A Comparative Study of the TEA, XTEA, PRESENT and Simon lightweight cryptographic schemes

Paul E.A. Adriaanse
Daily Supervisor: Miray Ayşen
Responsible Professor: Zekeriya Erkin

Cyber Security Group
Department of Intelligent Systems
Delft University of Technology

14-7-2021

Abstract

With the current fast paced growth in the number of devices connected to the internet, many of these having limited computational capability, security concerns are of increasing importance. To meet the necessity of providing secure encryption to constrained devices many lightweight cryptographic schemes have been developed. This paper provides a comparative study of four of these schemes, namely TEA, XTEA, PRESENT and Simon, explaining how they work and discussing their vulnerabilities and performances. This paper specifically considers the performance of Application Specific Integrated Circuit (ASIC) implementations with regards to their size, throughput, power usage and energy usage per bit encrypted. TEA and PRESENT were found to have potentially problematic vulnerabilities, while no attacks on XTEA and Simon are known that do not reduce the number of rounds used during encryption. Simon and PRESENT were found to perform well, while XTEA implementations were found to be too large for constrained devices. Out of the four, Simon seems the most promising for use in constrained devices. As such, further research into its vulnerabilities is advisable.

1 Introduction

The amount of Internet of Things (IoT) devices worldwide is growing at a fast pace, with (Balaji et al., 2019) stating this would grow from 25 billion devices in 2019 to 60 billion devices in 2025. Apart from the large number of devices, a great deal of different IoT applications exist, or may be viable in the future. Examples of this include things such as smart watches, e-bikes, smart homes with automated locks & lights and even skin patches (Sethi & Sarangi, 2017).

The security of these devices is however an area of concern, especially when the device could pose a physical threat to its users. One example for which major security concerns exist are modern cars. Video demonstrations of people hacking cars & taking over control are present in abundance on online media, such as demonstrations by security researchers described by Greenberg (2020). Oka et al. (2014) showed some bluetooth devices used in cars allow for very simple pairing with attackers, also stating this can be further abused to gain further access. This can potentially pose a danger to passengers, with the Federal Bureau of Investigation (FBI) even warning consumers and manufacturers of the potential dangers, including manipulation of the breaks, door locks and steering (FBI, 2016). Another example is an attack described by Ronen & Shamir (2016), pertaining to the use of smart lights to potentially cause seizures in people with epilepsy.

Apart from physical threats, the privacy of users could also be threatened when IoT devices deal with sensitive information. Skin patches generating data regarding the physical condition of users are a clear example of this. Many other applications that deal with human interaction however also pose privacy risks. Smart lights may for example measure the presence of people in a room, which could possibly be used to track individuals. Other privacy risks may also include the ability to discern behavior habits of users and identify individuals. Comparably, Liebers et al. (2021) have already shown this to be possible using the motion data generated by Virtual Reality (VR) users.

Abomhara & Køien (2015) state several reasons that may make IoT devices particularly interesting to attackers, stating many IoT devices cannot support complex security schemes due to hardware limitations such as memory limitations and low computational capabilities. Additionally, many devices use wireless networks or operate without human supervision.

As many encryption algorithms exist, the focus of this paper is on how several lightweight cryptographic block ciphers compare to one another, specifically the TEA, XTEA, PRESENT and Simon block ciphers. The vulnerability of each scheme is considered, as well as the performance of implementations of each scheme. This research is based on previous work, as such a literary survey was performed. Metrics used to compare performances are implementation size, throughput, power usage and energy usage per bit encrypted. This paper specifically looks at Application-Specific Integrated Circuit (ASIC) implementations, which are integrated circuits specifically designed for the application in question. This in contrast to Field-Programmable Gate Array (FPGA) implementations, which are implementations that make use of hardware components that can be re-programmed in order to implement the desired functionality. This choice was made as ASIC implementations generally perform better than FPGA's with lower power usage, while also having a lower unit cost, as noted by Intel (n.d.). FPGA's may however be preferable when changes to the initial design may need to be made in the future. Microcontrollers using software implementations are also a viable alternative. These are however not considered in this paper as both Simon and PRESENT are designed with hardware implementation in mind. The choice between ASIC and the use of microcontrollers is beyond the scope of this paper.

In an attempt to fairly compare different ASIC implementations on different architectures in respect to implementation size, the gate equivalents (GE) unit of measurement shall be used, which is generally defined by the size of one of the smallest logic gates in the respective implementation architecture, in particular the 2-input NAND gates.

This paper begins by shortly describing Feistel networks and Substitution-Permutation networks in section 3. The paper then describes the TEA, XTEA, PRESENT and Simon algorithms in sections 4 through 7 respectively, with each section describing their functionality, followed by subsections regarding their vulnerabilities and their performance given existing ASIC implementations. Section 8 comparing the vulnerabilities and performances of the considered algorithms will follow, after which section 9 discusses the findings. Section 10 shortly mentions any ethical implications of this research after which a conclusion follows in section 11, additionally mentioning possible future research topics.

Some additional extensions of TEA are briefly discussed in appendix A, with appendix B containing specific information regarding the functioning of PRESENT.

2 Related Work

Several studies into the security of different encryption algorithms exist, often directed at specific algorithms, along with research attempting to implement and analyse the performance of these algorithms. More general research regarding lightweight cryptographic schemes is mentioned below, with further research into the security and performance of the specific schemes under consideration being mentioned in sections 4 through 7.

One paper by Hatzivasilis et al. (2018) provides an extensive comparison of many lightweight cryptographic ciphers, including hardware comparisons with the criteria mentioned earlier. Interesting aspects of the paper include its cipher grouping based on different time periods and the amounts of ciphers and implementations under comparison. The paper however does not go into the functional details of the ciphers nor does it suggest changes or future research into the ciphers in question. Additionally, the Simon implementations it mentions only come from the original proposal (Beaulieu et al., 2013).

Another paper by Mohd et al. (2015) also compares a large number of lightweight block ciphers. It especially considers performance metrics, and tries to rank the considered ciphers according to these metrics based on other studies into the subject. The study however does not consider cipher vulnerabilities nor does it propose any improvements.

