A Comparative Study on Lightweight Authentication Protocols in IoT context

Tijmen van de Meent
University of Twente
P.O. Box 217, 7500AE Enschede
The Netherlands
t.j.vandemeent@student.utwente.nl

ABSTRACT

With the growth of the Internet of Things, the number of resource-constrained devices connected to the internet grows vastly. These devices are constrained in energy usage, processing, memory, storage, etc. The advancement in the comfort, several security challenges come up. Many researchers have attempted to address such issues. Every IoT device must be able to identify and authenticate other IoT devices. However, due to the nature of the IoT, this process can be quite difficult. To provide authentication in resource-constrained environments, in recent years lightweight authentication protocols have been proposed. These lightweight protocols are targeted at resource-constrained devices and provide less heavy cryptography than conventional protocols. However, lightweight protocols have not been investigated as thoroughly as conventional protocols. In this research we study and benchmark several lightweight protocols. We identify key metrics for lightweight cryptographic protocols and have collected data regarding these metrics from the literature. The metrics that the protocols will be evaluated and benchmarked on are memory usage, gate area, latency, throughput, and energy consumption. The result is a ranking of protocol performance for each of the metrics, as well as some possibilities for future research that is related.

Keywords

Resource-constrained devices, Authentication, Internet of Things

1. INTRODUCTION

In the past few years, the number of devices connected to the internet has grown immensely. A growing portion of these devices could be categorized as resource-constrained devices, devices that are constrained in terms of memory, processing, storage, bandwidth, and power, to name a few [22, 30, 33]. A parking garage sensor that connects to a central server which calculates the number of available spaces, or Wi-Fi trackers that send MAC addresses over the internet in real time is an example of such resource-constraint devices.

These IoT end-devices are often operating in vulnerable environments, which leads to several security challenges that should be taken into consideration. Authentication is considered an important feature for ensuring reliable and secure communication between devices in an IoT environment. However, traditional authentication methods based on traditional cryptographic protocols might not be suitable to be implemented on resource-limited IoT devices. Computing and resource requirements might have to include the lightweight requirements for protocols in order to provide an algorithm that fits the characteristics of different components in IoT systems.

Over the years, as lightweight cryptography increased in relevance, several papers have been published concerning these lightweight protocols, their characteristics, and their requirements [16, 17, 18, 22, 28, 32, 33, 36, 39, 45]. However, these publications compared established protocols without reviewing protocols that have been published in more recent years. This is especially relevant since the National Institute of Standards and Technology (NIST) of the United States is currently finalizing a standardization procedure for lightweight cryptography[42] protocols. In the research mentioned, none of the algorithms in this competition are reviewed, despite the fact that the competition will provide a standard that will be widely used in the coming years.

Providing an analysis including the finalists of the NIST competition is vital to provide standardized protocols that can benefit both researchers and developers. In this study we will provide a benchmark of all the available protocols both in literature and the final list provided by NIST based on several metrics that measure the lightweight features of these protocols.

We will do this by answering the following question:

> How can new authentication schemes be customized for resource-constrained devices in IoT applications?

To answer this question, we have defined the following research questions (RQ):

RQ1: What lightweight and ultra-lightweight authentication methods exist?

RQ2: What are properties of these authentication methods?

RQ3: What are advantages and disadvantages of these authentication methods compared to one another?

It is hypothesized that authentication schemes should be adapted to include lightweight authentication protocols, in

36th Twente Student Conference on IT Febr. 4th, 2022, Enschede, The Netherlands.

order to be better suited for resource-constrained devices in IoT applications. Moreover, it is expected that this research finds suitable protocols for several constrained environments, and that the research results in a protocol recommendation based on such constraints.

The rest of this paper will set out to answer the questions posed above. In section 2 we will start with explaining the security challenges that are especially relevant for lightweight authentication. Following the security challenges, works related to this research are presented, as well as some lightweight authentication protocols, in section 3. These lightweight authentication protocols will be analyzed and discussed in section 4, based on several metrics that were collected from works that can be found in the related works section. Following this analysis, in subsection 4.7 we discuss the methodology and results of this research. Lastly, in section 5 we will answer the research questions and conclude the paper, followed by ideas for future work.

2. SECURITY CHALLENGES

In this section, we consider the most basic architecture of IoT (three-layer architecture)[35], and discuss the security concerns, attacks and security requirements at each layer of the architecture.

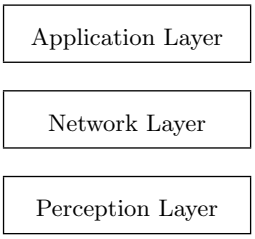

Figure 1: IoT Architecture Model

- **Perception Layer Security Issues and Requirements** The perception layer consists of sensors that are characterized by limited processing power and storage capacity[46]. Several security issues and attack risks rise due to such limitations. Several attacks on the perception layer are noticed: *Node capture, Denial of Service(DoS), Distributed Denial of Service (DDoS), Sybil, Replay, etc.* Taking into consideration the above mentioned risks, there is a need for node authentication to prevent fake node and illegal access, in addition to the need for data encryption to protect the confidentiality of data while being transmitted between nodes (end node, gateway or server). Due to the properties of the nodes with respect to the shortage of power and the limited storage capacity, there is a necessity for mature lightweight security schemes that include both lightweight cryptographic algorithms and security protocols[21].

- **Network Layer Security Issues and Requirements** The network layer is in charge of the diffusion of data from the perception layer to the application layer. This is where data routing occurs as well as the primary data analysis. In this layer, several network technologies are used such as the different technologies for mobile communication generations (2G, 3G, 4G and 5G) and wireless networks (Bluetooth, WiMAX, WiFi, LoRaWAN, etc.) Several attacks and risks on the network layer are identified: *DoS, Routing, Eavesdropping/sniffing, etc.*. These potential attacks at the network layer (wired or wireless) lead to the definition of the following security requirements: Hop-to-Hop Encryption, Point-to-Point Authentication, Key Agreement and Management, Security Routing and Intrusion Detection [21].

- **Application Layer Security Issues and Requirements** The application layer is responsible for providing services. It hosts a set of protocols for message passing [26], e.g. Constrained Application Protocol (COAP), Message Queuing Telemetry Transport (MQTT), Extensible Messaging and Presence Protocol (XMPP), Advanced Message Queuing Protocol (AMQP), etc. This layer directly interacts with the user. Given that the "traditional" application-layer protocols do not perform well within IoT, and since the IoT does not have its own international standards, several security issues arise at the application layer[29], which are related to the following aspects: Data Authentication, Data privacy, Availability of big data, etc. Regarding the security requirements for the application layer, authentication is required while protecting the privacy of users (respectively, data). In addition, there should be an information security management scheme that includes resource management and physical security information management.

3. LIGHTWEIGHT AUTHENTICATION PROTOCOLS AND RELATED WORK

It is important to note that lightweight authentication protocols should not be held to the same standard as heavyweight protocols. Therefore, in this section we discuss what is to be considered a lightweight cryptographic protocol. Moreover, we discuss previous works in order to define the research gap that this research will fill.

3.1 Lightweight Authentication Protocols

To ensure that all protocols that will be analyzed are lightweight protocols, we set up several requirements, based on the literature.

- Lightweight protocols should be targeted at resource-constrained devices [30]. As shown in Table 1, this concerns (among others) embedded systems, RFID and sensor networks.

- The cost metrics of lightweight protocols should be lower than those of conventional protocols. RAM and ROM usage and Gate Area are the cost metrics we will look at, as these are the most relevant metrics for reviewing protocol costs on resource-constrained devices [30].

- The performance metrics of lightweight protocols should be sufficient. The performance could be measured in several ways, so we chose the most important and relevant metrics. These are latency, throughput, and energy consumption [30].

The aforementioned metrics will be discussed in more detail in section 3.

3.2 Related Works

In the literature we found several reviews of lightweight cryptographic algorithms. These reviews contained useful information for this research, mainly in the form of performance and cost metrics for several protocols. Additionally, literature was collected on the proposals for several

Servers and Desktops	Conventional
Tablets and Smartphones	cryptography
Embedded Systems	Lightweight
RFID and Sensor Networks	cryptography

Table 1: Device Spectrum [30]

'Standard' protocols	NIST finalists
AES	ASCON
HIGHT	Elephant
KATAN	GIFT-COFB
KTANTAN	Grain128-AEAD
LEA	ISAP
Piccolo	Photon-Beetle
PRESENT	Romulus
PRINCE	Sparkle
QARMA	TinyJambu
RECTANGLE	Xoodyak
SIMON	
SPECK	
TWINE	
XTEA	

Table 2: Lightweight authentication protocols in literature

lightweight cryptographic protocols. Some of these protocols were included because they were mentioned frequently in literature and therefore metrics were readily available [22, 36], others were included as they are the finalists in the NIST Lightweight Cryptographic Standardization process [42]. This division in protocols is shown in Table 2.

