

DOI: <https://doi.org/10.38035/gijlss.v4i4><https://creativecommons.org/licenses/by/4.0/>

Received: 17 June 2026 | Revised: 07 August 2026 | Accepted: 3 October 2026

Encryption, Digital Governance, and Personal Data Protection: Developing a Juridical–Technical Enforcement Model under Indonesia’s Law No. 27 of 2022

Musa Darwin Pane^{1*}, Mari Maryati², Angga Setiyadi³¹Universitas Komputer Indonesia, Bandung, Indonesia, musa@email.unikom.ac.id²Universitas Komputer Indonesia, Bandung, Indonesia, mari.maryati@email.unikom.ac.id³Universitas Komputer Indonesia, Bandung, Indonesia, angga.setiyadi@email.unikom.ac.id*Corresponding Author: musa@email.unikom.ac.id¹

Abstract: The rapid digitalization of organizational data management has intensified concerns about personal data protection and the practical implementation of regulatory obligations. A significant gap remains between the normative requirements of Indonesia’s Law Number 27 of 2022 concerning Personal Data Protection and their technical implementation in information systems. This study aims to develop and evaluate a Juridical–Technical Model for encrypted digital data storage in cooperative information systems. The study employed an interdisciplinary design integrating juridical analysis and software engineering through Privacy by Design and a prototyping approach. Data were obtained through system observation, database analysis, and requirements identification. The prototype implemented XChaCha20-Poly1305 authenticated encryption through the Libsodium cryptographic library. The findings showed that the existing system stored directly identifiable personal information in plaintext, creating substantial risks of unauthorized access and manipulation. The proposed model converted protected data into authenticated ciphertext and successfully rejected modified records during decryption testing. These results demonstrate that selected legal obligations concerning security, confidentiality, unlawful processing, and unlawful access can be translated into executable technical safeguards. The study contributes a practical and replicable framework for strengthening secure, accountable, and privacy-preserving digital data governance in cooperatives.

Keywords: Personal Data Protection Law, Privacy by Design, XChaCha20-Poly1305, Authenticated Encryption, Digital Data Governance.

INTRODUCTION

The rapid advancement of information and communication technologies has fundamentally transformed how organizations collect, process, store, and exchange information. Digital transformation has become a strategic priority for public and private organizations because it improves operational efficiency, data accessibility, organizational

agility, and evidence-based decision-making (Bilan et al., 2023; Saleem et al., 2020). The adoption of database-driven information systems also enables organizations to manage large volumes of information in real time, reduce administrative burdens, improve service quality, and strengthen organizational performance (Atobishi et al., 2024; Chaudhuri et al., 2024; Das, 2024). Consequently, digital data are no longer merely administrative records but have become strategic organizational assets that support innovation, competitiveness, and institutional sustainability.

The transition from conventional administrative procedures to integrated digital systems is increasingly evident in organizations that process large volumes of personal information, including financial institutions, healthcare organizations, educational institutions, and cooperatives (Shojaei et al., 2024). In Indonesia, the government has actively promoted the digital transformation of cooperatives by targeting an increase in the number of modern digitally based cooperatives from 250 units in 2022 to 400 units in 2023 and 500 units in 2024 (Sekretariat Kabinet Republik Indonesia, 2022). This expansion demonstrates that cooperative activities, including membership registration, financial administration, savings and loan management, and member communication, are increasingly conducted through electronic information systems.

However, the increasing digitalization of cooperatives also expands their exposure to cybersecurity threats. The Indonesian National Cyber and Crypto Agency detected 103 suspected data-leak incidents in Indonesia during 2023, with monthly peaks of 20 incidents in March and 15 incidents in December (Kementerian Komunikasi dan Informatika Republik Indonesia, 2024). Although publicly available national statistics do not routinely separate cyber incidents involving cooperatives from those affecting other sectors, the combination of rapid digital adoption and the extensive collection of members' identity and transactional information demonstrates that cooperative information systems represent an increasingly important cybersecurity concern. At the case-study level, preliminary technical observation further revealed that personal information, including national identification numbers, names, addresses, and telephone numbers, was stored directly in plaintext format without adequate cryptographic protection.

The storage of personal information in plaintext creates a significant vulnerability because unauthorized access to the database can immediately reveal readable and identifiable information. A successful attack through structured query language injection, server misconfiguration, privilege escalation, stolen administrative credentials, or insider misuse may expose the entire membership database without requiring attackers to overcome an additional security layer (Arif et al., 2025; Naguib & Fouad, 2024; Samaraweera & Chang, 2021). Such exposure may result in identity theft, financial fraud, unauthorized profiling, discrimination, social engineering, and other forms of cybercrime (Al-Khater et al., 2020; Zaeem & Barber, 2025). Therefore, weaknesses in cooperative database architecture should not be regarded merely as technical deficiencies but also as risks to members' privacy rights and the legal accountability of cooperative management.

The economic value of personal information further increases these risks. Personal data have become valuable resources within digital economic ecosystems, which has contributed to the increasing frequency and sophistication of cyberattacks directed at organizational databases (Iman, 2024; Sánchez, 2022; Shukla et al., 2023). Incidents involving unauthorized access, ransomware, insider threats, data manipulation, and large-scale disclosure demonstrate that organizations cannot rely solely on passwords, user roles, or perimeter-based security controls. When sensitive records remain readable within the database, the compromise of a server or privileged account may directly expose the identities and confidential information of data subjects. Effective personal data protection must therefore incorporate security controls into the structure and operation of the information system itself.

From a legal perspective, privacy is increasingly recognized as an essential individual right in the digital era. International privacy frameworks require data controllers to implement appropriate technical and organizational measures to protect the confidentiality, integrity, and availability of personal information (Ibrahim et al., 2024; Sim et al., 2023). This principle is reflected in the concept of *privacy by design*, which requires privacy and data protection safeguards to be integrated from the earliest stages of system development rather than introduced only after an information system has been deployed (Georgiadis & Poels, 2021). Accordingly, personal data protection should operate as an inherent characteristic of system architecture rather than merely as an administrative policy or supplementary compliance document.

In Indonesia, the legal foundation for personal data governance was strengthened through the enactment of Law Number 27 of 2022 concerning Personal Data Protection. The legislation regulates the rights of personal data subjects and establishes obligations for personal data controllers and processors. Article 35 requires personal data controllers to adopt technical and operational measures and determine an appropriate security level by considering the nature and risks of the data being processed. Article 36 requires controllers to maintain the confidentiality of personal data, while Article 38 requires protection against unlawful processing. Furthermore, Article 39 obligates controllers to prevent unlawful access by implementing reliable, secure, and accountable electronic security systems. These provisions establish a legal basis for embedding confidentiality, access protection, and resistance to unauthorized manipulation within information-system architecture.

Nevertheless, practical implementation of the Personal Data Protection Law remains challenging. The legislation establishes mandatory security outcomes but generally does not prescribe a particular cryptographic algorithm, database architecture, or software-development method that every organization must use. This technology-neutral approach provides flexibility and accommodates technological development, but it also creates uncertainty concerning how normative legal obligations should be translated into measurable software requirements. Organizations, particularly cooperatives and smaller institutions with limited cybersecurity capabilities, may encounter difficulties in determining which encryption techniques, access controls, authentication mechanisms, and database protections are sufficient to demonstrate legal compliance (Hale & Gamble, 2019; Sun et al., 2022). Consequently, a gap persists between the normative requirements of personal data protection law and their implementation within operational information systems.

Existing literature has generally examined this problem from separate disciplinary perspectives. Legal studies primarily discuss data-subject rights, controller obligations, administrative sanctions, institutional authority, and regulatory enforcement without providing detailed technical models for implementation (Scheibner et al., 2020; Yeung & Bygrave, 2022). Conversely, information-security research commonly focuses on algorithmic strength, encryption performance, database-security models, and cybersecurity frameworks without sufficiently connecting those mechanisms to specific statutory obligations (Alghamdi & Munir, 2024; Radanliev, 2024; Ramakrishna & Ali Shaik, 2025). This disciplinary separation limits the development of practical models capable of translating legal requirements directly into executable security functions.

Previous studies on cryptographic implementation have also concentrated largely on computational performance, resistance to cryptographic attacks, and algorithmic efficiency. Limited attention has been devoted to explaining how authenticated encryption can be used as a legal-compliance mechanism within organizational databases. This limitation is important because encryption alone does not necessarily detect unauthorized modification. A legally relevant security architecture must protect the confidentiality of personal information while also enabling the system to identify and reject manipulated records. Authenticated encryption

is therefore particularly relevant because it combines encryption with an authentication mechanism that verifies whether stored ciphertext has been altered.

To address these theoretical and practical gaps, this study develops a Juridical–Technical Model that translates obligations under Indonesia’s Personal Data Protection Law into an encryption-based database-security architecture. The proposed model applies the *privacy-by-design* approach and implements the XChaCha20-Poly1305 authenticated encryption algorithm through the Libsodium cryptographic library. XChaCha20 transforms readable personal information into unintelligible ciphertext, while Poly1305 generates an authentication tag that enables the system to detect unauthorized modifications. Through this mechanism, the model is designed to support confidentiality, prevent unlawful access, preserve data authenticity, and reject manipulated records before they are presented to authorized users.

