must know the public key. Figure 3.4 involves (the sender) Bob and (the receiver) Alice. Bob sent Alice a public key, then Alice used it to generate a new one (in Plain Text), and she responded with a new code. Bob could then use his private key to decrypt Alice's response, but Alice would not be able to, because her public key is an added outer shell to Bob's private key.

The Diffie-Hellman key exchange protocol (Diffie and Hellman 1976), the ElGamal which was created by ElGamal as a legal descendant of the DHKEP in 1985, and the 1978 RSA (by Rivest, et al.) are the most frequently-utilized asymmetric cryptosystems.

RSA is one of the most famous and used public-key systems, and it handles large keys and integers. In fact, the digits and codes work exponentially, though finitely. Here's how it performs

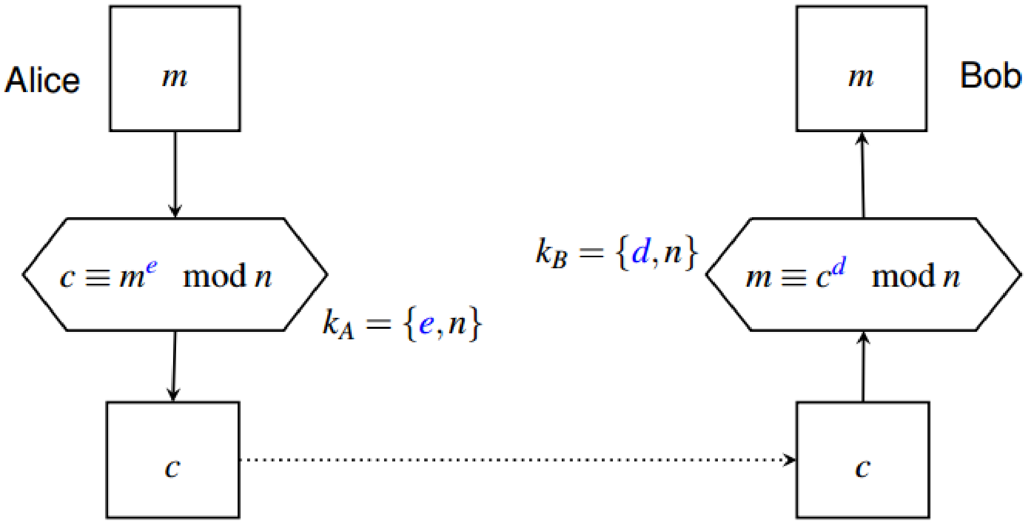


Figure 3.4 RSA Scheme

To encrypt a message, sender M obtains public key of recipient $PU = \{e, n\}$ and computes $C = M^e \, mod \, n$, where $0 \leq M < n$

To decrypt a cipher text C, receiver uses their private key $PR = \{d, n\}$, and computes $M = C^d mod \, n$

Random numbers and prime numbers play an important role in the use of encryption for various network security applications. In this thesis, off-the-shelf libraries such as Java BigInteger ports were used for generating random and prime numbers. Users generates public/private key by selecting to random large primes p and q, than computes the system modulus $n = p.q$, $ø(n) = (p - 1)(q - 1)$, the encryption key e generates where $1 < e < ø(n), gcd(e, ø(n)) = 1$ and solves $e.d = 1 \, mod \, ø(n) \, and \, 0 \leq d \leq n$ with decryption key d. and publish their public encryption key : $PU = \{e, n\}$ and keeps secret private decryption key $PR \; = \{d, n\}$.

3.1.3 Key Exchange

If any information is exchanged between the client and server, both sides should have a key for encryption, and one for decryption. In symmetric cryptosystems, the same key is used for both processes, but directly sharing this key between the sides is not appropriate. The security of a symmetrical cryptosystem depends on the security of its keys. In order to securely share in such a system, several mechanisms are utilized.

The key that is shared between its users is called a shared key, or session key. The key which is used at the end of the encryption process is not the session key; it is the key that was produced by using a session key. Therefore, it can be prevented from cryptanalytic attacks.

Generally, key exchange mechanism can be expressed as follows.

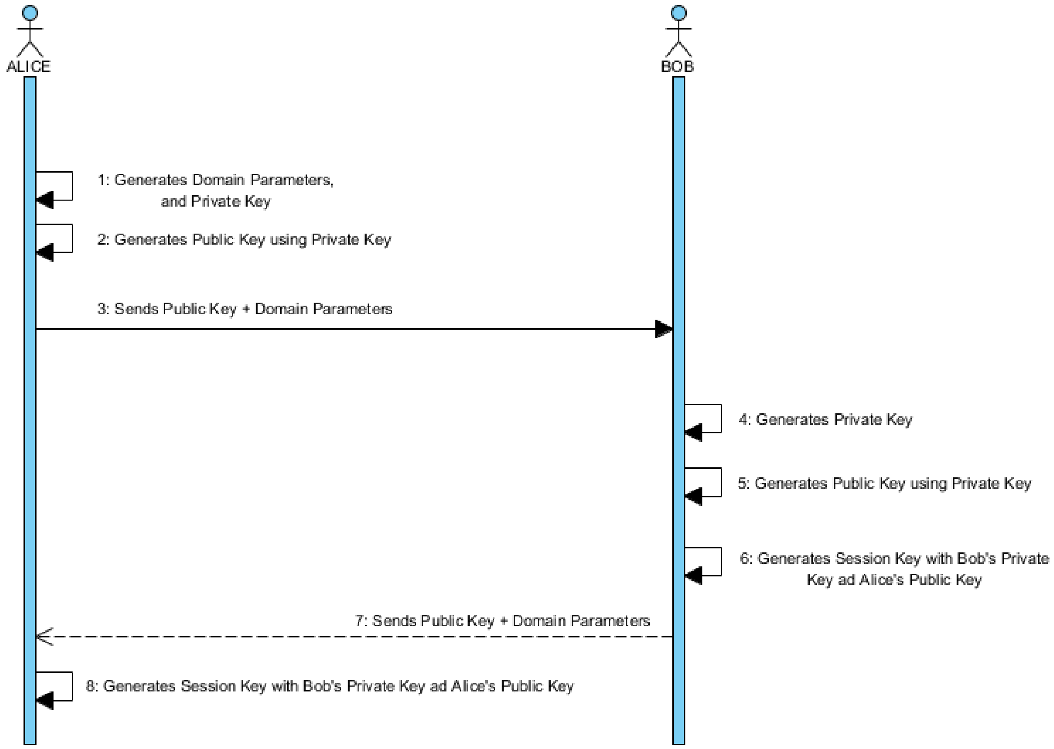

ALICE
BOB
1: Generates Domain Parameters, and Private Key
2: Generates Public Key using Private Key
3: Sends Public Key + Domain Parameters
4: Generates Private Key
5: Generates Public Key using Private Key
6: Generates Session Key with Bob's Private Key ad Alice's Public Key
7: Sends Public Key + Domain Parameters
8: Generates Session Key with Bob's Private Key ad Alice's Public Key

Figure 3.5 Key Exchange Scheme

3.1.4 Diffie-Hellman Key Exchange

In 1970, Williamson first described the idea of public-key cryptology, which was a method that could make an exchanged message secret, but it was hushed, even though it was only a numerical code. It wasn't until 1987 that his concept could be unveiled, and this didn't happen until years after Diffie and Hellman published the first public-key algorithm in a seminal print. Consequently, Diffie and Hellman were accredited to defining that type of cryptography when they released the Diffie-Hellman key exchange formula.

An example of Diffie Hellman key exchange is shown in the Figure 3.6.

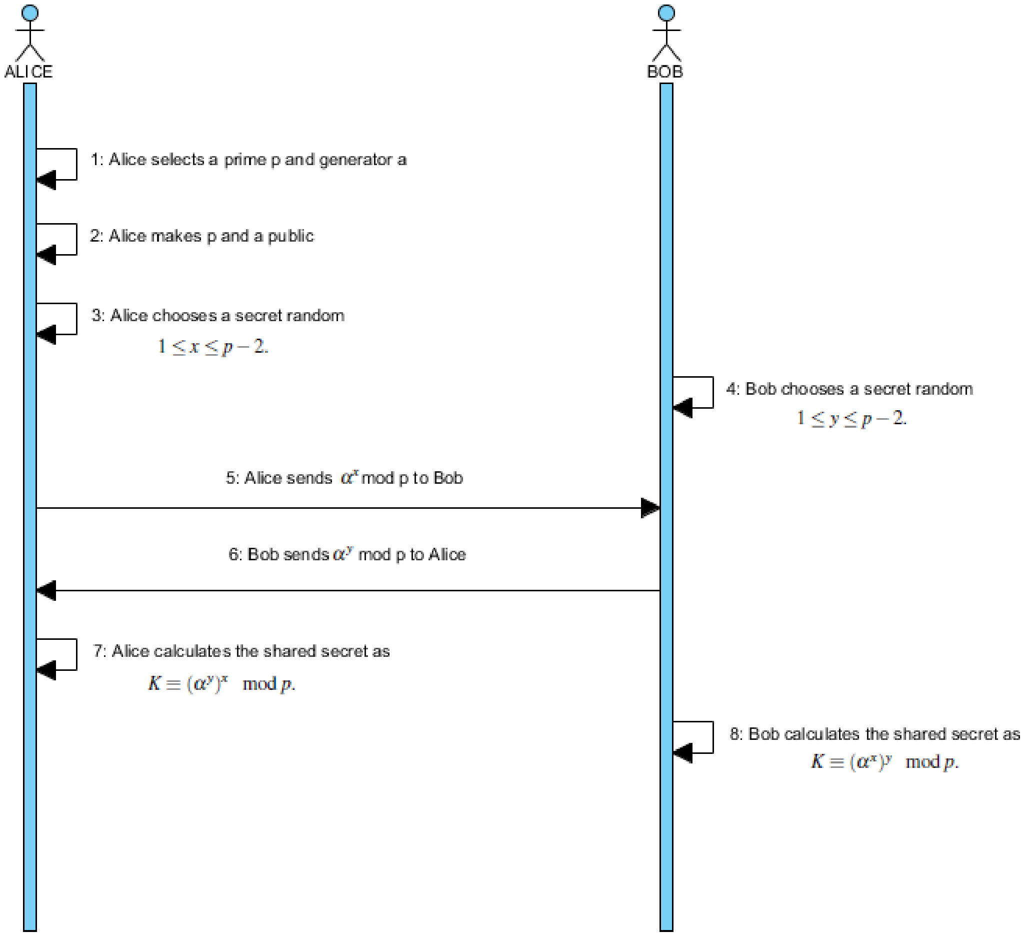

ALICE
BOB
1: Alice selects a prime p and generator a
2: Alice makes p and a public
3: Alice chooses a secret random
$1 \le x \le p-2.$
4: Bob chooses a secret random
$1 \le y \le p-2.$
5: Alice sends α^x mod p to Bob
6: Bob sends α^y mod p to Alice
7: Alice calculates the shared secret as
$K \equiv (\alpha^y)^x \mod p.$
8: Bob calculates the shared secret as
$K \equiv (\alpha^x)^y \mod p.$

Figure 3.6 Diffie-Hellman Key Exchange Scheme

The problems that can be occurred in Diffie-Hellman key exchange one problem that one can run into is a man-in-the-middle attack.

3.1.5 Man in the middle attack

Here is an example of this type of threat: Man in the middle creates two public and two private keys. Alice, sends her public key to Bob, and man in the middle catches Alice's public key. He exchanges it with his own, and transmits his key to Bob. Bob receives the middle man's public key, thinking that it is Alice's, and calculates the shared key. The man in the middle (MITM) also calculates a shared key by using Bob’s real, public one.

Bob sends his public key to Alice. The MITM captures this, transmits, second public key to Alice, and Alice receives that key. She then calculates the shared key. The MITM also calculates a shared key by using Alice's real public key. Thus, the middle man can decrypt and encrypt all messages between Alice and Bob. In this thesis's suggestion, in order to prevent man-in-the-middle attacks, SSL will be used, and HMAC will also be a barricade for man in the middle attack.

3.1.6 Elliptic Curve Diffie-Hellman Key Exchange

Elliptic curve cryptography has a mathematical foundation of elliptic curves in finite grids. The main advantage it promises over other systems (like an RSA system) is that it saves some of the storage and transfer load by using smaller keys than them while it offers an equal degree of protection. For an example, when a public RSA key takes up 3072 bits for a certain degree of security, the same degree could be achieved by using a 256-bit ECC key. (Koblitz, 1987)

Elliptic curves are consist of points based on this equation: $y^2 = x^3 + ax + b$

Along with a distinguished point at infinity, denoted ∞.

The algorithm can be described as: Alice and Bob agrees upon to use a randomly chosen point P on curve E as a key plus on a methodology to convert that point to an integer. Now, E is an elliptic curve over Fq and P is a starting point on E

Figure 3.7. Illustrates the Elliptic Curve Diffie Hellman Key Exchange

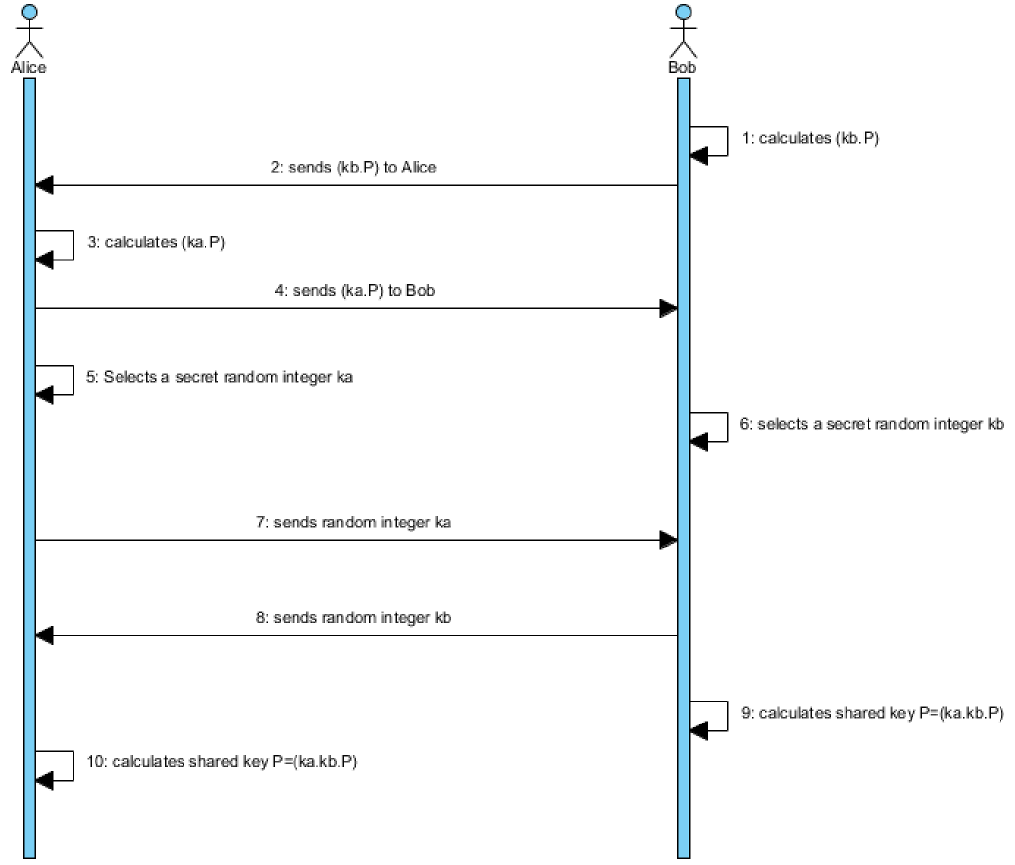

Alice
Bob
1: calculates (kb.P)
2: sends (kb.P) to Alice
3: calculates (ka.P)
4: sends (ka.P) to Bob
5: Selects a secret random integer ka
6: selects a secret random integer kb
7: sends random integer ka
8: sends random integer kb
9: calculates shared key P=(ka.kb.P)
10: calculates shared key P=(ka.kb.P)

Figure 3.7 Elliptic Curve Diffie-Hellman Key Exchange Scheme

3.1.7 Curve25519

Curve 25519 is a high-security, Diffie-Hellman elliptic curve function which set the highest speed record. It started to be used as key-exchange mechanism by many popular corporations, such as Apple Airplay, GNUnet, APPLE IOS 9.0, Android, Opera Browser, Google Chromium Browser, OpenSSH, TextSecure, Whatsapp, SigmaVPN, GoVPN, BoringSSL, and Google. OpenSSH made Curve 25519 its default key exchange in version 6.5. WhatsApp achieved support for its TextSecure protocol. It was also used as the base point in the Montgomery curve over the prime field defined by the prime number. (Bernstein, 2006)

Each user of Curve 25519, has a 32-byte secret key and a 32-byte private key. The main purpose is to activate a shared 32-byte secret key, which can be used to encrypt messages between users, or to act as a symmetric key and authenticate.

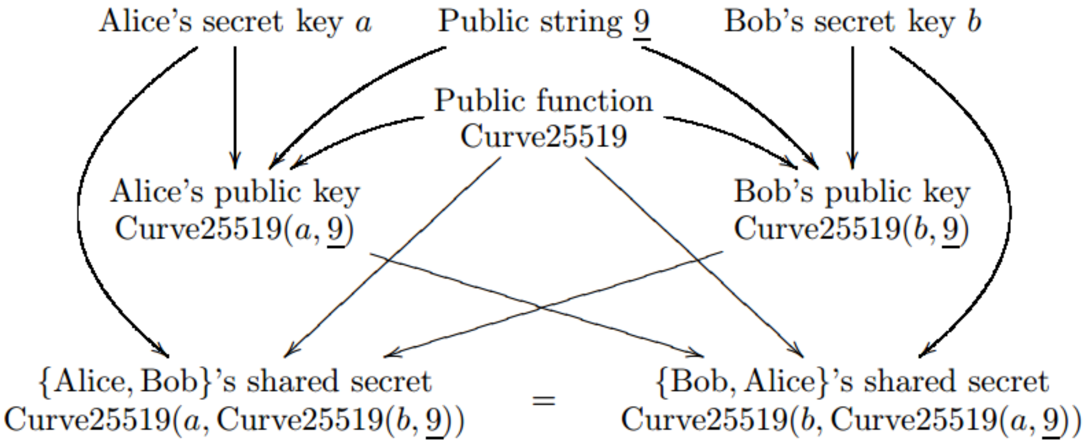

Alice's secret key a
Public string $\underline{9}$
Bob's secret key b
Public function
Curve25519
Alice's public key
Curve25519(a, $\underline{9}$)
Bob's public key
Curve25519(b, $\underline{9}$)
{Alice, Bob}'s shared secret
Curve25519(a, Curve25519(b, $\underline{9}$))
=
{Bob, Alice}'s shared secret
Curve25519(b, Curve25519(a, $\underline{9}$))

Figure 3.8 Data Flow from secret key through public keys to a shared secret (Bernstein, 2006)

Step 1 Alice creates a secret key, called "A."

Step 2 Bob creates a secret key, called "B."

The base point and public string for Curve 25519 is chosen to be 9.

Step 3 Alice's public key is generated by Curve 25519 (secret A, 9).

Step 4 Bob's public key is generated by Curve 25519 (secret B, 9).

Step 5 Alice's shared secret key is calculated by Curve 25519 (A, public B)

Step 6 Bob's shared secret key is calculated by Curve 25519 (B, A)

In the example, a hash of the shared secrets in Curve 25519 (A, Curve (B, 9)) is used as the key for a secret-key authentication system

Bernstein who is the creator of Curve25519 lists the efficiencies of Curve25519 as follows:

High Speed: For his test on Pentium III system Curve25519 computes 832457 cycles and 957904 cycles on a Pentium 4

Short secret keys: The Curve 25519 secret keys are only 32 bytes.

Short public keys: Also The Curve 25519 public keys are only 32 bytes. Normally ECDH functions use 64-byte public keys. And also free key validation and just 16kb of code size are efficient for an ECDH schema. (Bernstein, 2006)

3.1.8 SSL

The Secure Sockets Layer (SSL) is a set of regulations that NetScape made in 1996, which conducts how private files are sent through the internet. Eventually, ways of scrambling the data within the SSL server were discovered. At the same time, the URLs that were associated with SSL became more conventional, and they switched their beginnings from “https:” to “http:.” The SSL certificates now ensured more privacy in data exchanges because they encrypted the data before it was sent to the web-server.

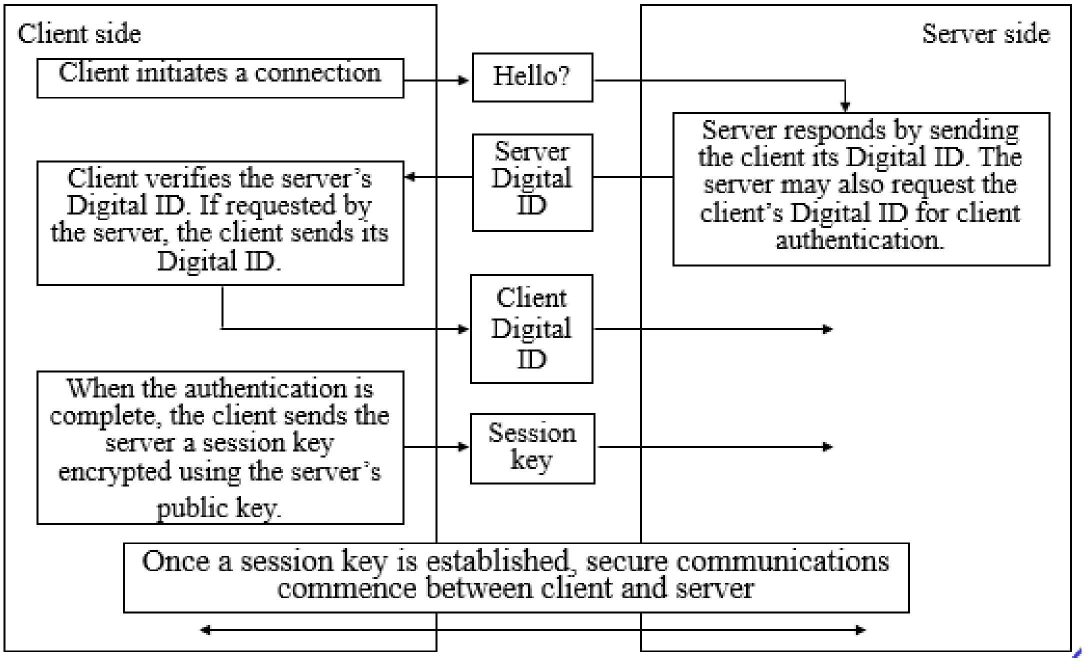

Client side
Server side
Client initiates a connection
Hello?
Server responds by sending the client its Digital ID. The server may also request the client's Digital ID for client authentication.
Server Digital ID
Client verifies the server's Digital ID. If requested by the server, the client sends its Digital ID.
Client Digital ID
When the authentication is complete, the client sends the server a session key encrypted using the server's public key.
Session key
Once a session key is established, secure communications commence between client and server

Figure 3.9 SSL Protocol Illustration (Foertsch, 2004)

3.1.9 OpenSSL

The OpenSSL not only functions as a multi-functional cryptography library, but it also commands a SSL and a Transport Layer Security (TLS) system. The SSL and TLS standards are saved in OpenSSL as an accessible library. The core database (which is written in C code) offers numerous utility features, including basic cryptographic ones. It is available to all users because it uses wrappers, which can be translated to different computer languages. (OpenSSL FIPS-140, 2015)

3.1.10 One-time Password

A one-time password is a disposable password that is generated instantaneously and then sent to users. This password is recommended to be sent from third-party services. Time synchronization is very important; without it, the password can be exposed to attacks. One-time passwords can be sent by SMS messages, local messages or web services. Such as in the case of voting (as this thesis focuses on it), the user receives a message through their mobile phone, which includes a one-time password. Then, the user is required to enter the password in the application. If the user enters the right password, then their vote will be delivered to the server.

