Kerberos Protocol: History, Steps, and Security Analysis

Volume 5 NO.1 YEAR 2023

مجلة العلوم والتطـبيقات الهـندسية

Journal of Science And Engineering Application

ISSN 2521-3911

Kerberos Protocol: History, Steps, and Security Analysis

1st Hassan M. Al-Jawahry
Faculty of Engineering-The Islamic University
Computer Science Department-Faculty of Computer Science and Mathematics
University of Kufa
Najaf, Iraq
hassanaljawahry@gmail.com
2nd Sahar Adill Kadum
Computer Science Department-Science collage for woman
Babylon University
Babylon, Iraq
dr.sahar.adill@gmail.com

بروتوكول Kerberos: التاريخ والخطوات وتحليل الأمان

م.م حسن الجواهري
كلية الهندسة -الجامعة الاسلامية
قسم علوم الحاسبات - كلية علوم الحاسبات والرياضيات
جامعة الكوفة
العراق-النجف الاشرف
hassanaljawahry@gmail.com
م.م سحر عادل كاظم
قسم علوم الحاسبات - كلية العلوم للبنات
جامعة بابل
العراق-بابل
dr.sahar.adill@gmail.com

Received 1/5/2023

Accepted 26/5/2023

Abstract –

Kerberos is a widely used authentication protocol that provides secure access to network resources by authenticating users and services. This paper provides an overview of the Kerberos protocol, its history, and its steps. The Kerberos protocol was initially developed at MIT in the 1980s to solve the problem of authentication in distributed systems. It has since become a widely adopted protocol in enterprise networks and is implemented in many operating systems and applications. This paper outlines the steps involved in the Kerberos protocol, including authentication, authorization, and ticket granting. It also covers the security features of the protocol, including encryption, mutual authentication, and delegation. The research on Kerberos protocol has mainly focused on improving its security, scalability, and usability. Various researchers have proposed different enhancements to the Kerberos protocol to address its weaknesses, such as using stronger encryption algorithms, improving the key distribution process, and implementing additional security features. Furthermore, this paper discusses the most important attacks on the Kerberos protocol, such as the Golden Ticket attack, Password guessing attacks and Replay attacks. These attacks exploit vulnerabilities in the protocol to gain unauthorized access to network resources or impersonate users

مجلة العلوم والتطـبيقات الهـندسية

Journal of Science And Engineering Application

ISSN 2521-3911

or services. The paper also presents the countermeasures and mitigations that can be employed to prevent these attacks and improve the overall security of the Kerberos protocol.

خلاصة-

Kerberosهو بروتوكول مصادقة مستخدم على نطاق واسع يوفر وصولاً آمنًا إلى موارد الشبكة عن طريق مصادقة المستخدمين والخدمات. تقدم هذه الورقة لمحة عامة عن بروتوكول Kerberos وتاريخه وخطواته. تم تطوير بروتوكول Kerberos في البداية في معهد ماساتشوستس للتكنولوجيا في الثمانينيات لحل مشكلة المصادقة في الأنظمة الموزعة. ومنذ ذلك الحين أصبح بروتوكولًا معتمدًا على نطاق واسع في شبكات المؤسسات ويتم تنفيذه في العديد من أنظمة التشغيل والتطبيقات. توضح هذه الورقة الخطوات المتضمنة في بروتوكول Kerberos، بما في ذلك المصادقة والترخيص ومنح التذاكر. يغطي أيضًا ميزات أمان البروتوكول ، بما في ذلك التشفير والمصادقة المتبادلة والتفويض. ركز البحث في بروتوكول Kerberos بشكل أساسي على تحسين الأمان وقابلية التوسع وسهولة الاستخدام. اقترح العديد من الباحثين تحسينات مختلفة على بروتوكول Kerberos لمعالجة نقاط ضعفه ، مثل استخدام خوارزميات تشفير أقوى ، وتحسين عملية توزيع المفاتيح ، وتنفيذ ميزات أمان إضافية. علاوة على ذلك ، تناقش هذه الورقة أهم الهجمات على بروتوكول Kerberos ، مثل هجوم التذكرة الذهبية وهجمات تخمين كلمة المرور وهجمات إعادة التشغيل. تستغل هذه الهجمات الثغرات الأمنية في البروتوكول للحصول على وصول غير مصرح به إلى موارد الشبكة أو انتحال صفة المستخدمين أو الخدمات. تعرض الورقة أيضًا الإجراءات المضادة ووسائل التخفيف التي يمكن استخدامها لمنع هذه الهجمات وتحسين الأمان العام لبروتوكول**Kerberos.**

Keywords: Kerberos Protocol, Authentication, Authorization, Ticket Granting, Security, Encryption, Mutual Authentication, Attacks.

الكلمات المفتاحية: بروتوكول Kerberos ، المصادقة ، التفويض ، منح التذاكر ، الأمان ، التشفير ، المصادقة المتبادلة ، الهجمات.

I. Introduction

Kerberos is a network authentication protocol that provides a secure method of authentication for client-server applications over an insecure network. The primary goal of the Kerberos protocol is to establish a secure channel of communication between a client and a server without transmitting any sensitive information in plain text over the network. The Kerberos protocol is based on the concept of a trusted third party, known as the Kerberos Key Distribution Center (KDC). The KDC acts as an intermediary between clients and servers, verifying the identity of clients and granting them access to servers based on their credentials. The Kerberos protocol was developed at the Massachusetts Institute of Technology (MIT) in the early 1980s as a solution to the problem of securing network communications. The original idea was to create a system that could authenticate users and provide secure access to network resources over an insecure network. The name "Kerberos" was inspired by the three-headed dog in Greek mythology that guarded the entrance to the underworld. The name was chosen to reflect the security goals of the protocol, which was to guard against unauthorized access to network resources [1, 2, 3, 4].

II. Kerberos Protocol Versions

There are several versions of the Kerberos protocol, each with different features and capabilities. The main versions of Kerberos are v1, v2, v3, and v4, with v5 being the current version. Here are the differences between each version [2, 4, 5, 6]:

1. Kerberos v1: Kerberos v1 was the first version of the protocol and was released in the early 1980s. It was based on the Needham-Schroeder protocol and was vulnerable to several security attacks, including replay attacks and man-in-the-middle attacks. It was also limited in scope and could only be used with Unix-based systems.