To the best of our knowledge, no study has been conducted to survey and analyze all available lightweight authentication protocols presented in the literature. Therefore, we decided to investigate and analyze the most recent lightweight authentication protocols elected by NIST [42].

Related protocol reviews consider many different metrics [7, 16, 22, 36, 39]. In this survey, we will consider the following metrics:

3.2.1 RAM and ROM occupation

Random Access Memory (RAM) and Read-Only Memory (ROM) are the most impacted cost metric for software implementations of cryptographic protocols. RAM is used to store and collect intermediate states for the protocols during and between cryptographic cycles [16], whereas ROM is used to store the compiled algorithm code, parameters, and constants [28]. These are important, since software implementations are stored on chips using RAM and ROM. During execution, the micro-controller communicates with the RAM and ROM to execute the algorithm. Ultimately, therefore, these metrics decide how much RAM and ROM capacity a chip must have in order to use the respective protocol. For this metric, lower is better.

3.2.2 Gate Area

Gate area or 'design area' measures the complexity of a hardware implementation for the respective protocol [33]. It is usually expressed in gate equivalents (GE), which can be converted to physical area (usually in μm^2) based on several constants for specific hardware platforms [36]. A protocol with a higher gate area requires more material to be manufactured, and therefore is more costly. For this metric, lower is better.

3.2.3 Latency

In cryptography, latency can be described as the time it takes from the start of the algorithm to the generation of the ciphertext [22, 36]. It is measured in clock cycles. For this metric, lower is better.

3.2.4 Throughput

Throughput is a measure of the amount of data a protocol can generate over time [33, 36, 22]. Combined with the latency, it shows the speed at which the protocol can operate [33]. Throughput depends on the frequency of the platform the protocol is implemented on, however normalizing the throughput negates this issue. For this metric, higher is better.

3.2.5 Energy Consumption

An important metric when considering protocols for resource-constrained devices is energy consumption [16]. In some environments, these devices might operate from a battery [33]. Especially in those situations, the energy consumption of a protocol is important. For this metric, lower is better.

3.3 NIST Lightweight Cryptography Standardization

In 2017, the NIST started the call for papers for their Lightweight Cryptography Standardization process [30]. As the publication describes, the NIST most often develops standards using one of three different approaches.

The first approach is a competition. This approach has been used for several years; for example, the AES cipher was chosen by using this approach [11], as well as the SHA-3 hash function [15]. However, this format takes place over many years: the competition for AES started in 1996, when the NIST announced the development of a new cryptographic standard [27], and ended in 2001 with the publication of the Advanced Encryption Standard (AES) [11].

The second approach is to adapt and revise standards from other standard development organizations. This was the case for the standardization of HMAC [44].

The last approach is in-house development. As [30] describes, the NIST has researchers who work together with experts in the work field to develop standards and guidelines if no suitable standard already exists.

For the Lightweight Cryptography Standardization Project, the NIST has been through 2 rounds of a protocol collection process. The first round started with 57 candidates after an initial review of all submissions. It promoted 32 protocols to the second round after reviewing cryptoanalysis and the maturity of the candidates [43]. In the second round, cryptographic security, performance, and side channel resistance were reviewed for all candidates, which resulted in a list of 10 finalists. Those 10 finalists can be seen in Table 2 and are included in the analysis of the next section.

3.4 Data

Using the reviews in literature, we found a lot of metric data for the mentioned protocols. This data can be found in Table 9.

4. PERFORMANCE ANALYSIS AND DISCUSSION

In this section we will analyze the protocols. To achieve a sound analysis, we will compare and analyze the metrics we collected in Table 9. For each metric, we will consider several data sources. To ensure that data is properly comparable, the data will first be reviewed for each data source. Some reviews, however, only discuss relevant metrics for a single protocol in Table 2. Since comparing the metrics from such reviews with those of other reviews can't be done without conversion and careful consideration of all environmental factors, in this research we will exclude them from the comparison.

4.1 RAM and ROM usage

In the literature, several papers contained data regarding RAM occupation [4, 16, 22, 28, 37, 47]. From these reviews, we will consider the metrics as produced in [16], [22] and [37]. These metrics are visualized in Figure 2.

In addition, there were several papers writing about the ROM occupation for lightweight cryptographic protocols [4, 8, 22, 28, 37]. Here as well, however, were a few that only contained metric data for a single protocol. The data for ROM occupation that we compare can be seen in Figure 3.

When looking at Figure 2 and Figure 3, it is clear that a few algorithms perform better when compared to others. Clearly, when looking at the results of Hatzivasilis [22], the SIMON algorithm is by far the most cost-friendly regarding the memory occupation. Furthermore, we can create a ranking when adding both metrics together for the 2 reviews that have data for both separate metrics. The results are shown in Table 3.

Combining the results of [22] and [37] is hard, if not impossible, because [22] only specifies that the data concerns implementations for 8-bit microcontrollers. However, no more details are given to the specifics. Therefore, we will let these results remain as it is.

We ranked the protocols based on the computed metric in Table 3 from lowest to highest value. It shows that SIMON, KATAN and SPECK use significantly less RAM and ROM than other protocols when comparing them to the non-NIST finalists. From the list of NIST finalists, ASCON, TinyJAMBU and Xoodyak perform significantly better than other finalists.

Protocol	RAM + ROM (bytes)
SIMON [22]	246
KATAN [22]	356
SPECK [22]	396
HIGHT [22]	1138
Piccolo [22]	1242
TWINE [22]	1467
AES [22]	2869
PRINCE [22]	4363
ASCON [37]	8408
TinyJAMBU [37]	8420
Xoodyak [37]	8540
GIFT-COFB [37]	12360
ISAP [37]	12888
Elephant [37]	14380
PHOTON-Beetle [37]	14852
Romulus [37]	15104
Grain-128AEAD [37]	23681

Table 3: Code Size

4.2 Gate Area

Multiple papers in the literature contained data regarding the Gate Area for lightweight protocols [1, 3, 17, 36, 38, 45]. We excluded [3] and [17] since they contained data for only a single protocol.

The data gathered from [1] was normalized to only a fraction of an expected value for the gate area. Luckily, both [1] and [38] provide a value for the gate area of ASCON. Therefore, we recreated the original values for the gate area of all other protocols in [1], by multiplying the values with the ratio of the value for ASCON in [38] to the same metric value in [1] ($\frac{GA_{ASCON,Rezvani}}{GA_{ASCON,Aagard}}$). The results are visualized in Figure 4.

It can be observed that TinyJAMBU has the smallest gate area of the protocols that were reviewed. Moreover, QARMA stands out as the protocol with the highest gate area, with a gate area more than 1.8 times larger than the second worst scoring protocol, SPARKLE.

As the Gate Area is not platform specific, we can compare the metric values from all 4 reviews. To do so, we take the lowest value for each protocol. The result is shown in Table 4

Protocol	Gate Area (GE)
TinyJAMBU [1]	496,79
KTANTAN [36]	688
SIMON [36]	763
Romulus [1]	1070,01
Piccolo [36]	1260
PHOTON-Beetle [1]	1413,95
ISAP [1]	1515,85
PRESENT [45]	1570
RECTANGLE [45]	1600
ASCON [38]	1898
GIFT-COFB [38]	1932
Xoodyak [1]	1987,17
Elephant [1]	2190,98
AES [36]	2400
HIGHT [36]	3048
PRINCE [36]	3491
SPARKLE [38]	4313
QARMA [36]	7844

Table 4: Gate Area, smaller is better

4.3 Latency

In several reviews in the literature latency is calculated or measured [36, 18, 8, 38, 22]. We review 3 of those, as the other 2 only contain data regarding a single protocol. Moreover, the data that remains still has metric values for the respective protocols.