3.2 System Environments

3.2.1 Cordova

Apache Cordova is a mobile development framework and it is open source. In using it, mobile applications can be created by utilizing brand new web technologies. Cordova is a solution for cross-platform implementation. Systems runs in wrappers which points to each platform, and Cordova endures compatible to standards API binning to reach device’s sensors, network status and hardware options. At the same time, Cordova can use both Objective-C (which is its native language), Java and Swift modules, or other modules can be written for it.

Through Cordova, applications can be built by using familiar web technologies that use native languages. Then, it will require less time and work when the program is used as a port for other mobile platforms. The applications that are accessible through Cordova are very distinct from a webpage setup, and they are built to be compatible with Apple or Google. The natural functions and sensor-based hardware of the mobile phones can be reached directly by Cordova. When they cannot be reached directly, then Objective-C and Java Swift can be mutually translated and swapped through Cordova. For instance, when the device ID of an Apple mobile phone stays in the phone, it can be retrieved. Thus, Objective-C (which is Apple's native language) should be copied and then computed in Cordova to retrieve the device's ID. (Cordova, 2015)

3.2.2 Cordova Security

The security of Cordova starts with the security of JavaScript html and css coding. Some soon-to-be explained security issues that the program has are in its uses of local storage and KeyChain, as well as in its JavaScript encryptions. When the keys are stored in variables, it can be harmful if the application can be decompiled, even if all precautions are taken. Therefore, temporary keys should be stored in local storage sandboxes, which cannot be used by other applications or devices. Permanent applications like private keys, should be stored officially in IOS's KeyChain with the native writing bridge.

3.2.3 Local Storage

The operating system typically provides an abstract layer for storing and retrieving application-specific data, like their preferences or how long they have been running. These values may be stored in the registry, INI files, XML files, or in some other place in the platform's conventions. If your native client application needs local storage beyond key/value pairs, you can embed your own database, invent your own file format, or utilize other solutions.

Historically, mobile/web applications use cookies, but cookies are limited to about 4KB of data, and they are included in every http request, so they slows down the applications. We need to create a large storage space for clients.

HTML5's storage is based on key/value pairs that users determine. If data is stored with a key that is given by the user, then the same data can be retrieved by using the same key. The key should be a string. The data can be of any type that JavaScript supports, including strings, Booleans, integers, and floats. However, the data is actually stored as a string. Local storage has 5 MBs of space.

3.2.4 KeyChain

KeyChain is an official password management system in OSX, which was developed by Apple. A “keychain” can contain various types of data, such as passwords, private keys, certificates, secure notes, credit card information, and Wi-Fi network information (etc.). The “keychain” files are stored by using Triple DES. The “keychain” are set to lock automatically. It is Apple's official key storage system. (Support Apple INC, 2015)

3.2.5 IOS Gyroscope Sensor

The iOS Gyroscope was created to sense when Apple gadgets are moved or shaken, and the devices' applications commonly use it. As Figure 3.10 shows, it detects when the x, y, and z axes shift, and matches its findings to the fixed positions.

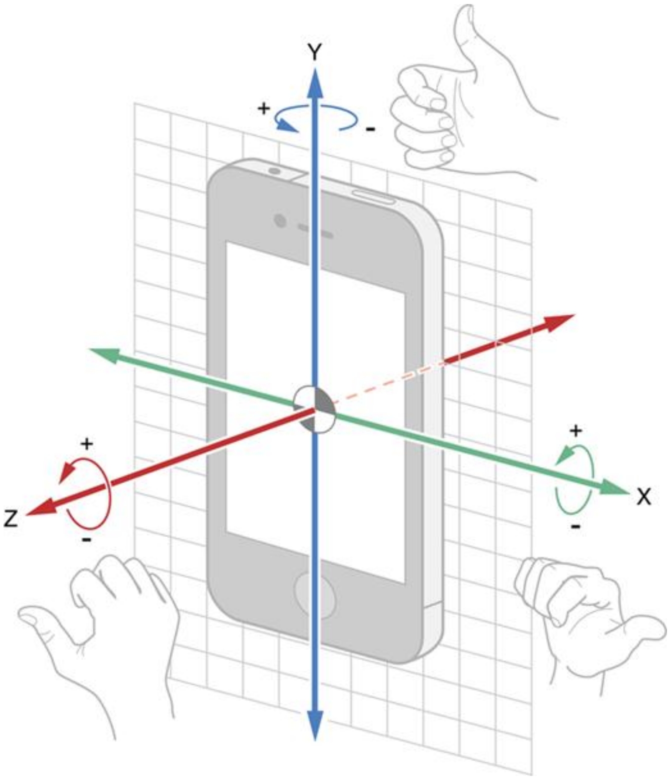
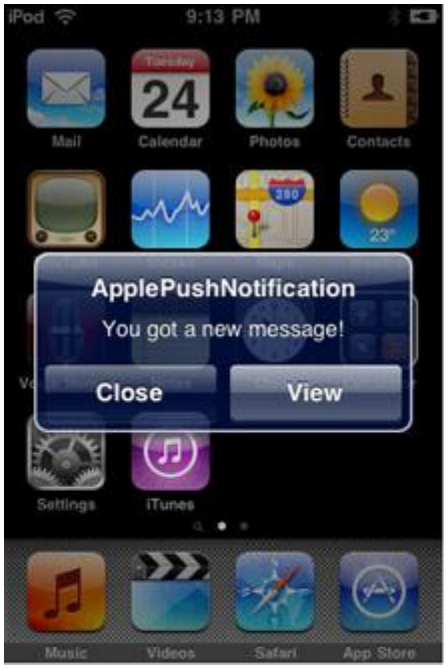


Figure 3.10 Apple Gyroscope Illustration

So, if someone snags the appliance from its owner, it will lock to prevent them from voting for them. This feature is enabled as an extra precautionary security measure for situations like those, which may involve a physical attack. The shake-detection sensitivity can also be changed so the phone will be as responsive as the user deems necessary for their circumstances.

3.2.6 Push-Notification Messages

If its users enable them to, applications can send push-notification messages when needed. Most program innovators prefer that type of mechanism because they do not cost anything to create, and they are commanded by the operator. The application that this thesis describes also uses one, and it works as parametric variable. It can fire at will, if the user allows it to. Figure 3.11 shows how notifications pop up.



ApplePushNotification

You got a new message!

Figure 3.11 The message which is sent with push notification

3.2.7 ASP.NET MVC

The Model View Controller (MVC) is one setup possibility for web applications. It lays out parts of the foundation of the program, such as a list of database records and reveals needed information which the administrator has access to. The MVC blueprint also grants full reign between CSS, JavaScript, and HTML coding.

MVC structures are divided between three parts; the model, the view, and the controller. The model it often fetches the information, ensures that is logical, and holds it. The view (which is commonly created out of the information) manages how the information is portrayed. The controller manages how users engage in the applications, watches their interactions through the view, and sends them the system.

MVC consists of three fields so that complex programs can be approached according to each function, and it makes testing easier, too. Additionally, the programs can be tackled from multiple angles at once when they are simplified in such a manner

3.2.8 Apple Touch ID

Apple Touch devices have a button with a silver, sapphire crystal ring around it, which senses that your finger is on it and notifies the Touch ID program to begin scanning your print. The mechanism is very sensitive, and it looks at the sun-epidermal layers of your skin to take multiple high-resolution images of sections of your fingerprint to piece together. After the image is constructed, finger is labelled as an arch, whorl, or loop, and unique, miniscule ridge details are identified; even directional variations from pores and edges are found. The probability of someone's fingerprint being misidentified to match another's is 1 in 50,000 identified prints, and Touch ID's limit for the number of unapproved entries is 5. (Support Apple INC, 2015)

THE SECURE ENCLAVE

A chip in the gadgets protect passkeys and fingerprint IDs by activating its secure enclave program for an extra security measure.

A numerical code of your fingerprint is stored in Touch ID instead of a photograph of it, and it is impossible to create an image based off of one of these codes. Then it is encrypted with a unique session key, and pushed through to the secure enclave on your Apple A7 to be recognized and verified. The session key was built into the sensor and security enclave. (Support Apple INC, 2015)

According to Apple, the enclave only uses your data to make sure that it matches the one that it recognizes, and it is isolated from the device and the rest of its chip. Therefore, nothing else can access or store it, and your biological pass is never backed up to anywhere else but the Touch ID program. Other fingerprint libraries cannot access it, either. The organization also guarantees that it's processor cannot read it.

Figure 3.12 Assemble of Apple Touch Sensor

Wherever data is stored, there is always the possibility that it can be accessed. Even then so, the program should not be fully trusted .Apple debates that the exchange used for fingerprint verification uses advanced encryption standards (AES) to protect the keys while they are in use by both the server and client, and that AES encodes then during their trip between the two. Apparently, the communication is just an "affirmed" or a "rejected" statement, and the key randomization enhances and optimizes protection. But is Apple's word really reliable?

3.2.9 JavaScript

JavaScript is object-oriented, cross-platform and superfast scripting language. It is a small and lightweight language.

JavaScript composes a standard library of objects, such as the array, date, and mathematic operations, as well as a core set of language elements such as those for operators, control structures, and statements. It also supports high-level operations like multi-threading, sockets, etc.

3.2.10 JQuery and AJAX

JQuery is a fast, small JavaScript library. It is for writing JavaScript faster and more fertile and garish. It makes things like HTML document traversal and manipulation, event handling, animation. With a combination of versatility and extensibility, millions of people use jQuery

AJAX updates parts of a web page without the whole page needing to be reloaded. This is because AJAX allows pages to be updated asynchronously by exchanging small amounts of data with the server behind the scenes. (jQuery Foundation, 2016)

3.2.11 JQuery Mobile

JQuery Mobile is based on the jQuery and is a HTML5 and CSS3 based mobile user interface system for designed to make responsive mobile applications. It allows AJAX page transitions, touch events and tons of widgets. It is lightweight, flexible and customizable framework.

3.2.12 JavaScript Security

The first phase of JavaScript's security is transmitting data securely with SSL used in https:// links.

The app may only use an F-sensitive code, JavaScript is not compiled – so the best we can do is obfuscate it to the point where it is as unreadable as possible. The source code should be minimized. Minimization, obfuscation and self-protection can apply to an app for JavaScript security.

Here is an example of a protected JavaScript code:

Unprotected JavaScript Code:

```
(function() {function add1(element, index, array) {    array[index]=element+1;    );
```

Protected JavaScript version of the code above:

```
/*...*/ ff)<<W);s=this[a](s,B);s=((s&0x 1ff` = A)|(s>>>Q` <)f);P^=s;P=((P&0x7`
H\"y)|(P>>>h);P=(P*z+((129.,0x187)<(91,0x1E0)?(0x72,0xe6546b64):(120.
```

The source codes of this thesis's application are secured by JScrambler. (Auditmark, 2007).

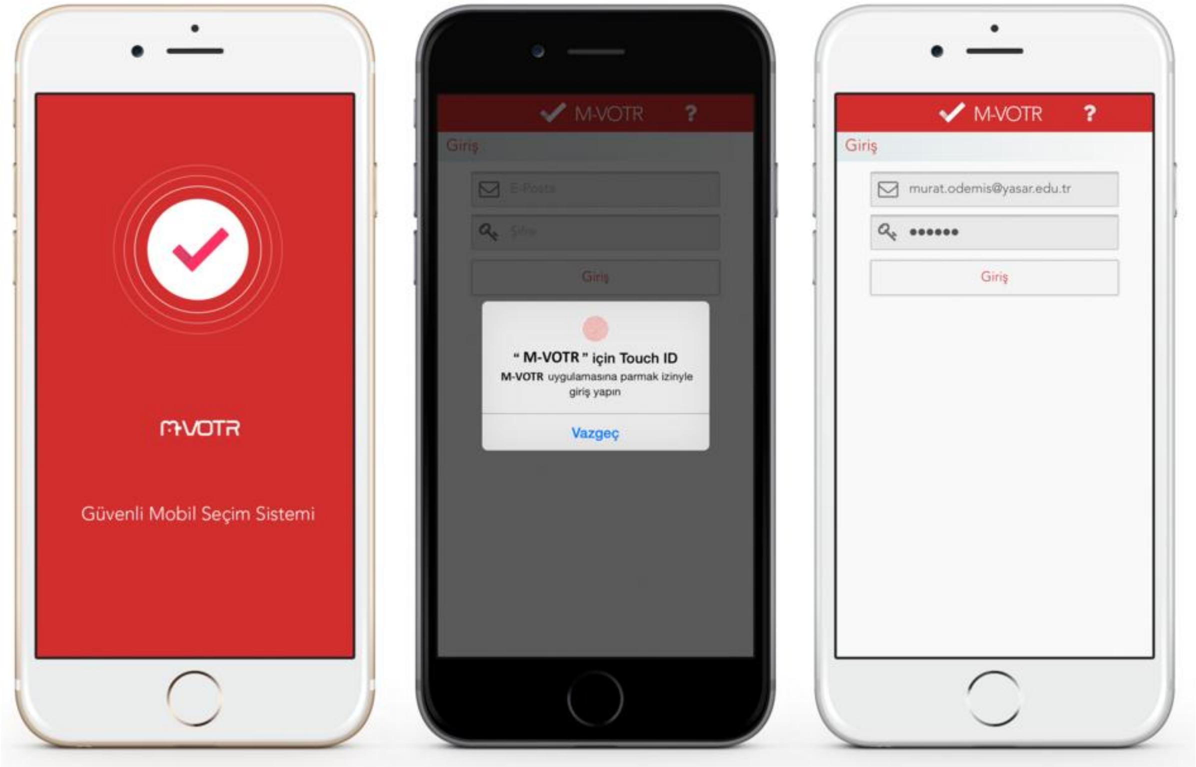
4 SYSTEM ARCHITECTURE

4.1 First Login

The home page of an application is independent from the activations within the application itself. On the home page, users enter the system using their official credential. The web-services in this part are provided by the institutions which will use this application. For instance, if this system will be used by a government, then users are going to enter the application by their e-government passports. So, they will be granted access to the application, as enabled by their government, and for the universities' elections, web-services are enabled by a school's information technology center. In the example that is illustrated in Figure 4.2, a user enters the system with the unique information that they were provided by their university.

To test the application's security, a user-management system is implemented, and the application is protected by SSL through its test domain https://mobilevoting.net. Users can be enrolled in the system and enter it with their information, and that information is solely for mobilevoting.net. In Figure 4.1, screenshots of its login page can be seen.

.



“ M-VOTR ” için Touch ID

M-VOTR uygulamasına parmak iziyle giriş yapın

murat.odemis@yasar.edu.tr

••••••

Figure 4.1 Screenshots of Login

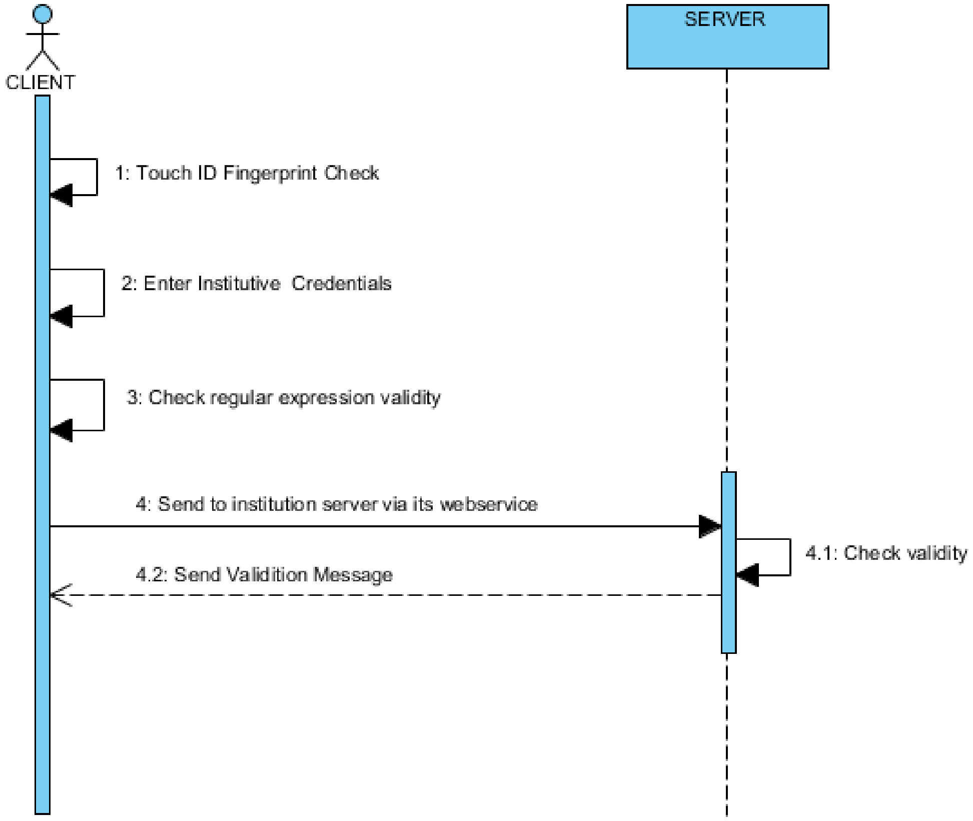

CLIENT
SERVER
1: Touch ID Fingerprint Check
2: Enter Institutive Credentials
3: Check regular expression validity
4: Send to institution server via its webservice
4.1: Check validity
4.2: Send Validition Message

Figure 4.2 Login Client-Server interaction diagram

Step 1: Fingerprint verification is performed by the application. Users show their fingers to the Apple Touch ID sensor, and then the application compares their fingerprint with the ones that it predefined. Then the application returns a Boolean message. If the application responds with a “success” message, it means the fingerprint is valid, then the user can enter their information; otherwise they are not allowed to put it in.

Step 2: Users enter their institutive credential in the appropriate parts.

Step 3: The application validates the regular expression of that information. Before sending the inputs to the server, the data is validated by the client.

Step 4: An e-mail and password are sent to the institution's main server by the web-services that the institution set up for the application. Also this server should have

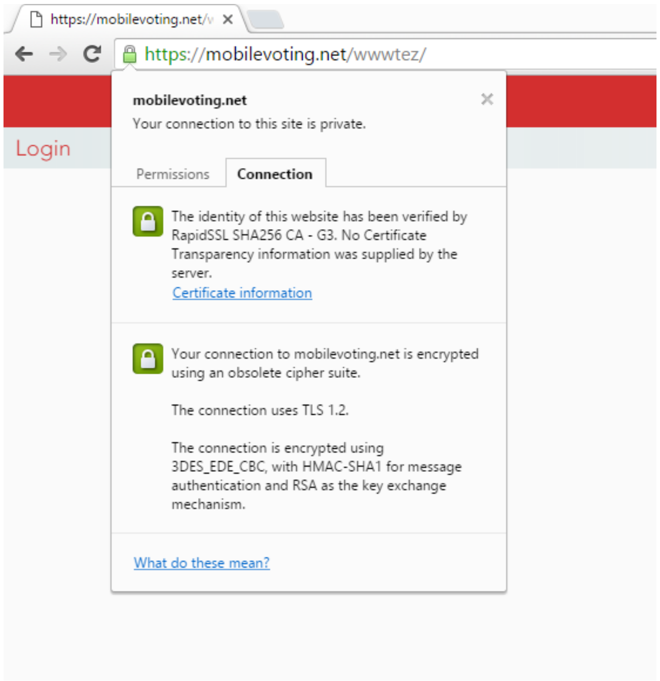
SSL too. If credentials are approved, then the user can move on to certify their identity. In this thesis, this part is demonstrated using mobilevoting.net demo web services and database. Users are registered to system on administration panel, and their warrants is arranged here by system admin

4.2 Certification

In this step, besides the vote-keeping server, there is also the Certificate Authority. The Certificate Authority is going to respond to the certificates' requirements and distribute the certificates. All the certificates will be X509 standards (Cooper et al., 2008). Technical details of the certification steps, key sizes and the examples of the algorithms have been explained in Section 5.

To secure the data in the certification process, a general SSL certificate is inserted to the domain. OpenSSL is used to create certification demands, certificates, keys, etc. Technical details of this phase can be seen on Section 5.3

On Figure 4.3 the secure connection of test domain https://www.mobilevoting.net which is protected by SSL is seen. The service uses RapidSSL.



https://mobilevoting.net/v ×
https://mobilevoting.net/wwwtez/
Login

mobilevoting.net

Your connection to this site is private.

The identity of this website has been verified by RapidSSL SHA256 CA - G3. No Certificate Transparency information was supplied by the server.

