

Secure Real-time Data Transmission for Drone Delivery Services using Forward Prediction Scheduling SCTP

Febby Ronaldo¹, Amang Sudarsono², Dadet Pramadihanto¹

¹ Department of Informatics and Computer Engineering, Politeknik Elektronika Negeri Surabaya

² Department of Electrical Engineering, Politeknik Elektronika Negeri Surabaya
Correspondence Author: dadet@pens.ac.id

Received March 2, 2022; Revised April 3, 2022; Accepted May 5, 2022

Abstract

Drone technology is considered the most effective solution for the improvement of various industrial fields. As a delivery service, drones need a secure communication system that is also able to manage all of the information data in real-time. However, because the data transmission process occurs in a wireless network, data will be sent over a channel that is more unstable and vulnerable to attack. Thus, this research, purposes a Forward Prediction Scheduling-based Stream Control Transmission Protocol (FPS-SCTP) scheme that is implemented on drone data transmission system. This scheme supports piggybacking, multi-streaming, and Late Messages Filter (LMF) which will improve the real-time transmission process in IEEE 802.11 wireless network. Meanwhile, on the cybersecurity aspect, this scheme provides the embedded option feature to enable the encryption mechanism using AES and the digital signatures mechanism using ECDSA. The results show that the FPS-SCTP scheme has better network performance than the default SCTP, and provides full security services with low computation time. This research contributes to providing a communication protocol scheme that is suitable for use on the internet of drones' environment, both in real-time and reliable security levels.

Keywords: Real-time Communication, SCTP, Secure, Hybrid Cryptography, Drones.

1. INTRODUCTION

Various industries are starting to use drone technology to improve work efficiency. This has a positive impact as well as a challenge for researchers to develop drone functions so that they can integrate with other technological innovations. Due to their ease of deployment and reconfiguration, drones research often focuses on missions such as monitoring, delivery service, agriculture, and even as future transportation systems [1,2]. In delivery services, drones are the most effective solution to the issue of traffic jams. No traffic jams mean increased mobility, expanded expedition area, and time fare

would be predictable. However, to achieve this goal, we need an environment in which drones can operate in an organized and centralized manner.

Cloud-Based Drone Management System (CBDMS) is a system that is currently being developed to support the internet of drones' environment. In general, CBDMS manages all traffic information data received or sent from drones to servers, including flight management and scheduling. Because of the many critical functions that exist, CBDMS requires a dynamic system that can work optimally in real-time. Thus, the most basic thing that can be highlighted to meet these requirements is the ability of drones and servers to transmit data to each other, state-by-state, in real-time over wireless networks [3].

However, the wireless network usually refers to an unstable and insecure network, so several aspects need to be considered. First, The real-time communication system intended for CBDMS should have an algorithm with low complexity and guarantee timeliness by considering the speed of message arrival time [4]. Second, providing real-time communication capability is not enough to ensure the exchange of data between entities in the system runs safely. Transmissions that occur on the wireless network must be prepared to face the risk of attack from illegal third parties (e.g., man-in-the-middle, eavesdropping, etc) by providing security guarantees for data privacy and entity authentication [1].

In this research, we purpose a secure real-time data transmission scheme for drone delivery services using Forward Prediction Scheduling-based Stream Control Transmission Protocol (FPS-SCTP). We chose SCTP as the basis for transport protocol because it can be enabled as a reliable or partially reliable transmission protocol [5,6]. This is reasonable because SCTP has more advantages including having the best effort of the TCP and UDP protocols [7–9]. In addition, our proposed scheme is closely related to wireless communication and each wireless device has only one active interface at a time. Thus, we also considered interesting features of SCTP such as piggybacking and multi-streaming. With this multi-streaming feature, SCTP can activate several independent parallel streams and combine them in one SCTP association.

Since FPS-SCTP has a specific implementation goal, we simplified a few existing methods for real-time communication between drones and servers [10]. We adapted a messages screening system that could potentially not reach the destination node on time. We called it as Late Messages Filter (LMF) Algorithm.

Meanwhile, we are also very concerned about the safety system that we propose so as not to burden the drone's performance in a real environment. We used the Lightweight Hybrid Cryptography scheme (LHC). We compared several symmetric cryptographic mechanisms to prove which one is more suitable for our system. Finally, we used AES as a form of data privacy protection, and ECDSA as a form of entity authentication guarantee.

This paper is composed of the following sections: related work is introduced in section 2. Then in section 3, it is explained about the originality

of the research. Our proposed FPS-SCTP scheme is described more in section 4, where the experimental analysis is in section 5. Section 6 is the conclusion.

2. RELATED WORKS

Drone Technology can be classified as a Real-Time System (RTS), so a system that supports real-time communication is needed. A real-time communication system should have a low complexity algorithm, timeliness/predictability, and security [4].

Until now, there has been a lot of research to create a reliable real-time communication system on drones. This will not be separated from the two transport-layer protocols that are currently the most commonly used, such as TCP and UDP. In our previous research, we have tried to implement the Message Queuing Telemetry Transport (MQTT) protocol as a communication medium between drones and servers. The MQTT protocol is a communication protocol at the application layer, which is built on top of the TCP protocol. From our previous research, we have analyzed that the communication side should be improved [3].