In Figure 5, the metric data is shown. We can observe an overlap in data, as several protocols have multiple values for different metrics. This is due to the difference in implementation, as will be discussed in subsection 4.7. Since latency is independent of technology [20], we will

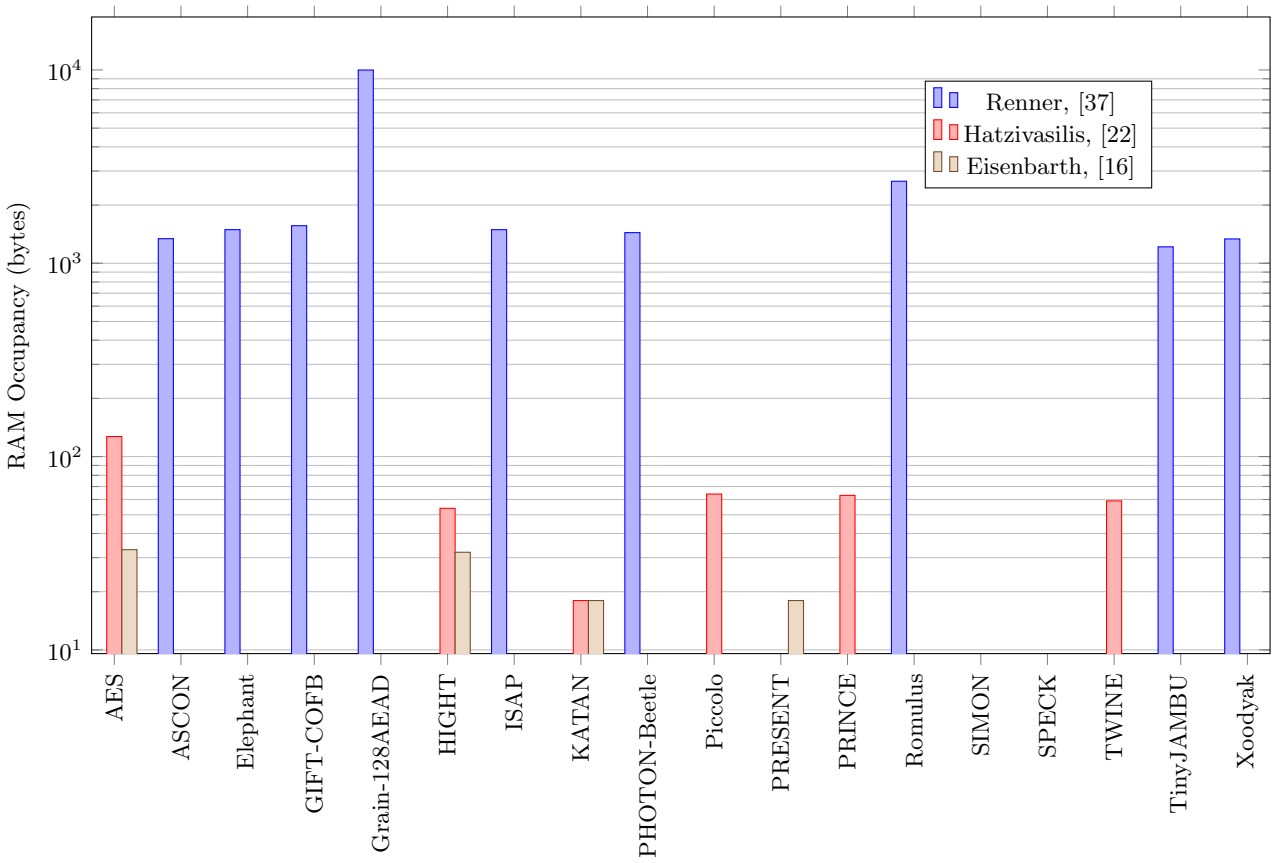

Figure 2: RAM Occupation

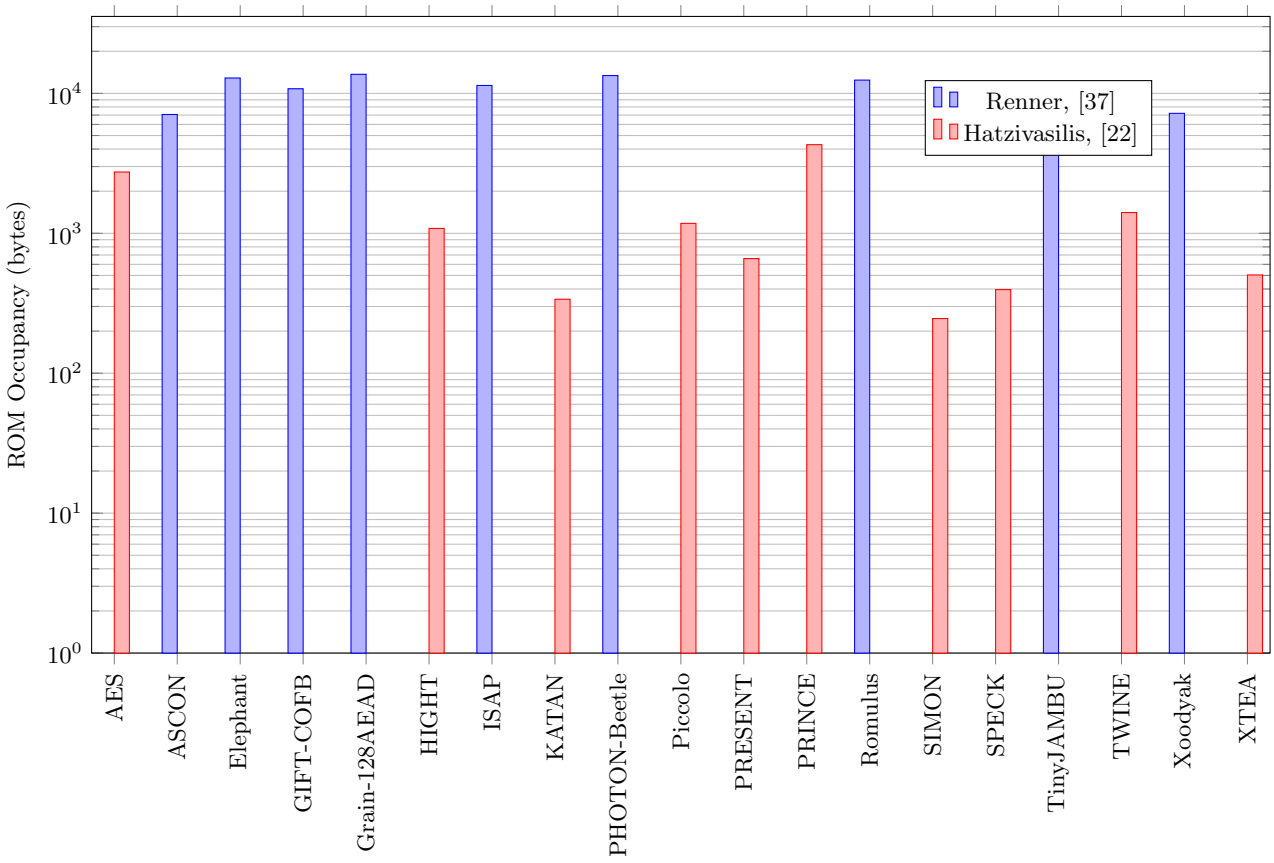

Figure 3: ROM Occupation

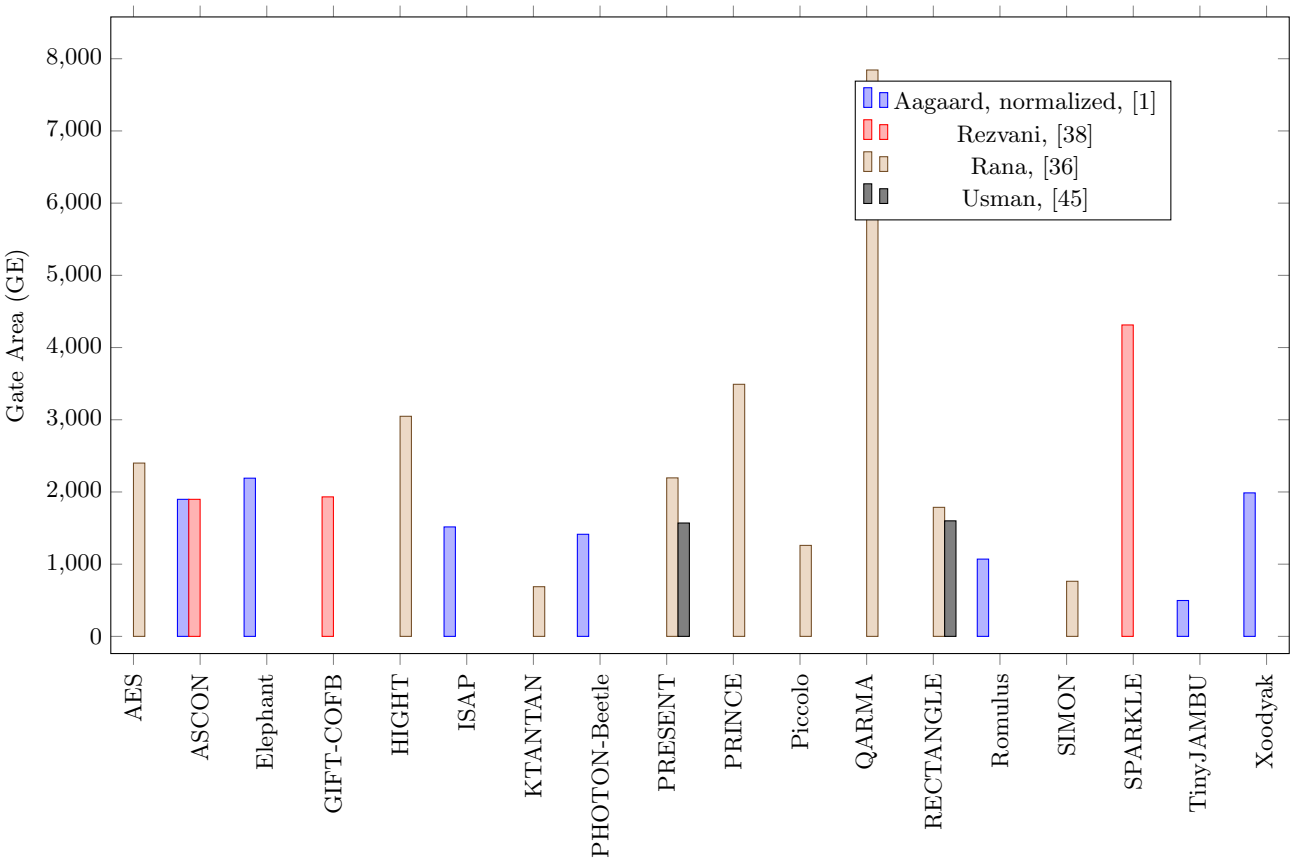

Figure 4: Gate Area

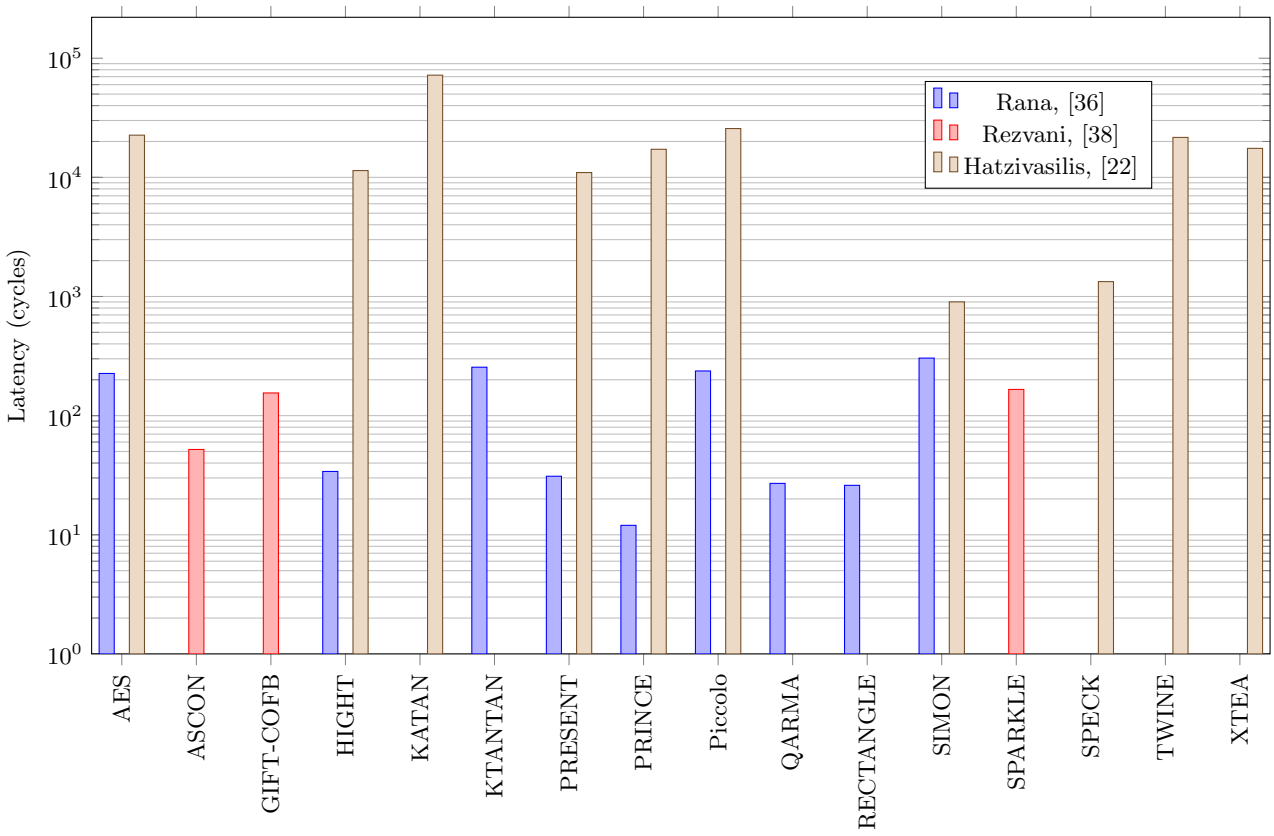

Figure 5: Latency

consider the best results for each protocol. As the data demonstrates, there is a big difference in latency across the protocols and platforms. For some protocols, the latency equals the number of rounds used in the implementation (PRINCE, PRESENT, QARMA), or perhaps 1 or 2 cycles more than the number of rounds (RECTANGLE, HIGHT). Consequently, this is reflected by the ranked results in Table 5.

Protocol	Latency (cycles)
PRINCE [36]	12
RECTANGLE [36]	26
QARMA [36]	27
PRESENT [36]	31
HIGHT [36]	34
ASCON [38]	52
GIFT-COFB [38]	155
SPARKLE [38]	166
AES [36]	226
Piccolo [36]	237,04
KTANTAN [36]	255
SIMON [36]	304
SPECK [22]	1333
XTEA [22]	17514
TWINE [22]	21637
KATAN [22]	72063

Table 5: Latency, smaller is better

4.4 Throughput

As Table 9 shows, a lot of data was collected when it comes to protocol throughput [1, 7, 8, 17, 18, 22, 36, 38, 45]. However, since we want to compare the values, several of the researches must be discarded, as the platform frequency was not specified. The frequency is an important factor in converting throughput for comparison, and therefore it is of the utmost importance to know the value.