2. Kerberos v2: Kerberos v2 was released in the mid-1980s and was a significant improvement over v1. It introduced several new features, such as support for mutual authentication and support for network services that run on non-Unix platforms. However, it was still vulnerable to some security attacks, and it lacked support for stronger encryption algorithms.

3. Kerberos v3: Kerberos v3 was a complete redesign of the protocol and was released in the early 1990s. It was much more secure than its predecessors and introduced support for stronger encryption algorithms, support for attribute-based authentication, and support for multiple realms. It also introduced a new message format that made it incompatible with earlier versions of Kerberos.
4. Kerberos v4: Kerberos v4 was released in the late 1980s and was widely used in the 1990s. It introduced several new features, such as support for proxy authentication and support for encrypted tickets. However, it had some significant security weaknesses, including a vulnerability to dictionary attacks, and it was replaced by Kerberos v5 in the early 1990s.
5. Kerberos v5: Kerberos v5 is the current version of the protocol and was released in the late 1990s. It is a complete redesign of the protocol and is much more secure than earlier versions. It supports strong encryption algorithms, supports attribute-based authentication, and is more flexible than earlier versions. It also includes support for cross-realm authentication, which allows users from different Kerberos realms to authenticate with each other.

III. Kerberos Protocol Steps

The protocol involves three main entities: the client, the Kerberos server, and the network service. Here are the steps of the Kerberos protocol in detail (as shown in Fig. 1) [2, 4, 7]:

1. Client authentication request: The first step is for the client to send an authentication request to the Kerberos server. The request contains the user's identity and a timestamp.
2. Kerberos server authentication: The Kerberos server receives the authentication request and verifies the user's identity. The server then generates a session key and creates a ticket-granting ticket (TGT) that includes the session key and the user's identity. The TGT is encrypted using the Kerberos server's secret key.
3. TGT delivery: The Kerberos server sends the TGT back to the client. The client then decrypts the TGT using the Kerberos server's secret key.
4. Network service ticket request: The client sends a ticket request to the Kerberos server, requesting a ticket to access the desired network service. The request includes the TGT and the user's identity.
5. Ticket-granting service (TGS) authentication: The Kerberos server receives the ticket request and verifies the TGT and the user's identity. The server then generates a session key and creates a service ticket that includes the session key and the user's identity. The service ticket is encrypted using the network service's secret key.

6. Service ticket delivery: The Kerberos server sends the service ticket back to the client. The client then decrypts the service ticket using the network service's secret key.
7. Network service authentication: The client sends the service ticket to the network service, along with a timestamp and a message indicating the desired service. The network service verifies the authenticity of the service ticket and the timestamp. If the ticket and timestamp are valid, the network service accepts the client's request and creates a new session key for the client and the service.
8. Service request fulfillment: The network service uses the new session key to encrypt and send data back to the client. The client then decrypts the data using the session key.

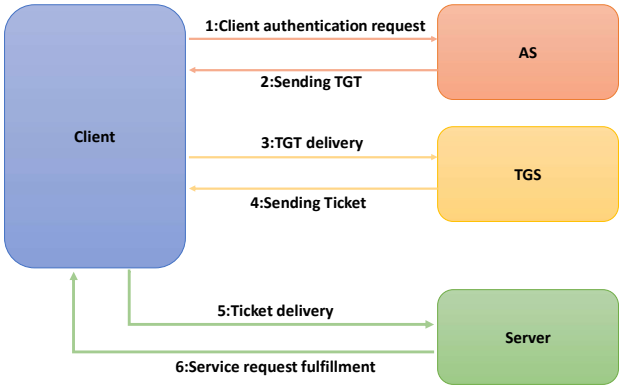


Fig. 1. Kerberos Protocol Steps

IV. Literature Review

[8] In 2016, Moin a. Khorajiya and Gardas Naresh Kumar presented a strategy to enhance the current framework which integrate PGP between users and the Kerberos server so that each user is signed in digitally for making the authentication to Kerberos, preventing man-in-the-middle attacks and securing non-repudiation.

An application called pretty good privacy can encode and decode emails sent by the internet. This confidentiality and authentication phenomenon was created by Phil Zimmerman and can be applied to secure communication and file storage. He incorporates encryption algorithms into general-purpose applications as building blocks.

PGP offers the following five services:

1) Authentication 2) confidentiality 3) compression 4) e mail compatibility 5) segmentation

Method for authentication in steps:

1. The user uses a key to encrypt and submit to PGP the specific information he wants to validate.

2. Digitally sign the data with PGP and sending to KDC

3. A ticket issues by KDC and return this ticket to PGP.

4. Sign by digitally the ticket once more with PGP and deliver it to the user. With his key, the user encrypts the ticket

5. The user has access to the server of his choice

Factors improved in the suggested design:

1. Protection from man-in-the-middle attacks: tickets have a digital signature encoded in them, preventing man-in-the-middle attacks that would allow someone to steal the secured data from the tickets.

2. User privacy: the information of the user that signatured digitally is sent, preventing unwanted parties from learning personal information.

3. User identity: because Kerberos can be used to identify users, only users that authorized are can access to reliable services.

4. Secure authentication: requests digitally signatured for authentication are sent, which improves security.

Fig 2. Shows the system that proposed by moin a. Khorajiya and gardas naresh kumar.

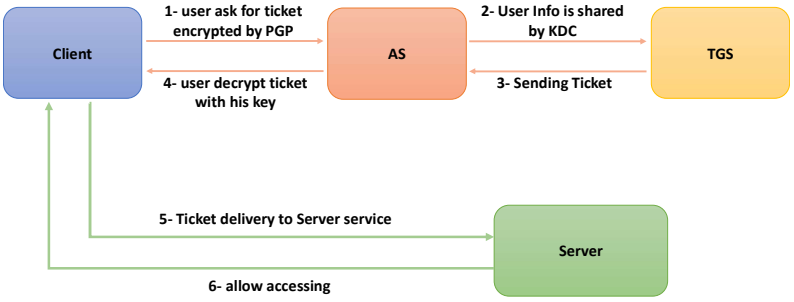


Fig. 2. The proposed system in [8]

[9] Linna, fan, et al., in 2016, developed an authentication in an anonymous way using Kerberos and HIBC protocols, named by KHIBC, to satisfy this criterion. It can satisfy the grid's need for authentication. Moreover, it can use an anonymous way to safeguard user identities. KHIBC able to satisfy requirements of anonymity, traceability, mutual authentication, and other factors through analysis.

