Privacy on IoT's Midware level 727 Report

Ariel Zhuang, rzhu035

I. Abstract

The social acceptance of IoT applications and services is strongly depending on the trustworthiness of information and the protection of private data. Since the IoT is a complex, distributed and heterogeneous system in nature, it faces several challenges regarding security and privacy; a reliable security technique for the IoT is still in demand to satisfy requirements of data confidentiality, integrity, privacy and trust [Atlam and Wills(2020)]. Numerous papers have identified the different levels and categories of sections within an IoT model, but there has been no breakdown of the classification of privacy. Though it has been touched upon in many papers, such as 'the first step toward defining the privacy is to define the classification of sensitive information at the context of any IoT device [Zolanvari and Jain(2015)]'. This report aims to establish a succinct privacy model and determine appropriate cryptographic measures to strengthen it within lightweight environments. Finally, over 500,000 pieces of accurate IoT data will be utilised to verify the hypothesis of the IoT privacy model.

II. Methodology

Firstly, a concise literature review is conducted after using 'IoT security' and 'IoT security guidance' to search on Google Scholar; several papers became the primary reference :

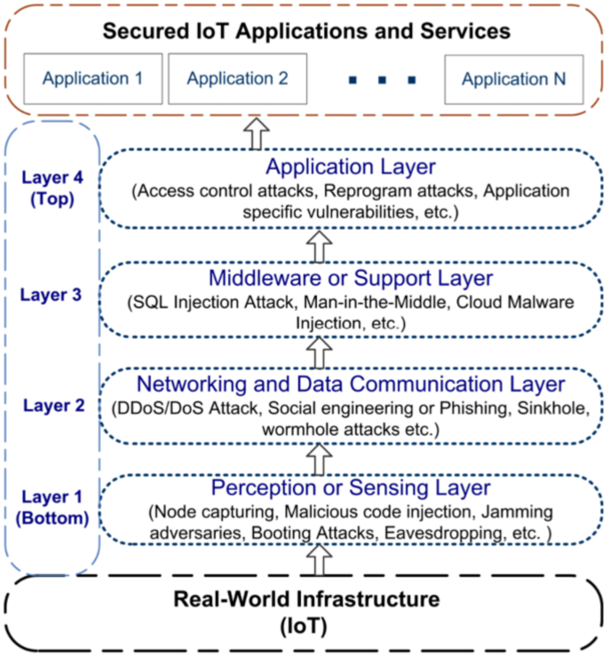
1) Internet of Things (IoT) Security Intelligence: A Comprehensive Overview, Machine Learning Solutions and Research Directions[Sarker et al.(2022)]
2) Internet of Things (IoT) Security: Current Status, Challenges and Countermeasures[Yousuf et al.(2015)]
3) IoT Security, Privacy, Safety and Ethics[Atlam and Wills(2020)].
4) Introduction to IoT Security[Jurcut et al.(2020)]
5) Landscape of IoT security[Schiller et al.(2022)]
6) IoT Security: Ongoing Challenges and Research Opportunities[Zhang et al.(2014)]
7) IoT Security: A Survey[Zolanvari and Jain(2015)]

In addition, we conducted a case study on a commercial IoT application to explore the best cryptographic measures for different privacy levels and data types.

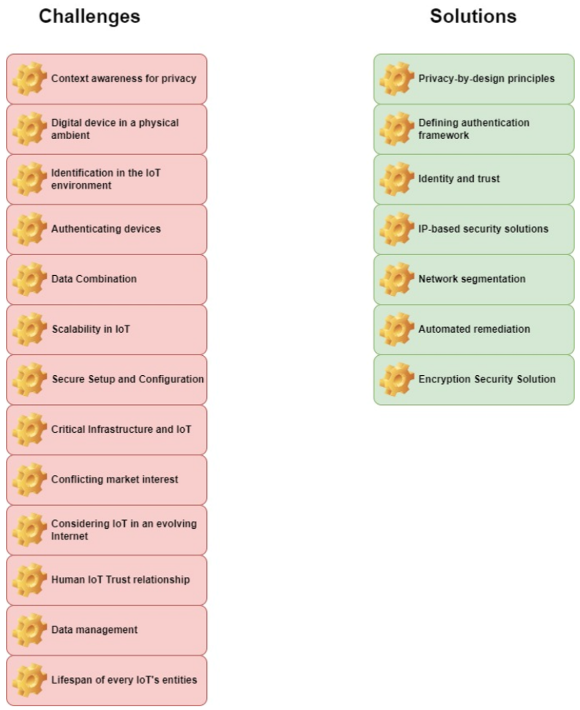
III. Result

A. IoT's layer model

As long as a person who learned network technology and cybersecurity will comprehend diverse network attacks aimed at one or several dedicated layers[Swire(2018)]. IoT is almost the same. However, compared with the OSI model of the network, or the most practical TCP/IP model, there is no standard for IoT's layers. However, talking about a recognised model can make identifying threats and developing countermeasures easier. According to many researchers [Zhao and Ge(2013)][Atzori et al.(2012)][Leo et al.(2014)], the IoT primarily operates on three layers which are the Perception, Network, and the Application layer. Each layer of IoT has inherent security issues associated with it[Yousuf et al.(2015)]. Meanwhile, some researchers imitated the OSI model and designed it as seven layers so that each layer provides additional information for establishing a common terminology[Atlam et al.(2018)]. However, the support or middleware layer is considered an important layer later, according to the needs for data processing and intelligent decision making, which lies between the network layer and the application layer[Sarker et al.(2022)], reflecting the cutting-edge software development and IoT environment precisely. See the below figure demonstrating the four-layer IoT model[Sarker et al.(2022)].


Secured IoT Applications and Services
Application 1
Application 2
Application N
Layer 4 (Top)
Application Layer
(Access control attacks, Reprogram attacks, Application specific vulnerabilities, etc.)
Layer 3
Middleware or Support Layer
(SQL Injection Attack, Man-in-the-Middle, Cloud Malware Injection, etc.)
Layer 2
Networking and Data Communication Layer
(DDoS/DoS Attack, Social engineering or Phishing, Sinkhole, wormhole attacks etc.)
Layer 1 (Bottom)
Perception or Sensing Layer
(Node capturing, Malicious code injection, Jamming adversaries, Booting Attacks, Eavesdropping, etc.)
Real-World Infrastructure (IoT)

The perception layer, network layer, middleware layer and application layer confront various threats in accordance with their attributes. Typical CIA (Confidentiality, Integrity and Availability) security requirements should be employed in the IoT system[Sarker et al.(2022)]. However, for the IoT system, confidentiality needs not only to be guaranteed inside the communication network but also when transmitting messages between various IoT devices[Sarker et al.(2022)]. The below figure exhibits the threats and traditional solutions[Zolanvari and Jain(2015)].