Certificate information

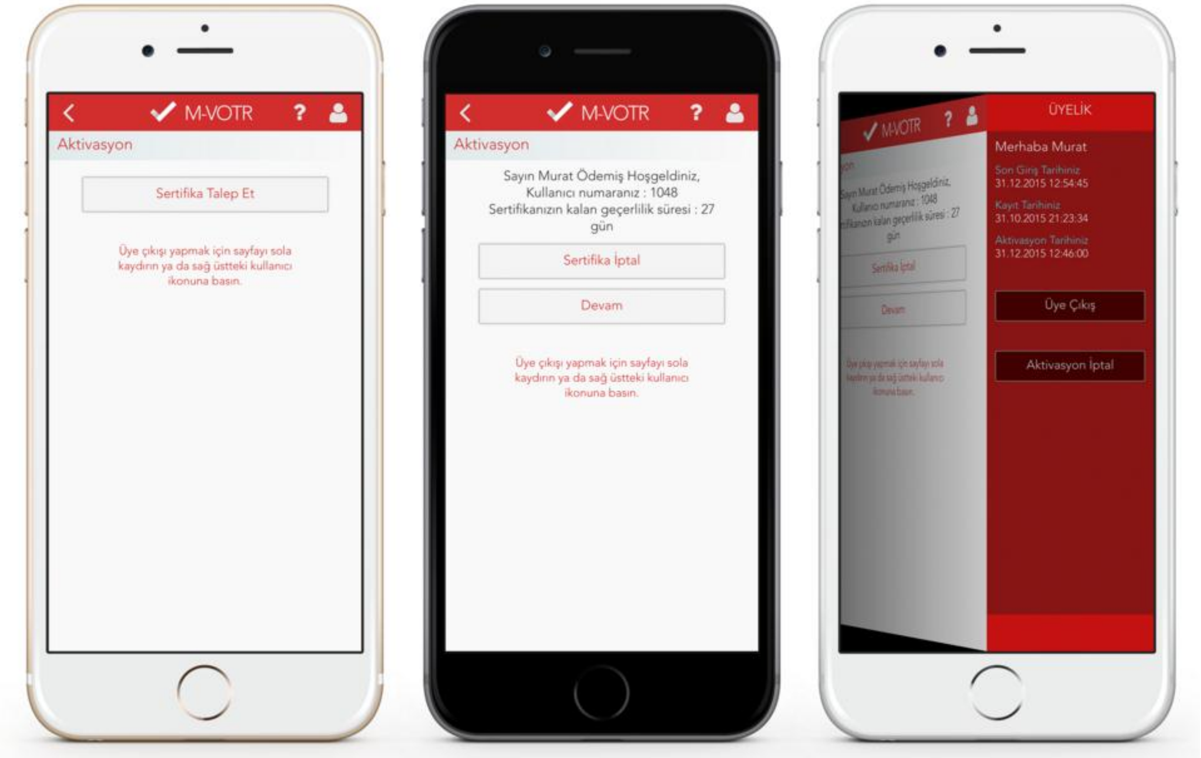
Your connection to mobilevoting.net is encrypted using an obsolete cipher suite.

The connection uses TLS 1.2.

The connection is encrypted using 3DES_EDE_CBC, with HMAC-SHA1 for message authentication and RSA as the key exchange mechanism.

What do these mean?

Figure 4.3 SSL Protected Test Domain and Server



Üye çıkışı yapmak için sayfayı sola kaydırın ya da sağ üstteki kullanıcı ikonuna basın.

Sayın Murat Ödemiş Hoşgeldiniz,
Kullanıcı numaranız : 1048
Sertifikanızın kalan geçerlilik süresi : 27 gün

Üye çıkışı yapmak için sayfayı sola kaydırın ya da sağ üstteki kullanıcı ikonuna basın.

Merhaba Murat

Son Giriş Tarihiniz
31.12.2015 12:54:45

Kayıt Tarihiniz
31.10.2015 21:23:34

Aktivasyon Tarihiniz
31.12.2015 12:46:00

Figure 4.4 Screenshots of Certification

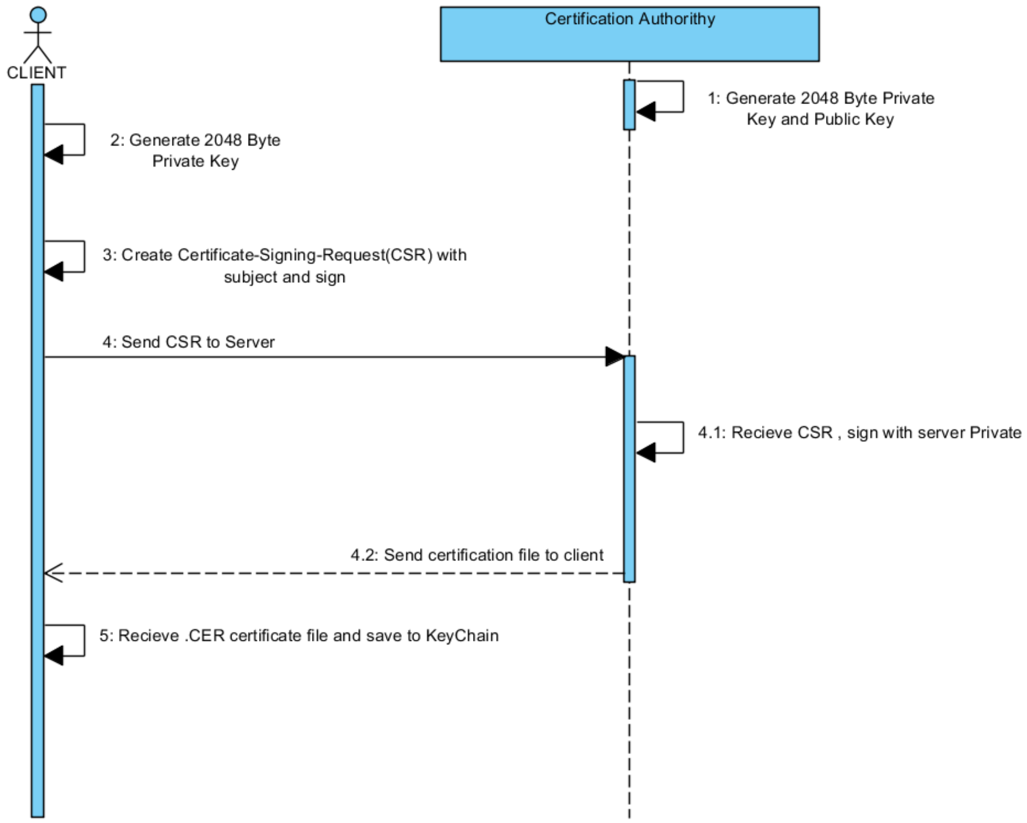

CLIENT
Certification Authorithy
1: Generate 2048 Byte Private Key and Public Key
2: Generate 2048 Byte Private Key
3: Create Certificate-Signing-Request(CSR) with subject and sign
4: Send CSR to Server
4.1: Recieve CSR , sign with server Private
4.2: Send certification file to client
5: Recieve .CER certificate file and save to KeyChain

Figure 4.5 Certification Client-Server Diagram

In this thesis, Curve25519 ECDH was used for some issues such as key exchange etc. In a session while voting, 3 or more key exchange processes can be performed so it is appropriate to use Curve25519. However, at this step, RSA was used for OpenSSL certification process. The reason is widespread usage of RSA and OpenSSL together and reinforcement of used plugins for RSA. The data transmission in this diagram and the security of the transmission between server and client are provided by SSL. OpenSSL communication is encrypted via RSA. The steps which is shown in Figure 4.5 is explained step by step and the Figure 4.4 illustrates the user interface of this phase.

Step 1: After the user logs in, the “Request Certificate” button appears for them to touch. The Certificate Authority creates a 2048-byte private key.

Step 2: Client side; A 2048-byte private key is created for the user and saved to KeyChain.

Step 3: The user completes a certificate-signing request (CSR) by filling in the missing information in the organization's form. The subject of it is filled as certificate preparation that is compatible with X509.

Step 4: The subject contains the organization's name, unit, country code, state or province, locality, and common name. This information completes the public key-signing request with this subject, and it is sent to the Certificate Authority as CSR.

Step 4.1, 4.2: The received CSR is signed with its private key, the expiration date is determined, and the certificate file is sent to the user.

The certificate's information is recorded to the database.

Step 5: The user receives the certificate file, and it is saved in KeyChain.

After that final step, the user has their certificate. If they want to revoke the certificate, then they can send a request to the Certificate Authority by touching the “Revoke Certificate” button.

Certificate Validation

To validate the certification, OpenSSL verification functions and RSA is used. The user uses their keys to generate a validation message. The server compares the received validation message to the user's certificate. If they match, then the application will check the certificate's expiration date. After that, the user is directed to the activation registration step. Certification Authority verification is provided by the paid service of RapidSSL.

4.3 Activation Register

If a user has entered the correct credential into the system, has a valid certificate, and has not activated the system before, then they can be shown directly to the registration screen. To ensure that the application is secure, account activations will not be granted until after the users have entered their valid default corporate information.

In this step, it is also required to create fake activation information. Thus, a user can enter the fake information into the application if they are under pressure. The system will not warn the user about fake information, so all processes will continue regularly, but their vote will be invalid. In addition to a character-based password, the application also requires a geometric pattern password from the user. As seen second screen on Figure 4.6, the application also supports pre-determined fake pattern passwords. The registration screen for activation saves all passwords to the database. In order to do so, all of the data is sent to the server and retrieved from the server in an encrypted form.

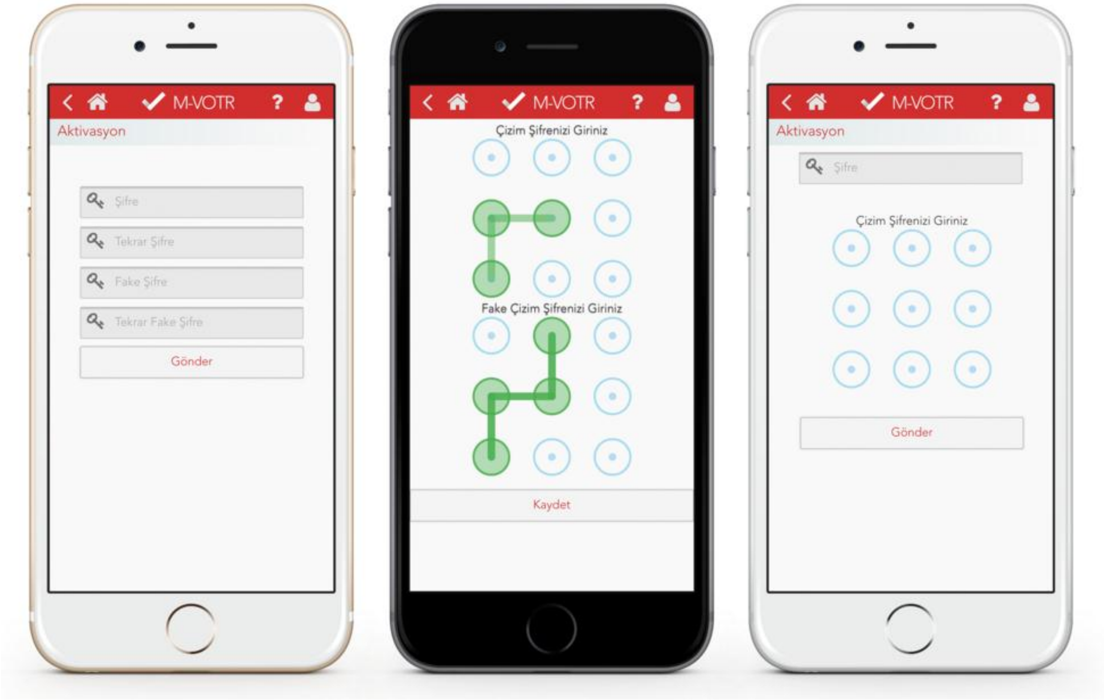


Figure 4.6 Screenshots of First Activation Phase

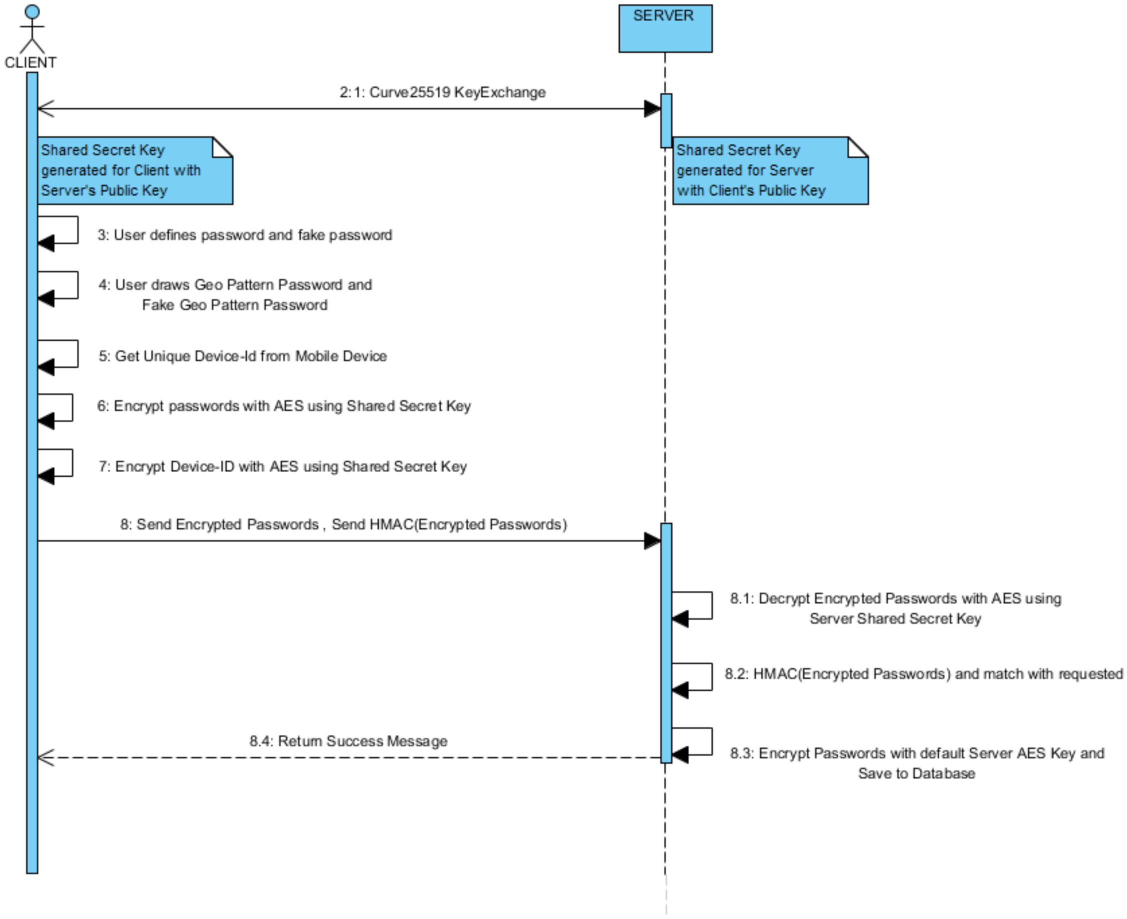

CLIENT
SERVER
2:1: Curve25519 KeyExchange
Shared Secret Key generated for Client with Server's Public Key
Shared Secret Key generated for Server with Client's Public Key
3: User defines password and fake password
4: User draws Geo Pattern Password and Fake Geo Pattern Password
5: Get Unique Device-Id from Mobile Device
6: Encrypt passwords with AES using Shared Secret Key
7: Encrypt Device-ID with AES using Shared Secret Key
8: Send Encrypted Passwords , Send HMAC(Encrypted Passwords)
8.1: Decrypt Encrypted Passwords with AES using Server Shared Secret Key
8.2: HMAC(Encrypted Passwords) and match with requested
8.4: Retum Success Message
8.3: Encrypt Passwords with default Server AES Key and Save to Database

Figure 4.7 Activation phase Client-Server interaction

The system sequence diagram in Figure 4.7 is explained step by step in the following.

Step 1-2: Curve25519 key exchange function, which is one of the critical part of this thesis, is called. Then keys are generated on both the client and server. Details about the function of the Curve25519 key exchange was given in Section 3.1.7.

Step 3: The user determines their real and fake character-based passwords in the first screen shown in Figure 4.6. Both must be at least 6 characters long and include one uppercase letter and one symbol. Then, with a single touch on the continue button, the user passes through to the geometric pattern password registration screen.

Step 4: The user draws their geometric pattern password and a fake geometric pattern password. The length of each must include at least 4 points.

Step 5-7: The unique device ID and user ID are fetched from the user's local storage, and encrypted by AES 256.

Step 6: All passwords, including the fake and geometric pattern ones, are encrypted by AES. In order to generate AES session key, Curve25519's shared secret is used as a passphrase, and it was generated after a key exchange. After the data is encrypted with AES, it is converted to the Base64 format to transfer easily. Details of AES's encryption libraries and the key sizes were explained in Section 3.1.1.

Step 8: The HMAC of the real character-based password, the fake one, the geometric pattern password, the fake geometric pattern password, and the device and user Ids are computed with using session key and SHA. HMAC technical details can be seen on Section 5.4.

The encrypted data posts to the server as a query string with AJAX, and its computed HMAC forms are sent on the request header.

Step 8.1: Encrypted passwords that come to the server are obtained from AJAX requests in the form of query strings. Encrypted passwords that are calculated with HMAC are given to the server by the https request headers.

Step 8.2: The encrypted data is recalculated with HMAC within the server. The server compares the request headers’ data with the checks, and if they are the same, it continues its process.

Step 8.3: AES encrypted data is decrypted by server session key which is derived from Curve25519 shared secret. Aforementioned, the Curve25519 shared secret was generated during the key exchange operation. After that step, all passwords and the device and user IDs are obtained as plain text.

Step 8.4: The server verifies if the decrypted device ID is the same as the registered device ID in the database.

If they are same, then the passwords and the activation date are saved to the database. Then, the server will send a response message about this process to the Ajax request system.

Step 9: If the returned message is “success,” then the user has completed their activation, and then they are directed to the activation login screen.

4.4 Activation Login

If the user has registered their activation successfully, or if they already have an activated account, then they will be directed to the activation login page. On this page, users log into the system with their predetermined character and geometric pattern-based passwords. If the users enter the system with either or both fake passwords, they can continue their processes, but their votes will be invalid.

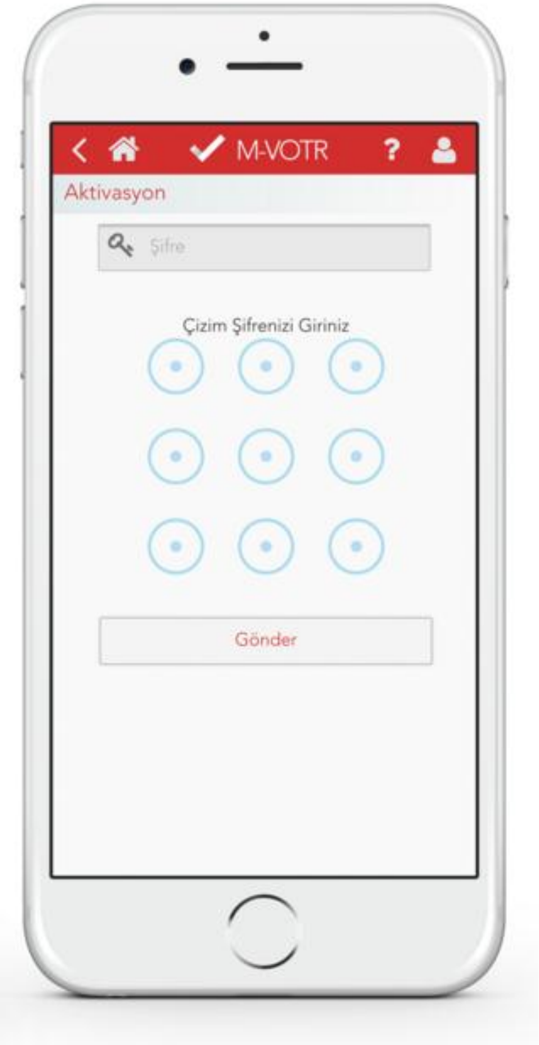


Figure 4.8 Screenshots of Activation Login Phase

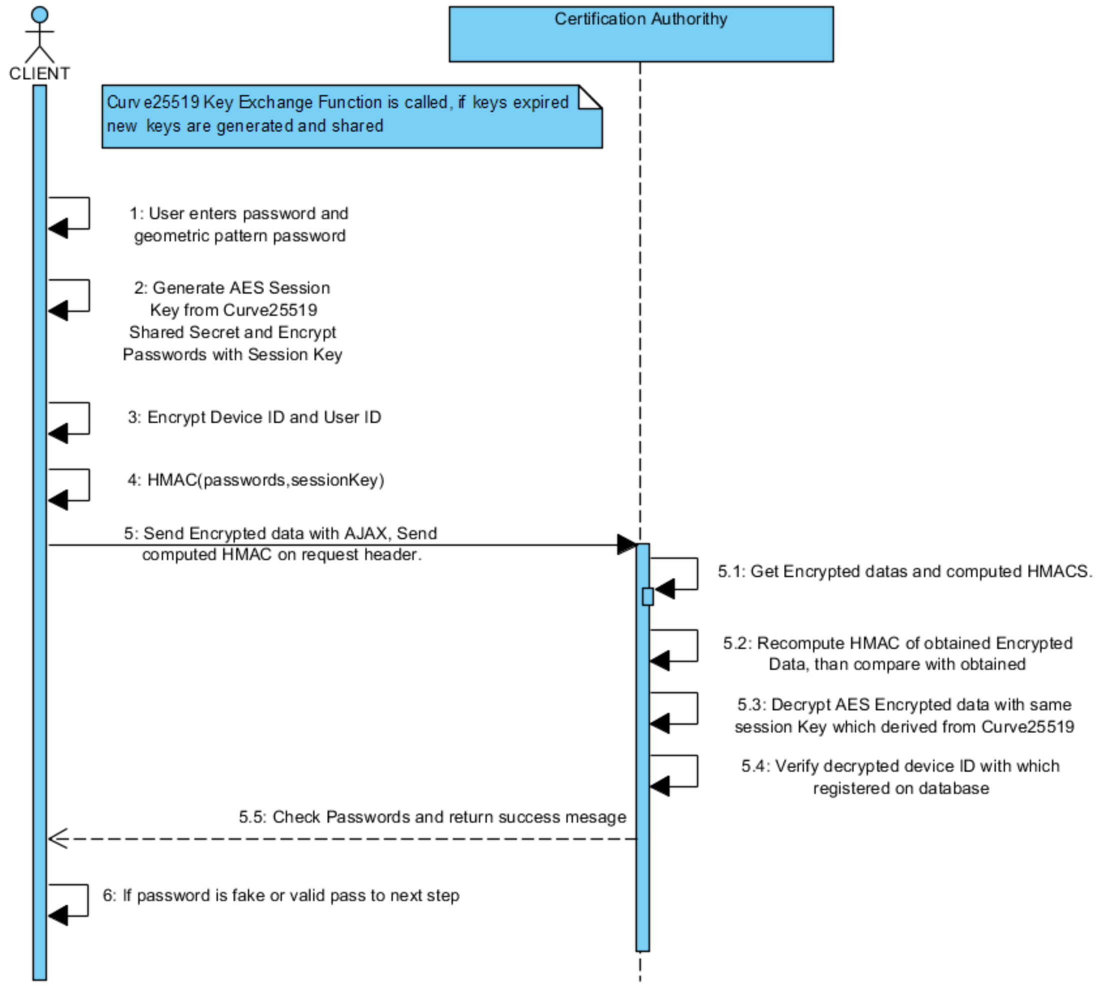

CLIENT
Certification Authorithy
Curv e25519 Key Exchange Function is called, if keys expired new keys are generated and shared
1: User enters password and geometric pattern password
2: Generate AES Session Key from Curve25519 Shared Secret and Encrypt Passwords with Session Key
3: Encrypt Device ID and User ID
4: HMAC(passwords,sessionKey)
5: Send Encrypted data with AJAX, Send computed HMAC on request header.
5.1: Get Encrypted datas and computed HMACS.
5.2: Recompute HMAC of obtained Encrypted Data, than compare with obtained
5.3: Decrypt AES Encrypted data with same session Key which derived from Curve25519
5.4: Verify decrypted device ID with which registered on database
5.5: Check Passwords and return success mesage
6: If password is fake or valid pass to next step

Figure 4.9 Activation Login Client-Server Diagram

Step 0: The Curve 25519 key exchange function is called, and if the keys has expired, then new keys are generated for encryption.