The data that remains originates from [1], [7], [17], [45], [18] and [38]. These values are plotted in Figure 6. As the figure shows, the implementation and platform have a huge difference on the throughput. Moreover, Xoodyak and ASCON have the highest throughput, with a throughput that is more than 4 times higher than the throughput of Elephant, the third highest. This is reflected in Table 6, where the protocols are ordered based on their throughput.

4.5 Energy Consumption

Several sources in the literature contain metric values for the energy consumption of lightweight cryptographic protocols [1, 3, 8, 16, 22, 47]. Of these 6 sources only 3 will be used for the comparison, since 3 of the sources only provide data for a single protocol and couldn't be compared. The sources used are [1], [16] and [22].

As can be seen in Figure 7, the NIST protocols are covered by [1], whereas the other protocols are covered by [16] and [22]. There is no overlap between the two; consequently a proper comparison cannot be made between all of the protocols. Regardless, both ASCON and TinyJAMBU perform equally well out of the NIST finalists, with Xoodyak as a close 3rd. PHOTON-Beetle performs the worst out of the protocols, consuming more than 8 times the energy that ASCON and TinyJAMBU use. Out of the other protocols, SIMON and SPECK perform well, with a distinct difference between them and the 3rd ranked protocol.

Protocol	Throughput (Mbps)
Xoodyak [1]	1102,8
ASCON [1]	1024
Elephant [1]	290,8
Romulus [1]	284,4
SPARKLE [38]	196,92
TinyJAMBU [1]	160
PHOTON-Beetle [1]	155,2
ISAP [1]	136,8
RECTANGLE [45]	98,4
GIFT-COFB [38]	96,6
PRESENT [17]	80
HIGHT [17]	75,28
SIMON [18]	58,18
AES [17]	4,96
Grain-128AEAD [7]	3,542

Table 6: Throughput at 40 MHz, higher is better

The full protocol ranking can be found in Table 7, where the best result from either [1], [16], or [22] was taken for each protocol. Moreover, the table shows the distinction between the NIST-finalists and the other protocols.