Challenges
Context awareness for privacy
Digital device in a physical ambient
Identification in the IoT environment
Authenticating devices
Data Combination
Scalability in IoT
Secure Setup and Configuration
Critical Infrastructure and IoT
Conflicting market interest
Considering IoT in an evolving Internet
Human IoT Trust relationship
Data management
Lifespan of every IoT's entities
Solutions
Privacy-by-design principles
Defining authentication framework
Identity and trust
IP-based security solutions
Network segmentation
Automated remediation
Encryption Security Solution

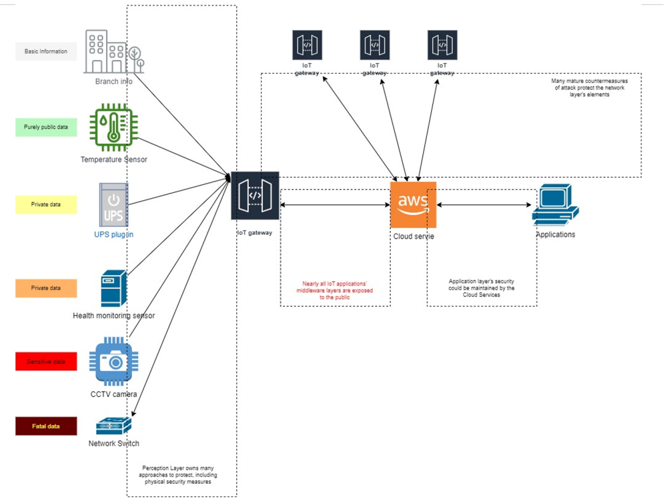
Suppose we apply those concepts to a typical IoT setting, which is suitable for over 90% of IoT applications. In that case, we can easily see that the middleware section - specifically the IoT gateway to the cloud service (whether it's Pass or Sass) - is the most vulnerable area lacking sufficient protection. Although the middleware layer is essential for delivering a secure and dependable IoT application, it is also vulnerable to attacks such as insider attacks, man-in-the-middle attacks, SQL injection attacks, signature wrapping attacks, cloud malware injection, cloud flooding attacks, and so on[Hassija et al.(2019)][Kügler(2003)]. Before the IoT, a security breach can lead to losing your money, but with IoT, security attack can literally result in losing your life[Atlam and Wills(2020)]. However, there are too many threats and countermeasures in the IoT world, we are not able to analyse them in this report. If we were transported back to the era of the Caser cypher and could ensure that a potential eavesdropper would not be able to understand the information they intercepted, any breached information would not significantly harm the IoT. Maintaining privacy can be viewed as the foundation of an IoT environment and a simple philosophy. This approach effectively addresses the vulnerability of the middleware layer, which is highly exposed to the public. Privacy is a notion that is associated with four main elements: information, communications, body and territory[Atlam and Wills(2020)]. The IoT's anywhere, anything, anytime nature could easily change these advantages into disadvantages, if privacy aspects would not be provided enough [Zhang et al.(2014)]. As privacy is the main concern of IoT to be ensured through impregnable access control schemes, the GDPR initiative is a timely solution established to constrict the IoT service providers (both software and hardware) from developed and marketing products with vulnerabilities[Jurcut et al.(2020)]. Notwithstanding, the first step toward defining privacy is to define the classification of sensitive information in the context of any IoT device [Zolanvari and Jain(2015)]. This research aims to classify privacy and explore practical cryptographic methods.

B. Privacy Mode

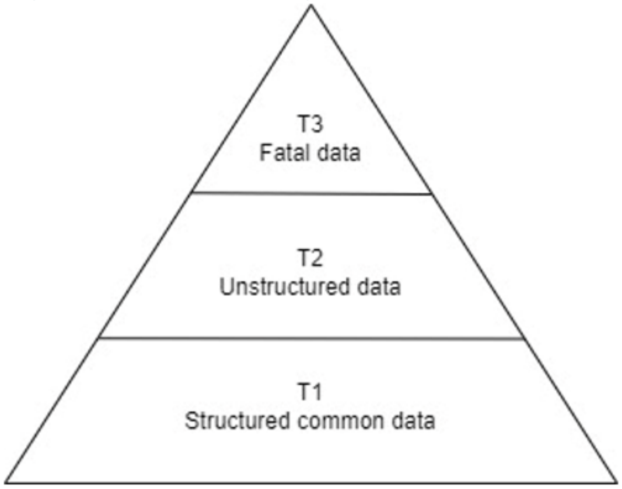
1) Six categories of IoT data: Six categories of IoT data exist based on their sensitivity levels and potential outcomes if they are compromised. Admittedly, diverse IoT systems which aim at various applications might have distinct forms and sources of data. However, the below categories stemming from the studied commercial IoT platform reflect a regular circumstance. (Below categorisation and consequence analysis depend on the vast majority of the scenarios, which means the categories and consequences might not suit exceptional scenes because IoT enables to improve several applications in various fields, such as, smart cities, smart homes, healthcare, smarts grids, as well as other industrial applications[Jurcut et al.(2020)].)

- A. Purely public data, everyone can see or obtain them readily via an ordinary approach; meanwhile, even if they were breached, no practically harmful consequence would be induced, such as the environmental temperature and humidity.
- B. Private data should not be obtained by other people who are not in the organisation; however, even if a MITM intercepted them, he/she could practically do nothing to impair the entity, such as the HDDs' capacity.
- C. Private data should not be obtained by other people who are not in the organisation; however, if a MITM intercepted them, he/she could not directly harm the entity by those data; nevertheless, the exposed 'opportunities' being leaked from the flaws might cause either physical or cyber security issues or both, such as servers' health status, UPS's status, CCTV system's health status.
- D. Sensitive data, which are critical information running upon or being generated by the IoT platform. If it were forged, that would lead to significant issues. To some extent, it might touch upon crime if an attacker falsified them, such as the CCTV live-stream or footage.
- E. Fatal data (control data), which owns an extremely high-security level. If a hacker hijacked or forged it, he/she even could dominate the whole system. Before the IoT, a security breach can lead to losing your money, but with IoT, security attack can literally result in losing your life[Atlam and Wills(2020)]. This category belongs to the 'might-lose-your-life' level, admittedly. This kind of data should be sophisticatedly maintained. The cost of maintenance could be neglected, such as remotely acknowledge on an alarm panel, remote switch operation and so forth.
- F. Basic information(Device identity spoofing), it is the basic information of an IoT environment, such as the sensor ID or the branch ID in an IoT platform. It seems

that breaching them means nothing. The reality is forging data causes original data's loss and fabricating data leads to fake information, which could be seen as a kind of high -level DoS attack.


Basic Information
Branch info
Purely public data
Temperature Sensor
Private data
UPS plug-in
Private data
Health monitoring sensor
Sensitive data
CCTV camera
Fatal data
Network Switch
Perception Layer owns many approaches to protect, including physical security measures
IoT gateway
IoT gateway
IoT gateway
IoT gateway
Many mature countermeasures of attack protect the network layer's elements
aws
Cloud servie
Applications
Nearly all IoT applications' middleware layers are exposed to the public
Application layer's security could be maintained by the Cloud Services

2) Three-layer IoT privacy model : To ensure adequate privacy protection, it would be better to have at least six cryptography approaches that correspond to six or more data categories. These approaches allow for targeted measures to be taken in different privacy scenarios. However, it should be considered that any solution for this aspect should be lightweight enough that it can be implemented on IoT[Zolanvari and Jain(2015)]. As a result, We recommend condensing the IoT messages' privacy as three levels after taking privacy requirements, data frequency and data types into account. See the below chart.



1) T1: Structured data: All structured data except for the control data belongs to this level, which is transmitted, processed, stored with an extremely high frequency, a high efficiency and lightweight encryption is able to address the privacy issue.
2) T2: Unstructured data: such as audio video information, which is widely used in the cutting-edge IoT system, could not be encrypted in real-time due to the time-sensitive issue and limited resources of IoT devices; signature approaches were able to address the privacy issue.
3) T3: The critical control data(instructions): The data is highly sensitive., which can literally result in losing your life[Atlam and Wills(2020)]., should be protected severely, neglecting the cost of data encryption.

Three-level privacy model of IoT depicts the IoT's privacy classification concisely. If we were able to make it encrypted effectively, IoT's privacy issues, or even all security issues of IoT could be solved immediately since encryption of information is a solution to protect the network from attack, which is widely used and popular[Zolanvari and Jain(2015)]. If that were true, having RSA2048 would be the ultimate solution. Even if a MITM had a quantum computer, it would take 8 hours to crack it[Jurvetson(2019)]. Unfortunately, that is not the case, cryptographic mechanisms and protocols, usually are the best choices to protect data, but sometimes we may not be able to implement these techniques on small elements. Therefore, we should have policies regarding how to manage any type of data with various policy mechanisms[Zolanvari and Jain(2015)]. The most common algorithms used for encrypting are: RSA, ECC, AES, 3DES, MD5 and SHA, which are heavily computational [Ukil et al.(2011)]. Such resource limitations prevent traditional security measures on IoT devices because due to resource-constrained devices are particularly susceptible to exhaustive attacks [Garcia-Morchon et al.(2019)]. The following part of the report, a commercial IoT application's data is utilised to verify the above privacy level, and various cryptography approaches are experimented.

