Design of a Progressive MAC for Drone RC Data Protection

Joungil Yun, Seungyong Yoon, Byoungkoo Kim, Daewon Kim, Yousung Kang
Cyber Security Research Division
Electronics and Telecommunications Research Institute (ETRI)
Daejeon, Republic of Korea
{sigipus, syyoon, bkkim05, dwkim77, youskang}@etri.re.kr

***Abstract*—This paper proposes a progressive message authentication code (MAC) scheme for securing remote control (RC) command data in drone systems. The scheme aims to ensure data integrity under unstable wireless conditions while reducing communication overhead for short-length messages that are frequently transmitted. To improve upon conventional MAC aggregation techniques, we introduce a modified structure that not only incorporates controlled inter-message dependencies but also minimizes inter-bit dependency overlap by employing randomized bit selection. This design is intended to enhance resilience against message loss while progressively strengthening authentication guarantees with ongoing data reception.**

Keywords— drone RC security, MAC aggregation, progressive MAC

I. Introduction

With the advancement of drone technology, the application of unmanned aerial vehicles (UAVs) has rapidly expanded across military, industrial, and civilian domains. Consequently, ensuring the security of drone systems—particularly the protection of RC command data—has become a critical requirement. RC commands are typically transmitted as short, real-time data streams over unstable and interference-prone wireless channels, necessitating the use of efficient and lightweight authentication mechanisms to ensure message integrity.

Authenticated Encryption with Associated Data (AEAD) is widely used to provide both confidentiality and integrity, and is applicable to constrained environments [1]. However, the 128-bit authentication tags it produces are considered excessive for short control messages, introducing significant communication overhead and latency—both of which are unsuitable for time-sensitive drone operations.

One approach to reducing this overhead involves using truncated MACs that produce 32-bit or 16-bit tags. These shorter tags provide benefits in reducing bandwidth consumption and transmission latency, which are important for real-time drone control. However, this efficiency comes at the cost of weaker security, significantly lowering brute-force resistance and increasing vulnerability to forgery and message loss in unreliable wireless environments.

In this paper, we propose a progressive authentication framework that balances efficiency and security for short-length, streaming control data, such as that used in drone systems. Our approach extends existing g-Sidon-set-based MAC aggregation techniques as proposed in [2] by minimizing inter-bit dependency overlap and introducing a tunable parameter e to control message index redundancy across dependency sets. This results in a flexible structure that limits the impact of message loss on overall tag validity and improves resilience under unreliable wireless conditions.

II. Previous Works

Various authentication strategies have been developed to secure real-time streaming data with minimal overhead. In particular, MAC aggregation techniques have been introduced to reduce transmission overhead while ensuring data integrity and authenticity under constrained conditions.

One fundamental approach is the compound MAC scheme [3], which aggregates multiple messages into a batch and generates a single authentication tag for the entire group. While this method reduces computational cost and the number of transmitted tags, it defers verification until all messages in the batch are received. As a result, it is unsuitable for real-time applications and is highly vulnerable to message loss.

To overcome these limitations, Progressive Message Authentication Codes (ProMACs) [4] were proposed. ProMACs adopt a progressive authentication strategy, beginning with minimal assurance and strengthening it as more messages are received. This method enables real-time verification and provides partial assurance of integrity even under moderate message loss. However, ProMACs are vulnerable to structural attacks, such as sandwich attacks, and their sliding-window-based dependency model requires continuous message reception to maintain full security.

In response to these structural weaknesses, the Randomized and Resilient Dependency Distributions (R2-D2) scheme [2] was introduced. R2-D2 utilizes randomized dependency mapping based on mathematical structures such as Golomb rulers or g-Sidon sets to evenly distribute dependency relationships, thereby reducing the likelihood of multiple authentication bits depending on the same message. Immediate protection bits—tag components relying solely on the current message—are introduced to preserve minimum security even during message loss or interference.

III. Proposed Scheme

The R2-D2 scheme improved ProMACs by introducing randomized, distributed dependencies to mitigate vulnerability from concentrated dependency. By using g-Sidon sets, it limited repeated differences within dependency sets, thereby reducing excessive reliance on any single message. However, g-Sidon sets alone cannot completely prevent overlapping dependencies, which may result in multiple tag bits depending on the same message and thus increase the impact of message loss.

