Research Journal of Computer Systems and Engineering (RJCSE)
Volume 4 Issue 1 (2023) | Pages: 74 – 83 | e-ISSN: 2230-8571; p-ISSN: 2230-8563
https://doi.org/10.52710/rjcse.65

Empowering Edge Security An Advanced Encryption Strategies for Enhanced Cyber Resilience in Edge Computing

Received: 15 March 2023; Revised: 12 May 2023; Accepted: 12 June 2023

Nadica Stojanovic
University of Kragujevac,
Faculty of Engineering, 6 Sestre Janjić STR.,
34000 Kragujevac, Serbia
https://orcid.org/0000-0002-4199-0587

Ritika Dhabliya
Director, Yashika Journal Publications Pvt. Limited, Wardha, Maharashtra, India
Email: ritikadhabalia@gmail.com

Abstract

Edge computing is quickly becoming popular in many fields because it promises lower delay and better performance. There are, however, serious security worries about the growing number of edge devices, especially when it comes to the safety and accuracy of data. This paper suggests using more complex encryption methods to make edge computing settings safer online. Traditional encryption methods, like AES (Advanced Encryption Standard), work, but they might not be enough to keep things safe in edge settings that change a lot. It is important to have lightweight encryption methods that can balance security and speed because edge devices often work in settings with limited resources. Hybrid encryption, which combines symmetric and asymmetric methods, is one way to do this. This makes it possible to secure data quickly with symmetric keys, while asymmetric keys are used to handle and trade keys, which improves security overall. It is also possible to use temporary key sharing methods to make the system even safer. Keeping track of encryption keys is another important part. Traditional ways of managing keys might not work well in edge computing, where devices are spread out and often move around. So, the study suggests using decentralized key management systems (DKMS) to make it safe to create, share, and remove encryption keys in edge settings. The study also talks about how safe startup and authentication methods are important for making sure that edge devices are trustworthy. Potential security risks can be lessened by making sure that devices are real and functioning properly before letting them connect to the network. The advanced encryption strategies that have been suggested, such as hybrid encryption, autonomous key management, and safe startup methods, can make edge computing settings much more secure online.

Keywords

Edge Computing, Encryption Strategies, Cyber Resilience, Hybrid Encryption, Decentralized Key Management

1. Introduction

As the number of IoT devices grows quickly, so does the amount of data that needs to be sent. This makes things very hard for standard IoT services. In most IoT setups, devices have to send data to cloud services so that it can be processed, which adds a lot of extra work to the transfer process. Even [1] though the cloud takes over computer jobs that used to be done by devices, the sheer amount of data that is sent through it causes network traffic problems and delay problems. The number of IoT-connected gadgets hit 11.2 billion in 2018. That number is expected to rise to 20 billion by 2020. The [2] exponential growth of IoT devices puts more stress on network speed, which isn't growing at the same rate. As a result, network speed has become a major problem for standard IoT services, making it harder to send and process data quickly. New ideas, [3] like edge computing, have come about to deal with these problems. Edge computing brings processing closer to the source of the data. This means that data doesn't have to travel as far to get to cloud computers. Edge computing can help networks handle more data by handling data closer to where it is created. This can make IoT services faster and less sensitive to delay. Also, technologies like 5G networks offer more speed and less delay, which could help IoT settings with network congestion even more [4]. But putting these technologies to use on a large scale and making sure they work with current systems are still big problems.