2.1 Stream Control Transmission Protocol (SCTP)

The SCTP is another transport-layer protocol, which in some studies, its performance can outperform both UDP and TCP protocols. [7–9]. SCTP protocol combines the best effort of TCP and UDP. It supports reliable transmissions, partially reliable transmissions, connection-oriented, congestion/flow control, SACKs, multi-homing, and multiple-stream [5,6,10,11]. In the multiple-stream services, each stream is independent [10]. It means that when one stream is blocked, the other streams are still able to transmit data, Transmission Sequence Number (TSN), Stream Identifier (SI), and Stream Sequence Number (SSN) are defined in the data chunk. Every stream in SCTP can be identified using SI. On the same SCTP stream, each data chunk can be distinguished by an SSN. Each data chunk is numbered using a unique TSN [5,6].

In 2004, a new concept of SCTP called Partially Reliable SCTP was introduced by the IETF [11]. PR-SCTP scheme allows nodes to specify a chunk retransmission threshold. However, the PR-SCTP mechanism can only work if both transmitter and receiver nodes agree to have agreed to enable the PR-SCTP function. In its implementation, several studies have proven that PR-SCTP is more suitable for real-time systems than common SCTP [10,11].

Meanwhile, in 2019, some researchers offered a new scheme that improved the performance of the SCTP protocol for real-time transmission in the wireless network. The scheme was called Cross-Layer SCTP (CL-SCTP). CL-SCTP is designed to work across the application layer and MAC layer. On the application layer, the late messages filtering algorithm is designed as a mechanism to prevent ineffective message transmission. Then, on the MAC layer, the redundant frame detection algorithm is designed as a backup mechanism to deal with redundant frames from the SCTP transport layer. In

In addition, this scheme also provides a function to calculate and broadcast the probability of a low signal-to-noise ratio to wireless end devices in a Wireless Access Point (WAP) or Internet of Things (IoT). Using the NS-2 simulator, this study compares the performance of PR-SCTP and SCTP based on several aspects. In the simulation of a wireless network having unstable channels, random loss rate and burst loss rate are analyzed. Both in terms of efficiency delivery ratio, average goodput, and average end-to-end delay, CL-SCTP has a better performance than SCTP and PR-SCTP [10].

Table 1. Existing standard security solutions for SCTP

Number	Title	Description
RFC 3436	Transport Layer Security over Stream Control Transmission Protocol [12]	<ul style="list-style-type: none"> - TLS is designed for ordered messages transmission in a transport protocol. - PR-SCTP should not be used (TLS assumes all messages are delivered). - Equal number of streams on both directions
RFC 6083	Datagram Transport Layer Security (DTLS) for Stream Control Transmission Protocol (SCTP) [13]	<ul style="list-style-type: none"> - Supports PR-SCTP - Provides messages fragmentation and retransmission timer - Supports unidirectional and bidirectional streams - Applies handshake only (DTLS can drop the packet containing DATA app.)
RFC 3554	On the Use of Stream Control Transmission Protocol (SCTP) with IPsec [14]	<ul style="list-style-type: none"> - IPsec used below SCTP - Uses Security Association (SA) for confidentiality and integrity protection - Requires to create new SA for each new dynamically connected IP address
RFC 4895	Authenticated Chunks for the Stream Control Transmission Protocol (SCTP) [15]	<ul style="list-style-type: none"> - Extension for data integrity and authentication mechanism - Implemented in COOKIE and AUTH chunk - Uses Hashed Message Authentication Code (HMAC)

Apart from the issue of real-time transmission, wireless networks are vulnerable to attack. SCTP, a new transmission protocol, already has several standard end-to-end security solutions as summarized in table 1. These solutions are not designed especially for SCTP, so some advanced techniques in SCTP may not be saved intact when the standard solution is used. In other studies, cryptography is used to facilitate data confidentiality and an SCTP auth extension is utilized to authenticate chunk each mess received. [16].

3. ORIGINALITY

We designed a new Stream Control Transfer Protocol (SCTP) scheme that can be optimally implemented on the Internet of Drones (IoD) environment. Since the system is close to the lightweight devices, we consider the complexity of the algorithm we use by choosing a lightweight algorithm and simplifying an existing mechanism. Forward Prediction Scheduling-based Stream Control Transmission Protocol (FPS-SCTP) is designed to offer the following functions:

- The scheme is designed to be suitable for communication between drones and servers in wireless networks.
- The Late Messages Filter (LMF) algorithm is used to predict and filter out messages that are unlikely to arrive on time. LMF requires retransmission time and delay budget to determine the urgency level of each packet. LMF is a simplification of the cross-layer algorithm of existing research [10], so it is more suitable for lightweight devices.
- FPS-SCTP scheme provides a higher level of security during the establishment process and data transfer using Lightweight Hybrid Cryptography (LHC). LHC combines AES-256 and ECDSA, which is a proven lightweight algorithm [3,17,18], to obtain full security services, including data confidentiality, integrity, authentication, and non-repudiation. This feature can be seen as an optional feature that can be enabled if both nodes agree.

4. SYSTEM DESIGN

We designed a huge system scheme to create a secure Internet of Drones (IoD) environment. Centralized in the cloud, the server was able to manage traffic, flight schedules, and forecast weather conditions at each point where the drone was located. However, to introduce our proposed scheme, we tried to narrow down our system illustration. The scenario of how servers and drones exchange data on a wireless network was illustrated in figure 1.