Protocol	Energy Consumption (nJ)
ASCON [1]	0.29
TinyJAMBU [1]	0.29
Xoodyak [1]	0.4
Elephant [1]	0.88
Romulus [1]	0.92
ISAP [1]	1.39
PHOTON-Beetle [1]	1.99
SIMON [22]	3600
SPECK [22]	5300
AES [16]	19200
PRESENT [22]	43100
HIGHT [22]	45500
PRINCE [22]	68800
XTEA [22]	70000
TWINE [22]	86500
Piccolo [22]	102700
KATAN [16]	289200

Table 7: Energy Consumption, smaller is better

4.6 Summary

A summary of the results that were produced in the previous sections is in Table 8. For each metric it shows the ranking, from best to worst scoring protocol.

4.7 Discussion

This research depends on a large set of literature. In reviewing the literature, several points of critique could come up when performing this research:

- Difference in implementation for data points
- Measured or calculated values
- Normalization of data

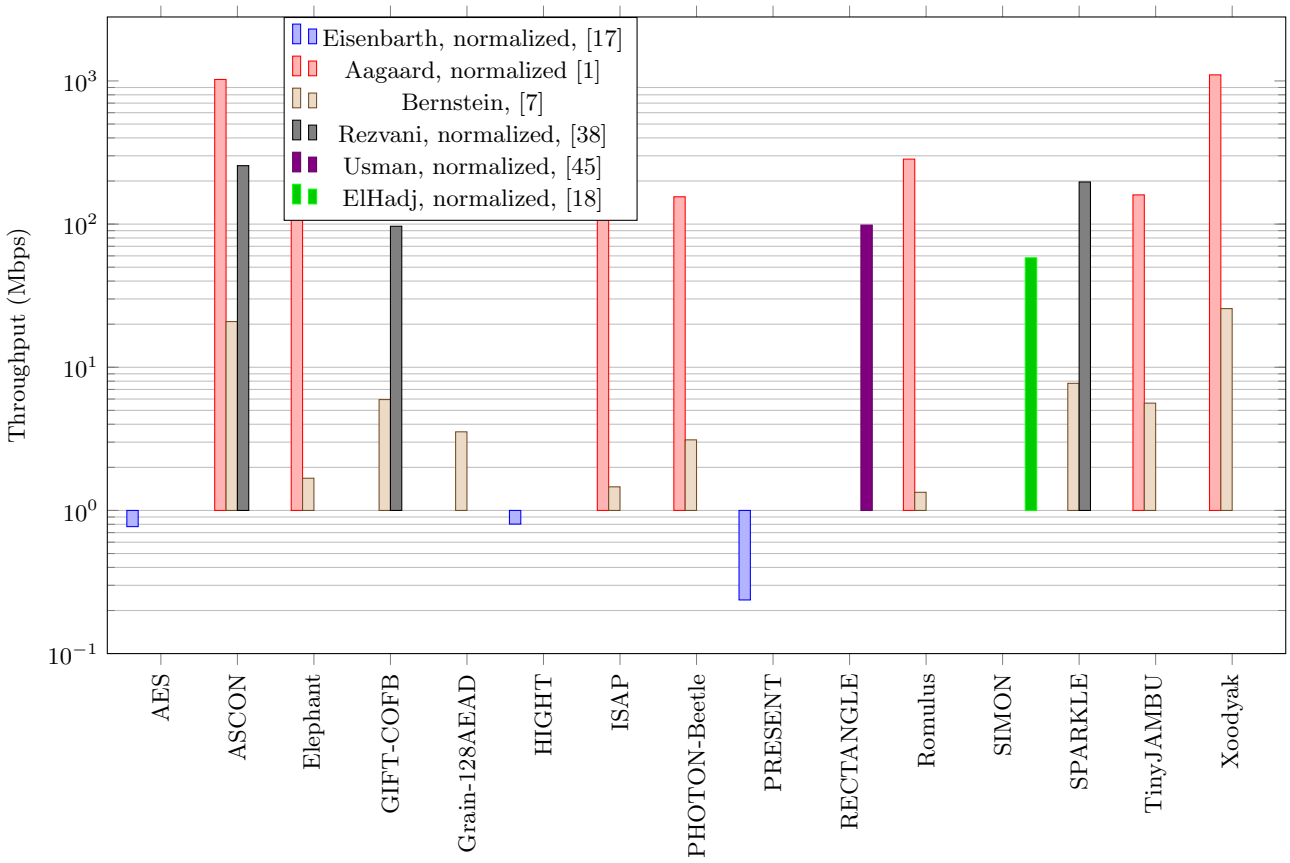

Figure 6: Throughput at 40 MHz

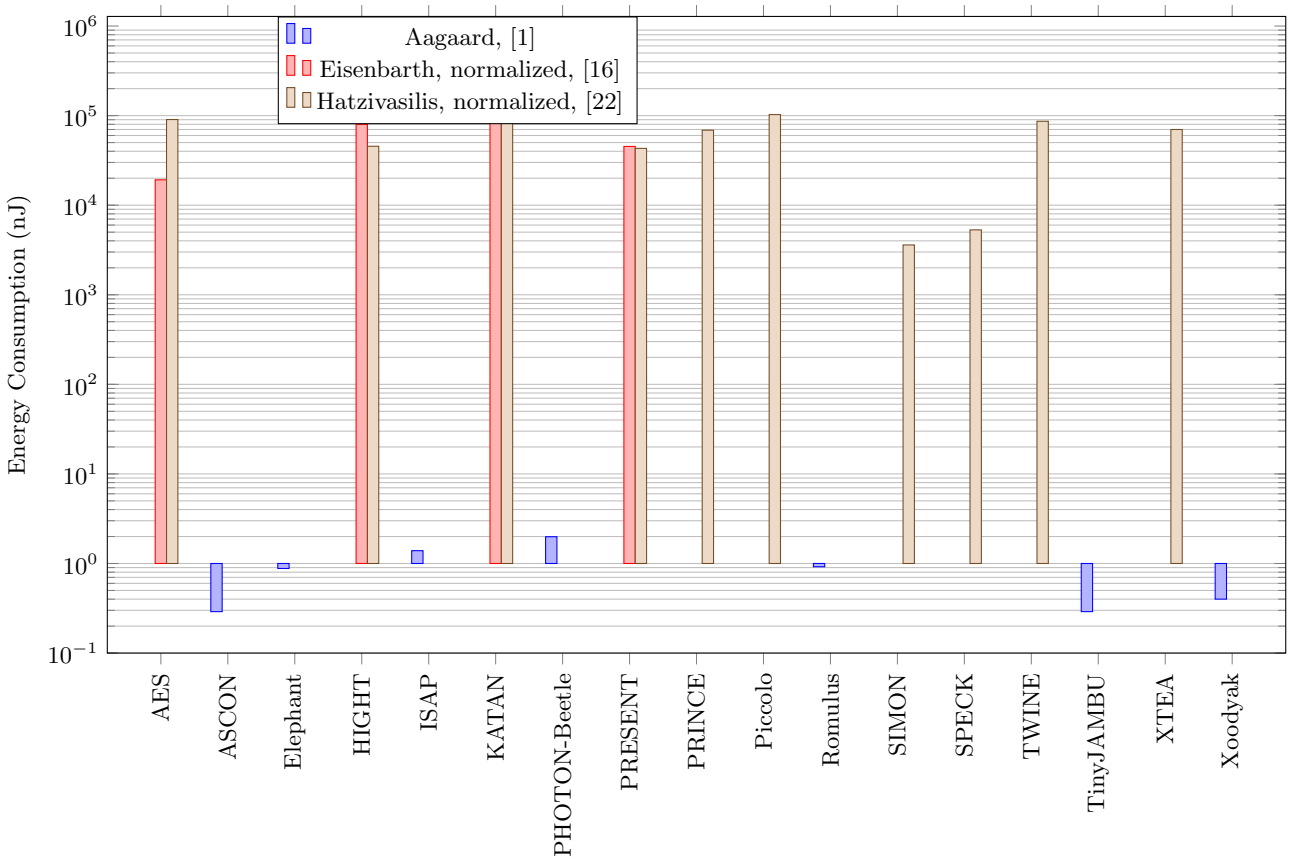

Figure 7: Energy Consumption

RAM & ROM	Gate Area	Latency	Throughput	Energy Consumption
SIMON	TinyJAMBU	PRINCE	Xoodyak	ASCON
KATAN	KTANTAN	RECTANGLE	ASCON	TinyJAMBU
SPECK	SIMON	QARMA	Elephant	Xoodyak
HIGHT	Romulus	PRESENT	Romulus	Elephant
Piccolo	Piccolo	HIGHT	SPARKLE	Romulus
TWINE	PHOTON-Beetle	ASCON	TinyJAMBU	ISAP
AES	ISAP	GIFT-COFB	PHOTON-Beetle	PHOTON-Beetle
PRINCE	PRESENT	SPARKLE	ISAP	SIMON
ASCON	RECTANGLE	AES	RECTANGLE	SPECK
TinyJAMBU	ASCON	Piccolo	GIFT-COFB	AES
Xoodyak	GIFT-COFB	KTANTAN	PRESENT	PRESENT
GIFT-COFB	Xoodyak	SIMON	HIGHT	HIGHT
ISAP	Elephant	SPECK	SIMON	PRINCE
Elephant	AES	XTEA	AES	XTEA
PHOTON-Beetle	HIGHT	TWINE	Grain-128AEAD	TWINE
Romulus	PRINCE	KATAN		Piccolo
Grain-128AEAD	SPARKLE			KATAN
	QARMA			

Table 8: Ranking of the best performing algorithms per metric

4.7.1 *Implementation differences*

It is true that implementations differ greatly between reviews. Therefore, a direct comparison is not always possible. However, the goal of this research is to set out to find the best protocols for each metric. With that goal in mind, the difference in implementation will help for a protocol to perform better.