It can be separating KHIBC to two trusted level domains. Kerberos is used in a first level trust domain. The HIBC is used in second level trust domain. The first trust level domain in several root pkgs made up in the second level.

Five steps can be used to represent the anonymous authentication mechanism's process.

(i) user a makes an anonymous authentication request to the ca along with the newly created NPK and seeks an anonymous certificate from the ca. The request is encoded using the public key of the ca.

(ii) ca confirms EEC of a and ensuring user “a” is one of the authorized users. The request is then decrypted using private key of ca to obtain subject of a. “epbca(npk)||hash′ epbca(npk)” is utilized to verify

if the encoded NPK has been altered or not. Finally, a receives the anonymous certificate that the ca has generated. If a is a fraudulent user, ca won't respond to it.

(iii) using an anonymous certificate, a requests resources from b.

(iv) after the certificate, which is anonymous, is received, a's NPK can be obtained by b, and then b verifies that a is at the other end of the communication. If so, the procedure advances to step v. Otherwise, the procedure will be ended.

(v) the anonymous certificate will be verified by b, and if ca is signed it and ca trusted directly by a, then anonymous authentication is passed through a and utilize b. If ca does not directly trust by b, it keeps vouching for ca's upper ca until it does. A's request will be rejected by b end the anonymous authentication procedure if it doesn't trust the anonymous certificate.

[10] The design of the implementation was presented in this poster/demo by araki, toshinori, et al in 2016 and shown in Fig. 3. The architecture demonstrated how to use low-cost servers to perform this high-

throughput three-party computing. The presentation showed that it is now feasible to compute secure multiparty Kerberos authentications in large enterprises.

The server of authentication in Kerberos protocol, that provides “Ticket Granting-Tickets” (TGT) with encryption method “AES”, has SMPC protocol implemented and is using it. Requirements of issuing tickets from the clients is taken by "secure KDC server". Instead of just using the AES encryption feature, this secure KDC server generates tickets with the aid of a new SMPC-proxy process within that server.

The design prevented the server from receiving the encrypted messages one at a time, even when it requests each encryption from the SMPC-proxy. Instead, it takes some time before it gets a batch of encrypted communications. As a result, the encryptions are carried out concurrently for several communications. They changed the Kerberos server such that it processes ticket generation in this manner.

For a brief while, the SMPC-proxy continues to receive requests for message encryption. They only take into account the counter mode of AES, therefore every request consists of a primary identification and counter pair. The proxy creates shares of these messages after receiving a series of them, and it delivers every part to the appropriate MPC.

When a request is getting from the SMPC-proxy, each of three MPC servers loads the given principal's part of the key first. The expanded round keys are the shared key. Then, MPC servers convert the "bit-slice" data structure in order to obtain extremely high throughput. The shared keys and counters are bit-sliced by the MPC server. Their SMPC protocol encrypts these bit-sliced counters using bit-sliced keys. The received bit-sliced ciphertexts are then returned to the SMPC-proxy after being bit-desliced into a series of ciphertexts. The KDC server now receives the results from SMPC-proxy, and it puts them together to create tickets.

مجلة العلوم والتطـبيقات الهـندسية

Journal of Science And Engineering Application

ISSN 2521-3911

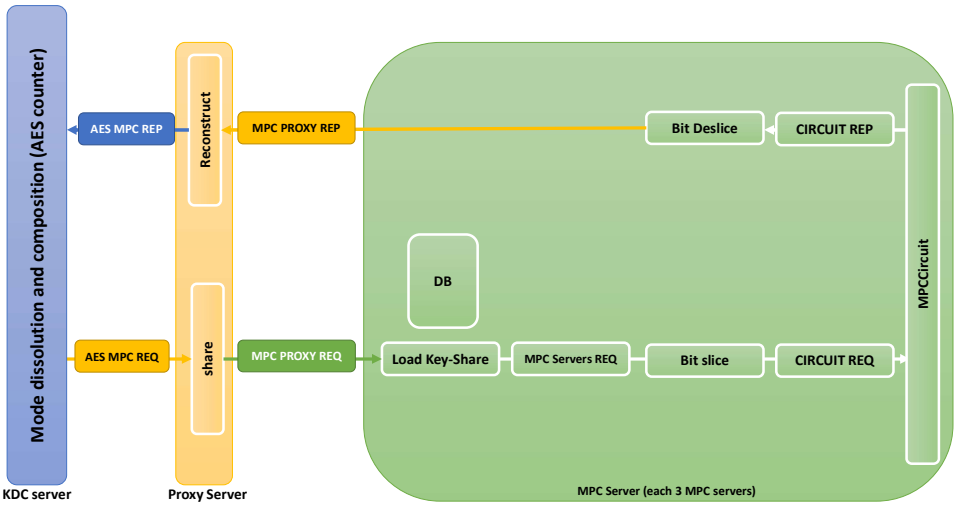


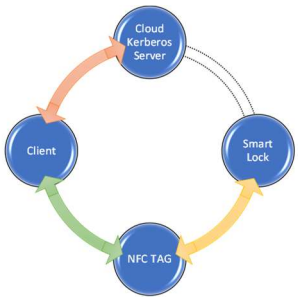
Fig. 3. The proposed system in [10]

[11] Chishti, mohd sameen, chung-ta king, and amit banerjee, in 2021, illustrated in their research the framework by considering a scenario: one user with an "NFC-enabled smartphone" attempts accessing to

an "NFC-enabled smart lock". Using fingerprint device, which is existing in utmost all current smart-phones, as one element of MFA. The mechanism that is proposed for MFA using NFC applies in following (Fig. 4 shown the proposed mechanism):

First, using a third-party authentication tool the smartphone of the user is authenticated. The third party calculate and send a ticket to access the smart lock. After matching, the device of fingerprint is used for authenticating the user. Authentication of two-factor is not only verifying the smartphone of user, but also verifies that the user is near the visible device or smart lock. They perform an experimental estimation of the structure via two NFC-enabled smartphones and passive tags of diverse dimensions. Additionally, it uses for authentication a third-party Kerberos for ticket generation and validation. The results showed only about 1 second was taken to connecting and generate a ticket from Kerberos, and six hundred milliseconds to open the locked door. Further, they evaluated resources that required to implement the algorithm, such as cpu utilization, power consumption, and smartphone internal temperature. They also conducted a security examination of the proposed scheme MFA.

