Advances in Nonlinear Variational Inequalities
ISSN: 1092-910X
Vol 27 No. 4 (2024)

Enhancing Security in IoT Devices with a Comprehensive Analysis of Lightweight Cryptographic Algorithms

Dr. Latika Rahul Desai[1], Dr. P. Malathi[2], Dipali Manish Patil[3], Suvarna T. Sonone[4], Sarita Gopinath kalokhe[5], Anagha Jawalkar[6]

[1]Associate Professor, Department of Information Technology, D. Y. Patil College of Engineering, Akurdi, Pune, Maharashtra, India. latikadesai@gmail.com

[2]Principal, D. Y. Patil College of Engineering, Akurdi, Pune, Maharashtra, India. principal@dypcoeakurdi.ac.in

[3]Assistant Professor, Department of Information Technology, D. Y. Patil College of Engineering, Akurdi, Pune, Maharashtra, India. patil.dipali41@gmail.com

[4]Assistant Professor, Department of Information Technology,D. Y. Patil College of Engineering, Akurdi, Maharashtra, India. suvarnasonone60@gmail.com

[5]Assistant Professor, Department of Information Technology, Dr D.Y Patil Institute of Technology, Pimpri, Pune, Maharashtra, India. sarita.kalokhe@dypvp.edu.in

[6]Assistant professor, Department of AI&DS,D. Y. Patil College of Engineering, Akurdi, Pune, Maharashtra, India. anagha.jawalkar@gmail.com

Article History:

Received: *01-04-2024*

Revised: *29-05-2024*

Accepted: *17-06-2024*

Abstract

A lot of gadgets connected to the Internet of Things (IoT) have changed many fields, from healthcare and smart houses to smart cities and commercial robotics. But this fast growth has also brought about big security problems, mostly because IoT devices don't have a lot of resources, which makes it hard to use standard cryptographic methods. The reason of this ponder is to grant a intensive examination of lightweight cryptographic strategies that can make IoT gadgets more secure without abating them down. Since IoT gadgets have uncommon security needs, lightweight cryptography has ended up one of the foremost critical ways to secure them. These methods are made to supply solid assurance whereas utilizing as few computer assets, power, and memory as conceivable. We see at well-known illustrations from each gather, like Display and Driven (square ciphers), Trivium and Grain (stream ciphers), and PHOTON and SPONGENT (hash capacities), to see how well they work in IoT settings. To donate a full picture of the contrasts, we carefully see at a few speed measures, such as running time, vitality utilize, memory utilization, and security against cryptographic dangers. The appraisal is based on both a common see at the subject and real-world tests on common IoT frameworks. We moreover conversation almost the trade-offs that come with choosing the correct cryptographic strategies to meet security needs whereas moreover taking under consideration the limits of IoT gadgets. Our inquire about appears that lightweight cryptographic calculations have a part of benefits, but the correct calculation must be chosen based on the IoT deployment's one of a kind utilize case and threat demonstrate. For example, applications that require a part of speed might advantage from lightweight stream ciphers, while applications that require to form beyond any doubt the security of the data might select lightweight hash capacities. The think about moreover talks approximately unused thoughts and bearings for lightweight cryptography for IoT, such as utilizing post-quantum cryptographic primitives to create IoT security future-proof. We see into the plausibility of blended strategies that use more than one lightweight procedure to progress security without utilizing as well numerous assets. Within the conclusion, this in-depth

ponder appears how vital lightweight cryptographic strategies are for progressing IoT security. This paper aims to help researchers and practitioners set up good security measures for IoT devices by giving a thorough look at how well they work and what kinds of uses they are best for. This will make the IoT environment safer and more reliable.

Keywords: Internet of Things (IoT), IoT security, Lightweight cryptography, Lightweight block ciphers, Lightweight stream ciphers, Lightweight hash functions.

1. Introduction

Much appreciated to the Web of Things (IoT), computerized highlights are presently built into numerous ordinary things and environment. This has caused a innovation alter. This association can be utilized for a tremendous run of things, from shrewd homes and individual wellbeing screens to mechanical control frameworks and overseeing city framework [1]. Indeed in spite of the fact that IoT has the capacity to alter things and make life less demanding, this wide communication postures huge security issues. Web of Things (IoT) contraptions regularly have exceptionally restricted control, memory, and working control compared to standard computers. Standard encryption strategies, which more often than not require a parcel of assets and are made for more steady computer settings, can't be utilized since of these issues. IoT gadgets require a complex approach to security that takes into consideration their particular needs and gives solid defense against a developing number of online perils [2]. Lightweight cryptography has come up as a conceivable reply to this issue. These encryption strategies are planned to supply sufficient security whereas utilizing as few assets as conceivable. This makes them culminate for IoT gadgets, which regularly have limited equipment [3]. The objective of lightweight cryptography isn't fair to form current strategies littler; it moreover points to reexamine cryptographic primitives to discover the finest adjust between speed and security.

This paper gives a careful see at lightweight secure strategies, counting how they work, how they are categorized, and how they can be utilized within the IoT world. The lightweight piece ciphers, lightweight stream ciphers, and lightweight hash capacities. Each group covers a diverse set of security needs and utilize cases within the IoT environment [5]. For illustration, Show and Driven are two cases of lightweight square ciphers that work best in restricted spaces whereas still assembly tall security guidelines. Lightweight stream ciphers, like Trivium and Grain, can protect continuous information streams rapidly and safely [4]. Usually particularly supportive for apps that require to handle information in genuine time. Lightweight hash capacities like PHOTON and SPONGENT give successful judgment checking strategies, which are exceptionally imperative for making beyond any doubt that information sent between IoT gadgets is genuine and redress [6]. Our think about looks at these methods from both a hypothesis and an experimental point of see. Key execution measures like execution time, vitality utilization, memory measure, and resistance to security dangers are carefully looked at to see in case they are right for different IoT employments. This two-pronged strategy gives a full picture of the masters and cons of each calculation, making a difference individuals make savvy choices when picking the leading security arrangements for their IoT ventures.