A comparative study by Kerckhof et al. (2012) studies different ciphers along with the effect of implementation decisions, including frequency. As noted by many other authors, energy usage is very dependant on the architectures and thus does not function as a fair comparison between different ciphers (Kerckhof et al. (2012) , Hatzivasilis et al. (2018), Beaulieu et al. (2013)). Kerckhof et al. however find the energy per bit metric to be much less dependant on the architecture used, making this a better metric to compare different cryptographic schemes with, regardless of the architecture used for implementation. As such this paper includes this metric.

3 Cipher Structure

Figure 1 depicts the structure of Feistel networks on the left and the structure of Substitution-Permutation (SP) networks on the right, both with two rounds. Other structures exist, however these are beyond the scope of this paper, as TEA, XTEA and Simon are all Feistel ciphers, while PRESENT is a Substitution-Permutation network.

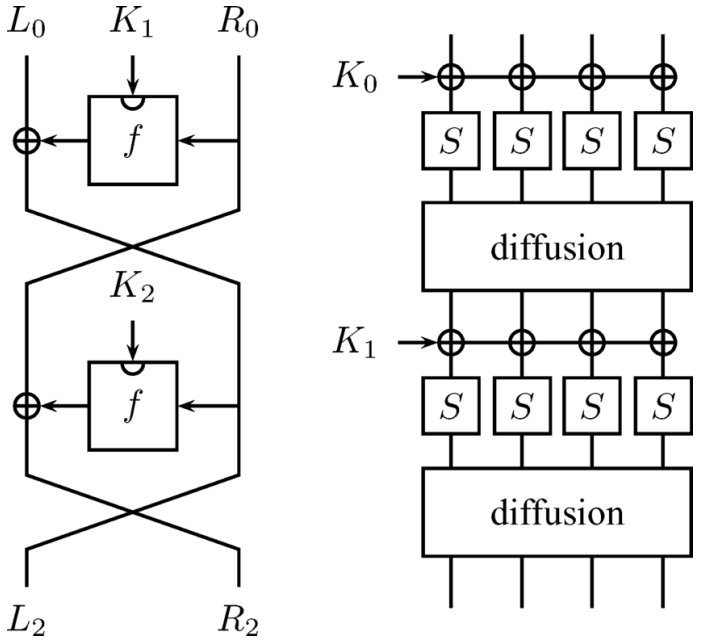

Figure 1: Structure of Feistel networks (left) and Substitution-Permutation networks (right) (De Canniere et al., 2006)

3.1 Feistel Networks

Feistel networks generally encrypt plaintext by splitting it into two halves. During several rounds a round function, making use of the round key, is applied to either of these halves, the result of which is then xor-ed with the other half. Decryption is achieved in a similar manner but in a reversed order. Notably the round function used does not need to be inversible.

3.2 Substitution Permutation Networks

As can be seen in figure 1, Substitution-Permutation (SP) networks function by having several rounds in which the plaintext is substituted using a substitution-block (S-block) and permuted/diffused using a permutation-block (P-block). Each roundf it is also combined with the round key, generally using a xor operation.

4 Tiny Encryption Algorithm

The Tiny Encryption Algorithm (TEA) is a Feistel cipher designed by Wheeler & Needham (1994).

TEA uses a key size of 128 bits with a block size of 64 bits and uses 32 cycles, or equivalently 64 Feistel rounds. One cycle of TEA, also 2 Feistel rounds, is described by figure 2. Each round the left part of the data is summed with a number gained by shifting the righter part and adding onto it in three different ways, then xor-ring these 3 resulting numbers. K[j] here refers to the j-th quarter of the key used, with K[0] referring to the first 32 bits. Delta meanwhile is a golden number equaling the constant $(\sqrt{5}-1)*2^{31} = 9E3779B9_{16}$, with $Delta_i$ referring to the multiple $Delta_i = (i+1)*Delta$, with i referring to the cycle number.

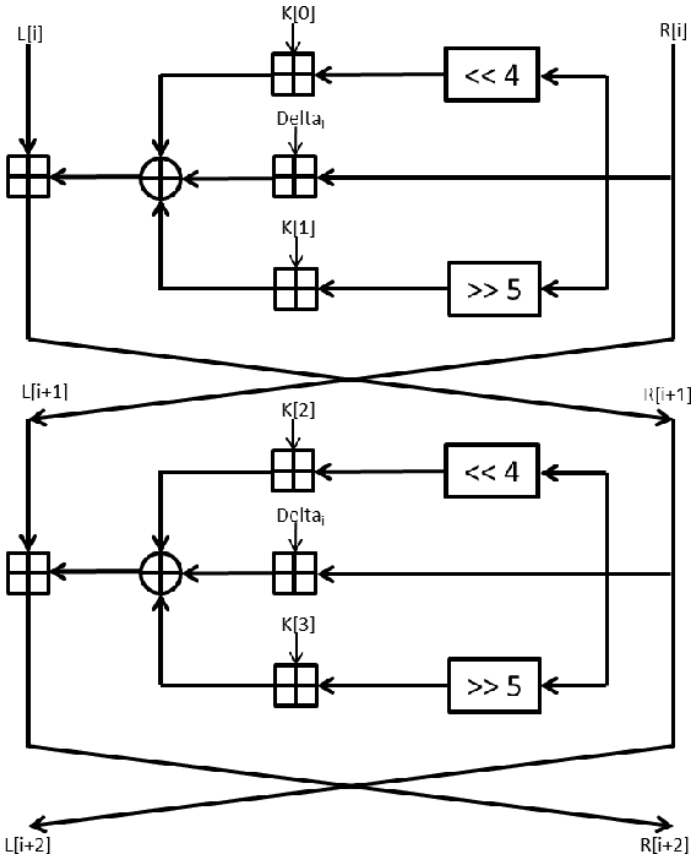

Figure 2: The i-th cycle of TEA (Appel et al., 2016)

Note that in contrast to most Feistel cyphers, the leftmost data half is summed with the result from the round function applied to the right half, instead of being xor-red with it. Decryption is again very similar to the reversal of encryption rounds as seen in Feistel decryption, but with subtraction operations replacing the xor-ring operations seen in figure 1.

4.1 Vulnerabilities

TEA can be shown to have an effective key space of 126 instead of 128, due to the round function yielding equivalent results for different keys. As described by Andem (2003), when the highest bits of the 2 subkeys used in the rounding function are inversed the subsequent xor in the rounding function results in the same number. As K[0] and K[1] are used together, as are K[2] and K[3], this thus means each key has 3 other equivalent keys, keys that will result in the same encryption/decryption. This thus happens when the first pair's most significant bits are inverted, when the second pair's most significant bits are inverted, and when this is done for both, as such the key space is effectively divided by 4.

