A Systematic Literature Review on Security and Privacy Design Patterns

Ebtehal Aljedaani, Maha Aljohani

Abstract**—**Privacy and security patterns are both important for developing software that protects users' data and privacy. Privacy patterns are designed to address common privacy problems, such as unauthorized data collection and disclosure. Security patterns are designed to protect software from attack and ensure reliability and trustworthiness. Using privacy and security patterns, software engineers can implement security and privacy by design principles, which means that security and privacy are considered throughout the software development process. These patterns are available to translate "security and privacy-by-design" into practical advice for software engineering. Previous research on privacy and security patterns has typically focused on one category of patterns at a time. This paper aims to bridge this gap by merging the two categories and identifying their similarities and differences. To do this, we conducted a systematic literature review of 40 research papers on privacy and security patterns. The papers were analyzed based on the category of the pattern, the classification of the pattern, and the security requirements that the pattern addresses. This paper presents the results of a comprehensive review of privacy and security design patterns. The review is intended to help future IT designers understand the relationship between the two types of patterns and how to use them to design secure and privacy-preserving software. The paper provides a clear classification of privacy and security design patterns, along with examples of each type. We found that there is only one widely accepted classification of privacy design patterns, while there are several competing classifications of security design patterns. Three types of security design patterns were found to be the most used.

Keywords**—**Design patterns, security, privacy, classification of patterns, security patterns, privacy patterns.

I. Introduction

RECENT scholarly attention has gravitated toward design, site engineering, and application development aimed at addressing prevalent security and privacy concerns. While some researchers have directed their focus solely on security issues and [1], [2], others exclusively on privacy matters [3], [4], the convergence of these domains remains relatively uncommon.

In response to the challenges posed by this divide, designers have formulated standardized solutions to address common security and privacy issues, termed as privacy and security design patterns. This paper seeks to delve into these design patterns, elucidate their interrelationships, and provide guidance to designers. While existing literature extensively covers privacy or security patterns in diverse contexts, it has been observed that no prior study has comprehensively amalgamated both types of patterns. Most scholarly works have tended to emphasize either privacy or security, neglecting a unified approach.

The subsequent sections of this research are structured as follows: Section II presents the background, Section III encompasses the reviewed related work, Section IV outlines the research methodology, Sections V and VI encompass the results and discussions, and Section VII encapsulates the conclusion.

II. Background

Christopher Alexander was a pioneer in introducing the concept of patterns within building architecture and subsequently applied this concept to the object-oriented environment [5]. Since 1994, design patterns have been extensively employed in software development to address persistent issues [6]. These Software Design Patterns (SDPs) serve as foundational solutions for recurring software challenges, aiding developers in the selection of optimal designs that enhance system reusability [7]. Each pattern elucidates a solution to commonly encountered problems in software development, offering a defined resolution that can be repeatedly applied without redundancy [8].

The discipline of Human-Computer Interaction (HCI) has witnessed a significant surge in privacy-related research, notably since the early 1990s, experiencing further notable growth in contemporary times. This heightened focus is substantiated by the proliferation of dedicated sessions addressing privacy concerns at HCI conferences and the emergence of specialized conferences such as the Symposium on Usable Privacy and Security (SOUPS), collectively illustrating the escalating scholarly attention directed towards privacy matters within the realm of HCI [9]. In defining privacy, [10] delineates it as "the individual's capacity to govern the conditions under which their personal data is obtained and utilized." Conversely, [9] characterizes privacy as the affirmation that individuals, institutions, or other entities possess the entitlement to manage the dissemination, timing, and extent of information shared about them. Nonetheless, the author of [11] identifies three pivotal aspects from the initial definition:

Firstly, privacy is rooted in the management of information flow, indicating a correlation between privacy concerns and issues within HCI. Secondly, both privacy and security revolve around assessing and regulating risk perception, particularly when past actions can lead to unforeseen future consequences, which may or may not align with prevailing standards of harm

E. Aljedaani and M. Aljohani are with Computer Science and Artificial Intelligence, Department, University of Jeddah, Jeddah KSA (e-mail: Ealjeddani0001.stu@uj.edu.sa, mmaljohani@uj.edu.sa).

or safety. Lastly, in societal contexts, privacy embodies control, trust, and empowerment. Consequently, it engenders ethical, political, and legal deliberations [11].

The concept of "Privacy by Design" (PbD) advocates integrating privacy considerations into all stages of system design. It is a set of principles endorsed by a unanimous resolution at the International Data Protection assembly. PbD's seven core principles emphasize proactive prevention of privacy breaches, default privacy settings, integrated privacy in system design, accommodating all interests without trade-offs, end-to-end security, transparency, and user-centric privacy prioritization [12], [13].

Privacy design patterns represent universal and reusable software designs devised to address specific privacy protection challenges within a defined setting. Each pattern is tasked with delineating the contextual parameters and intricacies inherent in resolving these concerns. Noteworthy among these patterns are user data confinement, asynchronous notice, and location granularity, recognized for their prominence. In a study [14], an observed disparity between HCI design methodologies and privacy-by-design approaches emerged, pinpointing contextual intricacies, particularly those pertaining to privacy, as the crux of the issue. This gap was identified by Mulligan and King in their work documented in [15]. As a solution, they advocated for the development of privacy patterns, positing it as a pivotal tool to bridge this schism. These patterns were envisaged as a means to transmute abstract concepts fostering privacy advocacy into practical engineering techniques effectuating its realization. Moreover, these proposed patterns were positioned as a "bottom-up" mechanism, adept at concretely manifesting privacy-by-design principles, thereby serving as a robust approach capable of delineating problems and corresponding solutions across diverse contextual landscapes [15].

Considering the escalating frequency of electronic assaults targeting organizations in recent times, the fortification of information security has emerged as a pivotal facet within enterprises. The fundamental aim of security design patterns resides in bolstering software resilience against prevalent attacks and instances of misuse. Consequently, there arises a necessity for security engineers and designers to rely upon solutions that mitigate the incidence of assaults on organizational systems. Hence, these security design patterns offer tailored solutions contingent upon the contextual exigencies, furnishing validated and established strategies to address recurrent security challenges [16].

III. Related Work

The corpus of available literature contains a considerable volume of scholarly articles focusing on patterns related to privacy and security. A multitude of approaches have been identified, encompassing articles that contextualize patterns pertinent to privacy, security, or both within the frameworks of their respective systems. Additionally, scholarly inquiries have contributed studies that either propose new patterns or seek to augment existing ones. However, an analysis of prior research underscores discernible differences in the manner of presentation and proposition of these patterns.

The subsequent scholarly works primarily concentrate on the introduction and exploration of novel security patterns within distinct technological domains. One such endeavor [16] contributes a novel collection of user-centric security patterns tailored specifically for social media platforms. These patterns are designed to fortify user privacy and security within interactive social environments while emphasizing ease of use for developers. The study introduced four distinctive patterns as remedies to recurrent security challenges encountered within social networking sites. Their evaluation involved a comparative analysis against Facebook interfaces, demonstrating higher user acceptance of the proposed interface integrated with the new patterns compared to the established Facebook interfaces.