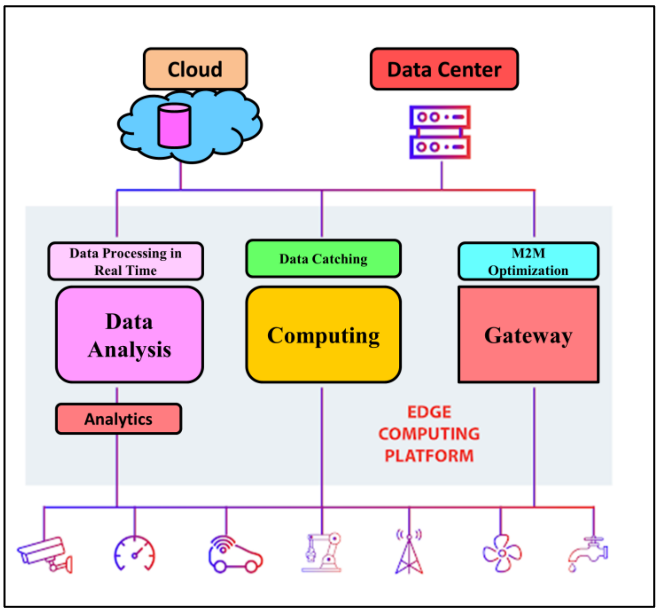

Figure 1: Overview of Component in Edge computing

Edge computing [5] is a new way of thinking about computing that moves computer power to the edge of the network, closer to where data is created and used. Edge computing has big benefits like lower latency, better bandwidth efficiency, and better real-time processing because it uses the fact that data sources are close by. But because edge computing is spread out, it brings new security challenges, especially when it comes to keeping data private, secure, and accessible. For edge computing operations to work, people must be able to prepare for, react to, and recover from cyberattacks. This is called cyber resilience. Encryption techniques that are more advanced are very important for making cyberspace safer because they protect data in edge settings. Even though traditional encryption methods like the Advanced Encryption Standard (AES) work, they might not be the best fit for edge devices because they are always changing and have limited resources. In this study, we suggest advanced encryption methods that are specifically designed to work in edge computing settings. One approach is hybrid encryption, which uses both symmetric and asymmetric algorithms to make security and speed work well together. Hybrid encryption is a strong way to protect edge devices because it uses symmetric keys to secure data efficiently and asymmetric keys to handle and trade keys [6].

To make edge computing [7] systems even safer, it is suggested that they use temporary key sharing methods along with mixed encryption. The safe creation, sharing, and removal of encryption keys is made possible by these methods. This keeps [8] data safe even as security threats change. Decentralized key management systems (DKMS) are also being considered as a way to deal with the problems that come up with managing keys in edge computing. The risk of a single point of failure is lower in DKMS because it spreads out the creation and sharing of encryption keys. This makes edge settings safer generally. The main goal of this study is to show how advanced encryption techniques can improve edge security and make edge computing more resilient to cyberattacks. Companies can lower security risks and use edge computing to its fullest for their applications by using these tactics.

2. Related work

Several works that are connected to this one focus on improving cyber resistance by making edges safer using advanced encryption techniques. [9] did one of these studies and looks into how lightweight encryption methods can be used in edge computing zones. The writers suggest a light encryption method based on the Lightweight MISTY1 algorithm. This method strikes a good mix between security and speed for edge devices that don't have a lot of resources. The study shows that the suggested method can ensure safe data transfer in edge computing settings without adding a lot of extra work to the computers.

The study [10] wrote a similar paper that talks about how homomorphic encryption can be used in edge computing to keep data handling safe. Homomorphic encryption lets you work on protected data without decrypting it, which keeps your data private. The writers suggest a homomorphic encryption method that works well in edge settings and keeps data processing safe while using as few computing resources as possible. The research shows that homomorphic encryption can be used in edge computing

Research Journal of Computer Systems and Engineering (RJCSE)
Volume 4 Issue 1 (2023) | Pages: 74 – 83 | e-ISSN:2230-8571; p-ISSN: 2230-8563
https://doi.org/10.52710/rjcse.65

for jobs like collecting and analyzing data. [11] study isn't just about encryption methods; it's also about safe key management in edge computing settings. The writers suggest a decentralized key management system (DKMS) that uses blockchain technology to keep encrypted keys safe in edge settings. The DKMS makes sure that encryption keys are made, shared, and taken away in a safe and decentralized way. This makes edge computing systems safer overall.

The [12] also talks about the security problems that edge computing can cause in industrial IoT (IIoT). The writers suggest a safe way for IIoT systems to collect data. It uses simple encryption methods to make sure that data is sent and processed safely at the edge. In IIoT settings, the study shows that the suggested method can protect private data well while keeping low delay and high speed. Overall, these linked works show how important advanced encryption methods are for making edges safer and more resilient in the online world. It is possible for edge computing settings to keep data transfer and processing safe while reducing computer waste and delay by using homomorphic encryption, lightweight encryption methods, autonomous key management systems, and secure data aggregation schemes.