Due to this key equivalence, TEA is not appropriate for use as a hash function, as described by Shepherd (2007). Shepherd also notes the usage of TEA as a hash function in Microsoft's XBOX console made it possible to start boot-loaders on RAM, instead of booting from ROM, by flipping 2 bits on the device.

Kelsey et al. (1997) show TEA is also susceptive to related-key attacks, providing several different examples. Shepherd (2007) however seems to imply this issue, amongst others, is not very problematic, stating they are impractical. Likewise, Andem (2003) states TEA is high resistance to attacks.

Security analysis of diffusion and confusion degrees of the TEA, KLEIN, KATAN and HIGHT algorithms by Alizadeh et al. (2012) showed all four to be very comparable.

4.2 Performance

One Verilog HDL implementation of TEA by Israsena (2005) is stated to require an area of $0.207mm^2$ using $0.35\mu m$ CMOS technology, reaching a maximum throughput rate of 53 million cycles per second and with power consumption stated to be $7.37\mu W$ when running at 25.6K cycles per second.

Analysis of the performance of a TEA software implementation on an AVR Atmel microcontroller, when compared to the KLEIN, KATAN and HIGHT lightweight encryption algorithms, by Alizadeh et al. (2012) indicated TEA to require a relatively low energy usage. Memory usage of TEA was however higher than that of HIGHT and KATAN.

Bogdanov et al. (2007) estimated TEA to require at least 2100 GE by assigning estimated GE to several operations (such as a 32-bit xor operation), although this is most probably too optimistic, as also noted in section 5.2.

5 Extended Tiny Encryption Algorithm

In an attempt to fix the issues mentioned in section 4.1 the eXtended Tiny Encryption Algorithm (XTEA) was devised by the same designers (Needham & Wheeler, 1997). It uses a different round function that also changes which parts of the key are used during each round. The i-th cycle of XTEA is depicted in figure 3, which is similar to the i-th cycle of TEA shown in figure 2.

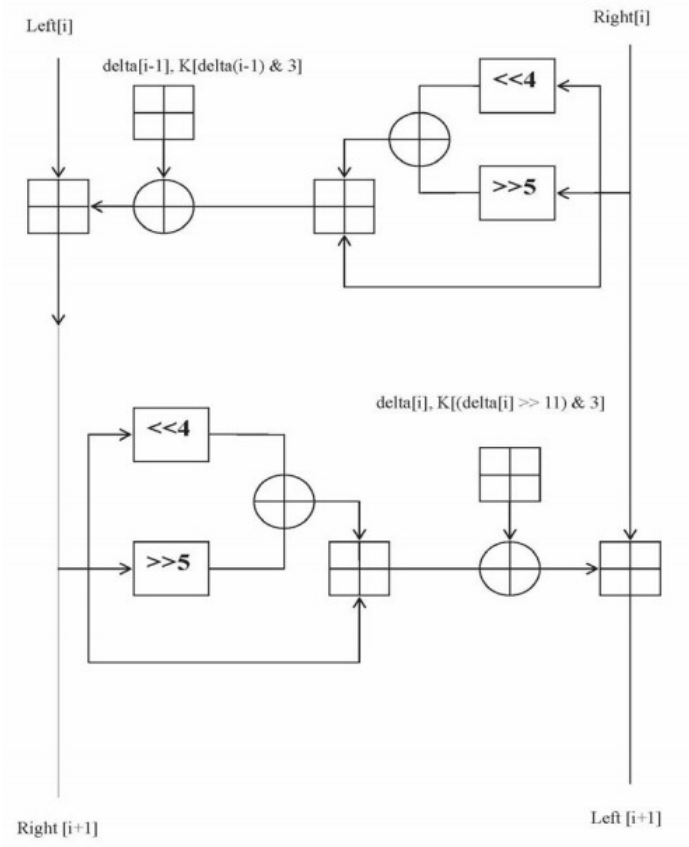

Figure 3: The i-th cycle of XTEA (Andem, 2003)

5.1 Vulnerabilities

Although intended as an improvement, attacks on XTEA with a reduced amount of rounds have still been found, including related-key attacks (Ko et al., 2004), but also others such as meet-in-the-middle attacks (Isobe & Shibutani, 2012). Many of these however rely on a reduction of the amount of rounds used by the algorithm, as such a reduction in the number of rounds used may not be advisable, to what extend this is viable is however unclear.

A paper by Khan & Moessner (2011) takes the stance this is viable up until 36 rounds (or equivalently 18 cycles). This was done in an attempt to save power consumption and defended by referring to a paper by Lu (2009), describing a related-key attack on a 36 round version of XTEA, with claims no better results for less reduced versions had been found at the time.

Although no full attack on XTEA is currently known, Khan & Moessner (2011) state XTEA is much weaker than AES.

5.2 Performance

Kaps (2008) reported on several implementations of XTEA, focussed on either speed or power efficiency. Designs for both being testing when implemented as ASIC's and using FPGA's. The authors also included comparisons with other reported implementations of different encryption schemes, such as AES, DESXL and PRESENT. In the ASIC case, the highest throughput of 36,571 Gbps was reached with 529.987 gate equivalents (GE), with a maximum delay of 1,75 ns. At 100 kHz the power-efficient designs reached throughputs of 26,7 and 57,1 Kbps at powers of 18,8 and 19,5 μW respectively, with 3.500 and 3.490 GE and maximum delays of 11,28 and 11,66 ns. The power-efficient designs used 703 and 341 pJ per bit. The optimized implementations are included in table 1.

Kaps concludes by stating XTEA is viable for applications that require high speeds, while noting it is slower than AES (the Advanced Encryption Standard). When power is a concern, Kaps concludes XTEA may however be a better choice than AES. An interesting note is also raised regarding the possibility of having different devices that use either hardware or software implementations in the same environment, as software implementations of XTEA are quite small.

