BAIT2023

Introduction to Internet Security

Assignment

February 2025

Semester 2025/26

TUNKU ABDUL RAHMAN UNIVERSITY OF MANAGEMENT & TECHNOLOGY

FACULTY OF COMPUTING AND INFORMATION TECHNOLOGY
RSD Y2S3
G5
Assignment
BAIT2023 Introduction to Internet Security
February 2025 Semester, AY 2025/2026

Name (Block Capital)	**Registration No.**	**Signature**	**Marks** (For Lecturer / Tutor use) **------- / 100%**
1. Liew Zi Li	2409077		
2. Low Kim Hong	2409085		
3. Ng Jhun Hou	2409091		
4. Boo Kai Jie	2409036		
5. Lim Jun Wei	2409078		
6. Ong Yi Xin	2409097		
7. Chia Ming Yi	2409040		

Lecturer/Tutor's Name: _____Dr Ng Yen Phing_____
Date of Submission: _____20 April 2025_____

Assignment Assessment Rubrics (Group: Programme: RSDY2S3 Tutorial: 5 5)

Assignment Final Report Assessment Criteria

The assessment of this final assignment report is based on the following criteria:

TASK 1 (GROUP) (CLO1)

Criteria	Overall Weight (%)	Excellent (5)	Good (4)	Average (3)	Below Average (2)	Poor (1)	Very Poor (0)	Weighted Marks for Group (%)	Marks
Part A Focus	10	Sharp, distinct controlling point made about a single topic with evident awareness of task	Clear and focused main point addressing the topic and demonstrating an understanding of the task.	Adequate main point established with some awareness of the task.	The limited or unclear main point with limited awareness of the task.	Absence of a discernible main point with little to no awareness of the task.	Significant lack of understanding or effort resulting in incomplete or irrelevant submission.	* 2	Excellent
Part B Content	20	Substantial, specific, and/or illustrative content demonstrating strong development and sophisticated ideas	Significant and relevant content demonstrating satisfactory development and coherent ideas.	Sufficient content with basic development and generally coherent ideas.	Limited or insufficient content with unclear or weakly developed ideas.	Inadequate or irrelevant content with little coherence or development.	Significant lack of understanding or effort resulting in incomplete or irrelevant submission.	* 4	Excellent
Part C Organization	5	Sophisticated arrangement of content with evident and/or subtle transitions	Functional arrangement of content that sustains a logical order with some evidence of transitions	Confused or inconsistent arrangement of content with or without attempts at transition	Limited or unclear arrangement of content with weak transitions.	Minimal control of content arrangement.	Significant lack of understanding or effort resulting in incomplete or irrelevant submission.	* 1	Excellent

Part D Style	5	Precise, illustrative use of a variety of words and sentence structure to create consistent writer's voice and tone appropriate to audience	Generic use of a variety of word and sentence structure that may or may not create writer's voice and tone appropriate to audience	Adequate use of words and sentence structure with a generally suitable writer's voice and tone.	Limited or repetitive use of words and sentence structure with an inconsistent writer's voice and tone.	Inaccurate or inappropriate use of words and sentence structure with an unclear writer's voice and tone.	Significant lack of understanding or effort resulting in incomplete or irrelevant submission.	___ * 1	Excellent
Total:	40	**Additional Feedback (if any):**						**Total Marks (/40):**	

TASK 2 (GROUP) (CLO3)

Criteria	**Overall Weight (%)**	**Excellent (5)**	**Good (4)**	**Average (3)**	**Below Average (2)**	**Poor (1)**	**Very Poor (0)**	**Weighted Marks for Group (%)**	**Marks**
Part A Background	10	Very clear description about the background of the topic selected.	Clear and concise description providing relevant background information about the selected topic.	Adequate description providing some relevant background information about the selected topic.	Limited or unclear description with insufficient background information about the selected topic..	Inadequate or irrelevant description with no or incorrect background information about the selected topic.	No description or complete absence of any background information about the selected topic.	* 2	Excellent
Part B Discussion on security related technology	20	Excellent understanding on the security related technology. Great amount of research on the technical issues.	Solid understanding of security-related technology with a satisfactory level of research on technical issues..	Basic understanding of security-related technology with limited research on technical issues.	Limited understanding of security-related technology with minimal research on technical issues.	Inadequate understanding of security-related technology with no or inaccurate research on technical issues.	No understanding of security-related technology with no research on technical issues.	* 4	Excellent
Part C Discussion on impacts	10	Sophisticated arrangement of content with evident and/or subtle transitions	Functional arrangement of content that sustains a logical order with some evidence of	Confused or inconsistent arrangement of content with or without attempts at transition	Limited or unclear arrangement of content with weak transitions.	Minimal control of content arrangement.	Significant lack of understanding or effort resulting in incomplete or	* 2	Excellent

			transitions				irrelevant submission.		
Life Long Learning	5	Able to provide very clear and reasonable evaluation with very details explanations on the technology	Provides a clear and reasonable evaluation with detailed explanations on the technology.	Offers an evaluation with some clarity and explanations on the technology.	Provides a limited or unclear evaluation with insufficient explanations on the technology.	Offers an inadequate or irrelevant evaluation with no or weak explanations on the technology.	No evaluation provided or a complete absence of explanations on the technology.	___* 1	Excellent
Originality	5	Good evidence of proving appropriate & relevance of selfresearch & information originality.	Presents compelling evidence demonstrating the appropriateness, relevance, and originality of self-research and information.	Offers satisfactory evidence demonstrating some appropriateness, relevance, and originality of self-research and information.	Provides limited or insufficient evidence of the appropriateness, relevance, and originality of self-research and information.	Presents inadequate or irrelevant evidence with no or weak connection to the appropriateness, relevance, and originality of self-research and information.	No evidence was provided or complete absence of any connection to the appropriateness, relevance, and originality of self-research and information.	____ *1	Excellent
Total:	40	**Additional Feedback (if any):**						**Total Marks (/40):**	

ORAL PRESENTATION SKILLS (INDIVIDUAL) (CLO1)

Criteria	Overall Weight (%)	Excellent (5)	Good (4)	Average (3)	Below Average (2)	Poor (1)	Very Poor (0)	Weighted Marks for Individual (%)	Marks
		Communicates ideas with enthusiasm, proper voice projection, appropriate language, and clear presentation delivery. Excellent visual aids. .	Communicates ideas with enthusiasm, reasonable voice projection, mostly appropriate language, and a generally clear presentation delivery. Effective visual aids are used.	Communicates ideas with some enthusiasm, inconsistent voice projection, language usage that may require improvement, and an overall satisfactory presentation delivery. Adequate use of visual aids..	Communicates ideas with limited enthusiasm, poor voice projection, language usage that needs significant improvement, and a presentation delivery that lacks clarity. Limited or ineffective use of visual aids.	Struggles to communicate ideas with little to no enthusiasm, very poor voice projection, inappropriate language usage, and a presentation delivery that is unclear or confusing. No or ineffective use of visual aids..	Unable to effectively communicate ideas, devoid of enthusiasm, extremely poor voice projection, inappropriate and incomprehensible language usage, and a presentation delivery that is entirely ineffective. No visual aids are used.		
Liew Zi Li	10							___* 2	5 **Excellent**
Low Kim Hong	10							___* 2	5 **Excellent**
Ng Jhun Hou	10							___* 2	5 **Excellent**
Boo Kai Jie	10							___* 2	5 **Excellent**
Lim Jun Wei	10							___* 2	5 **Excellent**

Ong Yi Xin	10							___* 2	5 **Excellent**
Chia Ming Yi	10							___* 2	5 **Excellent**
Additional Feedback (if any):									

Faculty of Computing and Information Technology
Plagiarism Statement

Read, complete, and sign this statement to be submitted with the written report.
We confirm that the submitted work are all our own work and are in our own words.

Name (Block Capitals)	Registration No.	Signature
1. …Liew Zi Li…	…24WMR09077…	……
2. …Ng Jhun Hou…	…24WMR09091…	……
3. …Low Kim Hong…	…24WMR09085…	……
4. …Boo Kai Jie…	…24WMR09036…	……
5. …Lim Jun Wei…	…24WMR09078…	……
6. …Ong Yi Xin…	…24WMR09097…	……
7. …Chia Ming Yi…	…24WMR09040…….	……

Tutorial Group : ……RSD Y2S3 G5…….
Date : ……20 April 2025……..

BAIT2023 Introduction to Internet Security

Individual Tasks Allocation

Indicate (√) in member name column if he/she have involved in that task.

Tasks		Liew Zi Li	Low Kim Hong	Ng Jhun Hou	Boo Kai Jie	Lim Jun Wei	Ong Yi Xin	Chia Ming Yi
1.	Task 1: Introduction					✅		
2.	Task 1: Fundamentals of Cryptography						✅	
3.	Task 1: Cryptography Types					✅	✅	✅
4.	Task 1: Major Cryptographic Protocols					✅	✅	✅
5.	Task 1: Challenges and Future Trends							✅
6.	Task 1: (Optional) Simple Code Example Implementation					✅		
7.	Task 1: Conclusion						✅	
8.	Task 1: References					✅	✅	✅
9.	Task 2: Background	✅	✅	✅	✅			
10.	Task 2: Detailed discussion on how the	✅	✅	✅	✅			

	security related technology works.							
11.	**Task 2: Detailed discussion on how the security related technology works.**	✅	✅	✅	✅			
12.	**Task 2: Discussion on the impact.**	✅	✅	✅	✅			
13.	**Task 2:Evaluate the effectiveness of current approaches and propose improvements**	✅	✅	✅	✅			

Contents

Task 1

Introduction

Approximately exabytes (EB) of data is transferred through the Internet every day and the total digital data stored on the Internet has exceeded 120 zettabytes (ZB) by 2023. (Taylor, 2024) Meanwhile, each of the data belongs to a specific user and only the owner or permitted user should be able to access it for security purposes. At this point, cryptography is the crucial key for realizing the protection of those data. Cryptography is a technique of securing information and communications through the use of codes so that only persons for whom the information is intended can understand and process it. (Kothari, 2019) It will explore and utilize different types of mathematical algorithms to convert the messages data into human unreadable messages and can only be converted back into readable messages using the correct key and algorithms.

Importance

Cryptography always obeys and applies these 5 important principles: encryption, data integrity, authentication, non-repudiation and key management. Thus, all the unauthorized parties would not be able to read the sensitive information easily such as users' account password, financial transactions and classified information. Not only that, cases of personal data exposure can be minimized if using the extreme high complexity of combination of algorithms to protect the data. So, the users can transmit or upload the sensitive data with confidence.

At the same time, it will also use techniques to promise the integrity of data and maximize the trustworthiness of the data. Any attempt of tampering with data changes will be detected and discarded immediately. Therefore, users can always believe the messages data received on the Internet without hesitation.

Moreover, every user action during message sending and transaction will be recorded and acts as a proof using the cryptography technique. The main purpose is to ensure accountability by preventing someone from denying their actions via providing the proofs recorded. This can effectively protect the security for each transaction and data transfer as any action made to the data can be quickly tracked without possibility of action denial.

Application of Cryptography in Modern Digital Communication

Secure Communication Protocols

Cryptography may seem far away, but it actually surrounds every person, software and even hardware all the time in daily lives. Everyday, users will browse and access a lot of web pages via web browsers. In order to secure the data that is viewed, created, modified, updated by users, cryptography will use SSL/TLS to encrypt and decrypt the data during transmission between web browsers and servers. (Tanushi Bandara, 2023) Meanwhile, it will also use Transport Layer Security (TLS) to ensure the end-to-end security for data sent over networks using symmetric and asymmetric encryption. (Tanushi Bandara, 2023)

Secure Data Storage

When dealing with the cloud storage services such as Dropbox, Google Drive and Microsoft Onedrive, cryptography will protect every data uploaded into the cloud by the users. It will use complex encryption algorithms such as AES to transform plaintext into ciphertext. (Schneider, 2024) This feature is aiming for avoiding unauthorized access by intended individuals. So, they will not be able to decrypt the data in drive or even read and alter the data inside the cloud storage.

E-commerce and Financial Transactions

Nowadays users usually purchase or order any wanted product on the e-commerce platform on the Internet using mobile devices. During the payment session, they will need to enter the credit card number, bank account number and passwords and other sensitive information. Without the presence of cryptography, all those transaction data will be easily disrupted or stolen by unauthorized parties. Thus, cryptography is very important for making those transactions data “unaccessible”, “invisible” and unreadable by other parties to secure the process of online transactions. (Vanderwall, 2018)

Secure Emails

Most of the users or organizations will use email as a communication or promotion channel to contact with other users. However, those email messages will easily get disrupted or altered without cryptography. Thus, the mail services will apply different file encryption to protect email communications, including sensitive conversations and attachments. (Wickramasinghe, 2023) For example, they use S/MIME encryption and Microsoft 365 message encryption to secure Microsoft 365 email messages. (Wickramasinghe, 2023) Eventually, the users can successfully receive the unaltered messages without being read by other unauthorized parties.

Fundamentals of Cryptography

Cryptography is a **computer science technique** that ensures the **security** of user communications by encrypting or converting user information into a format that is unreadable by humans, so that only authorized users can decrypt and read user information. Cryptography protects the privacy, confidentiality, integrity, and authority of user information. Its encryption and decryption processes play an extremely important role, otherwise the security of user information is threatened. Modern cryptography is based on mathematical principles and computational techniques that improve data security in various applications.

Key Concepts

The two most critical concepts of cryptography are **encryption and decryption**. Encryption refers to the use of encryption algorithms and keys to **convert human-readable text** or sentences directly **into ciphertext**, both data that cannot be read and understood by humans. For example, using the Advanced Encryption Standard (AES), the message "HELLO" might be encrypted into a seemingly random string such as "f9a8b7c6d5". Without the correct key, deciphering this encrypted data is nearly impossible.

Since there is encryption to keep unauthorized users from learning the contents of a digital message, there also needs to be decryption to allow users who are authenticated by their privileges to read the contents of a digital message. There are two main types of encryption techniques, **symmetric encryption and asymmetric encryption**. Symmetric encryption uses one key for both encryption and decryption, making it fast but requiring secure key distribution. Asymmetric encryption uses a public key for encryption and a private key for decryption, enhancing security but increasing computational cost.

Decryption is the **reverse process of encryption**, which uses the encryption key and algorithm to convert the **ciphertext back into plaintext**. Decryption must use the same key and algorithm used for encryption in order to convert the original message correctly, otherwise the converted message will become garbled or a completely unintelligible plaintext. This ensures that only authorized personnel with the correct key can read the data.

Goals

Cryptography has several key objectives to ensure the security and reliability of digital communications and data protection. The three most important goals include confidentiality, integrity, authentication.