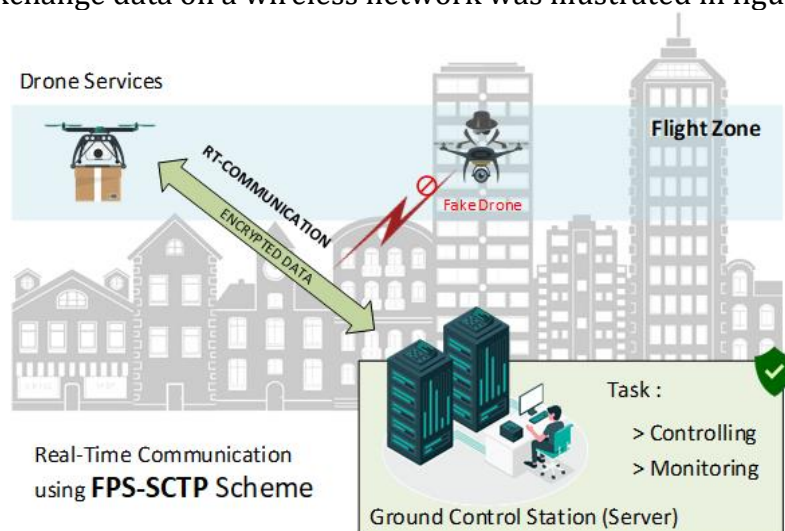


Figure 1. Illustration System of Proposed Secure Real-time Communication

The wireless network is often referred to as an unstable channel condition, so a scheme that generalizes the reliability level of chunk messages is not suitable to be adapted. In addition, since wireless networks are also vulnerable to cyber-attacks, it is necessary to guarantee the security of both data and registered entities. Thus, considering those two issues, we proposed Forward Prediction Scheduling based SCTP scheme for a drone data transmission process.

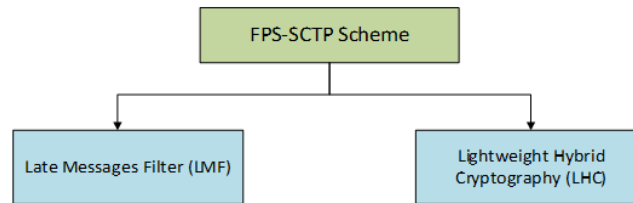


Figure 2. The FPS-SCTP scheme consists of two mechanisms, including Late Messages Filter (LMF) and Lightweight Hybrid Cryptography (LHC).

As shown in Figure 2, our proposed FPS-SCTP scheme consisted of two main components, including Late Messages Filter (LMF) and Lightweight Hybrid Cryptography (LHC). These two functions were embedded in each end device. We used a partially reliable chunk in the LMF algorithm. In addition, we gave users the flexibility to set the security level on FPS-SCTP provided that both ends of the node used the same security level.

4.1 Late Messages Filter (LMF)

LMF is a mechanism for filtering out potentially late messages when they arrive at the receiver. The block diagram of the LMF is as shown in figure 3, where it was implemented on the application layer. LMF consists of 3 steps, including how to determine Packet Loss Rate (PLR), expected retransmission time, and scheduling. For a real-time requirement, each chunk n has a Packet Delay Budget (PDB) value which depends on its Quality of Class Identifier (QCI).

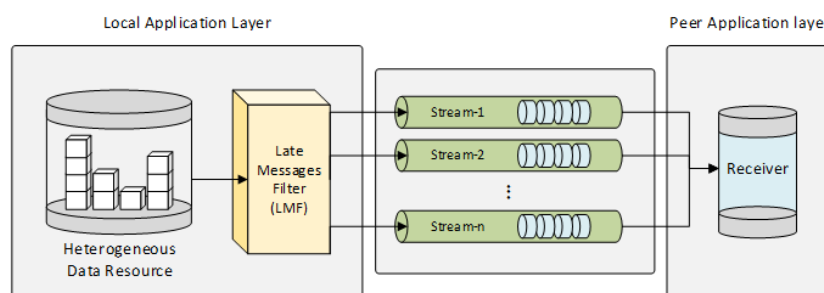


Figure 3. The block diagram of the LMF mechanism

Step 1. Get Packet Loss Rate value (P_{PLR})

Packet Loss Rate, P_{PLR} , is obtained from the number of a retransmission packet, N_R , divided by the number of transmissions, N_T , as in equation 1.

$$P_{PLR} = \frac{N_R}{N_T} = \frac{N_{FR} + N_{RTO}}{N_T} \quad (1)$$

The number of retransmission may consist of the number of retransmissions due to timeouts, N_{RTO} , number of retransmissions due to fast retransmit, N_{FR} , or both.

Step 2. Get an Expected Retransmission Time (t_{Rtx})

We need Retransmission Time Out (RTO) value, t_{RTO} , and fast retransmit time, t_{FR} , to compute t_{Rtx} , as shown in equation 2.

$$t_{Rtx} = P_{PLR} \times \left(\frac{N_{RTO}}{N_R} \times t_{RTO} + \frac{N_{FR}}{N_R} \times t_{FR} \right) \quad (2)$$

Based on RFC 4690 [5], The current RTO value, is obtained from two states variables. When a new RTT measurement, R^* , is exists, we computed SRTT of the i -th transmission, \dot{R}_i , and RTTVAR, R_v , as in equations 3 to 4. The range of α and β is $0 \leq [\alpha, \beta] \leq 1$. For SCTP protocol, recommended values of α are 0.125 and $\beta = 0.25$.

$$\dot{R}_i = (1 - \alpha) \times \dot{R}_i + \alpha \times R^* \quad (3)$$

$$R_v = (1 - \beta) \times R_v + \beta \times |\dot{R}_i - R^*| \quad (4)$$

The RTO value, t_{RTO} , is computed as in equation 3. Normally, the initial RTO value is 3 seconds, and it is updated when a new RTT is received. Whenever RTO is calculated, G is the minimum value that should be set. Recommended value of G is 100 milliseconds. If it is less than G, then RTO is rounded up to G.