Accordingly, this study aims to design, implement, and evaluate an interdisciplinary model of encrypted digital-data storage that bridges normative legal requirements and software-engineering implementation. Specifically, the study seeks to: (1) construct measurable technical parameters derived from the security obligations contained in Law Number 27 of 2022; (2) develop an encryption-based database architecture grounded in *privacy-by-design* principles; and (3) evaluate whether the proposed model satisfies both cybersecurity requirements and personal-data-protection objectives.

The novelty of this study lies in the direct translation of legal obligations into executable cryptographic functions within a cooperative information system. Unlike previous studies that examine regulatory compliance and information security separately, this study connects statutory requirements, legal parameters, system specifications, database architecture, authenticated encryption, and functional validation within a single model. The proposed model therefore offers not only a cryptographic prototype but also a governance framework through which personal data protection can be internalized within information-system architecture. Practically, the model is expected to provide cooperatives, software developers, data controllers, and policymakers with a replicable framework for developing secure, legally accountable, and privacy-preserving digital infrastructures.

METHOD

This study employed an applied interdisciplinary research design integrating juridical analysis and software engineering to develop and evaluate a legally compliant and technically secure digital data-storage model. The interdisciplinary approach was selected because personal data protection requires not only legal interpretation but also the translation of regulatory obligations into operational software requirements. Previous studies have shown that data-protection regulations influence requirements identification, system design, process modelling, and software-development practices (Leite et al., 2022; Negri-Ribalta et al., 2024). Accordingly, this study combined legal analysis with system development to bridge the gap between normative compliance and technical implementation.

The juridical approach was used to identify and translate obligations established under Law Number 27 of 2022 concerning Personal Data Protection into measurable security requirements. The legal analysis focused on Article 35 concerning technical and operational security measures, Article 36 concerning the confidentiality of personal data, Article 38 concerning protection against unlawful processing, and Article 39 concerning the prevention of unlawful access (Republic of Indonesia, 2022). These provisions were operationalized into technical requirements involving encrypted storage, controlled data retrieval, confidentiality protection, authentication verification, and detection of unauthorized data modification.

The software-engineering approach was used to implement the identified legal requirements within a functional prototype of a cooperative membership information system. The integration of legal and technical requirements was guided by the *Privacy by Design*

framework, which emphasizes that privacy and security safeguards should be embedded throughout the software-development lifecycle rather than added after system deployment (Alkubaisy et al., 2022; Andrade et al., 2022; Georgiadis & Poels, 2021). Therefore, legal compliance was treated as an intrinsic component of the database architecture, application logic, and security mechanism.

The unit of analysis consisted of the application architecture, database structure, data-storage process, and security functions of an anonymized cooperative membership information system in Indonesia. The identity and precise location of the cooperative were not disclosed to protect institutional confidentiality and reduce the security risks associated with publishing detailed information about the organization's digital infrastructure. Because the research focused on system development and technical validation rather than respondent attitudes or behaviour, it did not involve a statistical population or respondent sample.

Primary data were collected through non-participant observation and system-requirement analysis. The observation examined the membership-registration workflow, application interface, database structure, access mechanisms, and existing personal-data storage practices. Particular attention was given to directly identifiable information, including national identification numbers, full names, places and dates of birth, residential addresses, and telephone numbers. The analysis identified personal-data fields stored in plaintext and assessed the consequences that could arise if the database, server, application, or privileged user account were accessed unlawfully. Plaintext storage was treated as a significant vulnerability because unauthorized database access could immediately expose readable personal information (Naguib & Fouad, 2024; Samaraweera & Chang, 2021).

The identified data were mapped according to the statutory classification of general and specific personal data under Law Number 27 of 2022. The classification also considered the identification value of each field, the potential consequences of unauthorized disclosure, and the level of processing risk. Direct personal identifiers were treated as operationally sensitive even when they were not legally classified as specific personal data because their disclosure could facilitate identity theft, fraud, profiling, and social-engineering attacks (Al-Khater et al., 2020; Dhirani et al., 2023).

Secondary data were collected through document analysis and a review of relevant literature. The documents included Law Number 27 of 2022, studies on *Privacy by Design*, database security, software engineering, authenticated encryption, cryptographic implementation, and digital-security governance. The review was used to connect statutory requirements with system-security specifications and to determine an appropriate technical mechanism for protecting confidentiality and verifying the integrity of encrypted records.

Confidentiality and data-minimization principles were applied during system analysis and prototype testing. Access to technical information was limited to the system-development process, while actual members' personal information was not reproduced in the publication. For publication, Figure 1 and all system demonstrations were regenerated using simulated values that preserve the relevant field structure without reproducing any actual member NIK, name, address, telephone number, or other personal identifier. Personal identifiers shown in testing scenarios were therefore masked, anonymized, or replaced with simulated values. Application credentials, database connection details, cryptographic keys, and other security-sensitive information were excluded from the manuscript.

Tests involving the modification of encrypted records were conducted in a development or replicated database environment. The operational database was not intentionally altered for the integrity-testing scenario. This procedure was intended to prevent disruption of organizational services and avoid unauthorized modification of authentic membership records. The confidentiality measures applied in the study therefore included identity anonymization,

restricted technical access, data minimization, masking of identifiers, and the use of non-production data for security testing.

System development followed the System Development Life Cycle through a prototyping approach. The development procedure consisted of legal- and technical-requirement identification, vulnerability analysis, database redesign, prototype construction, cryptographic implementation, functionality testing, compliance evaluation, and iterative refinement. The use of an iterative development process supported the continuous alignment of legal requirements with software design, which is consistent with the principle that privacy requirements should be addressed throughout the system-development lifecycle (Andrade et al., 2022; Leite et al., 2022).

During the requirement-identification stage, the obligations contained in Articles 35, 36, 38, and 39 of the Personal Data Protection Law were converted into operational security specifications. Article 35 was translated into the requirement to implement technical safeguards proportionate to the risks of data processing. Article 36 was operationalized through encryption and restricted decryption to maintain confidentiality. Article 38 was translated into protection against unlawful processing, while Article 39 was operationalized through a secure electronic system designed to reduce unlawful access.

The vulnerability-analysis stage examined the existing membership system to identify weaknesses in data storage and retrieval. The analysis revealed that several personal-data fields were stored directly in plaintext format. Under this condition, a party that successfully obtained database-level access could read the stored information without overcoming an additional cryptographic security layer. Based on these findings, field-level encryption and authenticated integrity verification were selected as the principal requirements for the redesigned system.

The relational database architecture was redesigned using the MySQL Database Management System, while the application backend was developed using PHP. Personal-data fields requiring cryptographic protection were identified and allocated sufficient storage capacity for encrypted values. Encryption and verification functions were embedded into the application logic so that protected information was converted into ciphertext before permanent database storage. Decryption was performed only when the data were requested through an authorized application process.

The prototype implemented the XChaCha20-Poly1305 authenticated encryption algorithm through the Libsodium cryptographic library. XChaCha20-Poly1305 was selected because authenticated encryption combines confidentiality and authentication within a single cryptographic construction. ChaCha20 provides stream-encryption protection, while Poly1305 produces an authentication tag that enables the application to verify the authenticity of encrypted data (Degabriele et al., 2021; Nir & Langley, 2018). Authenticated encryption is particularly relevant to secure storage because it protects data confidentiality while enabling unauthorized alterations to be detected (Benadjila et al., 2022).

For every encryption transaction, the system generated a unique nonce through a cryptographically secure random-byte function. The plaintext was encrypted using XChaCha20 and authenticated through Poly1305. The nonce and encrypted output were combined and encoded in Base64 format before being stored in the database. During retrieval, the nonce was separated from the ciphertext and authenticated decryption was performed using the corresponding secret key. If the ciphertext, nonce, authentication tag, or secret key did not correspond to the original encryption process, the decryption operation failed and the application rejected the record.

This mechanism was designed to produce two principal security outcomes. First, confidentiality was achieved because personal information could not be read directly from the database without the correct cryptographic key. Second, integrity and authenticity were supported because unauthorized modification of encrypted data invalidated the authentication

tag and caused the decryption process to fail. Therefore, the proposed model not only concealed personal information but also enabled manipulated records to be detected and rejected.

The key included in the prototype was used exclusively for simulation and functionality testing. It was not treated as an appropriate key-management mechanism for production deployment. Cryptographic key management constitutes an essential component of secure storage because the confidentiality of encrypted data depends on protecting the corresponding encryption keys (Benadjila et al., 2022; Radanliev, 2024). In an operational environment, cryptographic keys should be separated from application source code and encrypted databases, restricted to authorized processes, and protected through secure generation, access control, logging, rotation, backup, revocation, and destruction procedures.

For systems processing large volumes of high-risk personal data, a dedicated key-management service or Hardware Security Module may provide stronger logical and physical protection. However, a Hardware Security Module was not implemented or tested in the present prototype. It was therefore positioned as a recommendation for future operational deployment rather than as a validated component of the developed model.