Step 1: The user enters the password to the required area, and then draws his geometric pattern password to the drawing password area.

Step 2: The character-based password and the geometric pattern password are taken from inputs and encrypted by AES with session key by using a shared secret from Curve25519 to send server.

Step 3: A unique device and user ID are obtained from the user's mobile phone and encrypted by AES.

Step 4: The HMAC of the character-based password, geometric pattern password, device ID and user ID are computed with using session key and SHA1.

Step 5: The encrypted data posts to the server as a query string with AJAX, and its computed HMAC form is sent on the request header.

Step 5.1: Encrypted passwords filter into the server via AJAX requests in the form of query strings. Encrypted passwords that are calculated with HMAC are given to the server by the https request headers.

Step 5.2: The encrypted data is recalculated with HMAC within the server. The server compares the request headers' data with the checks, and if they are the same, it continues its process.

Step 5.3: AES encrypted data is decrypted by server session key which is derived from Curve25519 shared secret. After that step, all passwords and the device and user IDs are obtained as plain text.

Step 5.4: The server verifies if the decrypted device ID is the same as the registered device ID in the database.

Step 5.5: If they are identical, then the server will compare the two inputted passwords with the passwords that have been registered in the database. After that, the server will respond with a request-status message. If both passwords are authentic, then it will display a "success" message. If one is the fake one, then the user will still receive a "success" message (which is fake). If both passwords are wrong or not fake, then it will respond with an "unsuccessful" message.

Step 6: If the returned message says that the process was successful, or fakely says it was successful, then the user is directed to the screen in which they can see possible options to vote for. If the returned message read "unsuccessful," then the user is required to enter the password again. If the user enters the wrong password several times, then the activation and the certificate are canceled.

4.5 Voting

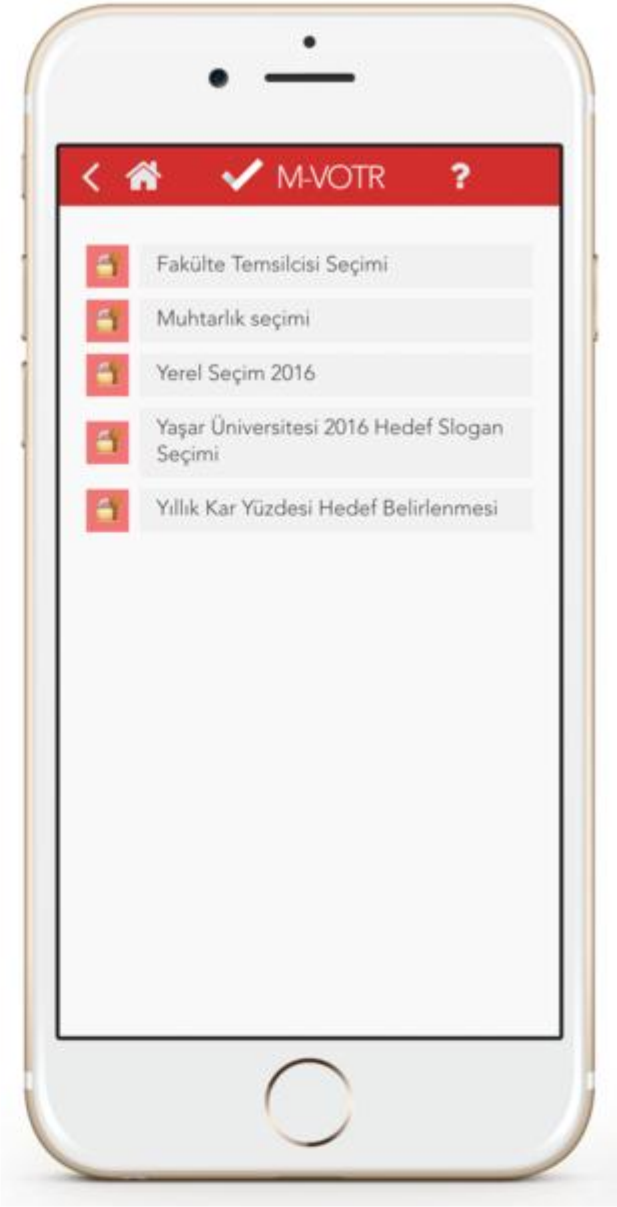
Users who were successfully certificated and activated pass through the login and activation stage and finally can see the screens of the elections.

All the elections that users are authorized to are listed on that screen. When a user clicks on the election, he can see the detailed questions and answer choices (candidates etc.), and start voting. A single election can contain more than one selection. The logic of the survey performs behind the actual election process. In one survey, there can be a single question or multiple questions, depending on the bases of the election.

For example, both elections for a representative of a faculty or of a department can be included in one election called “representative elections.” Thus, when a user partakes in it, they will see two questions in a single election. These two election votes can also be given in separate voting sessions.

All data, including election questions and choices, will be transferred in an encrypted way. Also, before sending the vote, a one-time password verification is required for voters using SMS authentication. User credentials won’t be saved with votes to supply the anonymity.

4.5.1 The List of Elections



Fakülte Temsilcisi Seçimi

Muhtarlık seçimi

Yerel Seçim 2016

Yaşar Üniversitesi 2016 Hedef Slogan Seçimi

Yıllık Kar Yüzdesi Hedef Belirlenmesi

Figure 4.10 Screenshots of Elections List

Figure 4.10 shows the active elections for the authorized users who are logged into the system. The system checks the related permissions for the users in the database before displaying this screen. This feature works with respect to real time. This means that if this screen is open in the user's mobile phone while a new election is added to this list, the user doesn't need to log off and log in again or refresh the page. The recently added election is automatically displayed on the already opened page, with the help of the AJAX technology.

Step 1: A user applies their fingerprint so they can see the listed elections.

Step 2: A Curve 25519 key exchange is requested, and if the key expired, then new keys are generated.

Step 3: When the user touches on an election, the ID of the touched election, the device ID and the user ID are retrieved from local storage and encrypted by AES with using session key which derived from Curve25519.

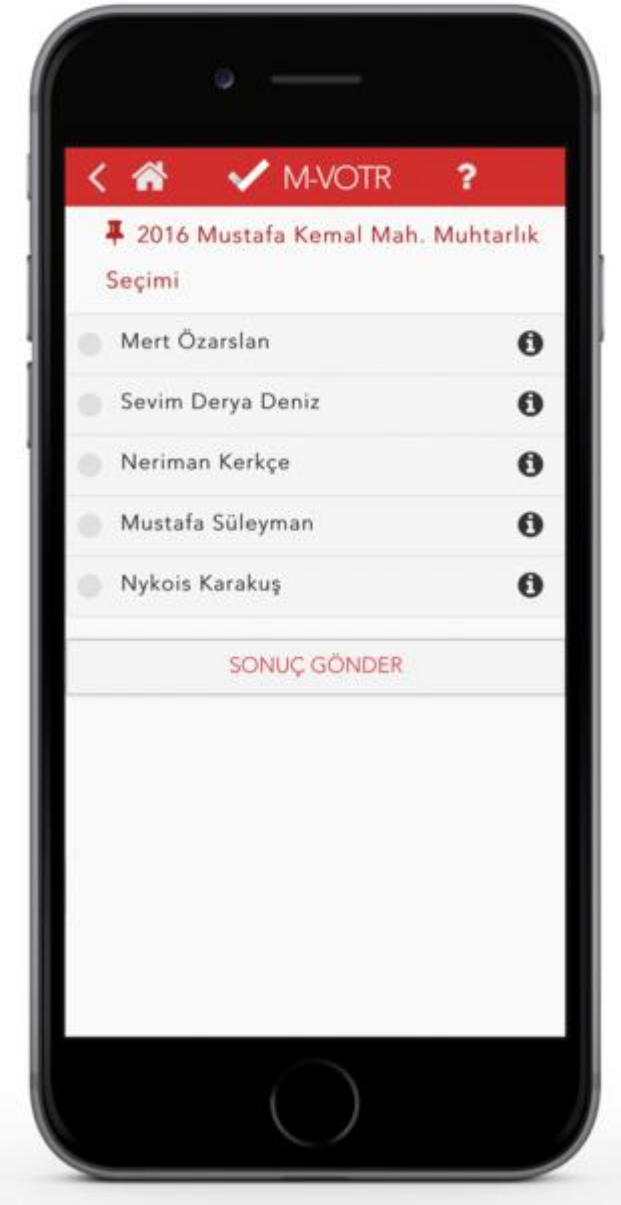
Step 4: The HMAC of the three IDs are computed.

Step 5: Ajax posts the encrypted data to the server as a query string, and its computed HMAC forms are sent to the request header.

Step 6: The server verifies the computed HMACs and device IDs; if verification is successful, it will decrypt the obtained data using the AES session key which is derived from Curve25519 shared secret.

Step 7: The server accesses the choices of the required elections from the database. Then, server encrypts their choices made in AES by using the Curve 25519 shared secret key, and it sends them to the application as a response. The computed HMACs are then sent with response header. User credentials or ID wont sent with votes.

4.5.2 Election Choices



2016 Mustafa Kemal Mah. Muhtarlık Seçimi

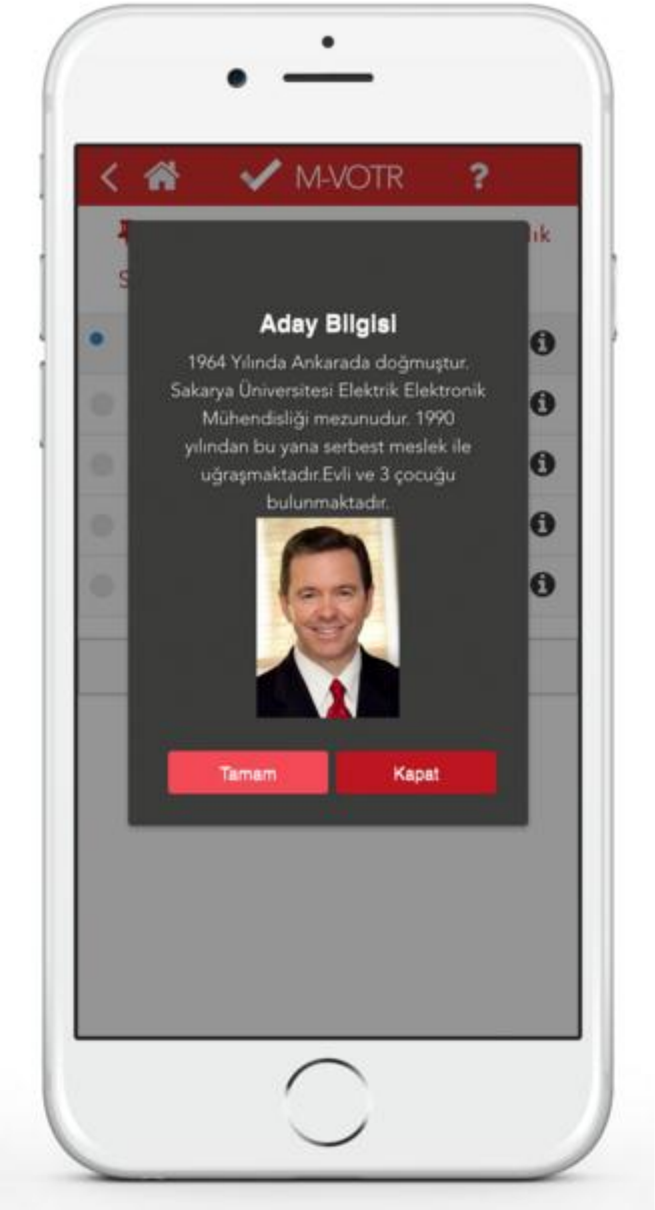
- ○ Mert Özarslan
- ○ Sevim Derya Deniz
- ○ Neriman Kerkçe
- ○ Mustafa Süleyman
- ○ Nykois Karakuş

Figure 4.11 Screenshots of Election Choices

Step 1: The HMAC codes are verified.

Step 2: Data that was encrypted by AES is now decrypted with same AES key.

Step 3: The user can display the choices of the ballot (if there are any surveys to answer) and the user can display information about each choice (candidate etc.) by clicking on the info button near it.

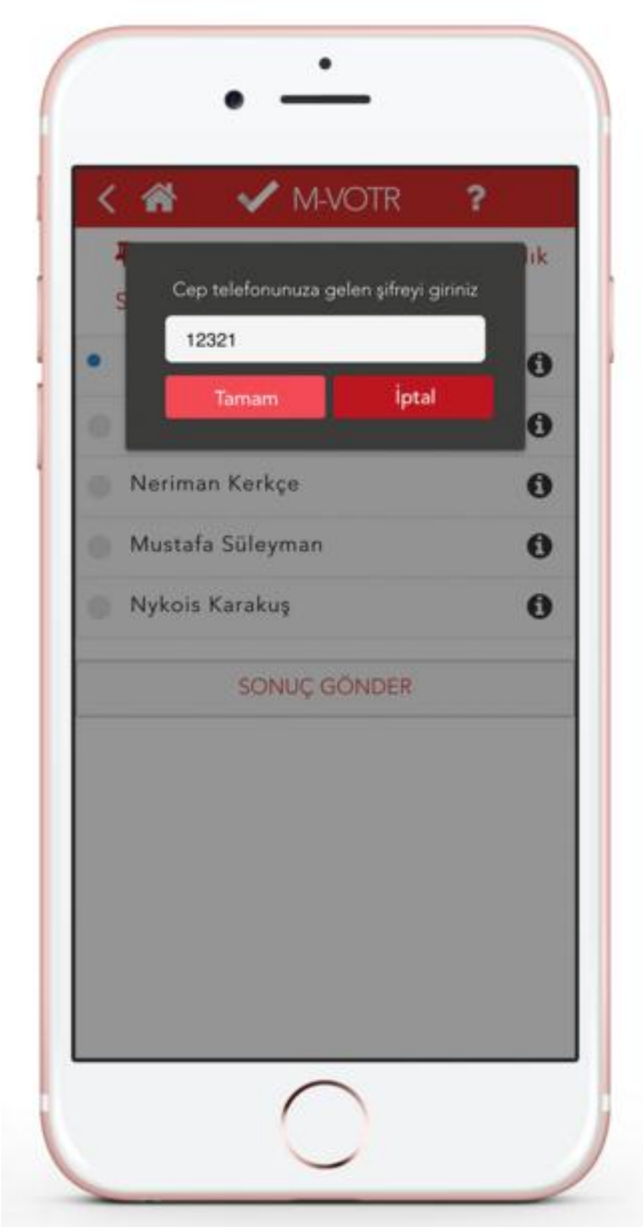


Aday Bilgisi

1964 Yılında Ankarada doğmuştur. Sakarya Üniversitesi Elektrik Elektronik Mühendisliği mezunudur. 1990 yılından bu yana serbest meslek ile uğraşmaktadır.Evli ve 3 çocuğu bulunmaktadır.

Figure 4.12 Screenshot of Candidate Information popup

4.5.3 Sending Vote



Cep telefonunuza gelen şifreyi giriniz

12321

Figure 4.13 Screenshot of OTP phase

Step 1: Users choose their option and click on the "VOTE" button to cast their votes.

Step 2: A One Time Password (OTP) process is enabled, then, by looking at the user's certificate, a one-time SMS password is sent to their mobile phone. The user then enters this SMS OTP verification code into the application, and if it is right, then they start voting.

Step 3: Curve 25519 is requested for a key exchange, and if the password expired, new keys are generated.

Step 4: The IDs of the chosen option and device ID retrieved from the user's local storage and encrypted by AES.

Step 5: IDs are converted to HMAC with using session key.

Step 6: The encrypted data is posted to the server, and computed HMAC are sent to the request header.

Step 7: The server verifies the HMAC and device ID; if they are verified successfully, it decrypts the obtained data through the Curve 25519 shared secret key. The server saves the ID of the marked option.

4.6 Extra Features of Mobile Application

If a mobile phone is shaken severely or someone tries to grab it from the user, the Apple Gyroscope sensor will detect that attack, and the application will be locked. It can be opened again when the user scans their fingerprint. This sensor will be enabled if the mobile phone is grabbed while the user is casting a vote. Also the location of user can be fetched from GPS service of mobile phone. With using google map plugin the users' current longitude and latitude can be detected. The application have this feature but it is not used because of privacy.

In all application pages, the user can use the "Question?" button which is in the top-right corner of the page. If a user touches that button, then all the information of that page will pop up along with directions.

If the user touches the user icon (which is also in the top-right corner of the page) or pulls the page from right to left, then a sliding menu appears. In this page, the user can display their membership information, can cancel activation from their account or logout.

Some features of the application are facial recognition, facial detection, pictorial recognition and image filtering. Facial detection algorithms run efficiently, but they do not enrich the application in term of security. Also, 3rd party facial recognition applications for mobile devices are pricey and raise suspicions that the users' images are saved. For this application, it was considered that its camera-enabled features would expose its users to such a risk while voting. However, if security is a primary concern, those functions wouldn't be desired, so it was decided that they would not be a part of this application. Some algorithms and frameworks were tested for these features, and they are listed in Appendix A.

4.7 The Administration Panel

In a general view, the plans of a mobile voting application and its security are explained in this thesis. However, in its inner workings, an extended administration panel was designed. In it, users can be managed within the system, departments and institutions can be added to the system, several elections can be activated and organized through it, an authorization process can take place, and the results of the elections can be announced instantaneously. Thus, the administration panel was designed to be very detailed. Its screenshots have been added to the APPENDIX B.

The admin panel is the part of the system where the system is controlled, elections are added, and authorizations are done. It is coded in ASP.NET MVC5. All of the processes executed in the admin panel, as well as their features, are summarized in the following section.

In the user-management part, users can be added, organized and controlled.

In the institution-management part, parent groups are determined. For example, a parent group consisting of the Yaşar University engineering faculty can be added in this part, and a system administrator from that specific faculty will be decided.

In the departmental management part, the parent groups' departments are organized and the administrator is assigned for each. An example of such a directory is under these headings: Yaşar University, Engineering Faculty, Computer Engineering (department), and its administrator can be listed in the last part.

In the election-management part, elections can be formed and authorizations can be done. The authorization step determines which users can take part in the elections. Also, election results can be displayed in it.

A single election can contain more than one question, and a question can contain multiple options. Also, there can be an infinite number of options in a single election.

The upcoming example shows the different combinations of users who may be authorized for a university election:

- Only the engineering faculty members can see the election.

- Only the members in the computer engineering department in the engineering faculty can see the elections.

- Just members of the computer engineering and financial affairs departments can see the elections.

- All the members of the health, culture, and sports department, the management dept. of administrative affairs, and all members of the engineering faculty

These kinds of combinations can be formed in the application.

The results of the elections can be displayed promptly. In order to display the results, it is not required to refresh the page. Ajax technology is used to display the results through the mobile application without refreshing. When all users have finished voting, then the results can be seen immediately.

5 DEVELOPMENT METHODS AND LIBRARIES

List of whole libraries and plugins and their author and creators, which are used in this thesis can be seen on APPENDIX A.