In a different domain, the work detailed in [17] delves into security patterns pertinent to Cloud Software as a Service (SaaS). The research endeavors to address a spectrum of security concerns encompassing system security, data security, and privacy issues. An examination of security patterns offered by Amazon Web Services (AWS) was conducted, aligning these patterns with corresponding solutions within AWS infrastructure. Additionally, the study presents the elucidation of each selected security pattern, offering a structured representation classified at a high-level taxonomy.

Furthermore, a separate investigation, documented in [18], conducts an empirical study to ascertain the efficacy of incorporating security patterns in achieving a more secure software environment. This study involved 64 participants enrolled in a software architecture course. Divided into two groups, participants were provided with a catalog comprising 36 security patterns and their respective solutions to fulfill design requirements. The research involved three phases: a training phase, a phase without security patterns, and a phase integrating security patterns. Findings suggest a tendency among designers, even those without specialized security expertise, to intuitively select solutions akin to established security patterns.

Moreover, [19] establishes a correlation between security patterns and essential security properties such as authentication, integrity, authorization, confidentiality, among others. The study categorizes security patterns based on their abstraction levels into architectural, design, and idiomatic patterns. A collection of security patterns is presented, each designed to address one or more specific security requirements. The research critically analyzes software security services, design patterns, and architectural patterns to discern relationships among these entities.

Several practical papers have undertaken diverse classification approaches for security patterns, as observed in [20], which classified patterns according to the layers of the Open Systems Interconnection (OSI) model.

Conversely, [6] views security patterns from an architectural perspective and explores their connection with software architecture classifications. This study introduces the concept of "Similar Forces" relations among patterns, signifying their interconnectedness and ability to produce analogous security outcomes, exemplified by patterns like Checkpoint patterns

Open Science Index, Computer and Systems Engineering Vol:18, No:8, 2024 publications.waset.org/10013769.pdf

mirroring effects similar to Full View with Error, Single Access Point, and Session patterns.

In contrast, several studies have focused on refining and enhancing privacy patterns, exemplified by research conducted by [21]. Their work aimed to enhance a set of privacy patterns by constructing a pattern system, facilitating the identification of contextually relevant patterns. Moreover, this system aimed to aid software developers in elucidating the handling of personal data within information systems. A notable aspect of their approach was the capacity to revise patterns to align with specific requirements, drawn from patterns cataloged by a privacy pattern collaboration. Furthermore, their proposed system sought to render these patterns implementable, coherent, well-structured, and interconnected, aligning with the requirements of Pattern-Oriented System Architecture (POSA) to ensure their effective utilization within the intended contexts. Additionally, this system facilitated improvements in selected patterns during implementation while elucidating interconnections between these patterns.

Another study by [22] amalgamated selected patterns into cohesive groups and subsequently enhanced these subgroups by constructing a pattern system. Their system aimed to identify contextually suitable patterns, demonstrate their usage and relationships, and provide guidance for software developers. To achieve these objectives, they endeavored to adapt patterns to fulfill specific contextual needs, particularly emphasizing the imperative of keeping users informed—a primary objective of their collection of privacy patterns designed for companies affected by General Data Protection Regulation (GDPR). Their pattern system adhered to Pattern-Oriented Software Architecture (POSA) standards and presented a pattern for user control, broadening the scope of the existing pattern system.

Additionally, [23] proposed a framework aimed at enhancing the application of patterns by scrutinizing the affected aspects during pattern application and outlining future steps in this domain. Employing various research methods such as systematic database searches and theoretical approaches, their framework consisted of two parts: an extraction process and an application process. The former involved pattern discovery, composition, organization, review, and publication, while the latter entailed contextual and problem recognition, pattern selection, instantiation, and evaluation. Their findings underscored significant opportunities for further research in this realm, emphasizing the necessity to encourage the adoption of privacy patterns and associated engineering strategies and tools.

Furthermore, [24] provided an overview of privacy transparency patterns, focusing on enhancing the transparency of privacy practices. By delineating two privacy patterns—the Privacy Policy Icons and the Personal Data Table—and categorizing them into four classifications, namely generic privacy information, real-time data insight, privacy awareness, and privacy marks, they aimed to offer comprehensive descriptions for each pattern, including specific contextual aspects such as user and application context.

In a separate investigation, [25] addressed concerns pertaining to the design of privacy-preserving systems and outlined nine privacy design patterns applicable across various domains, including Anonymity set, Morphed representation, Hidden metadata, Layered encryption, Cover traffic, Batched routing, Delayed routing, Constant length padding, and Constant link padding.

Conversely, [26] introduced a novel user control pattern system derived from an existing privacy pattern catalog, intending to streamline privacy patterns by ensuring coherence, consistency, maturity, and relevance. Employing 20 privacy patterns sourced from privacypatterns.org and incorporating control classifications from Hoepman's strategies, the system aligned with POSA requirements, delineating six specific requisites.

Moreover, another study [27] proposed seven privacy patterns tailored for mobile operating systems, presented alongside the RePa Requirements Pattern Template. Their approach encompassed two stages: first, identifying privacy requirements for mobile operating systems, and subsequently recognizing privacy patterns. These patterns included Authorized use of sensors or portals, Avoidance of privacy leakage in user behavior information collection, Guard for personal mobile data, Privacy protection over mobile cloud services, Authentication of mobile users, Financial information protection, and Mobile communication secrecy. Notably, Patterns 6 and 5 exhibited relevance to Pattern 7 due to the shared necessity for secure transport methods in both financial applications and cloud services, as elucidated in Pattern 7.

The data in Table I offer a comprehensive overview of the studies under examination. Nevertheless, it is notable that the entirety of these studies concentrates solely on either privacy or security patterns. In contrast, the present research presents a model aiming to integrate and amalgamate these distinct patterns.

IV. Research Method

In this study, we adopted the Systematic Literature Review (SLR) methodology, adhering to the prescribed steps delineated in the literature [28], comprising three principal phases: planning, conducting, and documenting the review process. As depicted in Fig. 1, each phase entails a sequence of specific activities. Within the initial phase, the focus is on formulating research inquiries, devising the review protocol, and validating this protocol. Subsequently, in the second phase, the emphasis shifts towards identifying pertinent literature, selecting primary studies, evaluating their methodological quality, extracting essential data, and synthesizing the gathered information. Moving to the third phase, the attention centers on generating a comprehensive review report and ensuring its validation. Finally, the ultimate step involves an in-depth discussion and interpretation of the outcomes, analyzing the primary studies thoroughly to address the research inquiries at hand.