Confidentiality is the central goal of cryptography, which ensures that only **authorized parties** have access to sensitive information. It prevents unauthorized access, interception, or disclosure of private data.Modern encryption methods such as AES (Advanced Encryption Standard) and RSA (Rivest-Shamir-Adleman) are widely used to protect sensitive data.

For example, cloud service providers such as Google Drive, Dropbox, and OneDrive use encryption to protect stored data. Even if an unauthorized person accesses a cloud server, encrypted files remain unreadable without the correct decryption key. This ensures that sensitive

information remains confidential even in the event of a data breach, highlighting the critical role of encryption in protecting digital privacy.

Authentication ensures that the **sender and receiver** in a communication process are **genuine and trustworthy**. It verifies the identity of a person, device or system before allowing access to sensitive information. Without authentication, an attacker could impersonate a legitimate user and gain unauthorized access to data and systems. Cryptographic authentication mechanisms such as digital signatures, certificates, and multi-factor authentication (MFA) play an important role in preventing impersonation attacks and securing communications.

For example, email services such as Gmail and Outlook use authentication protocols such as DKIM (DomainKeys Identified Mail) and SPF (Sender Policy Framework) to verify the authenticity of the sender and prevent email spoofing. These protocols ensure that email is sent from legitimate sources, thereby reducing the risk of phishing attacks and maintaining the integrity of email communications. In addition, cryptographic authentication methods such as Public Key Infrastructure (PKI) and certificate-based authentication are used in a variety of systems to securely verify identities and prevent unauthorized access.

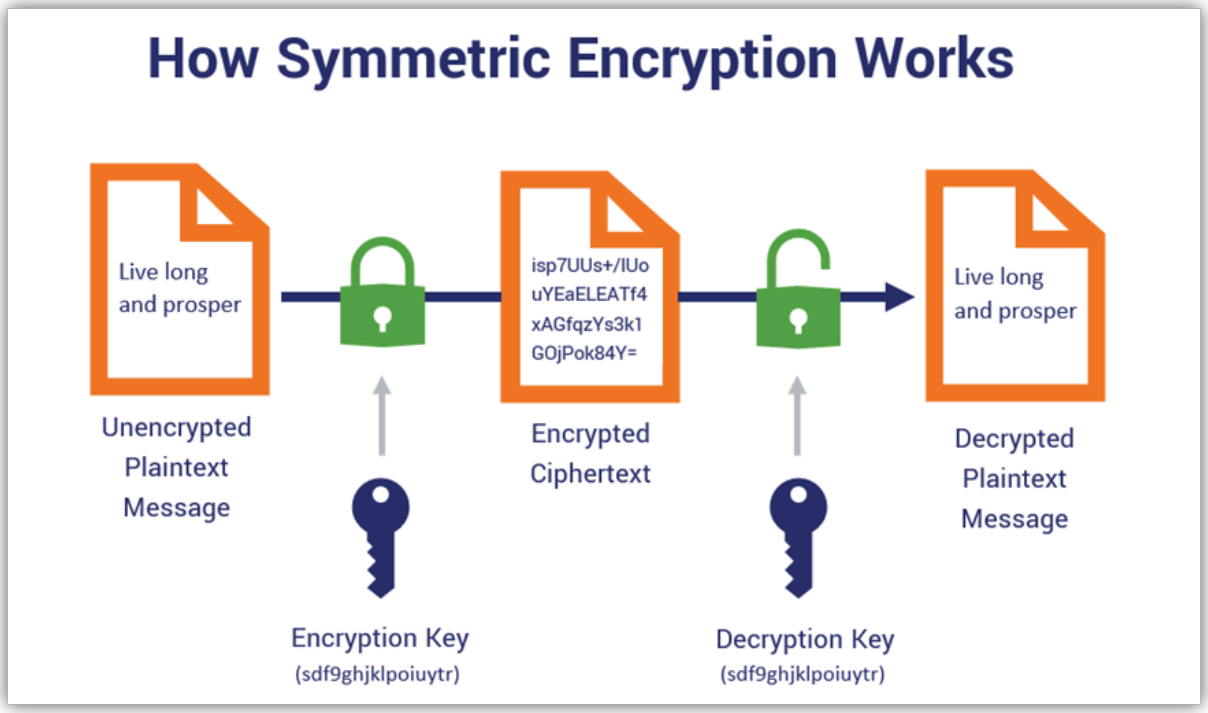
Integrity in cryptography ensures that **data remains unchanged and trustworthy** during storage, processing and transmission. It prevents accidental or malicious unauthorized modifications and allows detection of tampering. Cryptographic integrity mechanisms use **hash functions, digital signatures, and Message Authentication Codes (MACs)** to verify the consistency of data. Hash functions generate unique identifiers for data so that even the smallest changes can be detected. Digital signatures combine hashing and asymmetric encryption to verify authenticity and integrity, while MACs use keys to prevent unauthorized changes. By guaranteeing the consistency and authenticity of data, cryptographic integrity plays a vital role in ensuring that communications, transactions, and stored information cannot be tampered with or corrupted.

For example, blockchain technology uses cryptographic hashing algorithms to create a secure such as SHA-256 in Bitcoin, tamper-proof ledger of transactions. If an attacker attempts to modify a past transaction, the hash value of that block changes, disabling the entire chain. This ensures the immutability and trustworthiness of transactions. Similarly, cryptocurrencies such as Bitcoin and Ether rely on the integrity of the blockchain to prevent duplicate spending and fraudulent transactions, demonstrating the critical role of cryptographic integrity in maintaining the trustworthiness of digital systems.

Cryptography Types

Symmetric Cryptography

Symmetric cryptography is a type of encryption that uses a **single key for both** encryption and decryption processes. This method is often referred to as **"secret key" or "private key"** encryption because the same key is used by both the sender and the recipient to convert plaintext into ciphertext and vice versa. Symmetric encryption is widely used due to its **efficiency and speed**, making it ideal for encrypting **large volumes of data**, such as stored files or internal communications within a closed system. However, it requires secure key distribution and management, as both parties must be able to access the same key without it being compromised. Highly secure and efficient symmetric encryption algorithms such as AES and ChaCha20 are now widely used in the field of securing network information. Symmetric encryption involves two main types of ciphers, which are **block ciphers and stream ciphers**.



Algorithm

AES (Advanced Encryption Standard) is one of the algorithms of symmetric block cipher that encrypts data in f**ixed-size blocks** of **128 bits using a 128, 192, or 256 bit key**. It operates over a substitution permutation network (SPN) and performs multiple rounds of encryption on the way, including **steps** such as **SubBytes, ShiftRows, MixColumns, and AddRoundKey Nordlayer**.

The SubBytes operation refers to replacing each byte in the input block with the corresponding byte in an S-box, which is a fixed table of substitutions, thus providing nonlinearity to the cipher. Subsequently, the ShiftRows operation periodically shifts the bytes in each row of the state array to the left, with different offsets for different rows, thus ensuring that each column of the output state consists of bytes in each column of the input state. Next comes the MixColumns step, which transforms each column using a matrix multiplication operation that provides diffusion and enhances AES security by affecting all four bytes in a column at the same time. Finally, AddRoundKey uses bitwise XOR to combine the resulting data block with the round key in the key table, ensuring that each output data block byte depends on the corresponding byte of the key.

Depending on the key size, these steps are repeated for multiple rounds to ensure high security and attack resistance. For example, in AES-256, the encryption process consists of 14 rounds, with each round applying these transformations to ensure a high degree of security and diffusion.AES strikes a good balance between security and efficiency, and is therefore widely used as an encryption standard for large amounts of data.

Another algorithm of symmetric cryptography which is called **ChaCha20** is a fast, secure stream cipher designed by Daniel J. Bernstein. According to NordVPN, it works by generating a **pseudo-random keystream** with a **256-bit key, a 32-bit counter, and a 96-bit nonce**. The initialization process consists of setting the **initial state** to a **4x4 matrix of 32-bit words**, with the first row containing a constant string, the second and third rows populated with the 256-bit key, and the fourth row containing the counter and nonce. For example, to encrypt a message like ”HELLO” , ChaCha20 generates a keystream that is the same length as the message or a multiple thereof and then XORs it with the message to generate the ciphertext. This process is both fast and secure, making ChaCha20 suitable for high-performance applications.

Google's QUIC (Quick UDP Internet Connection) protocol is a good example of a ChaCha20 application that improves network performance by providing a faster, more secure encrypted connection than traditional TCP TLS. Typically, ChaCha20 is used with Poly1305 for authentication, forming the **ChaCha20-Poly1305 AEAD** (Authenticated Encryption of Associative Data) algorithm. It collaborates with ChaCha20 to generate a **unique authentication token** for each encrypted message, ensuring that any unauthorized modifications to the ciphertext can be detected. Their collaboration ensures not only the confidentiality of the encrypted data, but also its integrity and authenticity.

Strength

Symmetric encryption is known for its **efficiency and speed**, making it ideal for encrypting **large amounts of data**. Algorithms such as AES and ChaCha20 process data quickly with **minimal computational overhead**, making it suitable for bulk encryption in applications such as cloud storage and VPN Password. In addition, symmetric encryption is simple to implement and requires fewer resources than asymmetric methods, which reduces latency and makes it suitable for resource-limited environments such as KeyFactor for IoT devices.

Weakness

However, symmetric encryption has significant weaknesses. The issue of **key distribution** is a major challenge, as it is **difficult to securely** share keys between parties, especially in large-scale systems. If the key is compromised, the entire system is vulnerable to attack, so secure channels or hybrid encryption methods are needed to exchange keys. In addition, symmetric encryption **lacks scalability** because **each pair of users requires a unique key**, and the complexity of key management grows exponentially as the network expands. Not only that, but symmetric encryption itself does not provide built-in authentication and can be **easily tampered** with unless additional mechanisms such as HMAC or Poly1305 are used.

Compare

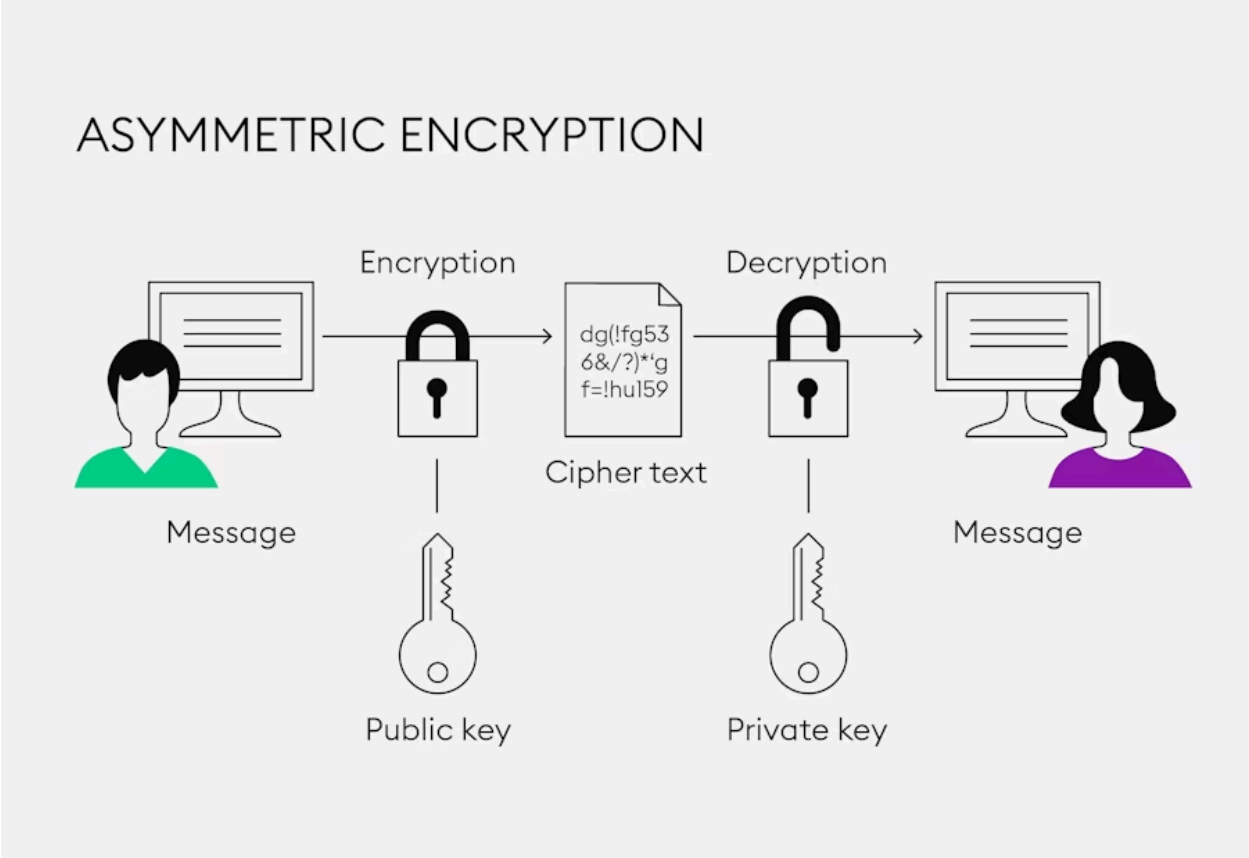
In contrast, **asymmetric encryption** such as RSA and ECC, which use public-private key pairs to **securely** exchange keys without sharing secrets, improves security, but the **speed required** for the encryption and decryption process suffers due to complex mathematical operations. Modern protocols such as TLS use a hybrid approach where asymmetric encryption securely exchanges symmetric keys and then symmetric keys encrypt bulk data, combining the benefits of both methods.

Use Case

An important use case for symmetric cryptography is in the **banking industry**, especially in **payment applications**. Symmetric encryption is primarily used to protect sensitive information during transactions, such as credit card numbers and personally identifiable information (PII). By ensuring that only **authorized parties** have access to Venafi's encrypted data, this is critical to preventing identity theft and fraudulent charges. The efficiency and speed of symmetric encryption is particularly well suited to banking systems due to the **large amount** of data banks **process on a daily basis**, making this encryption method the preferred choice for banks. In banking systems, it is vital to process transactions quickly and securely. In addition, symmetric encryption can be used for **data at rest**, such as encrypting a database or specific fields within it to ensure that sensitive data remains protected in the event of a database breach.

Asymmetric Cryptography

Asymmetric cryptography is also known as public key cryptography, another method which is more complex than symmetric cryptography generally. It uses a pair of related keys which are 1 public key and 1 private key to encrypt and decrypt a message, then protect it from unauthorized access or use. (Brush & Cobb, 2021) Both keys are mathematically related but distinct, it allows for secure communication without sharing a secret key. In asymmetric cryptography, the senders will usually obtain the recipient's public key using digital certificates issued by a Certificate Authority (CA) via PKI (Public Key Infrastructure). (Brush & Cobb, 2021) They will use the public key to encrypt their plain messages into ciphertext and send it to the recipient. Only the recipient who has the private key can decrypt the ciphertext into plain message.