When a protocol performs well on all metrics, that does not make it the best protocol overall, because of the same reason. Therefore, we must be careful when drawing conclusions from this research. Choosing a protocol based on a single metric is possible; still, nothing can be guaranteed about the results for the other metrics in that case.

4.7.2 *Measured vs. Calculated Data*

In some cases, the literature provided data that was calculated instead of measured. This is not an issue, as the literature has thoroughly been studied and the methodology is sound. Therefore, we see no issue in using calculated data.

The same applies for measured data: the methodology for measuring the metrics has been thoroughly studied, and was shown to be correct.

4.7.3 *Data Normalization*

In some of the previous sections, metric data was normalized to allow for comparison. We have normalized the data with care to ensure that no mistakes were made. Moreover, the literature has been studied and it was shown that the normalization was done using a scientifically and mathematically correct method.

5. CONCLUSIONS AND FUTURE WORK

In this research, we explained the purpose of this paper, namely, to inform the reader on the state of lightweight encryption at the time of publication. Then we provided the reason why lightweight encryption, and more specifically authentication, is necessary. Moreover, we explained what lightweight authentication entails, and the literature that was relevant for this research. Lastly, we performed an analysis on the data that was gathered from the relevant works and provided a result in Table 8.

To conclude, it is impossible to tell which protocol is best overall. Consequently, in the next sections, we give a recommendation based on the environmental requirements that an application of the protocol might need.

5.1 Costs

As previously discussed, the cost of cryptographic protocols can be split up in 2 separate parts: the software costs and hardware costs.

When it comes to software costs, Table 3 shows that SIMON, KATAN or SPECK perform the best out of the non-NIST finalists, while PRINCE performs the worst of all by requiring more than 10 times the ROM + RAM space than that of SPECK. Out of the NIST finalists, ASCON, TinyJAMBU and Xoodyak have the lowest software cost, while Grain-128AEAD requires 2.7 times the memory space required for Xoodyak. The lower the software cost, the less memory a chip needs in order for it to support the respective protocol. Therefore, especially on devices that are extremely limited with regard to memory capacity, SIMON, KATAN, SPECK, ASCON, TinyJAMBU or Xoodyak are advised.

When it comes to hardware costs, it can be concluded from Table 4 that TinyJAMBU can be implemented on the smallest hardware chip, while QARMA needs the most surface area to be implemented using hardware techniques. When it comes to manufacturing a chip that should support an authentication protocol, it is advised to implement TinyJAMBU, as it is the most cost-effective protocol when it comes to hardware implementations.

5.2 Performance

As discussed in section 3, the performance was measured using 3 distinct metrics: latency, throughput, and energy consumption.

In an environment where time delays should be kept to a minimum, it is best to choose either PRINCE, RECTANGLE or QARMA, since their latency is lowest of the researched protocols as shown in Table 5.

Table 6 shows the results concerning the throughput of

the reviewed protocols. It demonstrates that Xoodyak and ASCON are the best scoring protocols when it comes to throughput, with a significant gap to the other protocols. These two protocols could best be used when the application requires fast transfer of data, for example in an environment where there is only a small time frame to receive and transmit the authentication data.

If the application of a lightweight authentication protocol is required in an energy-constrained environment, there are several options that use less energy than other lightweight protocols. From the NIST finalists, ASCON and TinyJAMBU are suitable in such an environment; of the non-NIST protocols, SIMON or SPECK are the most suitable out of the protocols, as demonstrated in Table 7.

5.3 Future Works

This research can be the starting point of several other papers. The following topics are related:

- In-depth analysis for lightweight authentication

  In this research, a researcher would generate their own data, preferably in a single environment, to improve the research presented in this paper. This would add more value to the field of lightweight cryptography, as the comparisons could be more thorough.

- Review of currently used lightweight authentication protocols

  In this research, a researcher would find out what lightweight protocols are used in real-world applications and whether the security of IoT is up to standards. This research would be of value as over time the data could be used to make risk analyses of the state of IoT.

- Implementation of lightweight authentication protocols using resource-constrained devices

  In this research, a researcher would implement lightweight protocols themselves for use on resource-constrained devices, such as Arduino or Raspberry Pi, to test their computation and communication performance.

6. ACKNOWLEDGEMENTS

We would like to thank Dr. Mohammed El-hajj for his help in supervising the research. His help and expertise has been instrumental to the succes of the research.

References

[1] M. D. Aagaard and N. Zidaric. ASIC Benchmarking of Round 2 Candidates in the NIST Lightweight Cryptography Standardization Process. Cryptology ePrint Archive, Report 2021/049, 2021. `https://ia.cr/2021/049`.

[2] R. Avanzi. The qarma block cipher family. almost mds matrices over rings with zero divisors, nearly symmetric even-mansour constructions with non-involutory central rounds, and search heuristics for low-latency s-boxes. *IACR Transactions on Symmetric Cryptology*, 2017(1):4–44, Mar. 2017.

[3] S. Banik, A. Chakraborti, T. Iwata, K. Minematsu, M. Nandi, T. Peyrin, Y. Sasaki, S. M. Sim, and Y. Todo. Gift-cofb. *IACR Cryptol. ePrint Arch.*, 2020:738, 2020.

[4] Z. Bao, A. Chakraborti, N. Datta, J. Guo, M. Nandi, T. Peyrin, and K. Yasuda. Photon-beetle authenticated encryption and hash family, 2019.

[5] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, and L. Wingers. The simon and speck lightweight block ciphers. In *Proceedings of the 52nd Annual Design Automation Conference*, pages 1–6, 2015.

[6] C. Beierle, A. Biryukov, L. C. dos Santos, J. Großschädl, L. Perrin, A. Udovenko, V. Velichkov, Q. Wang, and A. Biryukov. Schwaemm and esch: lightweight authenticated encryption and hashing using the sparkle permutation family. *NIST round*, 2, 2019.

[7] D. J. Bernstein and T. Lange. eBACS: ECRYPT Benchmarking of Cryptographic Systems. `https://bench.cr.yp.to`, accessed 12 January 2022.

[8] A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. B. Robshaw, Y. Seurin, and C. Vikkelsoe. Present: An ultra-lightweight block cipher. In P. Paillier and I. Verbauwhede, editors, *Cryptographic Hardware and Embedded Systems - CHES 2007*, pages 450–466, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[9] J. Borghoff, A. Canteaut, T. Güneysu, E. B. Kavun, M. Knezevic, L. R. Knudsen, G. Leander, V. Nikov, C. Paar, C. Rechberger, P. Rombouts, S. S. Thomsen, and T. Yalçın. Prince – a low-latency block cipher for pervasive computing applications. In X. Wang and K. Sako, editors, *Advances in Cryptology – ASIACRYPT 2012*, pages 208–225, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[10] J. Daemen, S. Hoffert, M. Peeters, G. V. Assche, and R. V. Keer. Xoodyak, a lightweight cryptographic scheme, 2020.

[11] J. Daemen and V. Rijmen. Announcing the advanced encryption standard (aes). *Federal Information Processing Standards Publication*, 197, 2001.

[12] C. De Cannière, O. Dunkelman, and M. Knežević. Katan and ktantan — a family of small and efficient hardware-oriented block ciphers. In C. Clavier and K. Gaj, editors, *Cryptographic Hardware and Embedded Systems - CHES 2009*, pages 272–288, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[13] C. Dobraunig, M. Eichlseder, S. Mangard, F. Mendel, B. Mennink, R. Primas, and T. Unterluggauer. Isap v2.0. submission to nist. *Submission to the NIST LWC Competition*, 2019.