A. Research Questions

In this section, following the SLR methodology, the subsequent Research Questions (RQs) have been formulated to be addressed in the ensuing sections:

RQ1. What is the scope of approved classifications for privacy and security design patterns?

RQ2. What shared attributes exist between privacy and security design patterns, and which aspects represent their primary intersections?

TABLE I
SUMMARY OF EXAMINED STUDIES

Reference	Purpose	Pattern used (security or privacy)	Propose new pattern or use existing pattern
[16]	They provided new understandable security patterns to help social media developers when creating interactive environments.	4 security patterns	Propose new patterns
[17]	They explored the security pattern for cloud SaaS.	31 security patterns	Propose new patterns
[18]	They performed an empirical study to investigate whether such an audience's use of security patterns results in a more secure environment.	36 security patterns	Use existing patterns
[19]	They explained the relationship between security patterns and security properties.	27 security patterns	Use existing patterns
[20]	They classified the patterns depending on the network layer.	8 security patterns	Classified existing patterns
[6]	They view security patterns as architectural patterns, thus they looked at software architecture classifications.	8 security patterns	Classified existing patterns
[21]	They improve a set of privacy patterns by building a pattern system that helps find patterns suited for the context.	72 privacy patterns	Use existing patterns (improve)
[22]	They combined selected patterns into valuable groups, and then improved subgroups from these by building a pattern system.	31 inform privacy patterns	Use existing patterns (improve)
[23]	They proposed a framework to enhance the application of patterns.	Privacy patterns	Use existing patterns
[24]	They showed patterns of privacy transparency overview, which they concentrate on finding ways to make privacy more transparent.	2 privacy patterns	Use existing patterns
[25]	Describe in detail the existing privacy patterns with examples.	9 privacy patterns	Use existing patterns
[26]	They offered a new user control pattern system based on an existing privacy pattern catalog.	20 privacy patterns	Based on existing patterns, propose new patterns
[27]	Propose seven privacy patterns for the mobile operating systems and offered with RePa Requirements Pattern Template.	7 privacy patterns	Propose new patterns

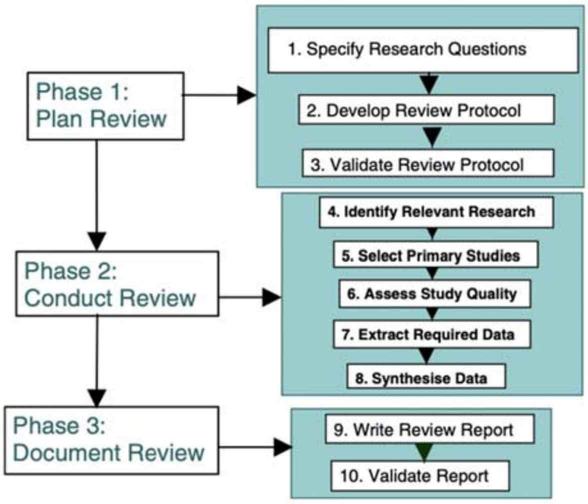


Fig. 1 Three primary phases of SLR [28]

The formulation of the first question (RQ1) stems from the aspiration to identify and elucidate the prevailing classifications associated with privacy and security design patterns, aiming to ascertain the central convergence between security and privacy paradigms. Conversely, the second inquiry (RQ2) emanates from an intent to discern common attributes that establish a nexus between privacy and security, thereby elucidating the nature of this juncture. Ultimately, these RQs aim to explicate the interface between privacy and security and pinpoint their intersecting elements.

B. Searching Strategy

1) Libraries

This systematic review exclusively utilizes electronic resources accessible through the Saudi Digital Library, accessible via the University of Jeddah's Blackboard platform. The search strategy involved querying specific databases housed within the digital library, notably encompassing IEEE, ACM, Science Direct, IEICE Electronics Express, Applied Science, Springer, among others. Moreover, supplementary relevant publications may be sourced from external research databases such as Google Scholar, Research Gate, and Semantic Scholar. The search protocol detailed in this review will be employed within the digital library using predefined search queries, with the selection of outcomes guided by the criteria set forth [9].

2) Search Queries

To ensure effective retrieval of relevant materials, our search strategy necessitates a balance between generality to encompass information tangential yet pertinent to the study topic and specificity to minimize unrelated articles. This entails a thorough examination of titles and abstracts among hundreds of articles, assessing their suitability for subsequent stages. Emphasizing the syntax of search queries becomes imperative to yield precise outcomes. Various syntax conventions, such as quotation marks for exact searches, logical connectors like 'AND' or 'OR,' and query structuring using brackets, are employed in several digital libraries to refine searches. Additionally, considerations extend to parameters like publication type, publication year, and occasionally, the publication's subject domain, exemplified by "computer information science." This study specifically targets computer science articles while focusing on privacy-preserving techniques and security measures within the scope of publications dated between 2006 and 2022 [19].

The search methodology encompasses the following keywords and phrases:

1. Security AND Design AND Patterns
2. Security AND Design AND Patterns AND Security requirement
3. Privacy AND Design AND Patterns
4. Privacy- preserving AND Privacy-patterns
5. Privacy AND Principle
6. Privacy OR Security AND Design AND Patterns
7. Security AND Pattern AND Software AND Development

3) Primary Study Selection

The final list of Primary Study Selection (PSS) is generated by limiting the list of search studies according to a set of criteria. Finding and analyzing the initial list of PSS articles is essential in the first stage. We gathered 40 articles in the first collected list of PSs articles. This list was filtered according to a set of rules and procedures. The first filter focuses on deleting the articles outside our topic range. The second filter is to retain only English-language-supporting items. The articles that respond to the RQs should be maintained on the list for the third filter. 25 articles are obtained as the result of this process.

V.Result

This section will present and discuss the primary studies to answer the RQs.

A. Overview of the PSs

The advantage of using design patterns of privacy and security is keeping the system safe from any outside attacks. Therefore, these presented design patterns will assist Software Engineers in maintaining their designs within the security and privacy framework, especially when dealing with sensitive information.

B. RQ1: How Many Classifications Are Approved for Privacy and Security Design Patterns?

Design patterns in software development are categorized into three primary groups: creational, structural, and behavioral patterns. Creational patterns focus on object creation mechanisms, ensuring objects are created in a way that is suitable for specific situations. Examples of creational patterns include Abstract Factory, Builder, and Singleton [30].

Structural patterns address the composition of classes and objects to form larger structures while maintaining flexibility and efficiency. Common examples of structural patterns include Adapter, Composite, and Decorator. These patterns are crucial in building systems that are both scalable and maintainable [30].

Behavioral patterns concentrate on object interaction and responsibility distribution, facilitating communication and collaboration among objects. Examples of behavioral patterns include Chain of Responsibility, Command, and Observer. These patterns ensure that the system's behavior is dynamic and adaptable. These classifications and their detailed descriptions are comprehensively presented in [30].