IV. A CASE STUDY

A. Background

The following section's analysis and research depend on a modelised commercial IoT system and its data. (The IoT platform's name and specific information have been concealed or altered to shield the owner's identity for commercial and legal reasons. In the following words, we name it Platform-S). Platform-S enables multi-site businesses or chain stores to delegate the monitoring and management of their site devices and systems to a trusted 3rd-party company. The 3rd-party company's continental-based monitoring centre remotely supervises and manages the subscribers' every site/branch via a huge IoT network which is composed of a Microsoft Azure-based core platform and ARM-architecture-based IoT gateways that are installed at each subscriber's site. Meanwhile, various soft and hard sensors manage servers, UPSs, power distribution units, network switches, CCTV cameras, access control systems, alarm panels, and environmental conditions at each site. Platform-S collects information on subscribers' devices and systems(upstream data) and can also carry out control instructions based on specific rules(downstream data). It has been promoted and evolved for almost a decade and deployed among numerous famed and Fortune 500 companies. This report's research relies on 500000 pieces of structured data (intentionally altered but with an identical structure) and the exact size of unstructured data. The primary hardware platforms used for the study are Wintel, Raspberry Pi 4B, and Raspberry Pi Zero ver. 1, which are similar to the operational

IoT gateways. Azure is not involved in the research, so communication is done through a free Internet-base MQTT platform broker.emqx.io.

B. *Experiment Plan*

1) Overview:

1) To diversify video bandwidth feeds at different intervals, use two algorithms to sign footages of sizes 15/30/45/60/90/120/135/180/240MB. Research the availability of these algorithms by analysing their time consumption on various platforms.
2) To add a signature to a live video stream's frames, utilise the up to 40 bytes of the IPv4 package header. Assess the capacity limits of platforms running under 25fps and establish a suitable number of frames that can be signed.
3) To evaluate the performance of an algorithm on a specific platform, use MQTT to transmit 500,000 data pieces as a benchmark. Send encrypted information using MQTT publisher and decrypt it at the MQTT subscriber to assess time consumptions.
4) To ensure security, we utilise an advanced encryption algorithm for controlling instructions and evaluate the impact on consumption of time and size capacity.

2) Experiment Environment: The below experiments depend on the specific environment: 1. Hardware OS:

- a. Intel X86 with WinOS: Few IoT gateways use this environment because of its high cost, despite the fact that it provides comprehensive security measures and commercial maintenance. This cost may discourage users, especially since IoT systems typically use a large number of devices. Nevertheless, it could serve as an excellent benchmark for experiments.
- b. Raspberry Pi 4B with Linux: Many people use the RP4B or similar platforms like Nvidia Jetson as the physical layer devices for IoT. These devices have impressive computational capacity and are cost-effective, with a price range of around NZD 100 before Covid-19 and NZD 350 after Covid-19 (RP4B), making them suitable for IoT applications[Kurniawan and Kurniawan(2019)].
- c. Raspberry Pi Zero (ver.1) with Linux: Certain IoT applications that rely solely on structured data use devices like RP0 or Arduino as a gateway. While these devices have limited computational capacity, their low price point (around 25NZD before Covid-19 and approximately 100NZD after Covid-19 for version 2) makes them an appealing option for users[Kurniawan and Kurniawan(2019)].

2. Development language:

- a. Python 3.8: This report takes a cautious approach by using a conservative evaluation language Python, despite the fact that many IoT developers prefer C and C++ because they are faster than Python[Plauska et al.(2022)]. However, a growing number of devices and microcontrollers now support Python (or MicroPython) in addition to traditional C/C++, making Python a reasonable choice for evaluation purposes [Hillar(2016)][Plauska et al.(2022)].

3. Transmission Protocol:

- a. MQTT: In IoT, device to device communications are considered through either Pushing or Polling protocol. Push protocol is more suitable for IoT devices because of its lightweight and high productivity. There are many Push protocols available for IoT such as XMPP, MQTT, AMQP in which MQTT is most widely used. The key feature of MQTT is its lightweight and bandwidth efficiency[Soni and Makwana(2017)].

4. Others:

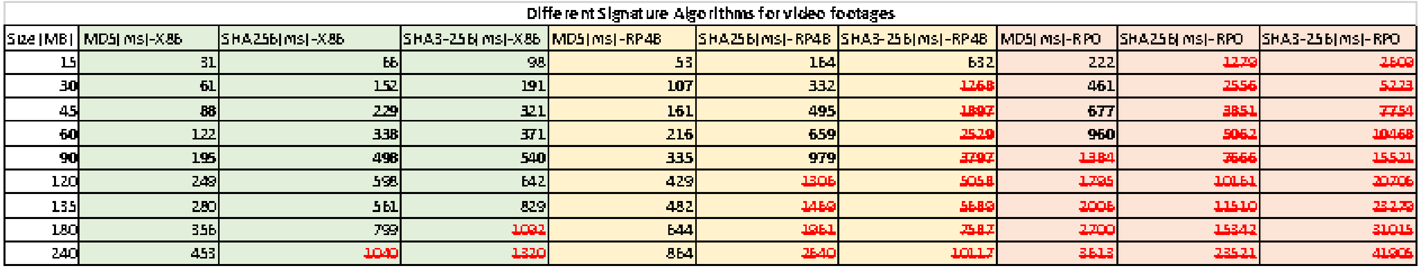
- a. Due to the restricted time and many other causes, numerous elements in an IoT environment, even those that exist in the Platform-S, are not able to be involved in the report's experiments, such as the Azure service. Meanwhile, it is only possible to experiment with several encryption and signature algorithms.

C. *Experiment 1-T2*

1) Overview: T2 refers to unstructured data or messages. The primary components of T2 are audio and video information, which can be live feeds or recorded footage. Compared with structured data, they spend much more capacity on transmission, storage, memory and even CPU resources. Live video feeds typically range from several seconds to several minutes in duration, which could be seen as a time-lapsed video record to some extent. Encrypting video or audio data can cause unnecessary problems and waste resources for both the sender and receiver. This is because modern computer architectures support GPU decoding[Shen et al.(2005)], meaning that encrypted videos require more CPU power not just for decryption but also for decoding. Therefore, signature technology is suitable for them, and hash technology owns a high priority. Cryptographic hash functions have another property that it is very difficult to find two different messages that produce the same message digest. To provide data integrity and data authentication, if a message digest of any information is changes, then the file itself has changed [Pittalia(2019)]. We experiment with MD5, SHA256 and SHA3-256, the three most prevailing signature algorithms in this report.

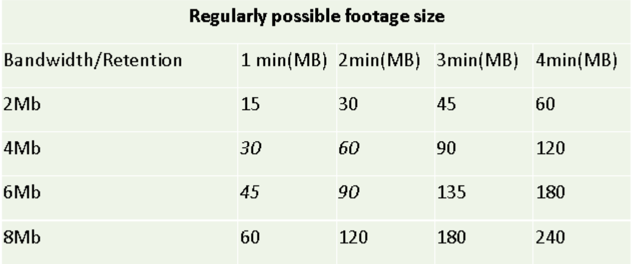
2) Experiment 1A-Video Footage:

a) Plan: CCTV footage is commonly used as evidence or for other purposes in IoT applications. However, it differs from traditional CCTV systems due to its lightweight feature. Usually, CCTV footage is stored for a short period of time, corresponding to the built-in storage of the CCTV camera. The table below shows footage sizes under different retention periods and compression ratios, with the most commonly used ratios in italics[Symes(2004)]. Concerning the storage form (standalone file on the IoT gateway) and decoding method, a separate signature file with the same filename but different postfix is the most reasonable approach.