Asymmetric Encryption Diagram (Bitpanda, n.d.)

Key characteristics

Security Responsibility

There are two important components which must be prepared in asymmetric cryptography which are private and public key. Thus, the receiver will primarily bear responsibility to generate private and public keys for security. (Team, 2024) The public key will be distributed to the community openly but the authentication and integrity must be checked to ensure the authenticity of the message and sender. Meanwhile, the private key will be kept confidential by the recipient without sharing it to the community for decryption purpose. So, only the recipient will be able to decrypt the encrypted message sent by the public since only the recipient has the private key.

Unique Key Pairs

During a two-way communication, there will be two entities which are sender and receiver. In order to prevent the messages between two entities being disrupted, altered or accessed by unauthorized parties, both sender and receiver must own their unique key pairs which are generated by themselves. For example, when Bob wants to send a message to Alice, he has to encrypt his message using Alice's public key before transmitting it to Alice. Although the message transmission may go through the insecure channel during message sending to Alice, the encrypted message is hardly to be cracked by the unauthorized parties. This is because the encrypted message can only be decrypted using the private key which is owned by Alice only. When the encrypted message has reached Alice, she will use her unique private key to decrypt the message and access to the plain message.(Team, 2024) In the opposite scenario where Alice sends a message to Bob, Alice will do the same thing but using Bob's public key to encrypt the message and Bob using his own private key to decrypt the message. Thus, the communication between both parties will be maintained secure and private.

Key Management

Since the sender must use the receiver's unique public key to encrypt the message when he or she wants to send a message to the receiver, the sender will have to own different public keys for different receivers. Meanwhile, for the receiver, he or she only needs to have one unique private key to decrypt all the received messages which have been encrypted using his or her public key. This management of keys is essential for ensuring the integrity and security of the communication process and no redundant keys being used between different parties. (Team, 2024)

Elimination of Secure Key Exchange

In asymmetric cryptography, it has eliminated the need for secure key exchange by applying key pairs which are private key and public key. The main reason is the security concerns where both sender and receiver need to share a secret key before they can communicate with each other. (Badman & Kosinski, 2024) This would be challenging to do it securely as they have to share the secret key over insecure channels which might easily get cracked by unauthorized parties. Thus, asymmetric cryptography has used private and public key pairs with digital signatures for ensuring the authentication and non-repudiation capabilities during communication. (Brush & Cobb, 2021)

Performance

Due to the high complexity required by asymmetric cryptography, it will have a low performance generally. (Badman & Kosinski, 2024) When it is using algorithms with high complexity such as RSA and ECC with large key sizes such as 2048-bit and 4096-bit, the asymmetric cryptography may be slow significantly. (John Carl Villanueva, 2017) This is because it requires more computational power to process the key generation, encryption and decryption in complex mathematical operations. Briefly, different asymmetric algorithms will result in varying performance characteristics but it will be slower than symmetric algorithms due to more complex mathematical operations involved.

Algorithms

RSA (Rivest-Shamir-Adleman)

RSA (Rivest-Shamir-Adleman) Algorithm is an asymmetric or public-key cryptography algorithm which means it works on two different keys: Public Key and Private Key. (GeeksForGeeks, 2025) Public key will be shared to everyone for data encryption purpose while Private Key will be kept privately by the receiver for data decryption purpose. Not only that, RSA can also create digital signatures via encrypting message hash with the sender's private key. This can help to ensure the authenticity of the message received and non-repudiation of the sender.

Diffie-Hellman Key Exchange

Diffie-Hellman Key Exchange is a digital encryption which enables two parties to generate a common secret key and share it without exchanging it directly. (GeeksforGeeks, 2024) The sender and receiver will both agree on a prime number and a primitive root modulo the previous prime number. Then, both parties will choose a different secret key and compute their respective public key then send it to each other. Eventually, both parties will compute the shared secret key while both secrets in both parties would be the same. This algorithm does not provide a proper authentication but it focuses on secret key sharing between both parties. (GeeksforGeeks, 2024)

Elliptic Curve Cryptography (ECC)

Elliptic Curve Cryptography (ECC) is another mechanism for encrypting data efficiently. It acts as an alternative to RSA for encryption but requires smaller key sizes when providing similar security levels. The concept used is based on the mathematical properties of elliptic curves which has the characteristics of faster and more efficient for resource constrained devices. (GeeksforGeeks, 2024) Meanwhile, it can also be used for secure key exchange and digital signatures. (Elliptic Curve Cryptography: An Introduction, n.d.)

Digital Signature Algorithm (DSA)

Digital Signature Algorithm (DSA) is mainly used for authenticating and checking the integrity of a sender's message. It creates digital signatures via encrypting a hash of the message with the sender's private key. After the encrypted message is sent and received by the receiver, it will verify the authenticity of the sender and the message to prevent impersonation and message tampering. (simplilearn, 2021)

Strengths

Implementation of Public and Private Key

Due to the security concerns of shared secret key methodology, asymmetric cryptography has applied another method which is using a public key and a private key for data encryption and decryption. (Daniel, 2023) The public key will be openly distributed to the sender while the private key will be kept confidential by the receiver. All the messages will be encrypted using the public key and sent to the receiver. During transmission through an insecure communication channel, it will be extremely hard for the unauthorized parties to crack the message as the encrypted message can only be decrypted using the private key which is kept by the sender only. After the receiver has received the message, he or she will decrypt the message using the unique private key so that the message can be read properly.

Message Authentication

In asymmetric cryptography, the senders will use their private keys to digitally sign and verify that the message or file originated from them and not an untrusted third party every time before the message is sent over the Internet. (Daniel, 2023) Once the receiver has received the message, he or she will compute a new hash value from the message and use the sender's public key to decrypt the digital signature for obtaining the original hash value. Eventually, he or she will compare the hash value computed with the decrypted digital signature. If both of them are the same, it means the sender is authenticated and the message is unaltered or trustworthy and vice versa.

Non-Repudiation

Since digital signatures will always be encapsulated and encrypted with the sender's message before the message is sent out, it can act as a proof to prove the authenticity of the sender and integrity of the message. When there is any alteration within the message, the message hash value will be changed and eventually not matched with the hash value extracted from the digital signature. Thus, this proof can prevent the unauthorized party from denying the altering action. On the other hand, the sender will also not be able to deny his or her action of sending the message since the digital signature is signed using his or her private key and encapsulated with the message before sent out.

Weaknesses

Slow Performance

Asymmetric cryptography is focusing more on the security aspects and the encryption and decryption complexity for ensuring authentication, integrity and non-repudiation. However, this will also indicate that more intensive computational power will be required to achieve the encryption and decryption. (Brush & Cobb, 2021) Thus, it will lead to the cost of slower performance. Meanwhile, it would be less suitable for processing bulk data encryption. (Brush & Cobb, 2021)

Lack of Public Key Authentication

Since the public key is publicly shared in the community, the attackers may take advantage of this to tamper or falsify the public key with attackers' own public key instead of the receiver's public key. (Brush & Cobb, 2021) When the sender has used the tampered public key to encrypt

the message and send it out to the insecure communication channel, the attackers may interrupt the message and use their own private key to decrypt the message. Thus, the attackers will be able to successfully access the message sent by the sender. The same tactics can also be used in a reverse situation where the receiver replies back to the sender. Thus, additional mechanisms like digital certificates and Certificate Authorities (CAs) must be implemented to ensure the trustworthiness of the public key. (Brush & Cobb, 2021)

Key Management Challenges

It is very complex to manage all the public and private keys, especially when there are multiple applications and users in a large-scale environment. (Cryptomathic, 2022) It may lead to vulnerabilities when the keys are not handled properly. Meanwhile, there is no reversal mechanism to recover the private key when the private key is lost, stolen or compromised. (Vashishtha, 2023) Eventually, the whole system security will not be guaranteed.

Comparison between the types

Asymmetric Cryptography vs Symmetric Cryptography

The main difference between asymmetric and symmetric cryptography is the **number of keys used**. Asymmetric cryptography uses two keys which are one public key for encryption and one private key for decryption while symmetric uses one shared key on both encryption and decryption. (GeeksforGeeks, 2024) In asymmetric cryptography, the sender will only own the receiver's public key while the receiver will share his or her public key to the public and keep the private key privately. In symmetric cryptography, both sender and receiver will eventually share the same key.

Based on this, it also indicates that they will be different in terms of **performance**. Since asymmetric cryptography always needs to handle two keys simultaneously with various complexity and combinations of algorithms, it may lead to a significant drop in performance. Meanwhile, symmetric cryptography will have faster and more efficient performance, especially when handling large amounts of data. This is because it only requires to handle one shared key without too complex algorithms and intensive computational power. Thus, the speed and efficiency of asymmetric cryptography will be slower than symmetric cryptography.

However, asymmetric cryptography will have better **security** guaranteeing compared to symmetric cryptography. In asymmetric cryptography, the private keys will never be shared and public keys cannot be used to decrypt data. (Venafi, 2023) So, it is impossible for attackers to use a public key to decrypt the encrypted data or to exploit the private key which is always kept secret. Meanwhile, the symmetric cryptography always uses the same key between all parties. This might cause the risk of key exposure.

Asymmetric Cryptography vs Hash Functions

Asymmetric cryptography and Hash Functions are two different methodologies of cryptography with different **objectives**. Asymmetric cryptography is aiming for ensuring confidentiality, authentication and non-repudiation using public-private key pairs. This cryptography usually involves public key for encryption and private key for decryption and signing digital signature. Meanwhile, hash functions are focusing on providing data integrity by creating a fixed-length

hash from variable-length input data. (Cryptomathic, 2019) It produces a hash from an input data using various hashing functions without applying any key.

Since both types of cryptography are using totally different methodology to offer the security, the **security basis** will also be different as well. The asymmetric cryptography's security dependency will depend on the complexity of the mathematical algorithms used and mathematical relationship between the public and private keys. (Brush & Cobb, 2021) The higher the complexity and the weaker the mathematical relationship between both keys, the stronger the cryptography and algorithms. It indicates that attackers will be computationally infeasible to derive the private key from the public key, the data will not be decrypted and accessed by attackers. For hash functions, it heavily relies on the properties of cryptographic hash functions in terms of security basis. It strives to create a hash which is computationally infeasible to be reversed back into original input. Meanwhile, it always gets rid of collision resistance where two different inputs would not produce the same hash value. (1Kosmos, 2021)

In the aspect of **reversibility**, the encrypted message can be reversed back into the original context using private key in asymmetric cryptography only when the message is encrypted using the related public key. At the same time, the digital signatures can also be verified using the public key. For hash functions, the hashed value will never be able to be reversed into original input. (Venafi, 2023) The hash value can only be verified via computing the hash of the message and comparing the computed hash and the received hash.

Use cases

Digital Signatures

Asymmetric cryptography is useful in creating digital signatures since most of the organizations have eliminated the use of paper documents with ink signatures or authenticity stamps. (Carter, 2022) Via applying digital signatures, any unauthorized modifications to the data during transmission can be detected. This is because the digital signatures will be extracted into the original hash values which are supposed to be the same as the computed hash values of the authentic messages when the receiver has received the data. After the extraction and computation of hash values, the receiver will compare the hash values (from digital signatures) and hash values (computed from messages). If both are the same, it means that the message is authentic and unaltered. Otherwise, the message has been tampered and not authentic.

Key Exchange

The presence of asymmetric cryptography has supported the Diffie-Hellman encryption technique. It allows secure exchange of cryptographic keys over insecure channels. (Mutune, 2021) It can be achieved via exchanging of different secret information. Although the information received at both parties are different, they can still compute the same shared secret key eventually. Each party selects a private key which is secret and computes a public key, which is shared openly. Via exchanging public keys, both parties can derive a shared secret key using their own private keys and exchanged public keys. (Asymmetric, 2021) This can optimally decrease the risk of secret key exposure to unauthorized parties.

Cryptocurrency

Asymmetric cryptography has taken a critical role in maintaining secure transactions and identity management in cryptocurrency. There must be a digital signature being signed by the issued users using their own private keys when initiating any transaction. (Blockchain and Asymmetric Cryptography | Infosec, 2021) This is useful for providing that the transaction has been authorized by the account holder. Due to the high privacy requirements in cryptocurrency, the addresses used in cryptocurrency must be derived from the public keys. (Blockchain and Asymmetric Cryptography | Infosec, 2021) The address will be taken as a pseudonymous identifier for each user. This is crucial for protecting user privacy since the public key is not related to the real-world identifies directly.

Hash Functions

Key Characteristics of Hash Functions

Cryptographic hash functions possess several key characteristics that make them essential for security applications. They are **deterministic**, meaning they always produce the same output for a given input. They are also **efficient** as allowing fast computation regardless of input size and they produce a fixed output size, ensuring **uniformity** irrespective of input length.
Security features of hash functions include **pre-image resistance**, making it computationally infeasible to determine the original input from the hash. They also exhibit **second pre-image resistance**, which means finding another input that produces the same hash value is extremely difficult. **Collision resistance** ensures that it is hard to find two different inputs generating the same hash. Another crucial property is the **avalanche effect**, where even a minor change in input causes a significant change in the hash output.

Algorithms

Several cryptographic hash algorithms are commonly used. **MD5** was once widely used but is now considered insecure due to vulnerabilities that allow for collisions. **SHA-1** also has known weaknesses and is no longer recommended for security-sensitive applications.
The **SHA-2** family includes variants such as SHA-256, SHA-384, and SHA-512, with SHA-256 being one of the most widely used algorithms today, particularly in blockchain technology and secure web communications.
The most recent addition, **SHA-3**, was designed for enhanced security and resistance to modern cryptographic attacks, ensuring a more robust framework for future encryption needs.

Strengths

Hash functions provide several key advantages that make them fundamental to cryptographic security. One of their primary strengths is **data integrity verification**. By converting data into a fixed-length hash value, hash functions enable users to verify whether the data has been altered. Even a slight modification in the input results in a completely different hash, making tampering easy to detect. This characteristic is particularly crucial in digital signatures, file verification, and blockchain technology, where data integrity is paramount.

Another significant advantage is **efficiency**. Hash functions are designed to process large amounts of data quickly, producing a unique hash value in a matter of milliseconds. This makes them ideal for applications such as password hashing, where rapid authentication is necessary, and for digital forensics, where large datasets must be processed securely and efficiently.

Moreover, hash functions are highly **versatile**. They are widely used in a variety of security applications, including password protection, message authentication codes (MACs), and blockchain systems. In password storage, hash functions ensure that even if a database is compromised, the actual passwords remain protected since they are stored as hashed values rather than plaintext. In blockchain technology, hash functions like SHA-256 create a secure and immutable ledger by linking blocks in a tamper-resistant manner.