The implementation by Kitsos et al. (2012) is comparable in size, while it reaches a higher output at the cost of a higher power usage, when compared to the power-efficient designs by Kaps (2008). Note the majority of this power usage is due to leakage, which is more generally the case for low power applications at this frequency, as noted by Kaps. Kitsos et al. note leakage power is proportional to both implementation size and the technology used. The difference in power usage in table 1 may thus be partially caused by the difference in technologies used.

Bogdanov et al. (2007) estimated XTEA to require at least 2000 GE, reached in the same way as their estimate for TEA mentioned in the previous section (2100 GE). This estimate is most likely highly optimistic, as the implementations shown in table 1 are significantly larger.

Table 1: Performances of several XTEA implementations at 100 kHz.

technology (nm)	source	area (GE)	throughput (kbps)	power(μW)	energy/bit (pJ/bit)
130	Kaps (2008)	3500	26,7	18,8	591,8
130	Kaps (2008)	3490	57,1	19,5	341,5
90	Kitsos et al. (2012)	3490	200	61	305

6 PRESENT

PRESENT is a Substitution-Permutation (SP) network designed by Bogdanov et al. (2007).

PRESENT uses 31 rounds with a key size of either 80 or 128 bits and a block size of 64 bits. Each round the data is combined with the current round key using a bit-wise xor operation, after which 16 parallel and identical S-boxes with an in- and output size of 4 bits is applied, followed by one P-box with an in- and output size of 64 bits. The specifics of these S- and P-boxes are provided in the appendix. After the last round the data is once more combined with an additional 32nd 'round' key providing the final ciphertext.

The round key consists of the leftmost 64 bits in the key register, which is initialised with the full key in use. After use, this register is shifted left by 61 bits (circularly), after which an S-box is applied to the leftmost 4 bits. In the event 128 bit keys are used, an S-box is also applied to the 4 following these initial bits (thus 2 S-boxes are applied to the leftmost 8 bits). Finally, the bits at position 19 through 16, or 66 through 62 in the case of 128 bit keys, are xor-ed with a 5 bit round counter. Note the least significant bits are on the right, at bit position 0.

6.1 Vulnerabilities

As with TEA, many reduced round attacks have been proposed, many of these however function on around 26 rounds, needing in between 2^{57} and 2^{64} known or chosen plaintext (Cho, 2010; Faghihi Sereshgi et al., 2016). As the block size is only 64 bits this is thus not very problematic. Blique attacks on a full (31 round) PRESENT have however been found for both 80 and 128 bit keysizes. The attacks with the fewest chosen plaintexts seeming to be one by Faghihi Sereshgi et al. (2016) on PRESENT-80, only requiring 2^{22} chosen plaintext, and one by Lee (2014) on PRESENT-128, reportedly only requiring 2^{19} known plaintexts. As such this cipher may not be suitable for applications that send much data. This is however in accordance with design goals for the scheme, as applications that don't send much data were mostly targeted.

Most interesting is the fact that both Abed et al. (2012) and Lee (2014) proposed attacks for both versions, with PRESENT-128 interestingly requiring fewer chosen plaintexts in both cases. This seems to indicate PRESENT-128 may possibly be less secure than its 80 bit counterpart, something 'playful research' by Hernandez-Castro et al. (2012) also seemed to suggest.

6.2 Performance

In their paper introducing PRESENT, Bogdanov et al. (2007) note PRESENT is particularly directed at applications in which a hardware implementation is used, and for which it is not necessary to encrypt large amounts of data, instead focusing on space, power usage & time requirements, in that order of importance. The authors also initially estimated PRESENT to require roughly 3600 GE by assigning GE to different operations.

The authors also implemented the 80 bit keysize version of PRESENT using VHDL, optimizing for area, with their implementation required only 1570 GE, of which most was comprised of the s-layer, the data & key state storage and the xor combining round key and data, while consuming 5 μW of power. They also note the addition of 53 GE could further reduce power consumption to 3.3 μW and estimate PRESENT-128 would require 1886 GE. Additional serialisation further reducing the size could also be performed as all S-boxes used by PRESENT are identical.

One ASIC implementation by Pandey et al. (2018) requires 1785 GE, with a throughput of roughly 133 Mbps at 100 MHz, with a power usage of 273 μW, consuming 16.36 nJ per bit. This implementation is however much more power-intensive than the implementation mentioned above and thus possibly not suitable for comparison for low power applications.

Implementations by Yap et al. (2011), in their paper proposing the EPCBC block cipher, yielded better results for both PRESENT-80 and PRESENT-180, as shown in table 2.

Table 2: Performance of different PRESENT implementations at 100 kHz.

cipher	technology (nm)	source	area (GE)	throughput (kbps)	power(μW)	energy/bit (pJ/bit)
PRESENT-80	180	Bogdanov et al. (2007)	1570	200	5	10
PRESENT-80	180	Yap et al. (2011)	1030	12,4	-	-
PRESENT-128	180	Yap et al. (2011)	1339	12,12	-	-

7 Simon

Simon is a family of lightweight block ciphers developed by Beaulieu et al. (2013) at the National Security Agency (NSA) that was designed specifically with hardware performance in mind. It was proposed together with the Speck family of lightweight block ciphers, which focused on software performance instead. An important design aspect of both ciphers was their flexibility, which refers to their ability to be efficiently implemented on different platforms, and to be implemented in different ways such that different requirements can be met. This includes the possibility of having small implementations while also allowing for larger implementations with more throughput. With this flexibility in mind the ciphers come with block sizes between 32 and 128 bits and with key sizes between 64 and 256 bits. All versions are given in table 3.