Different Signature Algorithms for video footages

Size (MB)	MD5(ms)-X86	SHA256(ms)-X86	SHA3-256(ms)-X86	MD5(ms)-RP4B	SHA256(ms)-RP4B	SHA3-256(ms)-RP4B	MD5(ms)-RPO	SHA256(ms)-RPO	SHA3-256(ms)-RPO
15	31	66	98	53	164	632	222	1279	2609
30	61	152	191	107	332	1268	461	2556	5223
45	88	229	321	161	495	1807	677	3851	7754
60	122	338	371	216	659	2529	960	5062	10468
90	195	498	540	335	979	3797	1384	7666	15521
120	249	598	642	429	1306	5058	1795	10161	20706
135	280	561	829	482	1469	5689	2006	11510	23279
180	356	799	1092	644	1961	7587	2700	15342	31015
240	453	1040	1320	864	2640	10117	3613	23521	41906



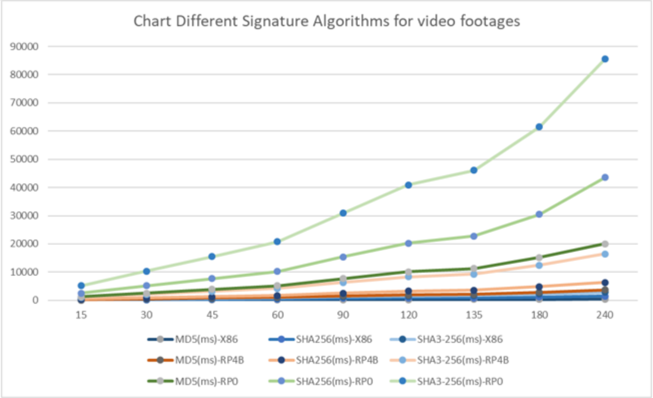
Regularly possible footage size

Bandwidth/Retention	1 min(MB)	2min(MB)	3min(MB)	4min(MB)
2Mb	15	30	45	60
4Mb	*30*	*60*	90	120
6Mb	*45*	*90*	135	180
8Mb	60	120	180	240

How to identify the outcome:

1) Most IoT systems and traditional CCTV systems typically retain data as only one or two minutes. Therefore, it is important that the signature's time consumption does not exceed one minute, and the faster it is, the better.
2) As the file size increases, it is best for the algorithm's time consumption to remain constant or increase linearly. An exponential increase is unacceptable.
3) For the footage, the signature's size is not sensitive.

Above and below is the experiment data



b) Analysis:

1) Intel Platform, either signature algorithm illustrates excellent performance, even SHA3-256 for 8Mb bandwidth with 4 minutes local retention, which is rarely used in an IoT environment, consumes 1.3 seconds, and the frequently utilised 30/45/60/90MB consume the maximum as 0.5s (under SHA3), which is thoroughly acceptable. However, no one would use i7 as an IoT gateway. Hence, it is merely a benchmark.
2) Raspberry Pi 4B, widely used as an IoT gateway, demonstrates a satisfactory performance while using MD5 and SHA256. MD5 suits either size, and SHA256 suits the ones less than 90MB (2-minute retention and 6Mbps bandwidth, a very common scenario for Full HD CCTV). However, SHA3 cannot be accepted due to the prolonged time consumption.
3) Raspberry Pi Zero (version 1), as a low-cost solution for IoT gateway, merely the combination of less-than-60MB with MD5 could be tolerated constrainedly, which means this solution is inappropriate for unstructured data's signature. (This conclusion matches the practical environment)

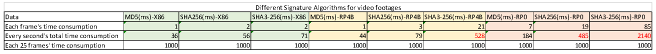
c) Conclusion: For Raspberry 4B or similar-level IoT gateway applications, MD5 and SHA256 are both options. MD5 is suitable for environments with complex multi-task applications, while SHA256 is better suited for gateways with less demanding workloads. Even the simplest MD5 cannot handle the unstructured data signature in gateway applicationslike Raspberry Pi Zero or others at a similar level

3) Experiment 1B-Live Video:

a) Plan: Live video differs from recorded video footage because it is a consecutive stream of frames. In terms of GPU decoding requirements, the signature for live video cannot be a separate file but instead must be a string up to 40 bytes long that is included in the optional section of IPv4's header[Barik et al.(2016)]. This is a commonly used solution. As a result, this solution does not impact the standard decoding but completes the signature without significant extra bandwidth consumption. How to identify the outcome:

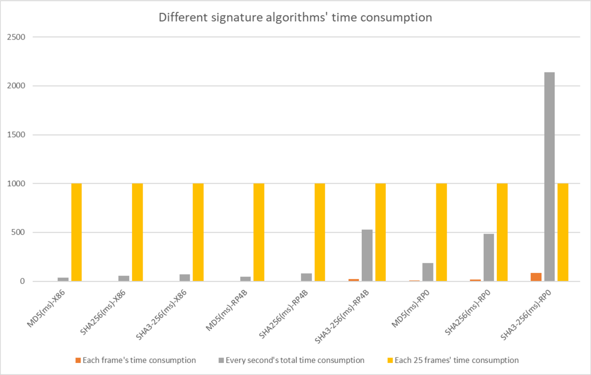
- Most CCTV cameras today use either H.264 or H.265 compression with GOP intervals of 25 or 100, meaning that there is only one I frame per second or four seconds. The remaining frames are P frames. Some older MPEG4 cameras use a GOP interval of 12. The least efficient solution is GOP=1, which uses MJPEG compression[Symes(2004)][Barik et al.(2016)]. Generally speaking, an I frame is 200KB to 1000KB, depending on the specific environment; 512 KB is a typical value for a complicated scene; and P frame is several KB [Symes(2004)][Barik et al.(2016)]. The experiment aims at an ultimate scenario that all 25 frames are all I frames, and at least one frame of 25 should be signed, which keeps the 25 frames could be handled by the specific hardware in one second.

b) Experiment data:



Different Signature Algorithms for video footages

Data	MD5(ms)-X86	SHA256(ms)-X86	SHA3-256(ms)-X86	MD5(ms)-RP4B	SHA256(ms)-RP4B	SHA3-256(ms)-RP4B	MD5(ms)-RP0	SHA256(ms)-RP0	SHA3-256(ms)-RP0
Each frame's time consumption	1	2	2	1	3	21	7	19	85
Every second's total time consumption	36	56	71	44	79	528	184	485	2140
Each 25 frames' time consumption	1000	1000	1000	1000	1000	1000	1000	1000	1000



c) Analysis: There are various ways to sign a live video stream compared to the signature on the video footage, which can result in different outcomes. To simplify our analysis, we will only examine the best-case scenario where one frame is signed per second and the worst-case scenario where each frame is signed within a second. We obtained the average time for signing a frame by examining the consumption of 5000 frames.

1) The best case, simply sign the I frame within a second. The most time-consuming process is SHA3-256 on RP0, taking approximately 85ms, which is still acceptable. Additionally, all other algorithms perform well on various platforms, taking less than 21ms. Overall, these results demonstrate excellent performance.
2) The worse case, in terms of performance, the X86 platform is the best option for running all three algorithms. However, on the RP4B, SHA3-256 takes around 0.5 seconds to sign one second's frames, while SHA256 takes the same amount of time in RP0; SHA3-256 takes more than two seconds to sign one second's frames on RP0, making it impractical to use any of these three combinations in reality. And RP0 spends 184 ms on MD5 also looks too long. In comparison, the others are in an acceptable scope.

d) Conclusion:

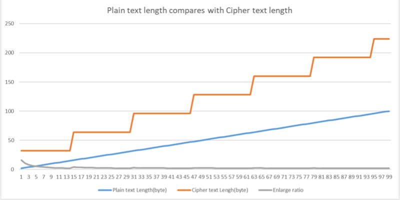
1) RP0 produces satisfactory results when using all three algorithms at a 1:25 signature ratio, and acceptable results with MD5 at a 1:1 signature ratio. However, RP0 is not a suitable choice for processing live video feeds on an IoT platform, especially when cryptographic technology is implemented.
2) Although X86(i5) demonstrates the best performance while running diverse signature algorithms, it is not a good choice of IoT's gateway.
3) The RP4B performs well in most situations, but it struggles when using SHA3-256 for a 1:1 signature scenario. This suggests that SHA3-256 may not be the best choice for live feeds. Based on the conclusion of the footage, it is recommended to use MD5 and SHA256 as the signature options for live feeds.