(a)



(b)

Fig. 4. (a) NFC Mode (b) The proposed mechanism in [11]

[12] In 2022, anakath, a. S., et al. Presented a paper using Kerberos for m-healthcare security. A fingerprint-based Kerberos authentication protocol protected the agreement key that is between the source and the destination is suggested to ensure m-healthcare security. Wireless sensor devices are used to monitor patient conditions, and the server then receives the data. The Kerberos protocol aids in preventing pointless transmission of authenticated data across cloud networks. In the majority of the authentication fields, the biometric security method offers the highest level of protection. In the cloud network, the trust node is in charge of transporting data packets from the sender to the recipient. Security is ensured in trust nodes using the Kerberos protocol. The minutiae form fingerprint feature, that mentions to the fingerprint

images of the transmitter and the receiver, is used to provide secure connecting the neighborhood health-center with the healthcare server.

the problems of transmitted data via cloud network between HS and LHC are looked up. A key is calculated, to solve the problem, using the Kerberos protocol for authentication by merging it with a fingerprint pattern. Both HS and LHC compute and verify each other's keys. The system security is therefore attained by examining mutual-authentication for the HC with LHC through exchanging the keys with each other. The execution of the system showed, compared with other researches, that the computational cost of the 100-node proposed system is “54%, 58.6% and 63.5%” improvements and the connection cost of 100-node in the system was determined and obtained superior performance of “17%, 28%, and 32.5%”. The energy costs of the system are found efficiencies of “36.7%, 70%, and 78%”.

Fig 5. Shows the system that proposed by anakath, a. S., et al.

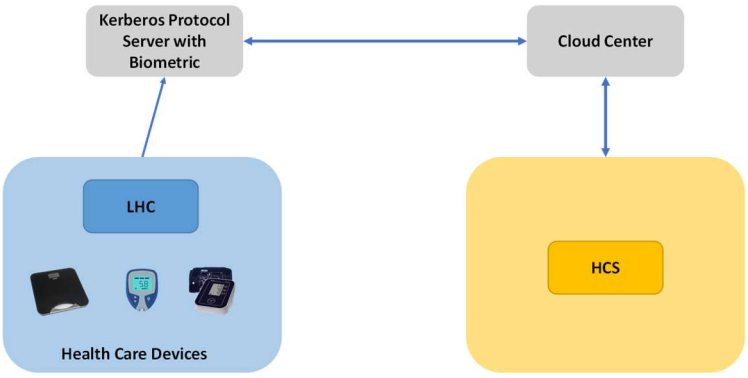


Fig. 5. Fingerprint agreement using enhanced Kerberos authentication protocol on m-health [12]

[13] In 2017, manunza l, marseglia s, romano sp. Described in their article a dynamically configurable system that can perform online fraud detection while still generating normal CDRS through appropriately combining actions made by OCS and related through only one call.

The telephone workers today depend on OCS (online billing systems) in order to monitor calls made by customers by generating events containing up-to-date data about network resource usage. Some data is naturally used for billing aims, but can also be used for fraud detection. To identify customers that make unauthorized use of the services provided. There two approaches also to cheating detection:

• “offline fraud detection”:

Which is the traditional method called “call detail records” (CDRS). A “CDR” is data related with a call. At the end of a call, the data is created. This data contains aggregate data about this call such as “caller, callee, call duration, etc.”. An offline method can detect fraud attempts only when the call is end.

• “online fraud detection”:

Which it is a new method based on real-time and analysis of call information is dynamic. This method detects fraud incidents by analyzing OCS events directly, rather than probing CDRS offline. Online detection is additional challenging in computational resources, but in other hand it allows the operator to detect fraud when the call is in progress, allowing appropriate action (e.g., fraud management) once a potential problem is detected. Component, etc.).

To achieving such of this area, Kerberos uses “Complex Event Processing (CEP)” methods in order to find subsets of actions that indicate the existence of potential fraud within the actual flow of actions generated by OCS. Kerberos is designed to be a powerful processing engine that also provides an easy-to-use web-based administration edge. It is planned and executed to the specifications of TISCALI, a well-known ITALIAN OTT “over the top” service provider that currently offers integrated cloud-based access (web, mobile,

مجلة العلوم والتطـبيقات الهـندسية

Journal of Science And Engineering Application

ISSN 2521-3911

PSTN, etc.), through the INDOONA platform, to IMS infrastructure (IP multimedia subsystem). In fact, TISCALI provide with general rules for the system, detailed information on basic fraud templates to consider, and an extensive dataset of his OCS events collected in the field.

[14] In 2017, Parmar et al. proposed a framework that allows MapReduce tasks to be executed directly on data stored in HDFS without requiring the complete file to be decrypted beforehand. This eliminates the need for an HDFS encryption zone. However, individual data blocks in MapReduce are decrypted during mapping, which is faster and more memory-efficient due to the smaller size of each block compared to the original file. It should be noted that HDFS encryption zones can still be used for data that will not be frequently accessed for extended periods.

In the proposed system, the Hadoop client splits data into blocks before encrypting each individual block and storing it in HDFS. This allows MapReduce to process each block independently, with decryption occurring during the map phase of the MapReduce job. The decryption key is provided through the job configuration, and the system is scalable and efficient. Using this approach with Kerberos and SASL for secure RPC communication and encrypting data in motion can provide three-dimensional security in Hadoop environments.

The Hadoop client is responsible for encrypting individual data blocks before they are stored in the data node for block-level encryption in Hadoop. MapReduce tasks can execute directly on the encrypted data in HDFS if the decryption key is known to the mapper class. When a file is read or copied to a local machine, the client reads each individual block and applies the decryption logic on the client side. It is important to note that encryption credentials are not stored in the Hadoop cluster but are securely managed on the client side using key management systems such as Kerberos. The choice of encryption algorithm is critical and must be memory efficient, fast, and secure. The Kuber framework allows clients to implement their own encryption algorithms based on their needs and is not dependent on a single technique. AES and Salsa20 are two reliable candidates. The current implementation of Kuber uses a variant of Salsa20 called chacha20, which improves Salsa20 in terms of diffusion per round and increasing resistance to cryptanalysis while preserving time per round.

Fig 6. show Providing Three-Dimensional Security in Hadoop Environment.