[14] C. Dobraunig, M. Eichlseder, F. Mendel, and M. Schläffer. Ascon v1.2. *Submission to the CAESAR Competition*, 2016.

[15] M. J. Dworkin. Sha-3 standard: Permutation-based hash and extendable-output functions, 2015.

[16] T. Eisenbarth, Z. Gong, T. Güneysu, S. Heyse, S. Indesteege, S. Kerckhof, F. Koeune, T. Nad, T. Plos, F. Regazzoni, F.-X. Standaert, and L. van Oldeneel tot Oldenzeel. Compact implementation and performance evaluation of block ciphers in attiny devices. In A. Mitrokotsa and S. Vaudenay, editors, *Progress in Cryptology - AFRICACRYPT 2012*, pages 172–187, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[17] T. Eisenbarth, S. Kumar, C. Paar, A. Poschmann, and L. Uhsadel. A Survey of Lightweight-Cryptography Implementations. *IEEE Design Test of Computers*, 24(6):522–533, 2007.

[18] W. El Hadj Youssef, A. Abdelli, F. Dridi, and M. Machhout. Hardware Implementation of Secure Lightweight Cryptographic Designs for IoT Applications. *Security and Communication Networks*, 2020:8860598, Nov 2020.

[19] C. Guo, T. Iwata, M. Khairallah, K. Minematsu, and T. Peyrin. Romulus v1.3, 2021.

[20] X. Guo and P. Schaumont. The technology dependence of lightweight hash implementation cost. In *ECRYPT Workshop on Lightweight Cryptography (LC '11)*, 2011.

[21] P. B. Hari and S. N. Singh. Security issues in Wireless Sensor Networks: Current research and challenges. In *2016 International Conference on Advances in Computing, Communication, & Automation (ICACCA)(Spring)*, pages 1–6. IEEE, 2016.

[22] G. Hatzivasilis, K. Fysarakis, I. Papaefstathiou, and C. Manifavas. A review of lightweight block ciphers. *Journal of Cryptographic Engineering*, 8(2):141–184, Jun 2018.

[23] M. Hell, T. Johansson, A. Maximov, F. Willi Meier, S. J. Sönnerup, and H. Yoshida. Grain-128aeadv2 - a lightweight aead stream cipher, 2019.

[24] D. Hong, J.-K. Lee, D.-C. Kim, D. Kwon, K. H. Ryu, and D.-G. Lee. Lea: A 128-bit block cipher for fast encryption on common processors. In Y. Kim, H. Lee, and A. Perrig, editors, *Information Security Applications*, pages 3–27, Cham, 2014. Springer International Publishing.

[25] D. Hong, J. Sung, S. Hong, J. Lim, S. Lee, B.-S. Koo, C. Lee, D. Chang, J. Lee, K. Jeong, H. Kim, J. Kim, and S. Chee. Hight: A new block cipher suitable for low-resource device. In L. Goubin and M. Matsui, editors, *Cryptographic Hardware and Embedded Systems - CHES 2006*, pages 46–59, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[26] V. Karagiannis, P. Chatzimisios, F. Vazquez-Gallego, and J. Alonso-Zarate. A survey on application layer protocols for the internet of things. *Transaction on IoT and Cloud computing*, 3(1):11–17, 2015.

[27] S. Kramer. Announcing development of a federal information processing standard for advanced encryption standard. *Federal Register*, 62:93–94, 1996.

[28] Y. W. Law, J. Doumen, and P. Hartel. Survey and benchmark of block ciphers for wireless sensor networks. *ACM Trans. Sen. Netw.*, 2(1):65–93, feb 2006.

[29] R. Mahmoud, T. Yousuf, F. Aloul, and I. Zualkernan. Internet of things (IoT) security: Current status, challenges and prospective measures. In *2015 10th International Conference for Internet Technology and Secured Transactions (ICITST)*, pages 336–341. IEEE, 2015.

[30] K. McKay, L. Bassham, M. Sönmez Turan, and N. Mouha. Report on lightweight cryptography. Technical report, National Institute of Standards and Technology, 2017.

[31] B. Mennink. Elephant v2, 2021.

[32] B. J. Mohd and T. Hayajneh. Lightweight Block Ciphers for IoT: Energy Optimization and Survivability Techniques. *IEEE Access*, 6:35966–35978, 2018.

[33] B. J. Mohd, T. Hayajneh, and A. V. Vasilakos. A survey on lightweight block ciphers for low-resource devices: Comparative study and open issues. *Journal of Network and Computer Applications*, 58:73–93, 2015.

[34] R. M. Needham and D. J. Wheeler. Tea extensions. *Report, Cambridge University*, 1997.

[35] F. Olivier, G. Carlos, and N. Florent. New security architecture for IoT network. *Procedia Computer Science*, 52:1028–1033, 2015.

[36] M. Rana, Q. Mamun, and R. Islam. Current Lightweight Cryptography Protocols in Smart City IoT Networks: A Survey, 2020.

[37] S. Renner, E. Pozzobon, and J. Mottok. NIST LWC Software Performance Benchmarks on Microcontrollers, 2020.

[38] B. Rezvani, F. Coleman, S. Sachin, and W. Diehl. Hardware Implementations of NIST Lightweight Cryptographic Candidates: A First Look. Cryptology ePrint Archive, Report 2019/824, 2019. `https://ia.cr/2019/824`.

[39] A. Shah and M. Engineer. A survey of lightweight cryptographic algorithms for iot-based applications. In S. Tiwari, M. C. Trivedi, K. K. Mishra, A. K. Misra, and K. K. Kumar, editors, *Smart Innovations in Communication and Computational Sciences*, pages 283–293, Singapore, 2019. Springer Singapore.

[40] K. Shibutani, T. Isobe, H. Hiwatari, A. Mitsuda, T. Akishita, and T. Shirai. Piccolo: An ultra-lightweight blockcipher. In B. Preneel and T. Takagi, editors, *Cryptographic Hardware and Embedded Systems – CHES 2011*, pages 342–357, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[41] T. Suzaki, K. Minematsu, S. Morioka, and E. Kobayashi. TWINE: A lightweight block cipher for multiple platforms. In L. R. Knudsen and H. Wu, editors, *Selected Areas in Cryptography*, pages 339–354, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[42] M. S. Turan, K. McKay, D. Chang, C. Calik, L. Bassham, J. Kang, J. Kelsey, et al. Status Report on the Second Round of the NIST Lightweight Cryptography Standardization Process. Technical report, Tech. rep. https://doi. org/10.6028/NIST. IR. 8369. Gaithersburg, MD, USA . . . , 2021.

[43] M. S. Turan, K. A. McKay, Ç. Çalik, D. Chang, L. Bassham, et al. Status report on the first round of the nist lightweight cryptography standardization process. *National Institute of Standards and Technology, Gaithersburg, MD, NIST Interagency/Internal Rep.(NISTIR)*, 2019.

[44] J. M. Turner. The keyed-hash message authentication code (hmac). *Federal Information Processing Standards Publication*, 198(1), 2008.

[45] M. Usman. Lightweight Encryption for the Low Powered IoT Devices, 2020.

[46] Q. Wen, X. Dong, and R. Zhang. Application of dynamic variable cipher security certificate in internet of things. In *2012 IEEE 2nd International Conference on Cloud Computing and Intelligence Systems*, volume 3, pages 1062–1066. IEEE, 2012.

[47] D. J. Wheeler and R. M. Needham. Tea, a tiny encryption algorithm. In B. Preneel, editor, *Fast Software Encryption*, pages 363–366, Berlin, Heidelberg, 1995. Springer Berlin Heidelberg.

[48] H. Wu and T. Huang. Tinyjambu: A family of lightweight authenticated encryption algorithms, 2019.

[49] W. Zhang, Z. Bao, D. Lin, V. Rijmen, B. Yang, and I. Verbauwhede. Rectangle: a bit-slice lightweight block cipher suitable for multiple platforms. *Science China Information Sciences*, 58(12):1–15, Dec 2015.

APPENDIX

A. DATA

In this appendix, the data for the analysis metrics has been collected and put in Table 9.

The table contains references to the reviews and researches that the data originated from, as well as some comments on the research method or variables that could impact the data. Moreover, some extra data is shown, such as the encryption and decryption power and code size.

