A Survey on Security Patterns and their Classification Schemes

Amjad Hudaib[1], Afaf Edinat[2], Bara'a Alhammad[3]

Computer Science Department, King Abdullah II School of Information Technology
University of Jordan, Amman, Jordan

Abstract: ***Security patterns are the best practices to solve recurring security problems in specific contexts. Finding the appropriate classification scheme for security patterns is still an obstacle encountered by architects where existing classifications considered only as a small number of patterns, and their purpose is often focused on implementation issues. Therefore, missing aspects in existing classifications are identified, and a new classification scheme we proposed based on Microsoft organizing table integrated with a certain criteria's Performance , Implementation Cost and Security Degree.***

Keywords: Security patterns , Architecure patterns, Zachmn framework

1. Introduction

Software systems usually have common structure for a certain kind of solutions observed through similarity over possible variations that can help to determine in what circumstances an approach can be used, and how flexible it is to be customized for specific system needs. This structure is known in software engineering as architectural patterns; a pattern is a description of set of predefined subsystems and their responsibilities, rules and guidelines for organizing their relationships. The patterns reflect the experience of many developers and generated when they solve certain types of problems in a similar way and produce a consensus on that particular way offering software reuse potential and common vocabulary of design solutions. Pattern's schema usually consists of three fundamental components; a context to describe on what situations it may apply, a problem to address including its urge forces and a solution's principles underlying the pattern [49,53]. The first person who used the pattern approach was Christopher Alexander. And in his book he indicated that each pattern describes a problem which occurs over and over again in our environment, and then states the core of the solution to that problem, in such a way you can use this solution million times over, without ever doing it the same way twice[72] .

Security and reliability issues are rarely considered at the initial stages of software development and are not part of the standard procedures in development of software and services and there is a very little work concerning the full integration of security and systems engineering. Although several approaches have been proposed for some integration of security, there is currently no comprehensive methodology to assist developers of security sensitive system; this shortage of support for security engineering is usually seen as a consequence of security requirements being generally difficult to analyses and model and developers lacking expertise in secure software development. The solution is security patterns which serve as a means of bridging the gap between developers and security experts [72,49].

Although several approaches have been proposed for some integration of security, there is currently no comprehensive methodology to assist developers of security sensitive system [72,91] .The lack of support for security engineering in those approaches for software systems development is usually seen as a consequence of :

1) Security requirements being generally difficult to analyses and model.
2) Developers lacking expertise in secure software development.

So the solution for this lacking , the Security pattern's which proposed as a means of bridging the gap between developers and security experts. Security patterns are intended to capture security expertise in the form of worked Solutions to recurring problems.[72,91]

Security patterns are reusable components that are guided by certain forces to be applicable in some contexts and solve specific problems; they are commonly described by general concepts related to their definition within a template of format. Assets are information that has value to an organization, stakeholders are the people who add value to these assets, security objectives are statements of intents to counter threats while threat is a potential of harm to an asset and an attack is an action to violate the security of an asset. Vulnerability is a weakness that may lead to breach the security of an asset and countermeasure is the action to be taken for asset protection, and risk is the probability of successful attack occurs [92].

In this paper ,we propose a new classification schema based on the Microsoft organizing table integrated with three security criteria's. The rest of the survey is organized as follows: The next section presents security patterns in details , section 3 describe classifications schemes in details , finally section 4 presents conclusion .

2. Security Patterns

Security design patterns approach the problem from a different perspective, by encapsulating expert knowledge in the form of proven solutions to common problems. and was later reused in the object-oriented world. Security patterns are such patterns, but applied for information security. These patterns will fit at different levels of abstraction and areas of concerns, resulting in many patterns that are not “design patterns” in the common sense of the expression[49].

Security architects only want to indicate which specific security mechanisms are needed not their implementation; therefore we need a set of patterns that define abstract security mechanisms and specify its fundamental characteristics as shown in *Figure 1* [33,65].

Other security service patterns doesn’t appear in the previous figure but they are tackled in [72,53] with detailed explanation of their intents, description and known uses such as reference monitor pattern, virtual address space access control, execution domain pattern, single access point pattern, check point pattern and session pattern.

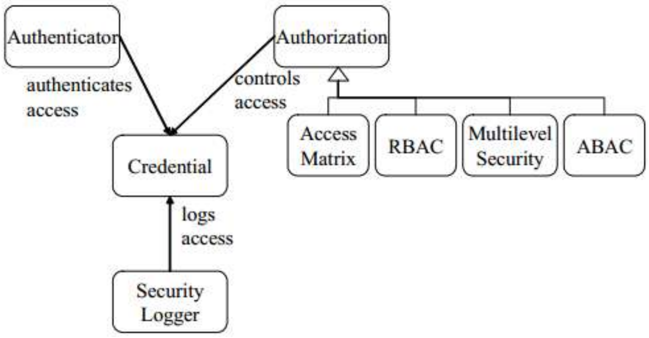


Figure 1: Basic Security Services [12]

2.1 Security properties

Software designers apply several design principles and heuristics to achieve different quality attributes. These principles and heuristics are called security properties, which provide a means to link appropriate patterns to a desired quality attribute .The most commonly cited security properties , As the following : [3, 82,89,91].

- ***Error management :*** A system should provide a robust error management mechanism to support error avoidance, error handing, fallback procedures and failure logging.
- ***Simplicity:*** A system should encapsulate initialization check processes, ensure security policy and low-level security, manage permissions and share global information. Systems should also be easy to use and keep the user interface consistent.
- ***Access Control:*** This property requires the system to support user identification , access verification, least privilege and privacy.
- ***Defense in depth:*** *This includes data verification, reduced exposure to attack, data protection, and communication and information protection.*

2.2 Security Basic Concepts

The most common security concepts , describe as the following :[95]

1) Asset: Is anything that has a value to the organization and the it’s mission.
2) Vulnerability: A weakness in any phase of the design, operation, implementation or any process in the system which expose the system to a threat.
3) Threat: Any possible danger that may result in harm of systems and organization.
4) Attack: An actual event done by a person; attacker to harm as asset of the software through exploiting a vulnerability.
5) Risk: a potential for loss, damage, or destruction of an asset as a result of a threat exploiting a vulnerability.
6) Software Security Requirement: is a non-functional requirement that elicit a control, constraint, safeguard to a void vulnerabilities from requirements design .
7) Confidentiality: means to disclose information to people or programs that are authorized to have access to that information.
8) Integrity: assures that a system performs its intended function, free from deliberate or inadvertent unauthorized manipulation of the system.
9) Availability: assures that systems work promptly, and service is not denied to authorized users.
10) Process: is an instance of a computer program that is being executed.
11) Secure software process: is a set of activities used to develop and deliver a secure software solution.

2.3 The Impact of Security Forces

Security usually has an impact on many other non-functionla requirements of a softwrae system (system properties) such as performance ,usability , avaliability. A specific solution can be easier to learn, slower, or more difficult to use. *Figure 2* shows how various forces can support or hinder one another.For example, performance is an important issue , the most suitable solution needs to be identified. The solution must balance such conflicting requirements or increase the performance[72].

The following security properties which are the most commonly used in the security domain are the ones considered in this study [3, 19, 44,55].

1) *Authentication:* It must be validated the identity of customers to frustrate any unauthorized access.
2) *Authorization:* This attribute defines the access privileges of entities to different resources and services of a system.
3) *Integrity*: To guarantee that data and communications will not be compromised by active attacks.
4) *Confidentiality*: The guarantee that information is not accessed by unauthorized parts.
5) *Attacker detection:* To be able to detect and register access or modification intents in the system coming from unauthorized users.
6) *Auditability:* To keep a log of user’s or other system’s interaction with a system and it helps detect potential attacks.
7) *Maintainability:* It facilitates the introduction or modification of the security policy during the software development life cycle.
8) *Availability:* It assures that authorized users can use the resources when they are required.
9) *Reliability*: It assures the system operations due to failures or configuration mistakes. Besides, it assures the system availability even when the system is being attacked.

10) *Performance:* It indicates the impact of the pattern on the functioning of a system.
11) *Implementation cost*: Costs accompanying the pattern use.
12) *Security degree*: It indicates the security level that the pattern has for the function it fulfills, that is, the more security properties the pattern covers, the more security degree will have.