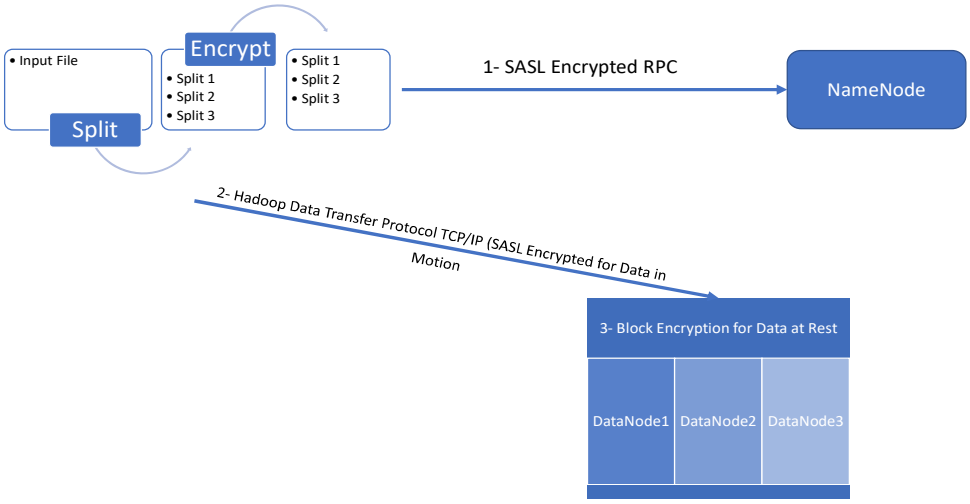


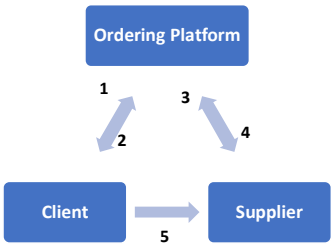
Fig. 6. 3-D Security in Hadoop Environment

[15] Li, hui, et al., in 2018, described the process of authentication in 3-stages as shown in Fig. 7.

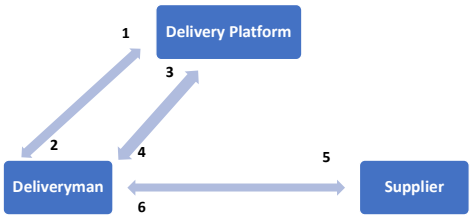
In the stage one, a client requests her TGT from the platform (m 1). More and more clients are opting to use their smartphones instead of computers. The platform replies with a TGT (m 2) noting that the encoded session key required for stage 2. This key can only be decoded by the client that knows his password. This client's password is never transmitted on the network. The 1st stage is used if the client first login into the system.

In the stage 2, the client shows the TGT to TGS (which is the m 3). TGS replies with SGT and the 2nd session key encoded with the 1st session key (m 4). The 2nd stage is essential when a customer confirms themselves with a new provider.

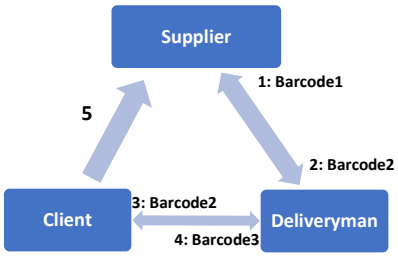
In the last stage, “SGT” is presented as a credential to a specific supplier (m 5). The supplier validates the authentication attempt by validating the signature of data that signed by the session key. The shipping business requires authentication offline “face-to-face” between both shipper and client. Table 5 compares electronic and in-person authentication implementations. In the real world, scanning a barcode image is preferable to starting a computer for sending and receiving messages. The suggested method is to stick the barcode on the dishes. Stamp and PND (for phone no.) Are included in the barcode authenticator.



(Stage 1)



(Stage 2)



(Stage 3)

Fig. 7. The Three-Stages of proposed system [15]

[16] Rao, k. And ram, n., (2016) in their article, applied the strategy of synchronizing time and freshness of information for exchanging key via internet which proposed by [17]. The proposed technique requires authentication or verification for multiple entities that connected with each other. A time synchronization scheme is included in the main and aggressive methods of internet key exchange for joint validation between some entities/elements. Each of the interested substances evaluates a piecemeal consistent capability, and if a match in qualities should exist in this approach, time modification is chosen as the neighborhood option.

In Kerberos v5 the time is used periodically such that Authenticator timestamp, ticket legitimacy or validity time, etc. The Kerberos attestation framework has difficulty maintaining normal behavior when it is difficult to keep the local good host time constant. Similarly, if the host's time changes, the host cannot use Kerberos protocols or rules. Also, whole authentication framework will break if serious problem with

the time of Kerberos server is existing. In addition, it is difficult to synchronize control in transport systems with different terminals. Tickets with long validity or longevity can easily withstand replay attacks. It is important to use secure time synchronization rules to tolerate time drift. The synchronizing of the time is essential for the ability of tickets issuing, inquiries and manage customers. Especially in the Kerberos framework, time synchronization is important before issuing ticket discount tickets and administrative tickets.

Authentication servers, service servers, ticket issuing servers, and Kerberos clients are considered NTP clients and may be kept in sync by discrepancies. Time can balance customers in two ways. Customers first follow the molds they had developed to see if they are suitable for customizing their own special timepieces in terms of durability, part consistency and balance evaluation. Otherwise, this developed form must be balanced by an external source such as a time server. In their protocol, pre-characterized resilience parameter between the local neighborhood time of client was take over from a Kerberos reference time.

[18] Saranya, s., et al., 2015, proposed a security model called the DPE technique for HDFS in big data, which used APACHE sentry and Kerberos for authorization and authentication to secure data from intruder prone operations. The paper addressed the security challenges faced by Hadoop and suggests an efficient technique based on the integration of sentry, Kerberos, and MONGODB. The DPE technique used a layered approach to provide security to HDFS.

Apache sentry is proposed as a solution that allows access control at the server, database, table, and view scopes, including various user controls such as select and insert. Sentry provides column-level security by creating a view of the allowed columns and restricting access to sensitive data. It also allows ease of administration through role-based authorization and grouping users into a single entity.

Sentry operates based on several key components: resources, privileges, roles, and groups. A resource is an object that controls access to data and can be a server, database, table, or URI. By default, sentry does not allow access to any resource unless specifically granted through a privilege. A privilege is a rule that specifies how a resource can be accessed. Roles are collections of privileges that can be combined to create a logical role in data processing. This allows multiple rules to be grouped together under a single template that can be assigned to an analyst all at once, simplifying permissions management. Groups are collections of users, and sentry's group mapping is extensible. By leveraging Hadoop’s group mapping, sentry can associate roles with groups, simplifying administration and avoiding the need to assign roles to individual users.

