Artificial Intelligence for Confidential Information Sharing Based on Knowledge-Based System

Bouchra Boulahiat, Salima Trichni, Mohammed Bougrine, Fouzia Omary
Faculty of Sciences of Rabat-Computer Science Department, Mohammed V University in Rabat, Rabat – Morocco

***Abstract*—Ensuring the security of sensitive data and protecting user privacy remains one of the most significant challenges in our contemporary landscape. Organizations Companies cannot adopt a new technology without reassurance regarding data confidentiality. To address these challenges, we present an innovative system that draws upon extensive knowledge and expertise in the field of cryptography, especially in encryption methods. This system tailors its strategies to align with specific scenarios, prioritizing data confidentiality. Our solution is based on one of the Artificial Intelligence techniques, which is Knowledge-Based Systems (KBS) and extends the intelligent encryption methods from our previous research. However, this new system has taken a novel approach by reconfiguring this within KBS architecture. We have introduced additional technical components, including knowledge bases, an inference engine, and the Nearest Neighbor (NN) search algorithm. As a result, this revised architecture not only enhances security and system performance but also showcases improved maintainability and scalability.**

***Keywords*—IT security; cryptography; confidentiality; Knowledge-Based system; artificial intelligence**

I. Introduction

IT security is an immensely influential domain that plays a pivotal role in shaping the trajectory of the broader IT landscape. The digital realm evolves exponentially, consistently introducing innovative concepts to simplify our lives and enhance our daily experiences. However, this technological evolution, when it intersects with fundamental human values and the need to uphold privacy, is stripped of its scientific significance and weight.

Security considerations have remained a persistent challenge from the inception of computer systems to the present day. The paramount question that incessantly prevails is that of data security. Initially, this question was primarily concerned with data confidentiality. Nevertheless, the emergence of the internet ushered in a multitude of additional requirements, including the assurance of data integrity, authenticity, availability, and non-repudiation. This evolution gave rise to various cryptographic primitives, encompassing both symmetric and asymmetric encryption algorithms, hash functions, digital signatures, and digital certificates.

However, the pace of creating new cryptographic tools tailored to evolving technological needs has significantly slowed down. Contemporary enterprises tend to gravitate towards well-established algorithms celebrated for their enduring performance and resilience over the years, as opposed to seeking novel cryptographic systems. Consequently, to adopt a new technology, these companies must establish a robust security policy to accommodate the requisites of the technology. This policy relies on a set of cryptographic primitives, along with additional methods for managing access and infrastructure.

Nonetheless, modern IT systems are increasingly characterized by their diversity, interactivity, and the need to interact with a continually expanding network of third parties. These third parties may not necessarily adhere to the same security standards as other established partners. This presents a dilemma of rigidity in security protocols: either rigidly enforces identical security policies across all interactions, a measure that may limit user participation, or continuously adapt the system to accommodate new customer requirements, an approach that can prove costly and impact existing customers. Existing methods might be too inflexible in their security protocols, making it challenging to adapt or align with varying security standards of third-party networks. This inflexibility could hinder effective collaboration with diverse partners.

To address this predicament, we propose a novel encryption model designed to flexibly align with the specific demands of each situation. By implementing this mediation, we can make informed decisions regarding the most suitable encryption algorithm.

The reasons that make the proposed encryption model suitable for addressing the challenges mentioned in the paragraph:

1) Flexibility and adaptability: The encryption model is designed to be flexible, allowing it to adapt to varying security standards and requirements specific to each situation.

2) Tailored security measures: It can accommodate diverse IT systems, interact with a broad network of third parties, and align security measures accordingly without compromising overall protection.

3) Customization for Different Situations: It enables customization based on individual demands, ensuring that security protocols can be adjusted as needed, even when dealing with partners who might not adhere to standard security practices.

4) Cost-effective solution: The model aims to balance the need for security enhancements with cost considerations, ensuring that implementing and adapting security measures remains economically viable.

5) Minimized Impact on Existing Customers: By offering

flexibility, it minimizes the impact on existing customers while accommodating new requirements, avoiding disruptions in service or user experience.

Overall, the proposed encryption model aims to provide a versatile and adaptive solution that addresses security challenges in modern IT systems without excessively limiting user engagement or imposing exorbitant costs.

In the following sections, we will delve into the intricacies of this approach. We will commence by outlining our motivation and the prior research undertaken in this domain. Subsequently, we will expound upon the proposed solution, detailing its underlying principles and the various modules that constitute it, starting from data classification and extending to its inference engine. Lastly, we will elucidate the application and experimental aspects of this approach, which have been tested against a diverse array of the most renowned encryption algorithms.

A. Background

Reviewing the literature alongside various research endeavors concerning established encryption algorithms [1] [2] [3], and drawing from our own hands-on experience with these algorithms [4] [5] [6] [11], it becomes apparent that the level of security within each encryption system remains far from constant. This security profile tends to fluctuate from one study to another. What one might deem the ideal algorithm, such as AES, in one scenario, may not hold the same status in another [7]. In essence, the performance profile of each algorithm undergoes variations contingent upon the specific context and the environment in which it operates.

To illustrate this, let us consider a comparative study mentioned in [8]. This study assessed the performance of symmetrical encryption algorithms, namely DES, AES, and Blowfish, in the context of processing images. The study involved several images of varying sizes, each accompanied by their respective histograms. By comparing the encryption and decryption times of these algorithms, the research presented a comparative diagram in “Fig. 1”, as demonstrated below:

Examining “Fig. 1”, the following conclusions can be drawn:

- DES excels when dealing with small images.
- Blowfish stands out for its efficiency in processing large images in terms of cipher time.
- Blowfish and AES exhibit nearly identical performance levels during decryption.