$$t_{RTO} = \max[G, \dot{R}_i + 4 \times R_v] \quad (5)$$

Fast retransmit occurs when a duplicate ack arrives at the sender up to at least 3 times. Fast retransmit time, t_{FR} , is computed as in equation 6. CW_i is the Congestion Window size of the i -th transmission and waiting time to trigger fast retransmit should be in range $\dot{R}_i + (\dot{R}_i / CW_i) * 2$ to t_{RTO} .

$$t_{FR} = \sum_{k=0}^{N_R-3} (P_{PLR}^k (1 - P_{PLR})^3) \times \min \left[\left(\dot{R}_i + \frac{\dot{R}_i \times 2}{CW_i} \right), t_{rto} \right] \quad (6)$$

Step 3. Scheduling packet transmission

We used a partially reliable chunk on our proposed scheme. In this step, we set the urgency level of the packet, C_n , as in equation 7.

$$C_n = \left\lceil \frac{PDB_n}{t_{Rtx}} \right\rceil \quad (7)$$

If the urgency level ≥ 1 , then the packet is forwarded to the transport layer for retransmission. We prioritized the retransmission of packets with an urgency level that has a value close to zero. It means a lower urgency level has a higher priority to transmit. Besides that, if the urgency level = 0, then we considered that the packet was impossible to arrive at its destination on time, and we decided to drop the packet.

4.2 Lightweight Hybrid Cryptography (LHC)

SCTP has a security mechanism that is embedded in the association establishment process namely “cookie”. This cookie is the result of baking the INIT data using a hash function and symmetric key. In the end, this mechanism provide authentication for SCTP. SCTP only enables hash functions like MD5 and SHA1 to create cookies in the initial configuration. In our proposed scheme, we slightly modified the SCTP kernel so that the HMAC-SHA256 function could be used. For one-to-one communication scenarios, this modified configuration was sufficient to authenticate peers.

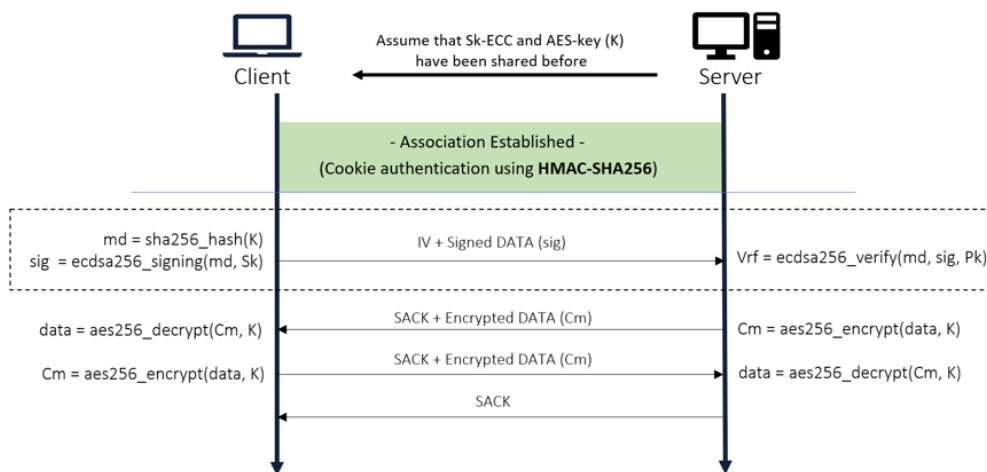


Figure 4. The Lightweight Hybrid Cryptography Mechanism

However, full security services were required in its implementation on drone transmission. HMAC-256 is only sufficient to ensure data integrity and authentication. We offered users the opportunity to use a LHC scheme. The LHC scheme consists of key generation, HMAC-256 authentication in the association establishment phase, digital signature, and encryption-decryption process. The digital signature is used to provide more security services including non-repudiation goals.

We used Elliptic Curve Digital Signatures Algorithm (ECDSA) for the signing and verifying process, as in Figure 4. It was supported by several studies which state that the elliptic curve algorithm produces a strong level of security by using a smaller key than RSA [3,18,19]. Thus, ECDSA has proven to be suitable for lightweight devices. Signatures and verification were only processed once after each association was established.

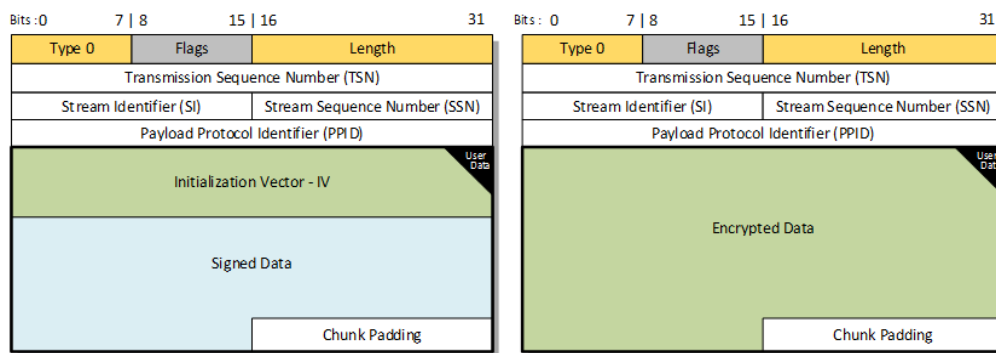


Figure 5. Structure of (a) the signed messages; and (b) encrypted messages

The basic chunk data structure of SCTP consists of a "DATA" chunk header and a "USER DATA". This research embedded initialization vector (IV) and signed messages in "USER DATA" during the first data exchange, as in Figure 5a. IV was required by symmetric cryptographic algorithms as a shared-key and a unique binary sequence. We separated IV data from encrypted data. It has two advantages: it avoids less efficient mechanisms such as repeated IV transmissions, and it reduces message size while encrypted data is sent. The cost of sending IV is 16 bytes and less than 80 bytes for signed data. Thus, the maximum cost of this first message is 96 bytes including the padding.