5.1 Curve25519 Key Exchange

On both the server and the client' side, plug-ins are translated to Javascript and C# .NET in the library of NaCL (Daniel J. B et al., 2012). A key exchange diagram can be seen in Figure 5.1

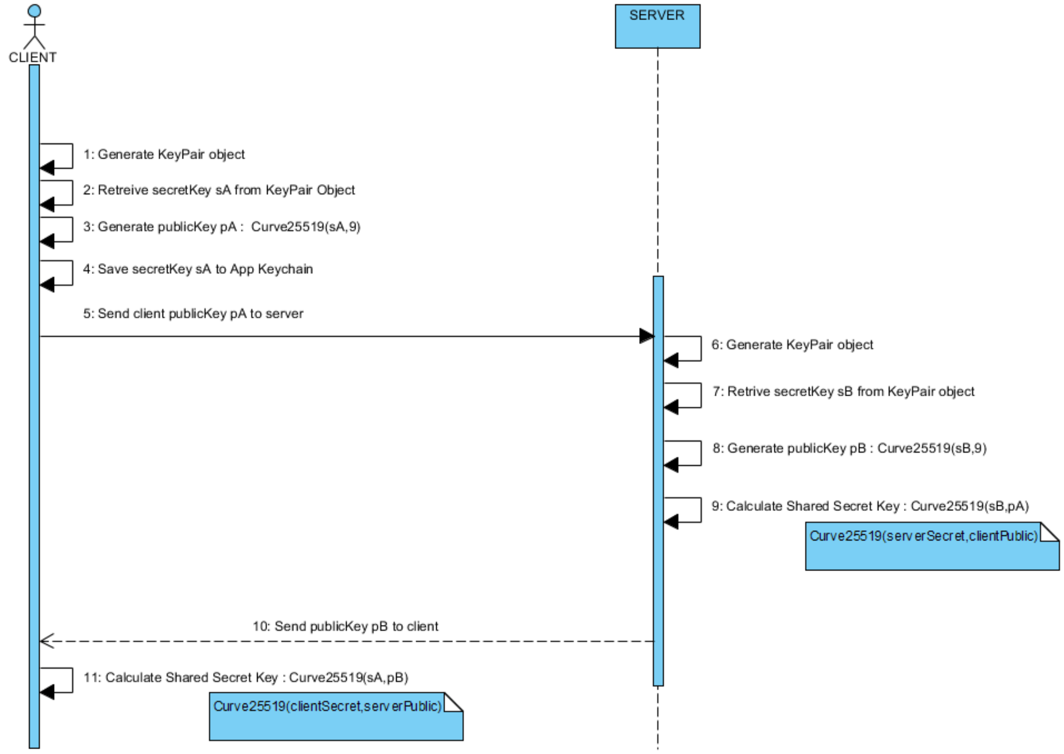

CLIENT
SERVER
1: Generate KeyPair object
2: Retreive secretKey sA from KeyPair Object
3: Generate publicKey pA : Curve25519(sA,9)
4: Save secretKey sA to App Keychain
5: Send client publicKey pA to server
6: Generate KeyPair object
7: Retrive secretKey sB from KeyPair object
8: Generate publicKey pB : Curve25519(sB,9)
9: Calculate Shared Secret Key : Curve25519(sB,pA)
Curve25519(serverSecret,clientPublic)
10: Send publicKey pB to client
11: Calculate Shared Secret Key : Curve25519(sA,pB)
Curve25519(clientSecret,serverPublic)

Figure 5.1 Using Curve25519 Key Exchange on development

5.1.1 Client Side

On the client's side, NaCL-JS and ECMA-NaCL plug-ins (which are coded in JavaScript) are used. A new KeyPair object is generated, and private and public 32-byte keys can retrieve it by using Curve 25519's function. The client's public key is sent to the server, and it retrieves it. A shared secret key is generated by combining the client's private key and the server's public key. Keys which are generated as arrays of bytes are shown.

Generated Client Secret Key Bytes:

139,188,188,90,103,39,91,39,127,89,128,18,141,188,131,96,4,54,189,61,85,164,172,22,180,204,138,139,10,214,245,204

Generated Client Public Key Bytes:

209,202,28,168,242,205,232,0,109,159,166,191,25,138,227,169,198,176,242,62,149,36,26,6,142,147,242,202,125,133,138,98

Retrieved Server Public Key Bytes:

117,147,62,100,145,91,78,112,85,124,23,48,156,218,54,115,119,62,25,106,191,210,196,249,120,153,145,120,124,174,228,32

Client Curve25519 Calculated Shared Secret Key:

Curve25519 (Generated Client Secret Key Bytes, Retrieved Server Public Key Bytes)

254,176,151,115,242,200,148,213,115,215,9,53,185,9,221,24,122,108,165,162,171,127,238,16,167,67,151,30,215,166,172,224

5.1.2 Server Side

Libraries of Chaos.Nacl (Christian Winnerlein) and Curve25519cs(Hans Wolf) are used on the server side.

Generated Server Secret Key Bytes:

8,86,138,33,121,34,161,37,39,134,25,10,6,7,235,249,166,254,59,42,33,150,205,151,80,135,106,211,55,223,96,114

Generated Server Public Key Bytes:

117,147,62,100,145,91,78,112,85,124,23,48,156,218,54,115,119,62,25,106,191,210,196,249,120,153,145,120,124,174,228,32

Retrieved Client Public Key Bytes:

209,202,28,168,242,205,232,0,109,159,166,191,25,138,227,169,198,176,242,62,149,36,26,6,142,147,242,202,125,133,138,98

Server Curve25519 Calculated Shared Secret Key:

Curve25519 (Generated Server Secret Key Bytes, Retrieved Client Public Key Bytes)

254,176,151,115,242,200,148,213,115,215,9,53,185,9,221,24,122,108,165,162,171,127,238,16,167,67,151,30,215,166,172,224

After a key exchange process, the keys clearly become identical on both the server's side and the client's side.

254,176,151,115,242,200,148,213,115,215,9,53,185,9,221,24,122,108,165,162,171,127,238,16,167,67,151,30,215,166,172,224

It is shown that shared secret is generated. This key is converted to base64 format to be used as passphrase for AES256.

5.2 AES Encryption/Decryption

The Object Oriented JavaScript Class Library is used by the C.NET Style (Ecoys, 2011) on the client's side. All methods and libraries in C.NET are translated to JavaScript, then inserted into this library. The same ported library is used for integrity of the system.

A 256-bit AES session key is generated by using a Curve 25519 shared secret key. This process's directory is: System.Security.Cryptography.Rfc2898DeriveBytes. Data is encrypted and decrypted by using this secret.

5.3 CERTIFICATION

The Certification Authority's operations are coded in the OpenSSL library as PHP.

To use PHP's OpenSSL support, PHP with OpenSSL[=DIR] must compiled. Then the php_openssl.dll must be uploaded to server and config changes must made in php.ini file.

Compiling OpenSSL for Apple OSX :

```
./Configure darwin64-x86_64-cc shared enable-ec_nistp_64_gcc_128 no-ssl2 no-ssl3 no-comp --openssldir=/usr/local/ssl/macos-x86_64

make depend

sudo make install
```

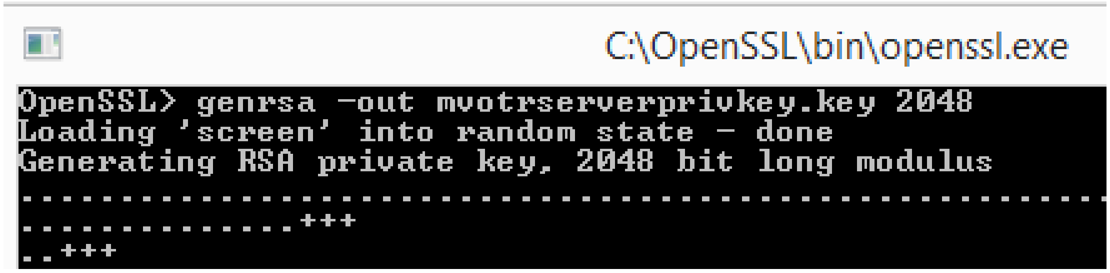
The JavaScript JSRSA library includes translations of OpenSSL-written functions, and it is used by the client.

An example of OpenSSL's certificate-generation process is shown as follows.

These are the OpenSSL default commands. JSRSA and PHP OpenSSL equivalent functions are used in this thesis instead of these commands.

Generating a private key

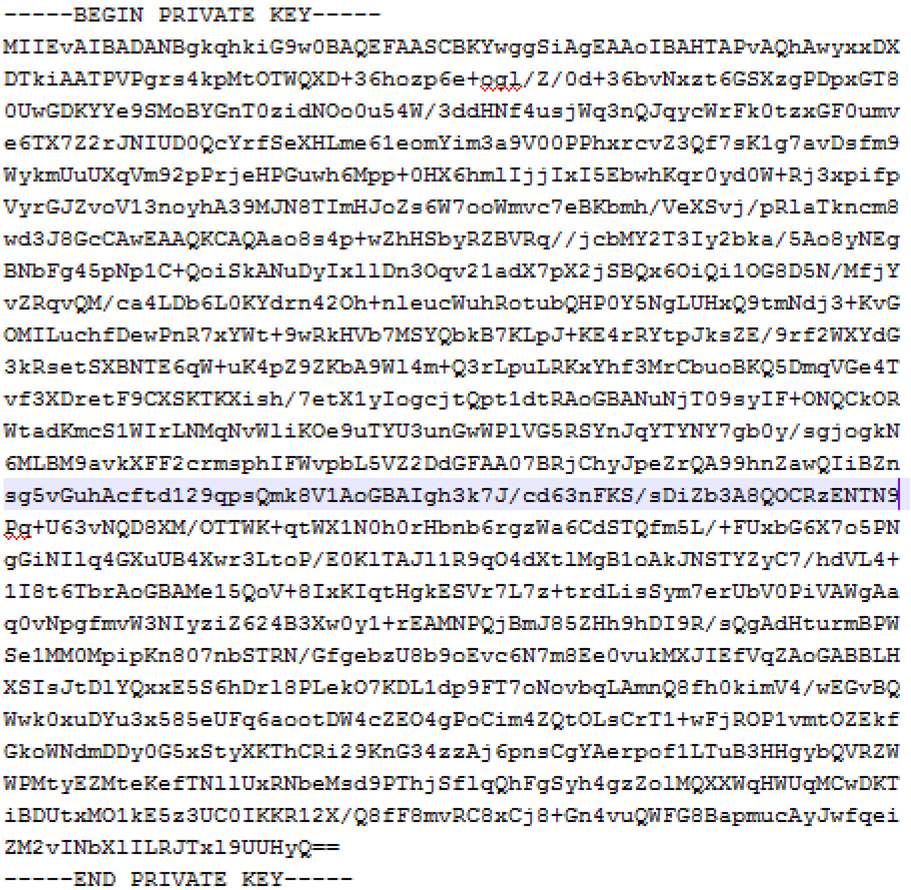
OpenSSL command: genrsa –out mvotrprivkey.key 2048



```
OpenSSL> genrsa -out mvotrserverprivkey.key 2048
Loading 'screen' into random state - done
Generating RSA private key, 2048 bit long modulus
.................................................
.............+++
..+++
```

Figure 5.2 OpenSSL Key Generation

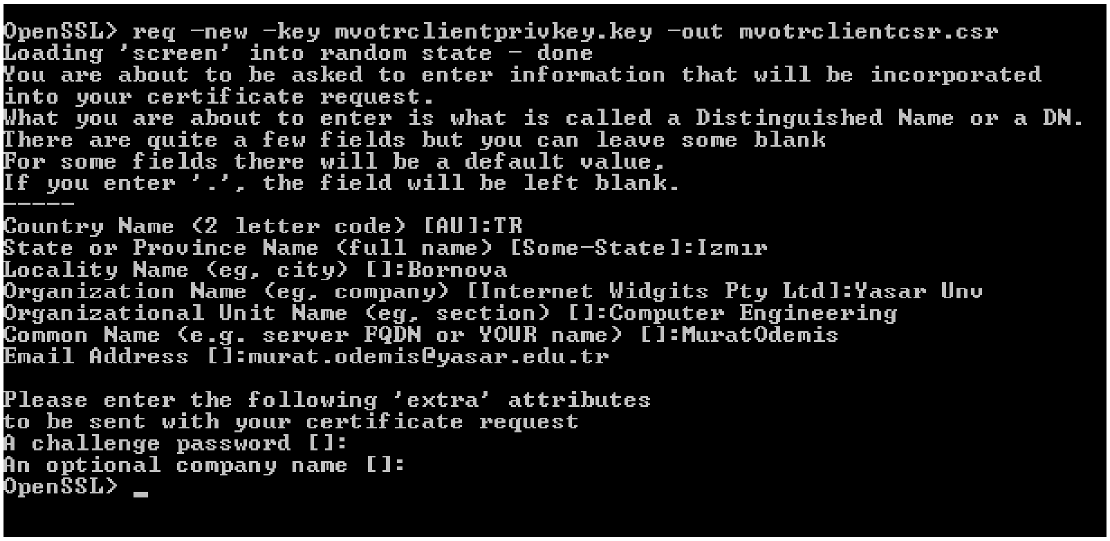
Private Key after this operation:



```
-----BEGIN PRIVATE KEY-----
MIIEvAIBADANBgkqhkiG9w0BAQEFAASCBKYwggSiAgEAAoIBAHTAPvAQhAwyxxDX
DTkiAATPVPgrs4kpMtOTWQXD+36hozp6e+qgl/Z/0d+36bvNxzt6GSXzgPDpxGT8
0UwGDKYYe9SMoBYGnT0zidNOo0u54W/3ddHNf4usjWq3nQJqycWrFk0tzxGF0umv
e6TX7Z2rJNIUD0QcYrfSeXHLme61eomYim3a9V00PPhxrcvZ3Qf7sK1g7avDsfm9
WykmUuUXqVm92pPrjeHPGuwh6Mpp+0HX6hmlIjjIxI5EbwhKqr0yd0W+Rj3xpifp
VyrGJZvoV13noyhA39MJN8TImHJoZs6W7ooWmvc7eBKbmh/VeXSvj/pRlaTkncm8
wd3J8GcCAwEAAQKCAQAao8s4p+wZhHSbyRZBVRq//jcbMY2T3Iy2bka/5Ao8yNEg
BNbFg45pNp1C+QoiSkANuDyIx11Dn3Oqv21adX7pX2jSBQx6OiQi1OG8D5N/MfjY
vZRqvQM/ca4LDb6L0KYdrn42Oh+nleucWuhRotubQHP0Y5NgLUHxQ9tmNdj3+KvG
OMILuchfDewPnR7xYWt+9wRkHVb7MSYQbkB7KLpJ+KE4rRYtpJksZE/9rf2WXYdG
3kRsetSXBNTE6qW+uK4pZ9ZKbA9W14m+Q3rLpuLRKxYhf3MrCbuoBKQ5DmqVGe4T
vf3XDretF9CXSKTKXish/7etX1yIogcjtQpt1dtRAoGBANuNjT09syIF+ONQCkOR
WtadKmcS1WIrLNMqNvWliKOe9uTYU3unGwWPlVG5RSYnJqYTYNY7gb0y/sgjogkN
6MLBM9avkXFF2crmsphIFWvpbL5VZ2DdGFAA07BRjChyJpeZrQA99hnZawQIiBZn
sg5vGuhAcftd129qpsQmk8V1AoGBAIgh3k7J/cd63nFKS/sDiZb3A8QOCRzENTN9
Pg+U63vNQD8XM/OTTWK+qtWX1N0h0rHbnb6rgzWa6CdSTQfm5L/+FUxbG6X7o5PN
gGiNIlq4GXuUB4Xwr3LtoP/E0K1TAJl1R9qO4dXt1MgB1oAkJNSTYZyC7/hdVL4+
1I8t6TbrAoGBAMe15QoV+8IxKIqtHgkESVr7L7z+trdLisSym7erUbV0PiVAWgAa
q0vNpgfmvW3NIyziZ624B3Xw0y1+rEAMNPQjBmJ85ZHh9hDI9R/sQgAdHturmBPW
Se1MM0MpipKn807nbSTRN/GfgebzU8b9oEvc6N7m8Ee0vukMXJIEfVqZAoGABBLH
XSIsJtDlYQxxE5S6hDrl8PLekO7KDL1dp9FT7oNovbqLAmnQ8fh0kimV4/wEGvBQ
Wwk0xuDYu3x585eUFq6aootDW4cZEO4gPoCim4ZQtOLsCrT1+wFjROP1vmtOZEkf
GkoWNdmDDy0G5xStyXKThCRi29KnG34zzAj6pnsCgYAerpof1LTuB3HHgybQVRZW
WPMtyEZMteKefTN11UxRNbeMsd9PThjSflqQhFgSyh4gzZolMQXXWqHWUqMCwDKT
iBDUtxMO1kE5z3UC0IKKR12X/Q8fF8mvRC8xCj8+Gn4vuQWFG8BapmucAyJwfqei
ZM2vINbXlILRJTxl9UUHyQ==
-----END PRIVATE KEY-----
```

Certification Signing Request:

OpenSSL command: req -new -key mvotrclientprivkey.key –out mvotrclientcsr.csr.

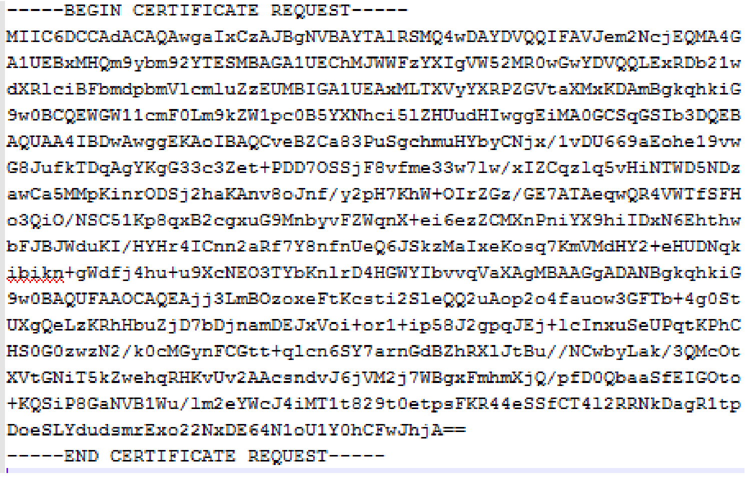


```
OpenSSL> req -new -key mvotrclientprivkey.key -out mvotrclientcsr.csr
Loading 'screen' into random state - done
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:TR
State or Province Name (full name) [Some-State]:Izmır
Locality Name (eg, city) []:Bornova
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Yasar Unv
Organizational Unit Name (eg, section) []:Computer Engineering
Common Name (e.g. server FQDN or YOUR name) []:MuratOdemis
Email Address []:murat.odemis@yasar.edu.tr

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
OpenSSL> _
```

Figure 5.3 OpenSSL Certification Signing Request

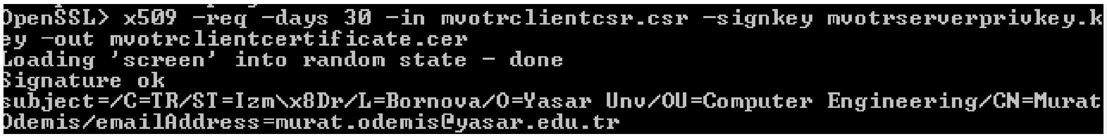
Certification request after this operation:



```
-----BEGIN CERTIFICATE REQUEST-----
MIIC6DCCAdACAQAwgaIxCzAJBgNVBAYTAlRSMQ4wDAYDVQQIFAVJem2NcjEQMA4G
A1UEBxMHQm9ybm92YTESMBAGA1UEChMJWWFzYXIgVW52MR0wGwYDVQQLExRDb21w
dXRlciBFbmdpbmVlcmluZzEUMBIGA1UEAxMLTXVyYXRPZGVtaXMxKDAmBgkqhkiG
9w0BCQEWGW11cmF0Lm9kZW1pc0B5YXNhci5lZHUudHIwggEiMA0GCSqGSIb3DQEB
AQUAA4IBDwAwggEKAoIBAQCveBZCa83PuSgchmuHYbyCNjx/1vDU669aEohe19vw
G8JufkTDqAgYKgG33c3Zet+PDD7OSSjF8vfme33w7lw/xIZCqzlq5vHiNTWD5NDz
awCa5MMpKinrODSj2haKAnv8oJnf/y2pH7KhW+OIrZGz/GE7ATAeqwQR4VWTfSFH
o3QiO/NSC51Kp8qxB2cgxuG9MnbyvFZWqnX+ei6ezZCMXnPniYX9hiIDxN6Ehthw
bFJBJWduKI/HYHr4ICnn2aRf7Y8nfnUeQ6JSkzMaIxeKosq7KmVMdHY2+eHUDNqk
ibikn+gWdfj4hu+u9XcNEO3TYbKnlrD4HGWYIbvvqVaXAgMBAAGgADANBgkqhkiG
9w0BAQUFAAOCAQEAjj3LmBOzoxeFtKcsti2SleQQ2uAop2o4fauow3GFTb+4g0St
UXgQeLzKRhHbuZjD7bDjnamDEJxVoi+or1+ip58J2gpqJEj+lcInxuSeUPqtKPhC
HS0G0zwzN2/k0cMGynFCGtt+qlcn6SY7arnGdBZhRXlJtBu//NCwbyLak/3QMcOt
XVtGNiT5kZwehqRHKvUv2AAcsndvJ6jVM2j7WBgxFmhmXjQ/pfD0QbaaSfEIGOto
+KQSiP8GaNVB1Wu/lm2eYWcJ4iMT1t829t0etpsFKR44eSSfCT4l2RRNkDagR1tp
DoeSLYdudsmrExo22NxDE64N1oU1Y0hCFwJhjA==
-----END CERTIFICATE REQUEST-----
```

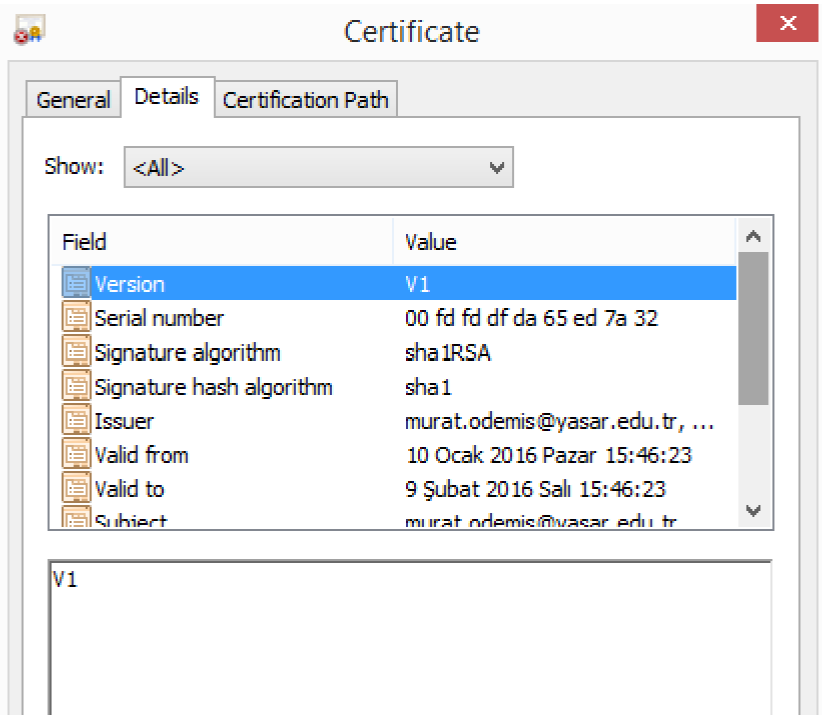
Distributing x509 Certificate to client with date of validity:

OpenSSL Command: x509 –req –days 30 –in mvotrclientcer.csr –signkey mvotrserverprivkey.key –out mvotrclientcertificate.cer



```
OpenSSL> x509 -req -days 30 -in mvotrclientcsr.csr -signkey mvotrserverprivkey.k
ey -out mvotrclientcertificate.cer
Loading 'screen' into random state - done
Signature ok
subject=/C=TR/ST=Izm\x8Dr/L=Bornova/O=Yasar Unv/OU=Computer Engineering/CN=Murat
Odemis/emailAddress=murat.odemis@yasar.edu.tr
```

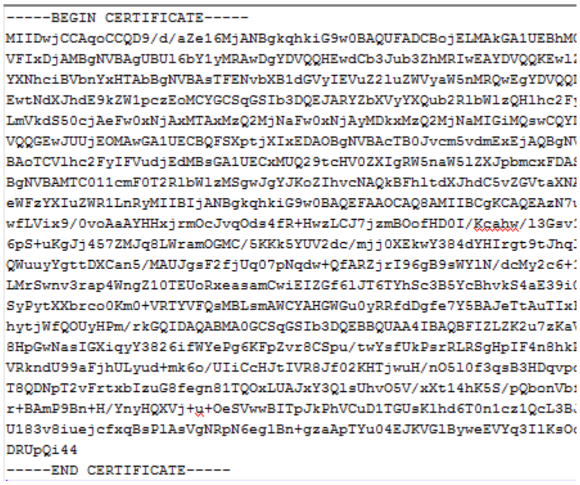
Figure 5.4 OpenSSL distributing certificate



Show: <All>

Field	Value
Version	V1
Serial number	00 fd fd df da 65 ed 7a 32
Signature algorithm	sha1RSA
Signature hash algorithm	sha1
Issuer	murat.odemis@yasar.edu.tr, ...
Valid from	10 Ocak 2016 Pazar 15:46:23
Valid to	9 Şubat 2016 Salı 15:46:23
Subject	murat.odemis@yasar.edu.tr

V1



```
-----BEGIN CERTIFICATE-----
MIIDwjCCAqoCCQD9/d/aZe16MjANBgkqhkiG9w0BAQUFADCBojELMAkGA1UEBhM
VFIxDjAMBgNVBAgUBU16bY1yMRAwDgYDVQQHEwdCb3Jub3ZhMRIwEAYDVQQKEwl
YXNhciBVbnYxHTAbBgNVBAsTFENvbXB1dGVyIEVuZ2luZWVyaW5nMRQwEgYDVQQ
EwtNdXJhdE9kZW1pczEoMCYGCSqGSIb3DQEJARYZbXVyYXQub2RlbW1zQH1hc2F
LmVkdS50cjAeFw0xNjAxMTAxMzQ2MjNaFw0xNjAyMDkxMzQ2MjNaMIGiMQswCQY
VQQGEwJUUjEOMAwGA1UECBQFSXptjXIxEDAOBgNVBAcTB0Jvcm5vdmExEjAQBgN
BAoTCVlhc2FyIFVudjEdMBsGA1UECxMUQ29tcHV0ZXIgRW5naW51ZXJpbmcxFDA
BgNVBAMTC011cmF0T2R1bW1zMSgwJgYJKoZIhvcNAQkBFhltdXJhdC5vZGVtaXN
eWFzYXIuZWR1LnRyMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAzN7
wfLVix9/0voAaAYHHxjrmOcJvqOds4fR+HwzLCJ7jzmBOofHD0I/Kcahw/13Gsv
6pS+uKgJj457ZMJq8LWramOGMC/5KKk5YUV2dc/mjj0XEkwY384dYHIrgt9tJhq
QWuuyYgttDXCan5/MAUJgsF2fjUq07pNqdw+QfARZjrI96gB9sWYlN/dcMy2c6+
LMrSwnv3rap4WngZ10TEUoRxeasamCwiEIZGf61JT6TYhSc3B5YcBhvkS4aE39i
SyPytXXbrco0Km0+VRTYVFQsMBLsmAWCYAHGWGu0yRRfdDgfe7Y5BAJeTtAuTIx
hytjWfQOUyHPm/rkGQIDAQABMA0GCSqGSIb3DQEBBQUAA4IBAQBFIZLZK2u7zKa
8HpGwNasIGXiqyY3826ifWYePg6KFpZvr8CSpu/twYsfUkPsrRLRSgHpIF4n8hk
VRkndU99aFjhULyud+mk6o/UIiCcHJtIVR8Jf02KHTjwuH/n0510f3qsB3HDqvp
T8QDNpT2vFrtxbIzuG8fegn81TQOxLUAJxY3QlsUhvO5V/xXt14hK5S/pQbonVb
r+BAmP9Bn+H/YnyHQXVj+u+OeSVwwBITpJkPhVCuD1TGUsKlhd6T0n1cz1QcL3B
U183v8iuejcfxqBsPlAsVgNRpN6eglBn+gzaApTYu04EJKVGlByweEVYq3IlKsO
DRUpQi44
-----END CERTIFICATE-----
```

Figure 5.5 Certificate which is will be to KeyChain

5.4 HMAC-SHA

A message authentication code (MAC) checks the sum of information that is being sent across an insecure channel. HMAC does the same thing, except it can use secret key, and it hashes with Secure Hash Algorithm (SHA). SHA size can be 1,128,256 or 512. HMAC provides integrity, as well as authenticity. When the client and server use pre-shared symmetric keys to calculate the HMACs, each side can be confident that the other's program has the secret key. This establishes the authenticity of both the server and client.

For another interpretation, this application's HMAC workflow is seen in Figure 5.6.

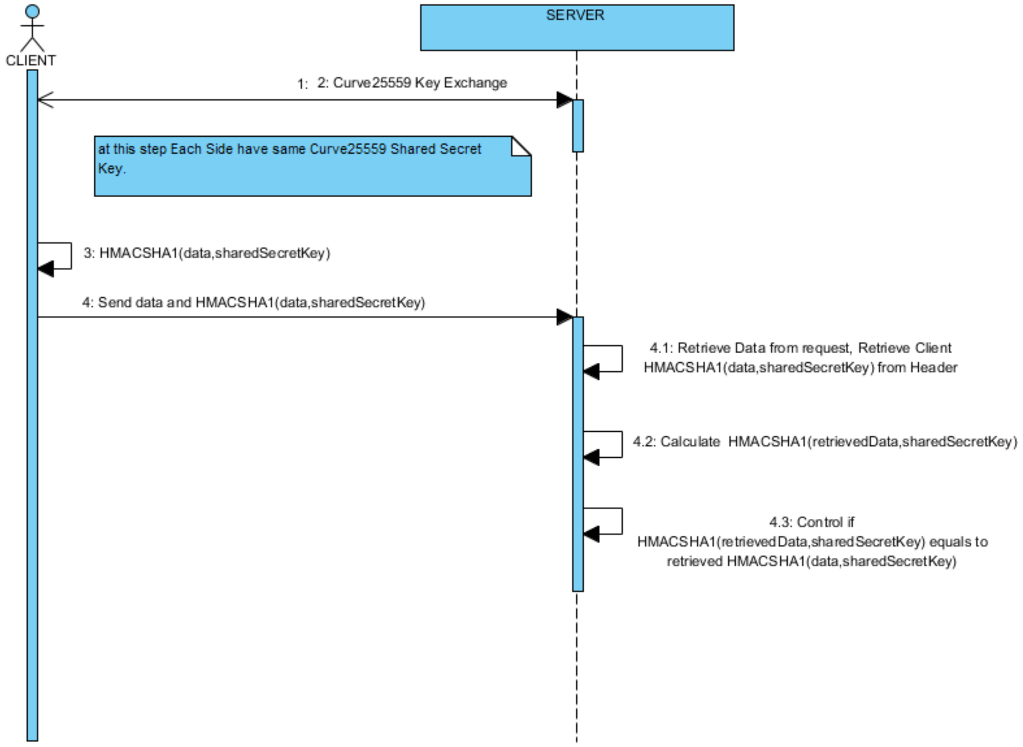

CLIENT
SERVER
1: 2: Curve25559 Key Exchange
at this step Each Side have same Curve25559 Shared Secret Key.
3: HMACSHA1(data,sharedSecretKey)
4: Send data and HMACSHA1(data,sharedSecretKey)
4.1: Retrieve Data from request, Retrieve Client HMACSHA1(data,sharedSecretKey) from Header
4.2: Calculate HMACSHA1(retrievedData,sharedSecretKey)
4.3: Control if HMACSHA1(retrievedData,sharedSecretKey) equals to retrieved HMACSHA1(data,sharedSecretKey)

Figure 5.6 HMAC illustration

Step 1: The key exchange function is requested and completed. In order to compute the HMAC of data, both sides (server and user) must have the same secret key.

Step 2: The MAC SHA1 data translation is computed by the client's shared secret key.

Step 3: Data is sent via an Ajax request. The computed HMACs are sent to the header.

Step 4: On the server's side, the HMAC SHA1 codes are computed by the same shared secret key that was generated by the key exchange.

Step 5: HMAC SHA1 data that is retrieved from the processor is compared to the “lost verification” data in the network. In other words, if computed HMACs are not identical then the process is stopped.

Example:

Client Side:

plainText = “helloworld”

Secret Key: i94k24PPa8/aN7Jbm4jx9JNMsuJrI5/dQxwtAli4sn4=

Message Digest Algorithm: SHA1

Computed HMAC: f1ddc491989a04e41e029dcbb6abeea47f2d6661

Server Side:

plainText = “helloworld”

Secret Key: i94k24PPa8/aN7Jbm4jx9JNMsuJrI5/dQxwtAli4sn4=

Message Digest Algorithm: SHA1

Computed HMAC: f1ddc491989a04e41e029dcbb6abeea47f2d6661

If computed HMACs are not identical then the process is stopped.

6 COMPUTATIONAL RESULTS

Table 6.1 shows the computational results of the application.

As a test device iPhone 6 Plus was used to implement the application. iPhone 6Plus has 1.4GHZ Apple A8 Processor and 1 GB of RAM.

This application is also tested for:

Table 6.1 Test Device Specifications

Mobile Device	CPU	RAM
iPhone 5S	Apple a7 1.3 GHz Cyclone	1GB Ram
iPhone 4s	Apple a5 1.0 GHZ	512 MB Ram
Samsung Galaxy Note 8	1.6 GHz quad-core Samsung Exynos 4412 quad SoC processor	2GB Ram

Table 6.2 Runtime of Algorithms

Mobile Device	Curve25519 KeyPair Generation	AES Encryption with 256 Bit Session Key	RSA KeyPair Generation	Curve25519 Shared Key Generation
iPhone 6 Plus	~5.5ms	~3.2ms	~169.6ms	~14.6ms
iPhone 5s	~5.8ms	~3.6ms	~186.3ms	~16.1ms
iPhone 4s	~6.9ms	~4.4ms	~203.3ms	~19.3ms
Samsung Galaxy Note 8	~8ms	~4.8ms	~212.2ms	~22.3ms

In table 6.2 the results of the average of 1000 iterations are shown. It can be seen that Curve25519 KeyExchange nearly 30 times faster than RSA KeyExchange and by using the Curve25519 KeyExchange speed of the application outperforms.

7 CONCLUSION AND FUTURE WORK

In this thesis, a reliable mobile voting application is tried to be developed. The developed application can be used by the government for the general and local elections and also it is adaptable for any type of election that carried out by the universities or the other institutions. The system is intended to provide authorization based voting. In other words, only the authorized people are able to display and cast their vote in the selections through the developed application.

Aforementioned rules such as democracy, privacy, accuracy, fairness, security, integrity, consistency, authenticity, eligibility, mobility, uniqueness, verifiability, uncoercibility, and cost effectiveness are some of the mobile voting rules. If the developed mobile voting and administration system in this thesis were considered, user authentication part satisfies the democracy rule. In the system, just the authenticated users can cast their votes. Also, the implemented system is appropriate for the multiple language options, so this feature is also satisfies the democracy rule.

The most important aspects of the e-voting and the mobile voting is privacy. In the developed system the id of the voters are never recorded. Users cast their votes, but their ids or names were never kept in the system so anonymity is provided. Just election id, voted option id and voting date are recorded to the database. While providing the anonymity, also the standard of the uniqueness should be provided. In the system user can cast only a single vote. Database records the election that is voted by the user. Therefore, system records that which user casted his/her vote to which election, not the option. All these processes in this system were performed by using database; also the more developed systems can be applied. In this system, an extra precautions were not taken into account for the vote counting process. In the future studies, Threshold Cryptosystem (Damgard and Jurik, 2003), which shares out the keys of the vote counting to the users, can be used and this system will enrich the developed application. For instance in the general elections, the keys, which are needed to display the results of the ballot, can be distributed to the representatives of the political party. The representatives cannot display the results without coming together at the same time. Basically, developing this kind of system will be more legitimate. Therefore, vote counting is going to be more reliable, people do not need to trust just a single institution. In the Threshold Cryptosystem, several parties must cooperate in the decryption protocol.

System provides the standard of fairness, but because the voting time is not added, after the end of the voting period, the result of the voting is not shown to the users from the application. Results of the votes just can be displayed by the institution or the system administrator. This part can be further improved. Security mechanisms that are used for the developed system are explained in the previous sections and several security steps were added. Thus, security and complex security rules were satisfied. However, implementing that much security steps can cause a conflict in the cost-effectiveness. Although, the voting process takes long time, Curve25519 key exchange scheme gives result quickly so the time tried to be minimized. In the system consumed time is insignificant for the cryptology schemes, but sms part, fingerprint detection, entering geometric pattern password take much more time. In the system, database and the server is reliable for the integrity but extra cautions can be considered in the future works. Mobility is totally provided, user do not have any limitation for the location, and they can cast their votes from everywhere. Authenticity is provided by SSL and user credential submission is improvement for authenticity. Also, in the system having another activation mechanisms that are explained before, provides Authenticity. Any extra study for the Verifiability is not carried out. User casts the vote, but if it is asked that the casted vote belongs to that user or not, then it is answered as a database query. This part can be further improved. Extra step is developed to satisfy uncoercibility rule, while most of the e-voting and mobile voting applications do not have any study for that rule. In this thesis, user can cast the vote by entering the system with his/her fake password or fake geometric password, but the vote will be invalid. Therefore, user does not have to cast the vote to the undesirable option while he/she is under pressure. Furthermore, if the phone is shaken over a predetermined threshold, then the application is locked and fingerprint detection system enables, so this feature also satisfies the uncoercibility.

System has been optimized with respect to the speed and the performance within the application. The security side of the application can be further improved. In this system, while generating the prime numbers or implementing the cryptology algorithms, frameworks and plugins are used in some cases. Using these frameworks and plugins causes dependence on other sources whereas most of them are open-source. For instance, in some encryption processes, library of System.Crypthology in Microsoft is used and this causes a dependence on Microsoft. All the implementations of the security processes should be carried out within the developed system. For the future work, unique libraries can be implemented for the processes of the system such as; generation of prime numbers, generation of random numbers, encryption etc.

Zero-Knowledge Proof (Baudron et al., 2001) system that is similar to the Threshold Cryptosystem can also be used. It is a method by which one political party (the prover) can prove to another political party (the verifier) that a given statement is true, without conveying any information apart from the fact that the statement is indeed true.

IOS operating system based mobile application was implemented in this thesis. Also the application based on Android operating system was developed but have not tested rather well. In addition to the mobile application, web site is also designed. However, extra methods about web security has not integrated. Completion of the Android based application and the secured web site will be studied in the future.

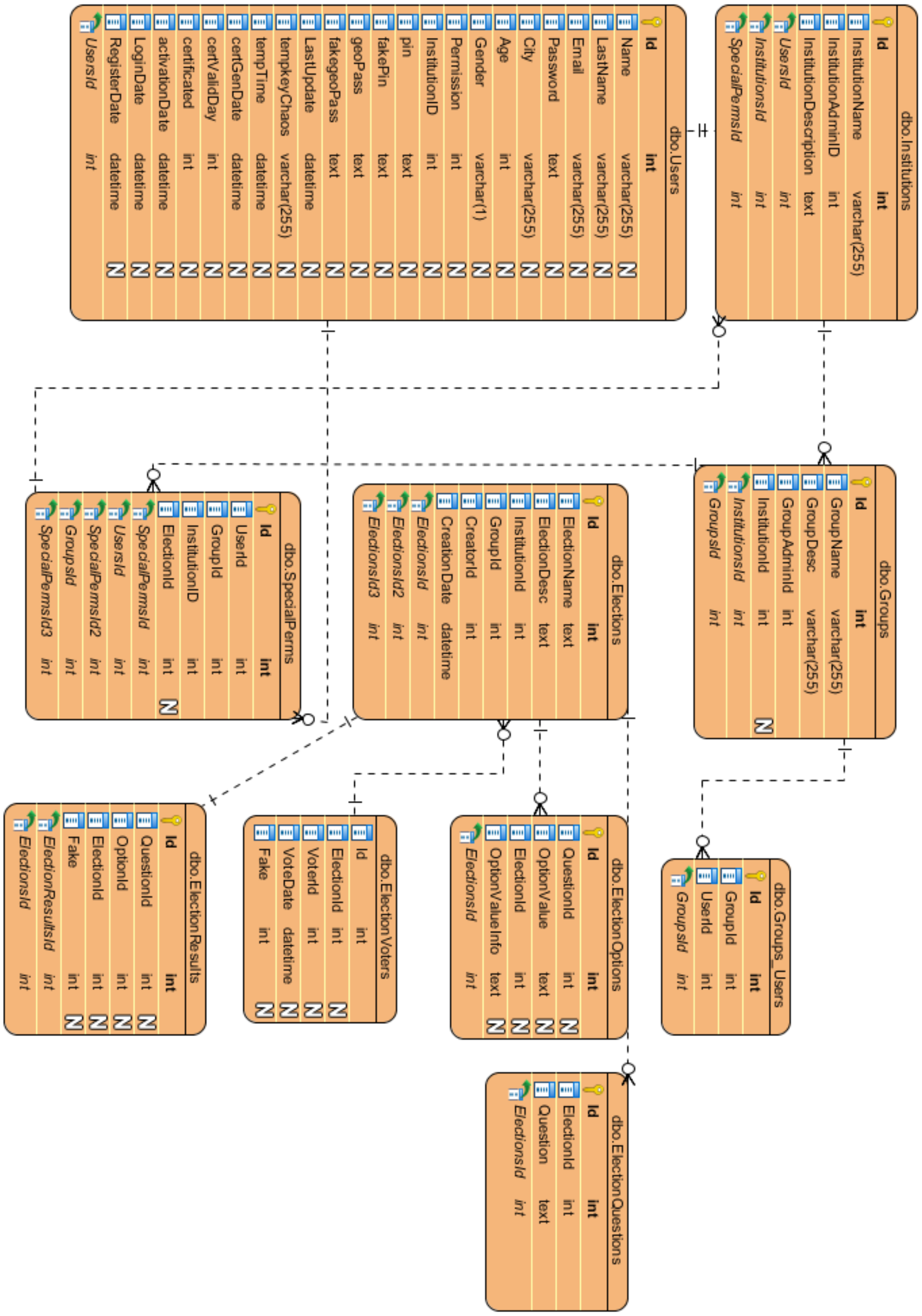
APPENDIX A

Frameworks & Plugin & Tools

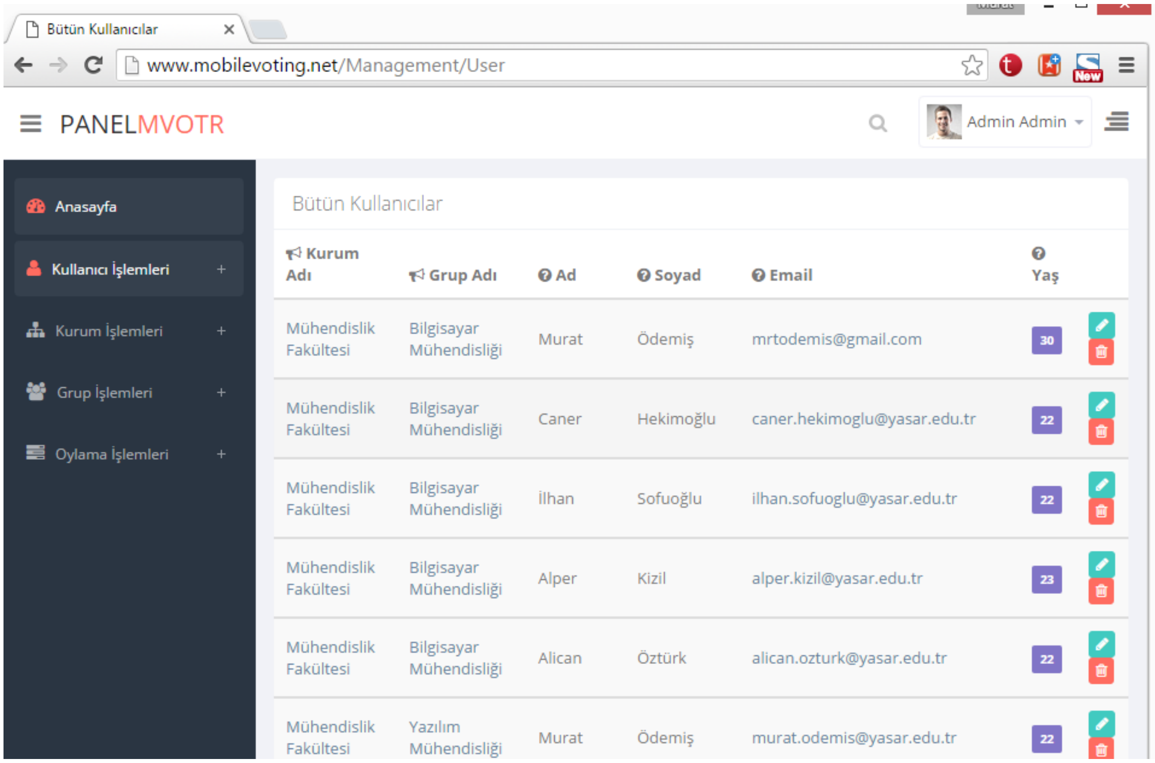
Cordova	https://cordova.apache.org/	Apache
Xcode	https://developer.apple.com/xcode/	Apple
Crypto-js	https://code.google.com/p/crypto-js/	Google / Jeff Mott
Ecma-Nacl	https://github.com/3nsoft/ecma-nacl	3nsoft
OOP JS .NET PORT	http://www.jocys.com/Common/JsClasses/Documents	Ejocys
Curve25519-JS	https://github.com/hanswolff?tab=repositories	Hans Wolff
Local Notification	https://github.com/katzer/cordova-plugin-local-notifications	Katzer
Device Identifier	https://github.com/mgcrea/cordova-secureudid	Oliver Lovignes
jQueryMobile	http://jquerymobile.com/	jQuery
jQuery	http:/jquery.com	jQuery
OpenSSL	https://www.openssl.org/	OpenSSL
patternLock	https://github.com/s-yadav/patternLock	Sudhanshu Yadav
Shake	https://github.com/leecrossley/cordova-plugin-shake	Lee Crossley
FastClick	https://ftlabs.github.io/fastclick/	FtLabs
Hammer.js	http://hammerjs.github.io/jquery-plugin/	Alexander schmitz
Modernizr	https://modernizr.com/	Modernizr
Wikitude FaceRecognation	https://www.wikitude.com/	Matteo Spinelli
Chaos.Nacl	https://github.com/CodesInChaos/Chaos.NaCl	Wikitude
Flavr popup	http://codecanyon.net/item/flavr-flat-jquery-popup-dialog/7021021	Dierg