The classification of security patterns reveals multiple categorizations, unlike privacy patterns, which generally adhere to a singular classification. Various studies provide different perspectives on security pattern classification. For instance, one study [24] focuses on safeguarding applications within the network layer of the OSI Model, presenting diverse security patterns tailored for networked applications. Another approach [31] categorizes security patterns based on software levels' structures into three tiers: Architectural-Level, Design-Level, and Implementation-Level.

An alternative perspective [31] partitions patterns based on the Characterization of Security NFRs (Non-Functional Requirements), encompassing aspects such as confidentiality, integrity, availability, non-repudiation, auditability, accountability, authorization, and authentication. This approach organizes security design patterns based on either the Characterization of Security NFRs or Structure Level. For instance, in [27], security patterns are delineated concerning their compatibility with different structural levels and the corresponding security characterizations achieved. This is substantiated through case studies assessing the Limited View design pattern's efficacy in meeting NFRs.

On the other hand, [32] functions as an extensive repository elucidating 23 Gang of Four (GOF) design patterns. It meticulously details each pattern with comprehensive explanations and illustrative examples, systematically categorizing them into Creational, Structural, and Behavioral design classifications. This catalog provides in-depth insights into their functional objectives, structural compositions, and practical applications. As such, it stands as an invaluable asset for software developers seeking comprehensive guidance.

In scientific discourse, an exploration of design pattern classification was detailed in [33]. The article delineated three primary categories for design patterns: Creational, Structural, and Behavioral. Creational patterns were depicted as focusing on object creation mechanisms, while Structural patterns concerned themselves with class and object composition, and Behavioral patterns centered on object interaction and communication.

Reference [34] explores the organization of security patterns in software development based on their life cycle phase, problem, and abstraction level. It emphasizes the importance of understanding the relationships among these patterns and proposes a classification system using standard methodologies. Architects have introduced various schemes, including those based on applicability, product, and process, while layered systems often classify security patterns according to system tiers. The paper suggests a classification scheme rooted in domain-level concepts to aid in pattern mining and navigation. Additionally, it mentions the Zachman framework, introduced in 1987, which outlines architectural views and levels of the information model.

Furthermore, [35] offered an in-depth analysis of various security design patterns, examining their effectiveness in addressing security threats within software systems. It categorizes these patterns based on their applications and specific security requirements they fulfill, providing a structured framework for evaluating their strengths and weaknesses. The study aims to assist software developers in integrating robust security measures into their designs by offering clear insights into the practical use and benefits of each pattern. This systematic evaluation helps in making informed decisions to enhance the security posture of software systems.

The research paper [2] acknowledges limitations in current security pattern classification. While not explicitly listing them, the paper critiques classifications based solely on threats (like STRIDE), development phases, or application contexts. These approaches struggle with patterns that address multiple concerns. The research proposes a new, multi-dimensional classification that considers factors like threat models and application contexts to create a more effective way to organize security patterns for easier use by developers.

Contrarily, the classification of privacy patterns lacks the uniformity observed in security pattern classifications. Instead, some research segregates patterns based on privacy strategies. Notably, the privacypatterns.org platform categorizes patterns akin to privacy strategies. For example, in [3], eight privacy design strategies—MINIMISE, HIDE, SEPARATE, AGGREGATE, INFORM, CONTROL, ENFORCE, and DEMONSTRATE—are presented, offering a framework for classifying privacy design patterns within the software development life cycle. This absence of standardized classifications for privacy patterns contrasts sharply with the diverse classifications observed in security design patterns.

Additionally, in [36], they presented a catalog that introduces a classification scheme organizing patterns according to their contextual usage, application permissions, and hierarchical interrelationships based on their level of generality. Nevertheless, the current framework suggests potential enhancements, including the addition of category, permission, and granularity filters. These improvements are planned for future integration to enhance and fine-tune the pattern classification system.

Furthermore, [4] enhances the definitions of these strategies and introduces a new level of abstraction termed 'tactics' to bridge the disparity between legal requirements and system development practices. These tactics complement the strategies, offering engineering methodologies for PbD. Additionally, the paper delves into the associations between strategies and GDPR entities, offering practical personal data processing examples. It also explores the correlations between these strategies and diverse activities influencing personal data.

Fig. 2 presents the classification scheme for classifying the security and privacy design patterns.

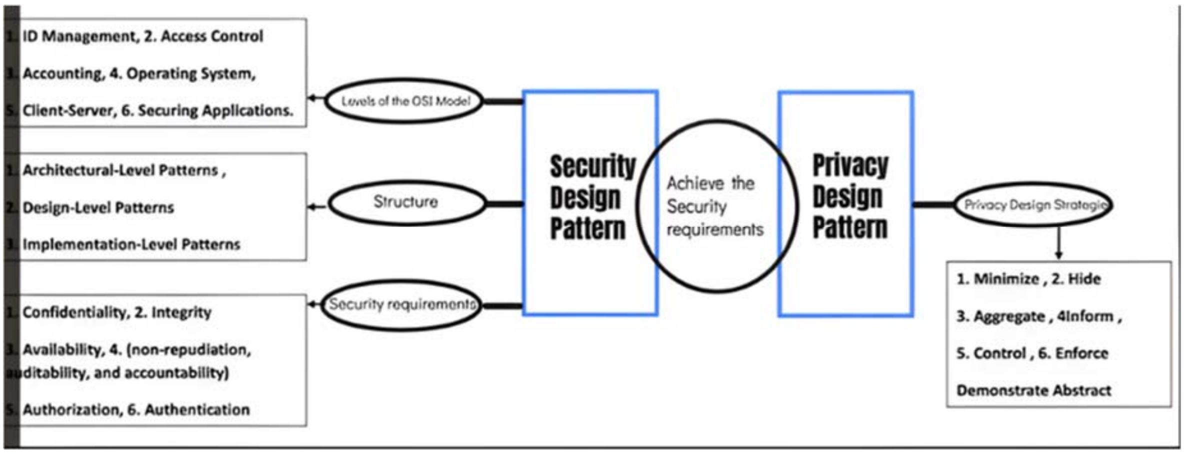

1. ID Management, 2. Access Control
3. Accounting, 4. Operating System,
5. Client-Server, 6. Securing Applications.
Levels of the OSI Model
1. Architectural-Level Patterns ,
2. Design-Level Patterns
3. Implementation-Level Patterns
Structure
1. Confidentiality, 2. Integrity
3. Availability, 4. (non-repudiation, auditability, and accountability)
5. Authorization, 6. Authentication
Security requirements
Security Design Pattern
Achieve the Security requirements
Privacy Design Pattern
Privacy Design Strategie
1. Minimize , 2. Hide
3. Aggregate , 4Inform ,
5. Control , 6. Enforce
Demonstrate Abstract

Fig. 2 Most classifications of privacy and security patterns

C. RQ2: What Are the Common Properties between Privacy and Security Design Patterns, and the Most Intersection Points between Them?