Our data confidentiality was obtained via the AES-256 algorithm [3,17,19]. We used Cipher Block Chaining (CBC) operation as the mode in the AES algorithm. The data should have a length that matches the full number of AES blocks (16 bytes per block). Thus, padding in AES is required for any messages that do not fill the last block.

5. EXPERIMENT AND ANALYSIS

As in figure 6, we used the Raspbian OS on both client and server nodes. The reason for using Raspbian OS was that the SCTP kernel implementation was built into the system. Raspberry Pi 4 as a portable device was also suitable for representing the state of drones. Tested in the real environment (indoor), two nodes placed in a line of sight/no obstacle scenario with a controlled distance. When an association was established, state information from the client node at fixed positions was sent to the server continuously in real-time. As soon as the server received the message, server sent a message of the same length to simulate two-way communication.

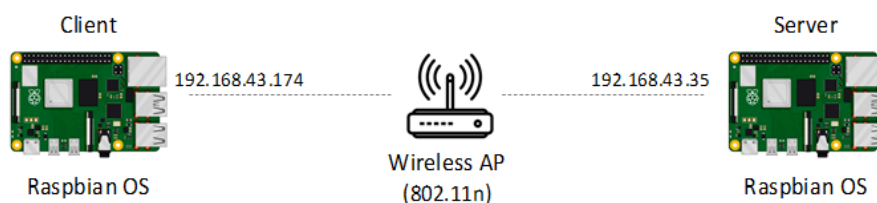


Figure 6. Testbed scenario for network performance evaluation

Table 2. Experiment Parameters

No	Parameters	Setting Value
SCTP Parameters		
1	Maximum Transmission unit for each path (PMTU)	1500
2	Chunks Size Estimation	1468 Bytes
3	Number Outbound Streams between sender and receiver	64
4	Congestion Window Size (cwnd)	MTU x 2
5	Number of repetitions to trigger fast retransmit	3
6	Size of Receiver Window (initialRwnd)	65536 Bytes
Wireless Parameters		
7	Network Size	20 x 20 meters
8	Transmission Range	20 meters
9	Wireless MAC	802.11n
Cryptography Parameters		
10	AES Cipher Key Size	256 bits
11	AES mode of operation	CBC
12	SHA-2 Digest Size	256 bits
13	ECDSA Key Size	256 bits

The parameters for FPS-SCTP, wireless network state, and hybrid cryptography were listed in table 2. However, there were two variables that we adjusted during the test, such as the size of the distance between the nodes and the size of the message sent. In addition, We considered timeliness in transmission. Thus, by looking at PDB and the cost of processing time, the client node would send messages at least every 100 milliseconds or less.

There were several measurements tested, which consisted of an analysis of network performance, the processing time of security schemes, and evaluation of security services. The variables measured in the network performance test included the amount of delay, throughput, and packet loss rate. In processing time analysis, we compared several symmetric cryptographies to determine which one had the lightest and fastest computation time. We also analyzed the cost of time required when the LHC scheme was used or not. Finally, we evaluated our proposed scheme, specifically on security services.

A. Network Performance Analysis

We compared the performance of our proposed FPS-SCTP scheme with the default SCTP when transmitting messages. Default-SCTP was the reliable channel, and FPS-SCTP was the partially reliable channel that provided a level of urgency for each message. Since it only used one IEEE 802.11n interface, we tested all two in parallel with the same parameters. At the point of network performance analysis, the results had been collected to visualize the different values in terms of packet loss rate, average throughput, and max smoothed RTT/delay (excluded RTO state).