Technical evaluation was conducted through black-box functionality testing. The evaluation examined system behaviour based on the relationship between inputs and outputs. Three principal scenarios were tested: encryption, decryption, and integrity verification. The encryption test assessed whether personal information entered through the registration interface was transformed into ciphertext before database storage. The decryption test examined whether an authorized process using the correct key could restore valid ciphertext to its original readable form. The integrity test examined whether the application rejected encrypted data that had been modified without authorization.

During the integrity test, one or more characters of the stored ciphertext were manually modified in the development database. The altered data were subsequently retrieved through the application. The test was considered successful when the authentication mechanism detected the modification, rejected the decryption operation, and produced an error notification instead of displaying inaccurate or manipulated personal information. This testing procedure was consistent with the authenticated-encryption principle that alteration of ciphertext or its authentication tag should result in verification failure (Degabriele et al., 2021; Nir & Langley, 2018).

The results were analyzed using descriptive qualitative analysis and a juridical–technical compliance matrix. Technical outcomes were compared with the operational parameters derived from Articles 35, 36, 38, and 39 of Law Number 27 of 2022. The evaluation examined whether the prototype implemented appropriate technical safeguards, maintained the confidentiality of stored information, reduced the risk of unlawful processing and access, and detected unauthorized modification through authentication failure.

Through this dual validation process, the prototype was assessed both as a software-engineering artifact and as a technical mechanism supporting compliance with Indonesia's personal-data-protection framework. The model was considered functionally feasible when it successfully encrypted information before storage, restored valid ciphertext through authorized decryption, and rejected manipulated ciphertext. It was considered juridically relevant when these functions corresponded to the statutory obligations concerning security measures, confidentiality, protection from unlawful processing, and prevention of unlawful access.

RESULTS AND DISCUSSION

This section presents the research findings and discusses them by integrating the technical dimensions of authenticated encryption with the legal obligations established under Law Number 27 of 2022 concerning Personal Data Protection. The results cover the existing condition of cooperative data management, the construction of juridical–technical parameters,

the conceptual architecture of the proposed model, database redesign, cryptographic implementation, technical functionality testing, and evaluation of legal-compliance support.

Existing Conditions of Cooperative Data Management

The cooperative membership information system examined in this study continued to rely on a conventional relational database architecture. In its daily operations, the system collected and processed various categories of directly identifiable personal information, including national identification numbers, full names, places and dates of birth, residential addresses, telephone numbers, occupations, and other membership-related information.

A technical review of the application interface and the structure of the operational MySQL database revealed a critical security vulnerability. Personal information was stored in plaintext without field-level encryption or authenticated integrity protection. Consequently, anyone who obtained unauthorized access to the database could immediately read, copy, modify, or distribute the stored information without having to overcome an additional cryptographic security layer.

SIMULATED DATA FOR PUBLICATION — NO REAL MEMBER INFORMATION									
member_id	full_name	place_of_birt	date_of_birth	address	RT	RW	postal	phone	status
SIM-001	Sample Member A	Bandung	1990-01-01	Example Street 1	001	002	40100	SIM-PHONE..	Active
SIM-002	Sample Member B	Jakarta	1992-02-02	Example Street 2	003	004	10100	SIM-PHONE..	Active
SIM-003	Sample Member C	Makassar	1994-03-03	Example Street 3	005	006	90100	SIM-PHONE..	Active

national_id_number (N)	SIM-NIK-0001
full_name	Sample Member A
place_of_birth	Bandung
date_of_birth	1990-01-01
address	Example Street 1, Example District
religion	Sample Value
phone_number	SIM-PHONE-01
occupation	Sample Occupation

Illustrative reconstruction of plaintext storage using synthetic identifiers and names.

Figure 1. Simulated Illustration of Current Plaintext Data Storage Practices in Cooperative Data Management

Source: Research Results (simulated data for publication).

As illustrated in Figure 1, simulated national identification numbers and simulated member names are displayed in a human-readable plaintext format to reproduce the structural vulnerability observed in the original system without disclosing actual member data. Figure 1 is therefore an illustrative reconstruction using synthetic values rather than a reproduction of operational membership records. Plaintext storage creates a substantial cybersecurity risk because a successful compromise of the database infrastructure may immediately expose personal information. Such compromise may occur through structured query language injection, server misconfiguration, stolen administrative credentials, privilege escalation, or misuse of access by authorized insiders (Naguib & Fouad, 2024; Samaraweera & Chang, 2021).

The absence of field-level encryption means that ordinary access-control mechanisms, such as usernames, passwords, and administrative permissions, constitute the primary barriers protecting the data. Once those barriers are bypassed, the information remains directly intelligible. This condition increases the potential for identity theft, financial fraud, unauthorized profiling, social engineering, privacy violations, and unlawful dissemination of members' personal information (Al-Khater et al., 2020; Dhirani et al., 2023).

From a juridical perspective, the identified condition is inconsistent with the obligation of personal data controllers to apply appropriate technical and operational safeguards, preserve confidentiality, protect personal data from unlawful processing, and prevent unlawful access, as provided under Articles 35, 36, 38, and 39 of Law Number 27 of 2022 (Republic of Indonesia, 2022). The result therefore demonstrates a significant gap between statutory personal-data-protection obligations and the security mechanisms implemented within the cooperative information system.

Construction of Parameters Based on Law Number 27 of 2022

In response to the identified vulnerabilities, the cooperative, in its capacity as a Personal Data Controller, is required to establish security measures proportionate to the nature and risks of the personal information it processes. To translate these legal obligations into measurable software-engineering requirements, this study constructed a juridical-technical parameter framework based on Articles 35, 36, 38, and 39 of Law Number 27 of 2022.

Article 35 requires Personal Data Controllers to protect and ensure the security of processed personal data by preparing and implementing technical and operational measures and determining an appropriate security level based on the nature and risks of the data. In the proposed model, this provision was operationalized through risk-based data classification, field-level encryption, authenticated decryption, restricted access to cryptographic keys, and integrity-verification mechanisms.

Article 36 requires Personal Data Controllers to maintain the confidentiality of personal data during processing. This obligation was translated into the confidentiality parameter. Technically, directly identifiable information should not remain permanently stored in a readable format. Instead, the information should be converted into ciphertext before database storage so that unauthorized parties cannot directly identify the original value even if they obtain access to the storage layer.

Article 38 requires Personal Data Controllers to protect personal data from unlawful processing. This provision was translated into safeguards that restrict data transformation and retrieval to authorized application processes. The system must reject data that do not pass cryptographic authentication and prevent manipulated records from being presented as valid personal information.

Article 39 requires Personal Data Controllers to prevent personal data from being accessed unlawfully by implementing an electronic security system that is reliable, secure, and accountable. This provision was operationalized through encrypted storage, authenticated decryption, access restrictions, and the requirement that sensitive information can only be restored using a valid cryptographic key.

Integrity was positioned as a technical security outcome supporting the obligations contained in Articles 35, 38, and 39 rather than being attributed to Article 40. Article 40 concerns the termination of personal-data processing following the withdrawal of consent and does not specifically regulate the prevention of unlawful data modification. Within the proposed model, integrity is supported through Poly1305 authentication tags, which enable the application to detect and reject unauthorized changes to encrypted records.

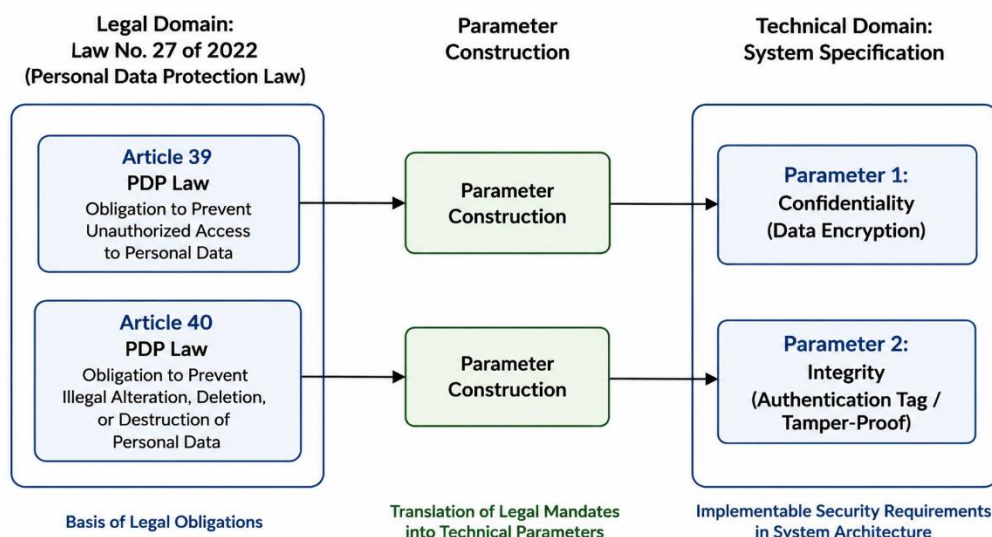


Figure 2. Development of Parameters Based on Law No. 27 of 2022

Source: Research Results.

Figure 2 demonstrates that the legal provisions were not treated merely as abstract administrative requirements. Each provision was converted into a technical parameter and subsequently linked to an operational security mechanism. Through this translation process, statutory obligations become measurable software requirements capable of being implemented and tested within the cooperative information system.