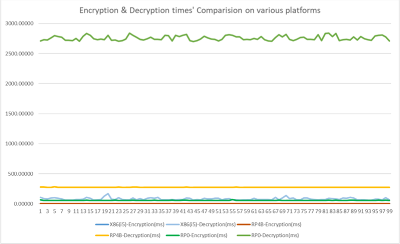
4) Experiment 2-T3:

a) Plan: T3 data in an IoT environment refers to critical control data for important devices. For instance, in Platform-S, restarting a network switch in a branch is considered T3-level information. If a malicious third party were to fabricate a fake command such as a restart or shutdown for a network switch, it could potentially disrupt the security system in the branch for several minutes or even disable it completely. When compared to the data of T1 and T2 levels, the data of T3 is usually just 0.01% to 1% or even less. It's worth noting that while more and more automation systems are adopting the concept of IoT, it's important to remember that IoT was not originally designed for device control. As a result, protecting T3-level data is able to neglect the CPU, time and capacity (the longer length of a command is encrypted) consumption. Implementing a solution disregarding cost might be easy, and the answers are lots; hence, the comparison seems meaningless. The experiment employs a mixed encryption that AES+RSA(2048), which comes from the idea of a ransomware Pandora, which utilises a mixed algorithm with AES and RSA, was shown on February 2022 and attacked the networks of a major automotive parts supplier, Denso Corp., a Japanese-based company[Avertium([n. d.])]. How to identify the outcome:

- Compare the encryption, decryption time and the length of the encrypted command in parallel.

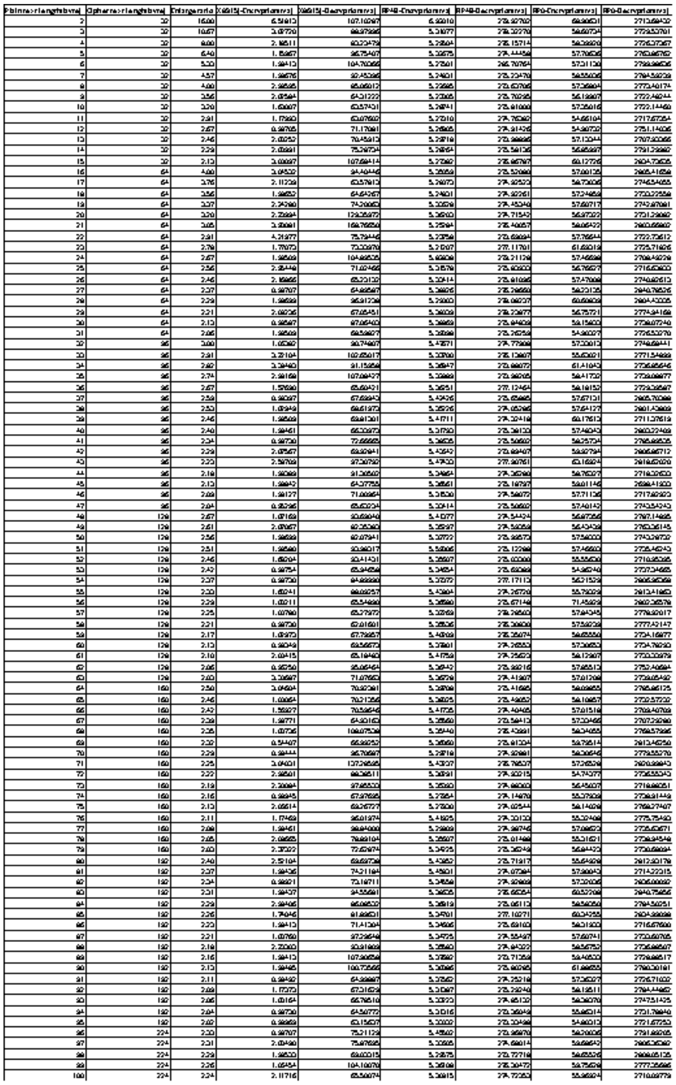
b) Experiment data: Below is the Experiment data





c) Analysis:

1) Cipher text's length, the above experiment examed the outcomes of encryption 2 bytes to 100 bytes' commands. Except for the commands less than 10 bytes, all the others' length are merely around double. However, from 1 byte to 15 bytes, the encrypted command's length is only 32 bytes, which can be accepted thoroughly.
2) Encryption time's consumption, In addition to having excellent performance on X86 (i5), the RP4B takes less than 6 milliseconds to process any command that is less than 100 bytes in length. The RP0, on the other hand, takes approximately 60 milliseconds. This means that all three platforms are suitable for RSA (2048) + AES256 combination encryption.
3) Decryption time's consumption, It takes much longer time than encryption. Besides X86(i5), RP4B demonstrates an utterly acceptable outcome of less than 280