The Kerberos protocol is used for authentication in the project, where the client requests a ticket from the key distribution center (KDC), which creates a ticket-granting ticket (TGT) for the client, encrypts it using the client's password, and sends it to the client. The client decrypts the TGT to prove its identity and can obtain additional tickets for specific services before the TGT expires.

Fig 8. Shows the system that proposed by Saranya, s., et al.

مجلة العلوم والتطـبيقات الهـندسية

Journal of Science And Engineering Application

ISSN 2521-3911

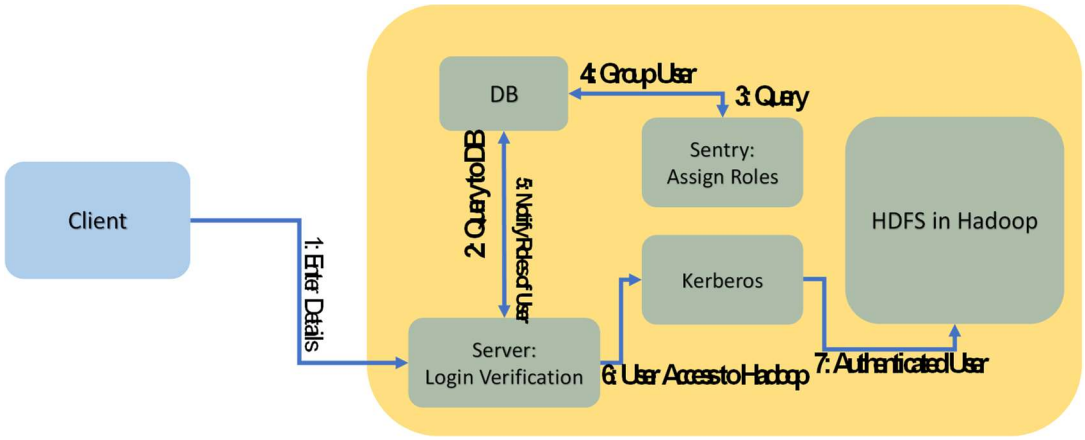


Fig. 8. The proposed system in [18]

[19] Pranay p. Gaikwad, et al., in 2015, described a smart home system created using internet of things technology. The system is presented in Fig. 9. The proposed system is designed to be cost-effective, efficient, and eco-friendly. It simplifies the process of home automation, enabling users to monitor and control home appliances remotely using the internet. The system is composed of embedded systems, GPRS modules, and rf modules. The paper also details the implementation of three-level Kerberos authentication to enhance security on the server-side, making it more secure than current smart home systems. Additionally, the paper provides information on the hardware and software design of the system.

The smart home architecture described in this paper includes a cloud server for storing user data, a GSM/GPRS modem to connect to the internet, a smart central controller, radio frequency (rf) modules, microcontrollers for switch modules, and household devices. The smart central controller serves as the main controller for the system and can be a microcontroller or another type of controller. The controller, along with the GSM/GPRS modem, acts as a link between household devices and the internet, enabling users to remotely monitor and control their devices. The smart central controller interfaces with the rf module and sends signals to rf module 1, which in turn sends signals to rf module 2 using the rf 433 MHZ frequency. A small microcontroller processes the signal from rf2 and sends it to the switch modules, which can switch the state of devices based on the given signal. Switch modules may include relays, actuators, wireless sensors, and other devices.

The proposed smart home system in this paper can be broken down into three components. The first part is an online server that utilizes Kerberos technology for secure authentication. The second component is the smart central controller, which comprises the GSM/GPRS modem, the main controller, and rf module 1. Lastly, the third part is composed of a small microcontroller that contains rf module 2 and switch modules.

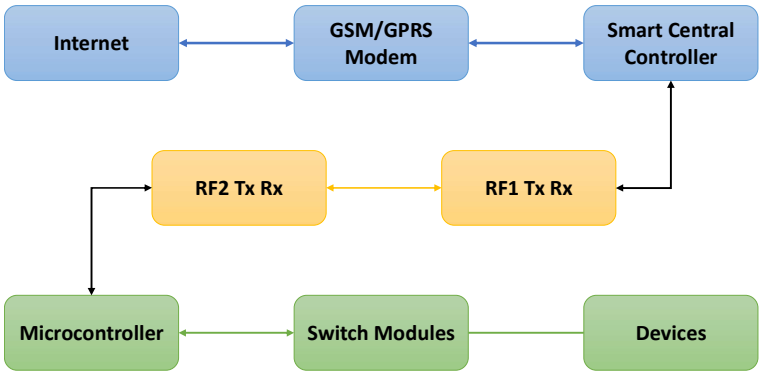


Fig. 9. The proposed smart home system in [19]

[20] In 2011, aruna kumari and dharmender singh kushwaha proposed a new model for secure authentication and authorization in distributed systems. The authors suggest a CTES (certificate trusted extended system) model that enhances the Kerberos authentication protocol by incorporating a certificate authority to verify user identities and permissions. The proposed model also includes a ticket granting service (tgs) for issuing tickets to users and a resource manager for managing access to resources. The authors present a comprehensive analysis of the CTES model and its security features, and compare it to other authentication models.

The process of client authentication includes four messages (message1, message2, message3, and message4). In the message 1, the client requests an as for authentication. The as then communicates with the message 2 superhost to verify the client and obtain the client's password, IP address, and trust level with message 3. The result of this process is the TGT assigned by the as to her message 4 client. Using two messages (message 5 and message 6) he gets the SGT and authorizes the client. Here the client makes a new request to the TGS with the TGT (received in message m4), an additional message known as message 4, and an additional message known as the authenticator. In their architecture, after receiving the SGT (using the message 6 message), the client sends the agent, not the server, a list of allowed services and messages message 7, message 8, message 9 and message 10. Clients access services from servers using message 11 and message 12 messages.

The researchers provided a performance evaluation of the model, demonstrating its efficiency and scalability. The paper concludes that the CTES model is a promising approach to secure authentication and authorization in distributed systems.

Fig 10. Shows the system that proposed by Saranya, s., et al.