Security patterns and privacy patterns both serve to protect data and systems, providing structured frameworks for developers. They share similarities in their aim to safeguard information and mitigate threats, while aiding in regulatory compliance. However, they diverge in focus and implementation. Security patterns prioritize system integrity, availability, and confidentiality through measures like authentication and encryption. In contrast, privacy patterns concentrate on protecting personal information and ensuring lawful data processing, focusing on data minimization and consent management. While security patterns address broader aspects of system security, privacy patterns specifically ensure compliance with privacy regulations like GDPR and CCPA. Together, they form a comprehensive approach to data protection [37], [38].

It is generally accepted that privacy means preserving as much sensitive information as possible from being disclosed. In contrast, security focuses on maintaining data integrity and preventing any unauthorized external change. Therefore, what is typical between privacy and security is to keep data from being disclosed or modified, as this applies to well-known security requirements such as confidentiality, integrity, availability, etc.

In [38], Weiss and Mouratidis propose a systematic approach to align security patterns with security requirements in software systems. The method involves first identifying and documenting the security needs of the system, and then mapping these requirements to suitable security patterns using a structured framework. This framework ensures that the selected patterns effectively address the specified security needs while considering the trade-offs and implications for the system's architecture, performance, and usability. The goal is to integrate security considerations into the software development lifecycle, ensuring that the chosen patterns are appropriate and effective in meeting the system's security requirements.

The study of [40] aimed to make it possible for higher-level security characteristics to be broken down into more specific ones and then matched to applicable patterns. They described two types of patterns (security and privacy) for their context IoT. Figs. 3 and 4 show the relationship between privacy and security with the most common security requirements.

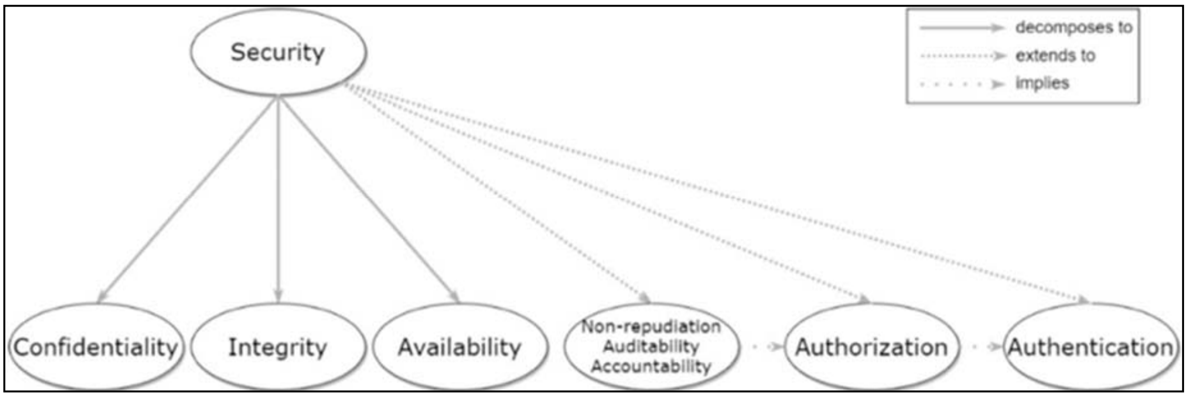


Fig. 3 Relationship between security with the most common security requirements [40]

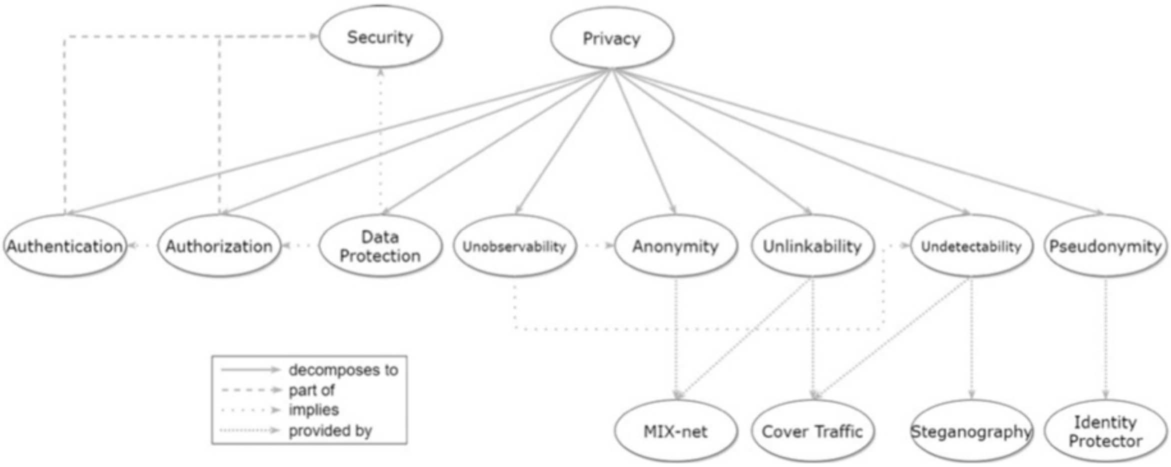


Fig. 4 Relationship between privacy with the most common security requirements [40]

Furthermore, [19] proposed a collection of security design patterns that somehow ensure one or more security requirement. Therefore, they matched one or several patterns for the related and ensured mentioned requirement. The relationship between security requirements, security services, security architectural patterns, and security design patterns is shown in Fig. 5. Another research [41] explained the security design patterns and their related security requirements. The difference between this research and the others is that the author clarified the relationship between the security patterns and the corresponding ones (level of software and the security requirements). Therefore, they combined two types of classification. Fig. 6 shows the relationship between security patterns with security requirements. Most research described the relationship between security patterns and security requirements, while privacy patterns are mentioned only in a few papers.

Table II shows the security requirements and each related pattern, either security or privacy, based on the results of the selected articles. Some security requirements have no direct privacy pattern or are reported in the research. Some privacy patterns may explain and fulfill the security requirement, but they are not directly mentioned in the research and approved sites. For example, the privacy pattern that may achieve the meaning of authorization security requirement can be Access Control.

Furthermore, we find the connection point between the patterns by understanding the relationship between privacy design strategies and security requirements, as each strategy may be achieved for a security requirement through the meaning of this strategy. For example, the two strategies, MINIMISE and HIDE can relate to the security requirements confidentiality and data protection. So, all privacy patterns of these two strategies can be related to the confidentiality and data protection requirements. Another example is the INFORM and CONTROL strategies that can relate to the integrity and non-repudiation security requirements. In this case, where any original data are by an unauthorized person, it will inform the administrator; this process can be achieved by controlling all systems. Therefore, the ENFORCE strategy relates to the authentication and authorization security requirements.