Table 3: Simon ciphers & parameters (Beaulieu et al., 2013)

block size	key size	rounds	sequence
32	64	32	z_0
48	72	36	z_0
48	96	36	z_1
64	96	42	z_2
64	128	44	z_3
96	96	52	z_2
96	144	54	z_3
128	128	68	z_2
128	192	69	z_3
128	256	72	z_4

Simon, like TEA and XTEA, as described in section 4, functions as a Feistel cipher. Its round function is relatively simple and is depicted in figure 4. Here S^j is a circular shift left by j bits, & is a bitwise AND, and $\oplus$ a bitwise xor. k_i is the round key. To find the round key we first define the constant z_j sequences below. The sequence used for each version of Simon is given in table 3. For how these sequences were reached, you are referred to the original paper (Beaulieu et al., 2013).

$$z_0 = 11111010001001010110000111001 10...$$

$$z_1 = 10001110111110010011000010110 10...$$

$$z_2 = 10101111011100000011010010011000101000010001111110010110110011...$$

$$z_3 = 11011011101011000110010111100000010010001010011100110100001111...$$

$$z_4 = 11010001111001101011011000100000010111000011001010010011101111...$$

The first m round keys are simply the m n-bit words of the key in use, where the word size n is half the block size and where the first round key k_0 is the rightmost part of this key. Each subsequent key is given by the formula below, where $c = 2^n - 4$ and where $(z_j)_i$ is the i-th value in the sequence.

$$k_i = \begin{cases} c \oplus (z_j)_i \oplus k_{i-m} \oplus (S^{-3}k_{i-1}) \oplus (S^{-4}k_{i-1}) & \text{if } m = 2, 3 \\ c \oplus (z_j)_i \oplus k_{i-4} \oplus (I \oplus S^{-1})((S^{-3}k_{i-1}) \oplus k_{i-3}) & \text{else } m = 4 \end{cases}$$

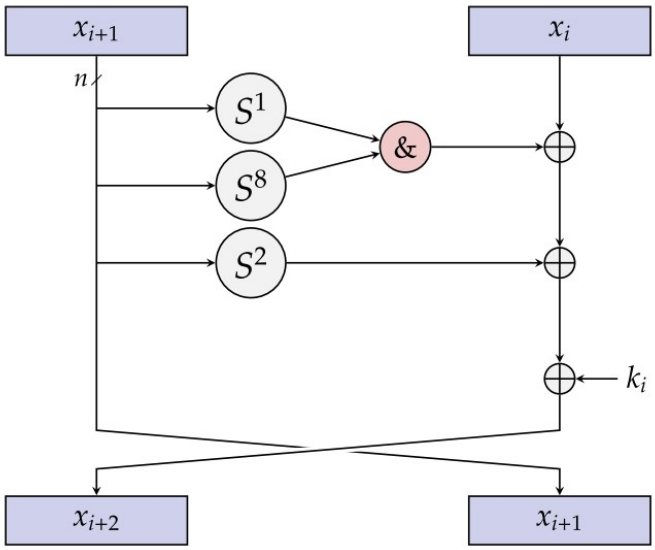

Figure 4: Round of Simon cipher. Beaulieu et al. (2013)

7.1 Vulnerabilities

As is the case with XTEA, no attacks on non-reduced versions of Simon seem to have been published. Cryptanalysis by AlKhzaimi & Lauridsen (2013) shows Simon is susceptive to attacks on reduced versions with more than half the original rounds, with the paper attacking 24 out of the 42 rounds of Simon64/96 and the 44 rounds of Simon64/128 using differential attacks.

Chen & Wang (2016) further attacked Simon using linear hull attacks, reaching 30 and 31 rounds for Simon64/96 and Simon64/128 respectively, further reducing the safety margins for these to 12 and 13 rounds respectively.

7.2 Performance

The original authors report on a large number of implementations for several Simon ciphers, with fully complete area minimized encryption only versions ranging from 523 to 1840 GE, depending on the block and key size. Other less optimized, but still small, implementations meanwhile reaching higher throughput. As Simon64/96 and Simon64/128 are best comparable with PRESENT, as these Simon ciphers have the same block size and the key size is (roughly) the same, some implementation results of these are included in table 4. Note 'Simon64/96' refers to the Simon cipher with a block size of 64 bits and a key size of 96 bits.

In a paper proposing Simeck, a cipher family based on Simon and Speck Yang et al. (2015) report on a multitude of Simon ASIC implementations. These vary in how far the designs have been serialised and by technology (namely semiconductor scale). Again some of the implementations with a block size of 64 are shown in the table below. The authors give the area for both before and after a 'place and route' phase of implementation, noting the implementations by Beaulieu et al. are only for before this phase. As such only these have been provided in table 3. In general all implementations by Yang et al. are smaller, this is also the counts for the other ciphers provided, with the smallest even reaching a GE of 517.

Interesting studies by Gulcan et al. (2015) and Rashidi (2019) both implement flexible designs of Simon that supports all Simon ciphers. Gulcan et al. compare their implementation to other block ciphers, concluding it to be smaller than implementations of other block ciphers such as AES, XTEA and PRESENT. Sadly Gulcan et al. use FPGA's, and as such their results are not comparable to other implementations mentioned in this paper. Rashidi in contrast uses ASIC to implement his design. The implementation is however quite large (5647,20 GE) and designed to achieve high throughput.

Table 4: Performances of several Simon implementations at 100kHz.

cipher	technology (nm)	source	area (GE)	throughput (kbps)	power(μW)	energy/bit (pJ/bit)
Simon64/96	130	Beaulieu et al. (2013)	809	4,4	-	-
Simon64/96	130	Beaulieu et al. (2013)	1216	142,2	-	-
Simon64/128	130	Beaulieu et al. (2013)	958	4,2	-	-
Simon64/128	130	Beaulieu et al. (2013)	1417	133,3	-	-
Simon64/128	13	Yang et al. (2015)	944	4,2	0,762	181,4
Simon64/128	13	Yang et al. (2015)	1403	133,3	1,239	9,295
Simon64/128	65	Yang et al. (2015)	845	4,2	2,336	556,2
Simon64/128	65	Yang et al. (2015)	1305	133,3	3,398	25,49

8 Comparison

Due to the security concerns mentioned in section 4 it is reasonable to conclude TEA is less secure than the other ciphers discussed in this paper, especially when used as a hash function. Full-round attack have also been shown for PRESENT, and as the design assumed little data to be sent this cipher may not be secure enough either, especially for applications that require large amounts of data to be encrypted. In contrast, no known full-round attacks on XTEA and Simon are known.

With respect to the performance of the XTEA, PRESENT and Simon ciphers a summary of relatively well performing implementations mentioned in this paper is shown in table 5. As can be seen, XTEA requires a relatively large implementation area, while implementations larger than 3000 GE are possibly no longer acceptable for constrained devices, as noted by Hatzivasilis et al. (2018). PRESENT and Simon both achieve low energy usage per bit and achieve much smaller sizes. Simon especially, with implementations of some versions even as small as 517 GE, as noted in section 7.2.