APPENDIX B

Administration System Entity Relationship and Database Diagram

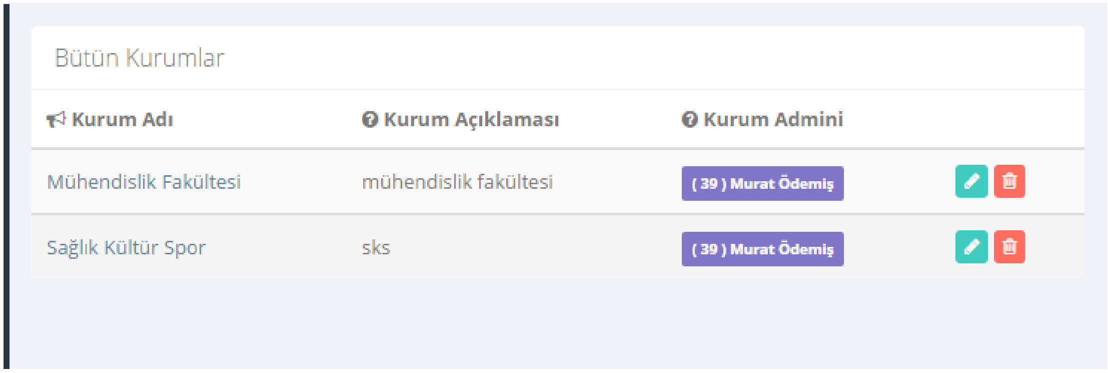


Administration Panel Screenshots



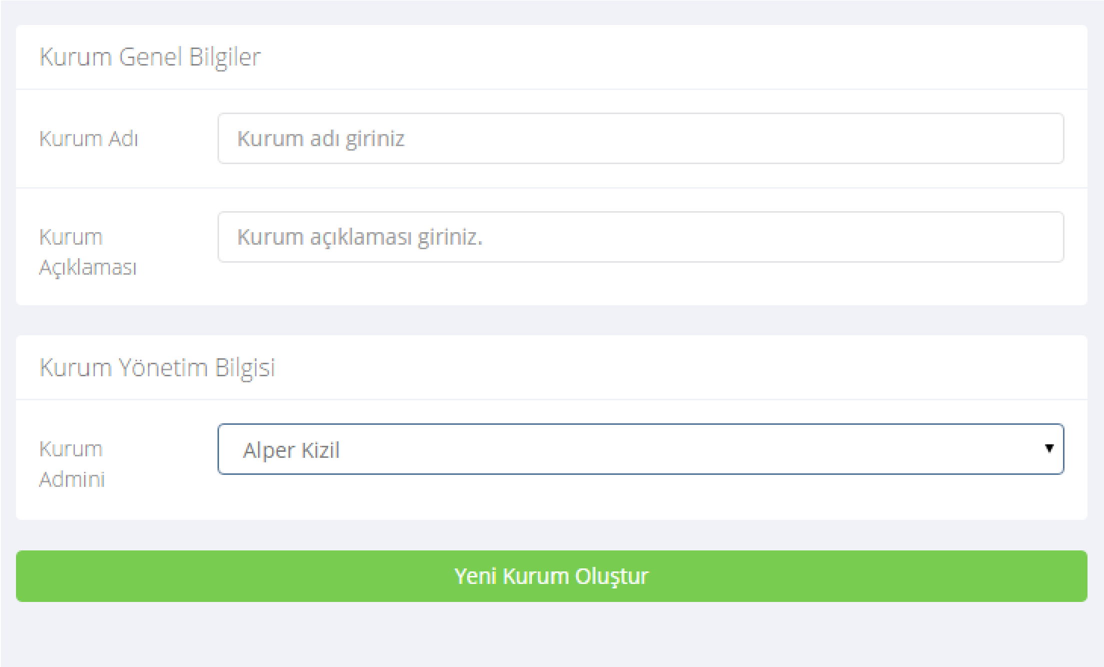
Bütün Kullanıcılar

Kurum Adı	Grup Adı	Ad	Soyad	Email	Yaş
Mühendislik Fakültesi	Bilgisayar Mühendisliği	Murat	Ödemiş	mrtodemis@gmail.com	30
Mühendislik Fakültesi	Bilgisayar Mühendisliği	Caner	Hekimoğlu	caner.hekimoglu@yasar.edu.tr	22
Mühendislik Fakültesi	Bilgisayar Mühendisliği	İlhan	Sofuoğlu	ilhan.sofuoglu@yasar.edu.tr	22
Mühendislik Fakültesi	Bilgisayar Mühendisliği	Alper	Kizil	alper.kizil@yasar.edu.tr	23
Mühendislik Fakültesi	Bilgisayar Mühendisliği	Alican	Öztürk	alican.ozturk@yasar.edu.tr	22
Mühendislik Fakültesi	Yazılım Mühendisliği	Murat	Ödemiş	murat.odemis@yasar.edu.tr	22



Bütün Kurumlar

Kurum Adı	Kurum Açıklaması	Kurum Admini
Mühendislik Fakültesi	mühendislik fakültesi	(39) Murat Ödemiş
Sağlık Kültür Spor	sks	(39) Murat Ödemiş

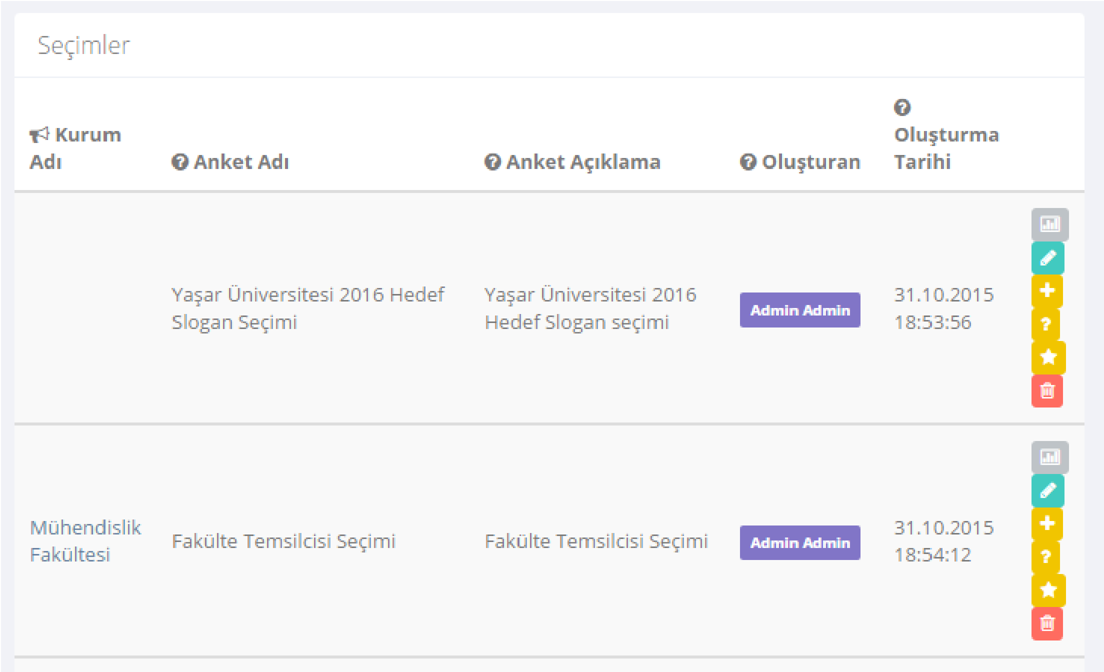


Kurum Genel Bilgiler

Kurum Adı	Kurum adı giriniz
Kurum Açıklaması	Kurum açıklaması giriniz.

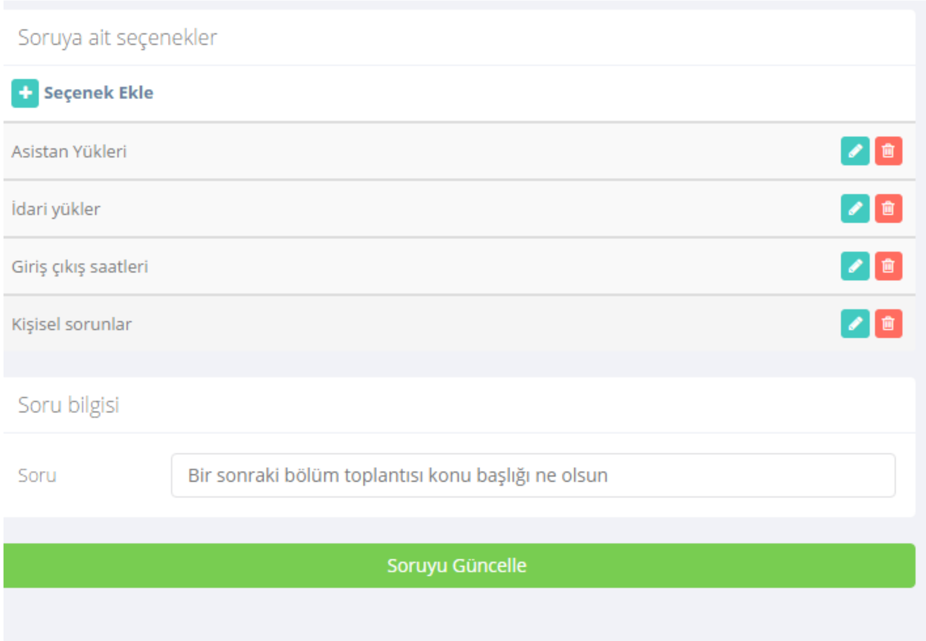
Kurum Yönetim Bilgisi

Kurum Admini	Alper Kizil ▾



Seçimler

Kurum Adı	Anket Adı	Anket Açıklama	Oluşturan	Oluşturma Tarihi	
	Yaşar Üniversitesi 2016 Hedef Slogan Seçimi	Yaşar Üniversitesi 2016 Hedef Slogan seçimi	Admin Admin	31.10.2015 18:53:56	✎ + ? ★
Mühendislik Fakültesi	Fakülte Temsilcisi Seçimi	Fakülte Temsilcisi Seçimi	Admin Admin	31.10.2015 18:54:12	✎ + ? ★



Soruya ait seçenekler

Asistan Yükleri

İdari yükler

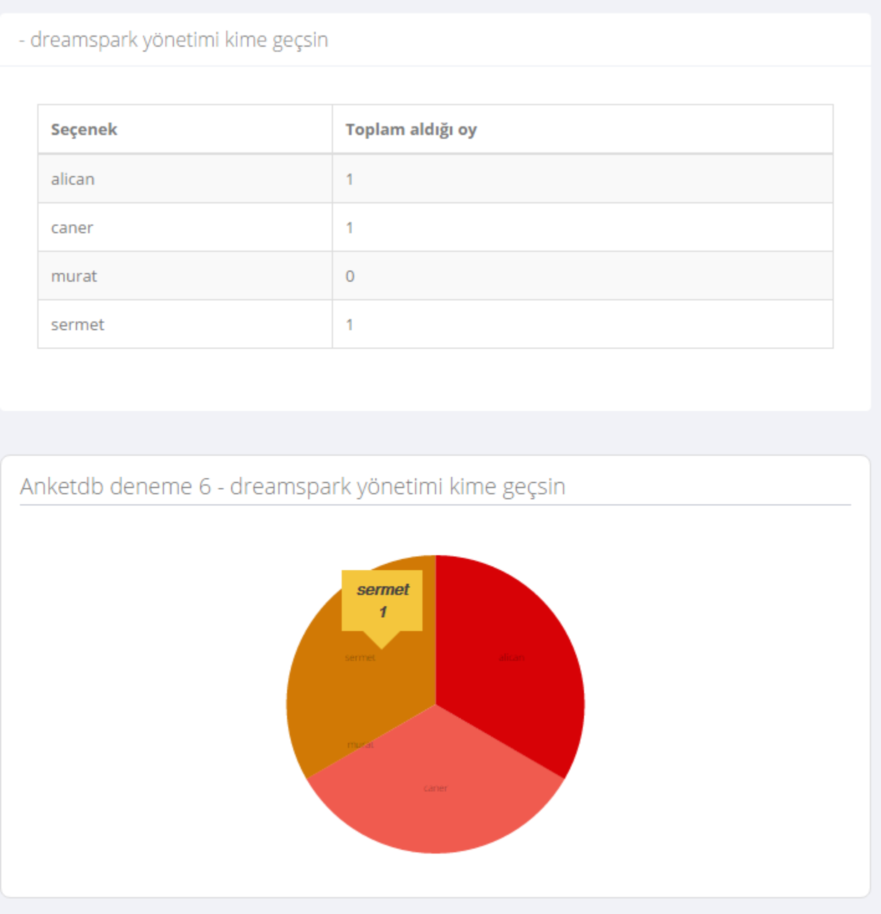
Giriş çıkış saatleri

Kişisel sorunlar

Soru bilgisi

Soru: Bir sonraki bölüm toplantısı konu başlığı ne olsun

Votes



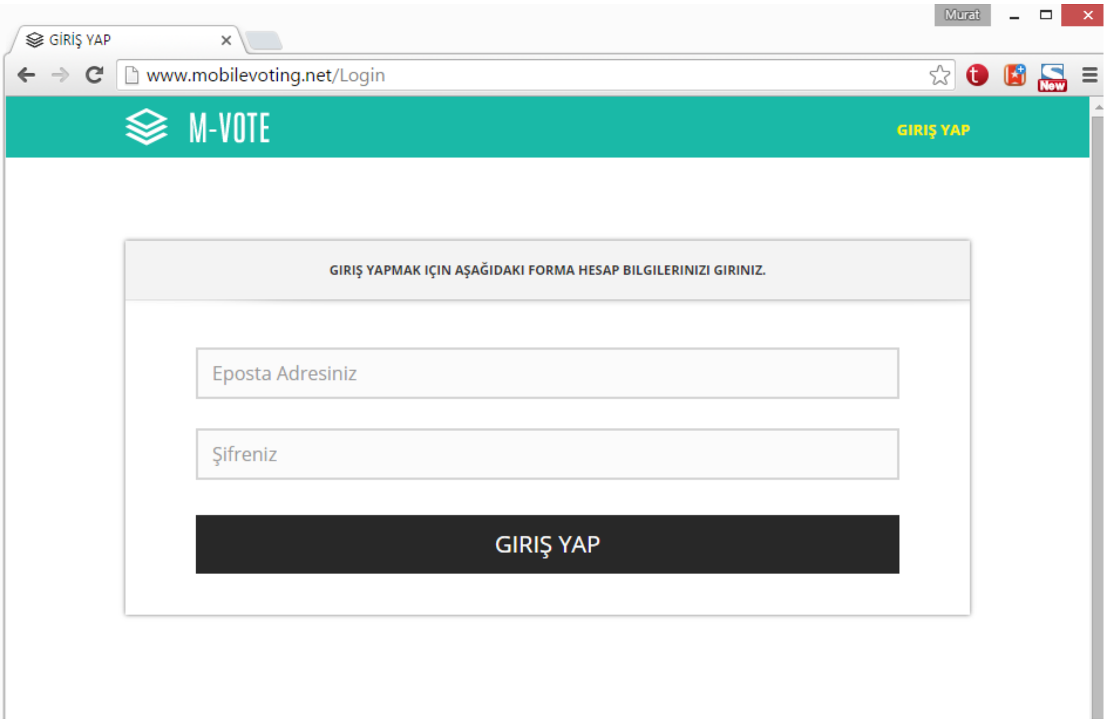
- dreamspark yönetimi kime geçsin

Seçenek	Toplam aldığı oy
alican	1
caner	1
murat	0
sermet	1

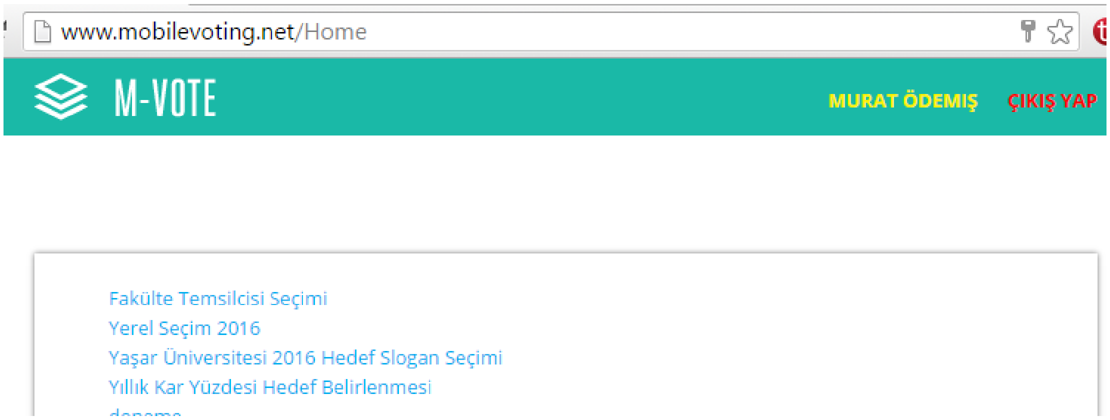
Anketdb deneme 6 - dreamspark yönetimi kime geçsin

WEBSITE SCREENSHOTS

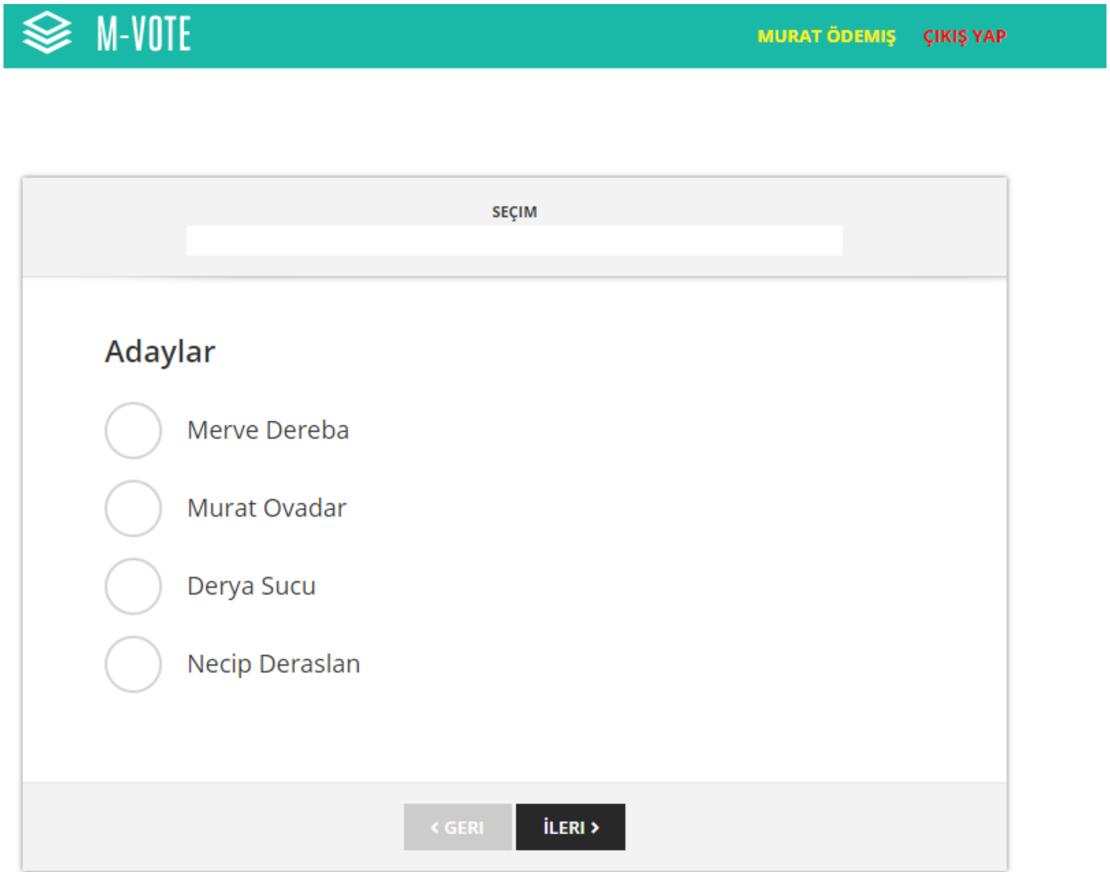
Website is designed for future work. It is working now but all the security phases doesn’t implemented.



GIRIŞ YAPMAK IÇIN AŞAĞIDAKI FORMA HESAP BILGILERINIZI GIRINIZ.