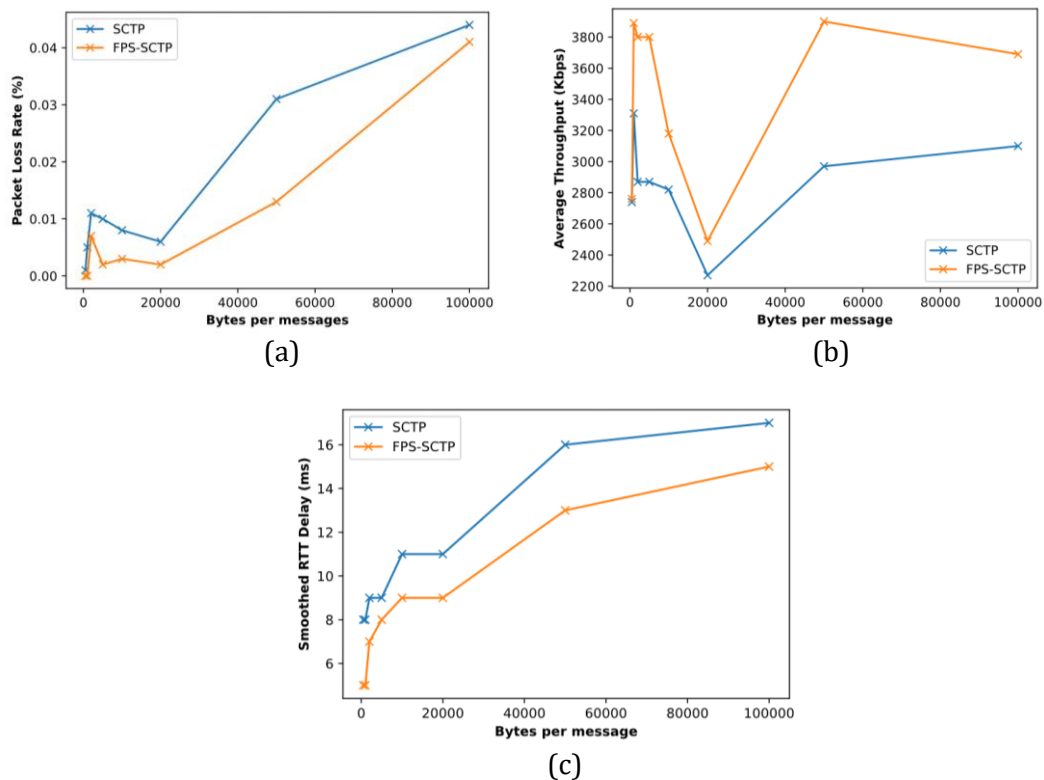


Figure 7. (a) Size of Messages versus Packet Loss Rate, (b) Average throughput, and (c) Max Smoothed RTT delay.

As in Figure 7a, results showed that, in various tests with different size of messages, FPS-SCTP had a lower average packet loss rate than the default SCTP. Figure 7a showed that our FPS-SCTP also had a more stable output than default SCTP in some tests. Changes up and down the value of the packet loss rate were influenced by the state of the wireless network at the time of testing. However, this value was not so significant because both had packet loss rates that were close to 0%.

The throughput value of both the default-SCTP and FPS-SCTP showed unstable results on the wireless network. The resulted throughput value was the average value of packet transmission observations for 10 seconds so that there could be changes in network quality that made the output pattern unstable. However, under the same environmental state, the throughput of FPS-SCTP showed a greater value than the default SCTP. As in figure 7b, FPS-SCTP showed the highest measured throughput value of 3.9 Mbps and 3.31 Mbps for SCTP default.

Meanwhile, the comparison of smoothed RTT performance between FPS-SCTP and default SCTP showed clear results. As in figure 7c, the results showed that FPS-SCTP had a lower delay value than the default SCTP. Moreover, in both FPS-SCTP and default SCTP, the smoothed RTT value increased with the size of the transmitted message. From these results, it

could be concluded that FPS-SCTP had better performance than the default SCTP in terms of average throughput, and smoothed RTT.

B. Processing Time Analysis

To evaluate the security level, the size of the message was adjusted. Oriented towards high-security levels with low complexity, we compared the processing time differences for several message encryption mechanisms, including AES, DES, 3DES, and blowfish. In addition, the processing time of LHC as optional features was also recorded to determine the difference between when this feature was used or not.

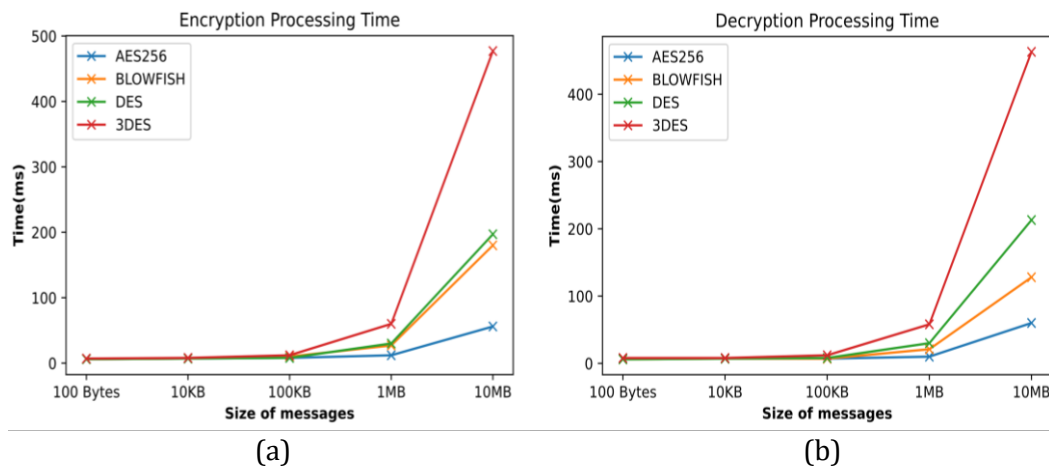


Figure 8. (a) encryption and (b) decryption processing time of several symmetry cryptography mechanisms

As in Figure 8, the performance of the 4 cryptographic mechanisms was compared. Both in the encryption and decryption process, the resulted processing time did not have a large difference in small messages. After 100 Kilobytes, it was known that a value gap appeared. As a sample, at a message size of 10 Megabytes, the AES-256 encryption time was at 56 ms, 180 ms for blowfish, 197 ms for DES, and 477 ms for 3DES. The test results concluded and at the same time proved that both in the encryption and decryption process, the AES-256 algorithm had a faster processing time than the other three algorithms. AES with a 256-bit key had excellent security and flexibility. Thus, the data confidentiality from the LHC scheme was carried out by AES-256 with CBC mode.

As listed in table 3, The server time required to generate an EC-key was relatively long, whereas if it included the time for generate a symmetric key, it took 27.36 milliseconds. Meanwhile, the client node only took less than 1 millisecond to generate a random IV key of 128 bits. The authentication process only took place on the client node because the server only echoed the cookie without changed the data. We measured it along with all association establishment process that took less than 1 millisecond.