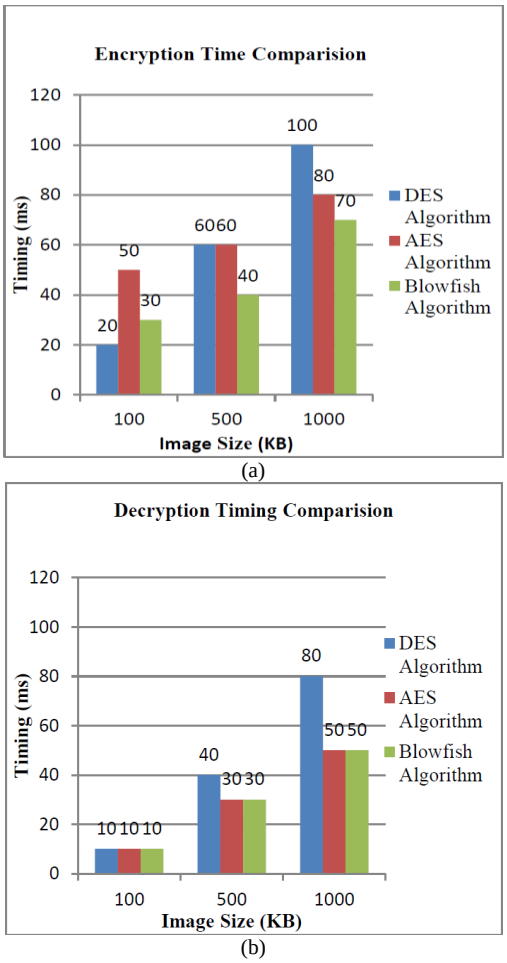
In a different study, which has also contributed to inspiring our approach and addresses various constraints, data types were considered [9]. This research explored the performance of encryption algorithms on mobile platforms, focusing on Triple DES, AES, and other methods based on elliptic curve arithmetic (ECC). The study sought to evaluate their performance across different Android platforms, including:

- Acer Iconia Tab A511
- Samsung Galaxy S4 i9505
- LG P500 Optimus One

Within each environment, numerous tests were conducted to assess algorithm performance based on the storage type, whether on an SD card or internal storage. The results from this study diverge from the previous one, highlighting the following insights:

- On the Samsung Galaxy S4 i9505, AES and DESede performed similarly, while ECC is not recommended.
- On the LG P500, AES remained the superior choice.
- On the Acer Iconia Tab A511, ECC and AES demonstrated comparable performance for smaller input file sizes, although ECC is not recommended for handling larger text files.

Hence, relying solely on a single encryption system for all communications could potentially undermine both security and system performance.



(a)

(b)

Fig. 1. Comparison between the timing of image encryption algorithms (a) and image decryption algorithms (b) using different image sizes.

In our prior works [5] [10], we devised a system that enables the categorization of various encryption scenarios within a decision-making database structured using a star modeling, comprising a central fact table and multiple dimensions. This system involves extracting the characteristics of each communication and employs data warehousing techniques to determine the most secure encryption algorithm for a case similar to or closely aligned with our own.

Building upon this foundational principle, this work introduces a substantial overhaul of the system, adopting a novel architecture based on a Knowledge-Based System (KBS). This innovative design offers distinct advantages over the basic idea, making the system more intelligent and autonomous, thus simplifying maintenance and enhancing result quality.

B. Related Works

To adapt the use of encryption systems to various environmental contexts and types of data exchanged, several research efforts have been undertaken to determine the most suitable encryption approaches. For instance, in [3], the authors directed their investigation toward the Smart Grid (SG), recognizing that devices within this network generate substantial data flows daily. To enhance security in this specific network, [3] proposed an approach grounded in multi-criteria analysis, designed to select the most appropriate encryption algorithm for optimizing energy production, consumption, and distribution. The PROMETHEE method was employed in this work, affirming the effectiveness of the AES-128 algorithm in alignment with decision-makers' preferences. Factors such as memory usage, encryption and decryption times, battery power consumption, and simulation time were taken into account.

Furthermore, the study in [10] represents another notable study that leverages a decision-making approach to determine the lightest and most secure encryption and authentication methods for Internet of Things (IoT) devices, particularly within the realm of IoHT (Internet of Healthcare Things). This research centered on data exchanges among IoHT devices operating within a healthcare environment. These devices generate sensitive data, possess limited processing capabilities, constrained bandwidth, and finite storage memory. The evaluation and decision-making approach introduced in [10] hybridized the CRITIC and TOPSIS methods, considering a diverse array of security criteria in line with the standards set by the International Organization for Standardization (ISO) and the National Institute of Standards and Technology (NIST). The experimentation from this study yielded insightful results, indicating KLEIN cipher as the most lightweight and secure choice among lightweight ciphers, including PRESENT-80, SEA, HIGH, LEA, AES Block Cipher, mCrypton, NOEKEON, Camellia, and the TEA numbers.

The uniqueness of our work lies in the fact that our proposal bases its decision-making on an extensive knowledge base, encompassing a wide array of encryption scenarios and cryptographic algorithms. Thus, our solution transcends the limitations of focusing on a specific domain or a well-defined environment. Instead, it strives to consolidate the wealth of expertise within the encryption field, with the aim of harnessing this knowledge for tailored decision-making in specific cases. Whether it's an ordinary network, an IoT network, or a P2P network, and whether the data is in the form of images, text, videos, or any other format, all these criteria serve as the parameters for our application in selecting the most appropriate encryption algorithm.

II. The Proposed Method

The proposed solution draws from one of the pioneering methods within the realm of artificial intelligence, known for its successful application in various fields. We've harnessed a Knowledge-Based System to address the domain of data encryption.