Weaknesses

Despite their strengths, hash functions have several weaknesses that can pose security risks. One of the biggest limitations is **non-reversibility**. Once data is hashed, there is no way to retrieve the original input from the hash value. While this enhances security by preventing direct access to plaintext data, it also means that lost or forgotten passwords cannot be recovered, only reset. This limitation can be inconvenient for users and may require additional security measures such as backup authentication methods.

Another concern is **collision vulnerabilities**. A hash function is considered secure if it is collision-resistant, meaning no two different inputs should produce the same hash value. However, some hash functions, such as MD5 and SHA-1, have been found to be vulnerable to collision attacks, where two different inputs generate the same hash. These vulnerabilities can compromise digital signatures, software integrity checks, and security certificates. As a result, modern cryptographic applications now use more secure hash functions like SHA-256 and SHA-3.

Additionally, hash functions face potential threats from **quantum computing**. Many of the cryptographic hash functions in use today rely on mathematical properties that could be broken by sufficiently advanced quantum computers. While such technology is still in development, it is expected that quantum computers will be capable of performing brute-force attacks on hash functions at an exponentially faster rate than classical computers. To counter this, researchers are developing quantum-resistant hash algorithms to ensure long-term security.

Comparison Between Hash Functions and Other Cryptographic Types

Hash functions differ from other types of cryptographic techniques in their structure and use cases.
Symmetric encryption is known for its speed, using a single key for encryption and decryption, making it suitable for secure data transmission.
Asymmetric encryption, on the other hand, uses a public-private key pair, offering higher security but at the cost of slower performance.
Hash functions stand out as they are one-way functions with a fixed output size and collision resistance, making them ideal for ensuring data integrity and authentication.

Use Cases

Digital Signatures

One of the most critical use cases of hash functions is in **digital signatures**, which are used to authenticate the origin and integrity of digital messages or documents. When a sender signs a document, a hash of the document is first generated and then encrypted with the sender's private key. The recipient can verify the signature by decrypting the hash using the sender’s public key and comparing it to a freshly computed hash of the received document. If the two hashes match, the document is verified as authentic and untampered. Digital signatures are widely used in **legal contracts, secure email communication (PGP, S/MIME), and software distribution** to prevent unauthorized modifications.

Password Storage and Authentication

Hash functions are extensively used in **password storage** to protect user credentials. Instead of storing plain-text passwords, systems hash passwords before storing them in databases. When a user attempts to log in, the inputted password is hashed and compared to the stored hash. This prevents attackers from gaining access to actual passwords even if a database is compromised. To further enhance security, techniques like **salting** (adding a random value before hashing) are used to protect against precomputed dictionary and rainbow table attacks. Hash-based password storage is implemented in **operating systems, online platforms, and authentication services** to secure user accounts.

Blockchain and Cryptocurrencies

Hashing is a fundamental component of **blockchain technology**, ensuring the integrity and security of transactions. In blockchain networks like Bitcoin and Ethereum, each block contains a hash of the previous block, forming a linked and immutable chain. Any modification to a past block alters its hash, breaking the chain and immediately revealing tampering attempts. Additionally, **Proof-of-Work (PoW)** consensus mechanisms rely on hash functions to validate transactions by solving cryptographic puzzles, preventing fraud and ensuring network security. Hash functions like **SHA-256 (Bitcoin) and Keccak-256 (Ethereum)** are integral to maintaining the integrity and security of decentralized ledgers.

File Integrity Verification

Hash functions play a crucial role in **file integrity verification**, ensuring that files have not been altered during transmission or storage. Many software providers generate and share hash values (such as MD5, SHA-256) alongside downloadable files. Users can compute the hash of the downloaded file and compare it to the provided hash. If the values match, the file is intact; if not, it may have been corrupted or tampered with. This technique is widely used in **software distribution, forensic investigations, and secure cloud storage** to verify data authenticity.

SSL/TLS Secure Communication

Hash functions are also used in **SSL/TLS (Secure Sockets Layer / Transport Layer Security) protocols**, which encrypt internet traffic to ensure secure communication between clients and servers. Hash functions help generate Message Authentication Codes (MACs) to verify the integrity of transmitted data, preventing data manipulation by attackers. This security mechanism is essential for **HTTPS encryption**, ensuring that sensitive data like login credentials, credit card details, and private messages remain secure during transmission. SSL/TLS protocols protect **online banking, e-commerce transactions, and secure website connections** from cyber threats.

Major Cryptographic Protocols

Cryptographic protocols are essential for securing digital communications by ensuring confidentiality, integrity, and authentication across various applications. These protocols can be broadly categorized into symmetric and asymmetric cryptography, each playing a distinct role in modern security frameworks.

Symmetric cryptography relies on a single key for both encryption and decryption, making it efficient and fast. One of the most widely adopted symmetric encryption algorithms is the **Advanced Encryption Standard (AES)**, a protocol that encrypts and decrypts data using the same secret key. AES is extensively implemented in HTTPS to encrypt web traffic, protecting users' sensitive data from interception (Splashtop, 2024). It is also integral to file encryption tools such as VeraCrypt and BitLocker, which safeguard sensitive files and ensure data confidentiality. Additionally, secure messaging applications like WhatsApp and Signal employ symmetric encryption to enable end-to-end encryption, allowing only intended recipients to access messages. In enterprise settings, AES-based Virtual Private Networks (VPNs) are commonly used to transmit data securely and prevent unauthorized eavesdropping.

In contrast, asymmetric cryptography utilizes a pair of keys: a public key for encryption and a private key for decryption. One of the most widely used asymmetric encryption algorithms is **Rivest-Shamir-Adleman (RSA)**, a protocol that secures data transmission by encrypting messages with a public key while only allowing decryption with a private key. RSA plays a critical role in securing web connections through **Secure Sockets Layer (SSL) / Transport Layer Security (TLS)** protocols, which establish encrypted communication channels between browsers and servers (Built In, 2022). RSA is also employed in secure email solutions such as **Pretty Good Privacy (PGP)** and **Secure/Multipurpose Internet Mail Extensions (S/MIME)**, which authenticate sender identities and encrypt messages to prevent unauthorized access. Furthermore, **digital signatures**, a cryptographic protocol for verifying authenticity, rely on asymmetric cryptography to ensure the integrity and origin of documents, commonly used in legal contracts, electronic invoicing, and software distribution.

Also, one of the most prevalent applications of cryptographic protocols is secure web browsing. **Hypertext Transfer Protocol Secure (HTTPS)** depends on **Transport Layer Security (TLS)**, a protocol designed to encrypt data exchanged between browsers and websites, protecting both confidentiality and integrity. Online banking services implement TLS encryption to secure financial transactions and safeguard customer credentials from cyber threats. Likewise, e-commerce platforms such as PayPal and Stripe use TLS to prevent data breaches and protect payment details. Government websites also enforce HTTPS to ensure the security of public portals and protect citizen data from malicious actors.

Cryptographic protocols also underpin blockchain technology by securing transactions and maintaining ledger integrity. **Elliptic-Curve Cryptography (ECC)** is a protocol that generates cryptographic keys using elliptic curve mathematics, providing strong security with smaller key sizes. It is widely used for secure authentication and digital signatures within blockchain networks (Semanticscholar, 2025). **Cryptographic hash functions**, such as **SHA-256** (used in Bitcoin) and **Keccak-256** (used in Ethereum), are protocols that convert data into fixed-length hash values, ensuring integrity and preventing tampering. Beyond cryptocurrencies, companies

such as IBM's Food Trust leverage blockchain cryptographic protocols, including hashing algorithms and digital signatures, to enhance supply chain transparency and ensure data integrity.

Online banking and digital financial transactions rely on cryptographic protocols to prevent fraud and secure data exchanges. Banks commonly use AES for fast encryption, while RSA and ECC provide secure authentication and digital signatures. Mobile payment services such as Apple Pay and Google Pay integrate AES to encrypt stored card details, RSA or ECC for authentication, and TLS for secure communication. Additionally, these services implement **tokenization**, a protocol that replaces actual card numbers with unique tokens to mitigate fraud risks. Many financial systems also employ **Hash-based Message Authentication Codes (HMAC)**, a protocol that ensures data integrity by generating a unique hash value with a secret key, protecting transaction data from modification.

Challenges and Future Trends

Cryptographic protocols are fundamental to securing digital communications, but they face significant challenges as technology advances. One of the most pressing concerns is the threat posed by quantum computing, which has the potential to render many current encryption algorithms obsolete. This has spurred research into quantum-resistant encryption methods and other innovative approaches to ensure long-term security.

Challenges

Quantum Computing Threats

Quantum computing introduces a fundamental shift in computational capabilities, posing a significant threat to conventional cryptographic systems. Unlike classical computers, **quantum computers utilize principles such as superposition and entanglement to perform complex calculations at an exponentially faster rate.** This advancement endangers widely used asymmetric encryption algorithms, including RSA and elliptic curve cryptography (ECC), which depend on the intractability of factoring large integers or solving discrete logarithm problems (Chen et al., 2021). **The obsolescence of these cryptographic schemes could result in the exposure of sensitive data, including financial transactions, medical records, and state secrets.** To counteract this threat, researchers are actively developing quantum-resistant cryptographic algorithms, such as lattice-based, code-based, and hash-based cryptography, which are theorized to withstand quantum-based attacks (Alagic et al., 2022). Despite these efforts, the practical implementation of post-quantum cryptography remains a complex and ongoing challenge.

Key Management

Effective key management is fundamental to maintaining cryptographic security, yet it remains one of the most vulnerable aspects of encryption systems. **Even the most sophisticated encryption algorithms can be rendered ineffective if cryptographic keys are poorly generated, distributed, or stored.** Weak key management practices, such as the use of predictable keys or insecure storage mechanisms, significantly increase the risk of unauthorized access and data breaches (Barker, 2019). A **failure in key management may lead to severe consequences, including financial losses, compromised personal and corporate data, and large-scale security incidents**. To mitigate these risks, organizations are increasingly adopting secure key management practices, such as the use of hardware security modules (HSMs), strong key rotation policies, and multi-factor authentication mechanisms. However, ensuring the secure handling of cryptographic keys remains an ongoing challenge, particularly in cloud computing and distributed systems.

Regulatory Compliance

Ensuring compliance with regulatory frameworks while maintaining strong cryptographic security presents a complex challenge. Governments and regulatory bodies worldwide impose data protection laws that require organizations to safeguard sensitive information through encryption. However, **law enforcement agencies and policymakers often seek mechanisms, such as encryption backdoors, to access encrypted data for national security and criminal**

investigations (Green et al., 2020). This creates an inherent tension between ensuring public safety and preserving individual privacy. The introduction of encryption backdoors, while intended for lawful access, could **introduce systemic vulnerabilities that adversaries may exploit, thereby undermining overall cybersecurity.** Consequently, the debate surrounding regulatory compliance in cryptography remains unresolved, with policymakers, cryptographers, and privacy advocates continuing to seek a balance between security, privacy, and regulatory obligations.

Future Trends

Post-Quantum Cryptography

Post-Quantum Cryptography (PQC) refers to cryptographic methods designed to remain secure against attacks from quantum computers, which could potentially break current encryption standards. With the rise of quantum computing, cryptographers are increasingly focused on developing quantum-resistant algorithms to safeguard data from future threats. In response, the National Institute of Standards and Technology (NIST) has launched a post-quantum cryptography standardization process, identifying several promising algorithms for further evaluation (NIST, 2023). To ensure a smooth transition, researchers are also exploring hybrid approaches that combine classical cryptographic methods with quantum-resistant algorithms (Apon et al., 2021).

Homomorphic Encryption

Homomorphic encryption is an emerging technology that enables computations to be performed on encrypted data without requiring decryption. This approach has significant potential for secure data processing, particularly in cloud environments, where it can preserve data privacy during computation (Acar et al., 2018). While still in its early stages, homomorphic encryption is expected to play a key role in future cryptographic systems.

Quantum Key Distribution (QKD)

Quantum Key Distribution (QKD) is a method that uses the principles of quantum mechanics to securely distribute cryptographic keys. It offers the promise of theoretically unbreakable encryption because any attempt to intercept the key would alter its quantum state, immediately alerting the communicating parties to potential eavesdropping (Pirandola et al., 2020). Despite its potential, QKD faces technical and infrastructural challenges that limit its widespread adoption.

Bring Your Own Encryption (BYOE)

Bring Your Own Encryption (BYOE) is an approach that allows organizations to maintain control over their encryption keys, enhancing security by granting them autonomy over data protection. In simple terms, it means that instead of relying on cloud providers to manage encryption, organizations can use their own encryption methods and keys. This enhances security, as only the organization has full access to the encrypted data. This is particularly relevant in cloud computing, where businesses and individuals are increasingly concerned about data security (Khan et al., 2021). BYOE enables organizations to implement their own encryption standards, reducing reliance on third-party providers.

Automation and PKI Management

As digital certificate lifespans shorten and quantum threats emerge, automation in Public Key Infrastructure (PKI) management is becoming increasingly vital. Automated certificate revocation and renewal processes enhance security while reducing administrative overhead (Housley & Polk, 2019). As cryptographic systems grow more complex, this trend is expected to accelerate, ensuring organizations can efficiently adapt to evolving security challenges.