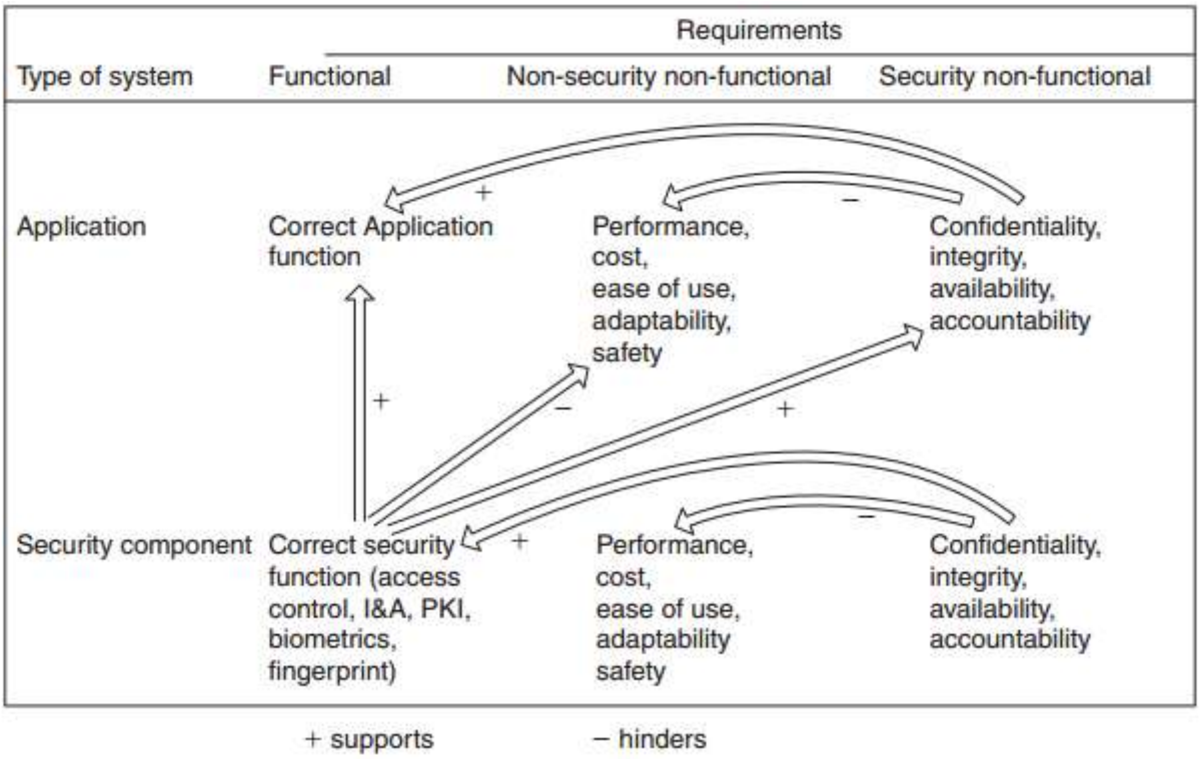

Requirements
Type of system
Functional
Non-security non-functional
Security non-functional
Application
Correct Application function
Performance, cost, ease of use, adaptability, safety
Confidentiality, integrity, availability, accountability
Security component
Correct security function (access control, I&A, PKI, biometrics, fingerprint)
Performance, cost, ease of use, adaptability safety
Confidentiality, integrity, availability, accountability
+
−
+
−
+
−
+
+ supports
− hinders

Figure 2: The Impact of Securit Forces [53]

2.5 Security Templete

A software pattern can be described through a set of properties (a template) such as name, problem, solution and so on. These templates allow authors to define new patterns, but respecting this structure . Template is based on existing design pattern templates with some additional sections, which we believe necessary for presenting the security patterns in an efficient way[3,72,53].

The templates on which security patterns are usually defined are multiple formats, but the general acceptable one in the literature consists of the common fundamental components for patterns beside a structure to detailed specification aspects of the pattern, dynamics to explain scenarios at run time, implementation instructions, example resolved, variants to clarify specializations, known uses and finally consequences and benefits of applying the pattern [38,42].

Pattern's templets usually consists of five fundamental components :

- **Name :** Is the common-usage short expression that encapsulates the pattern’s meaning.
- **Context :** To describe on what situations it may apply.
- **Problem :** Is a short description of the design problem that this pattern aims at solving.
- **Solution :** Is a textual description of the pattern that solves the problem .
- **Consequences:** To describe the trade-offs and results when we use the pattern.

3. Classification Schemes

Security patterns can be grouped into many categories based on multiple classification techniques proposed in the literature such as their relative software lifecycle phase, the problem they are attempting to solve and their abstraction level. Some security patterns are designed to tackle the information collected at the requirement phase while some other are meant to handle the intrinsic requests demanded by the detailed design schema on which the functional requirements are subject to change over time making it a challenging task under the spectrum of domain specific knowledge. Moreover, implementation phase impose a necessity for high security level upon user roles exchange and data interchange between subsystems other than the low-level one that deal with encryption and firewalls. Consequently; applying the appropriate security pattern to a certain phase in software lifecycle require developers to understand the relations among patterns and how they would communicate with each other [37,43].

To encounter the increasing number of patterns we need to develop classifications. A classification should be based on standard methodologies to organize the huge number of patterns. A classification organizes patterns into groups of patterns that share one or many properties such as the application domain or a particular purpose.

The kind of properties that should be used is not fixed. A pattern can have more than one specific property. Therefore, it may be included in more than one classification category[7].

Architects have introduced multiple classification schemes in an attempt to organize the security patterns landscape. Some have categorized them based on their purposes into creational structural and behavioral patterns [41] while others conducted survey effort to classify security patterns based on partitioning the system space . Here we will describe a list of these classification schemes [4,37,7,53]:

1) Based on applicability.

In this classification the patterns into two broad groups based on applicability; patterns for protected systems and patterns for available systems. The protected system patterns protect valuable resources against unauthorized use, expose, or modification. The available system patterns provide predictable and uninterrupted access to the services and resources. The advantage of this scheme is that it classifies the patterns according to the software architectural qualities that they address. However, this partitioning is too broad to be useful[53].

2) Based on product and process

Described their patterns under two broad classes :

- Structural patterns, are concerned with how classes and objects are composed to form larger structures. Structural class patterns use inheritance to compose interfaces or implementations[21].
- Procedural patterns, on the other hand, improve the process for development of security-critical software.

These patterns influence the organization or the management of a development process.

3) Based on System logical tiers.

The security patterns in layered systems are classified according to the system tiers. In this scheme, the patterns are classified as presentation or web tier patterns, business tier patterns and integration tier patterns. The web tier patterns intercept external requests and perform authentication and authorization. This classification schemes has the advantage that the partitioning is aligned with the system tiers. Hence the classification does not introduce new vocabulary for system architects and developers. However, the advantage of a classification scheme comes out of using the domain specific vocabulary [53,7].

4) Based on security concepts.

This classification based on the four key concepts of security - confidentiality, integrity, availability and accountability. A pattern classification scheme based on these domain level concepts, will facilitate pattern mining and pattern navigation.

5) Based on system viewpoints and interrogatives:

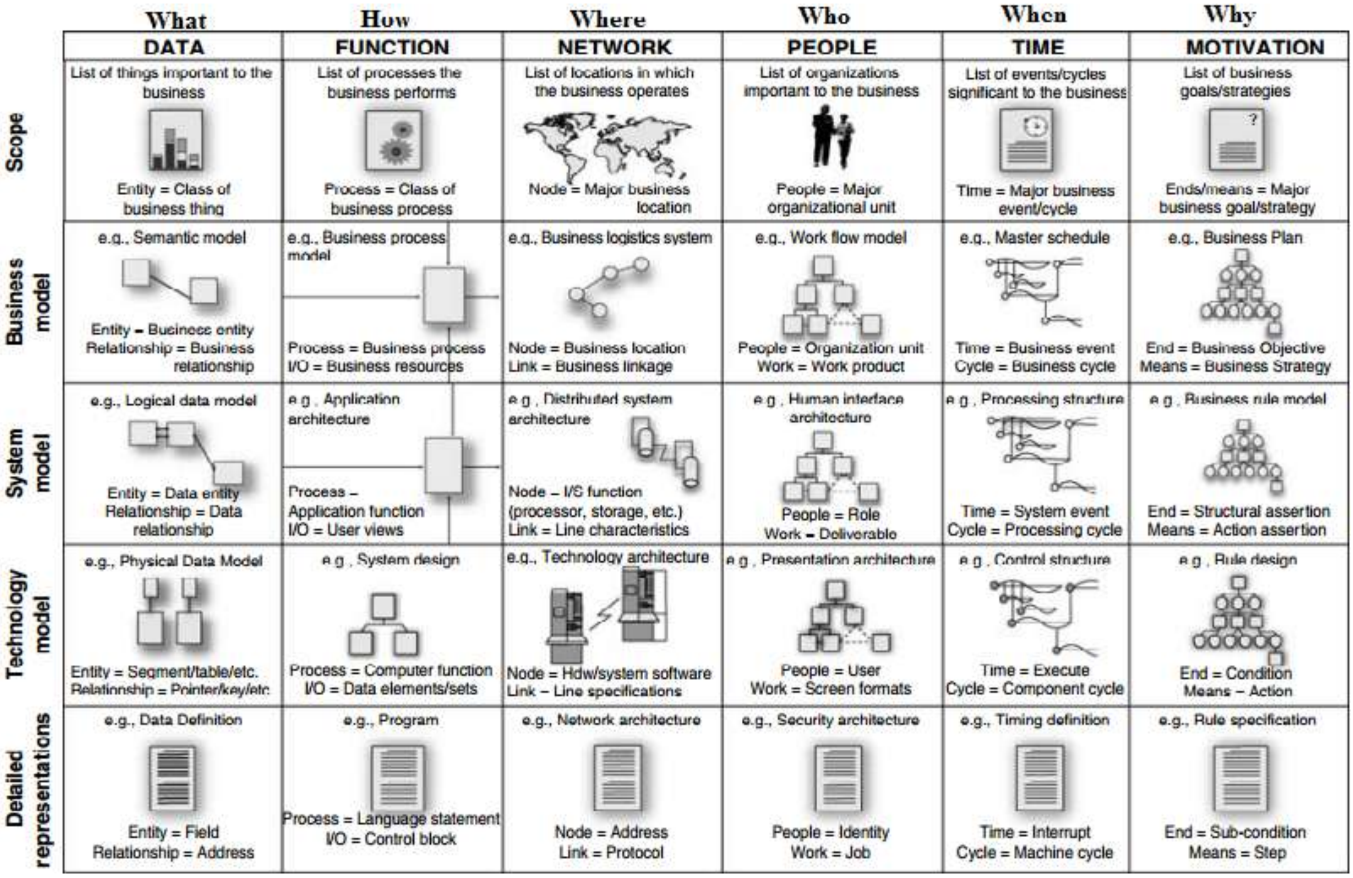
The security patterns book from Wiley publications introduced a classification scheme for security patterns that is based on the Zachman framework in *Figure 3*. Zachman framework was introduced in 1987 as a table with the rows describing the levels of information model and the columns describing the architectural views.

