Online Resource 2:
Experimentation Data, Client-side Encryption and Device Provisioning Details

February, 2022

0.1 Low-cost Client-Side Encryption and Secure IoT Provisioning

In this section, we present the details of our experimentation process towards provisioning the SAMG55 microprocessor equipped with the ATWINC1500 and ATECC608A onto the AWS IoT core. First, we experimented the degree of resource constrain in IoT devices by comparing a Laptop and SAMG55 implementations of AES128 and the low-cost algorithm to be used for client-side encryption. We experimented the avalanche effect test on the low-cost algorithm and compared the encryption completion time to that of lightweight CLEFIA, after which the low-cost algorithm is implemented as a client-side encryption solution to securely provision the sample IoT device using AWS cloud services programmatic access tools. Fig. 1 in the Online Resource 1 shows the experimental setup. We utilized the Amazon Web Services (AWS) Command line Interface (CLI) tools for enabling programmatic access to the AWS IoT core cloud service and provisioning of the SAMG55 device. Zerynth studio, which is a hybrid (C and python) Integrated Development Environment was used for implementing and compiling the executable binaries of the reduced round algorithm for the sample client-side encryption of data. The cloud end of this experimental setup requires the creation of a cloud account with the AWS cloud platform, enabled with the AWS lambda and AWS IoT Core services.

0.1.1 Comparative Analysis of IoT Devices' constraints

As an experimentation of the degree of resource constraint on a typical IoT device, we carried out a comparative analysis of the encryption completion time of the AES-128 and an efficient algorithm for constrained IoT devices detailed in [1]. We compared the implementations on a laptop computer (64-bits operating system, 8GB RAM and x64-based intel processor (core i5)) and a SAMG55 32-bit cortex M4 microprocessor which features 512Kbytes of Flash and 164Bytes of RAM, with the aim of estimating the degree of resource constrain on the SAMG55 with respect to the PC. According to the data sheet [18], The SAMG55 devices are general-purpose low-power micro-controllers that are suitable for wide range of IoT deployments. Firstly, we compared the PC and SAMG55 implementations of the standard AES algorithm. Fig. 1 shows a plot of the difference in the completion time of encrypting one byte of data on the laptop in comparison to the SamG55 micro-controller using the standard AES-128 algorithm. Consequent upon the constraint of computing resources of the SAMG55 device, our experimentation shows a 657% increase in the encryption completion time using the SAMG55 device in comparison to the encryption completion time using the PC, the constants being the message size, key-length, and encryption cipher while the variables are the computing resources available to the SAMG55 and the PC respectively. This is served as a demonstration of scarce computing resources in a typical IoT device and hence, the need for low-cost client-side encryption without compromise on security.

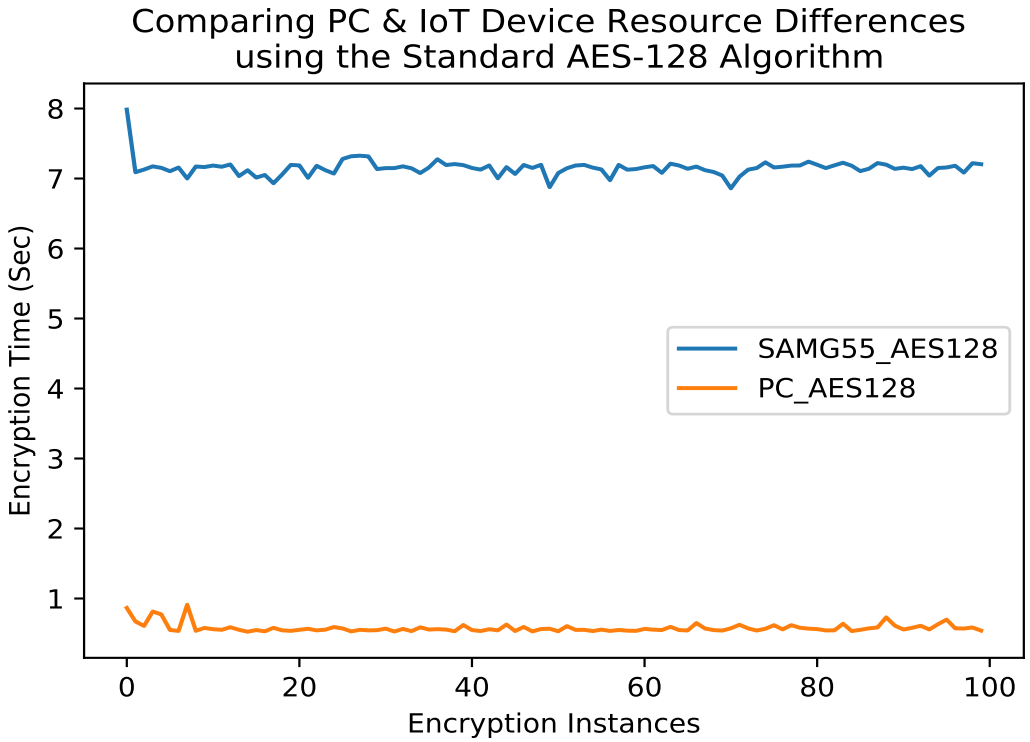

Figure 1: A comparison of the PC and SAMG55 Implementation of the Standard AES-128 Algorithm

The SAMG55 implementation of the reduced round algorithm however shows a 331% increase in the encryption completion time in comparison to the PC implementation. In contrast to the PC and SAMG55 comparison of the encryption completion time, the low-cost algorithm shows a 50.3% reduction in the encryption completion time in favour of the resource constrain of the IoT device. Similar to the comparative analysis for the AES-128, holding the constants message size (16bytes), key size and cipher, while the variables remain the difference in the availability of computing resources on the PC and the SAMG55 device. Fig. 2 shows the plot of the experimentation data.

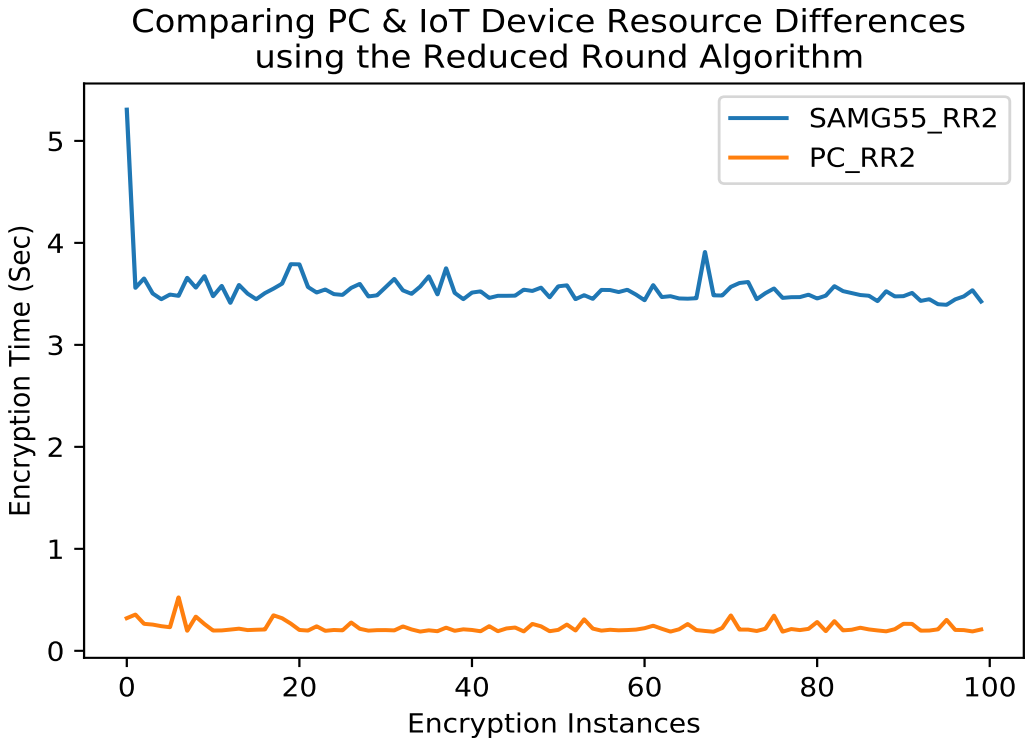

Figure 2: A comparison of the PC and SAMG55 Implementation of the Reduced Round Algorithm