Plaintext-length(bytes)	Ciphertext-length(bytes)	Enlarge ratio	X86I5(-Encryption(ms))	X86I5(-Decryption(ms))	RP4B-Encryption(ms)	RP4B-Decryption(ms)	RP0-Encryption(ms)	RP0-Decryption(ms)
2	32	16.00	6.51613	107.10987	6.92010	278.92702	69.90651	2713.69432
3	32	10.67	3.09730	88.97986	5.31077	278.32270	58.60734	2772.53701
4	32	8.00	2.16511	60.23479	5.29604	276.15714	58.29920	2776.37367
5	32	6.40	1.15967	96.75407	5.32675	274.44458	57.70636	2762.86762
6	32	5.33	1.99413	104.70066	5.27501	266.70764	57.31130	2798.98636
7	32	4.57	1.98676	92.45366	5.24501	275.23470	58.55406	2784.59239
8	32	4.00	2.98585	65.06013	5.22985	273.63706	57.36904	2773.40174
9	32	3.56	2.05584	64.31222	5.22065	275.70295	56.19987	2772.48044
10	32	3.20	1.60007	63.57431	5.29741	275.81000	57.35016	2772.14460
11	32	2.91	1.17960	63.07600	5.27310	274.76360	54.66104	2717.67354
12	32	2.67	0.99705	71.17081	5.26905	274.91426	54.90732	2751.14006
13	32	2.46	2.00252	70.45813	5.29718	273.98886	57.13044	2707.90066
14	32	2.29	2.00881	75.29734	5.29954	275.58136	56.85897	2791.29880
15	32	2.13	3.00097	107.69414	5.27382	276.96787	60.12736	2804.73635
16	64	4.00	3.04532	94.40446	5.35059	275.52980	57.00135	2805.41658
17	64	3.76	2.11238	63.57813	5.26073	274.92523	58.73006	2746.54055
18	64	3.56	1.99652	64.64367	5.24501	274.92261	57.24858	2730.22558
19	64	3.37	2.24380	74.20065	5.30559	274.45340	57.60717	2742.87081
20	64	3.20	2.22884	129.26973	5.36910	274.71542	56.97323	2731.29880
21	64	3.05	3.20081	169.76650	5.25594	276.40057	58.06422	2800.68880
22	64	2.91	4.21977	75.79446	5.22358	273.69084	57.76644	2722.73612
23	64	2.78	1.77073	73.20970	5.21087	277.11701	61.68019	2725.71606
24	64	2.67	1.98509	104.88535	5.60208	278.21129	57.46698	2708.49239
25	64	2.56	2.38449	71.02466	5.31579	275.60900	56.76677	2716.65800
26	64	2.46	2.16866	65.23132	5.30414	275.81096	57.47008	2740.80613
27	64	2.37	0.99707	64.88587	5.36806	275.26880	58.23125	2840.78636
28	64	2.29	1.99699	96.91738	5.29000	278.08737	60.60808	2804.43005
29	64	2.21	2.08736	67.05451	5.36009	278.23877	56.75721	2774.94169
30	64	2.13	0.99587	87.06400	5.36869	275.64808	58.15400	2738.07740
31	64	2.06	1.98509	69.59807	5.36069	275.26758	54.90027	2776.53270
32	96	3.00	1.05387	90.74907	5.47871	274.77908	57.20013	2749.69441
33	96	2.91	3.27104	102.65017	5.33700	276.13807	55.65021	2771.54888
34	96	2.82	3.39403	91.15856	5.36947	273.88073	61.41043	2736.85646
35	96	2.74	2.98169	107.08427	5.32888	273.98205	58.41732	2739.08877
36	96	2.67	1.57680	65.60421	5.36951	277.12464	58.18152	2779.36587
37	96	2.59	0.98097	67.69843	5.42426	275.65885	57.67131	2805.70088
38	96	2.53	1.09949	69.61973	5.35926	274.05785	57.64127	2801.43889
39	96	2.46	1.98509	69.81301	5.41711	274.32418	60.17613	2711.37619
40	96	2.40	1.99461	66.30973	5.31790	275.38130	57.48043	2800.22409
41	96	2.34	0.98730	72.66665	5.36635	275.50607	58.25734	2766.88535
42	96	2.29	2.07567	69.90941	5.43542	273.89407	59.92794	2806.86712
43	96	2.23	2.59709	97.30792	5.47930	277.90761	63.16804	2818.67020
44	96	2.18	1.99089	91.30542	5.34864	274.36780	58.76327	2718.20650
45	96	2.13	1.99842	64.37758	5.36561	275.18797	58.01146	2698.41920
46	96	2.09	1.98127	71.00864	5.31530	274.58072	57.71136	2717.80920
47	96	2.04	0.95296	65.65034	5.30414	275.50607	57.40143	2743.54343
48	128	2.67	1.07169	90.69040	5.41077	274.54424	56.87356	2787.14685
49	128	2.61	2.07067	92.35380	5.35987	274.58056	56.43438	2762.36145
50	128	2.56	1.98699	92.07941	5.30722	275.98573	57.56000	2743.38732
51	128	2.51	1.98580	90.98017	5.58006	275.12288	57.46800	2726.46243
52	128	2.46	1.69204	90.41431	5.28607	275.60088	55.55630	2710.95385
53	128	2.42	0.98754	65.94656	5.34654	275.69088	54.96740	2737.34685
54	128	2.37	0.98730	84.88880	5.37072	277.17113	56.21529	2806.96369
55	128	2.33	1.60241	88.08257	5.43804	274.26720	55.79029	2813.41860
56	128	2.29	1.00211	65.54690	5.36580	276.67148	71.45829	2802.38578
57	128	2.25	1.00780	65.27973	5.32269	278.39540	57.84345	2778.20017
58	128	2.21	0.98730	67.01601	5.28536	275.30830	57.58729	2777.42147
59	128	2.17	1.09973	67.79857	5.40909	276.35074	56.65550	2734.16877
60	128	2.13	0.99049	69.56673	5.37801	274.26553	57.30653	2734.78090
61	128	2.10	2.00415	65.19440	5.41759	274.25623	58.17907	2730.30879
62	128	2.06	0.95750	95.06464	5.36943	275.98716	57.95513	2752.40684
63	128	2.03	3.30987	71.07963	5.36778	274.41907	57.01708	2739.05492
64	160	2.50	3.04604	70.92381	5.36709	275.41695	58.09955	2785.86125
65	160	2.46	1.00064	70.21356	5.36025	275.49857	58.10857	2732.57732
66	160	2.42	1.59927	70.59646	5.41735	274.40405	57.01516	2708.40709
67	160	2.39	1.98771	64.90163	5.35560	273.58413	57.20465	2707.39090
68	160	2.35	1.00736	108.07538	5.35440	276.43881	58.24055	2769.57985
69	160	2.32	0.54407	66.99752	5.36560	275.81204	58.79514	2813.46750
70	160	2.29	0.99444	96.70687	5.29718	274.90881	58.30646	2779.55270
71	160	2.25	3.04031	137.29585	5.43307	276.78537	57.26536	2820.98843
72	160	2.22	2.98501	88.28511	5.30891	274.90215	54.74377	2736.58343
73	160	2.19	2.30084	97.85530	5.35580	274.88060	56.45407	2718.88051
74	160	2.16	0.98845	67.97695	5.72854	274.14870	55.37908	2738.91449
75	160	2.13	2.09614	69.26727	5.27800	274.02544	58.14028	2769.27407
76	160	2.11	1.17469	96.01974	5.41825	274.30130	55.30408	2775.75490
77	160	2.08	1.99461	98.84000	5.29809	274.98746	57.08672	2735.63671
78	160	2.05	2.09685	78.88104	5.36507	275.01488	55.31621	2738.94549
79	160	2.03	2.37322	72.62974	5.34925	275.36248	56.84422	2730.68084
80	192	2.40	2.52104	69.69738	5.43652	275.71917	55.64828	2812.90178
81	192	2.37	1.99436	74.21184	5.45801	274.07384	57.90043	2714.22315
82	192	2.34	0.98821	73.18711	5.34558	274.90808	57.30036	2806.00080
83	192	2.31	1.99437	94.55681	5.36535	276.66354	60.52708	2840.75456
84	192	2.29	2.99406	86.08539	5.36819	275.06113	58.54060	2784.56351
85	192	2.26	1.74046	81.88531	5.34701	277.10271	60.34755	2804.38088
86	192	2.23	1.99413	71.41304	5.34606	275.69103	58.31920	2716.67800
87	192	2.21	1.00760	97.39648	5.34735	274.55497	57.60741	2730.60705
88	192	2.18	2.22033	90.91808	5.35540	274.84327	58.56752	2736.88507
89	192	2.16	1.99413	107.90658	5.37880	273.71358	58.40532	2729.88517
90	192	2.13	1.99485	100.73566	5.30985	275.80085	61.88855	2780.30181
91	192	2.11	0.99492	64.99987	5.37567	274.25218	57.36327	2726.71022
92	192	2.09	1.17373	67.31679	5.31087	275.29240	58.19511	2784.44862
93	192	2.06	1.00164	66.78510	5.33923	274.85132	58.38078	2747.51425
94	192	2.04	0.98730	64.50772	5.31016	273.36049	55.86514	2731.79840
95	192	2.02	0.98859	63.15637	5.32002	273.30498	54.80013	2721.67753
96	224	2.33	0.98707	75.21129	5.45507	273.98970	58.70806	2791.88095
97	224	2.31	2.00490	75.87695	5.32605	274.68014	58.68543	2806.36362
98	224	2.29	1.98533	69.00015	5.29875	273.72718	58.65536	2808.05135
99	224	2.26	1.05454	104.10070	5.36108	276.30472	58.75629	2777.25686
100	224	2.24	2.11716	65.50074	5.36915	274.72352	55.96924	2710.09779

Fig. 1. Experiment 2-T3 data

ms. RP0 takes around 2.8s, it is usable in accordance with the property of T3-level data, although it does look too long.

d) Conclusion:

1) Based on time and capacity analysis, RSA (2048) + AES256 is undoubtedly suitable for encrypting and decrypting T3-level data.
2) Although RP4B is better suited for IoT gateways, RP0's encryption and decryption time still works well for IoT applications.