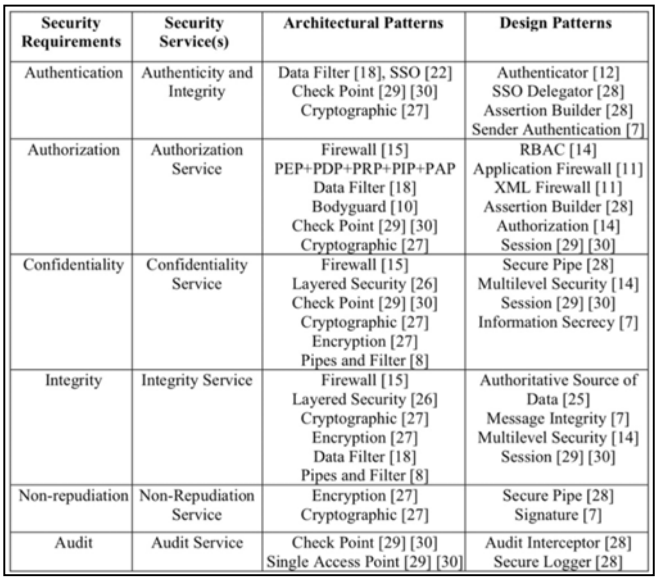
VI. Discussion

The results presented in the previous section will be comprehensively analyzed and elucidated in this section.

A. The Classification of Privacy and Security Design Patterns Related to RQ1

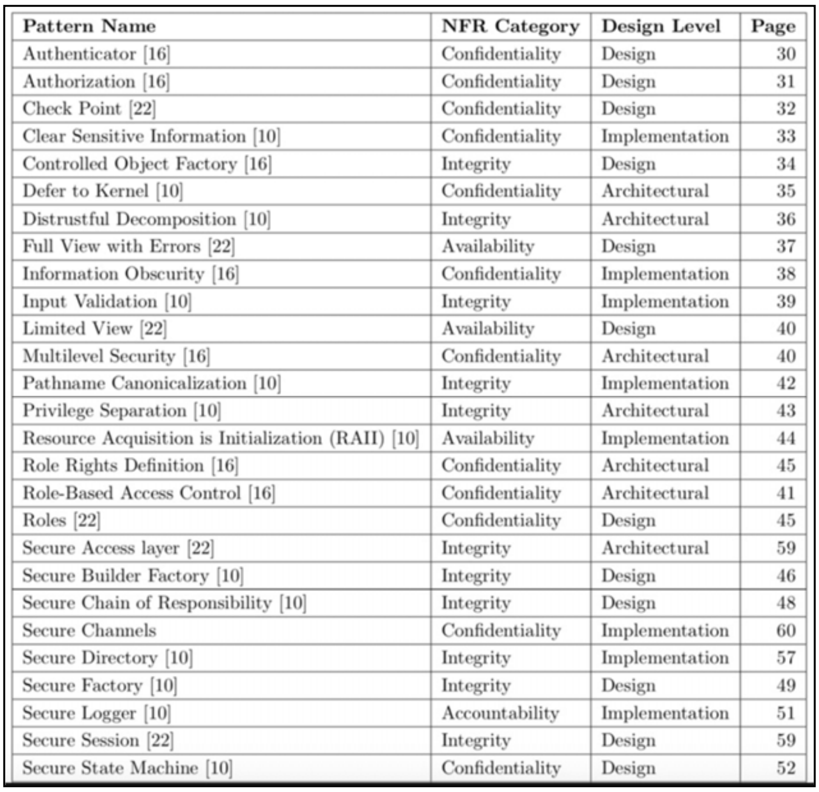
This discussion highlights the multifaceted categorization approaches in security patterns as opposed to the more singular classification structure observed in privacy patterns. Security patterns exhibit varying classifications across literature, such as segmentation based on network layers, software structure levels, and characterization of security NFRs. For instance, [20] focuses on safeguarding network applications, while [26] categorizes security patterns into architectural, design, and implementation levels.

Contrastingly, privacy pattern classifications lack uniformity and are often segregated based on privacy strategies. For instance, [39] introduces eight privacy design strategies for categorizing privacy patterns within the software development life cycle. In [30], a framework organizes patterns based on contextual usage, application permissions, and hierarchical relationships. Plans for future enhancements in this classification system include the integration of category, permission, and granularity filters.



Security Requirements	Security Service(s)	Architectural Patterns	Design Patterns
Authentication	Authenticity and Integrity	Data Filter [18], SSO [22] Check Point [29] [30] Cryptographic [27]	Authenticator [12] SSO Delegator [28] Assertion Builder [28] Sender Authentication [7]
Authorization	Authorization Service	Firewall [15] PEP+PDP+PRP+PIP+PAP Data Filter [18] Bodyguard [10] Check Point [29] [30] Cryptographic [27]	RBAC [14] Application Firewall [11] XML Firewall [11] Assertion Builder [28] Authorization [14] Session [29] [30]
Confidentiality	Confidentiality Service	Firewall [15] Layered Security [26] Check Point [29] [30] Cryptographic [27] Encryption [27] Pipes and Filter [8]	Secure Pipe [28] Multilevel Security [14] Session [29] [30] Information Secrecy [7]
Integrity	Integrity Service	Firewall [15] Layered Security [26] Cryptographic [27] Encryption [27] Data Filter [18] Pipes and Filter [8]	Authoritative Source of Data [25] Message Integrity [7] Multilevel Security [14] Session [29] [30]
Non-repudiation	Non-Repudiation Service	Encryption [27] Cryptographic [27]	Secure Pipe [28] Signature [7]
Audit	Audit Service	Check Point [29] [30] Single Access Point [29] [30]	Audit Interceptor [28] Secure Logger [28]

Fig. 5 Relation between security requirements and security patterns [19]



Pattern Name	NFR Category	Design Level	Page
Authenticator [16]	Confidentiality	Design	30
Authorization [16]	Confidentiality	Design	31
Check Point [22]	Confidentiality	Design	32
Clear Sensitive Information [10]	Confidentiality	Implementation	33
Controlled Object Factory [16]	Integrity	Design	34
Defer to Kernel [10]	Confidentiality	Architectural	35
Distrustful Decomposition [10]	Integrity	Architectural	36
Full View with Errors [22]	Availability	Design	37
Information Obscurity [16]	Confidentiality	Implementation	38
Input Validation [10]	Integrity	Implementation	39
Limited View [22]	Availability	Design	40
Multilevel Security [16]	Confidentiality	Architectural	40
Pathname Canonicalization [10]	Integrity	Implementation	42
Privilege Separation [10]	Integrity	Architectural	43
Resource Acquisition is Initialization (RAII) [10]	Availability	Implementation	44
Role Rights Definition [16]	Confidentiality	Architectural	45
Role-Based Access Control [16]	Confidentiality	Architectural	41
Roles [22]	Confidentiality	Design	45
Secure Access layer [22]	Integrity	Architectural	59
Secure Builder Factory [10]	Integrity	Design	46
Secure Chain of Responsibility [10]	Integrity	Design	48
Secure Channels	Confidentiality	Implementation	60
Secure Directory [10]	Integrity	Implementation	57
Secure Factory [10]	Integrity	Design	49
Secure Logger [10]	Accountability	Implementation	51
Secure Session [22]	Integrity	Design	59
Secure State Machine [10]	Confidentiality	Design	52

