COURSE PROJECT

Authenticating Machines

Instructions

The goal of this project is for you to apply the concepts we cover in this course to an example system. You should continue working on the same system as you used in the previous course.

In **Part One**, you will identify how shared key cryptography could be used to secure your system.

In **Part Two**, you will identify which machines need to be authenticated and design an authentication scheme for your system.

In **Part Three**, you will identify where public key encryption could be used to secure your system and revise your authentication scheme to include public key components where appropriate.

Except as indicated, use this document to record all your project work and responses to any questions. At a minimum, you will need to turn in a digital copy of this document to your facilitator as part of your project completion. You may also have additional supporting documents that you will need to submit. Your facilitator will provide feedback to help you work through your findings.

Note: Though your work will only be seen by those grading the course and will not be used or shared outside the course, you should take care to obscure any information you feel might be of a sensitive or confidential nature.

Complete each project part as you progress through the course. Wait to submit the project until all parts are complete. Begin your course project by completing Part One below. A Submit Assignment *button can be found on the final course project assignment page online. Information about the grading rubric is available on any of the course project assignment pages online. Do not hesitate to contact your facilitator if you have any questions about the project.*

Part One

Shared Key Cryptography

The first step is to identify what roles shared key cryptography could play in your system. Recall that shared key cryptography can be used to encrypt files on disk, to encrypt messages in transmission, or to authenticate principals. You should refer back to the security goals you identified in the previous course and determine which goals can be met using shared key cryptography. You should then decide how shared key cryptography will be used to secure your system. Be sure to consider details such as who generates the keys, where the keys are stored, how the keys distributed, and whether/when/how keys will be refreshed.

A ***shared key cryptography*** is a technique which involves encrypting and decrypting shared messages using a key. For shared cryptography to work, both sender and recipient of a message must have the same shared key, which must be kept secret from everyone else. So, if E and D are the encryption and decryption function respectively, then

- E(key, message) = cipher-text
- D(key, cipher-text) = message

In the previous project, we studied an electronic voting system specifically online voting system and the following goals were identified:

1. The system shall have a back up power to prevent power outages and crushes.
2. The system shall prevent erasing of electoral roll in the the database.
3. The system shall detect resources and client requests that slow down the server.
4. The system shall prevent over voting.
5. The system shall detect unauthorised user activities and prevent tempering with user credentials.

Note that meeting the above goals means that the **security, accuracy, verifiability and anonymity** of the system will, at the very least, be guaranteed and, using shared key cryptography will go along way in meeting these goals.
Roles the shared key cryptography could play in the electronic voting system are but not limited to the following:

1. **Encrypting the storage devices that will store the cast ballots and voters' credentials**. This will prevent attackers from viewing the votes and making it difficult for them to alter voters' choices. It also prevents attackers from tempering with user credentials.
2. **Encrypting the votes when they are in transit from the clients electronic device to the storage device.** This prevents attackers from altering a vote's choice during transmission.
3. **Decrypting the shuffled votes after disconnecting the server and storage devices from the internet when voting ends.** Note that it is necessary to disconnect the server and storage devices from the internet to prevent attackers from tempering with data during decryption.

The roles listed above shows that the following goals can be met.

- The system shall prevent erasing of electoral roll in the the database.
- The system shall detect resources and client requests that slow down the server.
- The system shall prevent over voting.
- The system shall detect unauthorised user activities and prevent tempering with user credentials.

To decide how shared key cryptography will be used to secure the electronic voting system, we first need to understand the architecture of the system and the flow of information in the system.

The architecture of the electronic voting system contains a base server hosting the voting program that will manage voters information and cast ballots and voters devices (laptops, smart phones, PCs) that will be connected to the base server via the internet. We will assume that the electoral roll for all eligible voters has being compiled and stored in a database. The following is an algorithm to follow for a successful election:

The online voting algorithm

1. Input: provide a blank database
2. Output: election result

3. Call procedure: voting programs(blank database)
4. While the the election time is not over:
   - Call procedure: Authenticate voter(voter's credentials)
   - Call procedure: Cast vote(voter)
   - Call procedure: Encrypt vote (vote)
   - Call procedure: Save encrypted vote in database (vote)

From the above algorithm, we may wan to authenticate a voter before they can access the voting system. Thus we follow the **gold standard.**

The online voting system is a client -server application and in view of the above, it will be advisable to use the Kerberos Authentication Protocol. The **Kerberos** is a computer network security protocol that authenticates service requests between two or more trusted hosts across an the internet. It uses shared key cryptography and a trusted third party for authenticating client-server application and verifying users' identities.

When a voter(client) wants to vote, he/she will need to login to the server to access the online voting system. There is the need for authentication.

Note the following keywords

- **V - voter**
- **W - workstation**
- **EVS – electronic voting system**
- **KDC – key distridution centre**
- **TGT – ticket-grating-ticket**
- **TGS – ticket-granting-service**

Authenticate voter (with a user ID and password)

The voter (V) opens the login page of the voting system and enters a user ID. For simplicity, let us assume the user ID is V. The user ID is sent to a Workstation (W). The Workstation tells the Key Distribution Centre (KDC) that the voter needs a ticket-granting ticket (TGT). The KDC then generates a fresh session key denoted S_V for the voter. The KDC goes to a database which stores encrypted information about the voter's credentials. Note that the KDC uses the key k_{KDC} which is known only to the KDC for the encryption. In the database, the KDC looks up the

master key k_v corresponding to the voter's credentials. Once the voter's credentials are verified, the KDC generates a TGT:={V, S_V , exptime}k_{KDC} for the voter V. The workstation W receives the encrypted message {S_V, TGT}k_V from KDC. The workstation the requests a password from, generates k_V with the password and decrypts {S_V, TGT}k_V to obtain S_V, TGT. It then forgets k_V.