To address this issue, we propose an enhanced dependency structure called the (g, e)-Sidon set, which introduces an additional parameter e to limit the overlap of elements across dependency sets. This enhancement is designed to improve the robustness of the authentication mechanism by reducing dependency overload and increasing tolerance to message loss.

In practice, the (g,e)-Sidon set is constructed through a parameter-guided search process defined by four parameters:

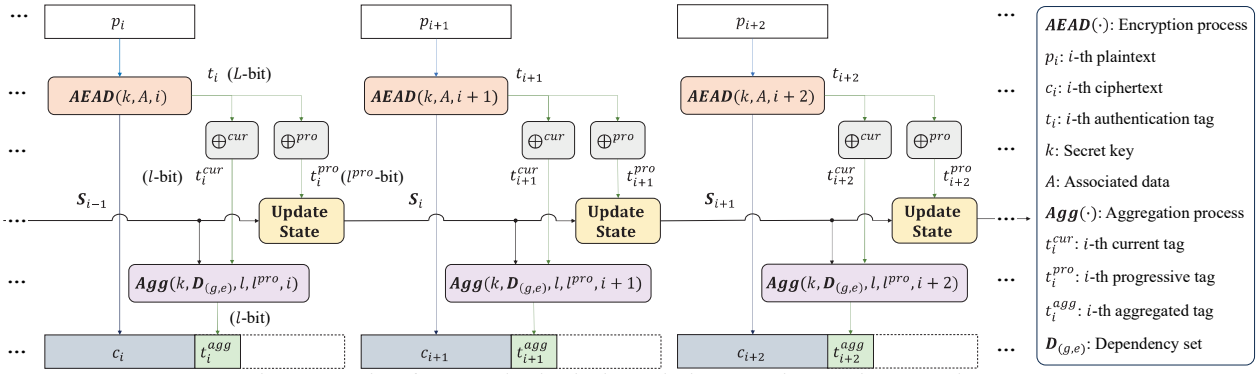

Fig. 1. Generation of aggregated authentication tag in the proposed progressive MAC scheme.

g, e, n, and N. These parameters enable flexible trade-offs among security strength, latency, and resilience. Their roles are as follows:

- g specifies the maximum number of times a particular difference between elements is permitted within a single dependency set. For example, when g is set to 1, no identical differences are allowed, ensuring that dependency distances are evenly distributed, which matches the original purpose of g in g-Sidon set.
- e defines the maximum number of times a specific message index may appear across all dependency sets. For example, if e is set to 2, each message index may participate in at most two different sets. This limits the influence of any single message on the overall authentication structure and localizes the security impact of message loss, thereby enhancing resilience.
- n determines the number of messages involved in computing each authentication bit, i.e., the size of each dependency set. A larger n improves security by increasing structural complexity but also increases computational and memory overhead.
- N is associated with the maximum range of message indices considered when generating the (g,e)-Sidon set used to construct candidate dependency sets. A larger N enables more diverse combinations and increases the likelihood of satisfying the given g, e, and n. As the (g,e)-Sidon set can be precomputed and shared in advance between the sender and receiver, N does not impact runtime computation.

Fig. 1 illustrates the overall process of the proposed progressive message authentication framework based on the (g,e)-Sidon set. When AEAD encryption is applied to the current plaintext message p_i, it yields a ciphertext c_i and an authentication tag t_i of length L bits (typically L=128). This tag is then condensed through a MAC aggregation process into a shorter tag t_i^{agg}.

In the proposed scheme, the aggregated tag t_i^{agg} consists of two components: immediate protection bits derived directly from the current tag t_i^{cur}, and progressive authentication bits computed using previously stored progressive tags t_{i-d}^{pro} from the state $\boldsymbol{S}_{i-1}$. This structure supports partial verification even in the presence of message loss and progressively strengthens authentication as more messages are received.