Table I: Related Work Summary

IoT Service	Application	Finding	Advantage
Edge Computing	Lightweight Encryption Algorithms	Balances security and performance for edge devices	Efficient data transmission
Edge Computing	Homomorphic Encryption	Secure data processing without decryption	Preserves data privacy
Edge Computing	Decentralized Key Management System	Securely manages encryption keys in edge environments	Distributed and secure key management
Industrial IoT (IIoT)	Secure Data Aggregation Scheme	Secures data transmission and processing at the edge	Protects sensitive data in IIoT environments
Smart Agriculture	Encrypted Sensor Data Transmission	Ensures data integrity and confidentiality	Secure and reliable data transmission
Smart Healthcare	Privacy-Preserving Data Sharing	Facilitates secure sharing of sensitive health data	Protects patient privacy
Smart Grid	Blockchain-based Energy Trading	Secures energy transactions and data integrity	Transparent and tamper-proof transactions
Smart Cities	Secure IoT Device Management	Ensures secure provisioning and management of IoT devices	Minimizes security vulnerabilities

3. Advanced Encryption Strategies

Hybrid encryption is a strong method that blends the best features of symmetric and asymmetric encryption algorithms to make things safer and faster. A symmetric encryption algorithm is used to protect the data in hybrid encryption, while an asymmetric encryption algorithm is used to protect the symmetric key that is used for encryption. As a result of this method, data is encrypted quickly and securely with symmetric encryption, and keys are managed safely and securely with asymmetric encryption. One of the best things about mixed encryption is that it can protect information over networks that you don't trust. An asymmetric method can be used to encrypt the symmetric key. This makes it safe for two people to share the symmetric key without a pre-shared key. This lets people who have never traded keys before talk to each other safely. It's perfect for private texting and online transactions, for example. One more benefit of mixed cryptography is that it works well. It is best to encrypt a lot of data at once with symmetric encryption algorithms because they are faster than asymmetric algorithms. Asymmetric encryption is used to handle keys and symmetric encryption is used to secure data. This makes hybrid encryption a good choice for both security and speed.

A. Hybrid Encryption

1. Symmetric and Asymmetric Algorithms

In hybrid encryption, symmetric and asymmetric encryption methods each do their own thing to make it safer and more efficient. When mixed encryption is used, symmetric encryption methods, like AES, are used to protect the data itself. Because these methods work quickly and efficiently, they are great for encrypting huge amounts of data. But symmetric encryption needs a shared secret key for both encrypting and decrypting. This makes it hard to keep track of keys, especially in edge computing, which is dynamic and spread out.

Asymmetric encryption methods, like RSA, are used to encrypt the symmetric key in mixed encryption. In asymmetric encryption, there are two keys: a public key and a secret key. This is different from symmetric encryption. When you want to decrypt something, you use the private key instead of the public key. This lets you swap keys safely without having to share a secret key first. However, asymmetric encryption is slower and less effective than symmetric encryption. This means that it is not as good for securing big amounts of data.

Hybrid encryption takes the best parts of both symmetric and asymmetric encryption methods and uses them together. Asymmetric encryption lets you securely share and handle keys, while symmetric encryption encrypts data quickly and efficiently. This mix makes it possible for a safe and effective encryption method that works well in edge computing settings.

B. Ephemeral Key Exchange Protocols

Ephemeral key exchange protocols are a type of cryptography that lets two people safely discuss and set up a shared secret key over an unprotected route of communication. These procedures are necessary to make sure that the shared key stays secret even if the way of communicating is broken.

DHE, or Diffie-Hellman key exchange, is one of the best known temporary key exchange systems. Alice and Bob agree on a big prime number, p, and a basic root modulo p, g. This is how the standard Diffie-Hellman system works. Next, each side makes a private key, which is a for Alice and b for Bob. They then use p, g, and their private key to make a public key. The public keys are sent back and forth, and then each person can use their own private key and the received public key to figure out the shared secret key.