Request the Electronic Voting System—EVS -- from the server

After the voter is authenticated, the voter requests to vote. This request first goes to the workstation. The workstation requests a ticket-granting-service TGS for the voter from the KDC by sending EVS and TGT to KDC. The KDC then generates a fresh session key $k_{V,EVS}$ that the workstation can access EVS on behalf of the voter. The KDC sends the encrypted message {EVS, k_{VEVS}, TGS}S_V to the workstation W, where k_{EVS} is the shared key shared between KDC and EVS and TGS:={V, EVS, k_{VEVS}, exptime}k_{EVS}. Note that TGS is the credentials that the voter will use to access the EVS to cast his or her vote.

Cast vote (Accessing the Electronic Voting System, EVS)

The workstation W now decrypts the message {EVS, k_{VEVS}, TGS}S_V using the session key S_V and extracts EVS, k_{VEVS} and TGS:={V, EVS, k_{VEVS}, exptime}k_{EVS}. The workstation W then sends TGS, and the a fresh time {T} k_{VEVS} encrypted under the session key k_{VEVS} to EVS. The EVS then extracts V, EVS, kVEVS and exptime from TGS using the key k_{EVS} If the EVS grants the request, the voter can then vote. Tr grant access to vote, the EVS checks that the fresh time T is smaller that the expiry time exptime.

Note that the Workstation and the server will be physical secured at the Electoral commision building.

Part Two

Authenticating Machines

In this step, you will identify which machines in your system need to be authenticated. For each such machine, design an authentication protocol that will authenticate that machine to the appropriate principals.

In this part of the project, we look at authenticating machines in the electronic voting system (EVS) using a protocol based on the public key cryptography.

Machine authentication is the process of determining whether a machine can interact with other machines to exchange information on both wired and wireless.

We learned how to use **digital signatures** to implement machine authentication through a network of federated Certificate Authorities (CAs).

The machines in the EVS include the following

- The electronic devices (ED) such as laptops, smartphones, tablets and desktop PCs used by voters- These devices are connected to the internet in order to access the electronic voting system.
- The server hosting the electronic voting system.

Denote the voter's device by VD and denote the server hosting the electronic voting system by EVSS. Let the Certificate Authority be CA.

Proposed authentication protocol

1. VD sends a request to EVSS to access the electronic voting system.
2. EVSS sends the public certificate of EVSS signed by CA to VD.
3. VD verifies the public certificate of EVSS by using the CA's public key.
4. The EVSS requests the public certificate of VD which is signed by CA.
5. VD signs an nonce using the private key of VD. VD sends the signed nonce together with

the public certificate of VD to EVSS.

6. EVSS retrieves the signed nonce and public certificate of VD. EVSS uses the public key of VD to verify that the nonce was signed by VD.
7. EVSS checks that the VD's public certificate has not expired and that it has not been revoked.
8. VD can access EVSS if all verifications have passed and there is a match of VD identity on EVSS.

Part Three

Public Key Cryptography

The final step is to identify what roles public key cryptography could play in your system. You should refer back to the security goals you identified in the previous course and determine which goals can be met using public key cryptography. You should then decide how public key cryptography will be used to secure your system. Be sure to consider details such as who generates the keys, where the keys are stored, and how principals discover public keys.

Public key cryptography involves a pair of keys known as a public key and a private key. This pair is associated with a user that needs to authenticate its identity electronically or to sign or encrypt data. Each public key is published in a certificate authority server and the corresponding private key is kept secret by the user.

Roles public key cryptography could play in the electronic voting system are

1. Securing the cast ballots
2. Keeping voter's identity anonymous.
3. Providing a way to verify vote accuracy.

The security goal from the previous project are

1. The system shall have a back up power to prevent power outages and crushes.
2. The system shall prevent erasing of electoral roll in the the database.
3. The system shall detect resources and client requests that slow down the server.
4. The system shall prevent over voting.
5. The system shall detect unauthorised user activities and prevent tempering with user credentials.

The goals that could be achieved are

1. The system shall detect unauthorised user activities and prevent tempering with user credentials.
2. The system shall prevent over voting.

Assume that the electoral commission has collected data (credentials) of eligible voters for identification and stored them in a database on the server.

The eligible voter is required to use their credentials to gain access to the electronic voting system server. Once the voter is authenticated and gains access to the electronic voting webpage, the electronic voting server will generate a public key K_{EVS} and a private key k_{EvS} pair. The public key will be sent to the voter and the private key will be store on another secure system different from the election server. The private key will be used for decrypting the encrypted votes. We assume the public key will be sent to the voter via a text message on their mobile phones.

Once a voter has received the election public key from the election server, he or she is required to vote immediately. The voter's electronic device also generates a public key K_V and a private key k_V which will be stored on the voter's device and kept a secret. The voter's private key will be used to generate a digital signature in order to sign the vote and also keep the voter anonymous. The election public key is then used by the voter to encrypt the ballot, the digital signature and other information from the voter. The voters public key will be used by the election server to check the validity of the vote.

REFERENCES

1. https://www.comparitech.com/blog/information-security/cryptography-secure-electronic-voting-systems/
2. Course Materials

To submit this assignment, please refer to the instructions in the course.