The levels of information model are based on three fundamental architectural representations one for each stakeholder:

- The customer or the owner has his own concept of the end product.
- The architect translates these perceptions into the designer's perspective.
- The builder then adds the constraints of the laws of nature and available technology to make a refinement of the architect's plan.

The model covers enterprise level in the first two rows(Business Model , Scope), and the next three rows covers the system level[53].

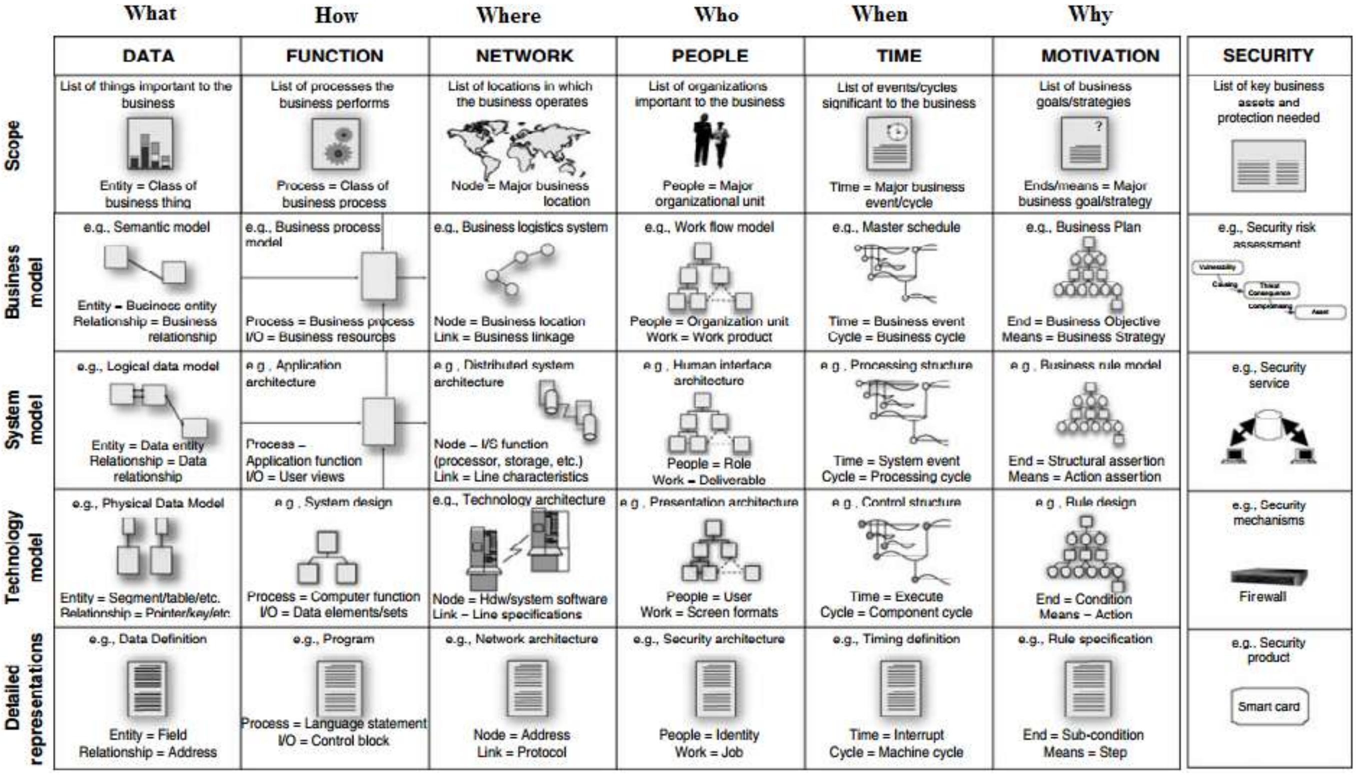
The Zachman views are represented in the six columns in the table: data (what?), function (how?), network (where?), people (who?), time (when?) and motivation (why?). These represent the different aspects of the object being described. Each of the views are orthogonal to each other, but they describe the same object and are associated tightly with each other.



	What	How	Where	Who	When	Why
	DATA	FUNCTION	NETWORK	PEOPLE	TIME	MOTIVATION
Scope	List of things important to the business Entity = Class of business thing	List of processes the business performs Process = Class of business process	List of locations in which the business operates Node = Major business location	List of organizations important to the business People = Major organizational unit	List of events/cycles significant to the business Time = Major business event/cycle	List of business goals/strategies Ends/means = Major business goal/strategy
Business model	e.g., Semantic model Entity – Business entity Relationship = Business relationship	e.g., Business process model Process = Business process I/O = Business resources	e.g., Business logistics system Node = Business location Link = Business linkage	e.g., Work flow model People = Organization unit Work = Work product	e.g., Master schedule Time = Business event Cycle = Business cycle	e.g., Business Plan End = Business Objective Means = Business Strategy
System model	e.g., Logical data model Entity = Data entity Relationship = Data relationship	e.g., Application architecture Process – Application function I/O = User views	e.g., Distributed system architecture Node – I/S function (processor, storage, etc.) Link = Line characteristics	e.g., Human interface architecture People = Role Work – Deliverable	e.g., Processing structure Time = System event Cycle = Processing cycle	e.g., Business rule model End = Structural assertion Means = Action assertion
Technology model	e.g., Physical Data Model Entity = Segment/table/etc. Relationship = Pointer/key/etc.	e.g., System design Process = Computer function I/O = Data elements/sets	e.g., Technology architecture Node = Hdw/system software Link – Line specifications	e.g., Presentation architecture People = User Work = Screen formats	e.g., Control structure Time = Execute Cycle = Component cycle	e.g., Rule design End = Condition Means – Action
Detailed representations	e.g., Data Definition Entity = Field Relationship = Address	e.g., Program Process = Language statement I/O = Control block	e.g., Network architecture Node = Address Link = Protocol	e.g., Security architecture People = Identity Work = Job	e.g., Timing definition Time = Interrupt Cycle = Machine cycle	e.g., Rule specification End = Sub-condition Means = Step

Figure 3: The Zachman framework (c) 1982-2006 John A. Zachman, www .zachman international.com [53]

To classify security patterns, the Zachman framework has been modified by adding a column representing the security view, as shown in *Figure 4*. The security view addresses all model levels,from the enterprise scope to technology model and detailed model representation model, from the enterprise scope to the detailed representations.