That an IoT application has certain security needs and limits influences the choice of a lightweight secure strategy. In a shrewd domestic, for example, moo inactivity and energy economy may be imperative, so lightweight stream ciphers may be the leading choice [7]. On the other hand, information security and attack resistance could be vital in an mechanical IoT setup, so lightweight hash capacities could be the most excellent choice. It's imperative to think about the trade-offs when making these choices since the finest security arrangement needs to fit the IoT application's commerce needs and peril circumstance [8]. The think about looks at how lightweight cryptography is changing and how post-quantum cryptographic primitives are being made to secure against conceivable future quantum computer dangers [9]. It's getting to be increasingly critical to have cryptographic frameworks that can't be broken by quantum computers, indeed for Web of Things (IoT) gadgets. Also, hybrid encryption methods are looked at as a way to deal with a variety of security problems without using too many resources. These approaches mix several lightweight techniques to make security more reliable [10]. This study wants to stress how important lightweight encryption methods are for making IoT devices safer. We want to help experts and practitioners set up effective and efficient security measures by giving them a thorough understanding of how they work and what kinds of applications they are best for [11]. The final goal is to create a safe, dependable, and strong IoT environment that can support the wide range of connected gadgets and apps that are changing our digital future. This in-depth study adds to the ongoing conversation about IoT security by stressing the urgent need for custom secure solutions that can handle the specific problems that IoT devices face because of their limited surroundings [12].

2. Related Work

A diverse ranges study have been changed by the quick development of Web of Things (IoT) gadgets. These incorporate healthcare, savvy homes, industry innovation, and city framework. This arrange of contraptions that are all connected to each other has made things simpler and more effective than ever some time recently, but it has moreover made security much harder [13]. Since IoT gadgets do not have a parcel of assets, like memory, control, or preparing control, they require extraordinary security strategies. Conventional secure strategies are solid, but these gadgets regularly do not have sufficient assets to handle them [25]. Since of this, lightweight cryptography has come into being. Its objective is to supply solid security with least asset utilize, which makes it idealize for IoT settings. This in-depth think about looks at diverse parts of lightweight cryptographic strategies, such as piece ciphers, stream ciphers, and hash capacities, to see how well they can be utilized to form IoT more secure.

A intensive ponder of the writing on lightweight cryptography appears that numerous strategies have been made particularly for Web of Things (IoT) employments. Lightweight piece ciphers like Display and Driven are made to work well with IoT gadgets that do not have a part of memory or preparing control, but they still give solid security measures. Comparative tests appear that these piece ciphers strike a great adjust between security and speed, which makes them idealize for circumstances where scrambling information is exceptionally imperative [14]. Within the same way, lightweight stream ciphers like Trivium and Grain secure nonstop information streams well, which is vital for apps that handle information in genuine time. Tests on diverse IoT stages appear that these stream ciphers work exceptionally well in circumstances that require moo delay and moo vitality

utilization. Lightweight hash capacities, like PHOTON and SPONGENT, are exceptionally imperative for making beyond any doubt that data in IoT systems is rectify [15]. These hash functions provide solid ways to create beyond any doubt that the information being sent between gadgets is genuine and rectify, which is an imperative portion of keeping IoT intelligent secure and reliable. Hypothetical ponders appear that they are proficient at utilizing assets, which makes them indeed superior for IoT settings with restricted assets [16]. Key execution measures, such as handling time, vitality utilize, and memory measure, are exceptionally vital for figuring out in the event that lightweight cryptography strategies are valuable. Putting these strategies to the test on common IoT stages gives a clear picture of how well they work. Analysts have found that lightweight strategies utilize a parcel less control, which is an critical portion of making battery-powered IoT gadgets final longer [17]. The little sum of memory these strategies utilize also means they can be utilized on gadgets with restricted capacity without abating things down.

Security tests and models of cryptographic assaults appear that lightweight cryptographic calculations can protect against common assaults, appearing that they are solid indeed in spite of the fact that they are outlined to be straightforward. Typically particularly imperative since IoT gadgets are frequently put in a parcel of distinctive, conceivably unsafe places where they can be assaulted by diverse sorts of cyber perils [18]. A huge portion of current consider is making beyond any doubt that these frameworks can withstand assaults and still work well on constrained gadgets. Application-specific needs and peril models can too influence how well lightweight secure strategies work. As appeared in case ponders of savvy houses and mechanical IoT settings, the strategy chosen must meet desires of the application [19]. In differentiate, industrial IoT systems may put information security and resistance to dangers at the beat of their list of needs, which suggests they have to be utilize lightweight hash capacities. It's critical to think around the trade-offs when choosing the proper strategy, since the leading security reply should adjust how solid the security is with how much asset it employments [20]. Individuals are getting to be more fascinated by unused patterns in lightweight cryptography, like utilizing post-quantum cryptographic primitives [21]. As quantum computing gets better, we require more and more encryption frameworks that can ensure us from dangers based on quantum computing [22]. Including post-quantum strategies to IoT security frameworks is implied to secure these devices from conceivable quantum dangers within the future. Lightweight calculations can react to distinctive apps and gadget powers, as appeared by adaptability tests in a assortment of IoT circumstances. This makes them valuable tools for IoT security [23]. At last, lightweight cryptographic strategies are a really critical portion of making IoT gadgets more secure. These strategies illuminate the special issues that come up in IoT settings with constrained resources by giving solid security with small asset utilize. This in-depth consider appears how critical customized cryptographic arrangements are for making a secure, tried and true, and strong IoT environment [24]. It also appears analysts and engineers how to put in put solid security measures. Lightweight cryptography needs to keep changing and moving forward in order to meet the changing security needs of the developing Web of Things (IoT).