5) Experiment 3-T1:

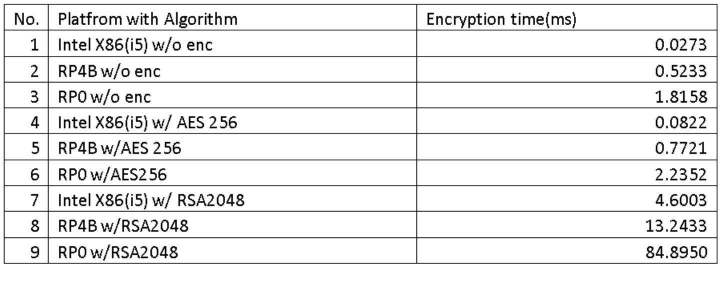
a) Plan: In an IoT application, T1-level data makes up most of the information and includes all six categories of data except for T2 and T3-level data. Ensuring the privacy of T1-level data is essential in addressing privacy concerns in IoT. However, processing T1-level data can significantly impact the performance of the middleware level. Many solutions are able to approach the requirements of diverse IoT applications. However, solutions for different security requirements have direct impact on the cost and time to market[Zolanvari and Jain(2015)]. Moreover, every solution has its own business requirements which may or may not be as strict [Jha and Sunil(2014)]. To demonstrate the results, we conducted three experiments: 1) using plain text with JSON format as a benchmark, 2) using AES256 which corresponds to the real world's Platform-S, and 3) using RSA 2048. All three experiments involved 1,000 items out of 500,000 (lengths are around 180 250bytes) and upon a public MQTT platform (broker.emqx.io). Compared with T2, T3's experiments, this one is the most complicated since:

1) To conduct the experiment accurately, we need to calculate the total time spent on both encryption and MQTT transmission. It's important to note that encryption not only takes time, but it also increases the size of the messages. Different algorithms can result in varying message lengths, which can affect the efficiency of MQTT transmission.
2) Although MQTT is not a real-time system[Soni and Makwana(2017)], it's still important for the publisher to synchronise with the subscriber to avoid losing messages and achieving a positive outcome. Calculating decryption consumption may not be relevant in this context, whereas it is deployed in experiments.

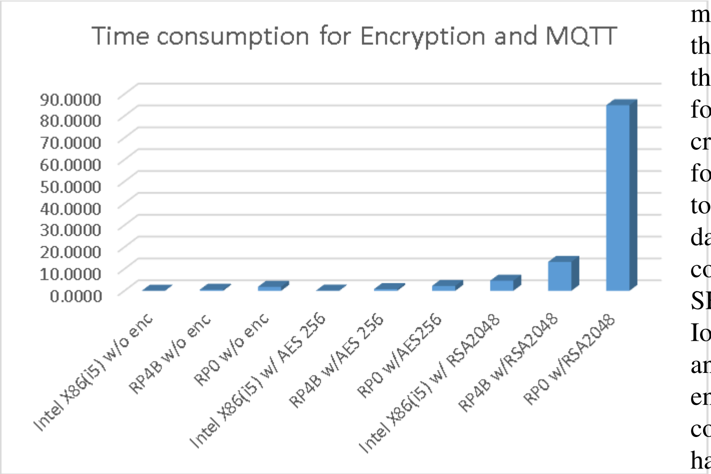
To conduct our experiment, we couldn't afford to rent an enterprise-level MQTT platform and lacked the means to set up a high-efficiency MQTT on home's LAN. As a solution, we utilised the free MQTT platform broker.emqx.io. The messages are sent one second apart. The commercial IoT application data has been approved for use in the case study, but the content has been distorted while keeping the same format. To simplify the programming, the below approaches are being taken.

1) The one-second interval is removed from the statistic.
2) The over-length message is truncated rather than pre-processed (for RSA 2048).

b) How to identify the outcome: Calculate the time consumption for 1,000 pieces of information, including both encryption and transmission, then average the time consumption for each piece of information. If the time consumption is longer than a typical value, such as 100ms, it is identified as useless. Plain text transmission is the benchmark, which is able to calculate the extra time consumption for different algorithms.



No.	Platfrom with Algorithm	Encryption time(ms)
1	Intel X86(i5) w/o enc	0.0273
2	RP4B w/o enc	0.5233
3	RP0 w/o enc	1.8158
4	Intel X86(i5) w/ AES 256	0.0822
5	RP4B w/AES 256	0.7721
6	RP0 w/AES256	2.2352
7	Intel X86(i5) w/ RSA2048	4.6003
8	RP4B w/RSA2048	13.2433
9	RP0 w/RSA2048	84.8950



c) Analysis:

1) All three platforms perform well in transmitting plaintext through MQTT, with message time consumption ranging from 0.027ms to 1.8ms. Intel X86(i5)'s performance, as another aspect of benchmark, is 0.027ms, 0.08ms, and 4.6ms while operating plaintext and the other two algorithms. While RSA2048 experiences higher consumption compared to plaintext and AES256, a message time consumption of 4.6ms is still acceptable.
2) The RP4B has an excellent performance outcome of 0.77ms per message when operating with AES256. Meanwhile, the RP0's performance of 13ms is also acceptable, especially when the IoT gateway's workload-established on RP0 or similar hardware - is medium.
3) The algorithm used by RSA2048 is more complex, which causes the encrypted message to be much longer. This results in longer processing time for encryption and MQTT transmission compared to the other two methods. While the 13 ms time consumption on RP4B is acceptable, RP0 is not suitable for RSA encryption as the 84 ms processing time means it can only handle around ten messages per second. This is not ideal for the IoT gateway's usual behaviour.

d) Conclusion:

1) The algorithm AES256 is suitable for various platforms, regardless of their cost. This makes it an excellent choice for IoT's middleware layer, particularly for low-cost platforms like RP0 or Arduino. And as a type of data which occupies 90%+ of a regular IoT application, employing a suitable algorithm is quite important.
2) RSA provides greater privacy, but it requires more resources. It is not suitable for low-cost or poorly equipped hardware, even if the message flow is minimal.

V. Discussion

The Internet of Things (IoT) is one of the emerging smart technologies for the Fourth Industrial Revolution (or Industry 4.0), which represents the ongoing automation of traditional manufacturing and industrial practices [Ślusarczyk(2018)]. Although the IoT brought infinite benefits, it creates several challenges, especially in security and privacy[Atlam and Wills(2020)]. This report aims at the middleware layer of the IoT, which is thoroughly exposed to the public, and reveals the probability of increasing privacy through diverse cryptography approaches on lightweight platforms. A clear and concise three-layer privacy model has been created for regular IoT applications by categorising six data forms. In addition, appropriate measures have been identified to enhance the privacy model, including encrypting structured data and signing unstructured data. To test our hypothesis, we conducted four experiments using AES, RSA, MD5, SHA, SHA3, and mixed algorithms on 500,000 pieces of reliable IoT data. Our findings indicate that current hardware, software, and platforms can effectively use cryptography methods to enhance privacy as long as they use algorithms that match their computational capabilities. Unfitted algorithms with specific hardware are also exhibited in the experiments. In total, due to the large amount numbers of data, limited (lightweight) hardware and low-cost trending (business purposes), concise but high-effective algorithms comply with IoT's privacy applications more. Nevertheless, the 'appropriate' is not a 'constant' but diverse in accordance with the applications of particular hardware. Notwithstanding, due to the restricted time, budget, human resources and knowledge, this report's experiments are scoped in a limited range; many algorithms have not taken part in the experiments, such as El-Gamal; and many classical elements of an IoT application are omitted, such as Cloud side's calculation; and many scenes are not simulated, such as multi-threading and multi-process. Last but not least, this report does not mention cutting-edge technologies, such as chip-level privacy protection, for example, Azure Sphere [Shi et al.(2019)]. All those could be deeply studied in the future. The definition provided by ITU in 2012 is the most common. It stated: 'a global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on, existing and evolving, interoperable information and communication technologies' [Atlam and Wills(2020)]. Currently, around 31 billion "things" are connected, and it is estimated that this number will rise to 75 billion by 2025 [Vega(2020)]. Although the IoT brought infinite benefits, it creates several challenges, especially in security and privacy[Atlam and Wills(2020)].; approximately 30 years after the birth of IoT, society is confronted with significant challenges regarding IoT security[Schiller et al.(2022)]. Hence, security and privacy are considered to be the major issues in the IoT system[Atlam and Wills(2020)]., which necessitates establishing and improving for IoT applications.