	What	How	Where	Who	When	Why	
	DATA	FUNCTION	NETWORK	PEOPLE	TIME	MOTIVATION	SECURITY
Scope	List of things important to the business Entity = Class of business thing	List of processes the business performs Process = Class of business process	List of locations in which the business operates Node = Major business location	List of organizations important to the business People = Major organizational unit	List of events/cycles significant to the business Time = Major business event/cycle	List of business goals/strategies Ends/means = Major business goal/strategy	List of key business assets and protection needed
Business model	e.g., Semantic model Entity – Business entity Relationship = Business relationship	e.g., Business process model Process = Business process I/O = Business resources	e.g., Business logistics system Node = Business location Link = Business linkage	e.g., Work flow model People = Organization unit Work = Work product	e.g., Master schedule Time = Business event Cycle = Business cycle	e.g., Business Plan End = Business Objective Means = Business Strategy	e.g., Security risk assessment Vulnerability Causing Threat Consequence Compromising Asset
System model	e.g., Logical data model Entity = Data entity Relationship = Data relationship	e.g., Application architecture Process – Application function I/O = User views	e.g., Distributed system architecture Node – I/S function (processor, storage, etc.) Link = Line characteristics	e.g., Human interface architecture People = Role Work – Deliverable	e.g., Processing structure Time = System event Cycle = Processing cycle	e.g., Business rule model End = Structural assertion Means = Action assertion	e.g., Security service
Technology model	e.g., Physical Data Model Entity = Segment/table/etc. Relationship = Pointer/key/etc.	e.g., System design Process = Computer function I/O = Data elements/sets	e.g., Technology architecture Node = Hdw/system software Link – Line specifications	e.g., Presentation architecture People = User Work = Screen formats	e.g., Control structure Time = Execute Cycle = Component cycle	e.g., Rule design End = Condition Means – Action	e.g., Security mechanisms Firewall
Detailed representations	e.g., Data Definition Entity = Field Relationship = Address	e.g., Program Process = Language statement I/O = Control block	e.g., Network architecture Node = Address Link = Protocol	e.g., Security architecture People = Identity Work = Job	e.g., Timing definition Time = Interrupt Cycle = Machine cycle	e.g., Rule specification End = Sub-condition Means = Step	e.g., Security product Smart card

Figure 4: Adding a security view to the Zachman framework (c) 1982-2006 John A. Zachman. www .zachman international.com [53]

3.1 Security Taxonomy

With the passing of years security experts have worked to establish security properties, approaches, and necessary services, for securing important enterprise assets by applying security engineering. To understand the relationships between these diverse security elements; they need to be organized into a usable taxonomy, as shown in *Figure 5*. In [53] the taxonomy is arranged to support development of enterprise security architecture , which is described in Table 1 .

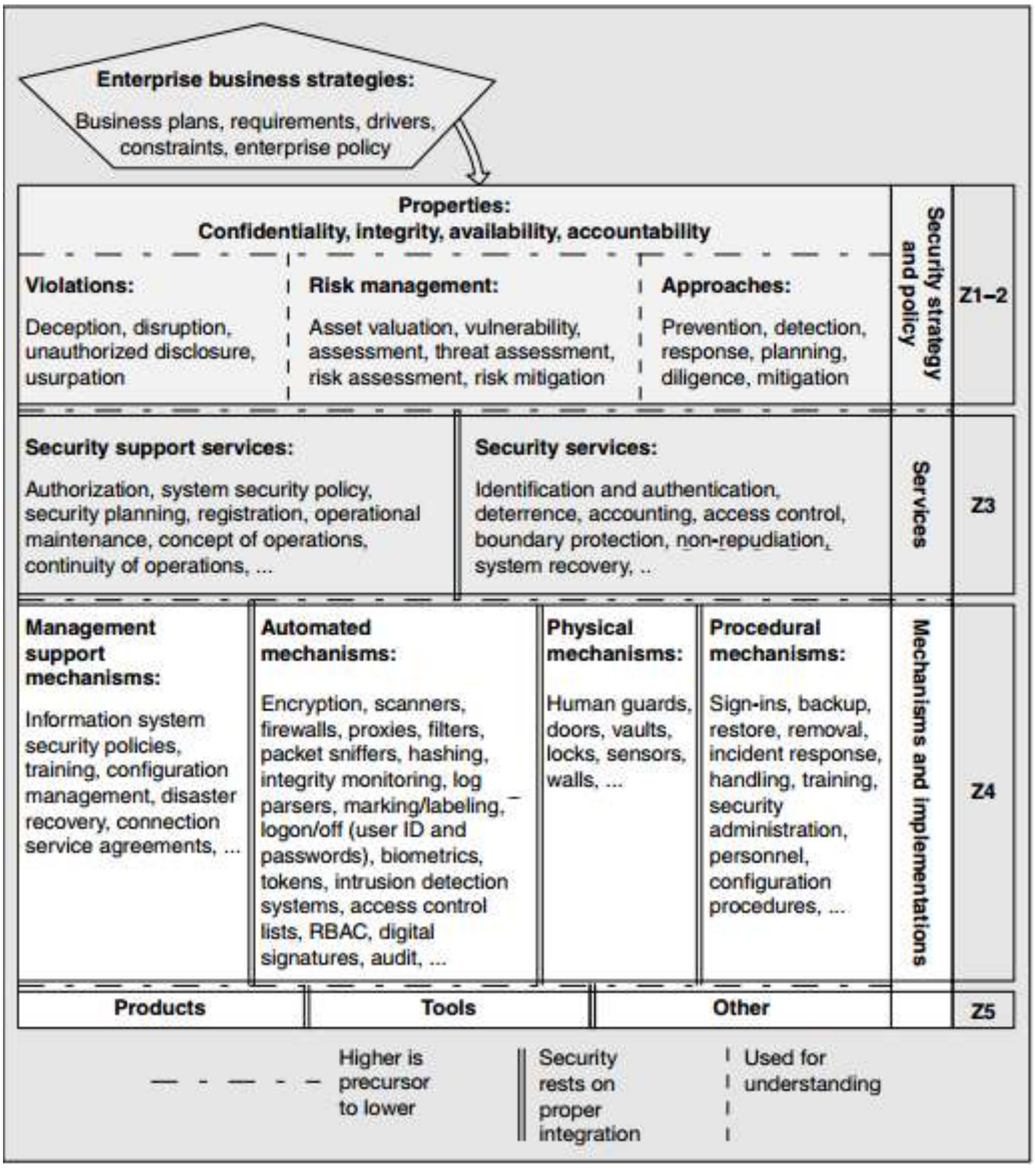

Enterprise business strategies:
Business plans, requirements, drivers, constraints, enterprise policy
Properties:
Confidentiality, integrity, availability, accountability
Violations:
Deception, disruption, unauthorized disclosure, usurpation
Risk management:
Asset valuation, vulnerability, assessment, threat assessment, risk assessment, risk mitigation
Approaches:
Prevention, detection, response, planning, diligence, mitigation
Security strategy and policy
Z1–2
Security support services:
Authorization, system security policy, security planning, registration, operational maintenance, concept of operations, continuity of operations, ...
Security services:
Identification and authentication, deterrence, accounting, access control, boundary protection, non-repudiation, system recovery, ..
Services
Z3
Management support mechanisms:
Information system security policies, training, configuration management, disaster recovery, connection service agreements, ...
Automated mechanisms:
Encryption, scanners, firewalls, proxies, filters, packet sniffers, hashing, integrity monitoring, log parsers, marking/labeling, logon/off (user ID and passwords), biometrics, tokens, intrusion detection systems, access control lists, RBAC, digital signatures, audit, ...
Physical mechanisms:
Human guards, doors, vaults, locks, sensors, walls, ...
Procedural mechanisms:
Sign-ins, backup, restore, removal, incident response, handling, training, security administration, personnel, configuration procedures, ...
Mechanisms and implementations
Z4
Products
Tools
Other
Z5
Higher is precursor to lower
Security rests on proper integration
Used for understanding

Figure 5: The Taxonomy of Security [53]

Table 1: The Description of Security Taxonomy

	Description	Elements	Example Pattern
Properties (z1,z2) (The scope and Business Model) Enterprise level)	Security is concerned with protection of assets, ensuring that actions are appropriate and holding actors responsible for their work.	• Confidentiality. • Integrity. • Accountability. • Availability	1) Security needs Identification patterns for enterprise assets. 2) Assets Valuation. 3) Threats Assessment. 4) Risk Determinations. 5) Enterprise Security Approaches. 6) Enterprise Security Services. 7) Enterprise Partner Communication.
Security strategy and policy(z3) (System Model)	Achieve the enterprise's objectives, which reflect the business strategy of the enterprise.	Violations : unauthorized , disclosure , deception are the major classes of vulnerability ,which can be attacked .	--------

		Risk Management: encompasses all form of the risk assessment and imagination for enterprise, such as Asset valuations, vulnerability, threats assessments, risk assessment, risk mitigation	1) Security needs Identification patterns for enterprise assets . 2) Assets Valuation. 3) Threats Assessment. 4) Risk Determinations. 5) Enterprise Security Approaches. 6) Enterprise Security Services. 7) Enterprise Partner Communication.
		Approaches : defines groups of related ways to address potential security violations , such as prevention ,detection ,response , planning .	1) Security needs Identification Patterns for Enterprise Assets. 2) Assets Valuation. 3) Threats Assessment. 4) Risk Determinations. 5) Enterprise Security Approaches. 6) Enterprise Security Services. 7) Enterprise Partner Communication.
Services(z4) (Technology Model)	Security Services are general safeguards that help achieve both enterprise and system security needs	Security supports services : Authorization , System security policy , security planning , ….. .	1) Authorization. 2) Role-Based Access Control. 3) Multilevel Security. 4) Reference Monitor. 5) Role Right Definition.
		Security Services : Identification and authentication , accounting , …. .	1) Identification and Authentication Patterns. 2) Access Controls Models Patterns. 3) System Access Control Architecture Patterns 4) Operating System Access Control Patterns. 5) Accounting Patterns. 6) Secure Internet Applications Patterns.
Mechanisms and implementations(5) (Detailed Representations)	Security services are dependent on the physical, procedural, or automated mechanisms available to implement those services. Mechanisms are dependent in turn on commercial products and other tools that implement those mechanisms.	Encryption, Scanners, Firewalls, Proxies, Filters, Intrusion, …	1) Identification and Authentication Patterns. 2) Access Controls Models Patterns. 3) System Access Control Architecture Patterns. 4) Operating System Access Control Patterns. 5) Accounting Patterns. 6) Secure Internet Applications Patterns.