Fig. 6 Categorization of Security Design Patterns [19]

TABLE II
SECURITY REQUIREMENTS AND EACH RELATED PATTERN

Security requirements	Security pattern	Privacy pattern
Authentication	Data Filter, SSO, Check Point, Cryptographic.	Attribute Based Credentials, Unusual Activities.
Authorization	Authenticator, SSO Delegator, Assertion Builder, Sender Authentication.	Access control and Privacy Rights Management.
Data Protection	Firewall, Data Filter, Bodyguard, Check Point, Cryptographic.	Select Before You Collect, Anonymization, Use Pseudonyms, and Attribute based Credentials.
Confidentiality	RBAC, Application Firewall, XML Firewall, Assertion Builder, Authorization, Session.	Encryption, Mix Networks, Anonymous Cash, Attribute based Credentials, Unlinkability, Unobservability, Use Pseudonyms, and Select Before You Collect.
Integrity	Firewall, Layered Security, Cryptographic, Encryption, Data Filter, Pipes and Filter, Controlled Object Factory, Distrustful Decomposition, Input Validation, Privilege Separation, Secure Access Layer, Secure Builder Factory, Secure Chain of Responsibility, Secure Factory, Secure Directory.	Platform for Privacy Preferences (P3P) and Data Breach Notifications.
Availability	Firewall, Layered Security, Check Point, Cryptographic, Encryption, Pipes and Filter.	-
Non-repudiation	Secure Pipe, Multilevel Security, Session, Information Secrecy.	Platform for Privacy Preferences (P3P) and Data Breach Notifications.
Audit	Firewall, Layered Security, Cryptographic, Encryption, Data Filter, Pipes and Filter.	-

Furthermore, the discourse introduces a new level of abstraction termed 'tactics' in [31], bridging legal requirements with system development practices in privacy strategies. These techniques enhance strategies by providing engineering approaches for implementing PbD. This paper examines the connections between these strategies and GDPR entities, presenting practical examples of personal data processing. Ultimately, the diverse classifications in security patterns contrast sharply with the lack of standardized classifications in privacy patterns, presenting a potential area for future development and standardization.

B. The Common Properties between Privacy and Security Design Patterns and the Most Intersection Points between them Related to RQ2

As previously noted, a crucial nexus between security and privacy lies in their mutual alignment with security requirements. While security design patterns prominently incorporate these requirements as a primary classification, privacy patterns, in contrast, tend to embody the attributes derived from these requirements. The exploration of prevalent attributes shared between security and privacy patterns necessitates a comprehensive review of a diverse array of scholarly articles.

VII. CONCLUSION

In summary, this systematic review of literature has outlined the primary research articles concerning design patterns within the domains of security and privacy. The analysis conducted has illuminated that security design patterns are categorized into three distinct types: based on the OSI model layers, structural elements, and the specific security requisites they address. Conversely, privacy design patterns exhibit a singular classification type reliant on strategies pertaining to privacy.

In this comprehensive overview, the article delineates the classifications of both security and privacy design patterns. It notes the varied categorizations in security patterns, contrasting them with the more standardized classifications in privacy patterns. Several studies offer different perspectives on categorizing security patterns, including structural and NFR-based approaches. The classification of privacy patterns tends to align with privacy strategies rather than uniform structural categorizations. Notably, [42] presents eight privacy design strategies to organize patterns in the software development lifecycle. Additionally, a paper [4] introduces tactics as a means to bridge legal requirements with system development practices in privacy design. This discourse highlights the diverse approaches to classifying both security and privacy design patterns and their significance in software development.

Notably, all features related to privacy and security share a unified objective, namely, meeting the security requirements expounded upon in this study. Moreover, an intersection is discernible between privacy and security patterns, notably converging at the point of security requirements. This convergence signifies that privacy and security are interconnected, striving toward a shared aim of safeguarding data and limiting unauthorized access as comprehensively as feasible.

ACKNOWLEDGMENT

We extend gratitude to Dr. Maha Aljohani of the University of Jeddah for their invaluable assistance. Furthermore, we wish to convey profound appreciation to all individuals whose contributions were instrumental in the completion of this research endeavor.

REFERENCES

[1] E. B. Fernández, "Two Patterns for Web Services Security," 2004. (Online). Available: https://www.researchgate.net/publication/220968149

[2] M. Hafiz, P. Adamczyk, and R. E. Johnson, "Organizing security patterns," IEEE Softw, vol. 24, no. 4, pp. 52–60, Jul. 2007, doi: 10.1109/MS.2007.114.

[3] J. H. Hoepman, "Privacy design strategies," IFIP Adv Inf Commun Technol, vol. 428, pp. 446–459, 2014, doi: 10.1007/978-3-642-55415-5_38.

[4] M. Colesky, J.-H. Hoepman, and C. Hillen, "A Critical Analysis of Privacy Design Strategies," 2017.

[5] C. Alexander, A Pattern Language: Towns, Buildings, Construction. Oxford university press, 1977.

[6] E. B. Fernandez, H. Washizaki, N. Yoshioka, A. Kubo, and Y. Fukazawa,

"Classifying security patterns," Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 4976 LNCS, no. January 2016, pp. 342–347, 2008, doi: 10.1007/978-3-540-78849-2_35.

[7] W. Hussain, D. Mougouei, and J. Whittle, "Integrating social values into software design patterns," Proceedings - International Conference on Software Engineering, pp. 8–14, 2018, doi: 10.1145/3194770.3194777.

[8] M. Z. Asghar, K. A. Alam, and S. Javed, "Software design patterns recommendation: A systematic literature review," Proceedings - 2019 International Conference on Frontiers of Information Technology, FIT 2019, pp. 167–172, 2019, doi: 10.1109/FIT47737.2019.00040.

[9] A. F. Westin, "Privacy and Freedom." (Online). Available: https://scholarlycommons.law.wlu.edu/wlulr/vol25/iss1/20

[10] M. J. Culnan, Protecting Privacy Online: Is Self-Regulation Working? on JSTOR, vol. 19, no.1. Journal of Public Policy & Marketing, 2000. Accessed: Dec. 23, 2023. (Online). Available: https://www.jstor.org/stable/30000484

[11] S. Patil, N. Romero, and J. Karat, "Privacy and HCI: Methodologies for studying privacy issues," in Conference on Human Factors in Computing Systems - Proceedings, 2006, pp. 1719–1722. doi: 10.1145/1125451.1125771.

[12] S. Gürses, C. Troncoso, and C. Diaz, "Engineering Privacy by Design."

