CLAIRVOYANT

design engineer deliver

Spring Security Basics

Team
Clairvoyant India Pvt. Ltd.

CLAIRVOYANT

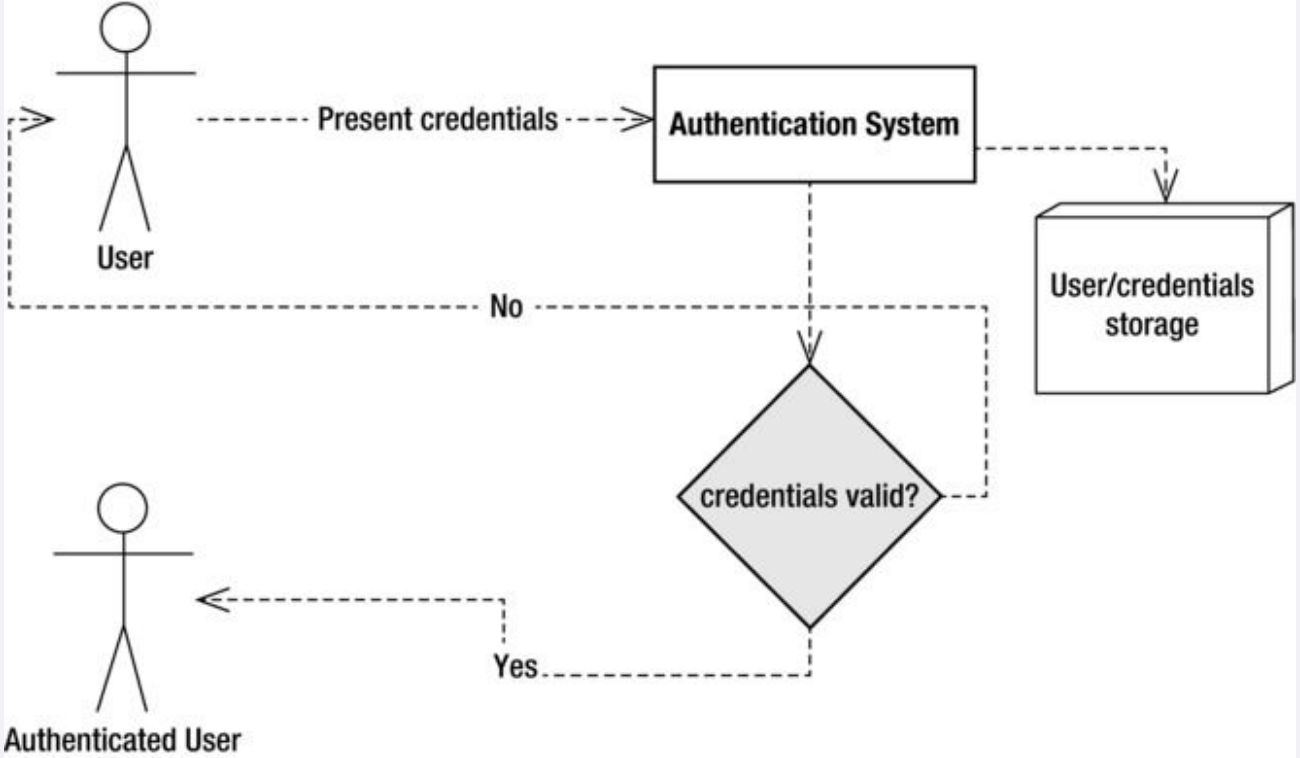
Agenda

- Security Basics
- Why spring security?
- Spring Security - architecture
- Code walkthrough
- Authentication
- Authorization
- Testing
- Spring security OAuth 2.0 Login