Our Knowledge-Based System comprises four fundamental modules [12], each encompassing a set of operations for knowledge processing and utilization:

- Module 1: Acquisition of Knowledge
- Module 2: Knowledge Representation
- Module 3: Knowledge Processing
- Module 4: Knowledge Utilization

In general, when crafting a Knowledge-Based System, three vital technical components are essential for system modeling and design [13]:

- The Knowledge Base
- The Fact Base
- The Inference Engine

Incorporating these technical components into a Knowledge-Based System, we adhere to the aforementioned modules, ensuring the system is enriched and leveraged effectively. In this work, we propose to adopt the architectural framework illustrated in the following diagram “Fig. 2” as the foundation for designing our decision-making Knowledge-Based System.

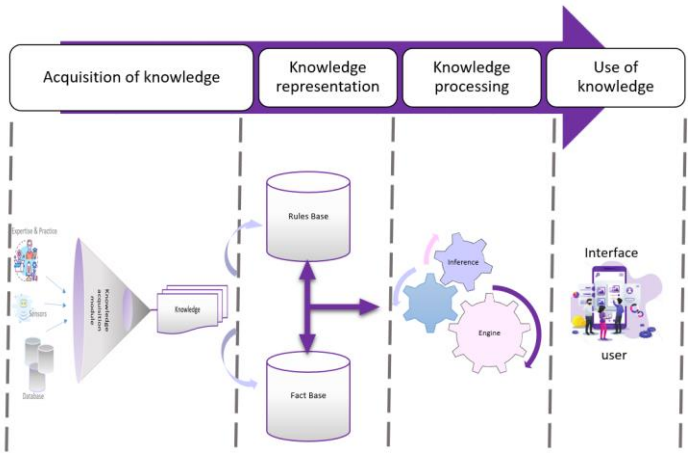


Fig. 2. The KBS Architecture for data ciphering.

This architecture offers precise control of the system, making maintenance more straightforward by segregating data analysis and classification from the decision-making component, which determines the most suitable rule.

To elaborate on the Ciphering Knowledge-Based System, the next section (see Section III) will begin by explaining the underlying principle of our proposed model within the context of data encryption. Subsequently, starting from Section III and continuing through to the end of this section, we will delve into the intricate steps carried out by Modules 1, 2, and 3 within this architectural framework. Notably, Module 4, which represents the graphical user interface (GUI) component of the application, is not covered in the forthcoming sections as it pertains to the user interaction and interface design.

III. RESEARCH METHOD

To construct a system of this nature, it's imperative to first establish a clear definition of knowledge and how it relates to our specific field of application. As articulated in [14], "Knowledge is the mobilization of one or more information in a well-defined context in order to trigger an action or produce knowledge" [14].

In the context of data exchange security, particularly in the pursuit of confidentiality, we will create a knowledge framework that encompasses various criteria influencing this confidentiality. As previously mentioned, these criteria span the source and destination environments, the network, data types, as well as the memory and energy capabilities of the devices involved in the process. Additionally, the encryption system itself must be highly reliable when employed under such conditions. Thus, our system must possess the capability to predict the most appropriate encryption algorithm to ensure that the ciphertext does not reveal any information about the plaintext. This can be quantified through metrics such as entropy or by considering factors like avalanche rate and other critical security criteria.

In the upcoming sections, we will provide a detailed explanation of the parameters that are taken into consideration within this solution.

A. Acquisition of Knowledge

The initial module crucial to the construction of this system is the acquisition of knowledge. Within this module, we must delineate the various potential sources for gathering knowledge pertinent to our issue. This acquisition can be realized through collaboration with experts in the field and/or by interfacing with databases from existing systems, which have been informed by established practices and historical data.

This module unfolds in three pivotal steps:

1) Data classification: As our approach involves data from diverse sources, our initial step is to classify this data based on the environment, structure, and nature of the information [15]. This classification allows us to extract the most critical insights. The outcomes of this classification process are subsequently consolidated in a temporary database.

2) Extraction of information: Next, we move on to extract the most relevant information that directly influences the effectiveness of the encryption system. This includes:

- Message type
- The percentage of images compared to all the data to be encrypted
- The percentage of the video to be encrypted compared to all the data to be encrypted
- The percentage of the Literal text compared to the set of data to be encrypted
- The size of the message
- Device type
- Device capacity
- Network type
- Network size
- Etc.

3) Building knowledge: The objective of this analysis is to enrich the knowledge base with the actual results of the encryption performed.

The goal of this analysis is to enrich the knowledge base with the real-world outcomes of the encryption procedures. During this phase of populating the knowledge base, we lay the foundation for utilizing it in the decision-making process. Here, we execute the integrated encryption algorithms within the system and subsequently compute indicators pertaining to their performance and security levels.

The knowledge we aim to capture during this phase includes:

Encryption execution time

Memory used for encryption

Decryption execution time

Memory used for decryption

Entropy of the encrypted message [16]

This stage is invoked at two distinct points in the life cycle of this approach:

In the Run-in phase: This phase is corresponds to the training phase of the system. It facilitates the generation of new experiences and the enrichment of the decision-making knowledge base.

In the Enrichment phase: This is the final step, occurring after each execution of this method, wherein the selected encryption algorithm is applied.

B. Knowledge Representation

Knowledge can take on various forms, including declarative knowledge, procedural knowledge, and a structured category that combines both, often referred to as meta-knowledge.

Declarative knowledge involves specifying how an action is performed using conditional statements (If... Then...). It's essentially the "facts" within a Knowledge-Based System (KBS). This type of knowledge encapsulates the logic of an operation, defining the objects and concepts that lead to a specific "fact." Declarative knowledge allows for the inclusion of diverse facts from different domains, making it versatile. However, this architecture tends to be slower because it relies on interpreting procedures during each execution.

On the other hand, procedural knowledge already provides instructions, as it outlines the logic of actions in the form of "rules, procedures, strategies, and agendas." The advantage of this architecture is speed, as knowledge is codified in compiled procedures, ready for execution. However, accessing data can be more complex, as it's embedded within the compiled code.