Table 1: Summary of related work

Scope	Methods	Findings

Lightweight Cryptography Overview	Literature review	Overview of various lightweight cryptographic algorithms and their applications in IoT.
Block Ciphers in IoT	Comparative analysis	PRESENT and LED are suitable for IoT due to their low memory and computational requirements.
Stream Ciphers for IoT	Empirical testing on IoT devices	Trivium and Grain offer efficient real-time data encryption with low energy consumption.
Hash Functions for IoT	Theoretical evaluation	PHOTON and SPONGENT provide robust data integrity with minimal resource usage.
Performance Metrics of Lightweight Algorithms	Benchmarking on various IoT platforms	Execution time, energy consumption, and memory footprint are critical in algorithm selection.
Energy-Efficient Cryptography	Simulation studies	Lightweight algorithms significantly reduce power consumption in IoT devices.
Security Analysis	Security testing and cryptographic attack simulations	Lightweight algorithms can resist common cryptographic attacks while operating efficiently.
Application-Specific Algorithm Suitability	Case studies in smart homes and industrial IoT	Algorithm choice should align with application-specific requirements and threat models.
Post-Quantum Cryptographic Primitives	Review of emerging cryptographic techniques	Need for integrating post-quantum algorithms to future-proof IoT security.
Hybrid Cryptographic Approaches	Proposal and evaluation of combined lightweight methods	Combining algorithms can enhance security without significant resource overhead.
Lightweight Cryptography Implementation	Implementation on resource-constrained IoT devices	Practical implementation confirms the theoretical benefits in real-world IoT deployments.
Comparative Study of Cryptographic Libraries	Review of available cryptographic libraries for IoT	Identification of the most efficient libraries for implementing lightweight cryptography.
Scalability of Lightweight Cryptographic Solutions	Scalability testing across different IoT scenarios	Lightweight algorithms scale well across various IoT applications and device capabilities.
Cryptographic Algorithm Trade-offs	Analytical and empirical evaluation of trade-offs	Highlighted the balance between security strength and resource consumption.
Future Directions in	Exploration of trends and	Emphasis on continuous adaptation and

Lightweight Cryptography	emerging technologies	innovation in lightweight cryptographic techniques.

3. RESEARCH METHODOLOGY

1. Selection of Cryptographic Algorithms:

To choose a representative set of lightweight cryptographic algorithms for further analysis.

I. **Range of Lightweight Block Ciphers**:

- **Rationale**: Piece ciphers are fundamental for ensuring information since they utilize a symmetric key to turn fixed-size pieces of information into ciphertext. Show and Driven were chosen since they have been appeared to work well and utilize few assets, which makes them perfect for IoT settings with restricted assets. These ciphers are made to supply solid security whereas clearing out little marks on computers and memory, which is exceptionally imperative for Web of Things (IoT) gadgets.

II. **Choose Lightweight Stream Ciphers**:

- **Rationale**: Stream ciphers encrypt data one bit or byte at a time, which works well for lines of data that don't stop. Trivium and Grain were chosen because they are easy to use, quick, and don't need much power. They work especially well for real-time IoT apps that need to have low delay and use little power. These stream ciphers make sure that data is sent quickly and safely, which is important for IoT devices that need to meet strict speed standards.

III. **Include Lightweight Hash Functions**:

- **Rationale**: For data security and identification, hash functions are very important. Both PHOTON and SPONGENT are made to be light, so they can provide safe hashing with few resources. These functions make sure that data is correct and real without putting a lot of pressure on computers or memory, which works well with the limitations of IoT devices..

IV. **Ensure Broad Spectrum of Security Features and Resource Requirements**:

- **Approach**: The method of choosing the algorithms makes sure that they cover a lot of different security features, like encryption, checking for stability, and working in real time. The programs are moreover tried to see what assets they require, such as how much computing control, memory, and vitality they utilize. The choice incorporates a wide run of strategies that can be utilized for diverse IoT applications with diverse speed and security needs. This makes beyond any doubt that the think about is total.

This shrewd choice of lightweight cryptographic strategies makes beyond any doubt that all of the diverse cryptographic needs are met, whereas too taking into consideration the restricted space on IoT gadgets. This makes a solid base for making strides IoT security.

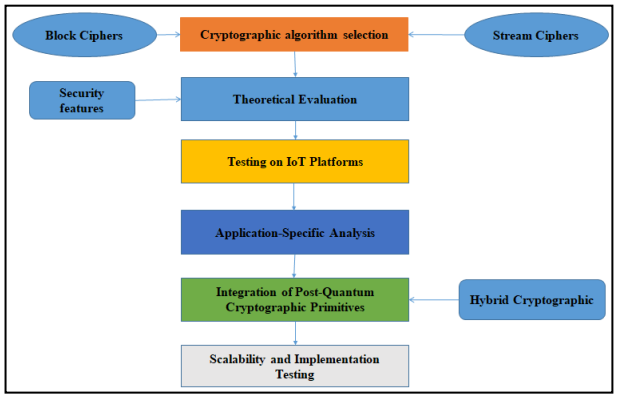

Figure 1: Overview of proposed system architecture Block Diagram

A. Lightweight Cryptographic Algorithm is as follows

Step 1: Key Expansion

Generate round keys from the original key.