Conceptual Architecture of the Juridical–Technical Model

The conceptual architecture of the Juridical–Technical Model was designed to bridge normative personal-data-protection obligations and their practical implementation through software engineering. The legal parameters derived from Law Number 27 of 2022 were translated into technical specifications through the application of the *Privacy by Design* approach.

The *Privacy by Design* approach requires personal-data protection to be incorporated from the earliest stages of system analysis, design, development, implementation, and evaluation rather than added only after the system has been deployed (Alkubaisy et al., 2022; Andrade et al., 2022). Accordingly, encryption, authentication, and integrity verification were embedded directly within the application logic and database-storage process.

As illustrated in Figure 3, the proposed model establishes a transformation pathway connecting legal obligations, privacy governance principles, software-engineering specifications, cryptographic implementation, technical testing, and compliance evaluation. Articles 35, 36, 38, and 39 constitute the regulatory foundation. These obligations are interpreted through *Privacy by Design* and translated into requirements concerning risk-based security, confidentiality, protection against unlawful processing, and prevention of unlawful access.

The requirements are subsequently implemented using the XChaCha20-Poly1305 authenticated encryption algorithm. XChaCha20 transforms readable personal information into ciphertext, while Poly1305 produces an authentication tag that enables the system to verify data authenticity. Technical testing then determines whether the functions operate as intended, while the final stage evaluates whether the technical outcomes support the relevant obligations under the Personal Data Protection Law.

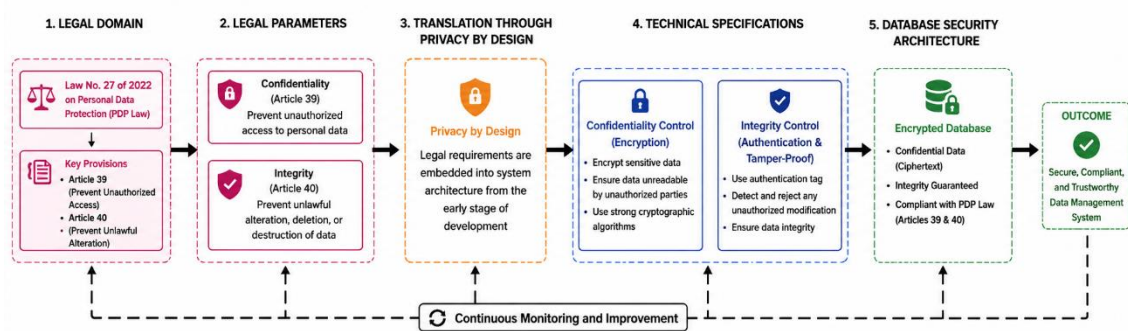


Figure 3. Conceptual Architecture of the Juridical–Technical Model for Translating Law No. 27 of 2022 into an Encryption-Based Database Security Framework through the Privacy by Design Approach.

Source: Research Results.

Based on Figure 3, the architecture of the conceptual model is explained through the following stages.

1. Legal and Regulatory Domain

This stage constitutes the legal foundation of the model and is derived from Law Number 27 of 2022 concerning Personal Data Protection. Article 35 establishes the requirement for risk-based technical and operational safeguards. Article 36 establishes the confidentiality obligation. Article 38 requires protection against unlawful processing, while Article 39 requires Personal Data Controllers to prevent unlawful access through reliable, secure, and accountable electronic systems.

These obligations form the normative basis for constructing the proposed technical security parameters. Integrity and authenticity are positioned as technical safeguards that support the implementation of these statutory obligations.

2. Conceptual Bridge

At this stage, the legal requirements are analyzed through the *Privacy by Design* framework. This framework functions as a conceptual bridge that converts juridical obligations into system requirements and architecture. Rather than treating privacy as an additional feature, the model incorporates personal-data protection into the system from the initial design stage.

3. Technical Implementation

The security requirements formulated in the previous stage are implemented through software engineering. At the application level, the XChaCha20-Poly1305 algorithm processes directly identifiable personal data before database storage. The resulting encrypted data are stored as ciphertext, replacing the plaintext format used in the existing system.

The application also performs authenticated decryption. This mechanism allows valid information to be restored when the correct key and unmodified ciphertext are supplied while rejecting records that have been altered or authenticated incorrectly.

4. Validation Stage

This stage evaluates whether the encryption, decryption, storage, retrieval, and integrity-verification functions operate correctly. Black-box testing is used to assess whether the system produces the expected output for each test scenario. The results are then compared with the juridical–technical parameters derived from the Personal Data Protection Law.

5. Compliance Support Evaluation

The final stage assesses the extent to which the technical implementation supports the statutory requirements. Successful functionality testing provides evidence that the model

implements relevant technical safeguards. However, it does not automatically establish complete organizational compliance because broader obligations, including governance, accountability, access policies, incident response, consent management, audit trails, and institutional supervision, must also be fulfilled.

Database Structure of the Membership Table

The technical implementation commenced with the redesign of the relational database using the MySQL Database Management System. The schema of the `m_anggota` membership table was reviewed by considering the personal-data classifications established under Article 4 of Law Number 27 of 2022 and the operational risks associated with each attribute.

The law distinguishes personal data into specific personal data and general personal data. Specific personal data include health information, biometric data, genetic data, criminal records, children's data, personal financial data, and other categories stipulated by legislation. General personal data include full name, gender, nationality, religion, marital status, and combined data capable of identifying an individual (Republic of Indonesia, 2022).

Therefore, religion should be classified as general personal data rather than specific personal data. A national identification number is also better described in this table as general personal data functioning as a direct identifier. Although it is not categorized as specific personal data under Article 4, it requires strong technical protection because it can uniquely identify an individual and may cause substantial harm if disclosed.

The classification of data as general personal data does not mean that encryption is unnecessary. The decision to apply encryption should also consider identification value, processing context, potential harm, and security risk. Accordingly, the proposed model applies enhanced protection to directly identifiable and operationally sensitive data, particularly national identification numbers and other information selected through the system's risk assessment.

Table 1. Structure of the `m_anggota` Database Table and Classification of Personal Data under Law No. 27 of 2022

Field Name	Data Type	Length	Classification under the PDP Law
<code>member_id</code>	VARCHAR	20	Internal Identifier
<code>national_id_number (NIK)</code>	VARCHAR	255	General Personal Data
<code>full_name</code>	VARCHAR	50	General Personal Data
<code>place_of_birth</code>	VARCHAR	30	General Personal Data
<code>date_of_birth</code>	DATE	–	General Personal Data
<code>gender</code>	VARCHAR	5	General Personal Data
<code>address</code>	VARCHAR	100	General Personal Data
<code>neighborhood_code (RT/RW)</code>	VARCHAR	10	General Personal Data
<code>regency/city, district, village</code>	VARCHAR	30	General Personal Data
<code>postal_code</code>	VARCHAR	10	General Personal Data
<code>phone_number</code>	VARCHAR	14	General Personal Data
<code>religion</code>	VARCHAR	20	General Personal Data
<code>occupation</code>	VARCHAR	20	General Personal Data
<code>marital_status</code>	VARCHAR	20	General Personal Data
<code>data_entry_officer_id</code>	VARCHAR	40	Internal Identifier
<code>entry_date</code>	DATETIME	–	System Metadata
<code>status</code>	VARCHAR	5	System Metadata

Source: Research Results.

The length of the `national_id_number` field was increased to accommodate the Base64-encoded combination of the nonce, ciphertext, and authentication tag. Under the redesigned architecture, the original national identification number is encrypted before it is written to the

database. Other fields may also be encrypted based on the organization's risk assessment, processing requirements, and need for search or reporting functionality.

Implementation of Cryptographic Simulation

The proposed model applies XChaCha20-Poly1305 to protect directly identifiable personal information against unauthorized disclosure and manipulation. This algorithm was selected because the Authenticated Encryption with Associated Data mechanism provided by Libsodium combines encryption and authentication within a single cryptographic process.

The algorithm provides two principal protection mechanisms:

1. Confidentiality. XChaCha20 transforms plaintext into ciphertext so that the stored information cannot be read directly by unauthorized parties.
2. Integrity and authenticity verification. Poly1305 generates an authentication tag associated with the ciphertext. If the encrypted information is modified, authenticated decryption fails and the application rejects the record.

The mechanism therefore protects data not only against direct disclosure but also against undetected manipulation. However, the term *tamper-proof* should be understood as the ability to detect unauthorized alteration rather than an absolute guarantee that alteration is physically impossible. An attacker may still modify the stored ciphertext, but the system is designed to detect the modification and prevent the altered record from being accepted as authentic.

The implementation can be observed through the data-protection simulation interface. The simulation demonstrates how a national identification number entered as plaintext is converted into an encrypted representation.

Data Protection Simulation (PDP Law)

Enter National Identity Number (Plaintext):
4738347282

Encrypt Now (Secure Data)

Ciphertext Result:
SkxFAQCxbtWkaym9JyQRMronwxaSoga/oOTjVmUsC0cXMIEGYdfASKYvu58PyuTjRyM=

Enter Ciphertext (Encrypted Text):
SkxFAQCxbtWkaym9JyQRMronwxaSoga/oOTjVmUsC0cXMIEGYdfASKYvu58PyuTjRyM=

Decrypt Now (Retrieve Data)

Figure 4. Implementation of Data Protection Simulation

Source: Research Results.