Here's how to figure out the shared secret key:

$$Shared\ secret\ key = gabmodp$$

After that, this shared secret key can be used as the symmetric key to send and receive encrypted data between Alice and Bob

C. Decentralized Key Management Systems

Decentralized key management systems (DKMS) are computer programs that make it safe to create, share, and control encryption keys without relying on one central location. In DKMS, there is no single body that makes or distributes encryption keys. Instead, a network of servers does this job. The use of blockchain technology is one way to put DKMS into action. A blockchain-based DKMS stores encryption keys on a blockchain, which is a shared ledger that can't be changed. A consensus method is used to make a new encryption key and send it to all the nodes in the network when one is needed. This keeps the process of making keys and giving them out safe and clear.

$$KDKMS = Encrypt(Kdata, KDKMS)$$

4. Secure Bootstrapping and Attestation Mechanisms

A. Importance of Secure Bootstrapping

In edge computing, secure bootstrapping is a key step that makes sure devices are real and intact when they connect to a network. This includes making sure the device is who it says it is and setting up a safe way for the device to talk to the network. It is important to use secure bootstrapping to keep unwanted devices from connecting to the network and to protect against security risks like phishing and man-in-the-middle attacks. An easy way to make sure that devices are who they say they are is to use pre-shared keys (PSKs) or certificates. The device and the network share PSKs. Certificates, on the other hand, are given out by a known certificate authority (CA) and are used to make sure the device is who it says it is. These steps help make sure that only known devices can connect to the network,

Research Journal of Computer Systems and Engineering (RJCSE)
Volume 4 Issue 1 (2023) | Pages: 74 – 83 | e-ISSN:2230-8571; p-ISSN: 2230-8563
https://doi.org/10.52710/rjcse.65

which lowers the chance of someone getting in without permission. Also, secure booting is important for getting devices on the network to trust each other. During the starting process, devices' identities can be checked to build trust. This lets devices communicate and work together safely. This is very important in edge computing settings where devices might be far away or in places that you don't trust.

Algorithm:

1. Key Generation: Each device generates a pair of cryptographic keys: a private key (SK) and a corresponding public key (PK).

$$SK\_Device, PK\_Device$$

2. Certificate Issuance: A trusted certificate authority (CA) issues a certificate for the device, binding its public key to its identity.

$$Certificate\_Device = IssueCertificate(PK\_Device, Identity\_Device)$$

3. Certificate Verification: When the device attempts to join the network, the network verifies the device's certificate using the CA's public key.

$$VerifyCertificate(Certificate\_Device, CA\_PublicKey)$$

4. Challenge-Response Authentication: The network challenges the device to prove its identity by encrypting a random challenge with its private key.

$$Challenge = GenerateChallenge()$$

$$Response = Sign(Challenge, SK\_Device)$$

5. Verification of Response: The network verifies the device's response using its public key.

$$VerifyResponse(Response, Challenge, PK\_Device)$$

B. Attestation Mechanisms for Device Integrity

Devices in edge computing settings use authentication methods to make sure they are honest and reliable. These features keep devices safe by making sure they haven't been hacked or messed with and can safely connect to the network. There are different ways to attestation, such as software-based attestation, hardware-based attestation, and online authentication. Hardware-based attestation uses hardware security modules (HSMs) or trusted platform modules (TPMs) to check that the hardware and software of the device are correct. To make sure the device hasn't been hacked, these parts store encryption keys and run safe boot steps. Software-based authentication, on the other hand, depends on the software stack of the device being correct. As part of this, the operating system, apps, and files that are working on the device may need to be checked to make sure they have not been messed with or changed. Remote attestation adds proof by a faraway entity, like a cloud server or network router, to the attestation process, in addition to the local device. This lets the device's stability be checked all the time and can help find and stop security threats in real time.

C. Verification and Authentication Processes

Verification and identification are important parts of edge computing are safe startup and verification methods. These steps make sure that gadgets are who they say they are and haven't been hacked or messed with. Verification uses security tools like digital signatures and certificates to make sure the device is who it says it is. Authentication is the process of showing the network that the device is who it says it is by using IDs like codes or passwords. Verification and authentication work together to build trust between devices and the network. This makes it possible for edge computing settings to communicate and work together safely. These steps help protect against security risks and holes by making sure that devices are real and complete. This makes edge computing systems more reliable and safe.

1. Key Generation:

Each device generates a pair of cryptographic keys: a private key (SK) and a corresponding public key (PK). Mathematically, this can be represented as:

$$SK\_A, PK\_A \text{ for Device } A$$

$$SK\_B, PK\_B \text{ for Device } B$$

2. Signing:
Device A wants to send a message (M) to Device B and needs to prove its identity. Device A signs the message using its private key:

$$Signature = Sign(M, SK\_A)$$

3. Verification:
Device B receives the message (M) and the signature from Device A. Device B verifies the signature using Device A's public key:

$$Verify(M, Signature, PK\_A)$$

4. Result:
If the verification is successful, Device B can be confident that the message originated from Device A, as only Device A possesses the private key corresponding to PK_A.