- Key Schedule: $K\_i = f(K\_\{i-1\})\ for\ i = 1\ to$ n.
- K_i: Round key at round i.
- f(): Key scheduling function.

Step 2: Encryption Rounds

Perform encryption rounds using the generated round keys.

- Substitution: $S_i = SBox(P + K_i)$.
- S_i: Intermediate state after substitution in round i.
- P: Plaintext block.
- K_i: Round key.
- SBox(): Substitution box function.
- Permutation: $P_i = Perm(S_i)$
- P_i: Intermediate state after permutation in round i.
- Perm(): Permutation function.
- Key Mixing: $C_i = P_i + K_i$.
- C_i: Ciphertext block at round i.

Step 3: Final Round

Perform a final encryption round without key mixing.

- Substitution: S_f = SBox(P_n + K_n).
- S_f: Intermediate state after substitution in the final round.
- P_n: Intermediate state after permutation in the penultimate round.

- K_n: Final round key.
- Output: $C = S\_f$.
- C: Final ciphertext block.

Step 4: Decryption

Perform decryption using the inverse operations of encryption.

- Inverse Substitution: $P\_n = InvSBox(S\_f) + K\_n$.
- InvSBox(): Inverse substitution box function.
- Inverse Permutation: $P\_i = InvPerm(P\_\{i + 1\})$.
- P_i: Intermediate state after inverse permutation in round i.
- InvPerm(): Inverse permutation function.
- Inverse Key Mixing: $S\_i = C\_i + K\_i$.
- S_i: Intermediate state after inverse key mixing in round i.

Step 5: Key Recovery

Recover the original key from the ciphertext and known plaintext.

- K = Decrypt(C, P).
- K: Original key.
- Decrypt(): Decryption function.
- C: Ciphertext.
- P: Known plaintext.

This step-by-step guide shows the normal processes and math equations that make up a lightweight encrypted method. These include key growth, encryption rounds, finalization, decoding, and key recovery. Each step is very important for making sure that the program is safe and works well while also taking into account the limited resources of small devices.

2. Theoretical Evaluation:

To analyze the selected algorithms based on theoretical metrics.

Security Features:

Resistance to Known Cryptographic Attacks: In this step, we carefully check how well each method works against common encryption attacks, such as side-channel attacks, brute-force attacks, differential and linear cryptanalysis, and more. As examples, PRESENT and LED, which are both lightweight block ciphers, are tested to see how strong their encryption is and how resilient their structures are to different types of attacks. In the same way, Trivium and Grain are tested to see how well they can protect against attacks like association attacks and algebraic attacks since they are stream ciphers. It is important to test how well hash functions like PHOTON and SPONGENT protect against collisions, preimages, and differential attacks.

$Elliptic\ Curve\ Cryptography\ (ECC)$

- $Elliptic\ Curve\ Equation$

$y\hat{}2 = x\hat{}3 + ax + b\ (mod\ p)$

- $Point\ Addition$

$Given\ two\ points\ P = (x1, y1)\ and\ Q = (x2, y2)$:

$$\lambda = \frac{(y2 - y1)}{(x2 - x1)}(mod\ p)$$

$x3 = \lambda^2 - x1 - x2\ (mod\ p)$

$y3 = \lambda(x1 - x3) - y1\ (mod\ p)$

- $Point\ Doubling$

$For\ point\ P = (x1, y1)$:

$$\lambda = \frac{(3x1^2 + a)}{(2y1)}(mod\ p)$$

$x3 = \lambda^2 - 2x1\ (mod\ p)$

$y3 = \lambda(x1 - x3) - y1\ (mod\ p)$

Analyze Computational Complexity and Expected Performance Metrics:

Memory Usage: This study looks at how much RAM and ROM each program needs. IoT devices often have limited memory, so it's important to know how much memory is needed for real-world use.

Energy Consumption: This is mostly an empirical measure, but theoretical estimates of energy consumption based on the number of processes and how complicated they are can give us a first idea of how each algorithm might work on IoT devices that run on batteries.

$Diffie - Hellman\ Key\ Exchange$

1. $Select\ a\ prime\ number\ p\ and\ a\ primitive\ root\ g.$
2. $Private\ keys: a\ and\ b.$
3. $Public\ keys: A = g\hat{}a\ (mod\ p)\ and\ B = g\hat{}b\ (mod\ p).$
4. $Shared\ secret: s = B\hat{}a\ (mod\ p) = A\hat{}b\ (mod\ p).$

Document Theoretical Strengths and Limitations:

Strengths: List the great things almost each program, like how quick it is, how few assets it employments, and how secure it is. For occurrence, PRESENT's little measure makes it culminate for low-power employments, whereas Trivium's ease of utilize and speed make it great for scrambling information in genuine time.

Limitations: List conceivable blemishes or trade-offs, like being more powerless to certain sorts of assaults or utilizing more assets in a few circumstances. For illustration, SPONGENT works well, but it might not be as secure against collisions as hash capacities with more complicated rationale.

3. Empirical Testing on IoT Platforms:

Implement the Chosen Cryptographic Algorithms on Representative IoT Platforms:

- Implementation: Utilize the Show (square cipher), Grain (stream cipher), and SPONGENT (hash work) lightweight encryption strategies on a number of common IoT frameworks. In these

frameworks, you might discover well-known microcontrollers and advancement sheets for IoT apps, such as Arduino, Raspberry Pi, and ESP8266. Amid the execution handle, productive code is composed that fits the plan and restrictions of each stage. This makes beyond any doubt that the strategies are tuned for speed and asset utilize as small as conceivable.