As shown in Figure 4, the identity number entered into the application is processed by the XChaCha20-Poly1305 cryptographic function. A unique nonce is generated for the encryption transaction. The plaintext is then encrypted, authenticated, combined with the nonce, and converted into Base64 format before being displayed or stored.

Base64 encoding itself does not provide security because it only converts binary output into a text-compatible representation. The actual security protection is provided by XChaCha20-Poly1305. The apparently random sequence displayed in the output field represents Base64-encoded cryptographic data consisting of the nonce and authenticated ciphertext.

To restore the information, the application performs authenticated decryption using the corresponding symmetric key. Only a process possessing the valid key and unmodified

encrypted data can successfully restore the original value. If the key is incorrect or the encrypted value has been modified, the authentication process fails.

Implementation of the Cryptographic Algorithm at the Application Level

The XChaCha20-Poly1305 algorithm was implemented in the backend of the cooperative membership-registration application. Every protected value entered through the application was processed through the authenticated-encryption function before database storage. This process enabled confidentiality and integrity verification to be applied automatically without requiring users to perform encryption manually.

The implementation code used in the simulation prototype is presented in Table 2. The code is retained because it corresponds directly with the application-side implementation displayed in Figure 5.

Table 2. Implementation Code of the Cryptographic Function

```
1. <?php
2.
3. define('KUNCI_RAHASIA', 'kunci-rahasia-simulasi-pdp-2024!');
4.
5. $aksi = $_POST['aksi'] ?? '';
6. $data = $_POST['data'] ?? '';
7.
8. if ($aksi == 'enkripsi') {
9.
10.     $nonce = random_bytes(SODIUM_CRYPTO_AEAD_XCHACHA20POLY1305_IETF_NPUBBYTES);
11.
12.
13.     $ciphertext = sodium_crypto_aead_xchacha20poly1305_ietf_encrypt(
14.         $data,
15.         '',
16.         $nonce,
17.         KUNCI_RAHASIA
18.     );
19.
20.
21.     echo base64_encode($nonce . $ciphertext);
22.
23. } elseif ($aksi == 'dekripsi') {
24.     try {
25.
26.         $decoded = base64_decode($data);
27.
28.
29.         $nonce_len = SODIUM_CRYPTO_AEAD_XCHACHA20POLY1305_IETF_NPUBBYTES;
30.         $nonce = mb_substr($decoded, 0, $nonce_len, '8bit');
31.         $ciphertext = mb_substr($decoded, $nonce_len, null, '8bit');
32.
33.
34.         $decrypted = sodium_crypto_aead_xchacha20poly1305_ietf_decrypt(
35.             $ciphertext,
36.             '',
37.             $nonce,
38.             KUNCI_RAHASIA
39.         );
40.
41.         if ($decrypted === false) {
42.             echo "Gagal: Kunci salah atau data rusak!";
43.         } else {
```

```

44.     echo $decrypted;
45.   }
46. } catch (Exception $e) {
47.     echo "Error: Data tidak valid!";
48. }
49. }
50. ?>
51.

```

Source: Research Implementation.

The code demonstrates the core functionality of the Juridical–Technical Model. The encryption function generates a unique nonce, encrypts the submitted value, combines the nonce and ciphertext, and converts the output into Base64 format. The decryption function decodes the stored value, separates the nonce from the ciphertext, and performs authenticated decryption.

The key shown in Table 2 is a simulation key used exclusively for prototype development and functionality testing. Hard-coding a secret key in application source code is not an acceptable production key-management practice. In an operational environment, the cryptographic key should be separated from the application source code and encrypted database, restricted to authorized processes, and stored using a protected key-management service, environment-secret mechanism, secure vault, or Hardware Security Module.

Operational key management should include secure key generation, role-based access, access logging, periodic rotation, protected backup and recovery, revocation, and secure destruction. The protection of encrypted information ultimately depends on the protection of its cryptographic keys (Benadjila et al., 2022; Radanliev, 2024). A Hardware Security Module was not implemented in this prototype and is therefore presented as a recommendation for production deployment rather than as a tested component.

The implementation represents a practical translation of legal security requirements into executable system behaviour. Nevertheless, it should be described as a technical mechanism that supports compliance, rather than as proof that encryption alone produces complete legal compliance. Organizational compliance also depends on governance, policies, accountability, authorization, supervision, incident management, consent management, and other legal obligations.

The screenshot displays the KOP-SMART SECURE application interface. On the left, the 'Add New Member' form is visible, showing fields for Member No. (KOP-2024-0002), ID No. / NIK (XXXXXXXXXXXX), Full Name (Budi Santoso), Place of Birth (Bandung), Date of Birth (06/01/1990), Gender (Male), Phone No. / HP (08XXXXXXXX), Complete Address (Jl. Merdeka No. 123, Bandung, West Java), RT (01), and RW (02). A green notification bar at the top states 'Member data has been successfully saved and encrypted securely!'. On the right, the 'Member List' table shows a single entry for Budi Santoso with member ID KOP-2024-0001, NIK XXXXXXXXXX, and status Active. On the far right, a detailed view of the member's data is shown, including fields like member_no, id_no, name, place_of_birth, date_of_birth, gender, address, rt, rw, city_regency, subdistrict, village, postal_code, phone_no, religion, occupation, marital_status, inputter_id, input_date, and status, all containing encrypted data.

Figure 5. Implementation of the Cryptographic Algorithm on the Application Side

Source: Research Results.

Figure 5 shows the implementation of the encryption and decryption functions within the cooperative membership application. The application automatically transforms protected input into ciphertext before storage and performs authenticated decryption when information is legitimately requested.

Technical Functionality Testing

Technical functionality testing was conducted to evaluate the operation of the Juridical–Technical Model. Black-box testing focused on the behaviour of the application in response to defined inputs and test scenarios. The evaluation examined three principal security functions: encryption, authorized decryption, and integrity verification.

The encryption test commenced by entering a national identification number through the cooperative membership-registration form. The application automatically executed the XChaCha20-Poly1305 function before the value was written to the MySQL database. The result demonstrated that the original identity number was replaced by Base64-encoded ciphertext and could no longer be identified directly from the database.

The decryption test assessed whether the authorized application process could restore the encrypted information. When the correct symmetric key and unmodified ciphertext were supplied, the application successfully recovered the original identity number. This result confirmed that encrypted information remained usable for legitimate organizational purposes.

The integrity test was conducted by manually modifying one character of the ciphertext in the development database. This scenario simulated unauthorized data alteration. When the modified ciphertext was submitted for decryption, the Poly1305 authentication verification failed, and the system rejected the record.

The failure response demonstrated that authenticated encryption prevented manipulated ciphertext from being silently converted into incorrect personal information. The function therefore operated as an authentication seal capable of detecting unauthorized changes. This finding is consistent with the authenticated-encryption construction described by Nir and Langley (2018).

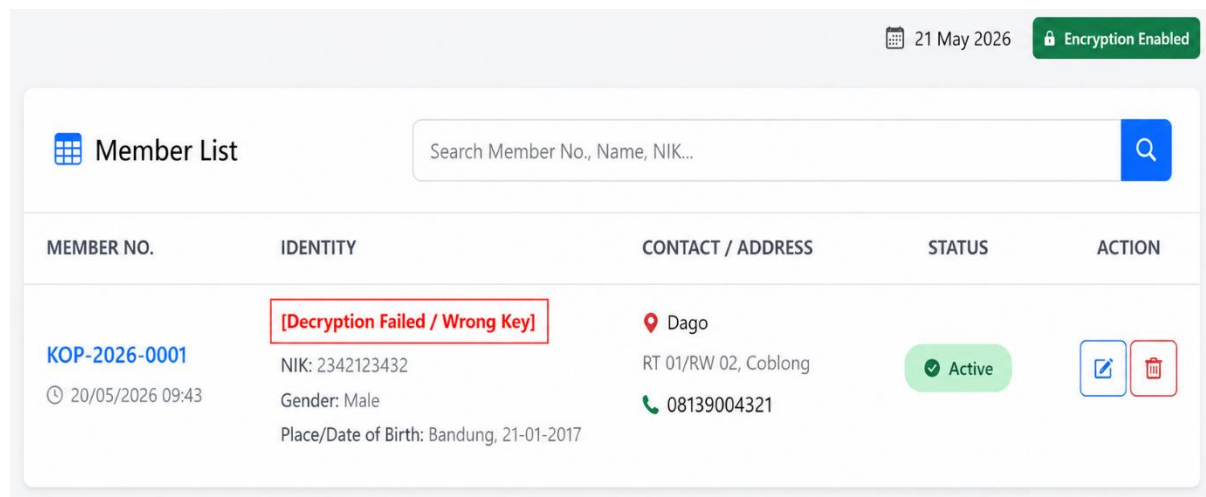
Table 3. System Testing Results

Testing Scenario	Procedure	Expected Result	Status
Encryption Test	Enter the original National Identity Number (NIK) in the registration form.	The data is stored as random text in the database.	Successful
Decryption Test	Retrieve member data from the dashboard.	The random text is converted back into the readable original NIK.	Successful
Integrity Test	Manually modify the data stored in the database.	The system rejects the decryption process and displays an error message: “Invalid data.”	Successful

Source: Research Results.

