Digital Signature Verification System

CSC 675-775-01 Summer 2025

Matthew Martinez
Student ID: 923707132
GitHub: matthewmartinez-java

Milestone	Date Submitted
M2	07/01/2025
M1	06/16/2025

Table of Contents

Project Description

Increasing dependence on digital communication and electronic documents, in a remote work setting, has made verification of document authenticity more necessary than ever. In sectors such as healthcare and legal services, unauthorized tampering or impersonations can result in serious consequences like data breaches and fraud. The motivation behind this project is to develop a safe Digital Signature Verification System that assists organizations in securing their workflow of electronic documents by providing secure, auditable digital signature verification.

At a high level, this database system will serve as the backbone of a digital signature verification service. It will store and manage user information, digital certificates, signed documents, audit logs, and signature validation records. The system will allow users to upload documents with attached digital signatures and verify their authenticity in real time. Factors like mismatches, expirations, or evidence of document tampering will result in a validation failure, and the system will automatically flag the issue and log it for further examination.

This database system will contain various unique features on the basis of a cybersecurity mindset and approach. Most importantly, it will:

- Support Public Key Infrastructure (PKI) integration for digital certificates/keys
- Offer role-based access control (RBAC) to protect sensitive data and operations
- Include timestamping and event logging for auditing and analysis

These features build upon existing systems by offering finer-grained access controls, transparency of the verification process, and detailed logging, which is common practice in a cybersecurity environment.

Two existing products that would benefit from using this database system are:

1. *DocuSign* - DocuSign currently has electronic signatures, but it would benefit from a customizable, internally-hosted database approach that allows for greater control over how verification events are stored and accessed. For example, firms with demanding data residency would be able to put this system in place to meet compliance without depending on a third party SaaS platform.

2. *Adobe Acrobat Sign* - Acrobat Sign can expand its current PKI functionality with stronger transparent audit trails and RBAC security by using this database system as a backend module. This is especially beneficial in industries like healthcare where secure logging is required for regulatory compliance.

Functional Database Requirements

1. User

1.1. A user shall create at most one account
1.2. A user shall have a unique user ID
1.3. A user shall be able to register using an email and password
1.4. A user shall have a role assigned via role entity
1.5. A user shall be linked to multiple sessions
1.6. A user without an account shall not be allowed to verify documents
1.7. A user shall verify their email address before gaining access to the system
1.8. A user shall have the ability to reset passwords via a secure token-based system
1.9. A user shall be locked out after 5 failed login attempts within 15 minutes
1.10. A user shall be alerted via notification if account activity is detected on a new device

2. Registered User

2.1. A registered user is also a user
2.2. A registered user shall have exactly one account
2.3. A registered user shall be able to upload and sign multiple documents
2.4. A registered user shall view verification history
2.5. A registered user shall be able to revoke their own signatures
2.6. A registered user shall be assigned to one organization

3. Admin

3.1. An admin is also a registered user
3.2. An admin shall manage all access control entities
3.3. An admin shall revoke certificates and monitor audit logs
3.4. An admin shall promote or demote other users to admin level
3.5. An admin shall trigger emergency shutdown or lockdown of the system
3.6. An admin shall be able to create and archive organization profiles

4. Role

4.1. A role shall be linked to many users
4.2. A role shall have a unique role ID

4.3. A role shall define access permissions
4.4. A role shall be customizable by the admin
4.5. A role shall include a descriptive title

5. Account

5.1. An account shall be associated with one user only
5.2. An account shall have a unique account ID
5.3. An account shall record the creation timestamp
5.4. An account shall include a status field
5.5. An account shall be updated only by the user or an admin

6. Digital Certificate

6.1. A certificate shall be assigned to one user
6.2. A certificate shall have an expiration and issue date
6.3. A certificate shall be issued by one certificate authority
6.4. A certificate shall include the public key
6.5. A certificate shall be revocable by an admin
6.6. A certificate shall be linked to multiple signatures
6.7. A certificate shall be validated against Certificate Revocation Lists
6.8. A digital certificate shall be exportable in PEM format by the owning user
6.9. A certificate shall include a fingerprint (SHA-256) for integrity verification

7. Certificate Authority

7.1. A certificate authority shall issue many certificates
7.2. A certificate authority shall have a unique ID and public key
7.3. A certificate authority shall have a verified status
7.4. A certificate authority shall be associated with an issuing organization
7.5. A certificate authority shall store revocation timestamps for invalid certificates

8. Signature

8.1. A signature shall be linked to one document
8.2. A signature shall be generated using a specific certificate
8.3. A signature shall store the hash of signed content
8.4. A signature shall include a timestamp and status
8.5. A signature shall be associated with a user
8.6. A signature may be a part of a recursive signature chain

9. Document

9.1. A document shall have a unique ID
9.2. A document shall be uploaded by one user
9.3. A document shall have one or more signatures

9.4. A document may be shared with multiple users
9.5. A document shall link to verification events
9.6. A document shall contain metadata

10. Verification Event

10.1. A verification event shall be triggered by a user
10.2. A verification event shall include status
10.3. A verification event shall generate an audit log
10.4. A verification event shall record document ID and timestamp
10.5. A verification event may trigger a notification

11. Audit Log

11.1. An audit log shall record every verification event
11.2. An audit log shall include timestamp, user ID, IP address, and result
11.3. An audit log shall be immutable once written
11.4. An audit log shall support compliance inspection and export
11.5. An audit log may be linked to a signature revocation event
11.6. An audit log shall support encrypted hashing to detect tampering
11.7. An audit log shall be linked to the specific action or API call that triggered it
11.8. An audit log shall differentiate between successful and failed actions