In the realm of AI, knowledge representation in each category is formalized differently and can take on various forms :

In “Fig. 3” we represent an example of the Triplet Knowledge such as: Triplet <object, attribute, value>:

In this context, the "object" refers to the subject that needs to undergo processing, the "attribute" represents the specific property or characteristic of interest, and the "value" corresponds to the specific value associated with this property.. Example:

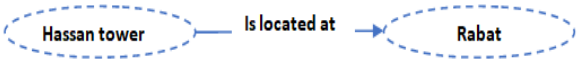


Fig. 3. Example of triplet knowledge representation.

And “Fig. 4” represent Logical formula: Indeed, by employing predicates and propositions in logical formulas, we can effectively depict a particular situation. Example:

If a person is female, and is a parent. Then that person is someone's mother:

- F(X): "x is a female", P(x): "x is a parent"
- M(x,y): "x is the mother of y"
- $\forall x\,(F(x) \wedge P(x) \rightarrow \exists\, y$ such that $M(x,y))$

Fig. 4. Example of Logical formula of Knowledge representation.

Semantic network: Exactly, in a semantic network, concepts are represented as nodes, and the relationships between these concepts are depicted as arcs or edges. This visual structure “Fig. 5” helps convey how different concepts are interconnected. Example:

Fig. 5. Example of Semantic network of Knowledge representation

Rule: Establishing connections between pieces of information, thereby extracting additional insights, is pivotal for drawing conclusions regarding relationships, strategies, directives, heuristics, and more. Example:

If <Flower, Color, Rose> Then “I like the Flower”

If “I like the Flower” then “I buy the Flower”

If “I like the Flower” and <Packaging, Price, Free> Then “I buy a bouquet of Flowers”.

The embodiment of knowledge within our Knowledge-Based System (KBS) can be described as a fusion of the Triplet and Rules concepts. Consequently,, to present an encryption experiment applied to certain values of the previously defined criteria, we consider tables for each object. Each table serves as a dimension within our decision analysis framework, with each criterion serving as an attribute within that dimension, encompassing multiple potential values. If we consider the criteria discussed earlier, we will find ourselves working with, at the very least, the following sheme; “Fig. 6”:

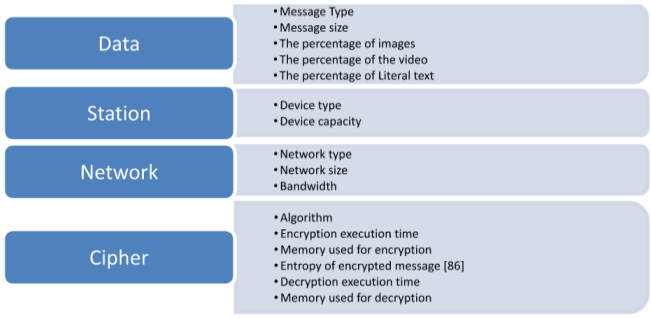


Fig. 6. Decision analysis of knowledge-based encryption.

For example, an experience in this knowledge base can be translated as following:

- If (Triplet <Data, type, 'image'> and Triplet <Data, size, '10587'> and Triplet <Station, type, IoT> and Triplet <Cipher, Time, 20ms>) Then Triplet “Cipher, Algo, AES”.

In a general context, each unit of knowledge can be subdivided into two distinct components. The first part embodies the conditions necessary to trigger an action, referred to as "Premises." The second part encompasses the consequences that result from the activation of this action, denoted as "Conclusions." These Conclusions, in turn, correspond to outcomes that may either initiate subsequent actions or determine the ultimate state of affairs, often characterized as "Facts."

To maximize the efficient utilization of the acquired knowledge, we opt for a knowledge base structured around a decision-making architecture in “Fig. 7”. This framework allows for the organization of these various elements into separate, independent structures. Specifically, we establish:

- A database of dimensions: This database consolidates the distinct characteristics pertaining to each object of analysis.

- A database of Facts: Within this repository, we house all the factual information that delineates diverse potential scenarios, along with their corresponding encryption outcomes.

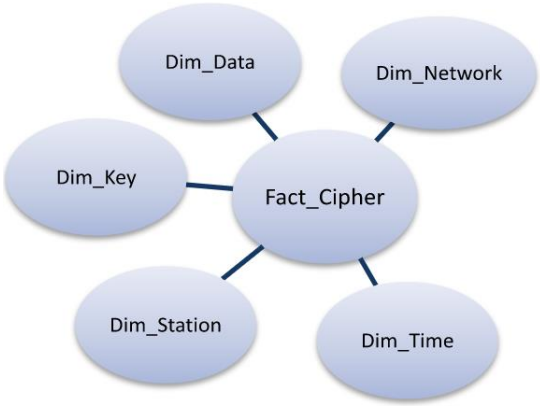


Fig. 7. Decision-making architecture of the knowledge-based encryption.

Going back to the example given above, we have:

- [Triple <Data, type, 'image'> and Triplet <Data, size, '10587'>]: represents a row of the Data dimension with a unique identifier: id_data.
- Triplet <Station, type, IoT>: represents a line of the Station dimension with a unique identifier: id_stat.
- Triplet <Cipher, Time, 20ms>: represents a line of the Cipher dimension with a unique identifier: id_cipher.

The rule is stored in the fact table as follows:

- If (id_data and id_stat and id_cipher) Then id_cipher.