The testing results confirmed that the three primary functions operated according to the expected behaviour. Nevertheless, the tests demonstrate functional feasibility rather than complete cybersecurity assurance. The study did not include large-scale performance benchmarking, source-code security auditing, penetration testing, side-channel analysis, or testing against all possible categories of cyberattack.

The implementation of the integrity test is presented in Figure 6.



MEMBER NO.	IDENTITY	CONTACT / ADDRESS	STATUS	ACTION
KOP-2026-0001 20/05/2026 09:43	[Decryption Failed / Wrong Key] NIK: 2342123432 Gender: Male Place/Date of Birth: Bandung, 21-01-2017	Dago RT 01/RW 02, Coblong 08139004321	Active	 

Figure 6. Technical Functionality Testing
Source: Research Results.

Figure 6 demonstrates the relationship between unauthorized ciphertext modification and authentication failure. The database value could be altered at the storage level, but the altered record could not pass the Poly1305 verification process. Consequently, the application rejected the compromised data instead of presenting it as authentic information.

Evaluation of Legal-Compliance Support

The evaluation assessed the extent to which the developed Juridical–Technical Model technically supports the obligations established under Law Number 27 of 2022. The assessment does not claim that cryptographic functionality alone proves complete legal compliance. Instead, it identifies the relationship between the tested functions and specific statutory security obligations.

1. Support for Technical and Operational Security Measures under Article 35

Article 35 requires Personal Data Controllers to implement technical and operational measures and determine a security level based on the nature and risks of the data being processed. The proposed model supports this obligation by identifying high-risk personal identifiers and applying field-level authenticated encryption before database storage.

The implementation changes the existing storage architecture from plaintext storage into an encryption-based system. Even when unauthorized database access occurs, the protected value is not immediately intelligible without the corresponding cryptographic key. The mechanism therefore constitutes one component of a risk-based technical safeguard.

However, Article 35 cannot be fulfilled by encryption alone. Operational implementation must also include access-control policies, key-management procedures, monitoring, personnel authorization, secure system configuration, backup protection, and incident-response mechanisms.

2. Support for Confidentiality under Article 36

Article 36 requires Personal Data Controllers to maintain the confidentiality of personal data. XChaCha20 supports this obligation by converting readable information into ciphertext before storage. The testing demonstrated that the national identification number could not be directly recognized from the database.

Confidentiality remains dependent on the secrecy of the cryptographic key. Therefore, production deployment must separate the key from the application source code and database and restrict key access to authorized processes or personnel.

3. Support for Protection against Unlawful Processing under Article 38

Article 38 requires Personal Data Controllers to protect personal data against unlawful processing. Authenticated decryption supports this requirement by preventing modified or unauthenticated encrypted records from being treated as valid information.

The integrity test demonstrated that unauthorized modification caused authentication failure. The system therefore prevented the altered ciphertext from being processed into a misleading readable value. Nevertheless, broader controls are still required to prevent unlawful collection, disclosure, transfer, profiling, deletion, and other forms of processing.

4. Support for the Prevention of Unlawful Access under Article 39

Article 39 requires Personal Data Controllers to prevent unlawful access by implementing reliable, secure, and accountable electronic security systems. Field-level encryption provides an additional security layer beyond ordinary usernames, passwords, and database permissions.

If unauthorized parties gain access to the database, encrypted values remain unintelligible without the corresponding key. Therefore, the model reduces the consequences of storage-layer compromise. However, encryption must be integrated with authentication, authorization, monitoring, access logging, secure software development, and vulnerability management.

5. Protection of Integrity and Authenticity

The Poly1305 authentication mechanism enables the application to detect unauthorized modification of encrypted data. The technical evaluation showed that changing one character of the ciphertext caused decryption to fail. This outcome supports the preservation of data authenticity and consistency.

Integrity is treated as a technical safeguard supporting Articles 35, 38, and 39. It should not be attributed to Article 40 because Article 40 regulates the termination of processing following withdrawal of consent.

6. Transformation of Digital Data Management

The proposed model demonstrates the integration of juridical and technical dimensions within cooperative data management. Legal obligations are not treated exclusively as documentary or administrative requirements but are translated into executable system functions.

This transformation strengthens the cooperative information system by adding cryptographic protection at the data-storage layer. The resulting architecture supports confidentiality, detects unauthorized modification, and reduces the exposure of directly identifiable personal information. Nevertheless, the model should be integrated with comprehensive organizational governance to establish a complete personal-data-protection framework.

Discussion

The findings demonstrate that the proposed Juridical–Technical Model is capable of translating selected obligations under Law Number 27 of 2022 into concrete technical safeguards within a cooperative membership information system. The implementation of XChaCha20-Poly1305 transformed the previous database architecture from plaintext storage into an authenticated encryption-based model.

This transformation indicates that personal-data protection cannot rely solely on administrative policies, consent documents, usernames, passwords, or procedural compliance. Legal obligations should also be reflected in the technical design and behaviour of information systems. The model therefore treats personal-data protection as a system requirement rather than as an external administrative addition.

The findings are consistent with Leite et al. (2022), who found that data-protection regulations influence software-engineering practices, particularly requirements identification,

system design, and process modelling. Negri-Ribalta et al. (2024) similarly emphasized that regulatory data-protection requirements must be translated into precise and operational system requirements. The present study extends these arguments by demonstrating how selected provisions of Indonesia's Personal Data Protection Law can be operationalized within the database and application layers of a cooperative information system.

From the confidentiality perspective, XChaCha20 ensured that the national identification number was no longer stored in a directly readable format. The conversion of plaintext into authenticated ciphertext prevented the original information from being immediately identified when the database was viewed. This result is aligned with the authenticated-encryption construction described by Nir and Langley (2018), in which ChaCha20 performs the encryption function and Poly1305 provides message authentication.

Base64 encoding should not be confused with encryption. It merely provides a text-compatible format for storing the binary output consisting of the nonce, ciphertext, and authentication tag. The confidentiality result originates from the cryptographic algorithm and the secrecy of the key rather than from the Base64 representation.

The integrity test demonstrated that the system could identify unauthorized modification of encrypted information. When one character was changed, the authentication process failed, and the application rejected the record. This result confirms that Poly1305 functioned as a cryptographic authentication mechanism that protected the authenticity of the stored information. The finding supports Degabriele et al. (2021), who examined the security of ChaCha20-Poly1305 within a multi-user setting.

The test also clarifies the meaning of *tamper detection*. The model does not prevent an attacker with database access from physically altering ciphertext. Instead, it prevents the modified ciphertext from being accepted as authentic during decryption. Therefore, the more precise description of the model is tamper-evident authenticated storage, rather than an absolute claim that the data cannot be altered.

The findings strengthen the relevance of *Privacy by Design* and *Data Protection by Design* in software development. Andrade et al. (2022) argued that privacy safeguards should be incorporated throughout the software-development lifecycle. Alkubaisy et al. (2022) similarly emphasized that privacy and security requirements should be analyzed from the early stages of system development to support regulatory compliance.

The present study provides a practical implementation of these principles within the Indonesian legal context. Security requirements derived from the Personal Data Protection Law were translated into database redesign, encryption, authenticated decryption, integrity verification, and testing functions. This approach illustrates how legal requirements can influence the behaviour of an information system.

The principal contribution of the study lies in integrating legal analysis and software-engineering implementation. In conventional organizational practice, compliance is frequently interpreted as the preparation of privacy notices, consent forms, internal policies, and administrative documentation. Although these instruments remain important, the findings demonstrate that selected legal obligations can also be internalized into the technical architecture.

In this model, the law is not merely treated as a set of external rules. It is translated into executable technical behaviour: the application encrypts information before storage, limits successful retrieval to valid cryptographic conditions, and rejects manipulated ciphertext. This embedded protection remains active even when direct human supervision is limited.

Within cooperative data management, the model provides an additional layer beyond conventional access controls. Access controls determine who may enter the system and what functions they may perform. Encryption protects the stored information when the storage layer

is accessed outside the authorized application process. The two mechanisms are complementary and should not be treated as substitutes.

The findings are particularly relevant for cooperatives and other organizations that process personal information but possess limited cybersecurity infrastructure. The prototype demonstrates that modern authenticated encryption can be integrated into a PHP- and MySQL-based application using the Libsodium library. Nevertheless, operational deployment requires stronger key management, server hardening, role-based access control, audit logging, secure backup procedures, employee training, and incident response.

The code used in the prototype contains a hard-coded simulation key. This arrangement was sufficient for controlled functional testing but is not appropriate for production use. A compromised source-code repository or server file could expose both the encrypted data and the key required to decrypt them. The model must therefore be extended through protected secret storage, key separation, access logging, rotation, revocation, and recovery procedures.

For higher-risk implementation, a dedicated key-management service or Hardware Security Module may strengthen protection by isolating key operations from ordinary application memory and database storage. However, this study did not implement or test a Hardware Security Module. Its effectiveness within the proposed architecture should therefore be examined through further research rather than presented as an achieved result.