12. Signature Revocation

12.1. A revocation shall be linked to one signature
12.2. A revocation shall store the reason for revocation
12.3. A revocation shall be time stamped and recorded in audit logs
12.4. A revocation shall be performed by users with revocation privileges
12.5. A revocation shall invalidate the associated certificate in the database

13. Access Control Entry

13.1. An access entry shall define permissions
13.2. An access entry shall be assigned to one user and one document
13.3. An access entry shall be editable only by an admin
13.4. An access entry shall define an access start and end period

14. Public Key

14.1. A public key shall be part of one certificate
14.2. A public key shall be uniquely stored using encrypted fields
14.3. A public key shall be used during the verification process
14.4. A public key shall support validation using hash comparison

15. Private Key

15.1. A private key shall be stored using industry-standard encryption

15.2. A private key shall never be directly queried
15.3. A private key shall be rotated on a schedule
15.4. A private key shall be associated with only one user
15.5. A private key shall be inaccessible through any API endpoints
15.6. A private key shall require MFA authentication for re-issue after deletion

16. Notification

16.1. A notification shall be generated for failed verification events
16.2. A notification shall be linked to one user and one document
16.3. A notification shall include content, timestamp, and urgency level
16.4. A notification shall support read/unread status tracking

17. Organization

17.1. An organization shall have a unique ID and name
17.2. An organization shall contain multiple users
17.3. An organization shall be associated with documents and certificate authorities
17.4. An organization shall have a designated admin contact

18. Session

18.1. A session shall be started at every login event
18.2. A session shall track login time, logout time, and IP address
18.3. A session shall be associated with one user
18.4. A session shall be automatically expired after timeout or logout

19. Hash Record

19.1. A hash record shall store a cryptographic hash of each document
19.2. A hash record shall be linked to a document ID and verification result
19.3. A hash record shall be used to detect tampering
19.4. A hash record shall be compared during signature verification
19.5. A hash record shall be calculated using SHA-256 or stronger algorithms
19.6. A hash record shall be immutable after the document is signed
19.7. A hash record mismatch during verification shall trigger an audit log alert

20. Device

20.1. A device shall be registered to a user for signature activity
20.2. A device shall store device fingerprinting metadata
20.3. A device shall be logged during session creation
20.4. A device shall store trust level status
20.5. A device shall require re-verification if flagged or inactive for 30+ days
20.6. A device shall be eligible for two-factor authentication binding
20.7. A device shall be deletable only by the user or an admin

21. IP Address Log

21.1. An IP address log shall track every user interaction
21.2. An IP log shall link to a session and verification event
21.3. An IP log shall be used for geo-location analysis
21.4. An IP log shall support compliance with access policies

Non-Functional Database Requirements

1. Performance

1.1. The database system shall support concurrent transactions for multiple users without race conditions
1.2. The system shall return query results for digital signature validation in under 2 seconds 95% of the time
1.3. Indexes shall be created on frequently queried fields such as ‘user_id’, ‘document_id’, and ‘certificate_id’ to optimize search performance
1.4. The system shall process and log at least 100 verification requests per minute during peak hours

2. Security

2.1. Only hashed and salted passwords shall be stored using secure algorithms like bcrypt
2.2. All connections to the database shall use TLS 1.2 or higher for encryption in transit
2.3. The database shall implement role-based access control (RBAC) at the query and table level
2.4. The system shall log all failed authentication attempts with timestamps and originating IP

3. Scalability

3.1. The database system shall support horizontal scaling of read replicas to manage growing query loads
3.2. The system shall be designed to handle at least a 10x increase in users and stored documents without performance degradation
3.3. Database tables such as ‘audit_log’, ‘session’, and ‘verification_event’ shall support partitioning by time or user ID for scalability

4. Capability

4.1. The system shall support document verification against both self-signed and certificate authority-issued certificates
4.2. The system shall be able to integrate with third-party APIs for certificate validation

4.3. The database shall support time-based queries to enable efficient log and session analysis

5. Environmental

5.1. The database system shall run on a Linux-based cloud environment (Ubuntu on AWS EC2)
5.2. The system shall consume less than 500 MB of RAM under normal load conditions
5.3. The system shall be deployable in a containerized environment using Docker
5.4. The system shall operate reliably in environments with intermittent internet by caching critical data locally