5. Implementation

A. Resource Constraints in Edge Computing

Edge computer devices usually work in places that don't have a lot of processing power, memory, or energy. This makes it harder to use encryption strategies, since these devices might not be able to handle standard encryption methods. Lightweight encryption methods can be used when there aren't enough resources. The goal of these methods is to use less computing power while still offering enough protection. Lightweight forms of AES, like AES-128, or stream ciphers, like ChaCha20, are two examples. These methods can make edge devices' processing easier without putting security at risk. In this way, the computing load can be spread out and encryption won't have a big effect on the speed of edge devices.

B. Performance Considerations

For real-time apps, low delay and fast speed are very important in edge computing. Encryption can add extra work that can slow things down, especially in apps that care about delay. To avoid speed problems, it's important to pick encryption methods and key sizes that are the right mix of fast and safe. For better performance, use symmetric encryption for data encryption and asymmetric encryption for key sharing. This is because asymmetric algorithms like RSA require more computing power than symmetric algorithms like AES. It is also possible to speed up encryption processes with hardware acceleration methods, such as using specific secure co-processors. These co-processors are designed to work best with security processes and can make encryption much easier on the computer.

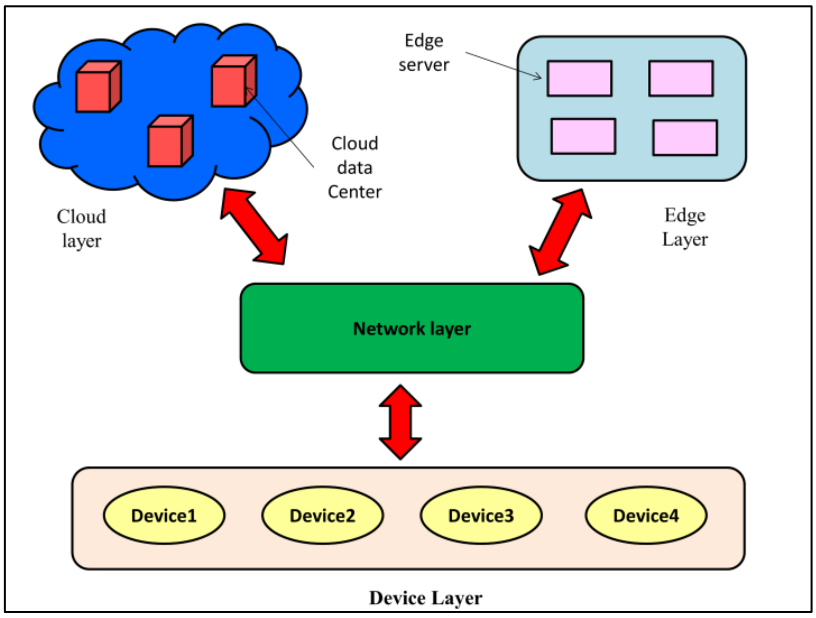

Figure 2: IoT base framework with Edge computing

C. Scalability and Flexibility

In edge computing, where the number of devices and the amount of data can change quickly, scalability is very important. Strategies for encryption should be able to grow as the number of devices and data streams increases. By spreading key management jobs across various nodes, decentralized key management systems (DKMS) can make things more scalable. For instance, blockchain-based DKMS can offer an adaptable and scalable way to handle encryption keys in edge settings. It's also important to be flexible, since edge computing settings often have a lot of different gadgets and apps. Different encryption methods, key sizes, and protocols should be able to be used with different encryption techniques so that they can work with all kinds of devices and applications.

6. Cryptographic Methods

A. Digital Signature

In edge computing, digital signatures are used to make sure that data and messages are real. Devices can show that data is real and came from the right place by signing it with a private key. Digital signatures are very important for making sure that software updates, setup changes, and contact between devices in edge areas are real.