The evaluation also indicates that successful encryption testing should not be equated with comprehensive legal compliance. Law Number 27 of 2022 establishes broader responsibilities relating to lawful processing, transparency, data-subject rights, consent, accountability, processing records, breach notification, supervision, and other governance dimensions. The proposed model addresses a specific technical component of that broader framework.

The model's novelty lies in the systematic pathway through which Indonesian statutory provisions are converted into measurable technical parameters, application functions, and testing criteria. Previous studies have frequently examined legal compliance and cryptographic performance separately. This study connects the legal domain, *Privacy by Design*, software architecture, authenticated encryption, technical validation, and compliance-support evaluation within a single framework.

The resulting contribution is therefore not limited to the selection of XChaCha20-Poly1305. Its primary contribution is the Juridical-Technical Model through which legal requirements are interpreted, operationalized, implemented, tested, and evaluated. This model may be adapted by cooperatives, micro and small enterprises, software developers, data controllers, and policymakers when developing practical technical guidance for organizations with limited cybersecurity resources.

The study nevertheless has several limitations. First, the prototype was implemented and tested only within the context of cooperative membership registration. The results cannot automatically be generalized to larger information systems, financial platforms, healthcare systems, or organizations with substantially different risk profiles.

Second, the technical testing focused on functional encryption, decryption, and integrity verification. The study did not conduct large-scale performance benchmarking, penetration testing, cryptographic side-channel analysis, source-code security auditing, or comparison with alternative authenticated-encryption algorithms.

Third, the prototype primarily addressed confidentiality and integrity verification. Other dimensions of personal-data governance, including secure key management, Hardware Security Module integration, detailed audit trails, role-based access-control policies, processing records, data-retention controls, incident-response procedures, consent withdrawal, and automated compliance monitoring, require further development.

Overall, the findings indicate that the Juridical–Technical Model provides a viable framework for translating selected personal-data-protection obligations into authenticated encryption-based controls. The implementation of XChaCha20-Poly1305 enabled the cooperative information system to conceal directly identifiable data, restore valid encrypted records through authorized processes, and reject manipulated ciphertext. The model therefore strengthens the technical foundation of cooperative data governance while recognizing that cryptographic protection must operate as part of a broader organizational and legal compliance framework.

CONCLUSION AND RECOMMENDATIONS

Conclusion

This study concludes that the proposed Juridical–Technical Model provides a feasible framework for translating selected security obligations under Indonesia’s Law Number 27 of 2022 concerning Personal Data Protection into executable safeguards within a cooperative membership information system. The model bridges normative legal requirements and software-engineering implementation by integrating *Privacy by Design*, database redesign, authenticated encryption, and technical functionality testing into a unified development framework.

The implementation of XChaCha20-Poly1305 successfully transformed the existing data-storage architecture from plaintext storage into authenticated ciphertext storage. Functional testing demonstrated that personal information entered through the application was encrypted before being stored in the database and could only be restored through an authorized decryption process using the corresponding cryptographic key. The integrity test further showed that unauthorized modification of the stored ciphertext caused authentication failure and resulted in the rejection of the altered record. Therefore, the system provided confidentiality, authenticity verification, and tamper-evident protection rather than merely concealing personal information.

From a juridical perspective, these technical functions support the implementation of Article 35 concerning technical and operational security measures, Article 36 concerning the confidentiality of personal data, Article 38 concerning protection against unlawful processing, and Article 39 concerning the prevention of unlawful access. However, the successful implementation of encryption should not be interpreted as proof of complete organizational compliance with the Personal Data Protection Law. Comprehensive compliance also requires lawful processing, transparency, protection of data-subject rights, accountability mechanisms, access-control policies, incident response, breach notification, consent management, processing records, institutional supervision, and secure cryptographic-key governance.

The principal theoretical contribution of this study lies in the development of a Juridical–Technical Model that connects legal interpretation, *Privacy by Design*, software architecture, authenticated encryption, technical validation, and compliance-support evaluation within a single framework. Unlike approaches that examine legal regulation and cybersecurity separately, this model demonstrates how statutory security obligations can be converted into measurable technical parameters and executable application behaviour.

Practically, the model provides a replicable foundation for cooperatives and other personal data controllers seeking to strengthen the protection of directly identifiable information. By embedding authenticated encryption into the database-storage process, organizations can reduce the consequences of unauthorized database access, detect manipulated encrypted records, and establish a more resilient digital data-management architecture. The model may therefore contribute to strengthening digital trust and supporting secure, accountable, and privacy-preserving data governance within Indonesia’s digital economy.

Limitations

This study has several limitations. First, the prototype was developed and tested only within the context of a cooperative membership-registration system. The findings therefore cannot automatically be generalized to larger information systems, financial platforms, healthcare systems, government databases, or organizations with different technological architectures and risk profiles.

Second, the technical evaluation was limited to black-box functionality testing involving encryption, authorized decryption, and ciphertext-integrity verification. The study did not include large-scale performance benchmarking, penetration testing, source-code security auditing, side-channel analysis, vulnerability scanning, or comparative testing with alternative authenticated-encryption algorithms. Consequently, the results demonstrate functional feasibility but do not establish complete resistance to all forms of cyberattack.

Third, the prototype primarily addressed confidentiality and integrity verification at the database-storage layer. Other important dimensions of personal-data governance, including secure key management, role-based access control, audit trails, data-retention management, processing records, consent withdrawal, incident response, breach notification, backup security, and automated compliance monitoring, were not fully implemented.

Fourth, the cryptographic key used in the prototype was a simulation key embedded for controlled functional testing. This arrangement is not appropriate for operational deployment. Although secure key-management services and Hardware Security Modules were identified as potential development mechanisms, these technologies were not implemented or evaluated in the present study.

Recommendations

Cooperative organizations and other personal data controllers are recommended to integrate authenticated encryption with broader technical and organizational security controls. The implementation of XChaCha20-Poly1305 should be accompanied by role-based access restrictions, multifactor authentication, access logging, secure software configuration, protected backups, vulnerability management, employee training, and incident-response procedures. Encryption should be treated as one component of a layered data-protection architecture rather than as a substitute for comprehensive information-security governance.

For operational deployment, cryptographic keys should be separated from application source code and encrypted databases. Organizations should establish formal procedures for secure key generation, storage, authorization, rotation, backup, recovery, revocation, and destruction. Cooperatives processing large volumes of personal or high-risk information should consider using a dedicated key-management service, secure secret vault, or Hardware Security Module to reduce the risk of unauthorized key exposure.

Future studies should test the model in larger and more diverse organizational environments, including financial cooperatives, microfinance institutions, micro, small, and medium enterprises, healthcare organizations, educational institutions, and government service systems. Further research should also conduct performance benchmarking, penetration testing, source-code review, comparative cryptographic analysis, and evaluation of secure key-management architectures. Integration with audit logs, role-based access control, automated compliance monitoring, incident notification, and data-retention management may further strengthen the model.

Regulators and relevant government institutions are recommended to develop sector-specific technical guidance for implementing Law Number 27 of 2022, particularly for cooperatives, micro, small, and medium enterprises, and organizations with limited cybersecurity infrastructure. Such guidance should establish minimum technical safeguards for encrypted storage, authenticated integrity verification, access control, key management, audit

logging, incident response, and periodic security evaluation. The Juridical–Technical Model proposed in this study may serve as an initial reference for preparing practical implementation standards that translate statutory obligations into measurable and accessible technical controls.

Through collaboration among regulators, cooperative authorities, software developers, cybersecurity professionals, and organizational data controllers, the proposed model can be further developed into a practical technical standard for secure digital-data governance. Such collaboration is necessary to ensure that personal-data protection is implemented not only through administrative documents but also through verifiable security mechanisms embedded directly within organizational information systems.

Author Contributions

All authors made substantial contributions to the completion of this study. The contributions included research conceptualization, development of the juridical–technical framework, analysis of Law Number 27 of 2022 concerning Personal Data Protection, system-requirement identification, database architecture design, prototype development, implementation of the XChaCha20-Poly1305 authenticated encryption algorithm, technical functionality testing, interpretation of the findings, manuscript preparation, critical revision, and final approval of the article. All authors have read and approved the final version of the manuscript and agree to be accountable for the accuracy and integrity of the research.