6. Coding Standard

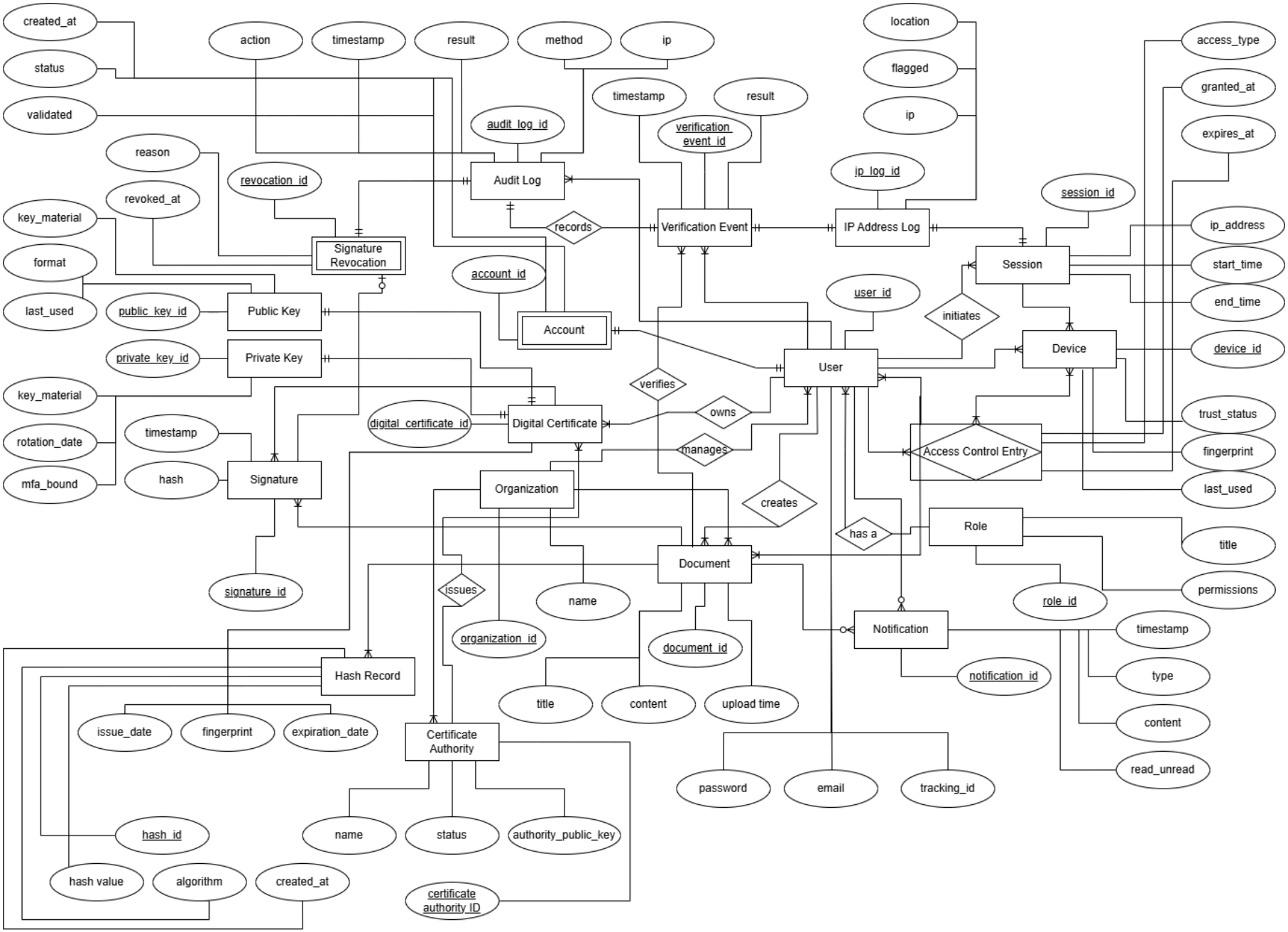
6.1. SQL queries and database schema scripts shall follow the ANSI SQL standard
6.2. All code shall follow naming conventions for clarity
6.3. All queries shall be parameterized to prevent SQL injection
6.4. Database triggers and stored procedures shall be version controlled and commented clearly

7. Storage

7.1. The database system shall support persistent storage for all core entities
7.2. Each table shall have a disk quota policy and alert administrators when 80% of capacity is reached
7.3. The system shall support automatic archival of records older than 5 years
7.4. Uploaded documents shall be stored in a dedicated, encrypted file system or object storage layer

8. Privacy

8.1. Personally identifiable information (PII) such as email and IP address shall be stored encrypted at rest
8.2. The system shall restrict access to sensitive fields based on user roles
8.3. Privacy actions such as document sharing or verification attempts shall require user consent where applicable


created_at
status
validated
action
timestamp
result
method
ip
audit_log_id
Audit Log
reason
revocation_id
revoked_at
Signature Revocation
key_material
format
last_used
public_key_id
Public Key
private_key_id
Private Key
key_material
rotation_date
mfa_bound
timestamp
hash
Signature
signature_id
records
Verification Event
timestamp
result
verification_event_id
account_id
Account
digital_certificate_id
Digital Certificate
verifies
owns
manages
Organization
issues
name
organization_id
Hash Record
hash_id
hash value
algorithm
created_at
issue_date
fingerprint
expiration_date
Certificate Authority
name
status
authority_public_key
certificate authority ID
location
flagged
ip
ip_log_id
IP Address Log
User
user_id
initiates
Session
session_id
ip_address
start_time
end_time
Device
device_id
access_type
granted_at
expires_at
trust_status
fingerprint
last_used
Access Control Entry
creates
has a
Role
title
permissions
role_id
Document
document_id
title
content
upload time
Notification
timestamp
type
content
read_unread
notification_id
password
email
tracking_id

Entity Set Description

1. **User (Strong)**
    a. Key Attributes
        i. user_id: numeric, primary key
    b. Non-Key Attributes
        i. email: alphanumeric, unique key
        ii. password: alphanumeric, hashed
        iii. tracking_id: numeric, unique key
        iv. role_id: foreign key (references Role)
        v. organization_id: numeric, foreign key (references Organization)

2. **Account (Weak)**
    a. Key Attributes
        i. account_id: numeric, primary key
        ii. user_id: numeric, foreign key
    b. Non-Key Attributes
        i. created_at: datetime
        ii. status: alphanumeric (active, suspended, etc.)
        iii. validated: boolean

3. **Role (Strong)**
    a. Key Attributes
        i. role_id: numeric, primary key
    b. Non-Key Attributes
        i. title: alphanumeric
        ii. permissions: alphanumeric