0.1.2 Low-cost Client-Side Encryption

Consequent upon the scarcity of computing resources on a typical IoT device as experimented and analyzed in section 0.1.1, a low-cost algorithm based on the (AES) is utilized for experimenting a client-side encryption of 16bytes of data. In [1], we detailed the reduced-complexity algorithm based on the AES in three stages viz: a cryptanalytic overview of the consequence of the reduction, a mathematical justification of complexity reduction based on the core algebraic properties of the AES cipher and a software implementation of the reduced round algorithm respectively. Based on the AES standard cipher, the reduced round algorithm executes the round function in two iterations using a total of 48bytes scheduled key as against the 176bytes of the standard AES, for every block of the plain text, following the process of key-whitening, initialization and the execution of the round function as diagrammatically shown in Fig. 3.

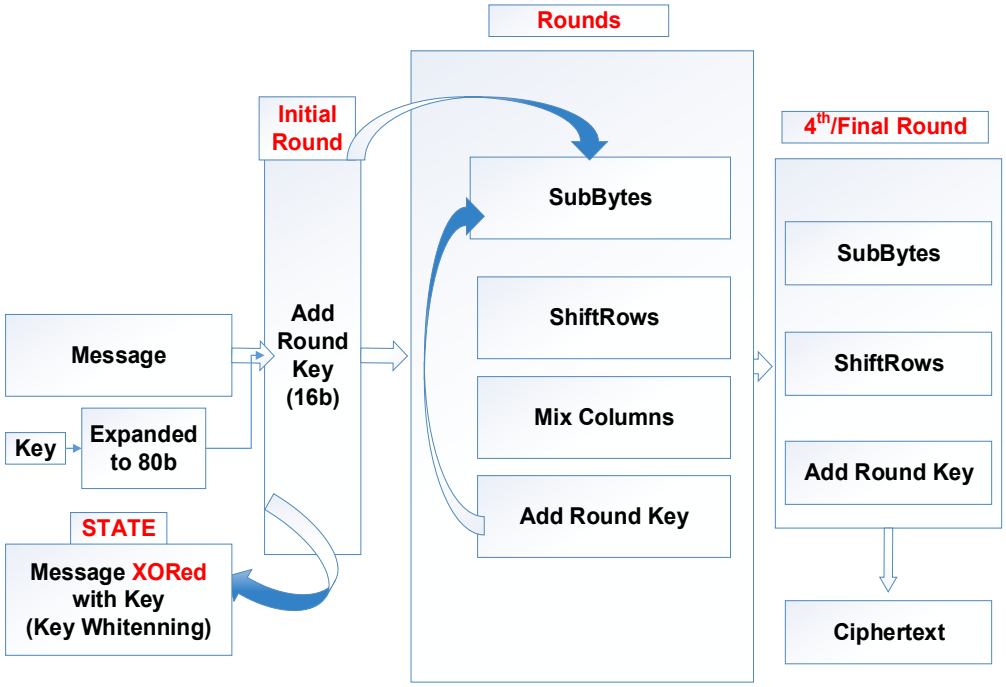

Figure 3: Diagrammatic structure of the Reduced Round Algorithm

The algorithm table 1 presents an algorithmic sequence of execution of the reduced round algorithm. Favourable to the experimentation and analysis of resource constrain of IoT devices presented in section 0.1.1 the reduced round algorithm is 35% cheaper than the standard AES algorithm in terms of the encryption completion time of a single block client data as shown in Fig. 4, and thus, a plausible candidate for client-side encryption of IoT device data before outsourcing to the cloud for storage or processing.

0.1.3 Experimental Setup

Experimentation tools used include a laptop computer and a SAMG55 microprocessor in terms of hardware. Software level components of the implementation tools include the Zerynth studio -which runs python optimized with C. Two platforms adjudged to suffice for the aim of experimentation include a laptop computer as a resource-sufficient platform and a SAMG55 microprocessor as a resource constrained platform as well as a typical IoT device. While the goal on one hand is to utilize the notion of percentages to express the impact of resource constraint on the sample constrained device in comparison to the resource-sufficient laptop, on the other hand is to observe and compare the attributes of complexity between the aforementioned algorithms on both platforms in a mutually exclusive context. For each instance of encryption, the encryption is repeated for one thousand iterations and the average execution time of the one thousand iterations is logged. The notion of average as detailed in [1] is applied to obtain statistically relevant values in terms of a representative

Algorithm 1 Client-Side-Encryption Execution Flow

Message, Key initialization of the counter $i = 0$ and $Nbr = 2$
Expand key to length: (block size) *Nbr + block size
STATE = message XORed with Key (Key whitening)
Invoke the round function:
while $i < Nbr$: **do**
    STATE = SubByte(STATE)
    STATE = ShiftRows(STATE)
    if $i < Nbr$: **then**
        STATE = MixColumn(STATE)
    end
    Invoke addRoundKey(STATE, NextRoundkey)
end
STATE as resulting Ciphertext

number for the encryption completion time of a single message block. Furthermore, this experiment is repeated for one hundred encryption instances for the distinct key lengths: 128, 192 and 256 of the standard AES algorithm on both platforms. Following the same setup and using the same platforms of experimentation, the lightweight clefia algorithm was also implemented, and finally the Efficient security algorithm for Constrained IoT devices. Data obtained with respect to the aforementioned experiments is presented in table table 5 and 6. The results and analysis detailed section in [1] and the Main document further show the details these experimental data presented in table 5 and 6 for the laptop-resource-sufficient platform and the SAMG55 resource-constrained platform respectively. The algorithm table 1 details the Efficient Security Algorithm for Constrained IoT Devices as implemented, whereby the iteration number of the round function is experimentally reduced to four and two respectively as evidenced by the experimentation data.