Table 5: Performances of best discussed implementations.

cipher	technology (nm)	source	area (GE)	throughput (kbps)	power(μW)	energy/bit (pJ/bit)
XTEA	130	Kaps (2008)	3490	57,1	19,5	341,5
XTEA	90	Kitsos et al. (2012)	3490	200	61	305
PRESENT-80	180	Bogdanov et al. (2007)	1570	200	5	10
Simon64/128	13	Yang et al. (2015)	944	4,2	0,762	181,4
Simon64/128	13	Yang et al. (2015)	1403	133,3	1,239	9,295

9 Discussion

The results found do not disagree with previous work. Notes on the insecurity of TEA, impracticality of hardware implementations of XTEA agree with the research presented by Hatzivasilis et al. (2018). The results showing PRESENT and Simon to perform relatively well are also in agreement. The attacks described by Faghihi Sereshgi et al. (2016) & Lee (2014) are however not considered by Hatzivasilis et al., as such PRESENT may be less secure than previously described.

As metrics were carefully chosen in an attempt to prevent architecture choice from influencing performance comparison, the results should hold regardless of the architectural differences in the implementations, although relatively small differences can be seen regarding the effect of architecture choice in table 3.

More interesting and less clear is the possible security difference between PRESENT-80 and PRESENT-128, as noted in section 6.1. As this is not yet well researched no strict stance can be taken on the topic however.

10 Responsible Research

The results found during this comparative study were reached without biased towards any particular cryptographic scheme and should be reproducible as all sources used to reach these results are referenced. New research into the vulnerabilities of the discussed cryptographic schemes may however make these results antiquated in the future.

Note that, as this comparative study is of limited scope, and as compromised IoT devices can potentially pose serious threats, as discussed in the introduction, this paper should not form the sole basis of important security decisions.

11 Conclusion and Future Work

The research presented in this report seems to suggest TEA is not a good fit for use as a secure cryptographic cipher in IoT, especially when compared to its XTEA. Similar concerns exist for PRESENT due to possible security vulnerabilities. XTEA also seems unacceptable due to its large footprint in ASIC implementations. The Simon ciphers meanwhile seem to provide good performance when implemented, both regarding implementation size and energy usage per bit. Additionally its flexibility may be useful as different applications may have different constraints and require different levels of security.

Future research into the security of Simon is advisable, as this may be a good candidate for use in IoT devices. Additionally, research into the security differences between PRESENT-80 and PRESENT-128 may be useful, as this could provide information regarding the cause of the potential vulnerability difference discussed in this paper, which could inform future design decisions. Research into the performance and security of the extensions presented in appendix A may also be useful in guiding future design decisions.

References

Abdelhalim, m. b., El-Mahallawy, M., & Ayyad, M. (2013, 06). Design and implementation of an encryption algorithm for use in rfid system. *International Journal of RFID Security and Cryptography*, *2*, 51-57. doi: 10.20533/ijrfidsc.2046.3715.2013.0007

Abed, F., Forler, C., List, E., Lucks, S., & Wenzel, J. (2012). Biclique cryptanalysis of the present and led lightweight ciphers. *IACR Cryptol. ePrint Arch.*, *2012*, 591.

Abomhara, M., & Køien, G. M. (2015). Cyber security and the internet of things: Vulnerabilities, threats, intruders and attacks. *Journal of Cyber Security and Mobility*, *4*(1), 65–88. doi: 10.13052/jcsm2245-1439.414

Alizadeh, M., Salleh, M., Zamani, M., Jafar, S., & Sasan, K. (2012). Security and Performance Evaluation of Lightweight Cryptographic Algorithms in RFID. *Recent Researches in Communications and Computers*(November 2015), 45–50. Retrieved from `http://goo.gl/ej5iEr`

AlKhzaimi, H., & Lauridsen, M. M. (2013). Cryptanalysis of the simon family of block ciphers. *IACR Cryptol. ePrint Arch.*, *2013*, 543.

Andem, V. R. (2003). *A cryptanalysis of the tiny encryption algorithm* (Unpublished doctoral dissertation). University of Alabama.

Appel, M., Pauer, C., & Wiesmaier, A. (2016, September). *Security aspects and comparison of block ciphers led and tea.* Retrieved from `http://tubiblio.ulb.tu-darmstadt.de/104854/`

Aradhyamath, S., & Paulose, J. (2018, 04). Multi-key modified tiny encryption algorithm for healthcare. *International Journal of Engineering & Technology*, *7*, 559. doi: 10.14419/ijet.v7i2.9894

Balaji, S., Nathani, K., & Santhakumar, R. (2019). IoT Technology, Applications and Challenges: A Contemporary Survey. *Wireless Personal Communications*, *108*(1), 363–388. Retrieved from `https://doi.org/10.1007/s11277-019-06407-w` doi: 10.1007/s11277-019-06407-w

Beaulieu, R., Shors, D., Smith, J., Treatman-Clark, S., Weeks, B., & Wingers, L. (2013). *The simon and speck families of lightweight block ciphers.* Cryptology ePrint Archive, Report 2013/404. (`https://eprint.iacr.org/2013/404`)

Bogdanov, A., Knudsen, L. R., Leander, G., Paar, C., Poschmann, A., Robshaw, M. J. B., ... Vikkelsoe, C. (2007). Present: An ultra-lightweight block cipher. In P. Paillier & I. Verbauwhede (Eds.), *Cryptographic hardware and embedded systems - ches 2007* (pp. 450–466). Berlin, Heidelberg: Springer Berlin Heidelberg.

Chen, H., & Wang, X. (2016). Improved linear hull attack on round-reduced simon with dynamic key-guessing techniques. In T. Peyrin (Ed.), *Fast software encryption* (pp. 428–449). Berlin, Heidelberg: Springer Berlin Heidelberg.

Cho, J. Y. (2010). Linear cryptanalysis of reduced-round present. In J. Pieprzyk (Ed.), *Topics in cryptology - ct-rsa 2010* (pp. 302–317). Berlin, Heidelberg: Springer Berlin Heidelberg.

De Canniere, C., Biryukov, A., & Preneel, B. (2006). An introduction to block cipher cryptanalysis. *Proceedings of the IEEE*, *94*(2), 346–356.