REFERENCES

- Alghamdi, Y., & Munir, A. (2024). Image encryption algorithms: A survey of design and evaluation metrics. *Journal of Cybersecurity and Privacy*, 4(1), 126–152. <https://doi.org/10.3390/jcp4010007>
- Al-Khater, W. A., Al-Maadeed, S., Ahmed, A. A., Sadiq, A. S., & Khan, M. K. (2020). Comprehensive review of cybercrime detection techniques. *IEEE Access*, 8, 137293–137311. <https://doi.org/10.1109/ACCESS.2020.3011259>
- Alkubaisy, D., Piras, L., Al-Obeidallah, M. G., Cox, K., & Mouratidis, H. (2022). A framework for privacy and security requirements analysis and conflict resolution for supporting GDPR compliance through privacy-by-design. In R. Ali, H. Kaindl, & L. A. Maciaszek (Eds.), *Evaluation of novel approaches to software engineering* (pp. 67–87). Springer. https://doi.org/10.1007/978-3-030-96648-5_4
- Andrade, V. C., Gomes, R. D., Reinehr, S., Freitas, C. O. A., & Malucelli, A. (2022). Privacy by design and software engineering: A systematic literature review. In *Proceedings of the XXI Brazilian Symposium on Software Quality*. Association for Computing Machinery. <https://doi.org/10.1145/3571473.3571480>
- Arif, T., Jo, B., & Park, J. H. (2025). A comprehensive survey of privacy-enhancing and trust-centric cloud-native security techniques against cyber threats. *Sensors*, 25(8), 2350. <https://doi.org/10.3390/s25082350>
- Atobishi, T., Moh'd Abu Bakir, S., & Nosratabadi, S. (2024). How do digital capabilities affect organizational performance in the public sector? The mediating role of organizational agility. *Administrative Sciences*, 14(2), 37. <https://doi.org/10.3390/admsci14020037>
- Benadjila, R., Khati, L., & Vergnaud, D. (2022). Secure storage—Confidentiality and authentication. *Computer Science Review*, 44, 100465. <https://doi.org/10.1016/j.cosrev.2022.100465>
- Bilan, Y., Oliynyk, O., Mishchuk, H., & Skare, M. (2023). Impact of information and communications technology on the development and use of knowledge. *Technological Forecasting and Social Change*, 191, 122519. <https://doi.org/10.1016/j.techfore.2023.122519>

- Chaudhuri, R., Chatterjee, S., Mariani, M. M., & Wamba, S. F. (2024). Assessing the influence of emerging technologies on organizational data-driven culture and innovation capabilities: A sustainability performance perspective. *Technological Forecasting and Social Change*, 200, 123165. <https://doi.org/10.1016/j.techfore.2023.123165>
- Das, D. K. (2024). Exploring the symbiotic relationship between digital transformation, infrastructure, service delivery, and governance for smart sustainable cities. *Smart Cities*, 7(2), 806–835. <https://doi.org/10.3390/smartcities7020034>
- Degabriele, J. P., Govinden, J., Günther, F., & Paterson, K. G. (2021). The security of ChaCha20-Poly1305 in the multi-user setting. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security* (pp. 1981–2003). Association for Computing Machinery. <https://doi.org/10.1145/3460120.3484814>
- Dhirani, L. L., Mukhtiar, N., Chowdhry, B. S., & Newe, T. (2023). Ethical dilemmas and privacy issues in emerging technologies: A review. *Sensors*, 23(3), 1151. <https://doi.org/10.3390/s23031151>
- Georgiadis, G., & Poels, G. (2021). Enterprise architecture management as a solution for addressing General Data Protection Regulation requirements in a big data context: A systematic mapping study. *Information Systems and E-Business Management*, 19(1), 313–362. <https://doi.org/10.1007/s10257-020-00500-5>
- Hale, M. L., & Gamble, R. F. (2019). Semantic hierarchies for extracting, modeling, and connecting compliance requirements in information security control standards. *Requirements Engineering*, 24(3), 365–402. <https://doi.org/10.1007/s00766-017-0287-5>
- Ibrahim, A. M., Abdel-Aziz, H. R., Mohamed, H. A. H., Zaghamir, D. E. F., Wahba, N. M. I., Hassan, G. A., Shaban, M., El-Nablaway, M., Aldughmi, O. N., & Aboelola, T. H. (2024). Balancing confidentiality and care coordination: Challenges in patient privacy. *BMC Nursing*, 23(1), 564. <https://doi.org/10.1186/s12912-024-02231-1>
- Iman, N. (2024). The fight for our personal data: Analyzing the economics of data and privacy on digital platforms. *International Journal of Law and Management*, 66(6), 774–791. <https://doi.org/10.1108/IJLMA-12-2022-0258>
- Kementerian Komunikasi dan Informatika Republik Indonesia. (2024, May 30). Wamen Nezar Patria ajak tingkatkan keamanan siber semua sektor digital. <https://www.komdigi.go.id/berita/pengumuman/detail/siaran-pers-no-372-hm-kominfo-05-2024-tentang-wamen-nezar-patria-ajak-tingkatkan-keamanan-siber-semua-sektor-digital>
- Leite, L., dos Santos, D. R., & Almeida, F. (2022). The impact of General Data Protection Regulation on software engineering practices. *Information & Computer Security*, 30(1), 79–96. <https://doi.org/10.1108/ICS-03-2020-0043>
- Naguib, A., & Fouad, K. M. (2024). Database security: Current challenges and effective protection strategies. In *2024 6th International Conference on Computing and Informatics (ICCI)* (pp. 120–130). IEEE. <https://doi.org/10.1109/ICCI61671.2024.10485182>
- Negri-Ribalta, C., Lombard-Platet, M., & Salinesi, C. (2024). Understanding the GDPR from a requirements engineering perspective: A systematic mapping study on regulatory data protection requirements. *Requirements Engineering*, 29, 523–549. <https://doi.org/10.1007/s00766-024-00423-4>
- Nir, Y., & Langley, A. (2018). *ChaCha20 and Poly1305 for IETF protocols* (RFC 8439). RFC Editor. <https://doi.org/10.17487/RFC8439>
- Radanliev, P. (2024). Digital security by design. *Security Journal*, 37(4), 1640–1679. <https://doi.org/10.1057/s41284-024-00435-3>

- Ramakrishna, D., & Ali Shaik, M. (2025). A comprehensive analysis of cryptographic algorithms: Evaluating security, efficiency, and future challenges. *IEEE Access*, 13, 11576–11593. <https://doi.org/10.1109/ACCESS.2024.3518533>
- Republic of Indonesia. (2022). *Undang-Undang Republik Indonesia Nomor 27 Tahun 2022 tentang Pelindungan Data Pribadi [Law of the Republic of Indonesia Number 27 of 2022 concerning Personal Data Protection]*. <https://peraturan.bpk.go.id/Details/229798/uu-no-27-tahun-2022>
- Saleem, F., Salim, N., Altalhi, A. H., Ullah, Z., Al-Malaise Al-Ghamdi, A., & Mahmood Khan, Z. (2020). Assessing the effects of information and communication technologies on organizational development: Business values perspectives. *Information Technology for Development*, 26(1), 54–88. <https://doi.org/10.1080/02681102.2017.1335279>
- Samaraweera, G. D., & Chang, J. M. (2021). Security and privacy implications on database systems in the big data era: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 33(1), 239–258. <https://doi.org/10.1109/TKDE.2019.2929794>
- Sánchez, M. (2022). A general approach on privacy and its implications in the digital economy. *Journal of Economic Issues*, 56(1), 244–258. <https://doi.org/10.1080/00213624.2022.2025729>
- Scheibner, J., Ienca, M., Kechagia, S., Troncoso-Pastoriza, J. R., Raisaro, J. L., Hubaux, J.-P., Fellay, J., & Vayena, E. (2020). Data protection and ethics requirements for multisite research with health data: A comparative examination of legislative governance frameworks and the role of data protection technologies. *Journal of Law and the Biosciences*, 7(1), lsaa010. <https://doi.org/10.1093/jlb/lsaa010>
- Sekretariat Kabinet Republik Indonesia. (2022, March 28). *Presiden Jokowi targetkan 20 juta UMKM masuk toko daring di tahun 2022*. <https://setkab.go.id/presiden-jokowi-targetkan-20-juta-umkm-masuk-toko-daring-di-tahun-2022/>
- Shojaei, P., Vlahu-Gjorgievska, E., & Chow, Y.-W. (2024). Security and privacy of technologies in health information systems: A systematic literature review. *Computers*, 13(2), 41. <https://doi.org/10.3390/computers13020041>
- Shukla, S., Bisht, K., Tiwari, K., & Bashir, S. (2023). *Data economy in the digital age*. Springer Nature Singapore. <https://doi.org/10.1007/978-981-99-7677-5>
- Sim, J., Kim, B., Jeon, K., Joo, M., Lim, J., Lee, J., & Choo, K.-K. R. (2023). Technical requirements and approaches in personal data control. *ACM Computing Surveys*, 55(9), 1–30. <https://doi.org/10.1145/3558766>
- Sun, N., Li, C.-T., Chan, H., Le, B. D., Islam, M. Z., Zhang, L. Y., Islam, M. R., & Armstrong, W. (2022). Defining security requirements with the Common Criteria: Applications, adoptions, and challenges. *IEEE Access*, 10, 44756–44777. <https://doi.org/10.1109/ACCESS.2022.3168716>
- Yeung, K., & Bygrave, L. A. (2022). Demystifying the modernized European data protection regime: Cross-disciplinary insights from legal and regulatory governance scholarship. *Regulation & Governance*, 16(1), 137–155. <https://doi.org/10.1111/rego.12401>
- Zaeem, R. N., & Barber, K. S. (2025). Economics of cybercrime: Identity theft and fraud. In *Encyclopedia of cryptography, security and privacy* (pp. 732–734). Springer Nature Switzerland. https://doi.org/10.1007/978-3-030-71522-9_1613