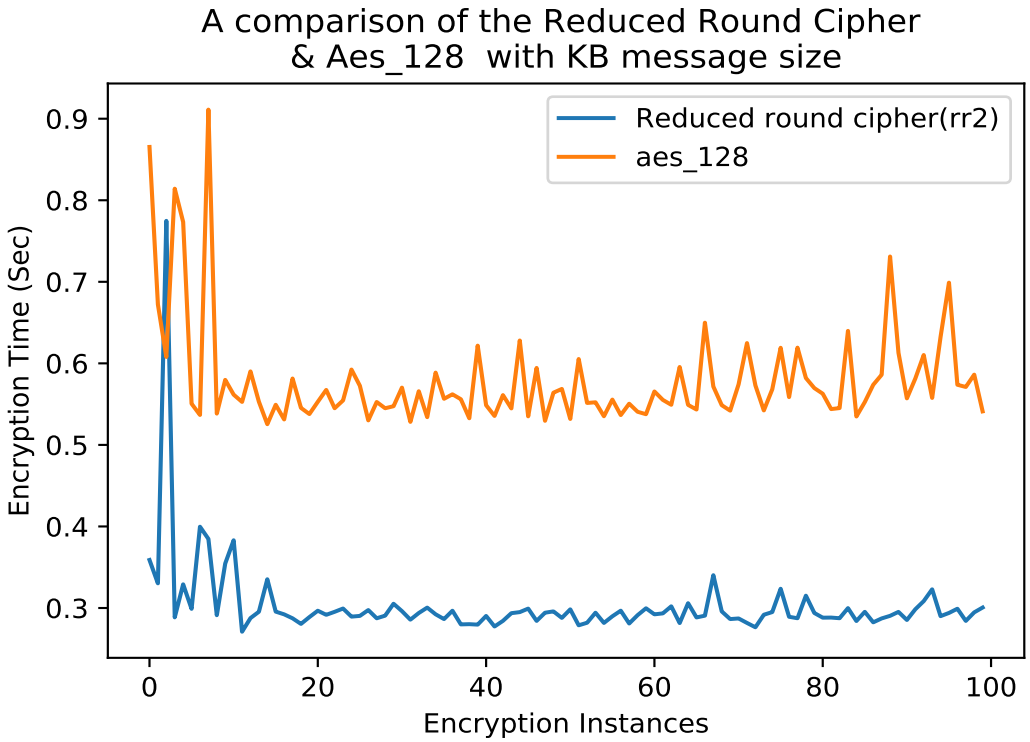

Figure 4: A comparison of the Reduced Round Cipher (rr2) and standard AES-128

Lightweight CLEFIA versus the Low-cost algorithm

We implemented and compared the cost in terms of encryption completion times, of lightweight CLEFIA and the Efficient Algorithm for Power Constrained IoT devices. First, we showed the complexities of the standard key lengths of CLEFIA as shown in Fig. 5, then a comparison of the least complex and the Efficient Algorithm for Constrained IoT devices followed as shown in Fig. 6. Further to the secure reduction of rounds of the AES round function, [1] trades-off by leveraging a tamper proof Secure Element, towards reduction of complexity in tandem with resource constrain in a sample SAMG55 IoT device as analyzed in section 0.1.1 and without compromise on security. In a related work detailed in [2], CLEFIA emerged a plausible lightweight cipher in terms of requiring the least computing resources when compared to CAMELLIA, SEED and the standard AES.

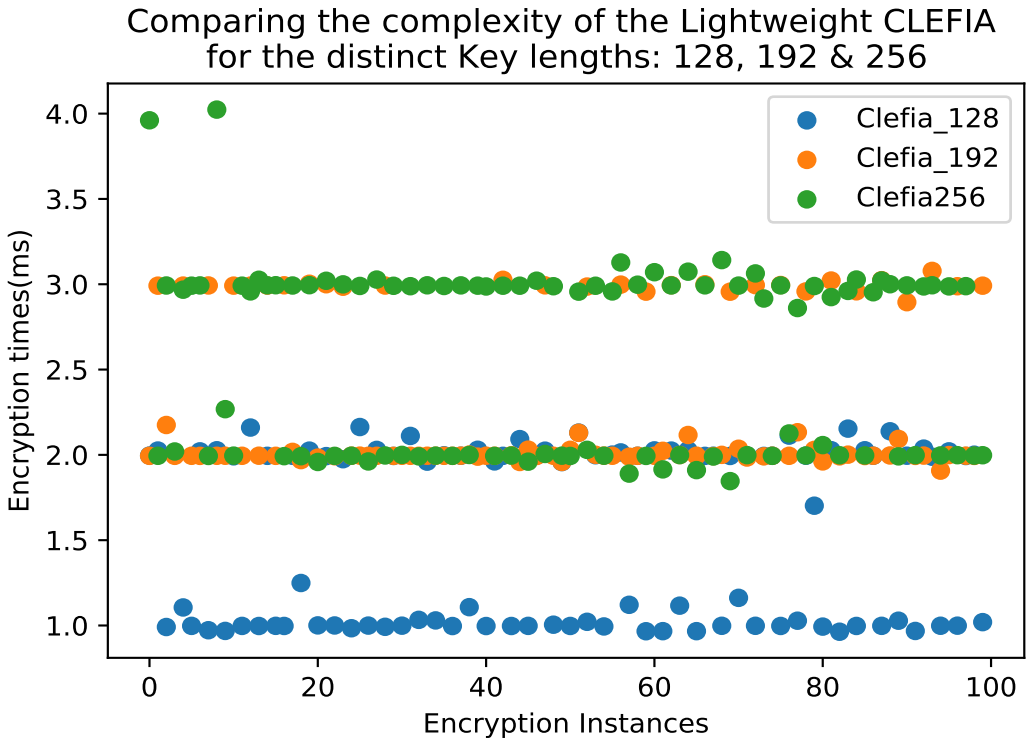

Figure 5: Complexity of CLEFIA distinct key lengths

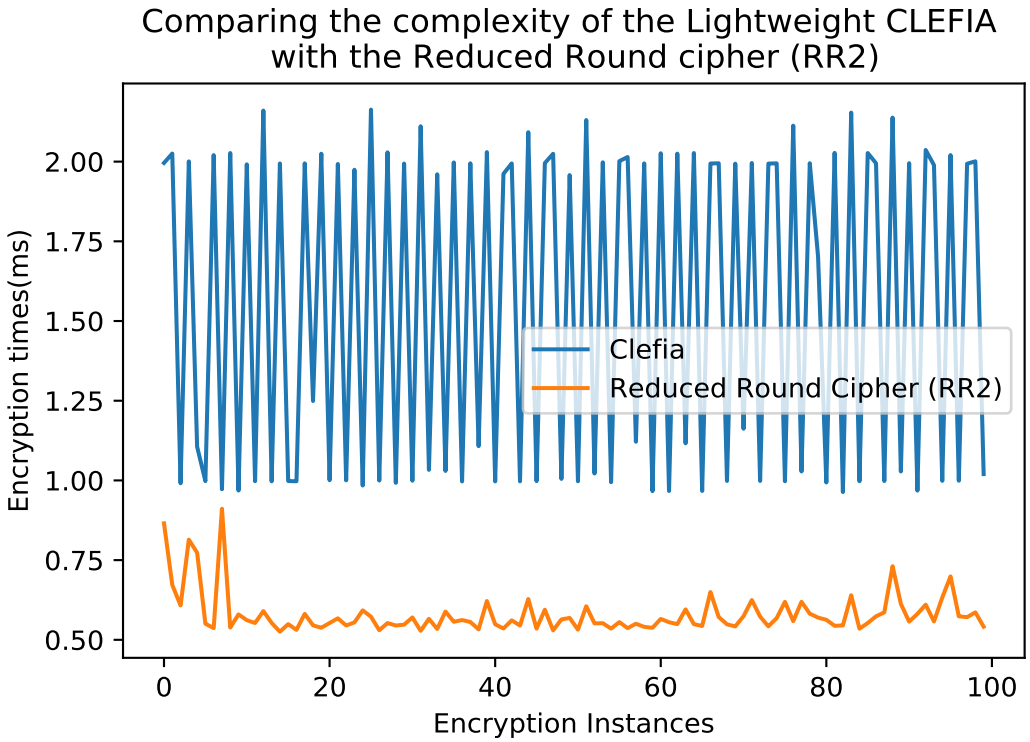

Figure 6: Comparison of lightweight CLEFIA and the Reduced Round Cipher (RR2)

Fig. 7, shows a plot of a comparison of the rate of growing complexity between the distinct key lengths of CLEFIA and that of the AES, on which the efficient algorithm is based. This was done with respect to the factors of compatibility between CLEFIA and the AES -and by extension the efficient algorithm, being: consisting of three key variants of 128, 192 and 256 bits and message blocksize of 16bytes. Our results reveal that although the encryption completion times for a single block, of CLEFIA as a lightweight algorithm is cheaper than the AES, the rate of growing complexity between the distinct key lengths between AES128 and AES256 is lower than the corresponding rate of growing complexity between CLEFIA128 and CLEFIA256.