De Leon, R. M., Sison, A. M., & Medina, R. P. (2019). A modified tiny encryption algorithm using key rotation to enhance data security for internet of things. In *2019 international conference on information and communications technology (icoiact)* (p. 56-60). doi: 10.1109/ICOIACT46704.2019.8938456

Faghihi Sereshgi, M. H., Dakhilalian, M., & Shakiba, M. (2016, January). Biclique cryptanalysis of mibs-80 and present-80 block ciphers. *Sec. and Commun. Netw.*, *9*(1), 27â33. Retrieved from `https://doi-org.tudelft.idm.oclc.org/10.1002/sec.1375` doi: 10.1002/sec.1375

FBI. (2016, March). *Motor vehicles increasingly vulnerable to remote exploits.* Retrieved from `https://www.ic3.gov/Media/Y2016/PSA160317#fn1`

Greenberg, A. (2020, Nov). *This bluetooth attack can steal a tesla model x in minutes.* Conde Nast. Retrieved May 18, 2021, from `https://www.wired.com/story/tesla-model-x-hack-bluetooth/`

Gulcan, E., Aysu, A., & Schaumont, P. (2015). A flexible and compact hardware architecture for the simon block cipher. In T. Eisenbarth & E. Öztürk (Eds.), *Lightweight cryptography for security and privacy* (pp. 34–50). Cham: Springer International Publishing.

Hatzivasilis, G., Fysarakis, K., Papaefstathiou, I., & Manifavas, C. (2018, Jun 01). A review of lightweight block ciphers. *Journal of Cryptographic Engineering*, *8*(2), 141-184. Retrieved from `https://doi.org/10.1007/s13389-017-0160-y` doi: 10.1007/s13389-017-0160-y

Hernandez-Castro, J. C., Peris-Lopez, P., & Aumasson, J.-P. (2012). On the key schedule strength of present. In J. Garcia-Alfaro, G. Navarro-Arribas, N. Cuppens-Boulahia, & S. de Capitani di Vimercati (Eds.), *Data privacy management and autonomous spontaneus security* (pp. 253–263). Berlin, Heidelberg: Springer Berlin Heidelberg.

Intel. (n.d.). *Comparing fpgas, structured asics, and cell-based asics.* Retrieved from `https://www.intel.com/content/www/us/en/products/programmable/fpga-vs-structured-asic.html`

Isobe, T., & Shibutani, K. (2012). Security analysis of the lightweight block ciphers xtea, led and piccolo. In W. Susilo, Y. Mu, & J. Seberry (Eds.), *Information security and privacy* (pp. 71–86). Berlin, Heidelberg: Springer Berlin Heidelberg.

Israsena, P. (2005). Design and implementation of low power hardware encryption for low cost secure rfid using tea. In *2005 5th international conference on information communications signal processing* (p. 1402-1406). doi: 10.1109/ICICS.2005.1689288

Kaps, J.-P. (2008). Chai-tea, cryptographic hardware implementations of xtea. In D. R. Chowdhury, V. Rijmen, & A. Das (Eds.), *Progress in cryptology - indocrypt 2008* (pp. 363–375). Berlin, Heidelberg: Springer Berlin Heidelberg.

Kelsey, J., Schneier, B., & Wagner, D. (1997). Related-key cryptanalysis of 3-way, biham-des,cast, des-x, newdes, rc2, and tea. In Y. Han, T. Okamoto, & S. Qing (Eds.), *Information and communications security* (pp. 233–246). Berlin, Heidelberg: Springer Berlin Heidelberg.

Kerckhof, S., Durvaux, F., Hocquet, C., Bol, D., & Standaert, F.-X. (2012). Towards green cryptography: A comparison of lightweight ciphers from the energy viewpoint. In E. Prouff & P. Schaumont (Eds.), *Cryptographic hardware and embedded systems – ches 2012* (pp. 390–407). Berlin, Heidelberg: Springer Berlin Heidelberg.

Khan, G. N., & Moessner, M. B. (2011). Secure authentication protocol for rfid systems. In *2011 proceedings of 20th international conference on computer communications and networks (icccn)* (p. 1-7). doi: 10.1109/ICCCN.2011.6006010

Kitsos, P., Sklavos, N., Parousi, M., & Skodras, A. N. (2012). A comparative study of hardware architectures for lightweight block ciphers. *Computers Electrical Engineering*, *38*(1), 148-160. Retrieved from `https://www.sciencedirect.com/science/article/pii/S0045790611001984` (Special issue on New Trends in Signal Processing and Biomedical Engineering) doi: https://doi.org/10.1016/j.compeleceng.2011.11.022

Ko, Y., Hong, S., Lee, W., Lee, S., & Kang, J. S. (2004). Related key differential attacks on 27 rounds of XTEA and full-round GOST. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *3017*, 299–316. doi: 10.1007/978-3-540-25937-4_19

Lee, C. (2014). Biclique cryptanalysis of present-80 and present-128. *The Journal of Supercomputing*, *70*(1), 95–103.

Liebers, J., Abdelaziz, M., Mecke, L., Saad, A., Auda, J., Gruenefeld, U., ... Schneegass, S. (2021). Understanding user identification in virtual reality through behavioral biometrics and the effect of body normalization. In *Proceedings of the 2021 chi conference on human factors in computing systems.* New York, NY, USA: Association for Computing Machinery. Retrieved from `https://doi-org.tudelft.idm.oclc.org/10.1145/3411764.3445528`

Lu, J. (2009). Related-key rectangle attack on 36 rounds of the xtea block cipher. *International Journal of Information Security*, *8*(1), 1–11. doi: 10.1007/s10207-008-0059-9

Mohd, B. J., Hayajneh, T., & Vasilakos, A. V. (2015). A survey on lightweight block ciphers for low-resource devices: Comparative study and open issues. *Journal of Network and Computer Applications*, *58*, 73-93. Retrieved from `https://www.sciencedirect.com/science/article/pii/S1084804515002076` doi: https://doi.org/10.1016/j.jnca.2015.09.001

Needham, R. M., & Wheeler, D. J. (1997). Tea extensions. *International Journal of Knowledge-Based and Intelligent Engineering Systems*, *12*(October 1996), 3–6. Retrieved from `http://iospress.metapress.com/content/4021v8r825k460n7/`

Oka, D. K., Furue, T., Langenhop, L., & Nishimura, T. (2014). Survey of vehicle IoT bluetooth devices. *Proceedings - IEEE 7th International Conference on Service-Oriented Computing and Applications, SOCA 2014*, 260–264. doi: 10.1109/SOCA.2014.20

Pandey, J. G., Goel, T., Nayak, M., Mitharwal, C., Karmakar, A., & Singh, R. (2018). A high-performance vlsi architecture of the present cipher and its implementations for socs. In *2018 31st ieee international system-on-chip conference (socc)* (p. 96-101). doi: 10.1109/SOCC.2018.8618487

Rashidi, B. (2019). High-throughput and flexible asic implementations of simon and speck lightweight block ciphers. *International journal of circuit theory and applications*, *47*(8), 1254–1268.