4. **Document (Strong)**
    a. Key Attributes
        i. document_id: numeric, primary key
    b. Non-Key Attributes
        i. title: alphanumeric
        ii. content: text (encrypted)
        iii. upload_time: datetime
        iv. organization_id: numeric, foreign key

5. **Signature (Strong)**
    a. Key Attributes
        i. signature_id: numeric, primary key
    b. Non-Key Attributes

i. hash: alphanumeric
ii. timestamp: datetime
iii. digital_certificate_id: numeric, foreign key

6. **Signature Revocation(Weak)**
    a. Key Attributes
        i. revocation_id: numeric, primary key
    b. Non-Key Attributes
        i. reason: text
        ii. revoked_at: datetime
        iii. signature_id: numeric, foreign key (references signature)

7. **Digital Certificate (Strong)**
    a. Key Attributes
        i. digital_certificate_id: numeric, primary key
    b. Non-Key Attributes
        i. issue_date: datetime
        ii. expiration_date: datetime
        iii. fingerprint: alphanumeric
        iv. user_id: numeric, foreign key

8. **Access Control Entry (Associative)**
    a. Key Attributes
        i. user_id: numeric, primary key and foreign key
        ii. document_id: numeric, primary key and foreign key
    b. Non-Key Attributes
        i. access_type: alphanumeric (read, write, revoke)
        ii. granted_at: datetime
        iii. expires_at: datetime

9. **Session (Strong)**
    a. Key Attributes
        i. session_id: numeric, primary key
    b. Non-Key Attributes
        i. user_id: numeric, foreign key (references User)
        ii. start_time: datetime
        iii. end_time: datetime
        iv. ip_address: alphanumeric

10. **Device (Strong)**
    a. Key Attributes

i. device_id: numeric, primary key

b. Non-Key Attributes

i. user_id: numeric, foreign key (references User)

ii. trust_status: alphanumeric (trusted, untrusted, flagged)

iii. fingerprint: alphanumeric

iv. last_used: datetime

11. Notification (Strong)

a. Key Attributes

i. notification_id: numeric, primary key

b. Non-Key Attributes

i. user_id: numeric, foreign key (references User)

ii. document_id: numeric, foreign key (references Document)

iii. timestamp: datetime

iv. type: alphanumeric (error, alert, success)

v. content: text

vi. read_unread: boolean

12. Audit Log (Strong)

a. Key Attributes

i. audit_log_id: numeric, primary key

b. Non-Key Attributes

i. user_id: numeric, foreign key (references User)

ii. verification_event_id: numeric, foreign key (references VerificationEvent)

iii. action: alphanumeric

iv. timestamp: datetime

v. result: alphanumeric (success, failure)

vi. method: alphanumeric (manual, auto)

vii. ip: alphanumeric

13. Verification Event (Strong)

a. Key Attributes

i. verification_event_id: numeric, primary key

b. Non-Key Attributes

i. user_id: numeric, foreign key (references User)

ii. document_id: numeric, foreign key (references Document)

iii. timestamp: datetime

iv. result: alphanumeric

v. audit_log_id: numeric, foreign key (references Audit Log)

14. IP Address Log (Strong)

a. Key Attributes
    i. ip_log_id: numeric, primary key
b. Non-Key Attributes
    i. session_id: numeric, foreign key (references Session)
    ii. verification_event_id: numeric, foreign key (references VerificationEvent)
    iii. ip: alphanumeric
    iv. location: alphanumeric
    v. flagged: boolean

15. Public Key (Strong)

a. Key Attributes
    i. public_key_id: numeric, primary key
b. Non-Key Attributes
    i. digital_certificate_id: numeric, foreign key (references Digital Certificate)
    ii. key_material: alphanumeric
    iii. format: alphanumeric
    iv. last_used: datetime

16. Private Key (Strong)

a. Key Attributes
    i. private_key_id: numeric, primary key
b. Non-Key Attributes
    i. digital_certificate_id: numeric, foreign key (references Digital Certificate)
    ii. key_material: alphanumeric (encrypted)
    iii. rotation_date: datetime
    iv. mfa_bound: boolean

17. Organization (Strong)

a. Key Attributes
    i. organization_id: numeric, primary key
b. Non-Key Attributes
    i. name: alphanumeric

18. Hash Record (Strong)

a. Key Attributes
    i. hash_id: numeric, primary key
b. Non-Key Attributes
    i. document_id: foreign key (references Document)
    ii. hash_value: alphanumeric
    iii. algorithm: alphanumeric (SHA-256, SHA-3, etc.)
    iv. created_at: datetime

19. Certificate Authority (Strong)

a. Key Attributes
    i. certificate_authority_id: numeric, primary key
b. Non-Key Attributes
    i. name: alphanumeric
    ii. authority_public_key: alphanumeric
    iii. organization_id: numeric, foreign key (references Organization)
    iv. status: alphanumeric (active, revoked)

Database Normalization

1. CREATE TABLE User (
   user_id INT PRIMARY KEY,
   email VARCHAR(50) UNIQUE NOT NULL,
   password VARCHAR(50) NOT NULL,
   tracking_id INT UNIQUE NOT NULL,,
   role_id INT,
   organization_id INT,
   FOREIGN KEY (role_id) REFERENCES Role (role_id)
   FOREIGN KEY (organization_id) REFERENCES Organization (organization_id)
   );
   Already passes 3NF

2. CREATE TABLE AccessControlEntry (
   user_id INT,
   document_id INT,
   access_type VARCHAR(50),
   granted_at DATETIME,
   expires_at DATETIME,
   PRIMARY KEY (user_id, document_id),
   FOREIGN KEY (user_id) REFERENCES User(user_id),
   FOREIGN KEY (document_id) REFERENCES Document(document_id)
   );

   Composite PK was the issue so I verified that access_type, granted_at, and expires_at are tied to the full user_id and document_pair.