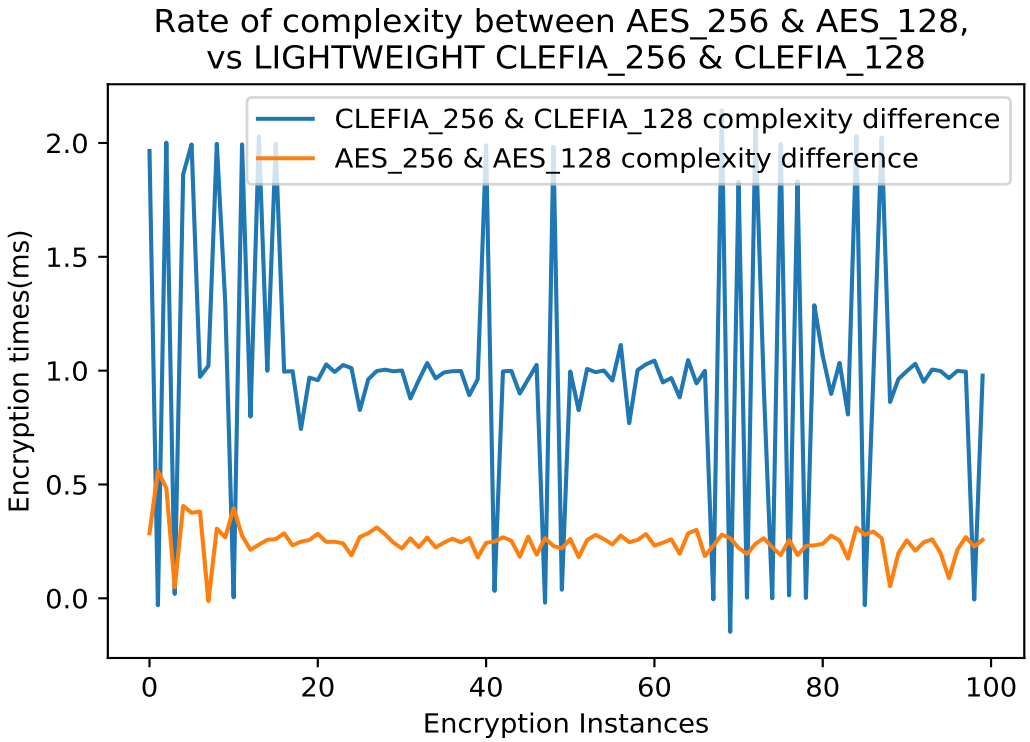

Figure 7: A measure of the rate of complexity between AES256 & AES128, vs CLEFIA256 & CLEFIA128

Avalanche Effect

The avalanche effect properties of a cipher is measured in terms of the rate of change in the cipher text with respect to changes in the plain text. According to [3], the ideal avalanche effect for a good cipher should be at least 50%, which means that a single change in the plain text should alter the outcome of the cipher text by at least 50% when encrypted by the cipher.

We experimented the avalanche effect test on the Low-cost algorithm to consolidate the mathematical justification of the round reduction detailed in [1], to demonstrate the proficiency of the low-cost algorithm for client-side encryption in provisioning the sample IoT device. The results of the avalanche effect experiments is detailed in tables 1 and 4. Table 1 shows the sample plain-text and cipher-text for the low-cost algorithm, against which the result in table 4 is compared whereby a byte level flip is performed on the block of the sample plain-text, and the observed avalanche effect is recorded. The block bytes which reoccur in the output cipher-text block when contrasted against the corresponding original cipher-text in 1 flagged red in the table 4. The arithmetic average of the five instances of avalanche effect experiment of the low-cost algorithm is then computed using $\sum_{i=1}^{5} x_i$, where x is an instance of the avalanche effect experiment, which yields 93.75%.

Furthermore, the efficient algorithm for constrained IoT devices is utilized as a client-side encryption solution for a sample IoT device, preceding the secure provisioning of the IoT

device onto the AWS IoT cloud platform. According to a comparative study on the features of the foremost Cloud Services Providers (CSP): AWS, GCP and Microsoft Azure -detailed in [4–6], the AWS cloud platform proves its leadership among the CSPs by maintaining about 33% share in the market for several notable CSP features including infrastructure, computation and storage to mention a few and thus, motivated the choice of use as an experi-mentation platform based on its aforementioned dominance as a platform leader among CSPs.

A summary of this experimentation and comparison of the efficient algorithm for constrained IoT devices and the lightweight CLEFIA is presented in the 0.1.4 section.

0.1.4 Secure Provisioning

The process flow for provisioning the sample IoT device is presented in the algorithm table 2. Beginning with the initialization of the IoT device equipped with the ATECC608A, the device-to-cloud authentication mechanism is invoked leveraging the tamper-proof security keys in the ATECC608A device.

Algorithm 2 Device Provisioning Process Flow

Initializing the IoT device and the ATECC608A secure element
Invoking device-cloud authentication leveraging the ATECC608A tamper-proof security keys
while *Creation and registration of a Certificate Authority (CA) and the IoT device's security credentials* **do**
    Create a Certificate Authority's root certificate
    $\leftarrow Certificate$
    Invoke the IoT device's certificate signing request to a certificate signer Certificate Authority
    sign the certificate signing request using the root certificate
    $\leftarrow Certificate$
    register the device's digital identity using the signed certificate.
    $\leftarrow DeviceUniqueID$
end
Connect the device to the IoT cloud by Via passing the network medium credentials to the WINC1500

Device authentication keys are often build into IoT devices at the point of manufacturing. This enforces the challenge of increasing the diameter of vulnerabilities associated with the secure handling of keys, from social engineering issues to third party mistrust. By design, The ATECC608A holds tamper-proof keys usable for invoking an authentication process that rids the aforementioned challenges of social engineering and mistrust in the handling of authentication keys. More so, these keys are completely isolated from the associated IoT device and the software vulnerabilities, owing to its taper-proof architecture. The AWS IoT core offers a bi-directional communication service between IoT devices and the AWS cloud, allowing for the registration of a device, associating it with certificates, custom attributes and requires a connecting IoT device to have a mutual authentication compatibility with Just-In-Time-Registration and Transport Layer Security (TLS1.2). The Just-In-Time registration ensures that an IoT device connecting to the cloud automatically invokes a process of self registration at the time of connection. Further to it's low-power compatibility with

constrained IoT devices is the provision of a device shadow service which makes it possible to adopt the constrained IoT devices power management strategy of devices going into sleep mode when it's not processing or transmitting information, allowing for the status of the device to be updated from the cloud end and the IoT device synchronizes the updates on the shadow upon re-connection. Integrated as an authentication security functionality of the ATECC608A is a security architecture based of the Elliptic Curve Cryptography (ECC), which is primarily a variant of the Elgamal algorithm. By leveraging the secure hardware accelerator in the ATECC608A, the SAMG55 microprocessor is securely authenticated to the AWS IoT core via utilizing it's tamper-proof keys. The SAMG55 is thus, provisioned on the cloud via utilizing the AWS CLI tool and executing the the process flow outlined in the algorithm table 2

The secure authentication process requires creation and signing of requisite digital credentials required to create a unique identity for the IoT device on the cloud. Thus, the creation of a Certificate Authority's (CA) root certificate leveraging the initialized tamper-proof security keys as sequenced in 2.