3.2 Based on application-domain.

Which is easy to understand and depicts the five target application domains which were discovered:

Enterprise, Software, Cryptographic, User, and network, as shown in Table 2.

Table 2: Application-Domain Based Classification[7].

Application Domain	Description	Number of Publication Describing Security Patterns	Patterns Examples
Enterprise	Security patterns deal with aspects that are important for enterprises to ensure security in several enterprise segments.	[13],[20], [22],[23], [32],[71], [79]	The *Manage Risk* pattern introduced by Elsinga and Hofman
User	Security patterns are focused on user behavior or awareness of security issues.	[67],[69], [81]	The *password lock box* pattern, which encourages the user to protect master passwords with the highest level of security
Cryptographic	Security patterns depict secure communication between two applications over a network.	[5], [11], [51], [50]	The *Sender Authentication* pattern. It presents the problem and solution how to guarantee that a received message has been sent by a person one expected
Network	Security patterns picture network infrastructures and their ideal composition	[1], [10], [16], [14], [80], [34], [15], [6], [28], [31], [40], [68], [61], [75], [78], [79], [84]	The *Packet Filter Firewall* to shield an internal network from Internet attacks or just tunneling the communication traffic though a single controllable instance
Software	Security patterns describe mostly how to structure parts of software to ensure security requirements.	[18], [44], [27], [29], [24], [35], [25], [26], [36], [41], [39], [46],	JEE patterns, which can be applied only at Java enterprise applications .

	Sometimes they also describe a specific behavior or way to manage or control a data flow in a secure way	[52], [58], [56], [57], [62], [76], [85], [86[, [90], [91], [94], [44]	

3.3 Classification Proposed of Microsoft.

In 2004, Microsoft Patterns and Practice s group introduced a tabular classification scheme for patterns, primarily based on the Zachman framework.
The classification scheme encapsulates the enterprise architectural space, and illustrates the relationship among artifacts in the enterprise space.

The classification scheme is based on four key pieces of work are :

1) The Zachman frame work.
2) The Architectural standards description from IEEE 1471 .
3) The Enterprise Architecture Framework .
4) The scheme is influenced by the principles of test- driven development.

3.4 Enterprise Architectural Space Organizing Table.

3.4.1 Architectural viewpoints

The table, as shown in *Table 3* has five grouped rows based on, the view points are :

- Business Architecture : The business and management perspective of software development.
- Integration Architecture : Is concerned with the integration between internal and external systems in an enterprise
- Application Architecture : Covers the system and software elements of an executable application.
- Operational Architecture : Is concerned with the operation of the production system
- Development Architecture : Covers the systematic implementation concerns of application and integration architecture.

3.4.1.1 Interrogatives. can be achieved based on the interrogatives in the Zachman framework and test driven development are illustrated by the seven columns.

- **Purpose (Why).**The reason behind an architectural decision.
- **Data (What).** Input and output of a decision making process.
- **Function (How).** The mechanism of architectural decision making.
- **Timing (When) .**Timing related issues of a decision or the decision making process.
- **Network(Where).**Communication related issues of architecture.
- **People (Who).** Issues concerning the stake holders and users of a system.
- **Scorecard (Test).** Checking for compliance with the requirements.

3.4.1.2 Business Architecture. Is partitioned using the four primary role-players. They are,

- Chief Executive Officer (CEO)
- General Manager
- Process Owner
- Process Worker

Table 3: Classification of Security Patterns [37]

Perspective	Viewpoint	Interrogative	Pattern Count	Example Pattern
Business Architecture	CEO	Function	5	*Security Needs Identification.* Create Architecture an association between enterprise assets and security needs.
Integration Architecture	Enterprise Architect	Function	2	*Single Sign On*. Allow a user to access Integration multiple services in a distributed Architecture Architect network environment without having to re-authenticate on every request.
Application Architecture	Architect	Data	2	*Error Detection and Correction.* redundancy added to data for error detection and correction.
		Function	27	*Single Access Point*. Single Entry Point for each process.
	Design	Network	4	*Stateful Firewall.* Filter traffic based on state information.
		Data	4	*Encrypted Storage.* Server data is protected by encryption.
		Function	12	*Server Sandbox.* Servers run with least privilege to limit client activities.
	Developer	Function	1	*Safe Data Structure.* Memory buffers contain length information that is checked before allocation.
		Test	1	*White Hats hack Thyself.* Test the system's security by attacking it.
Operational Architecture	System Architect	Function	1	*Low Hanging Fruit.* Get quick fixes rather than trying to re-design the system every time a vulnerability is found.

1-Based on Additional Information.
A classification based on the classification proposed by Microsoft with additional information to introduce finer partitioning in the cells of the organizing table.

2-Hierarchical Classification.
A classification based on the classification proposed by Microsoft with additional information to introduce finer partitioning in the cells of the organizing table. The user can use this schema to identify the relevant patterns for his task, as shown in *Figure 6*.

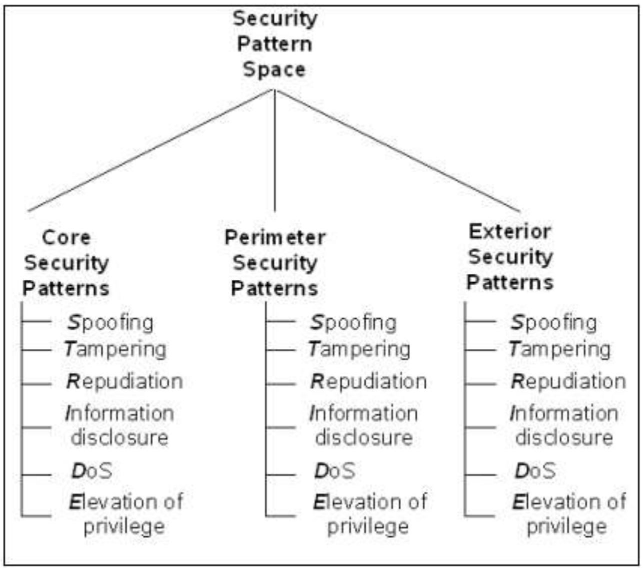


Figure 6: Hierarchal Classification [7]

3-Based on CIA Model

A classification based on the three key issues of security, Confidentiality, Integrity and Availability. The interpretation of these three issues varies based on the application context.

4-Based on Application Context

A Classification based on which part of the system they are trying to secure.

5-Based on The Security Wheel

A classification base on A security wheel represents the security features as in a spokes wheel. At the core of the hub of the wheel is the service or application that is under consideration. The spokes represent 12 core security services applicable to the service. These are authentication, authorization, confidentiality, integrity, policy, auditing, management, availability, compliance, logging, PKI and labeling. The edge of the wheel represent perimeter security.

6- Based on The McCumber Cube.

The classification space of the McCumber is identified by an information state, as shown in *Figure 7*:

- Based on security perspectives (CIA Model).
- Based on application context.
- Based on transmission (The perimeter security, Exterior security).

The X-axis represents the three primary categories of safeguards, i.e. Technology, policy and procedure, and human factor. The Y-axis of the model represents information states of transmission, storage and processing. The vertical axis comprises the three security perspectives of confidentiality, integrity and availability. The cub e is used for assessment and management of security risks in information technology systems.