C. Knowledge Processing

1) Inference engine: The inference engine, comprising a series of computer instructions, facilitates the process of logical reasoning in alignment with the knowledge and expertise encapsulated within the knowledge base. It harnesses the rule base, executing a sequence of logical inferences and ultimately deducing fresh insights to achieve predefined objectives [17]. To perform its task effectively, the inference engine must be able to detect and handle the following cases:

- Designate the set of rules of the BR to be compared with the facts of the BF.
- Specify the scheduling of these rules in line with the requested need.
- Trigger the execution of the chosen rules according to the sequence strategy previously specified.
- Detect and apply factbase updates.
- Manage duplicate rules and eliminate rules that are already in use.
- Detect rules that can cause confusion and eliminate contradictions.

a) Designate the set of rules: In this phase, we recommend employing the Nearest Neighbor search (NN) algorithm to identify the rules that closely align with our specific real-world case. This algorithm actively queries the knowledge base, taking into account the pre-established criteria, and retrieves all the rows that exhibit dominance concerning these criteria.

b) Nearest Neighbor Search Algorithm (NN)

i) Dominance concept: In a dataset comprising multidimensional objects, a relationship is deemed one of dominance if the dominant object excels in at least one dimension while maintaining a high level of performance across all other dimensions [17]. To identify all such dominant objects within a database of multidimensional entities, we leverage the Skyline operator. This operator empowers our query to retrieve a collection of points that remain unchallenged by any other object, aptly referred to as Skyline points [18].

Illustrating the concept of the Skyline operator, consider the common scenario of seeking a hotel near the beach at a significantly reduced cost. The criteria for this search may often entail trade-offs and appear somewhat contradictory, potentially yielding no results with a conventional selection query. However, it becomes crucial to assist the user in finding a combination that aligns more closely with their preferences. The Skyline operator fulfills this need by presenting the user with a set of the most appealing hotels in accordance with their specified requirements. As depicted in the example below, the curve highlights the Skyline hotels, each of which stands unchallenged by any other hotel in terms of the defined criteria specified in “Fig. 8”.

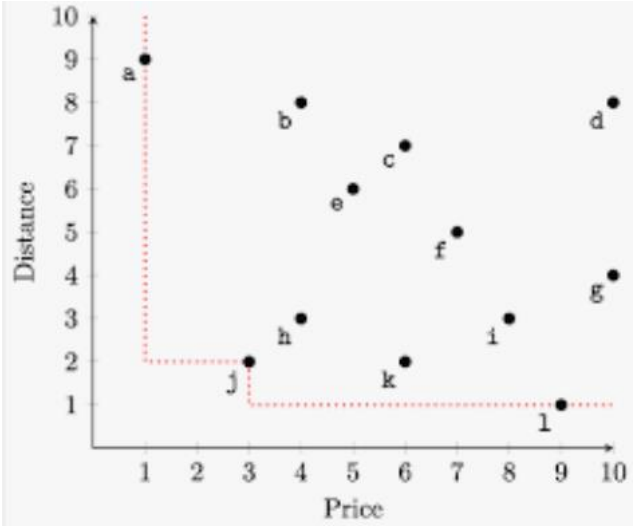


Fig. 8. Curve of elected skyline hotels.

Nonetheless, processing Skyline requests presents a substantial challenge due to the requirement of handling vast amounts of data in memory, a factor that considerably amplifies the algorithmic complexity of this operator [19]. To address this issue, multiple solutions have been developed, focusing on leveraging secondary memory in the Skyline point search process. These solutions fall into two distinct

categories: non-index-based algorithms and index-based algorithms.

In our research, we opt for the latter category, specifically embracing index-based algorithms. One such algorithm is the Nearest Neighbor (NN) algorithm, which employs a nearest neighbor search method based on an appropriate distance function suited to various values within the targeted search set. The algorithm effectively partitions the space into regions and systematically identifies points closest to the origin of each region based on a monotonically decreasing distance function, often exemplified by the Euclidean distance. Furthermore, the algorithm continuously evaluates the dominance of candidate objects within each region. Regions dominated by a candidate are progressively eliminated from consideration, and this process continues until the list is ultimately empty [20].

c) Specifying rule ordering: After obtaining all the Skyline points using the Nearest Neighbor (NN) method, the next step involves arranging these points in a user-preferred order and categorizing them based on the name of the encryption algorithm. The chosen encryption algorithm is then identified as the first entry in the sorted list.

d) Detect database updates: The selected encryption algorithm is subsequently applied to the data source to secure its exchange. Simultaneously, the pertinent values associated with the execution of this specific experiment are recorded within our knowledge base.

At this juncture, the application process reaches its conclusion. It's worth noting that the system does not incorporate steps related to managing duplicate rules or addressing potential confusions, as each encryption operation carried out within the system is treated as an independent case.

IV. Results and Discussion

A. Experiment

The tests carried out in this work are based on a technical environment of 16GB of RAM and an Intel(R) Core(TM) i7-6700HQ, 2.60GHz, 64bit OS processor.

The data stored in the knowledge base was generated via an application combining the following five encryption algorithms: AES, Blowfish, DES, TripleDES and ASEC.

The application takes different types of input; it performs an analysis to be able to identify their fixed characteristics (type, size, and environment). Then, it applies the encryption algorithms mentioned above, and as an output, it sends the performance study of each algorithm (execution time, memory capacity used, entropy). Initially, we focused on a sample of 200 different entries, including 100 with the same fixed characteristics.This makes a total of more than a thousand lines of tests.

1) Test scenarios: The experiments conducted within the scope of this contribution were structured around a defined set of test scenarios. Each test case was meticulously designed in accordance with the number of dimensions and their specific types. An effort was made to encompass as many diverse choices as possible, tailored to our specific context. The application, as part of its execution, queries the previously configured knowledge base and deploys the Skyline Nearest Neighbor (NN) algorithm [20], using the input criteria. Subsequently, it returns the Skyline points that correspond to the chosen encryption algorithm.

The primary objective of these experiments is to showcase the efficacy of Skyline algorithms across various specified cases, ranging from scenarios with as few as two dimensions to more complex scenarios with up to ten dimensions. Additionally, some test scenarios incorporate additional dimensions featuring fictitious values, enabling us to consider a broader spectrum of dimensions that are typically associated with network, platform, or transport channel criteria. Unfortunately, not all of these dimensions could be simulated and integrated into the system.