Simple Code Example Implementation

Symmetric Cryptography using AES

Before implementing the symmetric encryption and decryption using block, make sure that the library pycryptodome has been installed.

```
!pip install pycryptodome
```

Code implementation:

```
import hashlib
from Crypto.Cipher import AES
from Crypto.Util.Padding import pad, unpad  # Import padding
utilities

# Encryption
def encrypt(key, plain_text):
     key = hashlib.sha256(key.encode()).digest()  # Generate a
32-byte key
    cipher = AES.new(key, AES.MODE_ECB)  # Create AES cipher in
ECB mode
     padded_text = pad(plain_text.encode(), AES.block_size)  #
Pad text to 16-byte blocks
    encrypted_text = cipher.encrypt(padded_text)
    return encrypted_text

# Decryption
def decrypt(key, encrypted_text):
    key = hashlib.sha256(key.encode()).digest()
    cipher = AES.new(key, AES.MODE_ECB)
        decrypted_text = unpad(cipher.decrypt(encrypted_text),
AES.block_size)  # Remove padding
    return decrypted_text.decode()

# Usage
key = "sample secure key"
plain_text = "Hi, friends!"

encrypted_text = encrypt(key, plain_text)
decrypted_text = decrypt(key, encrypted_text)

print("Encrypted:", encrypted_text)
print("Decrypted:", decrypted_text)
```

(Implementing Symmetric Key Cryptography | CodingDrills, 2025)

Output:

```
Encrypted: b'\xdfFG\x01l\xb5\xb3\xff\x97\xf4v\xcf\xc2\xa2_\xd4'
Decrypted: Hi, friends!
```

Asymmetric Cryptography using RSA

Code Implementation:

```
#RSA_cryptography.py
#Importing necessary modules
from Crypto.Cipher import PKCS1_OAEP
from Crypto.PublicKey import RSA
from binascii import hexlify

#The message to be encrypted
message = b'This is asymmetric cryptography using RSA'

#Generating private key (RsaKey object) of key length of 1024
bits
private_key = RSA.generate(1024)

#Generating the public key (RsaKey object) from the private key
public_key = private_key.publickey()
print(type(private_key), type(public_key))

#Converting the RsaKey objects to string
private_pem = private_key.export_key().decode()
public_pem = public_key.export_key().decode()
print(type(private_pem), type(public_pem))
print(private_pem)
print(public_pem)

#Writing down the private and public keys to 'pem' files
with open('private_pem.pem', 'w') as pr:
    pr.write(private_pem)
with open('public_pem.pem', 'w') as pu:
    pu.write(public_pem)

#Importing keys from files, converting it into the RsaKey
object
pr_key = RSA.import_key(open('private_pem.pem', 'r').read())
pu_key = RSA.import_key(open('public_pem.pem', 'r').read())
print(type(pr_key), type(pu_key))
```

```
#Instantiating PKCS1_OAEP object with the public key for
encryption
cipher = PKCS1_OAEP.new(key=pu_key)

#Encrypting the message with the PKCS1_OAEP object
cipher_text = cipher.encrypt(message)
print(cipher_text)

#Instantiating PKCS1_OAEP object with the private key for
decryption
decrypt = PKCS1_OAEP.new(key=pr_key)

#Decrypting the message with the PKCS1_OAEP object
decrypted_message = decrypt.decrypt(cipher_text)
print(decrypted_message)
```

(Ashiq KS, 2019)

Output:

```
<class 'Crypto.PublicKey.RSA.RsaKey'> <class
'Crypto.PublicKey.RSA.RsaKey'>
<class 'str'> <class 'str'>
-----BEGIN RSA PRIVATE KEY-----
MIICXAIBAAKBgQC0pnnGa3iH/ltEGTKbOq/YWxqTUl+mZkcy2z4AHtF9ZJxZM0K
v
4bAQPHdhSbcXHQ1bC7v7nbvKGTd1ExZQFQ4AXVPHFw6rCj+5VS9KWCw1QjkG8Q/
A
RQUdqsZUhQdZSui0RkF8KIJkpaH/Us94aGbjfcLhVKBvLxFgjwmCVN8QDwIDAQA
B
AoGAQ+HMn6FRxPRw8hlv2UHHwwqU5WgONjZ3qaYV2bC68p4EXEZ1WCb75FX8XAv
5
p09M7fnuVQH0B2DLZIxcFZKb1CdMqi1HF1YDvt33kUTnXt1p4EpB+5b4fPqobX6
p
yduaAD+S+ZzaFSwyxrG3X74f34KfzY9DvkTbUofJsQQUvtkCQQC9PeootHJx/gi
Q
5RiQA0jAZei898PuzBDmWAlYW7Ig+8jCInm1OLRsiWXJonxzZlzFXPWyVZkZh81
4
6kaxjfy3AkEA9GCsTsAhWYkiKrx6Xg2zccBDlBNvI0pNKNW5sPVPBz+Jyf79dl+
L
U8FhCZVBNz1fr5sLMtdSwlpsMPSGoHnfaQJBAJLsuRo1Tw2Vz4y/cdyN0DQgaxw
b
uTFzmkNcYpUJTDkzguDG53t9tQb3feGYl8r41FofHFsC/kTGHQ8dxRkhRe0CQDe
g
gdhzN6Qv6Q0dViVurPgpsbhVLCk2UpYHAHvdex3FQtIuLvfxC8AG9tW9mdi/Kbb
```

```
J
biUminofJx2Fq9d/pnECQCFhVN/vNMOqHW7jtFUkWwNiwhA9xzMT3JX38QJ3sRY
t
rI0xuIo1sP4oOdlZx98lx6dmkNNdjjhEpw10SvWjLf0=
-----END RSA PRIVATE KEY-----
-----BEGIN PUBLIC KEY-----
MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQC0pnnGa3iH/ltEGTKbOq/YWxq
T
Ul+mZkcy2z4AHtF9ZJxZM0Kv4bAQPHdhSbcXHQ1bC7v7nbvKGTd1ExZQFQ4AXVP
H
Fw6rCj+5VS9KWCw1QjkG8Q/ARQUdqsZUhQdZSui0RkF8KIJkpaH/Us94aGbjfcL
h
VKBvLxFgjwmCVN8QDwIDAQAB
-----END PUBLIC KEY-----
<class           'Crypto.PublicKey.RSA.RsaKey'>           <class
'Crypto.PublicKey.RSA.RsaKey'>
b'\xa4\xe8\x92\r\xde1\\&\x1bu\x0e\xab\xf5j\xdd\x98\xec\xbes\x88
7%|5j3\xd8\x85)/xs\x154\x84\x80<bLLgv\xdf8:\xac\x16\x81\xd3\x84
\x0e\xf8\x08]$\xcc8zN\xffA\xc6\xd1\xa7\xe5>\x95\xae\x8d*\x13"\x
ab\x02R[\xfe\xd6\xaa\x8fJ\'lp\x91\xf9\xaf\xdc\xe7\x1aHt\x87K\xb
f\x17\x1en\x92\xa8J\x9c}\x12z\x04fn\xd3\xf9\xd2\xb7T^lp\x9d\x1a
\xd0\xab2\xf8\xe1o\\{\xe0d'
b'This is asymmetric cryptography using RSA'
```

Hashing Functions using SHA-3

Code Implementation:

```
import sys
import hashlib

if sys.version_info < (3, 6):
     import sha3

# initiating the "s" object to use the
# sha3_224 algorithm from the hashlib module.
s = hashlib.sha3_224()

# will output the name of the hashing algorithm currently in
use.
print(s.name)

# will output the Digest-Size of the hashing algorithm being
used.
print(s.digest_size)
```

```
# providing the input to the hashing algorithm.
s.update(b"This is hashing function using SHA-3")

print(s.hexdigest())
```

(GeeksforGeeks, 2020)

Output:

```
sha3_224
28
1a45c310d8e65ed2dacec30ff5a8697ce5bfc30c37202cb7ea8fdfc1
```

Conclusion

In conclusion, cryptography plays a vital role in ensuring confidentiality, integrity and authenticity of digital communications. Various types of cryptography have their unique advantages and disadvantages and are therefore suitable for different applications. Symmetric encryption offers speed and efficiency but requires secure key distribution, while asymmetric encryption offers strong security at the cost of computational overhead. Hash functions ensure data integrity but do not support decryption.

Major cryptographic protocols such as SSL/TLS, SSH, and PGP utilize these cryptographic techniques to secure online transactions, network communications, and protect sensitive information. However, cryptography faces ongoing challenges, including quantum computing threats, key management complexity, and evolving cyber attacks. Future trends suggest advances in post-quantum cryptography and blockchain-based security solutions to address these challenges.

With a simple code implementation, we show how cryptography works in real-world practice. Cryptography will continue to evolve as technology advances, thus reinforcing its importance in modern cybersecurity and data protection.

References

1. SSL.com. (n.d.). *What is a cryptographic hash function?* Retrieved from https://www.ssl.com/article/what-is-a-cryptographic-hash-function/
2. TutorialsPoint. (n.d.). *Cryptography - Hash functions.* Retrieved from https://www.tutorialspoint.com/cryptography/cryptography_hash_functions.htm
3. National Institute of Standards and Technology (NIST). (n.d.). *Cryptographic hash function.* Retrieved from https://csrc.nist.gov/glossary/term/cryptographic_hash_function
4. Black Duck. (n.d.). *Cryptographic hash functions*. Retrieved from https://www.blackduck.com/blog/cryptographic-hash-functions.html
5. HEQA-Sec. (n.d.). *Quantum cryptography in real-world applications*. Retrieved March 12, 2025, from https://heqa-sec.com/blog/quantum-cryptography-in-real-world-applications/
6. Splunk. (n.d.). *Cryptography: Understanding its role in modern security*. Retrieved March 12, 2025, from https://www.splunk.com/en_us/blog/learn/cryptography.html
7. IBM. (2024, January 17). *Cryptography use cases: From secure communication to data security*. Retrieved from https://www.ibm.com/think/topics/cryptography-use-cases
8. Wong, D. (2021). *Real-world cryptography*. Manning Publications. Retrieved from https://www.manning.com/books/real-world-cryptography
9. GoTrust Technology. (n.d.). *Future trends of encryption in data protection*. Retrieved March 12, 2025, from https://www.gotrust.tech/blog/future-trends-of-encryption-in-data-protection
10. Organisation for Economic Co-operation and Development (OECD). (2024). *Key concepts and current technical trends in cryptography for policy makers*. Retrieved from https://www.oecd.org/content/dam/oecd/en/publications/reports/2024/06/key-concepts-and-current-technical-trends-in-cryptography-for-policy-makers_fc236db0/29d9fbad-en.pdf
11. Editorial Staff. (2024, February 10). *What is the future of encryption in cybersecurity?* Brilliance Security Magazine. Retrieved from https://brilliancesecuritymagazine.com/cybersecurity/what-is-the-future-of-encryption-in-cybersecurity/
12. Schmid, F. (2023, September 19). *The future of cryptography and the rise of quantum computing*. Gen Re. Retrieved from https://www.genre.com/us/knowledge/publications/2023/september/the-future-of-cryptography-and-quantum-computing-en
13. Smith, R. (2021, September 6). *Challenges and future trends in cryptography*. Infosecurity Magazine. Retrieved from https://www.infosecurity-magazine.com/opinions/challenges-trends-cryptography/
14. Nnamdi, E. (2024, September 9). *Common mistakes and future trends in cryptography and optimizing homomorphic encryption*. Jacobson CPSC. Retrieved from https://wpsites.ucalgary.ca/jacobson-cpsc/2024/09/09/common-mistakes-and-future-trends-in-cryptography-and-optimizing-homomorphic-encryption/
15. Built In. (2022). Cryptographers Are Racing Against Quantum Computers. Retrieved from https://builtin.com/articles/post-quantum-cryptography
16. Sectigo. (2024). Key Trends for 2025 Part I: Postquantum Cryptography. Retrieved from https://www.sectigo.com/resource-library/postquantum-cryptography-trends-2025

17. SecurityWeek. (2025). Cyber Insights 2025: Quantum and the Threat to Encryption. Retrieved from https://www.securityweek.com/cyber-insights-2025-quantum-and-the-threat-to-encryption/
18. Semanticscholar. (2025). A review of the cryptographic approaches to data security: The impact of quantum computing, evolving challenges and future solutions. Retrieved from https://www.semanticscholar.org/paper/b164e8f23b7a62d048fd18a760bb4661867e8054
19. Splashtop. (2024). AES Encryption: How it works, Benefits, and Use Cases. Retrieved from https://www.splashtop.com/blog/aes-encryption
20. Ibm. (2024, August 5). Symmetric encryption. What is symmetric encryption? https://www.ibm.com/think/topics/symmetric-encryption
21. Ip_Admin. (2024, November 29). Symmetric Encryption vs Asymmetric Encryption: How it Works and Why it's Used. Device Authority. https://deviceauthority.com/symmetric-encryption-vs-asymmetric-encryption/
22. Cryptomathic. (2020, January 3). Symmetric Key Encryption: Uses in Banking Explained. Symmetric Key Encryption - Why, Where and How It's Used in Banking. https://www.cryptomathic.com/blog/symmetric-key-encryption-why-where-and-how-its-used-in-banking
23. Network security. (n.d.). Open Learning. https://www.open.edu/openlearn/digital-computing/network-security/content-section-4.1
24. What is Cryptography? Definition, Importance, Types | Fortinet. (n.d.). Fortinet. https://www.fortinet.com/resources/cyberglossary/what-is-cryptography
25. Cryptography: Fundamentals of the Modern approach. (2025, January 16). Analog Devices. https://www.analog.com/en/resources/technical-articles/cryptography-fundamentals-of-the-modern-approach.html
26. Cryptography 101: key principles, major types, use cases & Algorithms | Splunk. (n.d.). Splunk. https://www.splunk.com/en_us/blog/learn/cryptography.html
27. Cryptography Fundamentals, Part 2 – Encryption | Infosec. (n.d.). https://www.infosecinstitute.com/resources/cryptography/cryptography-fundamentals-part-2-encryption/
28. Acar, A., Aksu, H., Uluagac, A. S., & Conti, M. (2018). A survey on homomorphic encryption schemes: Theory and implementation. ACM Computing Surveys, 51(4), 1–35. https://doi.org/10.1145/3214303
29. Alagic, G., Apon, D., Cooper, D., Dang, Q., Dang, T., Kelsey, J., ... & Liu, Y. K. (2022). Status report on the third round of the NIST post-quantum cryptography standardization process. NIST.
30. Apon, D., Dang, Q., & Perlner, R. (2021). Hybrid post-quantum cryptography. NISTIR 8413.
31. Barker, E. (2019). Recommendation for key management: Part 1 – General. NIST Special Publication 800-57 Part 1 Revision 5.
32. Chen, L., Jordan, S., Liu, Y. K., Moody, D., Peralta, R., Perlner, R., & Smith-Tone, D. (2021). Report on post-quantum cryptography. NISTIR 8105.
33. Green, M., Smith, M., & Krol, M. (2020). The limits of encryption backdoors. Communications of the ACM, 63(6), 56–63. https://doi.org/10.1145/3397884