Table 3. LHC Processing Time

State	Processing Time (ms)	
	Client	Server
Preparation time (key generation)	0,8443	27,3618
Assoc. Establishment using HMAC-256	0,9911	-
ECDSA Signing	9,5467	-
ECDSA Verification	-	10,9976
AES-256-CBC Encryption	0,4489	0,4154
AES-256-CBC Decryption	0,3687	0,3013

After the association was established the client node signed the first message for 9.55 seconds, and the server verified the message in 11 milliseconds. It did not include the merging and parsing process between the IV and the signed data sent together. Lastly, because the size of the drone data sent was less than 128 bytes, AES could encrypt and decrypt messages in less than 0.5 milliseconds. With these results, we concluded that the LHC scheme was quite suitable to be implemented in the system because it had a low computational cost.

C. Security Services Evaluation

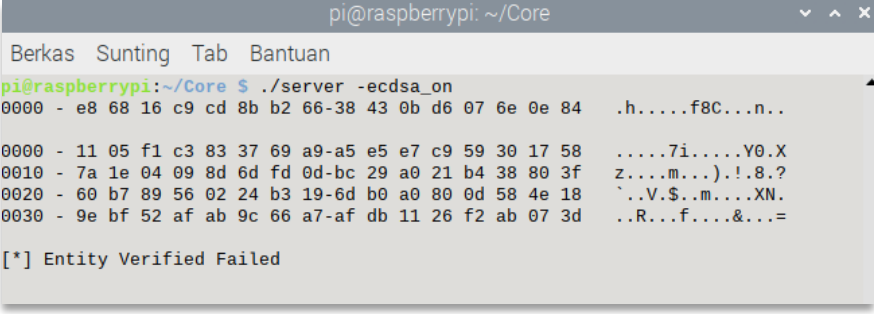
As listed in table 4, FPS-SCTP offered full-reliable security services for drone communication systems. We guaranteed the confidentiality of data from each node using AES cryptography which had been proven to had low computation time compared to other symmetric cryptographic algorithms. Meanwhile, the ECDSA algorithm existed for data integrity, authentication, and non-repudiation purposes. However, we still maintained the use of HMAC-256 as an authentication cookie in the association establishment phase.

Table 4. Security Services Comparison

Services	SCTP	S ² -SCTP 2.0 [16]	FPS-SCTP
Integrity	v	v	v
Authentication	v	v	v
Confidentiality	-	v	v
Non-repudiation	-	-	v

In our previous [3] and existing research[16], both had in common where the authentication process was used every time a message was sent. S²-SCTP 2.0 leveraged SCTP's built-in authentication extension, and our previous scheme used SHA-256 as the authentication mechanism. In this research, we considered that the use of a message authentication mechanism in each transmission might incur the double cost of processing each node. Thus, we proposed the use of ECDSA only at the start of the data transfer phase. This provided at least three advantages: 1) Without using authentication on every transmission, it meant that processing time for each transmission could be reduced. 2) ECDSA was a fairly complex algorithm but had a very strong level of security [17][18]. Uses wisely only at the beginning of the process, ECDSA

would provide strong data integrity and authentication so that it was difficult for hackers to attack. 3) Unlike message authentication codes (MAC), ECDSA is a digital signature algorithm with non-repudiation. Thus, providing more security services than the scheme without ECDSA. In addition, the unique asymmetric key of each node could be used as a way to identify which entity was connected to the server. This method would be very effective when applied to the one-to-many communication later.



```

pi@raspberrypi: ~/Core
Berkas  Sunting  Tab  Bantuan
pi@raspberrypi:~/Core $ ./server -ecdsa_on
0000 - e8 68 16 c9 cd 8b b2 66-38 43 0b d6 07 6e 0e 84  .h.....f8C...n..
0000 - 11 05 f1 c3 83 37 69 a9-a5 e5 e7 c9 59 30 17 58  ....7i.....Y0.X
0010 - 7a 1e 04 09 8d 6d fd 0d-bc 29 a0 21 b4 38 80 3f  z....m...).!.8.?
0020 - 60 b7 89 56 02 24 b3 19-6d b0 a0 80 0d 58 4e 18  `..V$.m...XN.
0030 - 9e bf 52 af ab 9c 66 a7-af db 11 26 f2 ab 07 3d  ..R...f....&...=

[*] Entity Verified Failed

```

Figure 9. The result of server refusal actions against fake drones or unauthorized associations. The server will return to idle state waiting for another association.

6. CONCLUSION

In this research, we propose FPS-SCTP scheme. FPS-SCTP offers Late Messages Filter (LMF) as a way to improve real-time communication performance and Lightweight Hybrid Cryptography (LHC) to provide reliable full security services on lightweight devices especially on the internet of drones' environment. On unstable wireless networks, the results show that FPS-SCTP performs better than default-SCTP in terms of average throughput and Round Trip Time/delay. Meanwhile, the results of the proposed lightweight hybrid cryptography scheme show low computation time, where 12.2 milliseconds on the client node and 36.08 milliseconds on the server node. In our future research, a scheme that can facilitate multiple drones communicating to the server will be proposed.

Acknowledgments

This research was fully supported by Robotics and Intelligent System Center (RoISC).

REFERENCES

- [1] Choudhary G, Sharma V, Gupta T, Kim J, You I. **Internet of Drones (IoD): Threats, Vulnerability, and Security Perspectives.** :14.
- [2] Gharibi M, Boutaba R, Waslander SL. **Internet of Drones.** IEEE Access. 2016;4:1148–62.
- [3] Ronaldo F, Pramadihanto D, Sudarsono A. **Secure Communication System of Drone Service using Hybrid Cryptography over 4G/LTE Network.** In: 2020 International Electronics Symposium (IES) [Internet].