Fakülte Temsilcisi Seçimi
Yerel Seçim 2016
Yaşar Üniversitesi 2016 Hedef Slogan Seçimi
Yıllık Kar Yüzdesi Hedef Belirlenmesi
deneme



SEÇIM

Adaylar

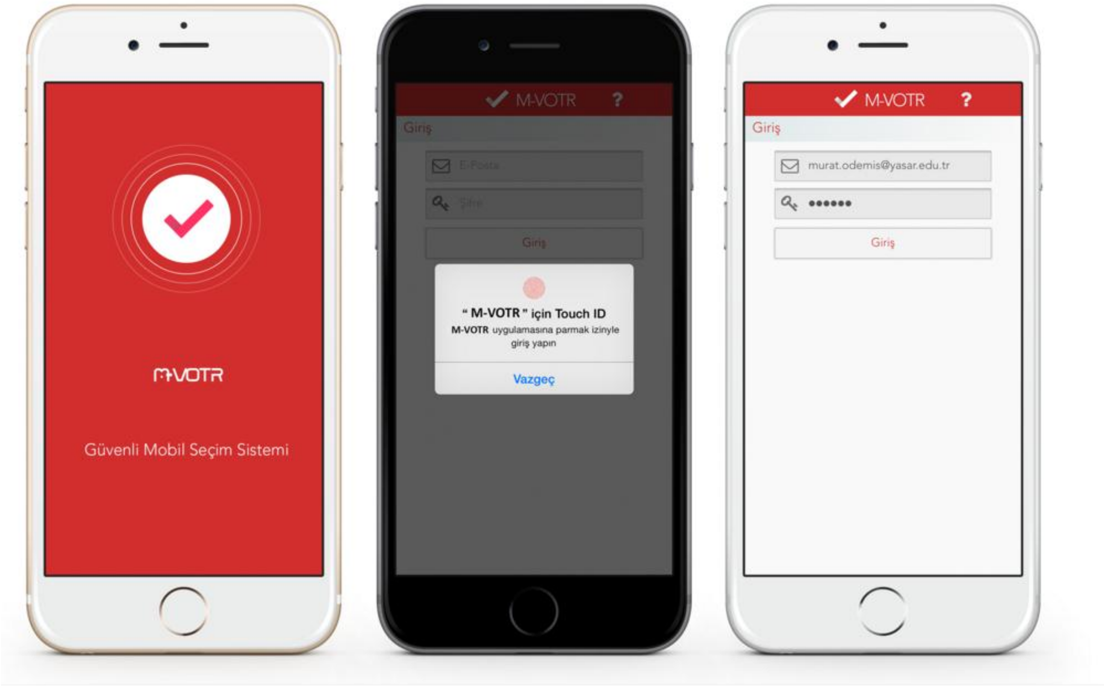
○ Merve Dereba

○ Murat Ovadar

○ Derya Sucu

○ Necip Deraslan

MOBILE APPLICATION SCREENSHOTS

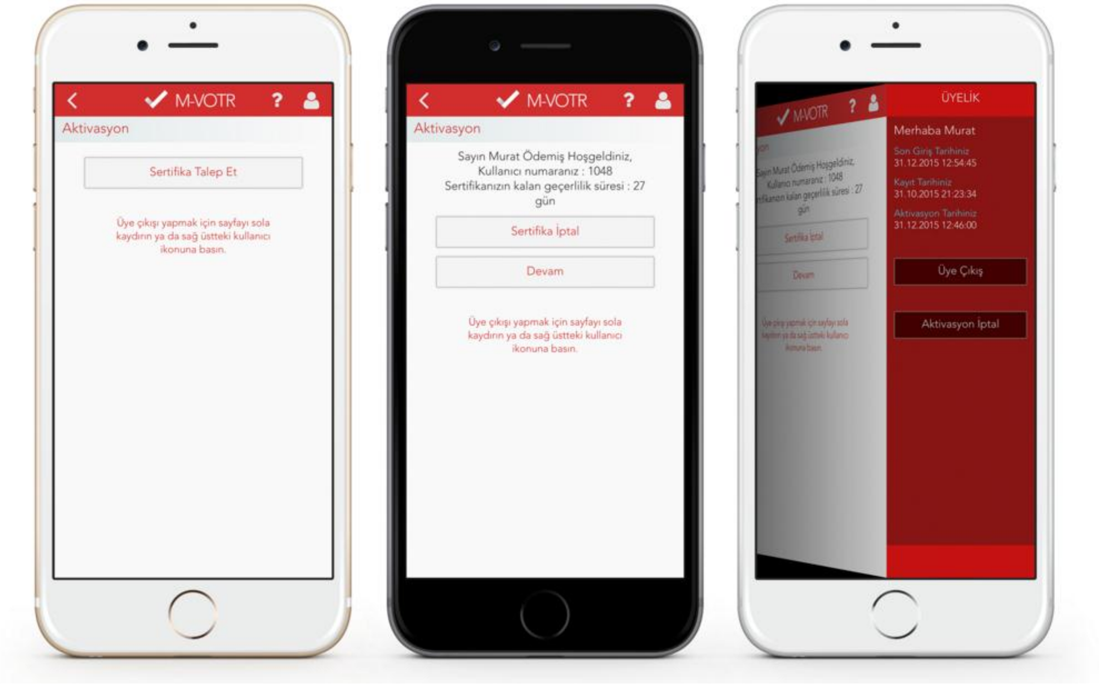


“ M-VOTR ” için Touch ID

M-VOTR uygulamasına parmak izinyle giriş yapın

murat.odemis@yasar.edu.tr

••••••



Üye çıkışı yapmak için sayfayı sola kaydırın ya da sağ üstteki kullanıcı ikonuna basın.

Sayın Murat Ödemiş Hoşgeldiniz,
Kullanıcı numaranız : 1048
Sertifikanızın kalan geçerlilik süresi : 27 gün

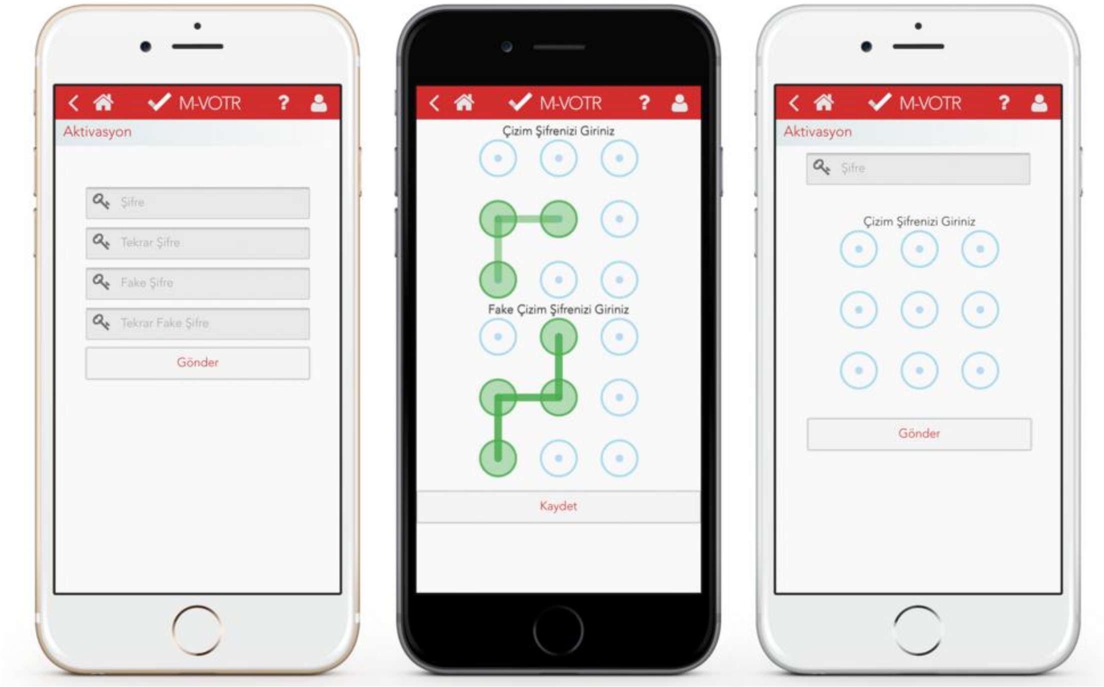
Üye çıkışı yapmak için sayfayı sola kaydırın ya da sağ üstteki kullanıcı ikonuna basın.

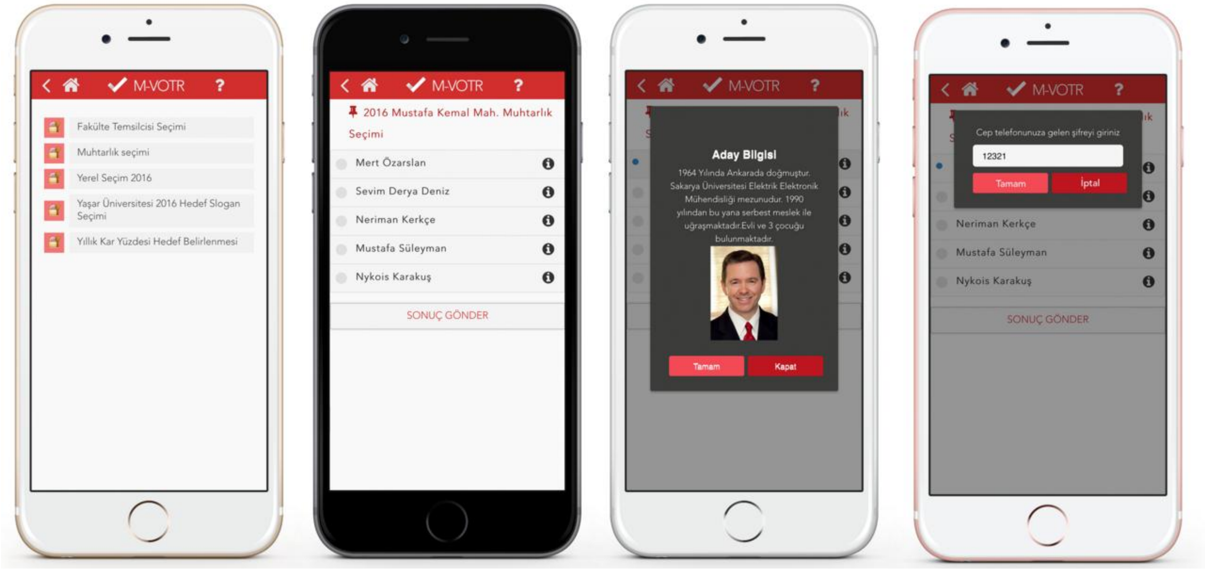
Merhaba Murat

Son Giriş Tarihiniz
31.12.2015 12:54:45

Kayıt Tarihiniz
31.10.2015 21:23:34

Aktivasyon Tarihiniz
31.12.2015 12:46:00





1964 Yılında Ankarada doğmuştur. Sakarya Üniversitesi Elektrik Elektronik Mühendisliği mezunudur. 1990 yılından bu yana serbest meslek ile uğraşmaktadır.Evli ve 3 çocuğu bulunmaktadır.

Cep telefonunuza gelen şifreyi giriniz

12321

8 REFERENCES

Cisco Systems Inc., “Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2014–2019” http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.pdf, (2015) (Access Date: 03.01.2016)

Ahmad T., Hu J., Han S., An Efficient mobile voting system security scheme based on Elliptic Curve Cryptography. NSS 2009 - Network and System Security, 474-479.

Alrodhan W.A., Alturbaq A., Aldahlawi S., A Mobile Biometric-Based E-Voting Scheme. 2014 World Symposium on Computer Applications and Research, WSCAR.

Auditmark, “Jscrambler”, https://jscrambler.com/en (2007), (Access Date: 12.01.2016)

Baudron O., Fouque P.A, Pointcheval D., Stern J., Poupard G., Practical multi-candidate election system. Proceedings of the Twentieth Annual ACM Symposium on Principles of Distributed Computing: PODC 2001, Newport, Rhode Island, United States. ACM, pp.274-283.

Bernstein J. Daniel, Curve25519: new Diffie-Hellman speed records. 2006. 9th International Conference on Theory and Practice in Public-Key Cryptography, pp.207-228

Bernstein D., Tanja L, Schwabe P., The security impact of a new cryptographic library. 2012, Proceedings of LatinCrypt pp159-176.

Biswas, Sujit, Gsm Verification Based Secure E-Voting Framework. 2015, International Journal of U- & E-Service, Science & Technology. Vol. 8 Issue 1, p231-237. 7p.

Campanelli S., Falleni A., Martinelli F., Petrocchi M., Vaccarelli A., Mobile implementation and formal verification of an e-voting system. 2008, Proceedings -

3rd International Conference on Internet and Web Applications and Services, ICIW: 476-481.

Campbell B.A., Tossell C.C., Byrne M.D., Kortum P., Voting On A Smartphone: Evaluating The Usability of An Optimized Voting System For Handheld Mobile Device. 2011, Proceedings of the Human Factors and Ergonomics Society Annual Meeting, vol. 55 no. 1 1100-1104.

Chaum D., Untraceable electronic mail, return addresses, and digital pseudonyms. 1981, Communications of ACM, vol 24, no. 2, pp-84-88.

Chang C. and Lee J., An anonymous voting mechanism based on the key exchange protocol. 2006, Computer and Security Volume 25 Issue 4, pp 307-314

CodesInChaos Inc., "Chaos.NaCl cryptography library", https://github.com/CodesInChaos/Chaos.NaCl (2013),(Access Date: 01.01.2016)

Cooper D, Santesson S, Farrell S, Boeyen S, Housley R, Polk W., "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", https://tools.ietf.org/html/draft-ietf-pkix-rfc3280bis-11, (2008) (Access Date: 14.1.2016)

CordovaApache,
"Overview",https://cordova.apache.org/docs/en/latest/guide/overview/ (2015) (Access Date: 11.12.2016)

Chun T.L and Hwang M.S., A Secure and Anonymous Electronic Voting Scheme Based on Key Exchange Protocol, 2013. International Journal of Security and Its Applications, vol. 7, no. 1, pp. 59-70

Chung Y.F. and Wu Z.Y., Casting ballots over internet connection against bribery and coercion. 2012, Comput. J., 55(10):1169-1179.

Cranor L. and Cytron R. K., Sensus: A Security-Conscious Electronic Polling System for the Internet. 1997, In Proc. of HICSS'97.

Damgard I. and Jurik J., A Length-Flexible Threshold Cryptosystem with Applications. 2003, BRICS Report Series ISSN: 0909-0878.

Diffie and Hellman, Exhaustive Cryptanalysis of the NBS Data Encryption Standard. 1979, IEEE Computer 10(6), June 1977, pp74-84.

Donovan G., Suresh S., Biometric Secured Mobile Voting. 2011, Second Asian Himalayas International Conference on Internet (AH-ICI), p1-6.

ElGamal Taher, A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms, 1985, IEEE Transactions on Information Theory 31 pp 469–472.

Firat M, "Securing Data in .NET" (2007), http://www.codeproject.com/Articles/21076/Securing-Data-in-NET (Access Date : 02.01.2016)

Foertsch Bob, SSL Certificates at UIIC (2004), www.cites.uiuc.edu, http://www.slideserve.com/taniel/ssl-certificates-at-uiuc,(2004)(Access Date: 05.01.2016)

NIST, FIPS PUB 197"Announcing the ADVANCED ENCRYPTION STANDARD AES", http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf, (2011) (Access Date: 03.01.2016)

GSMA, Global Mobile Economy Report, "The Mobile Economy 2015", http://www.gsmamobileeconomy.com/GSMA_Global_Mobile_Economy_Report_2015.pdf (2015) (Access Date: 03.01.2016)

Wolf H. ., "Curve25519cs cryptography library", https://github.com/hanswolff/curve25519 (2013),(Access Date: 01.01.2016)

Jaak T., Ilja T., Stanislav V., Wireless PKI Security and Mobile Voting. 2010, Computer, Vol. 43 Issue 6, p54, 7 p.

jQuery Foundation, “jQuery.ajax documentation”, http://api.jquery.com/jquery.ajax/ (2016 latest version), (Access Date: 11.01.2016)

Kao C.W., Yang C.W., Chen Y.W., Fan K.C., Hwang B.J., Huang C.P., Eye Gaze Tracking Based on Pattern Voting Scheme for Mobile Device. 2011, First International Conference on Instrumentation, Measurement, Computer, Communication & Control, p337-340, 4p.

Khelifi A., Yasmin G., Dima S., Dalya M., Shastry P.V.S., M-VOTE: A Reliable and Highly Secure Mobile Voting System. 2013 Palestinian International Conference on Information & Communication Technology, 2013, p90-98, 9p.

Koblitz N., Designs, Codes and Cryptography. 2000, Kluwer Academic Publishers, Boston, Netherlands, the State of Elliptic Curve Cryptography, 19, 173–193.

Koblitz N., Elliptic curve cryptosystems. 1987, Mathematics of Computation 48 (177): 203–209.

Koltuksuz A., Introduction to the Symmetrical Cryptography 02_symmetric lecture notes. 2012.

Kumar M., Kumar T.V.S., Hanumanthappa M., Geetha D.E., “Secure Mobile Based Voting System”. 2011, pp 324-326 [ONLINE] http://www.csi-sigegov.org/emerging_pdf/35_324-330. Pdf

Meida H., Guifa T., Chunshan W., Guandong G., “Research of Identity Authentication of the Mobile Terminal Voting System” Intelligent Information Hiding and Multimedia Signal Processing. 2013, Ninth International Conference. 2013. pp-198-201

Mohit M.S.S, Karthik M., Rajavel T., Sangeetha M.J., E-Voting System Using Android Application. 2014, IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 2, Issue 2, Apr-May, and ISSN: 2320 – 8791.

Monali S., Priyali P., Akshita G., Shivam M., An Efficient Mobile Voting System Scheme Based on Elliptic Curve Cryptography. 2015, International Journal of Scientific & Engineering Research, Volume 6, Issue 4, ISSN 2229-5518

NIST, "Voting Systems Performance and Test Standards", http://www.nist.gov/itl/vote/upload/Overview.pdf, (2009) (Access Date: 6.01.2016)

Ok K., Coskun V., Aydin M.N., Usability of Mobile Voting Using NFC Technology. 2010, International Conference on Software Engineering.

Ooijen P.M.A, Broekema A., Oudkerk M., An Interactive Image Based Voting System Using Windows Mobile Devices. 2011, International Journal of Medical Informatics Design and Implementation of I2VOTE – 80, pp-562-569.

OpenSSL FIPS 140, "Welcome to the OpenSSL Project", https://www.openssl.org/ (2015) (Access Date: 29.12.2015)

Support by Apple, "Frequently asked questions about iCloud Keychain", https://support.apple.com/en-us/HT204085 (2015) (Access Date: 21.12.2015)

Support by Apple, "About Touch ID security on iPhone and iPad", https://support.apple.com/en-us/HT204587 (2015) (Access Date: 21.12.2015)

Paillier P., Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. 199, EUROCRYPT. Springer. pp. 223–238.

Pandit P., Bhawar S., Desai M., Campus E-Voting for Android and Web Based Application. 2014, International Journal of Emerging Engineering Research and Technology Volume 2, Issue 7, PP 95-100.

Radicati S., "Mobile Statistics Report, 2014-2018.", http://www.radicati.com/wp/wp-content/uploads/2014/01/Mobile-Statistics-Report-2014-2018-Executive-Summary.pdf, (2014) (Access Date: 29.11.2015)

Rivcst R., Shamir A., and Adleman L., A Method for obtaining digital signatures and public-key cryptosystems. 1978, Communications of the ACM, vol. 21, no. 2, pp.120 -126.

Roonemaa H., and Lõugas H., "Putting the E in Estonia", http://review.gemalto.com/estonia/?/post/special-multimedia-feature-egov-leader-estonia, (2015) (Access Date: 03.01.2016)

Scammell R., "Internet voting a success in two European countries", http://www.eui.eu/News/2013/02-12-InternetvotingasuccessintwoEuropeancountries.aspx, (2013) (Access Date: 03.01.2016)

Sivagami V.M., Revathi N., Sumathi G., VoteGrid: A Mobile Ballot System For Decision Making In Grid Environment. 2011, Procedia Computer Science, Vol. 3, p287-291, 5p.

Stallings William, Cryptography and Network Security (4th Edition). 2005, p68

Supreme Electoral Council of Turkey, "Local Election Results." (2015) http://www.ysk.gov.tr/ysk/content/conn/YSKUCM/path/Contribution%20Folders/SecmenIslemleri/Secimler/2015MVES/96-A.pdf, (Access Date: 21.01.2016)

Thakur S., Olugbara O.O., Millham R., Wesso H.W., Sharif, M., Transforming voting paradigm-the shift from inline through online to mobile voting. 2014, Adaptive Science & Technology (ICAST), IEEE 6th International Conference, p1-7.

Ullah M., Iqbal U.A., Noor A., Nizamuddin., An efficient and secure mobile phone voting system. 2013 Eighth International Conference on Digital Information Management (ICDIM 2013), p332-336, 5p.

The Council of Europe,"Legal, Operational and Technical Standards for E-Voting", http://www.coe.int/t/DEMOCRACY/ELECTORAL-ASSISTANCE/themes/evoting/Rec-2004-11_en.pdf, (2004) (Access Date: 02.01.2016)

Vabariigi Valimiskomisjon,”Statistics about Internet Voting in Estonia”, http://www.vvk.ee/voting-methods-in-estonia/engindex/statistics (2015) (Access Date: 01.01.2016)

Velapure H., Rai S., Sharma S., Naiknavre P., Jadhav P., Bamane K., Android Based E-Voting. 2015, International Journal of Advance Foundation and Research in Computer (IJAFRC) Volume 2, Special Issue (NCRTIT 2015), ISSN 2348 – 4853.

Xun Y., Cerone P., Yanchun Z., Secure Electronic Voting for Mobile Communications. 2006, IEEE Vehicular Technology Conference, 2: 836-840.

Ying Q., Huafei Z., Somewhat Secure Mobile Electronic Voting Systems Based On the Cut-And-Choose Mechanism. 2009 International Conference on Computational Intelligence & Security, 2009, p446-450, 5p.

CURRICULUM VITAE

Murat Ödemiş was born in 1988, Izmir – Turkey. He holds a Bachelor of Science degree from Yaşar University, Computer Engineering Department after graduating in 2012. After starting his Master of Science (MSc.) education in Yaşar University, he started to study on mobile application development and security. His professional career started as a mobile application developer and project manager where his own company MobileCrea. He returned to career in academia after being accepted to be a research assistant in 2014 and he continues his job since then.