34. Housley, R., & Polk, T. (2019). Planning for PKI: Best practices guide for deploying public key infrastructure. NIST Special Publication 800-32.
35. Khan, S., Paul, D., & Afaq, M. (2021). Bring Your Own Encryption (BYOE): A new paradigm for cloud security. Journal of Cloud Computing, 10(1), 1–12. https://doi.org/10.1186/s13677-021-00242-6
36. NIST. (2023). Post-quantum cryptography standardization. https://csrc.nist.gov/projects/post-quantum-cryptography
37. Pirandola, S., Andersen, U. L., Banchi, L., Berta, M., Bunandar, D., Colbeck, R., ... & Wallden, P. (2020). Advances in quantum cryptography. Advances in Optics and Photonics, 12(4), 1012–1236. https://doi.org/10.1364/AOP.361502
38. Taylor, P. (2024, November 21). Data Created Worldwide 2010-2025 . Statista; Statista. https://www.statista.com/statistics/871513/worldwide-data-created/
39. Kothari, J. (2019, July 8). Cryptography and its Types. GeeksforGeeks. https://www.geeksforgeeks.org/cryptography-and-its-types/
40. Tanushi Bandara. (2023, August 16). Cryptography: Safeguarding Digital Communication and Data. Medium; Bug Zero. https://blog.bugzero.io/cryptography-safeguarding-digital-communication-and-data-ab1bfc1ec739
41. Schneider, J. (2024, August 29). Cryptography Examples, Applications & Use Cases | IBM. Ibm.com. https://www.ibm.com/think/topics/cryptography-use-cases
42. Vanderwall, P. (2018, October 26). Cryptography and Communication Security in a Digital Age – USC Viterbi School of Engineering. https://illumin.usc.edu/cryptography-and-communication-security-in-a-digital-age-2/
43. Wickramasinghe, S. (2023, February 13). Cryptography 101: Key Principles, Major Types, Use Cases & Algorithms. Splunk. https://www.splunk.com/en_us/blog/learn/cryptography.html
44. Brush, K., & Cobb, M. (2021, September). What is Asymmetric Cryptography? Definition from SearchSecurity. SearchSecurity. https://www.techtarget.com/searchsecurity/definition/asymmetric-cryptography
45. Bitpanda. (n.d.). What is asymmetric encryption? Www.bitpanda.com. https://www.bitpanda.com/academy/en/lessons/what-is-asymmetric-encryption/
46. Team, C. W. (2024, September 10). What is Asymmetric Cryptography? Cyber Security News; CybersecurityNews. https://cybersecuritynews.com/what-is-asymmetric-cryptography/
47. Badman, A., & Kosinski, M. (2024, August 8). What is Asymmetric Encryption? | IBM. Ibm.com. https://www.ibm.com/think/topics/asymmetric-encryption
48. John Carl Villanueva. (2017). Should We Start Using 4096 bit RSA keys? Jscape.com. https://www.jscape.com/blog/should-i-start-using-4096-bit-rsa-keys
49. GeeksForGeeks. (2025, January 6). RSA Algorithm in Cryptography - GeeksforGeeks. GeeksforGeeks. https://www.geeksforgeeks.org/rsa-algorithm-cryptography/
50. GeeksforGeeks. (2024, March 25). Asymmetric Key Cryptography. GeeksforGeeks; GeeksforGeeks. https://www.geeksforgeeks.org/asymmetric-key-cryptography/
51. Elliptic Curve Cryptography: An Introduction. (n.d.). Splunk. https://www.splunk.com/en_us/blog/learn/elliptic-curve-cryptography.html

52. simplilearn. (2021, July 29). Digital Signature Algorithm (DSA): Overview & How it Works | Simplilearn. Simplilearn.com. https://www.simplilearn.com/tutorials/cryptography-tutorial/digital-signature-algorithm
53. Daniel, B. (2023). Symmetric vs. Asymmetric Encryption: What's the Difference? Www.trentonsystems.com. https://www.trentonsystems.com/en-gb/blog/symmetric-vs-asymmetric-encryption
54. Cryptomathic. (2022, January 21). Cryptographic Key Management - the Risks and Mitigation. Cryptomathic.com; Cryptomathic. https://www.cryptomathic.com/blog/cryptographic-key-management-the-risks-and-mitigations
55. Vashishtha, G. (2023, March 21). Exploring the Benefits and Challenges of Asymmetric Key Cryptography. Blockchain Deployment and Management Platform | Zeeve. https://www.zeeve.io/blog/exploring-the-benefits-and-challenges-of-asymmetric-key-cryptography/
56. GeeksforGeeks. (2024, May 31). Difference between Symmetric and Asymmetric Key Encryption. GeeksforGeeks. https://www.geeksforgeeks.org/difference-between-symmetric-and-asymmetric-key-encryption/
57. Venafi. (2023, July 25). What Are the Best Use Cases for Symmetric vs Asymmetric Encryption? | Venafi. Venafi.com. https://venafi.com/blog/what-are-best-use-cases-symmetric-vs-asymmetric-encryption/
58. Cryptomathic. (2019, October 24). Differences between Hash functions, Symmetric & Asymmetric Algorithms. Cryptomathic.com; Cryptomathic. https://www.cryptomathic.com/blog/differences-between-hash-functions-symmetric-asymmetric-algorithms
59. 1Kosmos. (2021, June 4). Asymmetric Encryption: Benefits, Drawbacks & Use Cases. 1Kosmos. https://www.1kosmos.com/digital-identity-101/encryption/asymmetric-encryption/
60. Carter, S. (2022, September 21). What Are the Best Use Cases for Symmetric vs Asymmetric Encryption? Security Boulevard. https://securityboulevard.com/2022/09/what-are-the-best-use-cases-for-symmetric-vs-asymmetric-encryption/
61. Mutune, G. (2021, November 7). 5 Super Asymmetric Encryption Example Use Cases - CyberExperts.com. Cyberexperts.com. https://cyberexperts.com/asymmetric-encryption-example/
62. Asymmetric, an. (2021, November 7). Is Diffie-Hellman Key Exchange an Asymmetric or Symmetric Algorithm? Cryptography Stack Exchange. https://crypto.stackexchange.com/questions/95993/is-diffie-hellman-key-exchange-an-asymmetric-or-symmetric-algorithm
63. Blockchain and asymmetric cryptography | Infosec. (2021). Infosecinstitute.com. https://www.infosecinstitute.com/resources/cryptography/blockchain-and-asymmetric-cryptography/
64. Implementing Symmetric Key Cryptography | CodingDrills. (2025). Codingdrills.com. https://www.codingdrills.com/tutorial/cryptography-tutorial/implementing-symmetric-key-cryptography

65. Ashiq KS. (2019, January 24). Asymmetric Cryptography with Python - Ashiq KS - Medium. Medium. https://medium.com/@ashiqgiga07/asymmetric-cryptography-with-python-5eed86772731
66. GeeksforGeeks. (2020, September 22). SHA3 in Python. GeeksforGeeks. https://www.geeksforgeeks.org/sha3-in-python/

Task 2

Background

Blockchain technology is a decentralized and distributed ledger system designed to securely record, store, and protect data across a network of computers. Unlike traditional centralized systems that store data in a single location, blockchain distributes information across multiple nodes. Each block of data is securely linked to the previous one through cryptographic methods, forming an immutable chain that enhances transparency and prevents tampering. This distinctive structure makes blockchain a powerful solution for modern cybersecurity challenges.

A key feature of blockchain is its reliance on advanced cryptographic techniques to safeguard data. Every transaction or record added to the blockchain undergoes verification through consensus mechanisms like Proof of Work (PoW) or Proof of Stake (PoS), ensuring that only legitimate transactions are accepted. Once recorded, data cannot be altered unless a majority of network participants reach a consensus, making unauthorized modifications nearly impossible. This immutability is one of blockchain's most significant security advantages.

Additionally, blockchain enhances data privacy by allowing controlled access to sensitive information. While the ledger remains transparent and accessible to authorized users, encryption ensures the confidentiality of private data. This combination of transparency and security makes blockchain particularly valuable in areas such as financial transactions, healthcare records, and identity management.

Importance of Blockchain in Cybersecurity

The growing number of cyber threats such as ransomware attacks, identity theft, and data breaches has highlighted the need for robust security solutions. Traditional cybersecurity measures rely on centralized systems that are often prone to single points of failure. Blockchain technology has emerged as a transformative solution to the ongoing cybersecurity challenges. Blockchain's decentralization, transparency, and immutability provide powerful mechanisms for securing data, preventing tampering, and enhancing system resilience. The following are key aspects that highlight its importance in cybersecurity:

1. Enhanced data integrity and tamper resistance

Blockchain ensures that the data stored in its ledger remains unchanged. Each block is cryptographically linked to the previous block to form an unbroken chain. Any attempt to tamper with a block changes its encrypted hash, immediately exposing the tampering. This feature ensures that sensitive information is accurate and not tampered with, preventing unauthorized modifications and data leakage.

2. Decentralization: Eliminating Single Points of Failure

Traditional centralized systems are susceptible to cyberattacks, as compromising a single server can lead to catastrophic breaches. Blockchain's decentralized architecture distributes data across multiple nodes, making it highly resistant to attacks. Even if some nodes are attacked, the rest of the network remains operational and secure. This decentralization is particularly effective in protecting critical infrastructure such as financial systems, medical databases, and IoT networks.

3. Strong encryption and privacy mechanisms

Blockchain uses advanced encryption to ensure the security of transactions and data. End-to-end encryption ensures confidentiality, while permission networks allow access only to trusted parties. These features strike a balance between transparency and privacy, protecting sensitive information in industries such as finance and healthcare.

4. Fraud prevention and authentication

Blockchain provides secure identity management by storing credentials in an immutable ledger. Encryption keys verify user identity, reducing the risk of identity theft and fraud. In addition, blockchain supports multi-signature access control, which requires multiple verifications of administrative operations, thus further improving security.

5. Defending against cyber threats

Blockchain enhances system resilience by decentralizing infrastructure and implementing collaborative consensus algorithms to monitor malicious activities or anomalies in real time. This capability is critical to mitigate threats such as Distributed Denial of Service (DDoS) attacks or unauthorized access attempts.

Detailed discussion on how the security related technology works.

- Identify and discuss the components/technologies involved.
- Discuss current challenges and emerging solutions
- Explain the processes involved.
- Explain the threats that the security related technology can address. Discuss the examples of the security related technology usage.

Components/technologies involved in blockchain technology

Blockchain networks rely on several core components that ensure their secure, decentralized, and tamper-proof operations. Nodes are one of the fundamental elements of a blockchain, acting as individual computers that store and validate transaction data. There are different types of nodes, including full nodes, which maintain a complete copy of the blockchain and validate transactions, and partial nodes, also known as lightweight nodes, which store only hash values of transactions. Mining nodes, found in Proof of Work (PoW) systems, validate transactions and add new blocks through a process called mining. These nodes work together to maintain the blockchain's integrity and efficiency.

Another crucial component of blockchain networks is the distributed ledger, which records all transactions securely and immutably. Each block in the ledger contains a set of transactions, a timestamp, and a reference to the previous block, forming an unalterable chain of records. There are different types of ledgers, including public ledgers, which allow open access to transactions, distributed ledgers, where all nodes maintain a local copy of the database and collectively verify transactions, and decentralized ledgers, which operate without a central authority, ensuring transparency and trust in the network.

Transactions form the backbone of blockchain functionality, representing the transfer of value or data between participants. Each transaction contains details such as the sender's and receiver's addresses, the amount being transferred, and a digital signature to ensure authenticity. Before being recorded on the blockchain, transactions undergo verification by nodes to prevent fraud, such as double-spending. The security and consistency of transactions are maintained through consensus mechanisms, which allow network participants to agree on the validity of transactions. Different consensus algorithms are used depending on the blockchain type. Proof of Work (PoW) requires miners to solve complex mathematical puzzles to validate transactions, as seen in Bitcoin. Proof of Stake (PoS) selects validators based on the number of coins they hold and are willing to stake, reducing energy consumption compared to PoW. Another variation, Delegated Proof of Stake (DPoS), allows users to vote for trusted delegates who validate transactions on their behalf, improving scalability and transaction speed.

In addition to core components, supporting technologies enhance blockchain security, functionality, and usability. Cryptography is a critical element in securing blockchain transactions and maintaining privacy. Hash functions ensure that transaction data remains tamper-proof by converting information into a fixed-size hash value, which changes drastically if

even a small modification is made to the original data. Digital signatures play a vital role in authentication, as they verify the legitimacy of transactions using a sender's private key, which can be validated by others using the sender's public key. Public and private key pairs further enhance security by ensuring that only authorized individuals can access and sign transactions.

Smart contracts are another essential supporting component of blockchain networks. These self-executing contracts contain predefined terms written in code and automatically execute transactions when conditions are met. They eliminate the need for intermediaries, making processes more efficient and cost-effective. Smart contracts are widely used in industries such as decentralized finance (DeFi), supply chain management, and insurance claims, where automation is crucial. They help streamline processes, reduce human errors, and enhance transparency.

Tokens are also integral to blockchain ecosystems, representing various digital assets or utilities. Utility tokens grant users access to specific services within a blockchain platform, such as Ethereum's Ether, which powers decentralized applications and smart contracts. Security tokens represent ownership in real-world assets like stocks, real estate, or commodities, often requiring regulatory compliance. Stablecoins, on the other hand, maintain a stable value by being pegged to fiat currencies like the US dollar, making them useful for transactions that require price stability.

Challenges and solutions

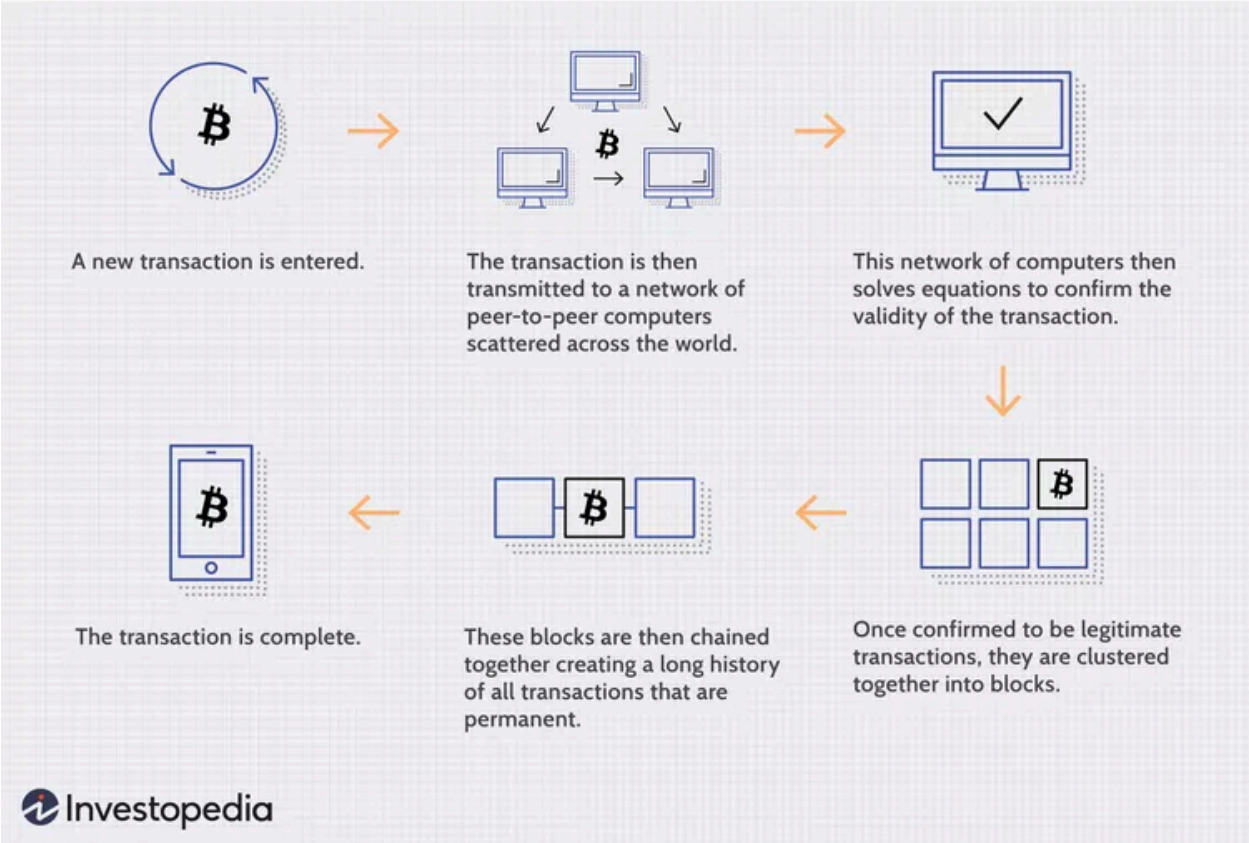
Blockchain technology faces several challenges that hinder its widespread adoption, but emerging solutions are addressing these issues. One major challenge is the **lack of trust among users**, as organizations are hesitant to rely on blockchain due to security concerns and the absence of regulatory clarity. Platforms like **TradeLens**, developed by Maersk and IBM, aim to build trust by implementing **permissioned blockchains** where participants are verified and transactions are immutable, private, and traceable. Another significant barrier is the **lack of financial resources**, as many organizations struggle to justify the high costs of blockchain implementation. However, as awareness of blockchain's benefits grows, companies are developing stronger business cases to highlight the long-term advantages of adoption. Additionally, blockchain's **high energy consumption** poses a major concern, especially with Proof-of-Work (PoW) systems that require vast computing power. To address this, energy-efficient alternatives like **Proof-of-Stake (PoS)** are being developed, reducing power consumption by assigning validation tasks based on participants' stakes rather than computational effort. Furthermore, **blockchain interoperability** remains a challenge, as different blockchain networks use independent protocols, limiting seamless communication between systems. Solutions like **Ark's SmartBridges** and **Cosmos' Interblockchain Communication (IBC) protocol** aim to enable cross-blockchain communication, allowing different platforms to exchange data efficiently. By tackling these challenges with innovative solutions, blockchain technology continues to evolve, paving the way for greater adoption across industries.