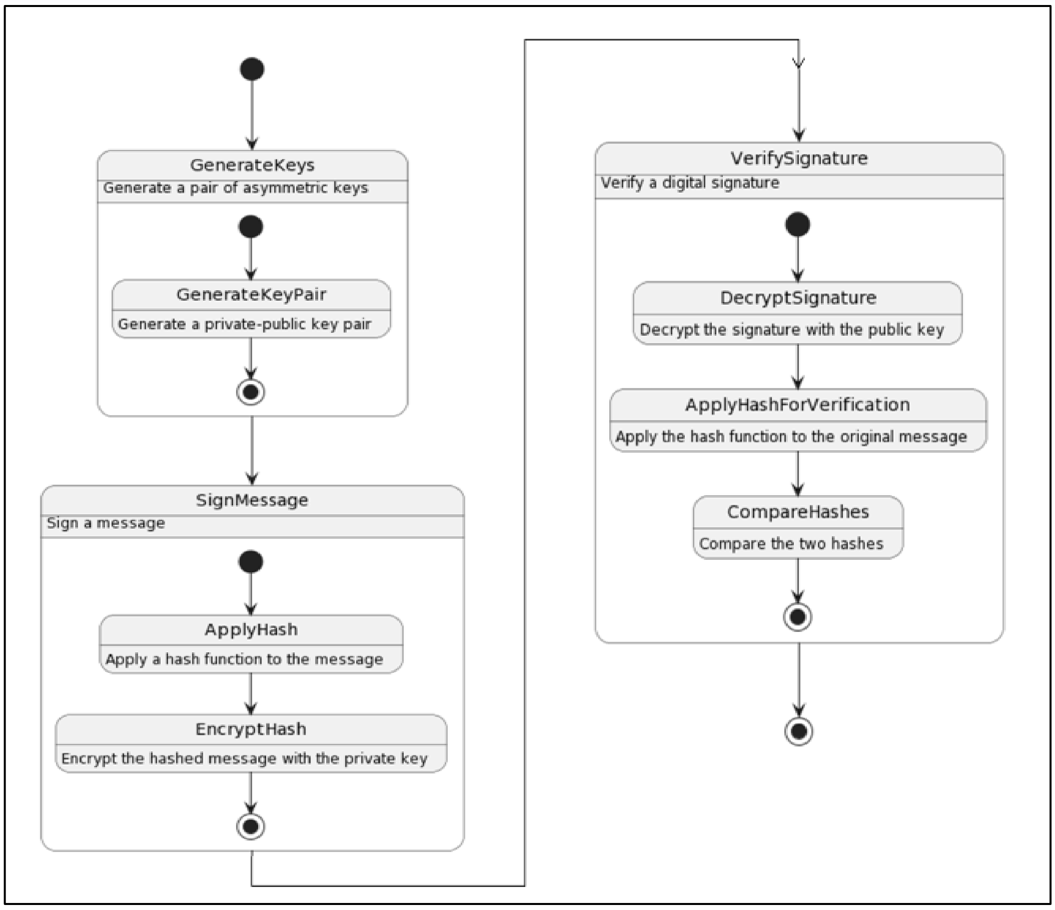
GenerateKeys
Generate a pair of asymmetric keys
GenerateKeyPair
Generate a private-public key pair
SignMessage
Sign a message
ApplyHash
Apply a hash function to the message
EncryptHash
Encrypt the hashed message with the private key
VerifySignature
Verify a digital signature
DecryptSignature
Decrypt the signature with the public key
ApplyHashForVerification
Apply the hash function to the original message
CompareHashes
Compare the two hashes

Figure 3: Flowchart for Digital Signature Generation and Verification

Mathematical modeling:

1. Key Generation:

- Generate a pair of asymmetric keys: a private key (SK) and a corresponding public key (PK).

$$SK, PK$$

2. Signing:

- To sign a message (M), apply a hash function (H) to the message to create a message digest.

$$Hash = H(M)$$

- Encrypt the message digest with the private key (SK) to create the digital signature (DS).

$$DS = Encrypt(Hash, SK)$$

3. Verification:

- To verify the digital signature, decrypt the digital signature (DS) with the public key (PK) to obtain the message digest.

$$Hash' = Decrypt(DS, PK)$$

- Apply the same hash function (H) to the original message (M) to obtain the message digest.

$$Hash = H(M)$$

- Compare the two message digests (Hash and Hash'). If they are equal, the signature is valid; otherwise, it is not.

$$Hash == Hash'$$

B. Secure Multiparty Computation

SMPC lets more than one person work together to compute a function over their inputs without sharing those inputs. This could be helpful in edge computing situations where private data from many devices needs to be handled safely without showing which devices sent the data.