The details of the different test scenarios are outlined in Table I.

TABLE I. Test Scenarios to Experiment the Knowledge-Based Encryption

N°	No. Dimension	Dimensions	Dominance criterion
Scenario 1	2	CipheringRuntime	Min
		Entropy	Max
Scenario 2	2	DecipheringRuntime	Min
		DecipheringMemory	Min
Scenario 3	3	CipheringRuntime	Min
		CipheringMemory	Min
		Entropy	Max
Scenario 4	4	CipheringRuntime	Min
		DecipheringRuntime	Min
		CipheringMemory	Min
		Entropy	Max
Scenario 5	6	CipheringRuntime	Min
		DecipheringRuntime	Min
		CipheringMemory	Min
		DecipheringMemory	Min
		Entropy	Max
		Dim_fictive_1* (limited No. values)	Min
Scenario 6	6	CipheringRuntime	Min
		DecipheringRuntime	Min
		CipheringMemory	Min
		DecipheringMemory	Min
		Dim_fictive_1* (limited No. values)	Max
		Dim_fictive_2* (limited No. values)	Min
Scenario 7	10	CipheringRuntime	Min
		DecipheringRuntime	Min
		CipheringMemory	Min
		DecipheringMemory	Min
		Entropy	Max
		Dim_fictive_1 (limited No. values)	Min
		Dim_fictive_2 (limited No. values)	Min

		Dim_fictive_3* (limited No. values)	Min
		Dim_fictive_4* (limited No. values)	Min
		Dim_fictive_5* (limited No. values	Min
Scenario 8	10	CipheringRuntime	Min
		DecipheringRuntime	Min
		CipheringMemory	Min
		DecipheringMemory	Min
		Dim_fictive_1 (limited No. values)	Max
		Dim_fictive_2 (limited No. values)	Min
		Dim_fictive_3 (limited No. values)	Min
		Dim_fictive_4 (limited No. values)	Min
		Dim_fictive_5 (limited No. values)	Min
		Dim_fictive_6* (limited No. values)	Min

[* : The Dim_fictive_i represent fictitious dimensions whose values have been generated with an approximate rule to designate other constraints. These dimensions have been added as an indication in order to test the impact of the number of dimensions on the performance of the system.]

B. Results

1) Test of Solution quality: To test the quality of the elected solution, we will focus more on the scenarios with 2, 3 and 4 dimensions containing the entropy and in which we have the real values of their executions. Indeed, focusing on the entropy dimension will allow us to decide whether the chosen cipher is well secured or not since this dimension reflects the amount of information on the source and contained in the cipher.

a) Result of scenario 1: The first scenario goes up 16 Skyline lines, the first four of which all designate the Blowfish algorithm, followed by the AES, ASEC and then 3DES algorithms. The following Table II shows the results of this run:

TABLE II. ELECTED SKYLINE ENCRYPTION OF SCENARIO 1

Algorithm	Runtime	Entropy
BLOWFISH	25	3.8870066417181426
BLOWFISH	49	3.8777830337525954
BLOWFISH	55	3.857533855872884
BLOWFISH	65	3.852401615754532
AES	67	3.821903635277186
AES	73	3.808205259874092
AES	77	3.769139091307226
AES	78	3.7276250338839367
ASEC	107	3.1246923931800934
ASEC	110	2.992846680894221
3DES	123	2.8588450604683873
3DES	229	2.8533695224817235
3DES	334	2.8444354220858403
3DES	353	2.837662090839393
3DES	500	2.830898867431037
3DES	721	2.8282605678748394

b) Result of scenario 2: Scenario 2 pulls up 5 Skyline lines, all of which point to the Blowfish algorithm. The following Table III shows the results of this execution:

TABLE III. ELECTED SKYLINE ENCRYPTION OF SCENARIO 2

Algorithm	Ciphering Time	Memory Used	Entropy
BLOWFISH	25	8.492485106920858	3.8870066417181426
BLOWFISH	29	7.911124302269485	3.9116333559383376
BLOWFISH	37	7.264124532937471	3.903903802424076
BLOWFISH	135	7.234850719105421	3.8894956802766205
BLOWFISH	169	7.231180846946539	3.856438845863682

c) Result of scenario 3: Scenario 3 pulls up 6 Skyline lines, all of which point to the Blowfish algorithm. The following Table IV shows the results of this run:

TABLE IV. ELECTED SKYLINE ENCRYPTION OF SCENARIO 3

Algorithm	Ciph-ering Time	Mem-ory Used	Entropy	Decip-hering Time
BLOWFISH	24	9.09971086299	3.922074380780725	36
BLOWFISH	25	7.785557142466666685	4.003422725072732	37.5
BLOWFISH	39	7.324437668623221	4.028405849999899	58.5
BLOWFISH	94	7.247478714871018 5	4.027019099838717	141
BLOWFISH	180	7.236124737916946	3.865031008 2953587	270
BLOWFISH	188	7.222306158361476 5	3.893535489 9007044	282

C. Discussion

1) Solution quality: Based on all the executed experiments, it is evident that the data intended for encryption in this study can be effectively secured using the BLOWFISH algorithm. The system's predictions consistently favored the selection of the BLOWFISH algorithm in all scenarios, particularly those emphasizing the "entropy" dimension, as well as dimensions related to encryption time and memory usage. These dimensions collectively provide a robust basis for evaluating the chosen solution.

Entropy, as a metric, serves as a reliable indicator of the security level and the solution's resistance to various forms of attacks [16], while the dimensions of encryption time and memory usage offer valuable insights into the solution's performance and associated costs.

Furthermore, it's noteworthy that the Skyline points returned by the system exhibit a growing level of agreement, consistently converging on the same solution [18]. This marks a notable departure from the previous encryption method [11], where multiple solutions were viable. Such convergence obviates the need for managing preferences and priorities among the criteria used.