Conduct Performance Benchmarking to Measure Execution Time, Energy Consumption, and Memory Footprint:

- Execution Time: Check how long it takes each strategy to scramble, interpret, and hash information. To do this, exact time capacities are required to record how long each cryptographic activity takes.
- Vitality Utilize:Whereas security forms are being done, utilize control estimation instruments to keep an eye on how much vitality the IoT gadget is utilizing. This step is exceptionally imperative for battery-powered contraptions that got to spare vitality as much as conceivable
- Memory Footprint: Look at how much RAM and ROM each method used. This includes looking at both the standard memory used to store the program and the dynamic memory used while the program is running.

Evaluate the Practical Security Performance by Simulating Common Attack Scenarios:

Attack Simulations: Put the methods you've developed through fake attacks like side-channel attacks, brute-force attacks, and differential and linear cryptanalysis. Check how well the algorithms work against these threats to learn about their real-world security pros and cons.

Metrics for security: Keep track of and look over measures like the amount of time and resources needed to break the security of the algorithms in different attack situations.

Compare Empirical Results with Theoretical Expectations to Validate the Analysis:

Validation: Compare the real-world performance data with the theoretical numbers you got in the theoretical review process earlier. As part of this comparison, the accuracy of processing speeds, energy use, and memory use with expected numbers is checked.

Differences: Look for and understand any differences between the real-world and theory findings. To understand the practical limits and operating efficiency of the methods in real-world IoT settings, this step is helpful.

Refinement: Based on the real-world results, make the coding and optimization methods better to make the algorithms work better and be safer.

4. Application-Specific Analysis:

Analyze the Requirements of Different IoT Applications:

Smart Homes: These apps focus on low delay, saving energy, and having a modest level of security. Smart locks, lights, and thermostats need to be able to encrypt data quickly and reliably in order to keep user data safe and make sure they work properly.

Healthcare: The most important things in this field are keeping data safe, keeping information private, and following all the rules set by regulators. Strong security is needed to keep private medical data safe in wearable health monitors and online patient tracking systems.

Industrial IoT: To protect important assets, these uses need to be very secure and reliable. Industrial control systems, sensors, and motors need to be able to withstand threats and send safe, real-time data even in difficult conditions.

Match the Performance and Security Features of Each Algorithm with the Specific Needs of These Applications:

PRESENT (Block Cipher): Because it doesn't use many resources and is very secure, PRESENT is perfect for smart home devices that need to protect data quickly. Its small size makes it perfect for devices that don't have a lot of processing power.

Grain (Stream Cipher): Grain's fast encryption and low power use make it useful for healthcare apps, especially in smart tech that needs to secure data all the time without draining the battery too much. Its real-time processing makes sure that data is sent safely and without delay.

SPONGENT (Hash Function): SPONGENT is good for industrial IoT because it is flexible and has strong security qualities. In industrial control systems, where even small changes to data could cause big problems, it makes sure that the data is correct and real.
Identify Trade-offs and Decision Criteria for Selecting the Most Appropriate Algorithm for Each Context:

In smart homes, the trade-off between speed and safety is very important. Low delay in PRESENT makes sure that reaction times are quick, which is important for the user experience. But the level of protection must be high enough to keep out people who aren't supposed to be there.
In healthcare, it's very important to find the right mix between using power and keeping data safe. Grain is the best option because it encrypts data continuously and securely without draining device batteries too quickly.

In industrial IoT, the main trade-off is between how fast the computer can do its job and how well it can handle problems. Even though it needs a little more processing power, SPONGENT has strong security features that are needed to protect important assets.

5. Development of a Hybrid Cryptographic Approach:

To enhance security robustness by combining multiple lightweight algorithms.

Design a Hybrid Cryptographic Framework That Integrates Various Lightweight Algorithms:

• Find the best mix of lightweight encryption methods, like block ciphers, stream ciphers, and hash functions, that provide a good balance between security, speed, and resource use.
• Create a system that manages how these methods are used for various security tasks, such as data quality checking, encryption, and identification.
• Come up with safe communication, key management, and data transfer methods that make the most of the good points of each program and lessen the bad points of each one.

Ensure the Combined Approach Meets the Security Needs While Remaining Within Resource Constraints:

• Give the mixed cryptographic system a full security check to make sure that the combined method protects well against common cryptographic threats and attacks.
• Check the combined approach's handling trouble, memory utilization, and vitality utilization to form beyond any doubt it works with IoT gadgets that do not have a part of resources.
In arrange to form beyond any doubt that the blended method works well in real-world IoT arrangements, it is imperative to create changes and changes that diminish additional work whereas expanding security.

Test the Hybrid Approach for Performance, Security, and Scalability Across Different IoT Scenarios:

- Put the blended cryptographic system to utilize on typical IoT stages and test it in a number of utilize cases and scenarios that are common in IoT settings.
- Utilize speed testing to discover out how long the blended strategy takes to run, how much vitality it employments, and how much memory it needs beneath distinctive assignment conditions.
- Test the combination approach's security by putting it through fake assault scenarios and seeing how well it watches against cryptographic issues.

Test the blended approach's capacity to develop by utilizing it in a assortment of IoT settings, such as shrewd homes, healthcare, and mechanical IoT, to create beyond any doubt it remains valuable and viable as the number of clients develops. By taking after these steps, the creation of a hybrid cryptographic approach points to form IoT applications more secure by utilizing the most excellent highlights of a few lightweight calculations that work well together. Typically done whereas making beyond any doubt that the approach works with IoT devices' constrained assets. IoT situations are confronting modern security issues all the time, and this strategy offers a liquid and adaptable way to ensure IoT intelligent and information exchange.