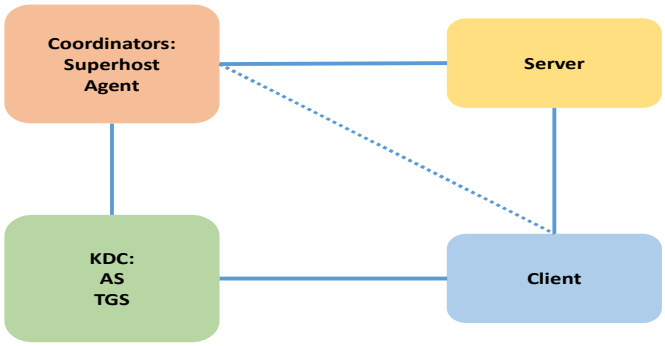


Fig. 10. The proposed system for secure authentication and authorization in [20]

[21] In 2009, a client-server model for a grid security architecture that adheres to the ogsa recommendations is described and created by moralis, athanasios, et al. This model offers improved near real-time services in grid applications by adopting symmetric cryptography during the actual operation. Because it offers superior interactive capabilities, this architecture is intended for grids with increased throughput and reaction time requirements (such as instrumentation grids).

The WSS Kerberos token profile was used in the security architecture's design and prototype because it performs better than the x.509 certificate token profile in terms of soap message authentication and integrity. The architecture includes common GSI features like single-sign-on authentication, delegation of credentials, message integrity (through digital signatures), and message secrecy (by encryption). It also has the option to add message authorization in response to user requests. Initial user authentication is dependent on their x.509 or VOMS proxy certificates for compatibility with GSI older systems. Furthermore, their design offers a flawless conversion between GSI and Kerberos credentials.

A major advantage of their architecture stems from the employed Kerberos token profile, which handles soap-level security more efficiently than the x509 profile, thus providing higher message throughput. Corresponding measurements have shown that the Kerberos token profile implementation consistently improves message throughput by up to 50% compared to the x.509 token profile when the server is under maximum CPU load. This difference is primarily due to the symmetric cryptography used by Kerberos implementations during messaging sessions, including the key derivation phase. It is this excellent performance in processing short messages at high speed that makes the proposed architecture attractive for applications with real-time constraints such as: typical of traditional grids

[22] Mahmoud khalifa, in 2017, proposed an enhanced version of the Kerberos authentication protocol that provides stronger security for distributed systems. The author argued that the traditional Kerberos protocol, which relies on a single server to provide authentication and authorization, is vulnerable to attacks that can compromise the security of the system.

To address these vulnerabilities, the author proposed a two-phase security model that adds an additional layer of security to the authentication process. In the first phase, the user authenticates with the Kerberos server using their username and password. In the second phase, the user is required to provide additional authentication factors, such as a biometric or a smart card, to further verify their identity.

The author presented a detailed analysis of their proposed protocol, including a security analysis and performance evaluation. He also provided a comparison of their protocol with other existing protocols, such as the traditional Kerberos protocol and the multi-factor authentication protocol. The results of his analysis demonstrate that their protocol provides stronger security and better performance than the existing protocols.

Overall, the paper presented a novel approach to enhancing the security of distributed systems through the use of a two-phase authentication protocol. The proposed protocol offered improved security and performance compared to existing protocols, and has the potential to be a valuable addition to the field of distributed systems security.

V. Main Attacks on Kerberos Protocol

There are many attacks on the Kerberos protocol. Some of the attacks, that have been mentioned in the researches that interested with this protocol, will be presented here:

مجلة العلوم والتطـبيقات الهـندسية

Journal of Science And Engineering Application

ISSN 2521-3911

1. Brute-force attacks: In a brute-force attack, an attacker tries each password combination until the right one is discovered in an effort to guess the user's password. Because Kerberos employs a symmetric key mechanism, the client and server share the same key. The key can be used by an intruder to pretend to be the client and access network resources without authorization [23, 24].
2. Replay attacks: To access the network, an attacker intercepts a legitimate authentication ticket and replays it. By using timestamps or nonces to make sure that each authentication request is distinct, this type of attack can be stopped [24].
3. Dictionary attacks: A dictionary attack is similar to a brute-force attack, except that the attacker employs a pre-built list of popular passwords rather than attempting every possible character combination. Using strong passwords and enforcing password rules can stop this kind of attack [23, 24, 25].
4. Man-in-the-middle attacks: A man-in-the-middle attack involves the attacker intercepting the communication between the client and server and then impersonating either one or both of them to obtain access to the network. Mutual authentication and encryption are effective ways to thwart this kind of assault [24].
5. Password guessing attacks: Using social engineering or other methods, an intruder attempts to predict a user's password in a password guessing attack. Users can be made aware of the value of using secure passwords and multi-factor verification to avoid this type of attack [23, 26].
6. Kerberos delegation attacks: Attackers can take advantage of Kerberos' sharing features to access resources beyond those for which they have been initially granted permission. Limiting delegation rights and/or utilizing constrained delegation methods can reduce the impact of this assault [27].
7. Golden Ticket attacks: Attackers can take complete control of a domain, including the power to add and remove user accounts, by using a falsified Kerberos ticket. Using strong password guidelines and keeping an eye out for strange activities can help to mitigate this attack [28, 29].

VI. Conclusion and Recommendations

This paper provides an overview of the Kerberos protocol, its history, steps, and security features, as well as the most important research and attacks on the protocol. It is hoped that this paper will provide valuable insights to researchers and practitioners in the field of network security and authentication.

Here are some recommended changes to the Kerberos protocol that have been proposed by researchers:

- Use stronger encryption algorithms: The current version of Kerberos uses DES encryption, which is considered weak. Researchers have proposed using stronger encryption algorithms, such as AES, to improve the security of the protocol.
- Improve the key distribution process: The key distribution process in Kerberos can be improved by using a trusted third-party to distribute keys or by implementing a hierarchical key distribution model.
- Enhance mutual authentication: Mutual authentication can be enhanced by using public-key cryptography or by implementing additional security features, such as time-based authentication.
- Implement additional security features: Additional security features, such as integrity checks, can be implemented to prevent attacks such as replay attacks and man-in-the-middle attacks.
- Improve the scalability of the protocol: The scalability of Kerberos can be improved by using load balancing and distributed key management techniques.
- Enhance usability: Usability can be improved by implementing user-friendly interfaces and simplifying the key distribution process.
- Implement continuous monitoring: Continuous monitoring can be implemented to detect and prevent attacks in real-time.