3. CREATE TABLE SignatureRevocation (
   revocation_id INT PRIMARY KEY,
   reason TEXT NOT NULL,
   revoked_at DATETIME NOT NULL,
   signature_id INT,
   FOREIGN KEY (signature_id) REFERENCES Signature(signature_id)
   );

   Partial dependency in signature_id was part of composite PK so I made revocation_id the only PK and used signature_id as a foreign key only

4. CREATE TABLE Session (
    session_id INT PRIMARY KEY,
    user_id INT,
    start_time DATETIME,
    end_time DATETIME,
    ip_address VARCHAR(255),
    FOREIGN KEY (user_id) REFERENCES User(user_id)
);

   Already passes 3NF

5. CREATE TABLE Device (
    device_id INT PRIMARY KEY,
    user_id INT,
    trust_status VARCHAR(50),
    fingerprint VARCHAR(255),
    last_used DATETIME,
    FOREIGN KEY (user_id) REFERENCES User(user_id)
);

   Made sure trust_status was device specific and not influenced by user_id

6. CREATE TABLE Notification (
    notification_id INT PRIMARY KEY,
    user_id INT,
    document_id INT,
    timestamp DATETIME,
    type VARCHAR(50),
    content TEXT,
    read_unread BOOLEAN,
    FOREIGN KEY (user_id) REFERENCES User(user_id),
    FOREIGN KEY (document_id) REFERENCES Document(document_id)
);

   Already passes 3NF

7. CREATE TABLE AuditLog (
    audit_log_id INT PRIMARY KEY,
    user_id INT,
    verification_event_id INT,
    action VARCHAR(255),
    timestamp DATETIME,

result VARCHAR(50),
method VARCHAR(50),
ip VARCHAR(255),
FOREIGN KEY (user_id) REFERENCES User(user_id),
FOREIGN KEY (verification_event_id) REFERENCES
VerificationEvent(verification_event_id)
);

Already passes 3NF

8. CREATE TABLE VerificationEvent (
verification_event_id INT PRIMARY KEY,
user_id INT,
document_id INT,
timestamp DATETIME,
result VARCHAR(50),
audit_log_id INT,
FOREIGN KEY (user_id) REFERENCES User(user_id),
FOREIGN KEY (document_id) REFERENCES Document(document_id),
FOREIGN KEY (audit_log_id) REFERENCES AuditLog(audit_log_id)
);

Already passes 3NF

9. CREATE TABLE PublicKey (
public_key_id INT PRIMARY KEY,
digital_certificate_id INT,
key_material VARCHAR(255),
format VARCHAR(50),
last_used DATETIME,
FOREIGN KEY (digital_certificate_id) REFERENCES
DigitalCertificate(digital_certificate_id)
);

Possible transitive dependency with digital_cerification_id and user_id. Verified all attributes depend only on public_key_id.

10. CREATE TABLE PrivateKey (
private_key_id INT PRIMARY KEY,
digital_certificate_id INT,
key_material VARCHAR(255),
rotation_date DATETIME,

```
    mfa_bound BOOLEAN,
    FOREIGN KEY (digital_certificate_id) REFERENCES
    DigitalCertificate(digital_certificate_id)
);
```

Possible transitive dependency with digital_cerification_id and user_id. Verified all attributes depend only on private_key_id.

11.
```
CREATE TABLE IPAddressLog (
    ip_log_id INT PRIMARY KEY,
    session_id INT,
    verification_event_id INT,
    ip VARCHAR(255),
    location VARCHAR(255),
    flagged BOOLEAN,
    FOREIGN KEY (session_id) REFERENCES Session(session_id),
    FOREIGN KEY (verification_event_id) REFERENCES
    VerificationEvent(verification_event_id)
);
```

Possible transitive dependency through session_id and verification_event_id.

12.
```
CREATE TABLE HashRecord (
    hash_id INT PRIMARY KEY,
    document_id INT,
    hash_value VARCHAR(255),
    algorithm VARCHAR(50),
    created_at DATETIME,
    FOREIGN KEY (document_id) REFERENCES Document(document_id)
);
```

Already passes 3NF

13.
```
CREATE TABLE Organization (
    organization_id INT PRIMARY KEY,
    name VARCHAR(255) NOT NULL
);
```

Already passes 3NF

14.
```
CREATE TABLE CertificateAuthority (
    certificate_authority_id INT PRIMARY KEY,
```

```
    name VARCHAR(255),
    authority_public_key VARCHAR(255),
    organization_id INT,
    status VARCHAR(50),
    FOREIGN KEY (organization_id) REFERENCES Organization(organization_id)
);
```

Already passes 3NF

15. CREATE TABLE Document (

```
    document_id INT PRIMARY KEY,
    title VARCHAR(255),
    content TEXT,
    upload_time DATETIME,
    organization_id INT,
    FOREIGN KEY (organization_id) REFERENCES Organization(organization_id)
);
```