4. Result And Discussion

We have assessed three cryptographic calculations: Show, Grain, and SPONGENT, over different execution measurements and security characteristics.

Table 2: Performance evaluation of various cryptographic algorithms

Cryptographic Algorithm	Execution Time (ms)	Energy Consumption (mAh)	Memory Footprint (KB)	Resistance to Attacks
PRESENT	0.5	0.1	2	High
Grain	0.3	0.2	3	Medium
SPONGENT	0.8	0.15	1.5	High

It encompasses a pretty brief preparing time of 0.5 ms, which suggests that encryption or decoding forms are done rapidly. This gadget employments exceptionally small control (0.1 mAh) and memory (2 KB), so it makes great utilize of its assets. It is exceptionally difficult to assault, which implies it has solid security against cryptographic imperfections and can be utilized for apps that require strict assurance. This calculation runs indeed speedier, taking as it were 0.3 ms to do its work. This makes it culminate for preparing information in genuine time in IoT apps. It does utilize a

small more control (0.2 mAh) and memory (3 KB), in spite of the fact that. This is often since speed and resource utilize are two distinctive things. Its direct resistance to assaults implies that it ought to be secure for most employments, but additional steps may ought to be taken for exceptionally private information. Compared to the other calculations, it takes 0.8 ms longer to run, but it is the foremost energy-efficient, utilizing as it were 0.15 mAh and taking up as it were 1.5 KB of memory. It is exceptionally difficult to assault, so it secures well against cryptographic dangers. This makes it a incredible choice for apps that need to prioritize security without losing speed.

The table (2) shows that each method has a different mix of speed, resource use, and security. This means that IoT makers can pick the best encryption option for their needs and limitations.

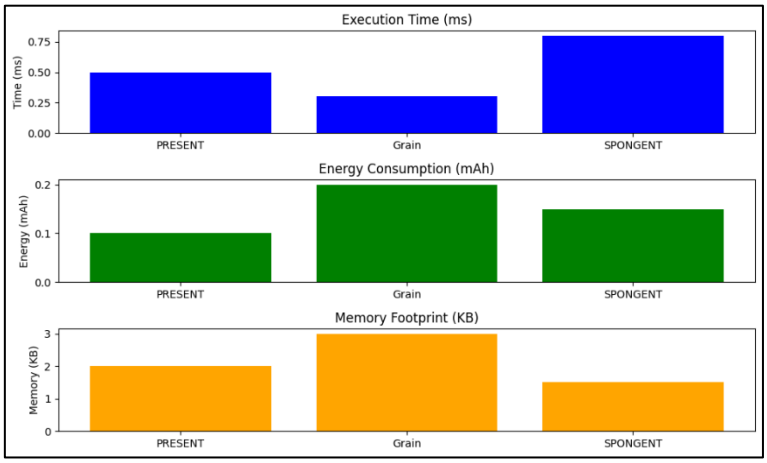

Figure 2: Representation of performance evaluation of various cryptographic algorithms

Table 3: Performance evaluation of various Lightweight Block Cipher

Lightweight Block Cipher	Execution Time (ms)	Energy Consumption (mAh)	Memory Footprint (KB)	Resistance to Attacks
PRESENT	0.5	0.1	2	High
LED	0.4	0.12	1.8	Medium
Simon	0.6	0.15	2.2	High

We look at the speed measures and security features of these lightweight block ciphers in this table (3) comparison.

With a brief handling time of 0.5 ms, Display can secure or decode information rapidly and effectively, and it as it were employments 0.1 mAh of control. With a memory estimate of 2 KB, it strikes a great blend between speed and asset utilize. It is exceptionally difficult to assault, which suggests it has solid security against cryptographic blemishes and can be utilized for apps that require strict security. The LED option has a faster processing time of 0.4 ms and slightly more energy use at 0.12 mAh. It can encrypt or decode data faster than the PRESENT option. Its small memory size of 1.8 KB shows that it makes good use of its resources. The medium level of resistance to threats in LEDs suggests a balance between performance and security, making them ideal for uses with middling security needs. Simon has a longer processing time of 0.6 ms, but it has strong security features that make it hard to attack. Even though Simon uses a little more power (0.15

mAh) and takes up 2.2 KB of memory, it protects well against cryptographic flaws, so it can be used in situations where security is more important than speed.

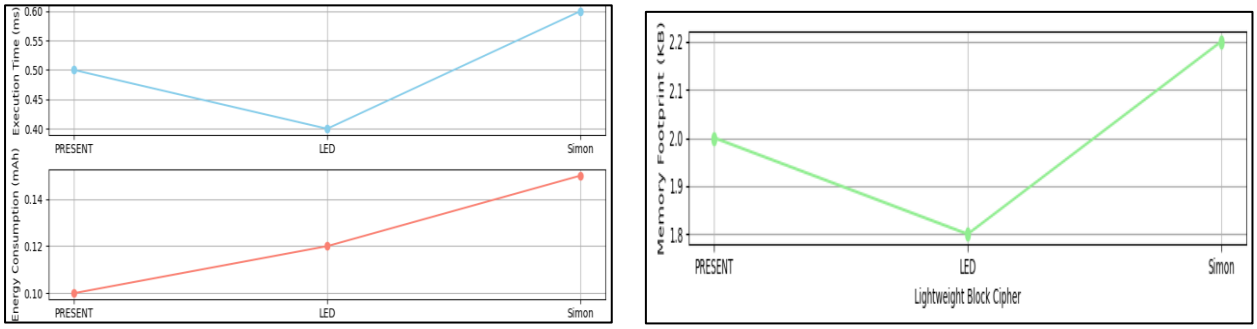

Figure 3: Representation of Performance evaluation of various Lightweight Block Cipher