Ronen, E., & Shamir, A. (2016). Extended functionality attacks on iot devices: The case of smart lights. In *2016 ieee european symposium on security and privacy (euros p)* (p. 3-12). doi: 10.1109/EuroSP.2016.13

Saarinen, M.-J. (1998). Cryptanalysis of block tea. *Unpublished manuscript, October*.

Sethi, P., & Sarangi, S. R. (2017). Internet of Things: Architectures, Protocols, and Applications. *Journal of Electrical and Computer Engineering*, *2017*. doi: 10.1155/2017/9324035

Shepherd, S. J. (2007). The Tiny Encryption Algorithm. *Cryptologia*, *31*(3), 233–245. doi: 10.1080/01611190601090606

Wheeler, D. J., & Needham, R. M. (1994). Tea, a tiny encryption algorithm. In *International workshop on fast software encryption* (pp. 363–366).

Wheeler, D. J., & Needham, R. M. (1998). Correction to xtea. *Unpublished manuscript, Computer Laboratory, Cambridge University, England*, *1*(2), 17.

Yang, G., Zhu, B., Suder, V., Aagaard, M., & Gong, G. (2015). The simeck family of lightweight block ciphers. *IACR Cryptol. ePrint Arch.*, *2015*, 612.

Yap, H., Khoo, K., Poschmann, A., & Henricksen, M. (2011). Epcbc - a block cipher suitable for electronic product code encryption. In D. Lin, G. Tsudik, & X. Wang (Eds.), *Cryptology and network security* (pp. 76–97). Berlin, Heidelberg: Springer Berlin Heidelberg.

A Tiny Encryption Algorithm Extensions

Many extensions to TEA exist in attempts to improve the security or speed of the algorithm, at least three of which were made by the original authors. As discussed, XTEA is one of those, however some additional extensions are mentioned below, although this does not comprise an extensive list.

A.1 Block Tiny Encryption Algorithm

The Block Tiny Encryption Algorithm (Block TEA) was introduced simultaneously with XTEA (Needham & Wheeler, 1997), and was designed for encrypting larger data blocks by cyclically repeating the XTEA round function on parts of the block.

Vulnerabilities

The description of an effective attack against Block TEA by Saarinen (1998) however lead to the creation of a Corrected Block TEA (also called XXTEA) Wheeler & Needham (1998). For this version Saarinen (1998) however also points out a possible vulnerability.

A.2 Multi-key Modified Tiny Encryption Algorithm

In a paper by Aradhyamath & Paulose (2018) a multi-key modified version of TEA (referred to as MMTEA in this paper) is suggested that uses 2 keys instead of 1 by repeating the round function on the original result of the single key round function, but using the second key.

Vulnerabilities

Testing of this modified algorithm provided along with its proposal (Aradhyamath & Paulose, 2018) seems to show that, when the two keys used are not identical, the equivalent key relation discussed before no longer holds.

A.3 Modified Tiny Encryption Algorithm I

Abdelhalim et al. (2013) suggest an additional Modified TEA extension, calling it MTEA (referred to as MTEA I in this paper), that relies on a pseudo random number generator (PRNG) in an attempt to improve the security of TEA. In this way, the key used each cycle is changed, also changing the key required for encryption to the last used key during encryption.

Vulnerabilities

In their proposal, (Abdelhalim et al., 2013) provides some completeness and avalanche effect testing results for both TEA and MTEA in an attempt to test the security strength of the algorithm, with

the results being slightly better.

Completeness here refers to the reliance of every part of the output on every part of the input, while the avalanche effect refers to the desire that small changes in the input should change a lot in the output.

Abdelhalim et al. also provided reference to a paper of theirs that seems to provide more of this testing of their proposed scheme, although this does not seem to be accessible.

A.4 Modified Tiny Encryption Algorithm II

De Leon et al. (2019) also suggest a Modified TEA extension (referred to as MTEA II in this paper) that shifts the 4 quarters of the key in use every cycle, in an attempt to improve TEA's security. The authors also refer to several other TEA modifications, including the two mentioned above.

Vulnerabilities

In the same manner as MTEA I, completeness and avalanche effect testing results were provided along with the proposal (De Leon et al., 2019). Equivalent key tests were also included, comparable to those discussed for MMTEA. The testing results seem promising, as MTEA II performed better in these tests than TEA. The equivalent key relation also no longer holds for this algorithm.

The authors notably suggest "cryptanalysis could be done to the modified TEA to determine its weakness". This also holds for the 2 other modified TEA algorithms mentioned above, as they are less studied than TEA and XTEA have been.

B PRESENT

A description of the S-box PRESENT uses is given in table 6 using hexadecimal notation, where x is the 4 bit input and S[x] is its 4 bit ouput. A description of the P-box PRESENT uses is given in table 7, where i refers to the bit position of input and P(i) refers to the corresponding bit position in the output of this input bit. Note due to the symmetry of this permutation, the bit position can be taken from both the left as right most bit.

Table 6: Substitution by the S-box used in PRESENT. (Bogdanov et al., 2007)

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
S[x]	C	5	6	B	9	0	A	D	3	E	F	8	4	7	1	2

Table 7: Permutation by the P-box used in PRESENT. (Bogdanov et al., 2007)

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
P(i)	0	16	32	48	1	17	33	49	2	18	34	50	3	19	35	51
i	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
P(i)	4	20	36	52	5	21	37	53	6	22	38	54	7	23	39	55
i	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
P(i)	8	24	40	56	9	25	41	57	10	26	42	58	11	27	43	59
i	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
P(i)	12	28	44	60	13	29	45	61	14	30	46	62	15	31	47	63