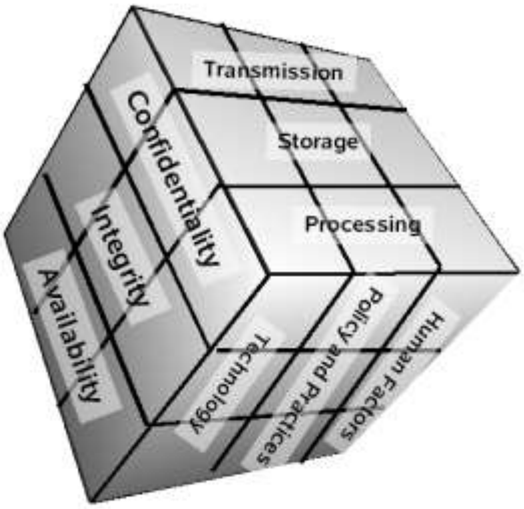
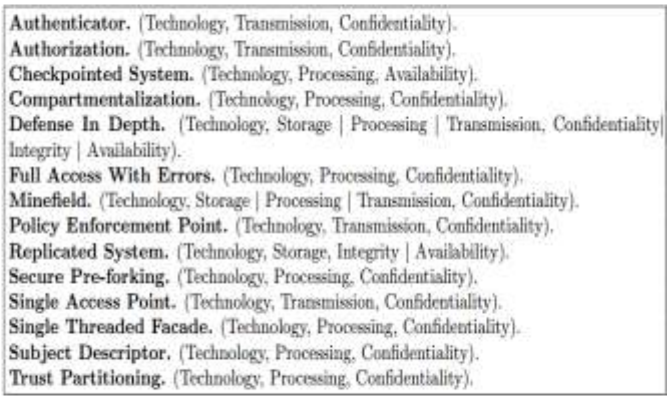


Figure 7: The McCumber Cube [7]

The classification of security patterns based on the McCumber cube ,as shown *in Figure 8*. The patterns and then provide the classification as a three tuple of (safe guard, information state , security persp e ctive).They us e the ‘ |’ symbol to describe multiple factors.



Authenticator. (Technology, Transmission, Confidentiality).
Authorization. (Technology, Transmission, Confidentiality).
Checkpointed System. (Technology, Processing, Availability).
Compartmentalization. (Technology, Processing, Confidentiality).
Defense In Depth. (Technology, Storage | Processing | Transmission, Confidentiality| Integrity | Availability).
Full Access With Errors. (Technology, Processing, Confidentiality).
Minefield. (Technology, Storage | Processing | Transmission, Confidentiality).
Policy Enforcement Point. (Technology, Transmission, Confidentiality).
Replicated System. (Technology, Storage, Integrity | Availability).
Secure Pre-forking. (Technology, Processing, Confidentiality).
Single Access Point. (Technology, Transmission, Confidentiality).
Single Threaded Facade. (Technology, Processing, Confidentiality).
Subject Descriptor. (Technology, Processing, Confidentiality).
Trust Partitioning. (Technology, Processing, Confidentiality).

Figure 8: Security Classification based on The McCumber Cube [7]

7- Based on Threat Modeling (Stride Model).

A classification based on threat modeling is used to identify and prioritize system security vulnerabilities.

8- Based on Meta-Model.

A classifications based on the patterns’ properties and relationship uniformly.

3.5 A Comparison Between Different Classifications Schemes of Security Patterns

In this section we will compare the various classification schemes to solve the problem of finding the appropriate pattern that we need to solve a particular problem in different system spaces, as shown in *Table 4* :

Table 4: A Comparison between Different Classifications Schemes of Security Patterns

Classification Schemes	Description	Advantages	Disadvantages
Based on Applicability	Classifies the patterns into two broad groups based on applicability: • Patterns for protected systems :valuable resources against unauthorized use	• Classifies the patterns according to software architectural qualities that they address.	• Too broad and useful

	- Patterns for available systems : provide predictable access to services and resources		
Based on Product and Process	Classifies the patterns under two broad classes : - Structural patterns: Can be implemented in the final products. - Procedural patterns: Improve the process for development of security critical software.	- Classifies according the development process in the enterprise.	- Too broad to be useful . - Influence on enterprise managements. - Break the system by finding the vulnerabilities.
Based on System logical Tiers	The security patterns in layered systems: Are classified according to the system tiers. In this scheme, The patterns are classified as presentation or web tier patterns, business tier patterns and integration tier patterns.	- The partitioning is aligned with the system tiers. - Using the Domain specific vocabulary	- The security concepts are not used for classifying.
Based on Security Concepts	Classifies patterns: Based on the four key concepts of security– confidentiality, integrity, availability and accountability.	- Will facilitate pattern mining and pattern navigation	- Too broad and useful - Does not cover all the security concepts
Based on System Viewpoints and interrogatives	Classifies security patterns, according the modified Zachman framework	- The security view addresses all model levels, from the enterprise scope to the detailed representations (5 view points).	- The same c classification as partitioning base d on system tiers.
Microsoft Classification	The classification scheme is based on four key pieces of work are : 1) The Zchman frame work. 2) The Architectural standards description from IEEE 1471 . 3) The Enterprise Architecture Framework. 4) The scheme is influenced by the principles of test-driven development.	- Pattern navigation becomes easier. - Identifying missing patterns	- Listing the same pattern in different contexts with varying granularity will create a huge number of patterns - The difficulty of managing the patterns.

Classification Schemes	Description	Advantages	Disadvantages
Based on Additional Information	A classification based on the classification proposed by Microsoft with additional information to introduce finer partitioning in the cells of the organizing table (3)	- Finer granularity, so that patterns can be classified with more specificity. - Uses domain specific vocabulary. - Make patterns navigation easier .	- Does not depend on the security terminology
Hierarchical	A classification based on the classification proposed by Microsoft with additional information to introduce finer partitioning in the cells of the organizing table (3) .The user can use this schema to identify the relevant patterns for his task .	- Simple classification notation (Tree). - Uses domain specific vocabulary.	- Does not depend on the security terminology
CIA Model	A classification based on the three key issues of security, Confidentiality, Integrity and Availability. The interpretation of these three issues varies based on the application context.	- Using the standard terminology from security literature to create partition.	- The problem is that the partitions are not disjoint from each other and most of the patterns would fall in a gray area. - Subset of the interrogatives in the Microsoft organize table(3).
Application Context	A Classification based on which part of the system they are trying to secure.	- The disjoining: there is a clear separation between the patterns	- The general patterns ,e.g. *(Defense in Depth*)cannot be classified using this scheme, because it impacts the core, The perimeter and the exterior security. - There is a lot of patterns would be classified as the core patterns without clear separation.
The Security Wheel	A classification base on A security wheel represents the security features as in a spokes wheel. At the core of the hub of the wheel is the service or application that is under consideration. The spokes represent 12 core security services applicable to the service. The edge of the wheel represent perimeter security.	- Finer granularity rather than application context based classification	- The overlap of classification patterns ,e.g.(*Policy Enforcement Point*) can be classified under authorization , authentication and policy .

Classification Schemes	Description	Advantages	Disadvantages
The McCumber Cube	The classification space of the McCumber is identified by an information state: • Based on security perspectives (CIA Model). • Based on application context. • Based on transmission (The perimeter security, Exterior security).	• This classification scheme is that it integrates three separate viewpoints.	• The dimensions of the cube does not provide a good partition. • The perspectives covered by the McCumber cube's information state is only a subset of the interrogatives in the Microsoft enterprise organizing table (what, how and where).
Threat Modeling	A classification based on threat modeling is used to identify and prioritize system security vulnerabilities.	• This classification is useful because it is uses security concepts.	• Patterns cannot be classified into a single group
Application Domain	This classification based on the five target application domains which were discovered: Enterprise, Software, Cryptographic, User, and network .	• This classification scheme fulfills the requirements of classifications in the terms of expandability, intuitive use, and is applicable • for security laymen	• Too broad and useful.
Meta –Model	A classification based the patterns' properties and relationship uniformly	• This classification is useful because it is based on the relationship between patterns .	• Too broad and useful.

3.6 The Proposed Classification Schema

In all previous classifications schemes, there is a lack of certain criteria's that needs to be considered as important issue for security patterns classifications. We will integrate the Microsoft organizing table with the most important of criteria's (Performance , Implementation Cost , Security degree) which have huge impact on the security pattern classifications.

- Performance: It indicates the impact of the pattern on the performance of the system.
- Implementation Cost: Costs accompanying the implementation of the pattern.
- Security Degree: It indicates the security level that the pattern has for the function it fulfills, that is, the more security properties the pattern covers, the more security degree will have.

When the same patterns listing in different contexts with varying granularity will create a huge number of patterns that exists in more than one cell in the table, Therefore we need additional criteria's to decrease the number of patterns that listed in the cells, to make it easy for choosing the appropriate pattern to solve a particular problem

4. Conclusion

It is very difficult to find the appropriate pattern to solve a particular problem because of the absence of a scientific classification scheme for security patterns. In this paper we identified several classification schemes and proposed a new schema based on Microsoft organizing table integrated with three important criteria's for security. Our proposed classification cover different aspects of security patterns based *Performance , Implementation Cost ,Security Degree*.

References

[1] B.BlakleyandC.Heath,*SecurityDesignPatterns*.TheOpen Group,Apr.2004,27.04.2011.[Online].Available:www.opengroup.org/onlinepubs/9299969899/toc.pdf

[2] B.Fernandez,S.Mujica,andF.Valenzuela,“Twosecuritypatterns,”in*Proc.ofAsianConferenceonPatternLanguagesofPrograms*,2010.

[3] Babar, Muahmmad Ali, Xiaowen Wang, and Ian Gorton. "Supporting security sensitive architecture design." *Quality of Software Architectures and Software Quality*. Springer, Berlin, Heidelberg, 2005. 140-154.