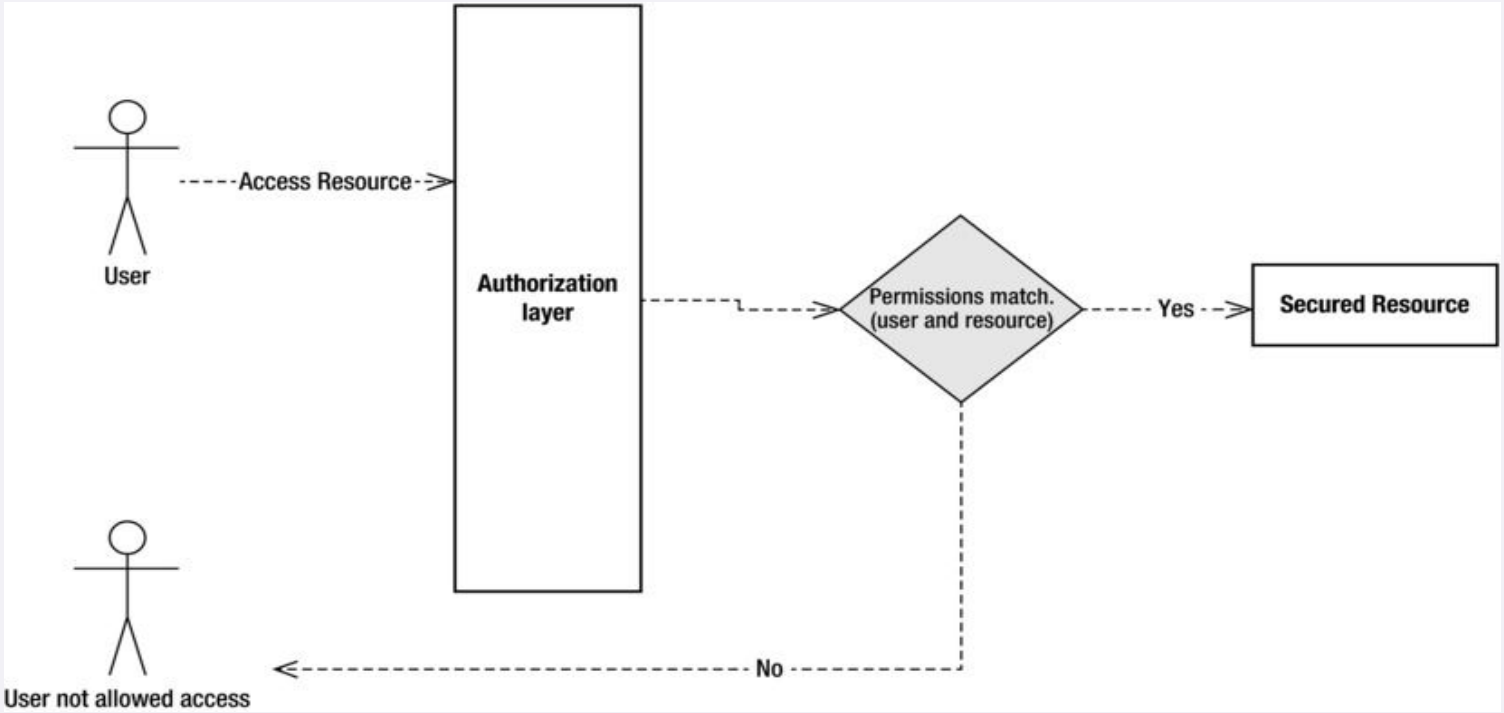
Concepts

- User
- Credentials
- Role
- Resource
- Permissions

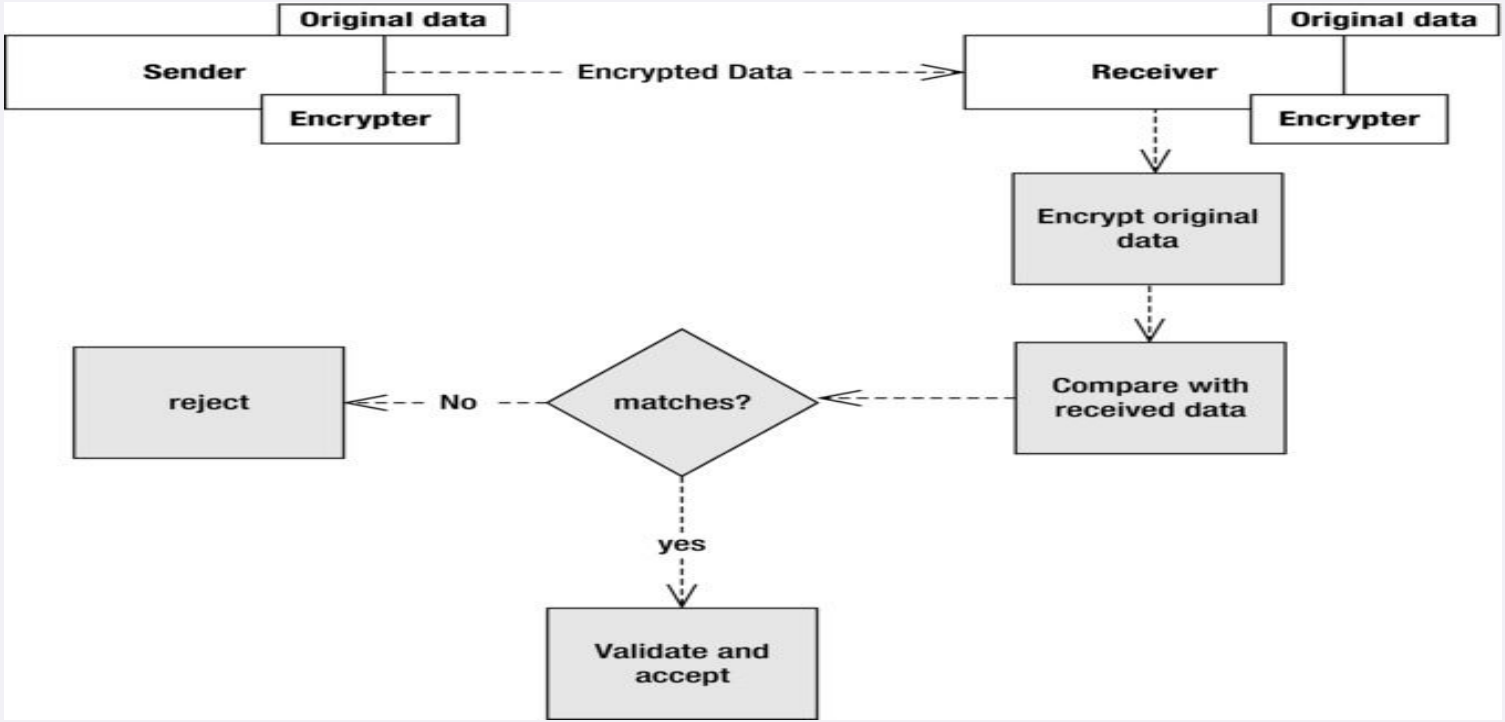
Authentication



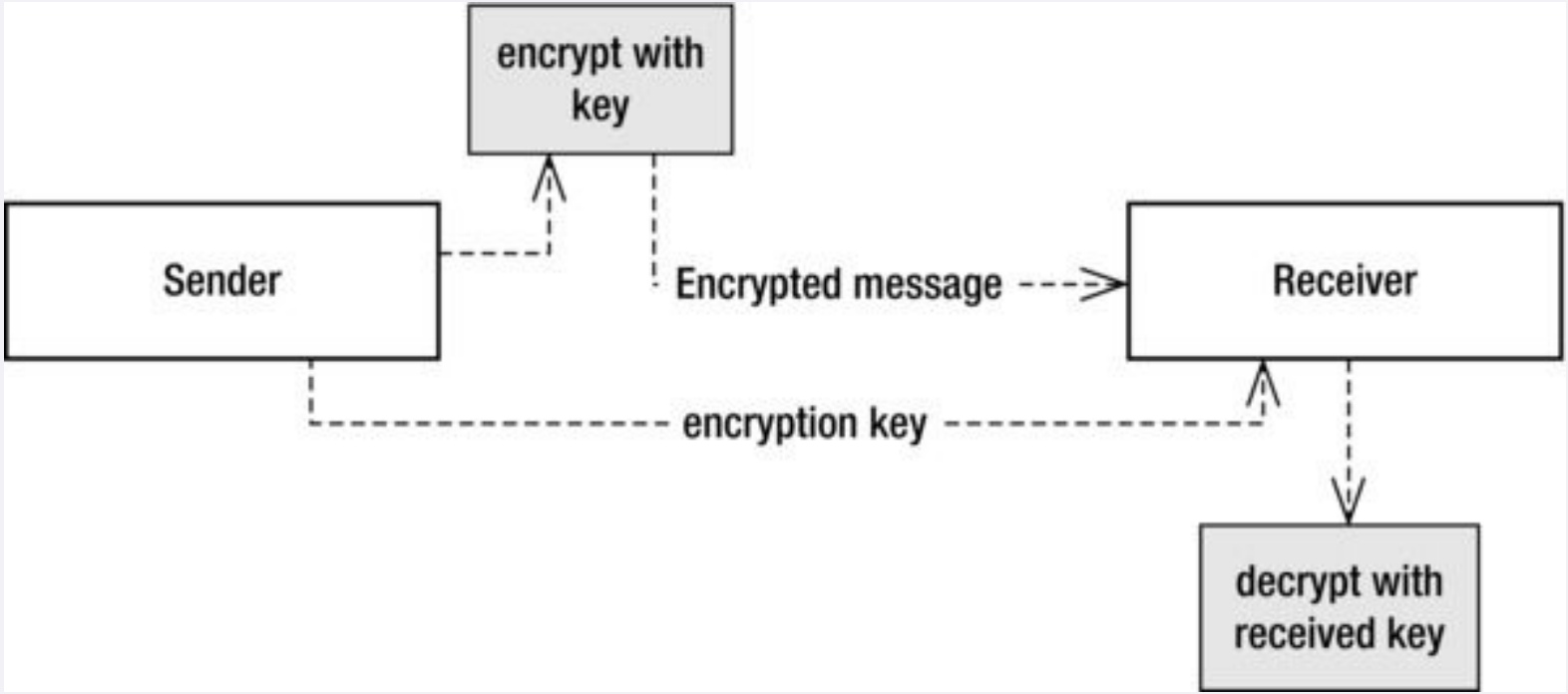
Authorization and ACLs



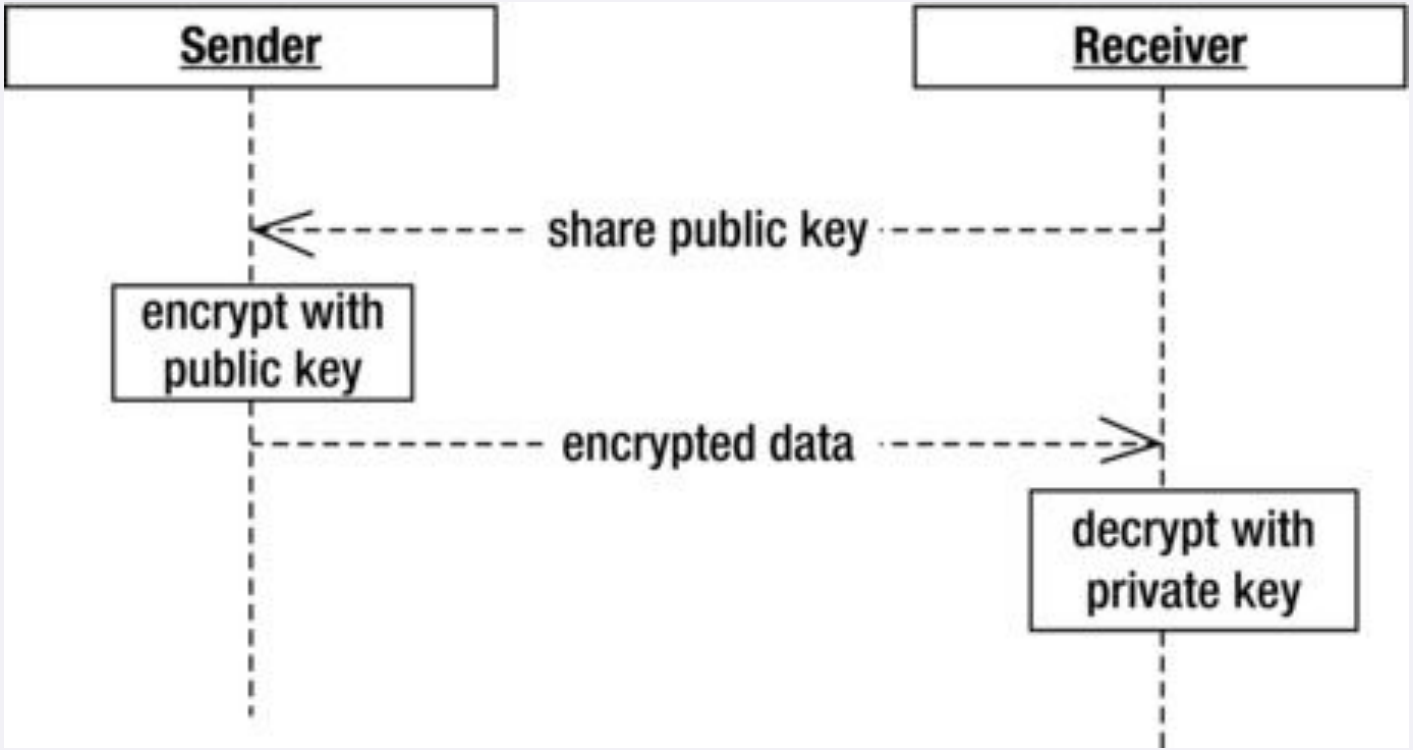
One-way encryption



Symmetric encryption



Public key cryptography



Java Options for Security

- Java Cryptography Architecture (JCA)
- Java Cryptographic Extensions (JCE)
- Java Certification Path API (CertPath)
- Java Secure Socket Extension (JSSE)
- Java Authentication and Authorization Service (JAAS)

Why spring security?

- It offers layered security services
- Comprehensive and extensible support for both Authentication and Authorization
- Protection against attacks like session fixation, clickjacking, cross site request forgery, etc
- Servlet API integration
- Optional integration with Spring Web MVC

Architecture

Code walkthrough

- Roles: Admin, Mgr, User
- Only admin can get details of all employees
- Manager can add employee details only if he/she is reporting to him/her
- Employee self/manager reporting to or admin can get details of employee