Table 9: Metrics for several lightweight authentication protocols

		Key size (bits)	Block size (bits)	Rounds	RAM Occupation (Bytes)	ROM Occupation (Bytes)	Code size (Bytes)	Encr or Decr Throughput (Mbps)	Encr and Decr Latency (Cycles)	Encr and Decr Power	Encr and Decr Energy Consumption	Gate Area (GE)	Notes
AES	[11]	128 / 192 / 256	128 bits	10 / 12 / 14									
	[36]	128	128	10				56,64	226			2400	Technology Value 0.13 µm
	[28]	128	128	10	302	8684	2600						
	[16]	128	128		33		1659				19,2 µJ		
	[22]	128	128		127	2742		0,02265	22603 (/block)		90,4 µJ		8bit microchip
	[17]	128	128				2606	0,496				3400	@ 4 MHz
ASCON	[14]	128	64 / 128	12, 6, 12									
	[1]							25,60 (bpc)			0,29 nJ	1,49	*Ascon GMU-v1
	[37]	128		12, 6, 12	1340	7068							*F7, ASCON128-12
	[38]	128	64	12, 6, 12				256	52	46,2 mW		1898	*fclk = 40MHz
	[7]	128	64	12, 6, 12				20,86					*median, 1536+1536 encrypt, 40 MHz, ascon128v12
Elephant	[31]	128	160 / 176 / 200	80									
	[1]							7,27 (bpc)			0,88 nJ	1,72	*Elephant-v5
	[37]	128		80	1492	12888							*F7, DUMBO
	[7]	128	200	80				1,679					*median, 1536+1536 encrypt, 40MHz, elephant200v1
GIFT-COFB	[3]	128	128							156,3 µW	1,31; 2,00; 2,69 nJ	3927	
	[37]	128			1564	10796							*F7
	[38]	128	128	40				96,6	155	32,1 mW		1932	*fclk = 40MHz
	[7]	128	128					5,95					*median, 1536+1536 encrypt, 40MHz, giftcofb128v1
Grain-128AEAD	[23]	128											
	[37]	128			9989	13692							*F7
	[7]	128						3,542					*median, 1536+1536 encrypt, 40MHz, grain128aead
HIGHT	[25]	128	64	32									
	[36]	128	64	32				188	34			3048	Technology Value 0.13 µm
	[16]	128	64		32		402				79,8 µJ		
	[22]	128	64		54	1084		0,02245	11399 (/block)		45,5 µJ		8bit microchip
	[17]	128	64				5672	7,528					@ 4 MHz

Table 9: Continued: Metrics for several lightweight authentication protocols

		Key size (bits)	Block size (bits)	Rounds	RAM Occupation (Bytes)	ROM Occupation (Bytes)	Code size (Bytes)	Encr or Decr Throughput (Mbps)	Encr and Decr Latency (Cycles)	Encr and Decr Power	Encr and Decr Energy Consumption	Gate Area (GE)	Notes
ISAP	[13]	128	400	16, 1, 8, 8				3,42 (bpc)			1,39 nJ	1,19	*ISAP-v1
	[1]												*F7, ISAP-K-128a
	[37]	128		16, 1, 8, 8	1492	11396							*median, 1536+1536 encrypt, 40MHz, isapk128av20
	[7]	128		16, 1, 8, 8				1,462					
KATAN	[12]	80	32 / 48 / 64	254									8bit microchip
	[32]	80	32 / 48 / 64	254									
	[16]	80	64		18		338				289,2 µJ		
	[22]	80	64		18	338		0,0035	72063 (/block)		289,2 µJ		
	[45]	80	32, 48, 64	254									
KTANTAN	[12]	80 bits	32 / 48 / 64	254									Technology Value 0.13 µm
	[36]	80	64	12				25,1	255			688	
	[22]												
LEA	[24]	128 / 192 / 256	128 bits	24 / 28 / 32									
PHOTON-Beetle	[4]	128		12	64	1122							*PHOTON-Beetle-v1
	[1]							3,88 (bpc)			1,99 nJ	1,11	*F7, PHOTON-Beetle-AEAD-128
	[37]	128			1440	13412							*median, 1536+1536 encrypt, 40MHz, photon-beetleaead128rate128v1
	[7]	128						3,112					
Piccolo	[40]	80 / 128	64	25 / 31								1260	Technology Value 0.13 µm
	[36]	80	64	25				237	237,04				8bit microchip
	[22]	80	64		64	1178		0,00996	25681 (/block)		102,7 µJ		
PRESENT	[8]	80 / 128	64	31		660		23,7	10792		43,1 µJ		Technology Value 0.13 µm
	[36]	80	64	31				206	31			2195	8bit microchip
	[16]	80	64		18		1000				45,3 µJ		
	[22]	128	64		-	660		0,0237	10972 (/block)		43,1 µJ		
	[45]	80	64	32								1570	@ 4 MHz
	[17]	80	64				936	8,00					
PRINCE	[9]	128 bits	64 bits	11									

Table 9: Continued: Metrics for several lightweight authentication protocols

		Key size (bits)	Block size (bits)	Rounds	RAM Occupation (Bytes)	ROM Occupation (Bytes)	Code size (Bytes)	Encr or Decr Throughput (Mbps)	Encr and Decr Latency (Cycles)	Encr and Decr Power	Encr and Decr Energy Consumption	Gate Area (GE)	Notes
PRINCE	[36]	128	64	12				533	12			3491	Technology Value 0.13 µm
	[22]	128	64		63	4300		0,01487	17207 (/block)		68,8 µJ		8bit microchip
QARMA	[2]	128, 256 bits	64, 128 bits	varies									
	[36]	64	64	27				2667	27			7844	Technology Value 0.13 µm, MODIFIED QARMA
RECTANGLE	[49]	80, 128 bits	64 bits	25									
	[36]	128	64	25				246	26			1787	Technology Value 0.13 µm
	[45]	80						0,246				1600	@ 100 KHz
Romulus	[19]	128	128	40									* previous version of specification used 56 rounds, which explains the result at 15
	[1]							7,11 (bpc)			0,92 nJ	0,84	*Romulus-v3
	[37]	128		56	2656	12448							*F7, Romulus-N1
	[7]	128		56				1,339					*median, 1536+1536 encrypt, 40MHz, romulusn1v1
SIMON	[5]	64 / 72 / 96 / 128 / 144 / 192 / 256	32 / 48 / 64 / 96 / 128	32 / 36 / 42 / 44 / 52 / 54 / 68 / 69 / 72									
	[36]	96	48	32				15,8	304			763	Technology Value 0.13 µm
	[22]	128	64		0	246		0,284	901 (/block)		3,6 µJ		8bit microchip
	[45]			22+									
	[18]	128	64	44				206,38	44	3 mW			@ 141,89 MHz
SPARKLE	[6]	128 / 192 / 256	16 / 24 / 32										
	[38]	128	256	4				196,92	166	39,1 mW		4313	*fclk = 40MHz
	[7]	128	256					7,726					*median, 1536+1536 encrypt, 40MHz, schwaemm256128v1
SPECK	[5]	64 / 72 / 96 / 128 / 144 / 192 / 256	32 / 48 / 64 / 96 / 128	22 / 23 / 26 / 27 / 28 / 29 / 32 / 33 / 34									
	[22]	128	128		0	396		0,384	1333 (/block)		5,3 µJ		8bit microchip
	[45]			22+									
TinyJAMBU	[48]	128 / 192 / 256	32										
	[1]							4,00 (bpc)			0,29 nJ	0,39	*TinyJAMBU TJT-v3
	[37]	128			1216	7204							*F7, TinyJAMBU-128

Table 9: Continued: Metrics for several lightweight authentication protocols

		Key size (bits)	Block size (bits)	Rounds	RAM Occupation (Bytes)	ROM Occupation (Bytes)	Code size (Bytes)	Encr or Decr Throughput (Mbps)	Encr and Decr Latency (Cycles)	Encr and Decr Power	Encr and Decr Energy Consumption	Gate Area (GE)	Notes
TinyJAMBU	[7]	128						5,619					*median, 1536+1536 encrypt, 40MHz, tinyjambu128
TWINE	[41]	80, 128 bits	64 bits	36									
	[22]	80	64		59	1408		0,01183	21637 (/block)		86,5 µJ		8bit microchip
Xoodyak	[10]	128											
	[1]							27,57 (bpc)			0,40 nJ	1,56	*Xoodyak GMU2-v2
	[37]	128			1336	7204							*F7, Xoodyakv1
	[7]	128						25,7					*median, 1536+1536 encrypt, 40MHz, xoodyakv1
XTEA	[47, 34]	128 bits	64 bits	32	24		648				30,3 µJ		
	[16]	128	64	64									TEA instead of XTEA
	[22]	128	64		-	504		0,0146	17514 (/block)		70,0 µJ		8bit microchip