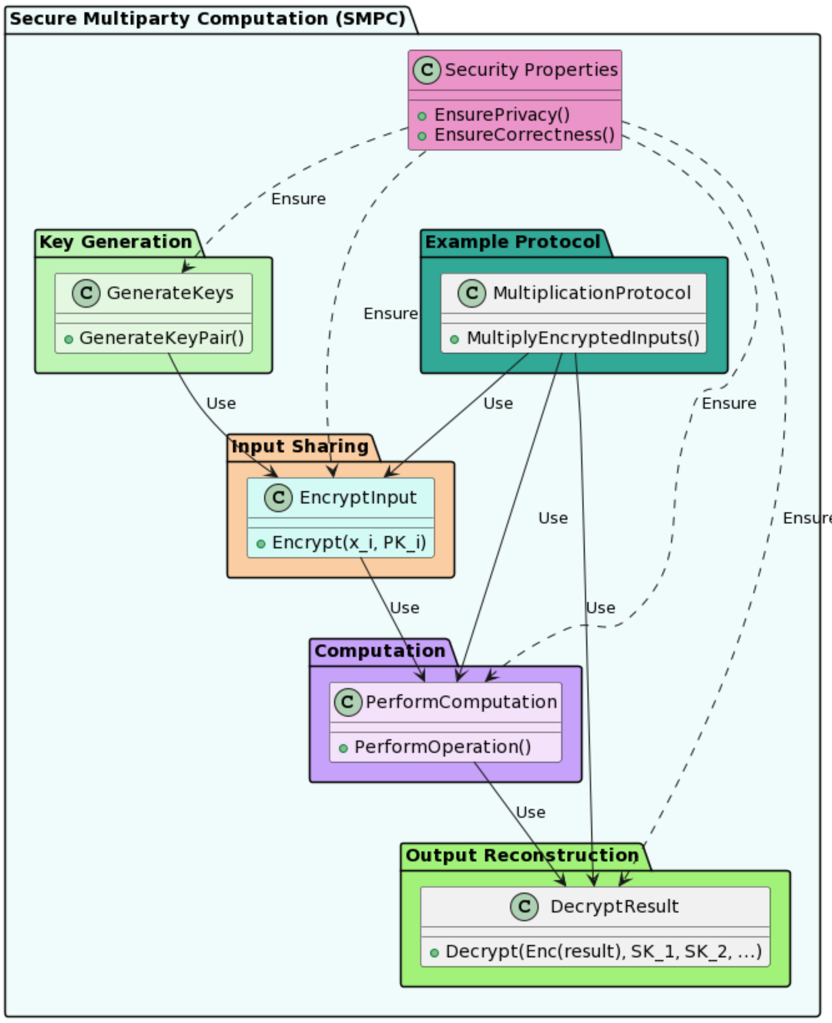

Figure 4: Overview of Secure Multiparty Computation

1. Key Generation:
Each party generates a pair of public and private keys for homomorphic encryption.

$$(PK\_i, SK\_i)\ for\ party\ i.$$

2. Input Sharing:
Each party i encrypts their input x_i using homomorphic encryption with their own public key.

$$Encrypted\ input: Enc(x_i) = Encrypt(x_i, PK_i).$$

3. Computation:

- Parties perform computations on the encrypted inputs without decrypting them, using homomorphic properties of the encryption scheme.

For example, for addition:

$$Enc(x_1) + Enc(x_2) = Encrypt(x_1 + x_2, PK_1) \times Encrypt(x_1 + x_2, PK_2)$$

4. Output Reconstruction:

- After the computation is done, parties jointly decrypt the result to obtain the final output.

Final output:

$$Dec\big(Enc(result)\big) = Decrypt(Enc(result), SK_1, SK_2, \dots).$$

5. Security Properties:

- SMPC ensures that no single party learns any other party's input, and the final result is correct.

Example Protocol:

- Parties have $inputs\ x\_1\ and\ x\_2$.
- Each party encrypts their input: $Enc(x\_1)\ and\ Enc(x\_2)$.

- Parties multiply their encrypted inputs together: $Enc(x\_1) \times Enc(x\_2)$.
- Parties jointly decrypt the result to obtain $x\_1 + x\_2$.

