PDF Encryption Software Architecture

Definition of Requirements:

- Key Stakeholders:
    a. Home Credit
        Field Coordinators
        Managers
    b. Regulatory Bodies
        National Privacy Commission (NPC)
        Bangko Sentral ng Pilipinas (BSP)
        Securities and Exchange Commission (SEC)
        Department of Trade and Industry (DTI)
        Intellectual Property Office of the Philippines (IPOPHL)
        Contractual Agreements

- Functional Requirements:
    a. Encryption and Decryption Capabilities
        The proposed software will utilize a password generation system and PDF encryption module to encrypt and decrypt files with passwords that expire every five minutes.
    b. File format support
        The proposed software will use an iText 7.x and iText 7 for .NET version from the iText/iTextSharp libraries which will offer PDF formats support up to PDF 2.0. Additionally, advantages such as a modern API, improved architecture, enhanced features, and performance optimization can be gained from this version.
    c. User interface requirements
        The proposed software will come with a simple graphical user interface that includes the selection of files, encryption, and decryption settings, as well as progress indicators and error handling/reports.

- Non-functional requirements
    a. Authentication and access control

The proposed software will include strong authentication mechanisms such as multi-factor authentication and role-based access controls for users accessing the software, as well as secure protocols for protected data transmissions and authentication.

b. Data encryption
c. Audit logging and monitoring

   Logging and monitoring of generated OTPs, password revocation, and other essential data.

d. Vulnerability and patch management

   Implementation of regular updates and patches to address any known vulnerabilities.

e. Secure Network and Security

   Integration of network segmentations, firewalls, and intrusion detection/prevention systems (IDS/IPS) to mitigate risks associated with network-based attacks, such as DDoS attacks.

f. Incident Response and Recovery

   Outlining comprehensive plans to prevent, identify, and address incidents and security breaches

g. Compliance and regulatory requirements

Software Architecture

Programming Language:

- Continue using Java on your team's expertise and project requirements.

Password Generation System:

- Utilize OTP for generating temporary passwords and SHA-256 for cryptographic functions, as previously discussed.

PDF Encryption Module:

- Use a PDF library compatible with your chosen programming language (e.g., iText for Java, iTextSharp for C#) to handle encryption and decryption of PDF files.

Decryption Process:

- Modify the decryption process to validate passwords against a static password or temporary password (generated by OTP) as before.
- Enhance the password validation logic to check if the password has been revoked due to the document being printed.
- Implement error handling and user feedback mechanisms to inform users when a password has been revoked.

MFA and Auditing/Logging:

- Continue implementing MFA and auditing/logging features as previously discussed to enhance security and accountability.
- Log events of password revocation along with relevant details such as the document identifier, user identity, and timestamp.
- Include information about the reason for revocation (e.g., document printed) in the audit log entries.
- MFA features include the sending of the OTP to the field coordinator's registered mobile number

Restricted File Actions and Conditional

- Set access permissions and restrictions within the PDF file to control what users can do with the document.
- Common permissions include prohibiting copying text or images, modifying content, or adding annotations.

Watermarking and Conditional Display and Rendering

- Add a visible watermark to the PDF document using software or PDF editing tools.
- Embed an invisible watermark within the content of the PDF document using the comprehensive features of the PDF library
- Configure the invisible watermark to include unique identifiers, timestamps, or user-specific information that can be detected through forensic analysis.
- Ensure that the invisible watermark is seamlessly integrated into the document's content layer and does not affect its visual appearance or readability.

- Implement logic or settings within the PDF viewer or printing software to control the visibility or rendering of watermarks based on specific conditions.
- Configure the PDF viewer to display the visible watermark when viewing the document digitally but suppress its rendering when printing or exporting the document.
- Ensure that the invisible watermark remains embedded within the document's content layer regardless of the viewing or printing settings.

Security and Deterrence:

- The visible watermark serves as a deterrent against unauthorized use or distribution of the document when viewed digitally, as it visibly indicates the document's protected status.
- The invisible watermark provides covert identification and traceability, enabling forensic analysis and accountability in the event of unauthorized access or misuse.
- By combining visible and invisible watermarks, you enhance document security, deterrence, and traceability across different usage scenarios.

Password Revocation:

- Develop a feature in your software that allows field officers to signal the system when a document has been printed. This could be through a dedicated button or interface.
- Implement logic to revoke the validity of passwords associated with documents that have been printed and signaled by field officers.
- Update the status or flag associated with the password record to mark it as revoked in the backend system or database.

User Interface (UI):

- Develop a user-friendly interface that allows users to interact with the software easily.
- Design intuitive screens for password input, document selection, and decryption operations.
- Provide clear feedback to users regarding password validation, decryption status, and any errors encountered.
- Include features for users to signal the system when a document has been printed and to view audit logs for security monitoring purposes.
- In-app notifications for the generation of passwords

Backend Server:

- Develop a backend server to handle password generation, validation, and revocation processes.
- Implement secure communication protocols (e.g., HTTPS) to transmit data between the client and server.
- Ensure that the backend server synchronizes its system time with a secure time source to maintain accurate time measurements.

Secure Time Source:

- Implement a secure time synchronization mechanism within your software to ensure accurate and consistent time measurements.
- Utilize an internet-based time server or network time protocol (NTP) to synchronize the system clock with a trusted time source.
- Implement error handling and redundancy measures to handle cases where the secure time source is unavailable or unreliable.

Additional Security Measures:

- Regularly update and patch software components to address any known vulnerabilities.
- Implement network segmentation and firewall rules to restrict access to sensitive systems and data.
- Conduct regular security assessments, including penetration testing and vulnerability scanning, to identify and address security weaknesses.

Integrating the PDF Encryption Software to MUCH

Assess Integration Requirements:

- Understand the requirements of the existing system and identify how the PDF encryption software will fit into it.
- Determine the specific features, functionalities, and data flows needed for integration.

Define Interfaces and APIs:

- Design clear interfaces and application programming interfaces (APIs) that allow communication between your PDF encryption software and the existing system.

- Specify the methods, parameters, and data formats used for exchanging information between components.

Develop Integration Points:

- Implement integration points within both the PDF encryption software and the existing system to facilitate data exchange and interoperability.
- Develop adapters, connectors, or middleware components as needed to bridge any gaps between different technologies or protocols.

Handle Data Synchronization:

- Determine how data will be synchronized between the PDF encryption software and the existing system to ensure consistency and accuracy.
- Implement mechanisms for data import/export, synchronization, and reconciliation to keep information up to date across systems.

Address Security and Authentication:

- Ensure that proper authentication and authorization mechanisms are in place to control access to the integrated components.
- Implement secure communication protocols (e.g., HTTPS) and encryption techniques to protect data transmitted between systems.