To validate the predictive results and ensure their accuracy, the chosen solution is executed, and pertinent measurements are calculated. The ensuing graph visually depicts the outcome of this comparative analysis.

The following graph shows in "Fig. 9" the result of this comparison:

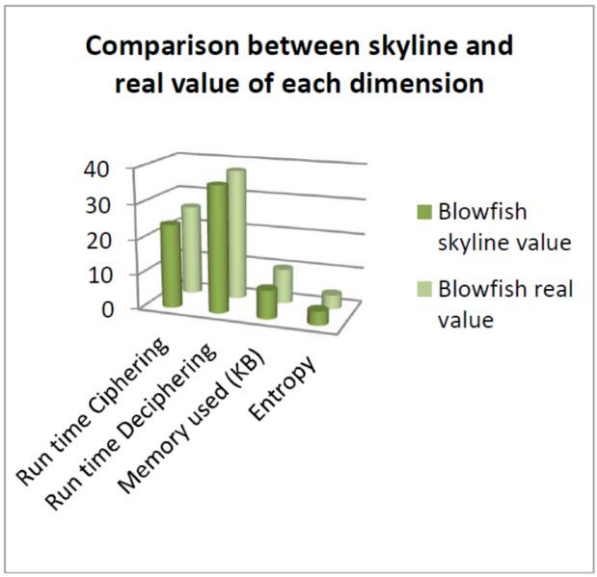


Fig. 9. Comparison between skyline solution value and real value of each dimension.

From this comparison, we can conclude that the difference between the values that were reported by the predictions of the system and the concrete values given by the application of the chosen solution is really negligible. The new values are very close to those predicted by the system.

2) System performance: Now, to see the performance of this system, we ran the set of predefined scenarios and we calculated the time that the system takes to return its results. The graph below "Fig. 10" shows the evolution of the execution time of the system according to the number of dimensions while comparing with the evolution of the old method:

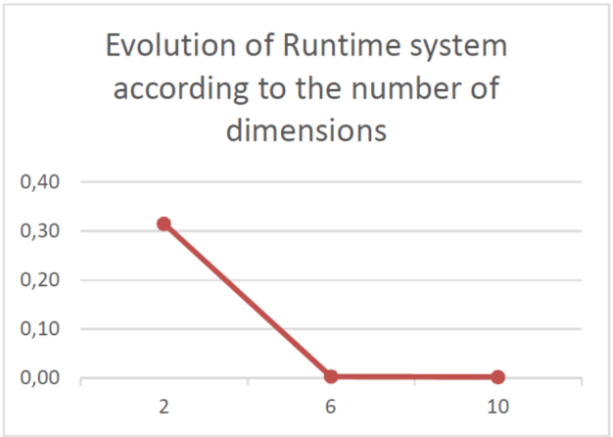


Fig. 10. Evolution of runtime system according to the number of dimensions.

As per the data illustrated in the graph, it becomes evident that the system's performance consistently improves as the number of dimensions increases. This observation suggests that the Nearest Neighbor (NN) algorithm is highly suitable for our configuration, offering enhanced scalability for accommodating various settings. In contrast, the previous intelligent encryption method [11], reliant on the BNL algorithm, experiences a decline in performance as the number of dimensions increases.

The accompanying graph in "Fig. 11" further elucidates this comparative analysis, emphasizing the advantages of the NN algorithm in handling increased complexity and dimensionality, making it a preferred choice for our system.

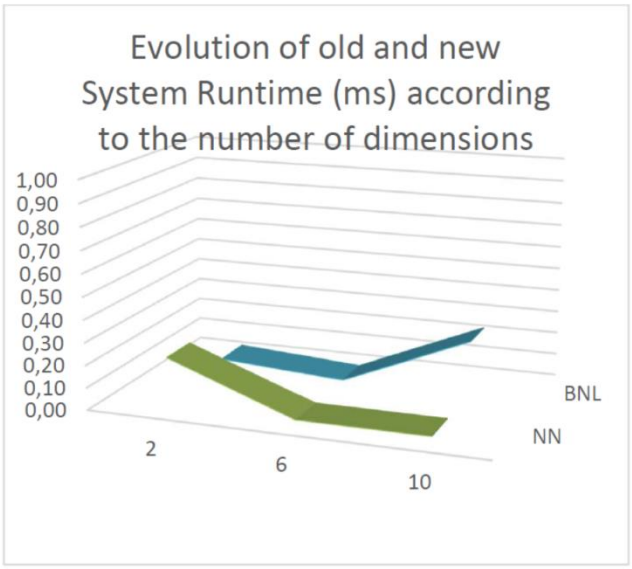


Fig. 11. Evolution of old and new System Runtime (ms) according to the number of dimensions.

3) System maintainability: As previously discussed, the new Knowledge-Based System (KBS) encryption system is constructed upon a modular architecture, where it comprises distinct modules, each responsible for specific functions. This modular design enhances code organization and simplifies system maintenance [22] [23]. Furthermore, the system incorporates a collection of well-established, standardized encryption algorithms, each with a proven track record of robustness and security.

These two key attributes, modularity and standardization, play a pivotal role in the quality of computer system development [21]. They promote system evolution and facilitate maintenance in a flexible and straightforward manner, ultimately contributing to the overall efficiency and effectiveness of the system.

V. CONCLUSION

Throughout this work, our focus has been on the novel approach to intelligent encryption, one grounded in decisional concepts to enhance the security of data exchanges. To boost its performance, this new approach harnesses the framework of a knowledge-based system. This architectural choice allows

us to effectively segregate the processes of data analysis and classification from the construction of knowledge to decision-making. This separation significantly bolsters the system's performance in terms of both the quality of the selected solution and the execution cost.