Processes involved in blockchain technology

The process of blockchain technology begins with its unique structure as a distributed database, where multiple copies of data are stored across different machines, ensuring consistency and security. Unlike traditional databases, blockchain structures data in blocks, which are linked together to form an immutable chain. When transactions occur, they are collected and stored in a block. Once a block reaches its maximum size—such as the 4MB limit in Bitcoin—it is processed through a cryptographic hash function, generating a block header hash. This hash is then embedded in the next block's header, securing the linkage between blocks and maintaining the integrity of the blockchain.

The transaction process varies depending on the blockchain but generally follows a structured sequence. In Bitcoin's blockchain, when a user initiates a transaction via a cryptocurrency wallet, the transaction is first sent to a memory pool, where it awaits validation. Miners pick up transactions from this pool and compile them into a block. Once a block is filled with transactions, mining begins. Each node in the network attempts to solve a cryptographic puzzle using a "nonce" (a randomly generated number). The nonce is repeatedly adjusted until a valid hash is found that meets the network's difficulty target. This process continues until a miner successfully generates a valid hash, at which point they add the block to the blockchain and receive a reward.

Once a block is added to the blockchain, transactions are not immediately confirmed. The network requires multiple additional blocks (typically five in Bitcoin) to be added before the transaction is considered fully validated. This process, taking approximately one hour in Bitcoin's network, ensures security and prevents double-spending. However, other blockchains, such as Ethereum, use alternative validation mechanisms like Proof of Stake (PoS), where validators are randomly chosen from those who have staked cryptocurrency. This method significantly reduces energy consumption and speeds up transaction processing compared to Bitcoin's Proof of Work (PoW) model. Regardless of the specific consensus mechanism used, all blockchain networks follow a structured process to ensure decentralization, security, and immutability.



Threats and example of security of usage of blockchain technology

Blockchain technology, despite its decentralized and secure design, is vulnerable to various cyber threats that can compromise its reliability and functionality. One of the most common threats is **phishing attacks**, where attackers use deceptive emails or fake websites to trick users into revealing sensitive information such as private keys or login credentials. Once fraudsters gain access to a user's blockchain wallet, they can steal funds or manipulate transactions. Another major threat is **routing attacks**, which occur when hackers intercept data as it moves between blockchain nodes or internet service providers. Since blockchain networks rely on real-time data transfers, attackers can secretly extract confidential information or even delay transactions, making the system appear normal while data is being compromised in the background.

Another significant risk is **Sybil attacks**, where an attacker creates multiple fake identities within a blockchain network to gain disproportionate influence. By flooding the network with these fraudulent nodes, they can manipulate voting processes, disrupt operations, or reduce network performance, potentially causing instability. **51% attacks** pose an even greater danger, especially for public blockchains. In this scenario, a single miner or a group of miners gains control over more than half of the network's computational power. With this control, they can rewrite transaction history, perform double-spending attacks, or prevent new transactions from being confirmed. Although private blockchains are generally not vulnerable to 51% attacks due to restricted access and centralized authority, public blockchain networks must implement strong security measures to mitigate these risks.

Blockchain security is widely applied across various industries to enhance data protection, prevent fraud, and ensure transparency. In **financial services**, blockchain technology secures transactions by providing a transparent and immutable ledger, reducing the risk of fraud, double-spending, and unauthorized alterations. Banks and financial institutions utilize blockchain to enhance the security and efficiency of transactions, minimizing errors and increasing trust among stakeholders. In **healthcare**, blockchain plays a crucial role in safeguarding patient records by ensuring their integrity, confidentiality, and accessibility. Medical institutions use blockchain to store and share sensitive health data securely, preventing unauthorized access while improving collaboration between healthcare providers. This helps maintain data privacy while streamlining patient care and medical research. In **supply chain management**, blockchain enhances traceability and accountability by providing a tamper-proof record of product movement. Companies use blockchain to verify the authenticity of goods, combat counterfeiting, and ensure ethical sourcing. By integrating blockchain into supply chains, businesses can track the entire lifecycle of products, from manufacturing to delivery, ensuring transparency and compliance with industry standards. These applications demonstrate how blockchain security strengthens data protection, prevents fraud, and promotes trust in various industries.

Discussion on the impact

Blockchain technology has revolutionized cybersecurity by offering enhanced security, transparency, and decentralization. Its implementation has had significant impacts on various industries, addressing key security concerns while also presenting certain limitations. The following discussion explores the benefits, challenges, and future potentials of blockchain technology in cybersecurity.

1. Benefits of Blockchain in Cybersecurity

Enhanced Data Security and Integrity

One of the most significant advantages of blockchain technology is its ability to provide strong data security and integrity. Due to its immutable nature, once data is recorded in a blockchain ledger, it cannot be altered without network consensus. This characteristic ensures that sensitive information remains untampered, reducing risks associated with data breaches and unauthorized modifications (Zheng et al., 2017). This feature is particularly valuable for industries dealing with highly sensitive data, such as healthcare, finance, and government institutions.

Decentralization Reducing Single Points of Failure

Traditional centralized systems often suffer from a single point of failure, making them attractive targets for cyberattacks. Blockchain's decentralized structure distributes data across multiple nodes, making it difficult for hackers to manipulate or compromise the system. Even if a few nodes are attacked, the rest of the network remains functional, ensuring data security and continuity. This decentralization significantly enhances resilience against cyber threats such as Distributed Denial of Service (DDoS) attacks and ransomware intrusions (Conti et al., 2018).

Improved Authentication and Identity Management

Blockchain technology facilitates secure identity verification through cryptographic authentication mechanisms. By leveraging blockchain for identity management, organizations can reduce identity theft and fraud, as personal data is stored securely and accessed only with authorized cryptographic keys (Zyskind et al., 2015). This feature has been widely adopted in financial services and healthcare industries to enhance security. Additionally, the implementation of self-sovereign identities allows users to maintain control over their personal information, reducing reliance on centralized authorities.

Transparency and Trust

The transparent nature of blockchain ensures that all transactions and modifications are traceable, increasing trust among stakeholders. This feature is particularly useful in supply chain management and financial transactions, where audit trails help verify the authenticity of transactions (Casino et al., 2019). Transparency also aids in regulatory compliance, as blockchain records can be used to demonstrate adherence to security protocols and prevent fraudulent activities.

2. Limitations of Blockchain in Cybersecurity

Scalability Issues

Blockchain networks, particularly those using Proof of Work (PoW) consensus mechanisms, face scalability challenges. Processing a large number of transactions requires significant computational power and time, leading to potential delays and high energy consumption (Khan & Salah, 2018). This limitation hinders its widespread adoption in high-speed transaction environments. Layer 2 solutions, such as sidechains and sharding, are being explored to enhance blockchain scalability while maintaining security.

Regulatory and Compliance Challenges

Since blockchain operates on a decentralized and global scale, regulatory challenges arise in terms of data protection laws, compliance, and jurisdictional issues. Different countries have varying regulations regarding data privacy, which may conflict with the decentralized nature of blockchain (Finck, 2018). Organizations implementing blockchain must ensure compliance with laws such as the General Data Protection Regulation (GDPR) and other regional policies.

Energy Consumption

The high computational requirements of blockchain networks, particularly PoW-based systems, contribute to significant energy consumption. Bitcoin mining, for example, requires vast amounts of electricity, raising environmental concerns. More energy-efficient consensus mechanisms like Proof of Stake (PoS) and hybrid models are being explored to mitigate this issue (Sedlmeir et al., 2020). Efforts are also underway to develop green blockchain solutions that reduce carbon footprints.

3. Future Potentials of Blockchain in Cybersecurity

Integration with Artificial Intelligence (AI) and IoT

The integration of blockchain with AI and the Internet of Things (IoT) can enhance security measures. AI-driven blockchain security models can detect anomalies and potential cyber threats in real-time, while blockchain can secure IoT devices against cyberattacks (Singh et al., 2021). This combination can lead to more autonomous cybersecurity frameworks capable of mitigating risks before they escalate.

Quantum-Resistant Cryptography

With the rise of quantum computing, traditional cryptographic methods could become obsolete. Future blockchain innovations aim to develop quantum-resistant encryption techniques to ensure continued security against quantum threats (Fernández-Caramés & Fraga-Lamas, 2020). Researchers are exploring lattice-based cryptography and other post-quantum encryption algorithms to fortify blockchain networks.

Adoption in Government and Public Services

Governments worldwide are exploring blockchain applications for securing digital identities, voting systems, and public records. Its adoption in public services can improve transparency, efficiency, and security, reducing fraud and corruption (Atzori, 2017). Blockchain-powered

e-governance platforms can enhance public trust by providing tamper-proof records and digital verification systems.

Evaluate the effectiveness of current approaches and propose improvements

1. Proof-of-Stake (PoS) and Delegated Proof-of-Stake (DPoS)

PoS replaces energy-intensive mining with token staking, reducing energy consumption by up to 99% compared to PoW. Ethereum's transition to PoS is a prime example of this shift, significantly cutting its energy usage. DPoS further optimizes efficiency by delegating validation tasks to elected nodes, minimizing computational overhead while maintaining a degree of decentralization. Platforms like EOS and Tron leverage DPoS to achieve high throughput with minimal energy expenditure. However, DPoS poses centralization risks, as a small group of entities controlling a significant portion of staked tokens or delegated nodes could undermine decentralization.

2. Proof-of-Authority (PoA)

PoA eliminates mining entirely by relying on pre-approved validators, reducing energy consumption to levels comparable to traditional databases. Projects like Fnality utilize PoA to secure their networks efficiently. However, PoA is primarily suited for permissioned or private blockchains due to its dependence on trusted validators, limiting its applicability in fully decentralized public networks.

3. Practical Byzantine Fault Tolerance (PBFT) and Its Variants

PBFT-based algorithms, such as S-PBFT, are optimized for specific applications. S-PBFT reduces latency and energy consumption by 30–40% compared to traditional PBFT through node partitioning and dual-layer validation mechanisms. This makes it particularly suitable for financial and consortium blockchains. However, PBFT still faces scalability challenges in large decentralized networks, where achieving consensus among numerous nodes remains complex.

4. Hybrid and Niche Consensus Mechanisms

Emerging mechanisms explore the integration of green energy with consensus algorithms. For example, Green-PoW modifies PoW by utilizing renewable energy for mining and adjusting difficulty dynamically, improving energy efficiency while maintaining security. Additionally, Raft and Federated Byzantine Agreement (FBA) excel in private networks, simplifying consensus in trusted environments and reducing energy waste.

Key Challenges in Current Consensus Approaches

1. Scalability vs. Efficiency Trade-offs
While PoS and DPoS improve energy efficiency, they still struggle with transaction throughput in large-scale networks.

2. Hardware Limitations
Even energy-efficient consensus mechanisms depend on high-performance hardware for optimal execution, creating barriers to entry.

3. Centralization Risks
DPoS and PoA enhance efficiency but often sacrifice decentralization, potentially undermining the core principles of blockchain.

4. Renewable Energy Integration Challenges
Although some projects experiment with solar or wind-powered mining, many blockchain networks lack the necessary infrastructure to fully leverage renewable energy, limiting sustainability gains.

Proposed Improvements

1. Adopt Adaptive Hybrid Consensus Models
Combining PoS/PBFT with sharding or off-chain solutions can enhance scalability while reducing on-chain computational load. For instance, a hybrid PoS-PBFT model could delegate routine transactions to PoS validators while reserving PBFT for high-security tasks, achieving an optimal balance between performance and energy efficiency.

2. Promote Decentralized Renewable Energy Integration
Encouraging validators and miners to use renewable energy through token rewards or carbon credit incentives can foster sustainable blockchain operations. Additionally, developing "green nodes" powered entirely by solar or wind energy, as explored in experimental blockchain projects, can further reduce environmental impact.

3. Research Quantum-Resistant Algorithms
With the advancement of quantum computing, traditional cryptographic algorithms may become vulnerable. Investing in post-quantum cryptographic consensus mechanisms can future-proof blockchain security while maintaining energy efficiency.

4. Hardware-Software Co-Design
Optimizing ASICs for low-power consensus algorithms can reduce reliance on general-purpose hardware. Furthermore, implementing lightweight clients and data compression techniques can lower storage and processing demands, enhancing overall energy efficiency.

5. Governance and Standardization

Establishing interoperability standards can unify energy-efficient consensus mechanisms across different blockchain networks. Additionally, introducing decentralized governance models in DPoS systems can mitigate centralization risks and enhance system fairness and security.

6. Sector-Specific Customization

Tailoring consensus algorithms for specific industries, such as finance or supply chain management, can balance efficiency and security. For instance, S-PBFT can be further optimized for environments that require both high efficiency and strong security, ensuring the best trade-off for industry-specific needs.