7. Conclusion

Using advanced encryption methods is a must if you want to make edge computing settings safer online. The large number of monitors and gadgets linked to the Internet of Things (IoT) has made it much more important to protect data and contact at the edge. Edge computing comes with its own problems, like limited resources, speed issues, and the need to be able to grow. To solve these problems, companies can use advanced encryption methods like hybrid encryption, temporary key exchange protocols, and autonomous key management systems. The benefits of both symmetric and asymmetric methods are combined in hybrid encryption, which strikes a balance between speed and safety. Key compromise attempts are harder to do when encryption keys are not reused. Ephemeral key sharing methods make sure that keys are only used once. It is very important for dynamic and distributed edge settings that key sharing and management are safe and flexible. This is made possible by decentralized key management systems. It's also impossible to say enough about how important safe bootstrapping, validation methods, and verification/authentication processes are. Edge computing devices and data are protected against harmful threats and illegal entry by these methods that build trust. Using advanced encryption methods is necessary to make edge computing more secure online. Not only do these strategies keep private data safe, they also make sure that edge computing systems are reliable and secure, which makes them less vulnerable to online dangers.

References

[1] H. Lee, S. Kim and H. K. Kim, "SoK: Demystifying Cyber Resilience Quantification in Cyber-Physical Systems," 2022 IEEE International Conference on Cyber Security and Resilience (CSR), Rhodes, Greece, 2022, pp. 178-183, doi: 10.1109/CSR54599.2022.9850312.

[2] S. Jawhar, C. E. Kimble, J. R. Miller and Z. Bitar, "Enhancing Cyber Resilience with AI-Powered Cyber Insurance Risk Assessment," 2024 IEEE 14th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 2024, pp. 0435-0438, doi: 10.1109/CCWC60891.2024.10427965.

[3] M. N. Halgamuge, "Leveraging Deep Learning to Strengthen the Cyber-Resilience of Renewable Energy Supply Chains: A Survey," in IEEE Communications Surveys & Tutorials, doi: 10.1109/COMST.2024.3365076.

[4] S. D. Roy and S. Debbarma, "Enhancing Cyber-Resilience of Power Systems' AGC Sensor Data by Time Series to Image Domain Encoding," in IEEE Transactions on Smart Grid, doi: 10.1109/TSG.2024.3361014.

[5] J. C. Haass, "Cyber Threat Intelligence and Machine Learning", 2022 Fourth International Conference on Transdisciplinary AI (TransAI), pp. 156-159, 2022.

[6] G. Langner, J. Andriessen, G. Quirchmayr, S. Furnell, V. Scarano and T. J. Tokola, "Poster: The Need for a Collaborative Approach to Cyber Security Education", 2021 IEEE European Symposium on Security and Privacy (EuroS&P), pp. 719-721, 2021.

[7] M. Neumann, M. Rauschenberger and E. -M. Schön, "We Need to Talk About ChatGPT": The Future of AI and Higher Education", 2023 IEEE/ACM 5th International Workshop on Software Engineering Education for the Next Generation (SEENG), pp. 29-32, 2023.

[8] L. Li, W. He, L. Xu, A. Ivan, M. Anwar and X. Yuan, "Does Explicit Information Security Policy Affect Employees' Cyber Security Behavior? A Pilot Study", 2014 Enterprise Systems Conference, pp. 169-173, 2014.

[9] Ajani, S. N. ., Khobragade, P. ., Dhone, M. ., Ganguly, B. ., Shelke, N. ., & Parati, N. . (2023). Advancements in Computing: Emerging Trends in Computational Science with Next-Generation Computing. International Journal of Intelligent Systems and Applications in Engineering, 12(7s), 546–559

[10] V. Sundararajan, A. Ghodousi and J. E. Dietz, "The Most Common Control Deficiencies in CMMC non-compliant DoD contractors", 2022 IEEE International Symposium on Technologies for Homeland Security (HST), pp. 1-7, 2022.

[11] P. P. Roy, "A High-Level Comparison between the NIST Cyber Security Framework and the ISO 27001 Information Security Standard", 2020 National Conference on Emerging Trends on Sustainable Technology and Engineering Applications (NCETSTEA), pp. 1-3, 2020.

[12] Chandu Vaidya, Prashant Khobragade and Ashish Golghate, "Data Leakage Detection and Security in Cloud Computing", GRD JournalsGlobal Research Development Journal for Engineering, vol. 1, no. 12, November 2016.