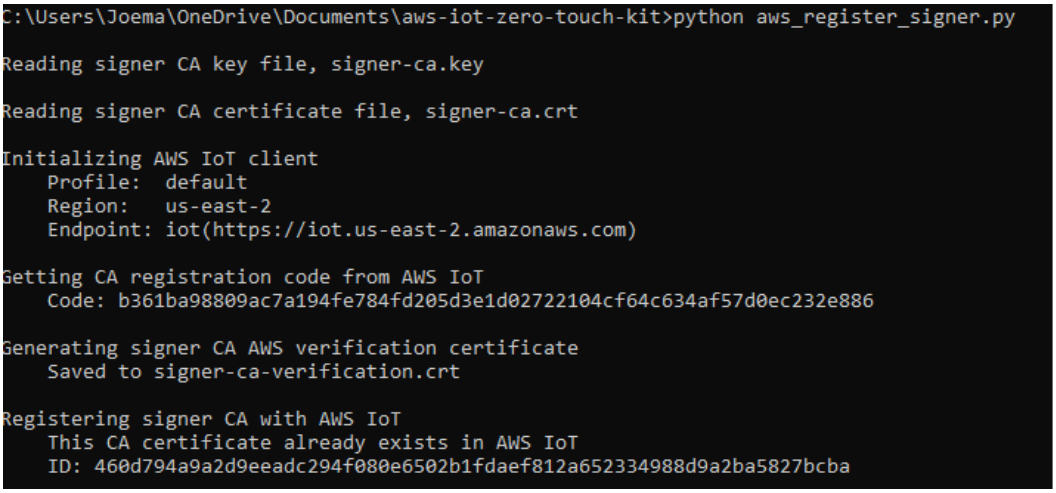

Figure 8: Certificate Authority Registration and Creation of a Unique Device ID on the Cloud

The tamper-proof functionality of the ATECC608A is leveraged to uphold the secrecy of the device private keys. Using the AWS CLI tool, a certificate signing request is invoked to a certificate signer CA and the certificate gets signed using the root certificate, and a digital identity is then created for the associated IoT device using the singed certificate. The ATWINC1500 is a low-power 802.11b/g/n that enables a network interface extension to the SAMG55 microprocessor to aid the process of connecting the device to a network medium. For this process, a WIFI network equipped with Wireless Protected Access (WPA2) security is used. The WIFI medium credentials viz: the Service Set Identifier (SSID) and password of the WIFI is then successfully passed onto the ATWINC1500 using the AWS CLI programmatic access tool as shown in 9

For each of 2, 3 and 4, a byte level flip is performed on the block of the sample plaintext, and the observed avalanche effect is recorded. The block bytes which reoccur in the

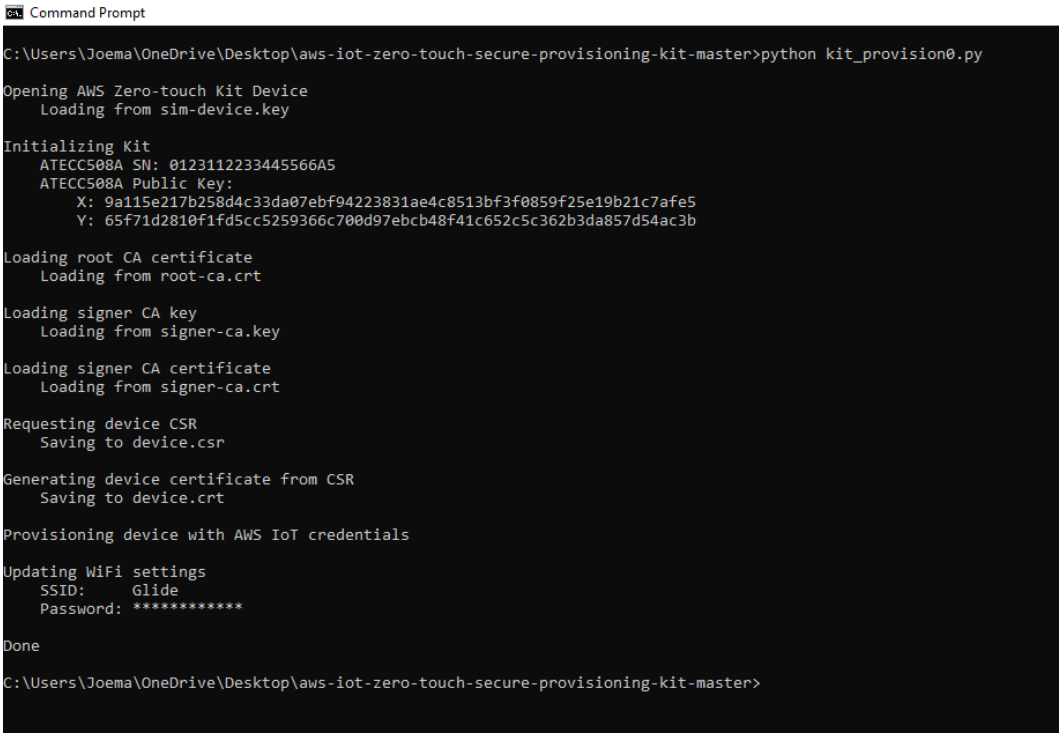

Figure 9: IoT Device Connection to a Wifi Medium

output cipher-text block when contrasted against the corresponding original cipher-text in 1 is presented in red in tables 2, 3 and 4. The arithmetic average of the five instances of avalanche effect experiment per variant of the low-cost algorithm is then computed using $\sum_{i=1}^{5} x_i$, where x is an instance of the avalanche effect experiment, which yields 23.75%, 98% and 93.75% for 2, 3 and 4 respectively for the distinct variants of the low-cost algorithm.

According to [7], Lightweight cryptography is generally defined as the cryptography for resource constrained devices, for which Radio Frequency Identification (RFID) tags are mentioned as examples. Consequent upon the constraints on low power devices in terms of area, processing capabilities, memory and scarce power resources, Light weight cryptography emerged in efforts to ensure the security of these devices in the digital communication space. As rightly put by [2], Lightweight Cryptography (LWC) or protocols are tailored for implementations in constrained environments including RFID tags, sensors, contact less smart cards, health care devices and so on. The development of lightweight cryptographic protocols therefore, is chiefly anchored on the need to develop cheaper security schemes that are compatible with the constrained nature of these devices and without compromise to security. In [3], the performance analysis of two lightweight ciphers: CLEFIA and PRESENT was presented with respect to security strengths, throughput and resource utilization, which had the latter outperforming the former in terms of memory usage, security, and the former outperforms the latter in terms of throughput. PRESENT is a lightweight algorithm with a Substitution Permutation Network (SPN) structure, and utilizes thirty one rounds of:XOR RoundKey, S-box layer and P-layer, with a final (32nd) round which XORs the STATE