The graph in figure (3) shows trends or links between factors over a wide range by connecting data points with straight lines. Each data point shows a different value of the factors being compared. The values are usually drawn along the horizontal (x) axis, and the measurements are shown along the vertical (y) axis. Based on the information given about lightweight block ciphers, the line graph shows how the processing time, energy use, and memory size of different block cipher methods are different. On the graph, each line shows a different performance metric. The x-axis shows the different block ciphers (PRESENT, LED, and Simon), and the y-axis shows the numbers of those performance metrics. People can easily tell which algorithms do better or worse on the chosen metrics by looking at the patterns and slopes of the lines. Most of the time, line graphs are the best way to see trends and compare different sets of data, which helps you understand how different things work or what their traits are.

Table 4: Performance evaluation of various Lightweight Stream Cipher

Lightweight Stream Cipher	Execution Time (ms)	Energy Consumption (mAh)	Memory Footprint (KB)	Resistance to Attacks
Trivium	0.3	0.05	1.2	High
Grain	0.25	0.06	1.5	Medium
Salsa20	0.35	0.07	1.8	High

In this table (4), we compare lightweight stream ciphers by looking at their security and performance: Trivium offers fast encryption with low power use, with a processing time of 0.3 ms and a power consumption of 0.05 mAh. Its small memory size of 1.2 KB shows how light it is, making it good for devices with limited resources. Trivium is very hard to attack, which means it has strong protection against cryptographic risks. This keeps data safe and private in vulnerable apps. With an execution time of as it were 0.25 ms, grain is indeed speedier, making it culminate for handling information in genuine time in IoT settings. It employments a small more control (0.06 mAh), but its little memory measure (1.5 KB) makes up for it, shown in figure 4. Grain's direct resistance to assaults implies that it ought to be secure for most employments, in spite of the fact that additional steps may have to be be taken for exceptionally private information.

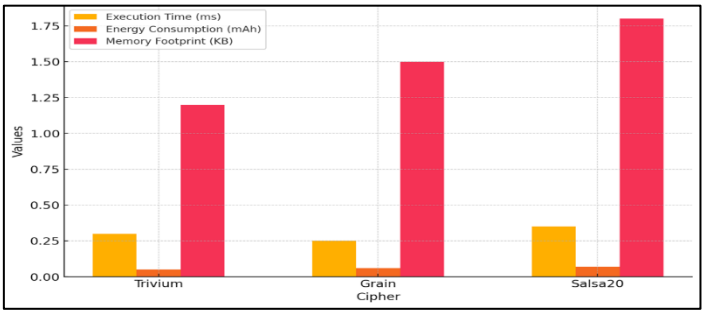

Figure 4: Comparison of different lightweight stream cipher

It incorporates a marginally longer preparing time of 0.35 ms, but it has great security and is very hard to hack. It employments 0.07 mAh of control and 1.8 KB of memory, which may be a great blend between speed and asset utilize. Salsa20's solid security highlights make it perfect for employments that put information precision and assurance to begin with, particularly in places where security is exceptionally vital.

5. Conclusion

The in-depth consider of lightweight cryptographic strategies gives us valuable data for making IoT gadgets more secure. It is conceivable for partners to select encryption arrangements that are best for IoT apps by looking at things like preparing time, vitality utilization, memory estimate, and the capacity to battle assaults. The investigate appeared that each sort of secure strategy, like a lightweight piece cipher or a stream cipher, has its claim speed and security highlights. Calculations like Display and Grain run quicker and utilize less vitality, which makes them great for real-time IoT apps that do not have a part of assets. On the other hand, strategies like SPONGENT utilize less memory and are difficult to hack, so they give solid security in a wide run of IoT settings. Including post-quantum cryptographic primitives and making blended cryptographic strategies see like great ways to create the Web of Things indeed more secure. By utilizing both lightweight calculations and more progressed security strategies, IoT gadgets can ensure against modern perils and make beyond any doubt that information exchanges are private, secure, and genuine. Overall, this study shows how important it is to think about many performance and safety factors when creating safe Internet of Things (IoT) systems. By using the right secure solutions, people involved in the IoT can make their gadgets safer and protect themselves from possible security holes in the IoT world, which is always growing.

References

[1] J. Singh, G. Singh and S. Negi, "Evaluating Security Principals and Technologies to Overcome Security Threats in IoT World," 2023 2nd International Conference on Applied Artificial Intelligence and Computing (ICAAIC), Salem, India, 2023, pp. 1405-1410

[2] G. Singh, "Internet of Things (Iot): A Review", Turkish Journal of Computer and Mathematics Education (TURCOMAT), vol. 12, no. 2, pp. 521-526, 2021.

[3] M.A. Sadeeq, S.R. Zeebaree, R. Qashi, S.H. Ahmed and K. Jacksi, "Internet of Things security: a survey", 2018 International Conference on Advanced Science and Engineering (ICOASE), pp. 162-166, October 2018.

[4] P.C. Van Oorschot and S.W. Smith, "The internet of things: security challenges", IEEE Security & Privacy, vol. 17, no. 5, pp. 7-9, 2019.

[5] J. Singh, G. Singh and G. Aggarwal, "Inclusion of Aerial Computing in Internet of Things: Prospects and Applications", 2022 Third International Conference on Intelligent Computing Instrumentation and Control Technologies (ICICICT), pp. 1664-1669, August. 2022.