[13] P. Schaar, "Privacy by Design," Identity in the Information Society, vol. 3, no. 2, pp. 267–274, Aug. 2010, doi: 10.1007/s12394-010-0055-x.

[14] N. Doty and M. Gupta, "Privacy Design Patterns and Anti-Patterns: Patterns Misapplied and Unintended Consequences," A Turn for the Worse: Trustbusters for User Interfaces Workshop, pp. 1–5, 2013, (Online). Available: http://cups.cs.cmu.edu/soups/2013/trustbusters.html

[15] D. Mulligan and J. King, "Bridging the gap between privacy and design," U. Pa. J. Const. L., pp. 989–1034, 2011, (Online). Available: http://heinonlinebackup.com/hol-cgi-bin/get_pdf.cgi?handle=hein.journals/upjcl14&section=32

[16] R. Ortiz, S. Moral-García, S. Moral-Rubio, B. Vela, J. Garzás, and E. Fernández-Medina, "Applicability of security patterns," Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 6426 LNCS, no. PART 1, pp. 672–684, 2010, doi: 10.1007/978-3-642-16934-2_49.

[17] K. Alemerien, "User-friendly security patterns for designing social network websites," International Journal of Technology and Human Interaction, vol. 13, no. 1, pp. 39–60, 2017, doi: 10.4018/IJTHI.2017010103.

[18] K. Yskout, R. Scandariato, and W. Joosen, "Do security patterns really help designers?," Proceedings - International Conference on Software Engineering, vol. 1, pp. 292–302, 2015, doi: 10.1109/ICSE.2015.49.

[19] D. G. Rosado, C. Gutiérrez, E. Fernández-Medina, and M. Piattini, "Security patterns related to security requirements," Proceedings of the 4th International Workshop on Security in Information Systems, WOSIS 2006, in Conjunction with ICEIS 2006, no. January, pp. 163–173, 2006.

[20] S. Romanosky, "Security Design Patterns Part 1," Proceedings of PLoP, pp. 1–19, 2001, (Online). Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.13.7808&rep=rep1&type=pdf

[21] S. Y. Chia, X. Xu, H. Y. Paik, and L. Zhu, "Analysing and extending privacy patterns with architectural context," Proceedings of the ACM Symposium on Applied Computing, pp. 1390–1398, 2021, doi: 10.1145/3412841.3442014.

[22] M. Colesky and J. C. Caiza, "A system of privacy patterns for informing users: Creating a pattern system," ACM International Conference Proceeding Series, 2018, doi: 10.1145/3282308.3282325.

[23] J. C. Caiza, J. M. D. Alamo, and D. S. Guamán, "A framework and roadmap for enhancing the application of privacy design patterns," Proceedings of the ACM Symposium on Applied Computing, pp. 1297–1304, 2020, doi: 10.1145/3341105.3375768.

[24] J. Siljee, "Privacy transparency patterns," ACM International Conference Proceeding Series, vol. 08-12-July, 2015, doi: 10.1145/2855321.2855374.

[25] (25) M. Hafiz, "A collection of privacy design patterns," PLoP 2006 - PLoP Pattern Languages of Programs 2006 Conference Proceedings, pp. 1–26, 2006, doi: 10.1145/1415472.1415481.

[26] M. Colesky, J. C. Caiza, J. M. Del Lamo, J. H. Hoepman, and Y. S. Martín, "A system of privacy patterns for user control," Proceedings of the ACM Symposium on Applied Computing, pp. 1150–1156, 2018, doi: 10.1145/3167132.3167257.

[27] X. Xuan, Y. Wang, and S. Li, "Privacy requirements patterns for mobile operating systems," 2014 IEEE 4th International Workshop on Requirements Patterns, RePa 2014 - Proceedings, pp. 39–42, 2014, doi: 10.1109/RePa.2014.6894842.

[28] P. Brereton, B. A. Kitchenham, D. Budgen, M. Turner, and M. Khalil, "Lessons from applying the systematic literature review process within the software engineering domain," vol. 80, no. 4, pp. 571–583, 2007, doi: 10.1016/j.jss.2006.07.009.

[29] E. and H. R. and J. R. and V. J. Gamma, Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley Professional, 1994.

[30] C. Dougherty and R. C. Seacord, "Secure Design Patterns," Structure, no. October, 2009.

[31] H. Zhang, "Software Design and Patterns (Object Oriented Design)—Catalog of 23 GOF Design Patterns | by Hanwen Zhang | Medium." Accessed: Jan. 07, 2024. (Online). Available: https://hanwenzhang123.medium.com/software-design-and-patterns-catalog-of-23-gof-design-patterns-f336989f7d99

[32] R. Grimm, "Classification of Design Patterns." Accessed: Jan. 08, 2024. (Online). Available: https://www.linkedin.com/pulse/classification-design-patterns-rainer-grimm/

[33] A. K. Edinat, A. Hudaib, and A. E. Bara'a Alhammad, "A Survey on Security Patterns and their Classification Schemes," 2016. (Online). Available: https://www.researchgate.net/publication/330473622

[34] M.-A. Laverdì, A. Mourad, A. Hanna, and M. Debbabi, "Security Design Patterns: Survey and Evaluation," 2003.

[35] O. Drozd and S. Kirrane, "Towards an Interactive Privacy Pattern Catalog," 2016. (Online). Available: https://www.researchgate.net/publication/305811615

[36] J. W. J. W. Yoder and J. Barcalow, "Architectural patterns for enabling application security," Proceedings of PLoP 1997, vol. 51, p. 31, 1998, (Online). Available: http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Architectural+patterns+for+enabling+application+security#0

[37] S. Romanosky, A. Acquisti, J. Hong, L. F. Cranor, and B. Friedman, "Privacy patterns for online interactions," PLoP 2006 - PLoP Pattern Languages of Programs 2006 Conference Proceedings, no. October, 2006, doi: 10.1145/1415472.1415486.

[38] M. Weiss and H. Mouratidis, "Selecting security patterns that fulfill security requirements," in Proceedings of the 16th IEEE International Requirements Engineering Conference, RE'08, 2008, pp. 169–172. doi: 10.1109/RE.2008.32.

[39] M. Papoutsakis, K. Fysarakis, G. Spanoudakis, S. Ioannidis, and K. Koloutsou, "Towards a collection of security and privacy patterns," Applied Sciences (Switzerland), vol. 11, no. 4, pp. 1–42, 2021, doi: 10.3390/app11041396.

[40] Jeremiah Y. Dangler, "Categorization of Security Design Patterns," Categorization of Security Design Patterns, p. 144, 2013.

[41] "Privacy Patterns." Accessed: Aug. 05, 2024. [Online]. Available: https://privacypatterns.org/

[42] UC Berkeley School of Information, "Privacy Patterns." Accessed: Aug. 05, 2024. Online. Available: https://privacypatterns.org/