Our Knowledge-Based System (KBS) for Ciphering incorporates well-defined technical components. The knowledge base, for instance, has been meticulously modeled in a multidimensional fashion, and the inference engine is enriched with the Nearest Neighbor (NN) algorithm within its inference engine to formulate the encryption policy to be adhered to. We have subjected the system to various test cases, each of which explores a different combination by altering the number and type of dimensions (criteria). This approach has allowed us to quantitatively assess the quality and performance of this innovative solution.

Consequently, we've conducted a comprehensive comparative study, juxtaposing the primitives used in this work with the old method. This examination serves to highlight the advantages and advancements brought about by our new architecture, which is built upon separate modules and established security standards. Ultimately, this design choice renders the system more flexible and easier to maintain.

In summary, our system has demonstrated the potential to offer an excellent quality-to-cost ratio for the encryption processes it facilitates, underscoring its efficiency and effectiveness in securing data exchanges.

REFERENCES

[1] Md. M. Ahamad and Md. I. Abdullah, "Comparison of Encryption Algorithms for Multimedia," *Rajshahi Univ. j. sci. eng.*, vol. 44, pp. 131–139, Nov. 2016, doi: 10.3329/rujse.v44i0.30398.

[2] S. R. Ellis, "Chapter 63 - Fundamentals of Cryptography," in *Computer and Information Security Handbook (Second Edition)*, J. R. Vacca, Ed. Boston: Morgan Kaufmann, 2013, pp. 1031–1038. doi: 10.1016/B978-0-12-394397-2.00063-5.

[3] R. Mouachi *et al.*, "A Choice of Symmetric Cryptographic Algorithms based on Multi-Criteria Analysis Approach for Securing Smart Grid," *Indian Journal of Science and Technology*, vol. 10, no. 39, pp. 1–9, Dec. 2017, doi: 10.17485/ijst/2017/v10i39/119856.

[4] F. Omary, A. Mouloudi, A. Tragha, and A. Bellaachia, "A new ciphering method associated with evolutionary algorithm," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 3984 LNCS, pp. 346–354, 2006, doi: 10.1007/11751649_38.

[5] S. Trichni, F. Omary, A. Idrissi, M. Bougrine, and M. Abourezq, "New intelligent strategy for encryption decisional support system," *International Journal of High Performance Systems Architecture*, vol. 9, no. 4, pp. 173–181, 2020, doi: 10.1504/IJHPSA.2020.113678.

[6] M. Bougrine, F. Omary, and S. Trichni, "Security of a new hybrid ciphering system," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 6, pp. 694–699, 2020.

[7] M. N. A. Wahid, A. Ali, B. Esparham, and M. Marwan, "A Comparison of Cryptographic Algorithms: DES, 3DES, AES, RSA and Blowfish for Guessing Attacks Prevention," p. 7, 2018.

[8] A. Devi, A. Sharma, and A. Rangra, "Performance analysis of Symmetric Key Algorithms: DES, AES and Blowfish for Image encryption and decryption," vol. 4, no. 6, p. 6, 2015.

[9] M. Oulehla and D. Malanik, "Comparison of cryptographic methods Triple DES, AES and a method based on the arithmetic of elliptic curves (ECC) on the Android mobile platform. - extended version," *International Journal of Computers and Communications*, vol. 9, p. 62, Jan. 2015.

[10] L. Ning, Y. Ali, H. Ke, S. Nazir, and Z. Huanli, "A Hybrid MCDM Approach of Selecting Lightweight Cryptographic Cipher Based on ISO and NIST Lightweight Cryptography Security Requirements for Internet of Health Things," *IEEE Access*, vol. 8, pp. 220165–220187, 2020, doi: 10.1109/ACCESS.2020.3041327.

[11] S. Trichni, F. Omary, and M. Bougrine, "New Smart Encryption Approach based on Multidimensional Analysis Tools," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 5, pp. 666–675, 2021, doi: 10.14569/IJACSA.2021.0120579.

[12] Systèmes et applications intelligents: Actes de la conférence sur les systèmes intelligents 2021 (IntelliSys) Volume 1.

[13] Janiesch, C., Zschech, P. et Heinrich, K. Apprentissage automatique et apprentissage profond. Marchés électroniques 31.685-695 (2021). https://doi.org/10.1007/s12525-021-00475-2.

[14] C.Zhang, Yu.Xie, H. Bai, B. Yu, W. Li, Y. Gao. "A survey on federated learning" Knowledge-Based Systems,Volume 216, 15 March 2021, 106775

[15] T. Hamon, "Modélisation et Représentation des Connaissances - Introduction," Institut Galilée - Université Paris 13, 2019.

[16] M. Lefevre, "CM8 : Système à Base de Connaissances," p. 61, 2020.

[17] N. Sendrier, "Introduction à la théorie de l'information," p. 70.

[18] J.-L. Ermine, "Les systèmes de connaissances," p. 145.

[19] M. ABOUREZQ, "Cloud Service Selection using the Skyline and Multi Criteria Decision Aiding," Mohammed V University of Rabat, 2017.

[20] S. Börzsönyi, D. Kossmann, and K. Stocker, "The Skyline operator," *Proceedings 17th International Conference on Data Engineering*, 2001, doi: 10.1109/ICDE.2001.914855.

[21] S. Berchtold, C. Böhm, D. Keim, and H. Kriegel, "A cost model for nearest neighbor search in high-dimensional data space," 1997. doi: 10.1145/263661.263671.

[22] Younoussi, Siham & Roudies, Ounsa. (2016). Capability and maturity model for Reuse: A comparative study. 302-308. 10.1109/CloudTech.2016.7847714.

[23] Zhenan Tu, "Research on the Application of Layered Architecture in Computer Software Development." Journal of Computing and Electronic Information Management. 11. 34-38. 10.54097/jceim.v11i3.08.