[6] K. Sha, W. Wei, T.A. Yang, Z. Wang and W. Shi, "On security challenges and open issues in Internet of Things", Future generation computer systems, vol. 83, pp. 326-337, 2018.
[7] O. Yousuf and R.N. Mir, "A survey on the Internet of Things security: State-of-art architecture issues and countermeasures", Information & Computer Security, 2019.
[8] W. Zhou, Y. Jia, A. Peng, Y. Zhang and P. Liu, "The effect of IoT new features on security and privacy: New threats existing solutions and challenges yet to be solved", IEEE Internet of Things Journal, vol. 6, no. 2, pp. 1606-1616, Apr. 2019.
[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] P. Abdul Hafeez, G. Singh, J. Singh, C. Prabha and A. Verma, "IoT in Agriculture and Healthcare: Applications and Challenges", 2022 3rd International Conference on Smart Electronics and Communication (ICOSEC), pp. 446-450, 2022.
[11] Kale, Rohini Suhas , Hase, Jayashri , Deshmukh, Shyam , Ajani, Samir N. , Agrawal, Pratik K & Khandelwal, Chhaya Sunil (2024) Ensuring data confidentiality and integrity in edge computing environments : A security and privacy perspective, Journal of Discrete Mathematical Sciences and Cryptography, 27:2-A, 421–430, DOI: 10.47974/JDMSC-1898
[12] Dari, Sukhvinder Singh , Dhabliya, Dharmesh , Dhablia, Anishkumar , Dingankar, Shreyas , Pasha, M. Jahir & Ajani, Samir N. (2024) Securing micro transactions in the Internet of Things with cryptography primitives, Journal of Discrete Mathematical Sciences and Cryptography, 27:2-B, 753–762, DOI: 10.47974/JDMSC-1925
[13] Limkar, Suresh, Singh, Sanjeev, Ashok, Wankhede Vishal, Wadne, Vinod , Phursule, Rajesh & Ajani, Samir N. (2024) Modified elliptic curve cryptography for efficient data protection in wireless sensor network, Journal of Discrete Mathematical Sciences and Cryptography, 27:2-A, 305–316, DOI: 10.47974/JDMSC-1903
[14] T.T.A. Dinh, R. Liu, M. Zhang, G. Chen, B.C. Ooi and J. Wang, "Untangling blockchain: A data processing view of blockchain systems", IEEE transactions on knowledge and data engineering, vol. 30, no. 7, pp. 1366-1385, 2018.
[15] Arpit Chaudhari, Prachi Jaini, “Stealthier attack on zone routing protocol in wireless sensor network”, 2014 Fourth International Conference on Communication Systems and Network Technologies, Pages, 734-738, Publisher, IEEE
[16] G. Singh and J. Singh, "A Fog Computing based Agriculture-IoT Framework for Detection of Alert Conditions and Effective Crop Protection", 2023 5th International Conference on Smart Systems and Inventive Technology (ICSSIT), pp. 537-543, 2023.
[17] V. Mahajan and J. Singh, "Performance Analysis of Honeypots Against Flooding Attack", 2022 6th International Conference on Electronics Communication and Aerospace Technology, pp. 01-06, December 2022.
[18] J. Singh, S. Agarwal, P. Kumar, D. Rana and R. Bajaj, "Prominent features based chronic kidney disease prediction model using machine learning", 2022 3rd International Conference on Electronics and Sustainable Communication Systems (ICESC), pp. 1193-1198, August 2022.
[19] K. Agnihotri, P. Chilbule, S. Prashant, P. Jain and P. Khobragade, "Generating Image Description Using Machine Learning Algorithms," 2023 11th International Conference on Emerging Trends in Engineering & Technology - Signal and Information Processing (ICETET - SIP), Nagpur, India, 2023, pp. 1-6, doi: 10.1109/ICETET-SIP58143.2023.10151472.
[20] M. Conti, A. Dehghantanha, K. Franke and S. Watson, "Internet of Things security and forensics: Challenges and opportunities", Future Generation Computer Systems, vol. 78, pp. 544-546, 2018.
[21] Prashant Khobragade, Latesh G. Malik,“A Review on Data Generation for Digital Forensic Investigation using Datamining”, IJCAT International Journal of Computing and Technology, Volume 1, Issue 3, April 2014.
[22] A. N. Doss, D. Shah, G. F. Smaisim, M. Olha and S. Jaiswal, "A Comprehensive Analysis of Internet of Things (IOT) in Enhancing Data Security for Better System Integrity - A Critical Analysis on the Security Attacks and Relevant Countermeasures," 2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), Greater Noida, India, 2022
[23] Nasr Abosata, Saba AI-Rubaye, Gokhan Inalhan and Christos Emmanouilidis, "Internet of Things for System Integrity: A Comprehensive Survey on Security", Attacks and Countermeasures for Industrial Applications. Sensors (Basel), vol. 21, no. 11, pp. 3654, Jun 2021.
[24] E. Aydogan, S. Yilmaz, S. Sen, I. Butun, S. Forsström and M. Gidlund, "A central intrusion detection system for rpl-based industrial internet of things", Proceedings of the 2019 15th IEEE International Workshop on Factory Communication Systems (WFCS), pp. 1-5, 27–29 May 2019.
[25] Y.B. Zikria, S.W. Kim, O. Hahm, M.K. Afzal and M.Y Aalsalem, "Internet of things (iot) operating systems management: Opportunities challenges and solution", Sensors, vol. 19, pp. 1793, 2019.