Reference

1. IBM. (2021, August 4). *What is blockchain security?*. Retrieved March 12, 2025, from *https://www.ibm.com/topics/blockchain-security*
2. Kaspersky. (2024, December 12). *What is blockchain security? | Is blockchain safe?*. Retrieved March 12, 2025, from *https://www.kaspersky.com/resource-center/definitions/what-is-blockchain-security*
3. PixelPlex. (2025, March 6). *Blockchain cybersecurity: Use cases, benefits & challenges*. Retrieved March 12, 2025, from *https://pixelplex.io/blog/blockchain-cybersecurity/*
4. SecureWorld. (n.d.). *Blockchain beyond cryptocurrency: Enhancing cybersecurity*. Retrieved March 12, 2025, from *https://www.secureworld.io/industry-news/blockchain-beyond-crypto-cybersecurity*
5. UpGuard. (n.d.). *The role of cybersecurity in blockchain technology*. Retrieved March 12, 2025, from *https://www.upguard.com/blog/the-role-of-cybersecurity-in-blockchain-technology*
6. Infosys. (n.d.). *Rethinking cybersecurity through blockchain*. Retrieved March 12, 2025, from https://www.infosys.com/insights/cyber-security/cybersecurity-blockchain.html
7. Chainalysis. (2023). *The importance of blockchain security*. Retrieved March 12, 2025, from *https://www.chainalysis.com/blog/blockchain-security*
8. CM-Alliance. (2023, July 20). *Blockchain technology: Enhancing security in the digital age*.*https://www.cm-alliance.com/cybersecurity-blog/blockchain-technology-enhancing-security-in-the-digital-age*
9. Investopedia. (2024). *Blockchain Facts: What Is It, How It Works, and How It Can Be Used*. *https://www.investopedia.com/terms/b/blockchain.asp*
10. Nature. (09 April 2024). A secure and highly efficient blockchain PBFT consensus algorithm for microgrid power trading. *https://www.nature.com/articles/s41598-024-58505-w*
11. IJRASET. (29 May 2024). Sustainable Blockchain: A New Horizon for Energy-Efficient Consensus Mechanisms. *https://www.ijraset.com/research-paper/new-horizon-for-energy-efficient-consensus-mechanisms*
12. RAPID INNOVATION. (2025). Eco-Friendly Blockchain: Green Innovations and Shaping Environmental Futures in 2024. *https://www.rapidinnovation.io/post/sustainable-blockchain-green-innovations-environmental-impact-2024*
13. Atzori, M. (2017). Blockchain technology and decentralized governance: Is the state still necessary? *Journal of Governance and Regulation*. Retrieved from https://www.virtusinterpress.org/Blockchain-technology-and.html
14. Casino, F., Dasaklis, T. K., & Patsakis, C. (2019). A systematic literature review of blockchain-based applications: Current status, classification, and open issues. *Telematics and Informatics*. Retrieved from https://www.sciencedirect.com/science/article/abs/pii/S0736585318304941
15. Conti, M., Kumar, E. S., Lal, C., & Ruj, S. (2018). A survey on security and privacy issues of blockchain technology. *Future Generation Computer Systems*. Retrieved from https://www.sciencedirect.com/science/article/abs/pii/S0167739X18300274

16. Fernández-Caramés, T. M., & Fraga-Lamas, P. (2020). Towards post-quantum blockchain: A review on blockchain cryptography resistant to quantum computing attacks. Retrieved from https://ieeexplore.ieee.org/document/8968985
17. Finck, M. (2018). Blockchains and data protection in the European Union. *European Law Journal*. Retrieved from https://onlinelibrary.wiley.com/doi/abs/10.1111/eulj.12234
18. Singh, S., Chatterjee, S., & Ruj, S. (2021). Blockchain and AI: Opportunities and challenges. *IEEE Internet of Things Journal*. Retrieved from https://ieeexplore.ieee.org/document/9207703
19. Zheng, Z., Xie, S., Dai, H., Chen, X., & Wang, H. (2017). An overview of blockchain technology: Architecture, consensus, and future trends. *Proceedings of IEEE* https://ieeexplore.ieee.org/document/8029379*International Congress on Big Data*. Retrieved from
20. Ibm. (2025, February 13). *What is Blockchain Security?*. IBM. https://www.ibm.com/think/topics/blockchain-security#:~:text=Routing%20attacks,threat%2C%20so%20everything%20looks%20normal
21. *Blockchain security: Key concepts, threats, and future trends*. Sangfor Technologies. (n.d.). https://www.sangfor.com/glossary/cybersecurity/blockchain-security-key-concepts-threats-and-future-trends
22. Admin. (2024, October 2). *Top 8 challenges of Blockchain Adoption & Their Solutions*. Parangat Technologies. https://www.parangat.com/top-8-challenges-of-blockchain-adoption-their-solutions/
23. GeeksforGeeks. (2024, October 22). *Components of Blockchain Network*. https://www.geeksforgeeks.org/components-of-blockchain-network/

Appendix

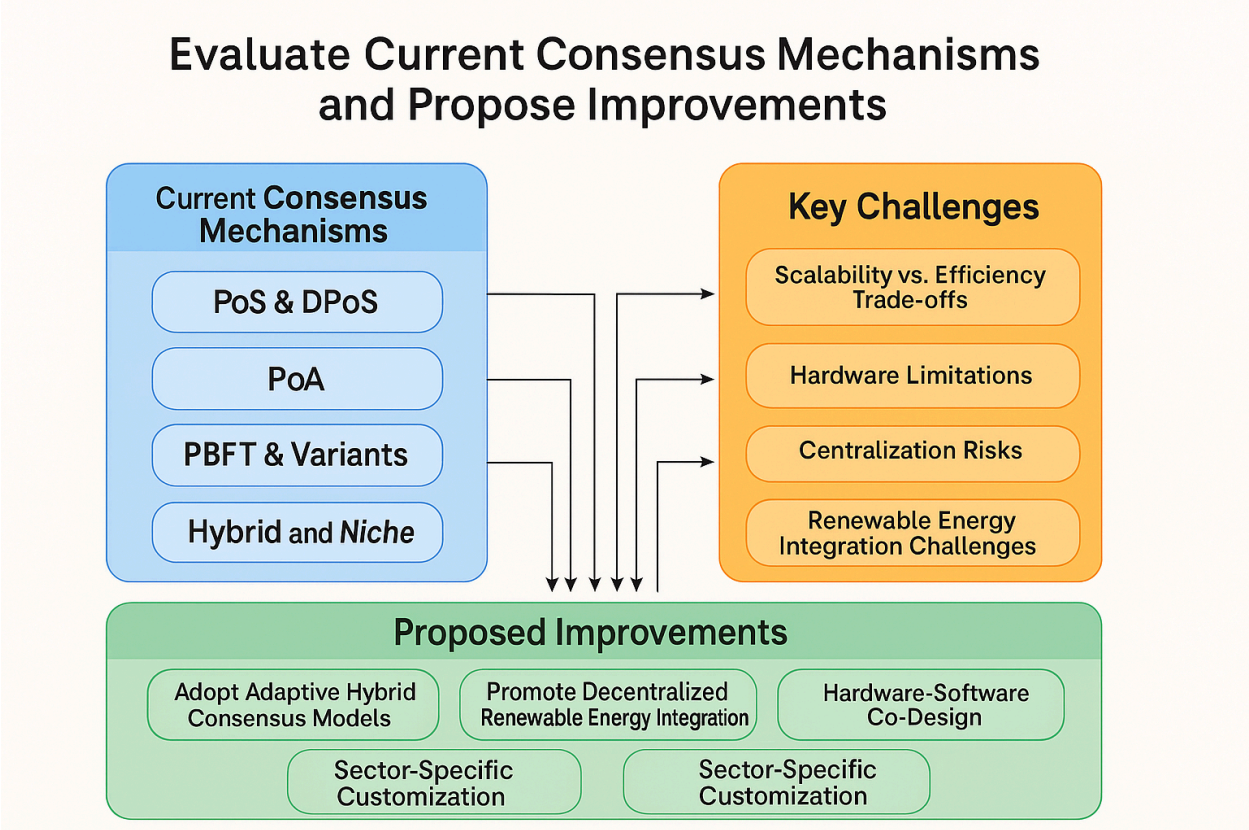


Figure 1 : Evaluate Current Consensus Mechanisms and Propose Improvements

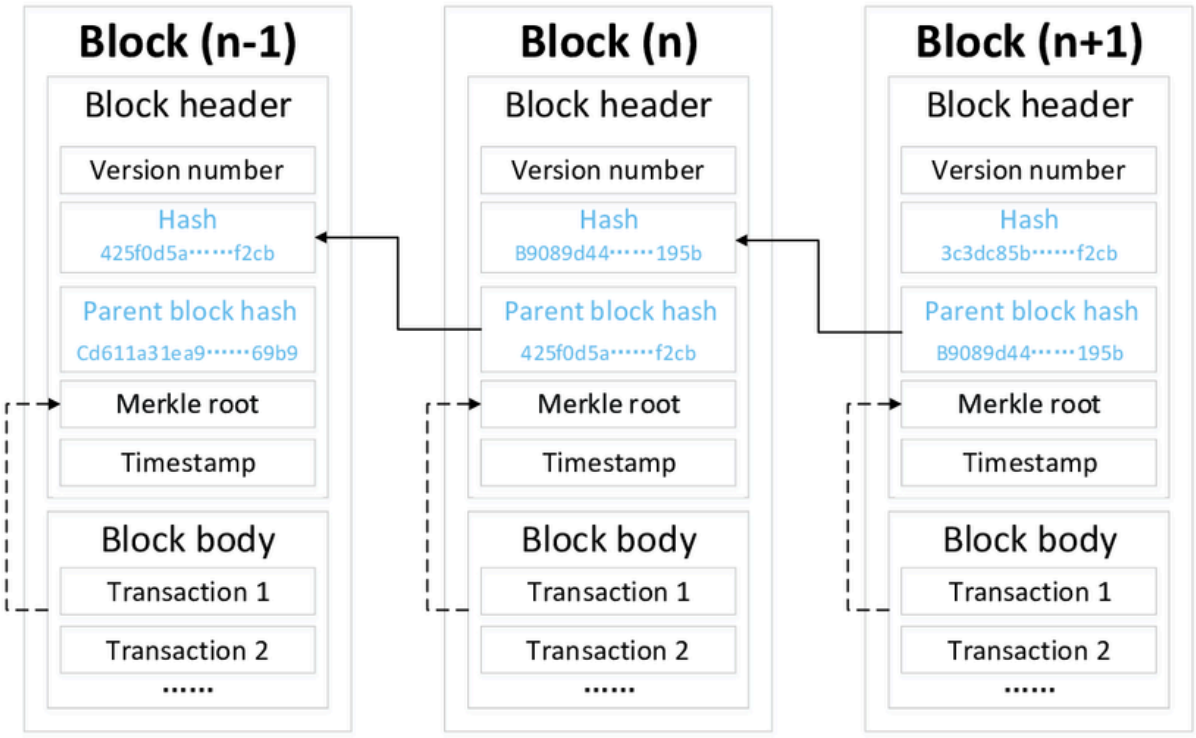


Figure 2: Blockchain Structure Diagram

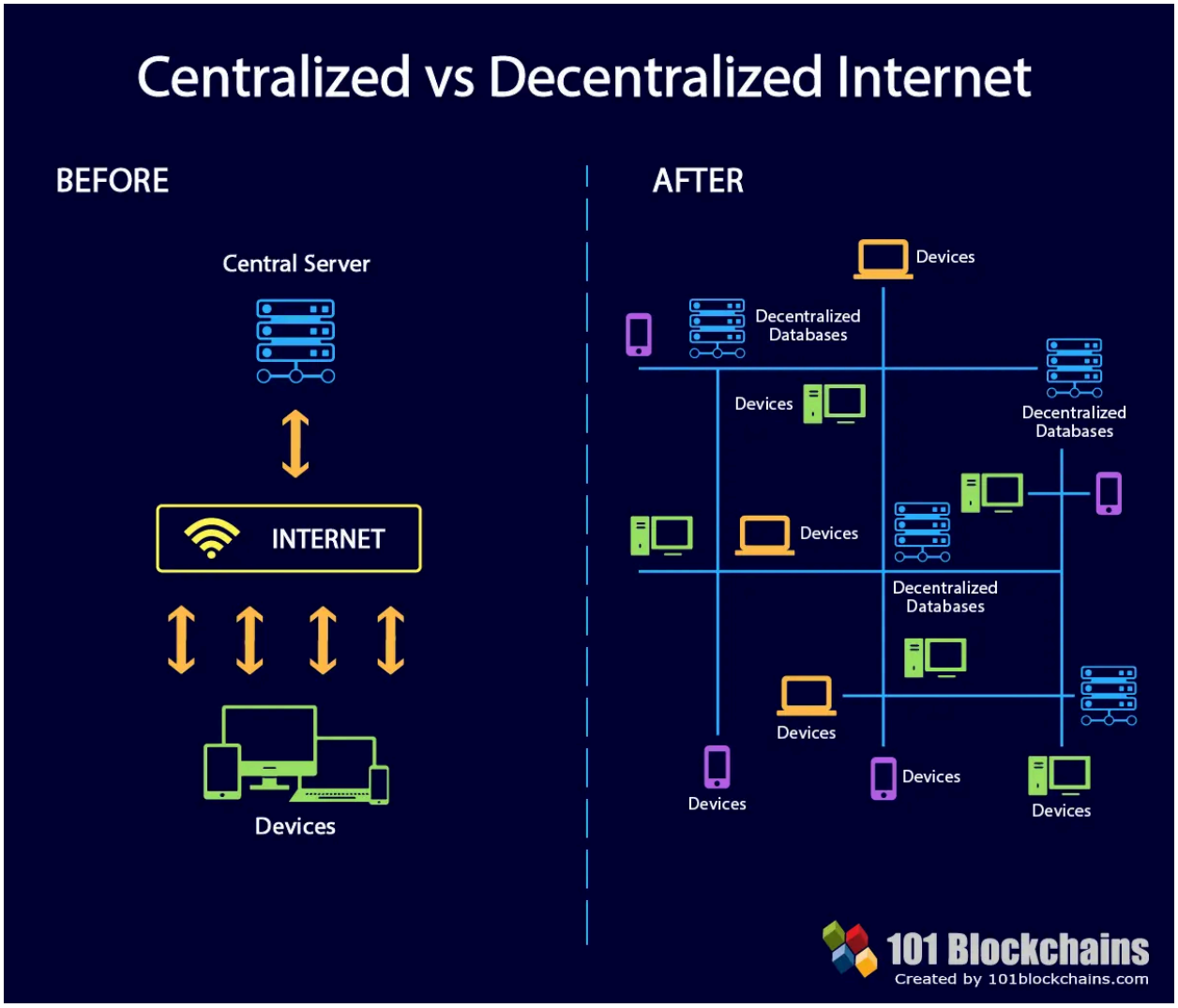


Figure 3: Centralized vs. Decentralized Systems