[4] Booch, Grady. "Architectural organizational patterns." *IEEE software* 25.3 (2008).

[5] Braga, A., C. Rubira, and Ricardo Dahab. "Tropyc: A pattern language for cryptographic software." (1999).

[6] Brown, F. L., DiVietri, J., de Villegas, G. D., & Fernandez, E. B. . "The authenticator pattern." *Proceedings of PLoP*. Vol. 99. 1999.

[7] Bunke, Michaela, Rainer Koschke, and Karsten Sohr. "Application-domain classification for security patterns." Proceedings of the International Conferences on Pervasive Patterns and Applications, IARIA Conferences. XPS (Xpert Publishing Services). 2011.

[8] Bunke, Michaela. "On the description of software security patterns." *Proceedings of the 19th European Conference on Pattern Languages of Programs*. ACM, 2014.

[9] Cabrera, Ralph, Brad Appleton, and Stephen P. Berczuk. "Software reconstruction: Patterns for reproducing software builds." *Pattern Languages of Programming'99* (1999).

[10] Cuevas, A., El Khoury, P., Gomez, L., Laube, A., & Sorniotti, A. . "A security pattern for untraceable secret handshakes." Emerging Security Information, Systems and Technologies, 2009. SECURWARE'09. Third International Conference on. IEEE, 2009.

[11] Cuevas, A., El Khoury, P., Gomez, L., & Laube, A. "Security patterns for capturing encryption-based access control to sensor data." *Emerging Security Information, Systems and Technologies, 2008. SECURWARE'08. Second International Conference on*. IEEE, 2008.

[12] D.M.Kienzle,M.C.Elder,D.Tyree,andJ.EdwardsHewitt,“ Securitypatternsrepository,version1.0,”2003,(28.08.2011).[Online].Available:http://www.scrypt.net/～celer/securitypatterns/repository.pdf

[13] Dallons, G., Massonet, P., Molderez, J. F., Ponsard, C., & Arenas, A.. "An analysis of the chinese wall pattern for guaranteeing confidentiality in grid-based virtual organisations." *Security and Privacy in Communications Networks and the Workshops, 2007. SecureComm 2007. Third International Conference on*. IEEE, 2007.

[14] Delessy, Nelly, Eduardo B. Fernandez, and T. Sorgente. "Patterns for the extensible access control markup language." *Proceedings of the 12th Pattern Languages of Programs Conference (PLoP2005)*. 2005.

[15] Delessy, N., Fernandez, E. B., Larrondo-Petrie, M. M., & Wu, J. . "Patterns for access control in distributed systems." *Proceedings of the 14th Conference on Pattern Languages of Programs*. ACM, 2007.

[16] Delessy-Gassant, N., Fernandez, E. B., Rajput, S., & Larrondo-Petrie, M. M.. "Patterns for application firewalls." *Proceedings of the Pattern Languages of Programs (PLoP) Conference*. 2004.

[17] Deng, Y., Wang, J., Tsai, J. J., & Beznosov, K. "An approach for modeling and analysis of security system architectures." *IEEE Transactions on knowledge and data engineering* 15.5 (2003): 1099-1119.

[18] Dougherty, C., Sayre, K., Seacord, R. C., Svoboda, D., & Togashi, K. . “ Secure design patterns”. No. CMU/SEI-2009-TR-010. CARNEGIE-MELLON UNIV PITTSBURGH PA SOFTWARE ENGINEERING INST, 2009.

[19] Dwivedi, Ashish Kumar, and Santanu Kumar Rath. "Incorporating security features in service-oriented architecture using security patterns." ACM SIGSOFT Software Engineering *Notes* 40.1 (2015): 1-6.

[20] Dyson, Paul, and Andy Longshaw. "Patterns for Managing Internet-Technology Systems." *EuroPLoP*. 2003.

[21] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, “Design Patterns: Elements of Reusable Object-Oriented Software “Addison-Wesley, 1994.

[22] Elsinga, Ben, and Aaldert Hofman. "Security Paradigm Pattern Language." *EuroPLoP*. 2003.

[23] Ernst, Alexander M. "Enterprise architecture management patterns." *Proceedings of the 15th Conference on Pattern Languages of Programs*. ACM, 2008.
EuroPLoP. 2003.

[24] Fernandez, Eduardo B. "Patterns for operating systems access control." *Procs. of PLoP 2002* (2002).

[25] Fernandez, Eduardo B., and David laRed Martinez. "Patterns for the secure and reliable execution of processes." *Proceedings of the 15th Conference on Pattern Languages of Programs*. ACM, 2008.

[26] Fernandez, Eduardo B., and Günther Pernul. "Patterns for session-based access control." *Proceedings of the 2006 conference on Pattern languages of programs*. ACM, 2006.

[27] Fernandez, Eduardo B., and John Sinibaldi. "More Patterns for Operating System Access Control." *EuroPLoP*. 2003.

[28] Fernandez, Eduardo B., and Reghu Warrier. "Remote authenticator/authorizer." *Procs. of PLoP* (2003).

[29] Fernandez, Eduardo B., and Rouyi Pan. "A pattern language for security models." *In Proc. of PLoP*. Vol. 1. 2001.

[30] Fernandez, Eduardo B., and Sergio Mujica. "Building secure systems: From threats to security patterns." *Chilean Computer Science Society (SCCC), 2010 XXIX International Conference of the*. IEEE, 2010.

[31] Fernandez, E. B., Larrondo-Petrie, M. M., Seliya, N., Delessy-Gassant, N., & Schumacher, M "A pattern language for firewalls." ernandez, E. B., Larrondo-Petrie, M. M., Seliya, N., Delessy-Gassant, N., & Schumacher (2003).

[32] Fernandez, E. B., Ballesteros, J., Desouza-Doucet, A. C., & Larrondo-Petrie, M. M. "Security patterns for physical access control systems." IFIP Annual Conference *on Data and Applications Security and Privacy*. Springer, Berlin, Heidelberg, 2007.

[33] Fernandez, Eduardo B., Hironori Washizaki, and Nobukazu Yoshioka. "Abstract security patterns." *Proceedings of the 15th Conference on Pattern Languages of Programs*. ACM, 2008.

[34] Fernandez, Eduardo B., Juan C. Pelaez, and Maria M. Larrondo-Petrie. "Security patterns for voice over ip networks." *Computing in the Global Information Technology, 2007. ICCGI 2007. International Multi-Conference on*. IEEE, 2007.

[35] Fernandez, Eduardo B., Tami Sorgente, and Maria M. Larrondo-Petrie. "Even more patterns for secure operating systems." *Proceedings of the 2006 conference on Pattern languages of programs*. ACM, 2006.

[36] Gondi, Vivek. "Multiple secure observers using j2ee." *Proceedings of the Conference on Pattern Languages of Programs*. 2010.

[37] Hafiz, Munawar, and Ralph E. Johnson. "Security patterns and their classification schemes." *University of Illinois at Urbana-Champaign Department of Computer Science, Tech. Rep* (2006).

[38] Hafiz, Munawar, Paul Adamczyk, and Ralph Johnson. "Patterns Transform Architectures." *Software Architecture (WICSA), 2011 9th Working IEEE/IFIP Conference on*. IEEE, 2011.

[39] Hafiz, Munawar, Ralph Johnson, and Raja Afandi. "The security architecture of qmail." *Proceedings of the 11th Conference on Patterns Language of Programming (PLoP’04)*. 2004.

[40] Hafiz, Munawar. "A collection of privacy design patterns." *Proceedings of the 2006 conference on Pattern languages of programs*. ACM, 2006.

[41] Hafiz, Munawar. "Secure pre-forking-a pattern for performance and security." *Proceedings of the Conference on Pattern Languages of Programs*. 2005.

[42] Harrison, N., and P. Avgeriou. "Leveraging Architecture Patterns to Satisfy Quality Attributes, In proc." *First European Conference on Software Architecture*.

[43] Heyman, T., Yskout, K., Scandariato, R., & Joosen, W. "An analysis of the security patterns landscape." Software Engineering for Secure Systems, 2007. SESS'07: ICSE Workshops 2007. Third International Workshop on. IEEE, 2007.

[44] Hofman, Aaldert, and Ben Elsinga. "Control the Actor-Based Access Rights." *EuroPLoP*. 2002.

[45] I, Gorton. "Architecture Patterns." *Essential Software Architecture/Software Engineering*. Amsterdam: Springer,2014. Page 35

[46] Kodituwakku, Saluka R., Peter Bertok, and Liping Zhao. "APLRAC: A Pattern Language for Designing and Implementing Role-Based Access Control." *EuroPLoP*. Vol. 1. 2001.

[47] Konrad, S., Cheng, B. H., Campbell, L. A., & Wassermann, R. . "Using security patterns to model and analyze security requirements." *Requirements Engineering for High Assurance Systems (RHAS'03)* 11 (2003).