OAuth 2.0 integration

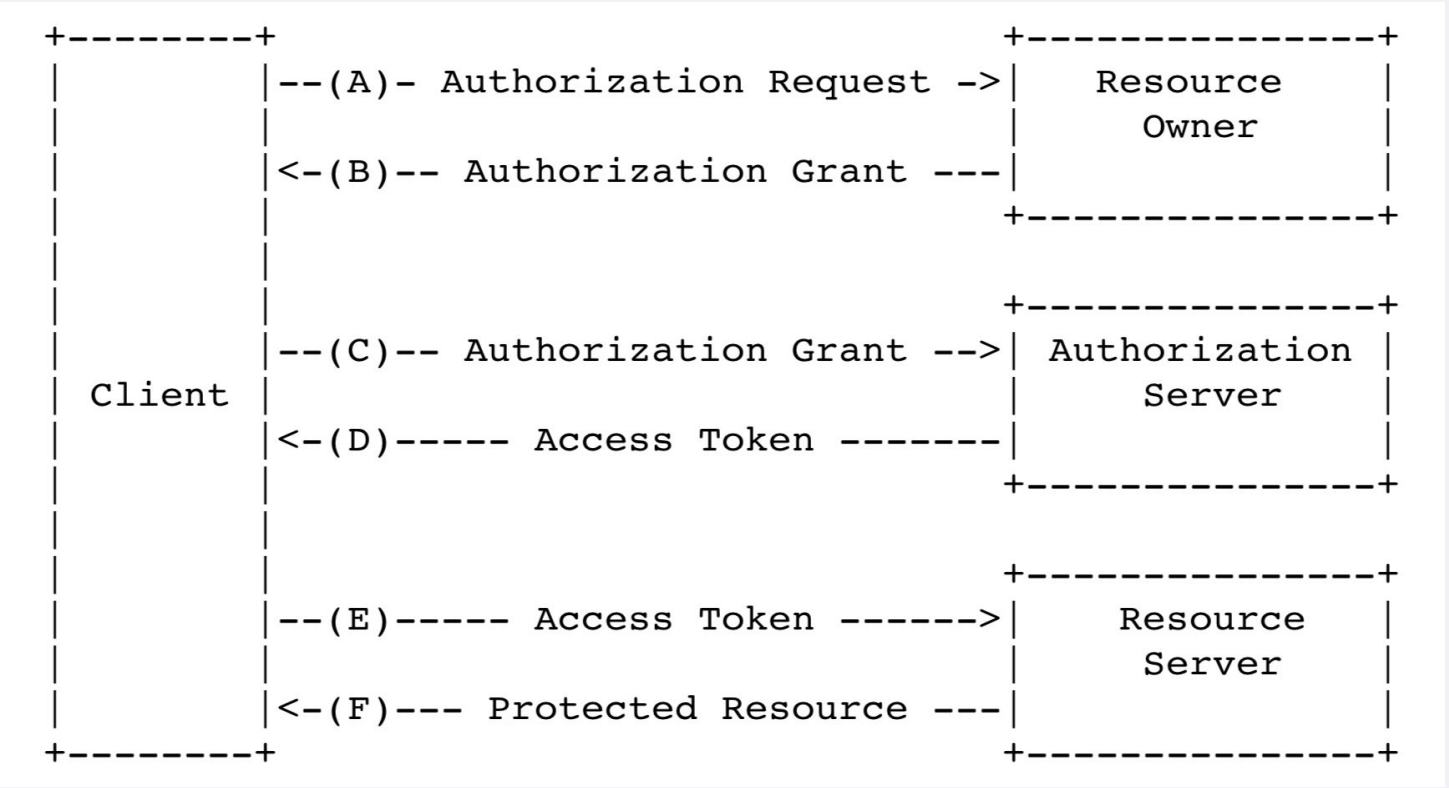
OAuth 2.0

- An authorization framework that enables user to applications to obtain limited access to user accounts on an HTTP service, such as Facebook, GitHub and LinkedIn.
- It works by delegating user authentication to the service that hosts the user account, and authorizing third-party applications to access the user account.

OAuth 2.0: Roles

- Resource Owner
- Resource Server
- Client
- Authorization Server

Protocol Flow



Authorization Grant

An authorization grant is a credential representing the resource owner's authorization (to access its protected resources) used by the client to obtain an access token.

- Authorization Code
- Implicit
- Resource Owner Password credentials
- Client Credentials

Authorization Grant- Authorization Code

- The application opens a browser to send the user to the OAuth server
- The user sees the authorization prompt and approves the app's request
- The user is redirected back to the application with an authorization code in the query string
- The application exchanges the authorization code for an access token

Authorization Grant- Implicit

- The application opens a browser to send the user to the OAuth server
- The user sees the authorization prompt and approves the app's request
- The user is redirected back to the application with an access token in the URL fragment.

Authorization Grant- Client Credentials

- The client credentials can be used as an authorization grant when the authorization scope is limited to the protected resources under the control of the client.

Authorization Grant- Resource Owner Password Credentials

- The resource owner password credentials (i.e., username and password) can be used directly as an authorization grant to obtain an access token.
- Happens for only one request.

Questions ???

References

- Pro Spring Security - by Carlo Scarioni
- https://www.owasp.org
- https://docs.spring.io/spring-security/site/docs/5.0.6.RELEASE/reference/htmlsingle/
- https://www.youtube.com/watch?v=1pegQ4Bj_aM&t=1s - GOTO 2017
- Building Layers of Defense with Spring Security-Joris Kuipers
- https://spring.io/guides/tutorials/spring-security-and-angular-js/
- https://github.com/nitinkasat/meetups.git
- https://www.youtube.com/watch?v=8rnOsF3RVQc