VI. Conclusion

Sophisticated security algorithms are not suitable for these constrained devices as they are not able to execute complex processing operations in real time[Atlam and Wills(2020)]. Instead, constrained devices typically only employ fast, lightweight encryption algorithms [Jurcut et al.(2020)]. It makes sense to categorise common IoT data into six types and then condense them into three levels of privacy. Three various algorithms can be used to confront the privacy requirements for each level. AES suits the vast majority of the upstreaming messages due to its efficiency although it looks not so complicated, RSA or even RSA+AES (resembling a ransomware's algorithm) requires adequate hardware, which doesn't fulfil frequent use, in low-configured hardware particularly; however, it is secure for high privacy required and low-frequency messages, even on a low-configured platform. It is important to understand the uses of encryption and signature to protect privacy. Signature methods work well for unstructured messages with a lot of data, while high-level encryption like SHA3 is more effective but may not be suitable for lightweight IoT platforms. MD5 is a simpler encryption method that can work for various applications, even with low-equipped hardware. However, it is not recommended to use low-equipped hardware to process unstructured IoT messages. All experiments' programs and data can be downloaded from below address: https://github.com/Lemon-Ice-Tea/727ReportExperiment.git

References

[Atlam et al.(2018)] Hany Fathy Atlam, Robert Walters, and Gary Wills. 2018. Internet of things: state-of-the-art, challenges, applications, and open issues. *International Journal of Intelligent Computing Research (IJICR)* 9, 3 (2018), 928–938.

[Atlam and Wills(2020)] Hany F Atlam and Gary B Wills. 2020. IoT security, privacy, safety and ethics. *Digital twin technologies and smart cities* (2020), 123–149.

[Atzori et al.(2012)] Luigi Atzori, Antonio Iera, Giacomo Morabito, and Michele Nitti. 2012. The social internet of things (siot)–when social networks meet the internet of things: Concept, architecture and network characterization. *Computer networks* 56, 16 (2012), 3594–3608.

[Avertium([n. d.])] Avertium. [n. d.]. An in-depth look at pandora ransomware. https://explore.avertium.com/resource/in-depth-pandora-ransomware

[Barik et al.(2016)] Runa Barik, Michael Welzl, and Ahmed Elmokashfi. 2016. How to say that you're special: Can we use bits in the IPv4 header?. In *Proceedings of the 2016 Applied Networking Research Workshop*. 68–70.

[Garcia-Morchon et al.(2019)] Oscar Garcia-Morchon, Sandeep Kumar, and Mohit Sethi. 2019. Internet of Things (IoT) security: State of the art and challenges. (2019).

[Hassija et al.(2019)] Vikas Hassija, Vinay Chamola, Vikas Saxena, Divyansh Jain, Pranav Goyal, and Biplab Sikdar. 2019. A survey on IoT security: application areas, security threats, and solution architectures. *IEEE Access* 7 (2019), 82721–82743.

[Hillar(2016)] Gaston C Hillar. 2016. *Internet of Things with Python*. Packt Publishing Ltd.

[Jha and Sunil(2014)] Ajit Jha and MC Sunil. 2014. Security considerations for Internet of Things. *L&T Technology Services* (2014).

[Jurcut et al.(2020)] Anca D Jurcut, Pasika Ranaweera, and Lina Xu. 2020. Introduction to IoT security. *IoT security: advances in authentication* (2020), 27–64.

[Jurvetson(2019)] Steve Jurvetson. 2019. How a quantum computer could break 2048-bit RSA encryption in 8 hours. *MIT Technology Review, May* 30 (2019), 9.

[Kügler(2003)] Dennis Kügler. 2003. "Man in the middle" attacks on bluetooth. In *Financial Cryptography: 7th International Conference, FC 2003, Guadeloupe, French West Indies, January 27-30, 2003. Revised Papers 7*. Springer, 149–161.

[Kurniawan and Kurniawan(2019)] Agus Kurniawan and Agus Kurniawan. 2019. Introduction to raspberry pi. *Raspbian OS Programming with the Raspberry Pi: IoT Projects with Wolfram, Mathematica, and Scratch* (2019), 1–25.

[Leo et al.(2014)] Marco Leo, Federica Battisti, Marco Carli, and Alessandro Neri. 2014. A federated architecture approach for Internet of Things security. In *2014 Euro Med Telco Conference (EMTC)*. IEEE, 1–5.

[Pittalia(2019)] Prashant P Pittalia. 2019. A comparative study of hash algorithms in cryptography. *International Journal of Computer Science and Mobile Computing* 8, 6 (2019), 147–152.

[Plauska et al.(2022)] Ignas Plauska, Agnius Liutkevičius, and Audronė Janavičiūtė. 2022. Performance Evaluation of C/C++, MicroPython, Rust and TinyGo Programming Languages on ESP32 Microcontroller. *Electronics* 12, 1 (2022), 143.

[Sarker et al.(2022)] Iqbal H Sarker, Asif Irshad Khan, Yoosef B Abushark, and Fawaz Alsolami. 2022. Internet of things (iot) security intelligence: a comprehensive overview, machine learning solutions and research directions. *Mobile Networks and Applications* (2022), 1–17.

[Schiller et al.(2022)] Eryk Schiller, Andy Aidoo, Jara Fuhrer, Jonathan Stahl, Michael Ziörjen, and Burkhard Stiller. 2022. Landscape of IoT security. *Computer Science Review* 44 (2022), 100467.

[Shen et al.(2005)] Guobin Shen, Guang-Ping Gao, Shipeng Li, Heung-Yeung Shum, and Ya-Qin Zhang. 2005. Accelerate video decoding with generic GPU. *IEEE Transactions on circuits and systems for video technology* 15, 5 (2005), 685–693.

[Shi et al.(2019)] Jiong Shi, Liping Jin, and Jun Li. 2019. The integration of azure sphere and azure cloud services for internet of things. *Applied Sciences* 9, 13 (2019), 2746.

[Ślusarczyk(2018)] Beata Ślusarczyk. 2018. Industry 4.0: Are we ready? *Polish Journal of Management Studies* 17 (2018).

[Soni and Makwana(2017)] Dipa Soni and Ashwin Makwana. 2017. A survey on mqtt: a protocol of internet of things (iot). In *International conference on telecommunication, power analysis and computing techniques (ICTPACT-2017)*, Vol. 20. 173–177.

[Swire(2018)] Peter Swire. 2018. A pedagogic cybersecurity framework. *Commun. ACM* 61, 10 (2018), 23–26.

[Symes(2004)] Peter D Symes. 2004. *Digital video compression*. McGraw Hill Professional.

[Ukil et al.(2011)] Arijit Ukil, Jaydip Sen, and Sripad Koilakonda. 2011. Embedded security for Internet of Things. In *2011 2nd National Conference on Emerging Trends and Applications in Computer Science*. IEEE, 1–6.

[Vega(2020)] Malvina Vega. 2020. Internet of things statistics facts & predictions [2020's update]. *Retrieved Novemb* 30 (2020), 2020.

[Yousuf et al.(2015)] Tasneem Yousuf, Rwan Mahmoud, Fadi Aloul, and Imran Zualkernan. 2015. Internet of things (IoT) security: current status, challenges and countermeasures. *International Journal for Information Security Research (IJISR)* 5, 4 (2015), 608–616.

[Zhang et al.(2014)] Zhi-Kai Zhang, Michael Cheng Yi Cho, Chia-Wei Wang, Chia-Wei Hsu, Chong-Kuan Chen, and Shiuhpyng Shieh. 2014. IoT security: ongoing challenges and research opportunities. In *2014 IEEE 7th international conference on service-oriented computing and applications*. IEEE, 230–234.

[Zhao and Ge(2013)] Kai Zhao and Lina Ge. 2013. A survey on the internet of things security. In *2013 Ninth international conference on computational intelligence and security*. IEEE, 663–667.

[Zolanvari and Jain(2015)] Maede Zolanvari and R Jain. 2015. IoT security: a survey. *Computer Scientists & Computer Engineers at WashU* (2015).