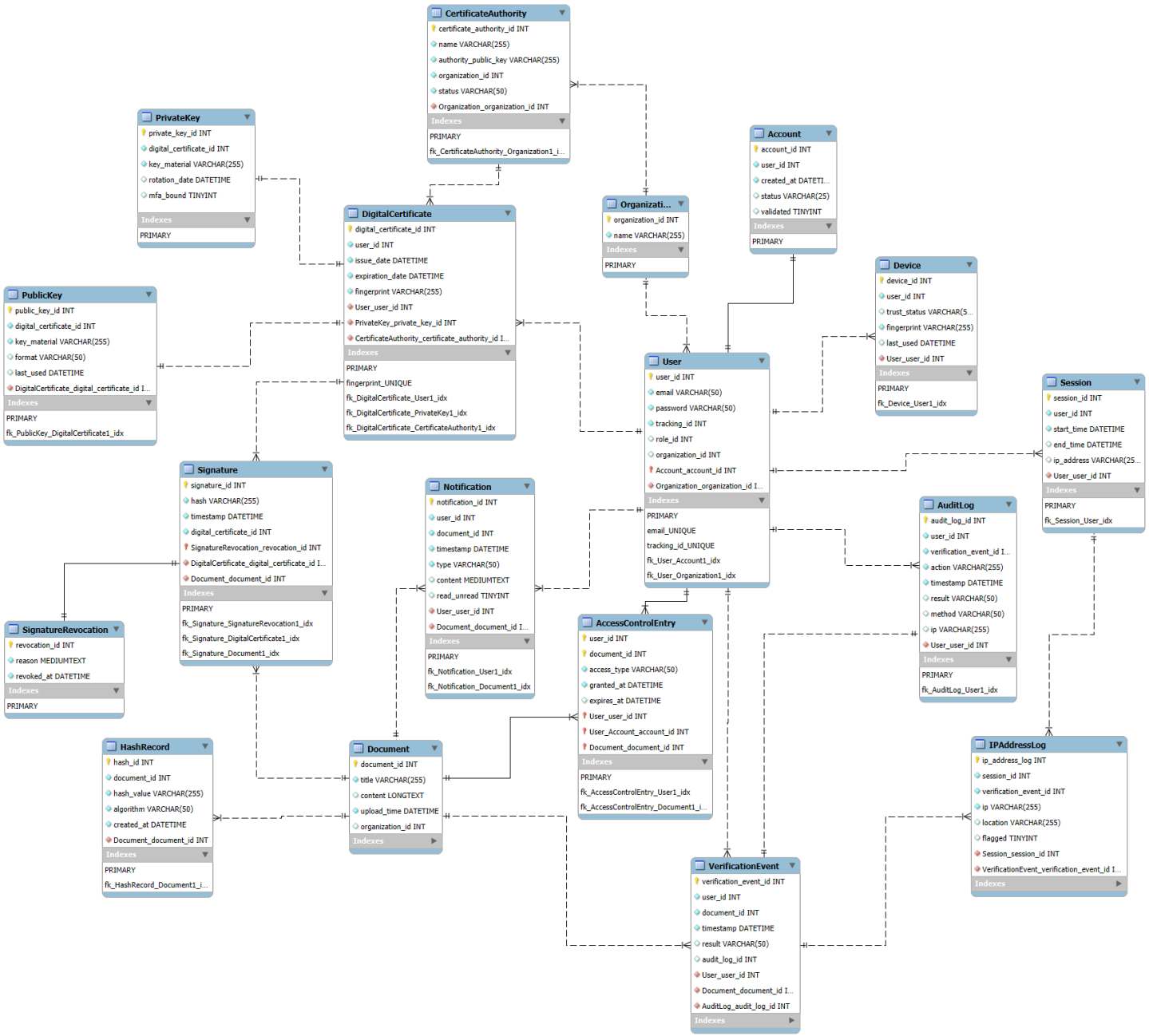
Already passes 3NF

16. CREATE TABLE Role (

```
    role_id INT PRIMARY KEY,
    title VARCHAR(50),
    permissions VARCHAR(255)
);
```

Already passes 3NF

EER and Database Schema


CertificateAuthority
certificate_authority_id INT
name VARCHAR(255)
authority_public_key VARCHAR(255)
organization_id INT
status VARCHAR(50)
Organization_organization_id INT
Indexes
PRIMARY
fk_CertificateAuthority_Organization1_i...
PrivateKey
private_key_id INT
digital_certificate_id INT
key_material VARCHAR(255)
rotation_date DATETIME
mfa_bound TINYINT
Indexes
PRIMARY
DigitalCertificate
digital_certificate_id INT
user_id INT
issue_date DATETIME
expiration_date DATETIME
fingerprint VARCHAR(255)
User_user_id INT
PrivateKey_private_key_id INT
CertificateAuthority_certificate_authority_id I...
Indexes
PRIMARY
fingerprint_UNIQUE
fk_DigitalCertificate_User1_idx
fk_DigitalCertificate_PrivateKey1_idx
fk_DigitalCertificate_CertificateAuthority1_idx
Organizati...
organization_id INT
name VARCHAR(255)
Indexes
PRIMARY
Account
account_id INT
user_id INT
created_at DATETI...
status VARCHAR(25)
validated TINYINT
Indexes
PRIMARY
Device
device_id INT
user_id INT
trust_status VARCHAR(5...
fingerprint VARCHAR(255)
last_used DATETIME
User_user_id INT
Indexes
PRIMARY
fk_Device_User1_idx
PublicKey
public_key_id INT
digital_certificate_id INT
key_material VARCHAR(255)
format VARCHAR(50)
last_used DATETIME
DigitalCertificate_digital_certificate_id I...
Indexes
PRIMARY
fk_PublicKey_DigitalCertificate1_idx
User
user_id INT
email VARCHAR(50)
password VARCHAR(50)
tracking_id INT
role_id INT
organization_id INT
Account_account_id INT
Organization_organization_id I...
Indexes
PRIMARY
email_UNIQUE
tracking_id_UNIQUE
fk_User_Account1_idx
fk_User_Organization1_idx
Session
session_id INT
user_id INT
start_time DATETIME
end_time DATETIME
ip_address VARCHAR(25...
User_user_id INT
Indexes
PRIMARY