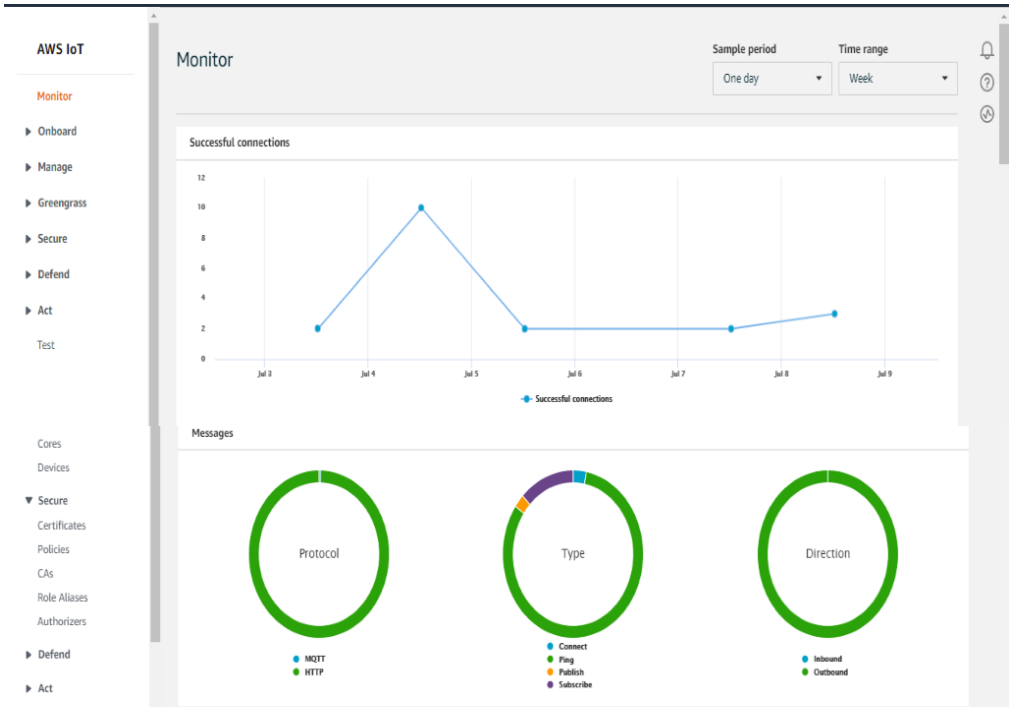

Figure 10: Cloud-end View of Provisioned IoT Device and Connection Records

produced from the first thirty one rounds and the round key. It carries out message encryption in sixty four (64) bits blocks and and supports key lengths of sixty four (64)bits and one hundred and twenty eight (128)bits. The efficiency of a lightweight cipher (CLEFIA) in comparison to other conventional ciphers such as the AES, Camelia and Seed is detailed in [2], wherein the efficiency of lightweight ciphers defined as a ratio of throughput and gate size is presented with respect to energy consumption.

0.2 Additional Details of Reseach tools

0.2.1 The secure element (SE):

ATECC608x extension board is an integral component of the SAMG55 microprocessor, equipped as a hardware cryptographic accelerator with tamper-proof capabilities to hold sensitive data and facilitate the running of secure applications. Developed by the ARM in collaboration with LoRaWAN and the Things industry, it holds the potential to automatically offload all cryptographic operations, such that keys will never be visible nor accessible, even when the associated IoT device is compromised [8]. The ARM Cortex M4 is a microprocessor designed for low energy efficient devices which supports IoT application development via the Atmel studio Integrated Development Platform and compatible with the IEEE 802.15.4 specification and lower power communications protocols including: Bluetooth Low Energy (BLE), Zigbee, Low Power Wide Area Network (LoRaWAN), Rout-

Table 1: REDUCED ROUND ALGORITHM PLAIN-TEXT AND CORRESPONDING CIPHER-TEXT

RR Variant	Message	Ciphertext
RR2	[87, 104, 97, 116, 32, 105, 115, 32, 97, 32, 99, 114, 101, 101, 108, 63]	[138, 243, 252, 95, 61, 75, 45, 57, 161, 83, 125, 88, 154, 94, 31, 120]
RR3	[87, 104, 97, 116, 32, 105, 115, 32, 97, 32, 99, 114, 101, 101, 108, 63]	[209, 22, 245, 185, 40, 157, 157, 106, 74, 172, 60, 74, 114, 85, 56, 185]
RR4	[87, 104, 97, 116, 32, 105, 115, 32, 97, 32, 99, 114, 101, 101, 108, 63]	[118, 142, 29, 129, 217, 196, 51, 217, 1, 187, 202, 118, 155, 183, 109, 5]

ing Protocol for Low power devices (RPL), Constrained application protocol (CoAP) and the Message Queueing Telemetry Transport (MQTT) messaging protocol for constrained IoT devices and support the LoRaWAN 1.x and 1.1. The ATECC608A functionality wades against implementation attacks which aim to exploit the recovery of encryption keys through manipulating the efficient security algorithm for power constrained IoT device at the point of hardware implementation. This serves as compensation for the reduced rounds and consolidating the preserved security properties of the algorithm in the perspective of brute-force and analytical attacks as detailed in section [1]. Moreover, the ATECC608A-MAHTN-T microchip offers a cost-efficient secure authentication of the associated IoT device onto the network infrastructure; as a preceding requirement to secure message encryption and decryption hence, facilitating the aforementioned two-step process of secure authentication and client-side encryption using the efficient security algorithm for power constrained IoT as detailed in the Main Document. The algorithm table: 'Device Provisioning Process Flow' and table 1 detail the provisioning sequence and the client-side encryption execution respectively.

0.3 Experimentation Data

Table 5: ENCRYPTION TIMES DATA GENERATED FROM LAPTOP IMPLEMENTATIONS OF STANDARD AES VARIANTS, RR4 AND RR2

Instances	AES-128	AES-192	AES-256	4Rounds	2Rounds
1	0.796481	1.063513	1.150296	0.358913	0.319948
2	0.938084	1.087956	1.231361	0.33023	0.355408
3	0.678996	0.958239	1.091734	0.774768	0.264625
4	0.840003	0.779829	0.862399	0.288513	0.257222
5	0.699007	0.820399	1.179502	0.329047	0.241581
6	0.871251	1.024502	0.926096	0.29894	0.231383
7	0.659962	0.760395	0.91801	0.399697	0.523993
8	0.647439	0.744336	0.898726	0.3847	0.196657
9	0.646284	0.745524	0.84429	0.291152	0.335179
10	0.642803	0.784104	0.847197	0.354563	0.262377

11	0.668974	0.775927	0.95745	0.383063	0.198691
12	0.656269	0.746075	0.827348	0.270903	0.199992
13	0.669492	0.786635	0.803035	0.287626	0.208207
14	0.651663	0.721784	0.788633	0.295411	0.217068
15	0.65963	0.762982	0.782648	0.335264	0.203221
16	0.644828	0.74586	0.808947	0.295549	0.206765
17	0.678491	0.746925	0.816973	0.292222	0.209141
18	0.640799	0.773456	0.814105	0.287358	0.348329
19	0.646623	0.730141	0.794316	0.28049	0.319346
20	0.652565	0.725419	0.793985	0.288851	0.266928
21	0.640252	0.764499	0.836343	0.296679	0.204343
22	0.64393	0.767146	0.814982	0.29164	0.198584
23	0.637668	0.745981	0.793461	0.295319	0.240341
24	0.633993	0.755575	0.795994	0.299369	0.196061
25	0.652754	0.732563	0.78073	0.28942	0.204323
26	0.662431	0.761634	0.842196	0.29031	0.200725
27	0.637823	0.811709	0.815935	0.297496	0.277536
28	0.65987	0.771289	0.863997	0.287387	0.215974
29	0.658937	0.734334	0.826035	0.290604	0.197776
30	0.636545	0.775253	0.791555	0.305222	0.20276
31	0.662086	0.751674	0.78919	0.296161	0.203225
32	0.644834	0.729328	0.791722	0.28568	0.200307
33	0.667148	0.786044	0.790263	0.293869	0.239411
34	0.634388	0.735546	0.801227	0.300489	0.20953
35	0.676704	0.82529	0.812097	0.292282	0.188598
36	0.635997	0.742736	0.801363	0.286407	0.20107
37	0.643702	0.71343	0.824174	0.296671	0.192198
38	0.641621	0.772996	0.801476	0.279898	0.227209
39	0.631685	0.743731	0.797509	0.280169	0.196283
40	0.639805	0.759186	0.801189	0.279618	0.210804
41	0.674406	0.751732	0.79255	0.29026	0.204457
42	0.724704	0.752392	0.784566	0.277302	0.191754
43	0.650104	0.720907	0.829447	0.284235	0.241602
44	0.67115	0.771285	0.797776	0.293654	0.192311
45	0.658444	0.720485	0.810518	0.29496	0.219324
46	0.637829	0.797668	0.805972	0.299274	0.228382
47	0.657	0.7275	0.785955	0.284178	0.190228
48	0.641977	0.861797	0.793709	0.294227	0.263724
49	0.642562	0.770741	0.793679	0.295848	0.240137
50	0.655844	0.752201	0.788024	0.287839	0.192244
51	0.640611	0.740977	0.792546	0.298359	0.205322
52	0.675508	0.849912	0.786215	0.278771	0.257448
53	0.676227	0.734257	0.807021	0.281918	0.199227
54	0.652618	0.763153	0.831031	0.294223	0.308798
55	0.651402	0.750046	0.794944	0.281681	0.217811
56	0.653366	0.786155	0.79249	0.289811	0.197847
57	0.741723	0.760848	0.812319	0.296721	0.206297
58	0.65985	0.775478	0.796919	0.280992	0.201199
59	0.650861	0.758782	0.797387	0.291227	0.20358
60	0.663025	0.772559	0.819652	0.299667	0.208102
61	0.645486	0.843913	0.797077	0.292224	0.221715
62	0.660474	0.755524	0.799229	0.293403	0.246193
63	0.647182	0.732947	0.808645	0.302071	0.21593
64	0.642671	0.741091	0.791025	0.281508	0.188439
65	0.639493	0.82436	0.834648	0.305902	0.210972
66	0.661803	0.737679	0.844368	0.288427	0.26281
67	0.647245	0.746962	0.835154	0.2906	0.203778
68	0.655512	0.724558	0.802613	0.340236	0.194663
69	0.632978	0.743145	0.828356	0.295826	0.187151
70	0.657448	0.756955	0.806119	0.286422	0.223906