[48] Kumar, A., and E. Fernandez. "A security pattern for a virtual private network." *Proceedings of the Latin American Conference on Pattern Languages of Programming*. 2010.

[49] Laverdiere, M. A., Mourad, A., Hanna, A., & Debbabi, M. (2006, May). Security design patterns: Survey and evaluation. In *Electrical and Computer Engineering, 2006. CCECE'06. Canadian Conference on* (pp. 1605-1608). IEEE.

[50] Lehtonen, Sami, and Juha Pärssinen. "A pattern language for key management." *Procs. of PLoP 2001* (2001).

[51] Lehtonen, Sami, and Juha Pärssinen. "A pattern language for cryptographic key management." *Procs. of PLoP 2001* (2002).

[52] Mahmoud, Q. "Security policy: A design pattern for mobile Java code." Proceedings of the 7th Conference on Pattern Languages of Programming (PLoP'00). 2000.

[53] Schumacher, M., Fernandez-Buglioni, E., Hybertson, D., Buschmann, F., & Sommerlad, P ."Security Patterns: Integrating Security and System Engineering." (2006).

[54] Me, Gianantonio, Giuseppe Procaccianti, and Patricia Lago. "Challenges on the Relationship between Architectural Patterns and Quality Attributes." *Software Architecture (ICSA), 2017 IEEE International Conference on*. IEEE, 2017.

[55] Monroe, R. T., Kompanek, A., Melton, R., & Garlan, D. "Architectural styles, design patterns, and objects." *IEEE software* 14.1 (1997): 43-52.

[56] Morrison, Patrick, and Eduardo B. Fernandez. "Securing the Broker Pattern." *EuroPLoP*. 2006.

[57] Morrison, Patrick, and Eduardo B. Fernandez. "The credentials pattern." *Proceedings of the 2006 conference on Pattern languages of programs*. ACM, 2006.

[58] Mouratidis, Haralambos, Paolo Giorgini, and Markus Schumacher. "Security patterns for agent systems." (2003).

[59] Muñoz-Arteaga, Jaime, Ricardo Mendoza González, and Jean Vanderdonckt. "A classification of security feedback design patterns for interactive web applications." *Internet Monitoring and Protection, 2008. ICIMP'08. The Third International Conference on*. IEEE, 2008.

[60] Nhlabatsi, A., Bandara, A., Hayashi, S., Haley, C. B., Jurjens, J., Kaiya, H., ... & Tun, T. T. "Security patterns: Comparing modeling approaches." *Software engineering for secure systems: Industrial and research perspectives* (2010): 75-111.

[61] Okubo, Takao, and Hidehiko Tanaka. "Web security patterns for analysis and design." *Proceedings of the 15th Conference on Pattern Languages of Programs*. ACM, 2008.

[62] Ortega-Arjona, Jorge L., and Eduardo B. Fernandez. "The secure blackboard pattern." *Proceedings of the 15th Conference on Pattern Languages of Programs*. ACM, 2008.

[63] Pasquale, L., Ghezzi, C., Pasi, E., Tsigkanos, C., Boubekeur, M., Florentino-Liano, B., ... & Nuseibeh, B. "Topology-Aware Access Control of Smart Spaces." *Computer* 50.7 (2017): 54-63.

[64] Peters, Joeri, Jan Martijn EM van der Werf, and Jurriaan Hage. "Architectural pattern definition for semantically rich modular architectures." *Software Architecture (WICSA), 2016 13th Working IEEE/IFIP Conference on*. IEEE, 2016.

[65] Ponde, Poonam, Shailaja Shirwaikar, and Sharad Gore. "Hierarchical Cluster Analysis On Security Design Patterns." *Proceedings of the International Conference on Advances in Information Communication Technology & Computing*. ACM, 2016.

[66] Priebe, T., Fernandez, E. B., Mehlau, J. I., & Pernul, G.. "A pattern system for access control." *Research Directions in Data and Applications Security XVIII*. Springer, Boston, MA, 2004. 235-249.

[67] Riehle, D., Cunningham, W., Bergin, J., Kerth, N., & Metsker, S. "Password Patterns." *EuroPLoP*. 2002.

[68] Roedig, U., and M. Schumacher. "Security engineering with patterns." *Pattern Languages of Programs*. Vol. 2001. 2001.

[69] Romanosky, S., Acquisti, A., Hong, J., Cranor, L. F., & Friedman, B. "Privacy patterns for online interactions." *Proceedings of the 2006 conference on Pattern languages of programs*. ACM, 2006.

[70] Romanosky, Sasha. "Enterprise security patterns." (2002).

[71] Romanosky, Sasha. "Security Design Patterns Part 1 v1. 4." Google Scholar (2001).

[72] Rosado, D. G., Gutierrez, C., Fernandez-Medina, E., & Piattini,. "A study of security architectural patterns." Availability, Reliability and Security, 2006. ARES 2006. The First International Conference on. IEEE, 2006.

[73] Ryoo, J., Kazman, R., & Anand, P. (2015). Architectural analysis for security. *IEEE Security & Privacy*, *13*(6), 52-59.

[74] Ryoo, Jungwoo, Phil Laplante, and Rick Kazman. "In search of architectural patterns for software security." *Computer* 42.6 (2009).

[75] Sadicoff, Mauricio, Maria M. Larrondo-Petrie, and Eduardo B. Fernandez. "Privacy-aware network client pattern." *Proceedings of the Pattern Languages of Programs Conference*. 2005.

[76] Saridakis, Titos. "Design Patterns for Fault Containment." *EuroPLoP*. 2003.

[77] Sarmah, Achyanta, Shyamanta M. Hazarika, and Smriti K. Sinha. "Security pattern lattice: A formal model to organize security patterns." *Database and Expert Systems Application, 2008. DEXA'08. 19th International Workshop on*. IEEE, 2008.

[78] Schleinzer, Benjamin, and Nobukazu Yoshioka. "A security pattern for data integrity in p2p systems." *Proceedings of the 17th Conference on Pattern Languages of Programs*. ACM, 2010.

[79] Schumacher, M., Fernandez-Buglioni, E., Hybertson, D., Buschmann, F., & Sommerlad, P. Security Patterns: Integrating security and systems engineering. John Wiley & Sons, 2013.

[80] Schumacher, Markus. "Firewall Patterns." *EuroPLoP*. 2003.

[81] Schumacher, Markus. "Security Patterns and Security Standards - with selected security patterns for anonymity and privacy" *EuroPLoP*. 2002.

[82] Schwanke, Robert W. "Layers, decisions, patterns, styles, and architectures." *Software Architecture, 2001. Proceedings. Working IEEE/IFIP Conference on*. IEEE, 2001.

[83] Sion, L., Yskout, K., van Den Berghe, A., Scandariato, R., & Joosen, W ."MASC: modelling architectural security concerns." *Modeling in Software Engineering (MiSE), 2015 IEEE/ACM 7th International Workshop on*. IEEE, 2015.

[84] Sommerlad, Peter. "Reverse Proxy Patterns." *EuroPLoP*. 2003.

[85] Sorensen, Kristof Elof. "Session Patterns." *EuroPLoP*. 2002.

[86] Steel, Chritopher, and Ramesh Nagappan. Core Security Patterns: Best Practices and Strategies for J2EE", Web Services, and Identity Management. Pearson Education India, 2006.

[87] Thomsen, Dan. "Practical policy patterns." Proceedings of the first ACM conference on Data and application security and privacy. ACM, 2011.

[88] Uzunov, Anton Victor. Engineering security methodologies for distributed systems. Diss. 2014.

[89] Vlissides, J., Helm, R., Johnson, R., & Gamma, E ."Design patterns: Elements of reusable object-oriented software." *Reading: Addison-Wesley* 49.120 (1995): 11.

[90] Weiss, Michael. "Credential delegation: Towards grid security patterns." *Procs. of the Nordic Pattern Languages of Programs Conference (VikingPLoP)*. 2006.

[91] Yoder, Joseph, and Jeffrey Barcalow. "Architectural patterns for enabling application security." *Urbana* 51 (1998): 61801.

[92] Yoshioka, N., Washizaki, H., & Maruyama, K. (2008). A survey on security patterns. *Progress in informatics*, *5*(5), 35-47.

[93] Yskout, Koen, Riccardo Scandariato, and Wouter Joosen. "Does organizing security patterns focus architectural choices?." *Proceedings of the 34th International Conference on Software Engineering*. IEEE Press, 2012.

[94] Zhou, Yu, Qian Zhao, and Mark Perry. "Policy enforcement pattern." Proceedings of the Conference on Pattern Languages of Programs. 2002.

[95] A.Hudaib , M.Alshraideh , O.Surakh and M. Khanfash ,"A Survey on Desig Methods for Secure Software Development " , INTERNATIONAL JOURNAL OF COMPUTERS & TECHNOLOGY 16.7 (2017): 7047-7064.