- Surabaya, Indonesia: IEEE; 2020 [cited 2021 Dec 1]. p. 116–22. Available from: <https://ieeexplore.ieee.org/document/9231951/>
- [4] H. Kopetz, P. Puschner. **Real-Time Communication [Internet]**. Insitute of Computer Engineering - TU WIEN Informatics; 2017 [cited 2020 Sep 26]. Available from: <https://ti.tuwien.ac.at>
- [5] Stewart R. **Stream Control Transmission Protocol [Internet]**. RFC Editor; 2007 Sep [cited 2021 Dec 1] p. RFC4960. Report No.: RFC4960. Available from: <https://www.rfc-editor.org/info/rfc4960>
- [6] B. A. Forouzan. **Stream Control Transmission Protocol (SCTP)**. In: TCP/IP PROTOCOL SUITE, FOURTH EDITION. New York, America: McGraw-Hill; 2010. p. 502–38.
- [7] Wiss T, Forsstrom S. **Feasibility and performance evaluation of SCTP for the industrial internet of things**. In: IECON 2017 - 43rd Annual Conference of the IEEE Industrial Electronics Society [Internet]. Beijing: IEEE; 2017 [cited 2021 Dec 1]. p. 6101–6. Available from: <http://ieeexplore.ieee.org/document/8217060/>
- [8] Sun W, Yu S, Xing Y, Zhang D. **A Multi-path Switching Method Based on SCTP for Heterogeneous Wireless Networks in Smart IoT**. In: 2018 IEEE International Conference on Smart Internet of Things (SmartIoT) [Internet]. Xi'an: IEEE; 2018 [cited 2021 Dec 1]. p. 15–22. Available from: <https://ieeexplore.ieee.org/document/8465519/>
- [9] Vivekananda GN, Reddy PC. **PERFORMANCE EVALUATION OF TCP, UDP, AND SCTP IN MANETS**. 2018;13(9):6.
- [10] Lai WK, Jhan J-J, Li J-W. **A Cross-Layer SCTP Scheme With Redundant Detection for Real-Time Transmissions in IEEE 802.11 Wireless Networks**. IEEE Access. 2019;7:114086–101.
- [11] Stewart R, Ramalho M, Xie Q, Tuexen M, Conrad P. **Stream Control Transmission Protocol (SCTP) Partial Reliability Extension [Internet]**. RFC Editor; 2004 May [cited 2021 Dec 1] p. RFC3758. Report No.: RFC3758. Available from: <https://www.rfc-editor.org/info/rfc3758>
- [12] Jungmaier A, Rescorla E, Tuexen M. **Transport Layer Security over Stream Control Transmission Protocol [Internet]**. RFC Editor; 2002 Dec [cited 2021 Dec 1] p. RFC3436. Report No.: RFC3436. Available from: <https://www.rfc-editor.org/info/rfc3436>
- [13] Tuexen M, Seggelmann R, Rescorla E. **Datagram Transport Layer Security (DTLS) for Stream Control Transmission Protocol (SCTP) [Internet]**. RFC Editor; 2011 Jan [cited 2021 Dec 1] p. RFC6083. Report No.: RFC6083. Available from: <https://www.rfc-editor.org/info/rfc6083>
- [14] Bellovin S, Ioannidis J, Keromytis A, Stewart R. **On the Use of Stream Control Transmission Protocol (SCTP) with IPsec [Internet]**. RFC Editor; 2003 Jul [cited 2021 Dec 1] p. RFC3554. Report No.: RFC3554. Available from: <https://www.rfc-editor.org/info/rfc3554>
- [15] Tuexen M, Stewart R, Lei P, Rescorla E. **Authenticated Chunks for the Stream Control Transmission Protocol (SCTP) [Internet]**. RFC

- Editor; 2007 Aug [cited 2021 Dec 1] p. RFC4895. Report No.: RFC4895. Available from: <https://www.rfc-editor.org/info/rfc4895>
- [16] Hasselstro N, Hjern G, Hoorn R, Hult M, Syre J, Alfredsson S, et al. **The Design, Implementation, and Performance Evaluation of Secure Socket SCTP 2.0**. Sci Technol. :52.
- [17] Dutta IK, Ghosh B, Bayoumi M. **Lightweight Cryptography for Internet of Insecure Things: A Survey**. In: 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC) [Internet]. Las Vegas, NV, USA: IEEE; 2019 [cited 2021 Dec 1]. p. 0475–81. Available from: <https://ieeexplore.ieee.org/document/8666557/>
- [18] Mallouli F, Hellal A, Sharief Saeed N, Abdulraheem Alzahrani F. **A Survey on Cryptography: Comparative Study between RSA vs ECC Algorithms, and RSA vs El-Gamal Algorithms**. In: 2019 6th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/ 2019 5th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom) [Internet]. Paris, France: IEEE; 2019 [cited 2021 Dec 1]. p. 173–6. Available from: <https://ieeexplore.ieee.org/document/8854027/>
- [19] Purnamasari DN, Sudarsono A, Kristalina P. **Medical Image Encryption Using Modified Identity Based Encryption**. Emit Int J Eng Technol [Internet]. 2019 Dec 1 [cited 2022 Jun 2];7(2). Available from: <http://emitter.pens.ac.id/index.php/emitter/article/view/405>