71	0.660351	0.775812	0.795449	0.287188	0.346972
72	0.639	0.788759	0.819285	0.281686	0.20823
73	0.644025	0.717566	0.81187	0.276355	0.208011
74	0.686817	0.783176	0.806307	0.291758	0.194373
75	0.686989	0.74995	0.792521	0.294978	0.216359
76	0.669858	0.73884	0.808769	0.32362	0.345135
77	0.660895	0.743109	0.813774	0.289241	0.187716
78	0.991533	0.801572	0.80964	0.287427	0.213549
79	0.646495	0.760508	0.812479	0.315125	0.203112
80	0.658818	0.732363	0.802092	0.293908	0.215188
81	0.659467	0.739964	0.802328	0.288141	0.282489
82	0.655914	0.748594	0.819241	0.288273	0.193076
83	0.641507	0.850415	0.799361	0.287449	0.290985
84	0.653858	0.790232	0.813728	0.299868	0.200441
85	0.649888	0.718799	0.845874	0.284034	0.206625
86	0.66349	0.755336	0.831243	0.295317	0.226451
87	0.672976	0.760736	0.867467	0.282433	0.209834
88	0.645787	0.793376	0.850376	0.287209	0.199801
89	0.654096	0.745713	0.783989	0.290388	0.191049
90	0.650949	0.751276	0.811824	0.295241	0.211907
91	0.668338	0.741746	0.812153	0.285306	0.264901
92	0.653422	0.734143	0.78999	0.29843	0.264115
93	0.652116	0.815095	0.856662	0.308453	0.197625
94	0.648468	0.772482	0.816817	0.322918	0.198779
95	0.649995	0.781928	0.831812	0.290007	0.210326
96	0.652253	0.759551	0.787012	0.293948	0.30342
97	0.649166	0.717744	0.787099	0.299119	0.204918
98	0.673989	0.741115	0.840173	0.28418	0.203088
99	0.677502	0.825463	0.813646	0.294659	0.190189
100	0.649556	0.745381	0.797679	0.300579	0.209313

Table 6: ENCRYPTION TIMES DATA GENERATED FROM PC AND SAMG55 IMPLEMENTATIONS

Instances	PC AES128	SAMG55 AES128	PCRR2	SAMG55RR2
1	0.8653966	7.9845381	0.3199476	5.306837
2	0.6722173	7.0898513	0.3554082	3.558903
3	0.6078102	7.1278026	0.2646247	3.651014
4	0.8142278	7.1740311	0.2572222	3.503775
5	0.7733388	7.1522892	0.2415805	3.447525
6	0.5505474	7.1040291	0.2313827	3.493384
7	0.5367047	7.1572788	0.5239930	3.480428
8	0.9111719	7.0003486	0.1966569	3.657735
9	0.538378	7.1715739	0.33517940	3.560993
10	0.5797766	7.1636076	0.2623766	3.674276
11	0.5616452	7.1852247	0.1986907	3.476701
12	0.5526894	7.1682235	0.1999917	3.578366
13	0.5901672	7.2007712	0.2082066	3.411661
14	0.5529859	7.0344392	0.2170684	3.587528
15	0.525381	7.1181350	0.20322110	3.502104
16	0.5492386	7.0129556	0.2067645	3.447414
17	0.5311931	7.0507333	0.2091410	3.507083
18	0.5814451	6.9322417	0.3483287	3.551365
19	0.5453837	7.0578825	0.3193460	3.600353
20	0.5378147	7.1943727	0.2669280	3.791293
21	0.5528532	7.1874095	0.2043434	3.789522
22	0.5674588	7.0099481	0.1985842	3.566956
23	0.5449442	7.1819301	0.2403406	3.513176

24	0.5545544	7.1191668	0.1960610	3.543044
25	0.5924382	7.0705010	0.20432250	3.4973440
26	0.5724627	7.2794395	0.2007245	3.489710
27	0.5300381	7.3177725	0.2775364	3.558582
28	0.5526312	7.325782	0.21597360	3.597799
29	0.5449595	7.316189	0.1977761	3.474725
30	0.5474292	7.1351984	0.2027602	3.485705
31	0.5702799	7.1486028	0.2032251	3.565918
32	0.5282262	7.1488287	0.2003072	3.646651
33	0.5658316	7.174035	0.2394114	3.534058
34	0.5339959	7.1457394	0.20953	3.500095
35	0.5886807	7.0776997	0.1885983	3.571149
36	0.5565304	7.1565013	0.2010696	3.671927
37	0.5620644	7.2755302	0.1921977	3.494774
38	0.5560042	7.1925013	0.2272087	3.75103
39	0.5326382	7.2063509	0.1962833	3.50964
40	0.6217603	7.1911279	0.2108037	3.448011
41	0.5486803	7.1507099	0.204457	3.512652
42	0.5354819	7.1270511	0.1917535	3.524758
43	0.5610803	7.1868825	0.2416022	3.460276
44	0.544726	7.0024638	0.1923111	3.481107
45	0.6280711	7.1631181	0.219324	3.481105
46	0.5349875	7.0648929	0.2283822	3.482311
47	0.5944224	7.1938023	0.190228	3.54122
48	0.5295156	7.1509642	0.2637239	3.528512
49	0.5640447	7.1958179	0.2401371	3.561153
50	0.5686602	6.876114	0.1922442	3.466134
51	0.5318003	7.0786747	0.205322	3.574118
52	0.6053492	7.1465078	0.2574483	3.583825
53	0.5514164	7.1856221	0.1992267	3.447737
54	0.5522621	7.1944632	0.3087982	3.487202
55	0.5351141	7.1557555	0.2178106	3.450933
56	0.5556107	7.1297972	0.1978465	3.538909
57	0.5365631	6.9756328	0.2062973	3.537691
58	0.5505253	7.1945475	0.2011987	3.517696
59	0.5406325	7.1260282	0.2035803	3.540433
60	0.5376175	7.135184	0.2081023	3.493113
61	0.5655508	7.1600694	0.2217153	3.437365
62	0.5551825	7.1781874	0.2461931	3.585943
63	0.5491718	7.0807266	0.2159295	3.468655
64	0.595563	7.2125069	0.1884392	3.477246
65	0.5490705	7.1857306	0.2109718	3.455186
66	0.5433521	7.1386586	0.2628103	3.452635
67	0.6499065	7.1710358	0.203778	3.457152
68	0.5715895	7.1193347	0.1946626	3.911168
69	0.5486831	7.0932761	0.1871505	3.485562
70	0.5419936	7.0433712	0.2239057	3.483925
71	0.5741845	6.8596342	0.3469724	3.568834
72	0.6248896	7.0255559	0.2082296	3.606569
73	0.5728943	7.1265729	0.2080106	3.616531
74	0.5422509	7.1497671	0.194373	3.447822
75	0.5681231	7.2313748	0.2163594	3.506609
76	0.619145	7.1577117	0.3451346	3.552993
77	0.5585851	7.1688576	0.1877159	3.460477
78	0.6192985	7.1847087	0.2135492	3.467442
79	0.581996	7.1859872	0.2031121	3.46825
80	0.5696902	7.2415987	0.2151881	3.492075
81	0.5627767	7.1967067	0.2824886	3.454579
82	0.5439197	7.148875	0.1930755	3.48244
83	0.5451428	7.1879384	0.290985	3.575745