مجلة العلوم والتطـبيقات الهـندسية

Journal of Science And Engineering Application

ISSN 2521-3911

References

[1] J. Rosenberg *et al.*, "SIP: session initiation protocol," 2070-1721, 2002.
[2] J. Kohl and C. Neuman, "The Kerberos network authentication service (V5)," 2070-1721, 1993.
[3] B. C. Neuman and T. Ts'o, "Kerberos: An authentication service for computer networks," *IEEE Communications magazine,* vol. 32, no. 9, pp. 33-38, 1994.
[4] S. William, *Cryptography and Network Security - Principles and Practice, 7th Edition*. Pearson Education India.
[5] R. Perlman, C. Kaufman, and M. Speciner, *Network security: private communication in a public world*. Pearson Education India, 2016.
[6] D. Gollmann, "Computer security," *Wiley Interdisciplinary Reviews: Computational Statistics,* vol. 2, no. 5, pp. 544-554, 2010.
[7] S. Garfinkel, "Practical UNIX & Internet Security 2nd ed.; O'Reilley & Associates," ed: Inc, 1996.
[8] M. A. Khorajiya and G. N. Kumar, "A Security based Architecture using Kerberos and PGP," in *Proceedings of the International Conference on Advances in Information Communication Technology & Computing*, 2016, pp. 1-4.
[9] F. Linna *et al.*, "An anonymous authentication mechanism based on Kerberos and HIBC," in *Proceedings of the 10th International Conference on Education Technology and Computers*, 2018, pp. 392-396.
[10] T. Araki, A. Barak, J. Furukawa, Y. Lindell, A. Nof, and K. Ohara, "High-throughput secure three-party computation of kerberos ticket generation," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016, pp. 1841-1843.
[11] M. S. Chishti, C.-T. King, and A. Banerjee, "Exploring half-duplex communication of NFC read/write mode for secure multi-factor authentication," *IEEE Access,* vol. 9, pp. 6344-6357, 2021.
[12] A. Anakath, S. Ambika, S. Rajakumar, R. Kannadasan, and K. Kumar, "Fingerprint Agreement Using Enhanced Kerberos Authentication Protocol on M-Health," *COMPUTER SYSTEMS SCIENCE AND ENGINEERING,* vol. 43, no. 2, pp. 833-847, 2022.
[13] L. Manunza, S. Marseglia, and S. P. Romano, "Kerberos: A real-time fraud detection system for IMS-enabled VoIP networks," *Journal of Network and Computer Applications,* vol. 80, pp. 22-34, 2017.
[14] R. R. Parmar, S. Roy, D. Bhattacharyya, S. K. Bandyopadhyay, and T.-H. Kim, "Large-scale encryption in the Hadoop environment: Challenges and solutions," *IEEE Access,* vol. 5, pp. 7156-7163, 2017.
[15] H. Li, Y. Niu, J. Yi, and H. Li, "Securing offline delivery services by using Kerberos authentication," *Ieee Access,* vol. 6, pp. 40735-40746, 2018.
[16] K. Rao and N. Ram, "Application of time synchronization process to Kerberos," *Procedia Computer Science,* vol. 85, pp. 249-254, 2016.
[17] "Kerberos The Network Authentication Protocol " https://web.mit.edu/kerberos/ (accessed.
[18] S. Saranya, M. Sarumathi, B. Swathi, P. V. Paul, S. S. Kumar, and T. Vengattaraman, "Dynamic preclusion of encroachment in hadoop distributed file system," *Procedia Computer Science,* vol. 50, pp. 531-536, 2015.
[19] P. P. Gaikwad, J. P. Gabhane, and S. S. Golait, "3-level secure Kerberos authentication for smart home systems using IoT," in *2015 1st International conference on next generation computing technologies (NGCT)*, 2015: IEEE, pp. 262-268.
[20] A. Kumari and D. Singh Kushwaha, "Kerberos style authentication and authorization through CTES model for distributed systems," in *Computer Networks and Intelligent Computing: 5th International*

Conference on Information Processing, ICIP 2011, Bangalore, India, August 5-7, 2011. Proceedings, 2011: Springer, pp. 457-462.

[21] A. Moralis, V. Pouli, S. Papavassiliou, and V. Maglaris, "A Kerberos security architecture for web services based instrumentation grids," *Future Generation Computer Systems,* vol. 25, no. 7, pp. 804-818, 2009.

[22] M. Khalifa, "Enhanced Kerberos Authentication For Distributed Environment Using Two Phases Security," *An international journal of advanced computer technology,* vol. 6, no. 4, 2017.

[23] S. M. Dhiman Saha, and Koushik Majumder "A Comprehensive Review of Kerberos Security Threats and Countermeasures," *Journal of Network and Computer Applications,* vol. 147, 2019.

[24] A. A. a. X. Zhang, "Kerberos Authentication Protocol Vulnerabilities and Security Measures," *IEEE Access,* vol. 8, 2020.

[25] M. Weir, S. Aggarwal, B. De Medeiros, and B. Glodek, "Password cracking using probabilistic context-free grammars," in *2009 30th IEEE symposium on security and privacy*, 2009: IEEE, pp. 391-405.

[26] P. K. N. H. Vijendrasinh P. Thakur, "Hash Based Dynamic Password Authentication Mechanism For Kerberos Environment," *INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT),* vol. 02, no. 07, 2013, doi: 10.17577/IJERTV2IS70115.

[27] A. F. Z. Danilo Costa, and Altair Olivo Santin, "Exploiting Kerberos Vulnerabilities," *IEEE Communications Surveys & Tutorials,* vol. 23, no. 02, 2021.

[28] M. Soria-Machado, D. Abolins, C. Boldea, and K. Socha, "Kerberos golden ticket protection," *Mitigating Pass-the-Ticket on Active Directory, CERT-EU Security Whitepaper,* vol. 7, p. 2016, 2014.

[29] S. Pocarovsky, M. Koppl, M. Orgon, and A. Bohacik, "Kerberos Golden Ticket Attack," in *Data Science and Algorithms in Systems: Proceedings of 6th Computational Methods in Systems and Software 2022, Vol. 2*: Springer, 2023, pp. 677-688.

مجلة العلوم والتطـبيقات الهـندسية

Journal of Science And Engineering Application

ISSN 2521-3911

مجلة العلوم والتطـــبيقات الهـندسية

Journal of Science And Engineering Application

ISSN 2521-3911