fk_Session_User_idx
Signature
signature_id INT
hash VARCHAR(255)
timestamp DATETIME
digital_certificate_id INT
SignatureRevocation_revocation_id INT
DigitalCertificate_digital_certificate_id I...
Document_document_id INT
Indexes
PRIMARY
fk_Signature_SignatureRevocation1_idx
fk_Signature_DigitalCertificate1_idx
fk_Signature_Document1_idx
Notification
notification_id INT
user_id INT
document_id INT
timestamp DATETIME
type VARCHAR(50)
content MEDIUMTEXT
read_unread TINYINT
User_user_id INT
Document_document_id I...
Indexes
PRIMARY
fk_Notification_User1_idx
fk_Notification_Document1_idx
AuditLog
audit_log_id INT
user_id INT
verification_event_id I...
action VARCHAR(255)
timestamp DATETIME
result VARCHAR(50)
method VARCHAR(50)
ip VARCHAR(255)
User_user_id INT
Indexes
PRIMARY
fk_AuditLog_User1_idx
SignatureRevocation
revocation_id INT
reason MEDIUMTEXT
revoked_at DATETIME
Indexes
PRIMARY
AccessControlEntry
user_id INT
document_id INT
access_type VARCHAR(50)
granted_at DATETIME
expires_at DATETIME
User_user_id INT
User_Account_account_id INT
Document_document_id INT
Indexes
PRIMARY
fk_AccessControlEntry_User1_idx
fk_AccessControlEntry_Document1_i...
HashRecord
hash_id INT
document_id INT
hash_value VARCHAR(255)
algorithm VARCHAR(50)
created_at DATETIME
Document_document_id INT
Indexes
PRIMARY
fk_HashRecord_Document1_i...
Document
document_id INT
title VARCHAR(255)
content LONGTEXT
upload_time DATETIME
organization_id INT
Indexes
IPAddressLog
ip_address_log INT
session_id INT
verification_event_id INT
ip VARCHAR(255)
location VARCHAR(255)
flagged TINYINT
Session_session_id INT
VerificationEvent_verification_event_id I...
Indexes
VerificationEvent
verification_event_id INT
user_id INT
document_id INT
timestamp DATETIME
result VARCHAR(50)
audit_log_id INT
User_user_id INT
Document_document_id I...
AuditLog_audit_log_id INT
Indexes

Forward Engineering (MySQL Workbench)

Populating my Database:

The following is what ChatGPT gave me when I prompted it for a sql file with sample data to populate my tables:

```
USE dsvs;
-- Sample data insert script for Digital Signature Verification System

INSERT INTO Role (role_id, title, permissions) VALUES
(1, 'admin', 'full'),
(2, 'user', 'basic'),
(3, 'verifier', 'verify_only');

INSERT INTO Organization (organization_id, name) VALUES
(1, 'CyberOrg'),
(2, 'DataTrust'),
(3, 'SecuServe');

INSERT INTO User (user_id, email, password, tracking_id, role_id, organization_id) VALUES
(1, 'alice@example.com', 'pass123', 1001, 1, 1),
(2, 'bob@example.com', 'pass456', 1002, 2, 2),
(3, 'carol@example.com', 'pass789', 1003, 3, 3);

INSERT INTO Account (account_id, user_id, created_at, status, validated) VALUES
(1, 1, NOW(), 'active', 1),
(2, 2, NOW(), 'suspended', 0),
(3, 3, NOW(), 'active', 1);

INSERT INTO Document (document_id, title, content, upload_time, organization_id) VALUES
(1, 'Policy.pdf', 'EncryptedContent1', NOW(), 1),
(2, 'Report.docx', 'EncryptedContent2', NOW(), 2),
(3, 'Plan.txt', 'EncryptedContent3', NOW(), 3);

INSERT INTO DigitalCertificate (digital_certificate_id, user_id, issue_date, expiration_date,
fingerprint) VALUES
(1, 1, NOW(), DATE_ADD(NOW(), INTERVAL 1 YEAR), 'ABC123'),
```

```
(2, 2, NOW(), DATE_ADD(NOW(), INTERVAL 1 YEAR), 'DEF456'),
(3, 3, NOW(), DATE_ADD(NOW(), INTERVAL 1 YEAR), 'GHI789');

INSERT INTO PublicKey (public_key_id, digital_certificate_id, key_material, format, last_used)
VALUES
(1, 1, 'pubkey1', 'PEM', NOW()),
(2, 2, 'pubkey2', 'PEM', NOW()),
(3, 3, 'pubkey3', 'PEM', NOW());

INSERT INTO PrivateKey (private_key_id, digital_certificate_id, key_material, rotation_date,
mfa_bound) VALUES
(1, 1, 'privkey1', NOW(), 1),
(2, 2, 'privkey2', NOW(), 0),
(3, 3, 'privkey3', NOW(), 1);

INSERT INTO Signature (signature_id, hash, timestamp, digital_certificate_id) VALUES
(1, 'hash1', NOW(), 1),
(2, 'hash2', NOW(), 2),
(3, 'hash3', NOW(), 3);

INSERT INTO SignatureRevocation (revocation_id, reason, revoked_at, signature_id) VALUES
(1, 'Key compromise', NOW(), 1),
(2, 'User left organization', NOW(), 2),
(3, 'Expiration', NOW(), 3);

INSERT INTO CertificateAuthority (certificate_authority_id, name, authority_public_key,
organization_id, status) VALUES
(1, 'GlobalTrust', 'CAKey1', 1, 'active'),
(2, 'SecureSign', 'CAKey2', 2, 'active'),
(3, 'CertifyNow', 'CAKey3', 3, 'revoked');

INSERT INTO Session (session_id, user_id, start_time, end_time, ip_address) VALUES
(1, 1, NOW(), NOW(), '192.168.1.10'),
(2, 2, NOW(), NOW(), '192.168.1.20'),
(3, 3, NOW(), NOW(), '192.168.1.30');

INSERT INTO Device (device_id, user_id, trust_status, fingerprint, last_used) VALUES
(1, 1, 'trusted', 'devfp1', NOW()),
(2, 2, 'flagged', 'devfp2', NOW()),
(3, 3, 'trusted', 'devfp3', NOW());

INSERT INTO Notification (notification_id, user_id, document_id, timestamp, type, content,
read_unread) VALUES
```

```
(1, 1, 1, NOW(), 'alert', 'Verification failed', 0),
(2, 2, 2, NOW(), 'success', 'Document verified', 1),
(3, 3, 3, NOW(), 'error', 'Invalid signature', 0);

INSERT INTO VerificationEvent (verification_event_id, user_id, document_id, timestamp, result, audit_log_id) VALUES
(1, 1, 1, NOW(), 'success', 1),
(2, 2, 2, NOW(), 'failure', 2),
(3, 3, 3, NOW(), 'success', 3);

INSERT INTO AuditLog (audit_log_id, user_id, verification_event_id, action, timestamp, result, method, ip) VALUES
(1, 1, 1, 'verify_signature', NOW(), 'success', 'auto', '192.168.1.10'),
(2, 2, 2, 'verify_signature', NOW(), 'failure', 'manual', '192.168.1.20'),
(3, 3, 3, 'revoke_certificate', NOW(), 'success', 'admin', '192.168.1.30');

INSERT INTO HashRecord (hash_id, document_id, hash_value, algorithm, created_at) VALUES
(1, 1, 'hashVal1', 'SHA-256', NOW()),
(2, 2, 'hashVal2', 'SHA-3', NOW()),
(3, 3, 'hashVal3', 'SHA-256', NOW());

INSERT INTO IPAddressLog (ip_address_log, session_id, verification_event_id, ip, location, flagged) VALUES
(1, 1, 1, '192.168.1.10', 'NYC', 0),
(2, 2, 2, '192.168.1.20', 'LA', 1),
(3, 3, 3, '192.168.1.30', 'SF', 0);

INSERT INTO AccessControlEntry (user_id, document_id, access_type, granted_at, expires_at) VALUES
(1, 1, 'read', NOW(), DATE_ADD(NOW(), INTERVAL 30 DAY)),
(2, 2, 'write', NOW(), DATE_ADD(NOW(), INTERVAL 30 DAY)),
(3, 3, 'revoke', NOW(), DATE_ADD(NOW(), INTERVAL 30 DAY));
```

I ran the sql script and can confirm it works as expected.

Note: Foreign key constraints were omitted in the schema file (dsvs.sql) due to persistent errors in MySQL Workbench. I instead documented them in a foreign_keys.sql file and manually added them after table creation.

23

Result Grid | Filter Rows: | Edit: | Export/Import:

document_id	title	content	upload_time	organization_id
1	Policy.pdf	EncryptedContent1	2025-07-01 16:17:17	1
2	Report.docx	EncryptedContent2	2025-07-01 16:17:17	2
3	Plan.txt	EncryptedContent3	2025-07-01 16:17:17	3
NULL	NULL	NULL	NULL	NULL

Result Grid | Filter Rows: | Edit: | Export/Import:

certificate_authority_id	name	authority_public_key	organization_id	status
1	GlobalTrust	CAKey1	1	active
2	SecureSign	CAKey2	2	active
3	CertifyNow	CAKey3	3	revoked
NULL	NULL	NULL	NULL	NULL

Result Grid | Filter Rows: | Edit: | Expor

hash_id	document_id	hash_value	algorithm	created_at
1	1	hashVal1	SHA-256	2025-07-01 16:17:17
2	2	hashVal2	SHA-3	2025-07-01 16:17:17
3	3	hashVal3	SHA-256	2025-07-01 16:17:17
NULL	NULL	NULL	NULL	NULL

Result Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content: | F

audit_log_id	user_id	verification_event_id	action	timestamp	result	method	ip
1	1	1	verify_signature	2025-07-01 16:17:17	success	auto	192.168.1.10
2	2	2	verify_signature	2025-07-01 16:17:17	failure	manual	192.168.1.20
3	3	3	revoke_certificate	2025-07-01 16:17:17	success	admin	192.168.1.30
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Result Grid
Form Editor

Result Grid | Filter Rows: | Edit: | Ex

device_id	user_id	trust_status	fingerprint	last_used
1	1	trusted	devfp1	2025-07-01 16:17:17
2	2	flagged	devfp2	2025-07-01 16:17:17
3	3	trusted	devfp3	2025-07-01 16:17:17
NULL	NULL	NULL	NULL	NULL