84	0.6398623	7.2256915	0.2004411	3.526932
85	0.5348342	7.1825497	0.2066249	3.507894
86	0.5525812	7.1058369	0.2264506	3.48867
87	0.5737178	7.1396526	0.2098343	3.481596
88	0.5860344	7.2206878	0.1998014	3.42864
89	0.731048	7.1978417	0.1910487	3.525836
90	0.6128936	7.1376596	0.2119074	3.475132
91	0.5570411	7.1550742	0.2649007	3.477388
92	0.5813946	7.134312	0.2641149	3.509621
93	0.6102484	7.1769933	0.1976251	3.430877
94	0.557563	7.0415045	0.1987785	3.447758
95	0.6324176	7.1500664	0.210326	3.398484
96	0.6989839	7.1576659	0.30342	3.392671
97	0.5737455	7.1824543	0.2049178	3.446055
98	0.5710415	7.0841743	0.2030884	3.475504
99	0.5861858	7.2191732	0.1901892	3.535517
100	0.5410907	7.204138	0.2093132	3.424857

Table 2: RR2 AVALANCHE EFFECT LOG FOR THE FIVE BYTE FLIP INSTANCES

Byte flip	Message	Ciphertext
1	[87, 120, 97, 116, 32, 105, 115, 32, 97, 32, 99, 114, 101, 101, 108, 63]	[138, 243, 252, 60, 61, 75, 136, 57, 161, 142, 125, 88, 107, 94, 31, 120]
2	[87, 104, 97, 116, 32, 105, 115, 32, 97, 32, 99, 114, 101, 101, 120, 63]	[138, 15, 252, 95, 115, 75, 45, 57, 161, 83, 125, 75, 154, 94, 153, 120]
3	[87, 104, 97, 120, 32, 105, 115, 32, 97, 32, 99, 114, 101, 101, 108, 63]	[120138, 108, 252, 95, 28, 75, 45, 57, 161, 83, 125, 85, 154, 94, 14, 120]
4	[87, 104, 97, 116, 32, 105, 115, 32, 97, 32, 120, 114, 101, 101, 108, 63]	[254, 243, 252, 95, 61, 75, 45, 193, 161, 83, 200, 88, 154, 26, 31, 120]
5	[87, 104, 97, 116, 32, 105, 115, 32, 97, 32, 99, 114, 120, 101, 108, 63]	[138, 243, 252, 2, 61, 75, 233, 57, 161, 204, 125, 88, 221, 94, 31, 120]

Table 3: RR3 AVALANCHE EFFECT LOG FOR THE FIVE BYTE FLIP INSTANCES

Byte flip	Message	Ciphertext
1	[87, 120, 97, 116, 32, 105, 115, 32, 97, 32, 99, 114, 101, 101, 108, 63]	[79, 63, 93, 127, 26, 253, 55, 48, 163, 29, 153, 10, 70, 194, 85, 170]
2	[87, 104, 97, 116, 32, 105, 115, 32, 97, 32, 99, 114, 101, 101, 120, 63]	[239, 249, 61, 174, 22, 74, 78, 132, 104, 215, 179, 185, 163, 112, 246, 135]
3	[87, 104, 97, 120, 32, 105, 115, 32, 97, 32, 99, 114, 101, 101, 108, 63]	[81, 139, 48, 212, 99, 101, 87, 90, 93, 66, 132, 69, 118, 59, 160, 78]
4	[87, 104, 97, 116, 32, 105, 115, 32, 97, 32, 120, 114, 101, 101, 108, 63]	[46, 130, 149, 228, 28, 203, 159, 26, 252, 79, 28, 97, 42, 189, 38, 97]
5	[87, 104, 97, 116, 32, 105, 115, 32, 97, 32, 99, 114, 120, 101, 108, 63]	[237, 154, 223, 145, 162, 240, 61, 201, 130, 93, 211, 210, 8, 109, 9, 178]

Table 4: RR4 AVALANCHE EFFECT LOG FOR THE FIVE BYTE FLIP INSTANCES

Byte flip	Message	Ciphertext
1	[87, 120, 97, 116, 32, 105, 115, 32, 97, 32, 99, 114, 101, 101, 108, 63]	[219, 111, 183, 151, 108, 67, 54, 0, 146, 174, 27, 187, 120, 11, 96, 1]
2	[87, 104, 97, 116, 32, 105, 115, 32, 97, 32, 99, 114, 101, 101, 120, 63]	[27, 190, 233, 46, 102, 113, 27, 196, 217, 148, 33, 61, 27, 132, 76, 228]
3	[87, 104, 97, 120, 32, 105, 115, 32, 97, 32, 99, 114, 101, 101, 108, 63]	[186, 45, 240, 227, 6, 249, 217, 248, 40, 237, 101, 103, 146, 33, 198, 157]
4	[87, 104, 97, 116, 32, 105, 115, 32, 97, 32, 120, 114, 101, 101, 108, 63]	[150, 116, 145, 14, 239, 30, 192, 50, 243, 47, 167, 253, 185, 115, 60, 40]
5	[87, 104, 97, 116, 32, 105, 115, 32, 97, 32, 99, 114, 120, 101, 108, 63]	[254, 171, 72, 30, 27, 247, 181, 235, 160, 163, 205, 215, 1, 94, 21, 117]

Bibliography

[1] J. N. Mamvong, G. L. Goteng, B. Zhou, and Y. Gao, "Efficient security algorithm for power-constrained iot devices," *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5498–5509, 2021.

[2] M. Katagi, S. Moriai *et al.*, "Lightweight cryptography for the internet of things," *Sony Corporation*, vol. 2008, pp. 7–10, 2008.

[3] M. Jangra and B. Singh, "Performance analysis of clefia and present lightweight block ciphers," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 22, no. 8, pp. 1489–1499, 2019.

[4] M. A. Kamal, H. W. Raza, M. M. Alam, and M. M. Su'ud, "Highlight the features of aws, gcp and microsoft azure that have an impact when choosing a cloud service provider," *International Journal of Recent Technology and Engineering (IJRTE)*, 2020.

[5] P. Pierleoni, R. Concetti, A. Belli, and L. Palma, "Amazon, google and microsoft solutions for iot: architectures and a performance comparison," *IEEE Access*, vol. 8, pp. 5455–5470, 2019.

[6] P. Dutta and P. Dutta, "Comparative study of cloud services offered by amazon, microsoft & google," *International Journal of Trend in Scientific Research and Development*, vol. 3, no. 3, pp. 981–985, 2019.

[7] S. B. Sadkhan and A. O. Salman, "A survey on lightweight-cryptography status and future challenges," in *2018 International Conference on Advance of Sustainable Engineering and its Application (ICASEA)*, 2018, pp. 105–108.

[8] h. y. microchip.com, title=Network and Accessories secure authentication.