Let the length of t_i^{agg} be l, and the portion dedicated to progressive authentication be $l^{pro}(< l)$. Two types of condensed tags are computed as follows:

$$t_i^{cur} = \oplus^{cur} = \oplus_{0 \le k < \lfloor L/l \rfloor} [t_i]_{k \cdot l+1:(k+1) \cdot l} \quad (1)$$

$$t_i^{pro} = \oplus^{pro} = \oplus_{0 \le k < \lfloor L/l^{pro} \rfloor} [t_i]_{k \cdot l^{pro}+1:(k+1) \cdot l^{pro}} \quad (2)$$

Here, the operator $\oplus_k$ denote the bitwise XOR operation over the specified bit blocks, and $[X]_{a:b}$ represents the bit segment of the X from bit position a to b, inclusive. L is the bit length of the AEAD tag t_i, while l and l^{pro} are the bit length of t_i^{cur} and t_i^{pro}, respectively. The notation $\lfloor \cdot \rfloor$ indicates the floor function, which returns the greatest integer less than or equal to its argument.

In the **Update State** stage (see Fig. 1), t_i^{pro} is added to the state while the oldest entry is discarded, resulting in an updated state $\boldsymbol{S}_i$. The number of elements retained in $\boldsymbol{S}_i$ is determined by the maximum value among the elements in the dependency set $\boldsymbol{D}$, which will later be selected from the precomputed (g,e)-Sidon set. For example, if the maximum index is v, the states are updated as follows:

$$\boldsymbol{S}_{i-1} = \{t_{i-v}^{pro}, \cdots, t_{i-1}^{pro}\} \rightarrow \boldsymbol{S}_i \ \{t_{i-v+1}^{pro}, \cdots, t_i^{pro}\}$$

If a message cannot be properly received due to interference, adversarial tampering, or other disruptions, a flag value (e.g., -1) is stored to mark the entry as invalid. This mechanism enables efficient identification of missing messages during aggregation and allows unnecessary computations to be skipped.

MAC aggregation is performed by the function $\boldsymbol{Agg}(\cdot)$ shown in Fig. 1. This function takes as input the previously accumulated state $\boldsymbol{S}_{i-1}$ and the current tag t_i^{cur}, and operates based on the candidate of dependency pool $\boldsymbol{D}_{(g,e)}$, drawn from the precomputed (g,e)-Sidon set. And a secret key k and message index i, which are securely shared in advance between the sender and receiver, are used both to deterministically generate the AEAD nonce and to seed the pseudorandom selection of dependency subsets.

The progressive authentication procedure consists of the following steps:

1. Dependency Set Selection: The set of candidate dependency subsets, denoted by $\boldsymbol{D}_{(g,e)}$, is derived from the precomputed (g,e)-Sidon set and used as the pool for constructing actual dependency set. To ensure sufficient diversity for progressive authentication, the total number of available subsets

in (g,e)-Sidon set must exceed l^{pro}. For memory efficiency, $\boldsymbol{D}_{(g,e)}$ may be composed of subsets with smaller maximum element values.

2. Randomized Subset Selection: The sender and receiver use a shared pseudorandom function seeded by the secret key k to randomly select l^{pro} subsets without duplication. These selected subsets form the dependency set $\boldsymbol{D} = \{\boldsymbol{D}_1, \dots, \boldsymbol{D}_{l^{pro}}\}$, which remains fixed as long as k is unchanged.
3. Per-Message Randomization: As required for AEAD encryption and decryption synchronization, a per-message nonce shared between the sender and receiver is used as the seed for a pseudorandom permutation. The permutation reorders the bits of t_i^{cur} to produce $t_i'^{cur}$.
4. Final Tag Aggregation: Using the permuted tag $t_i'^{cur}$ and the previously accumulated progressive tags in $\boldsymbol{S}_{i-1}$, the final aggregated tag t_i^{agg} is computed as follows:

$$\left[t_i^{agg}\right]_m = \begin{cases} \bigoplus_{d \in \boldsymbol{D}_m} \delta(m,d) & \text{for } 1 \le m \le l^{pro} \\ \left[t'^{cur}_i\right]_m & \text{otherwise} \end{cases} \quad (3)$$

$$\text{where } \delta(m,d) = \begin{cases} \left[t'^{cur}_i\right]_m & \text{if } d = 0 \\ \left[t_{i-d}^{pro}\right]_m & \text{otherwise} \end{cases}$$

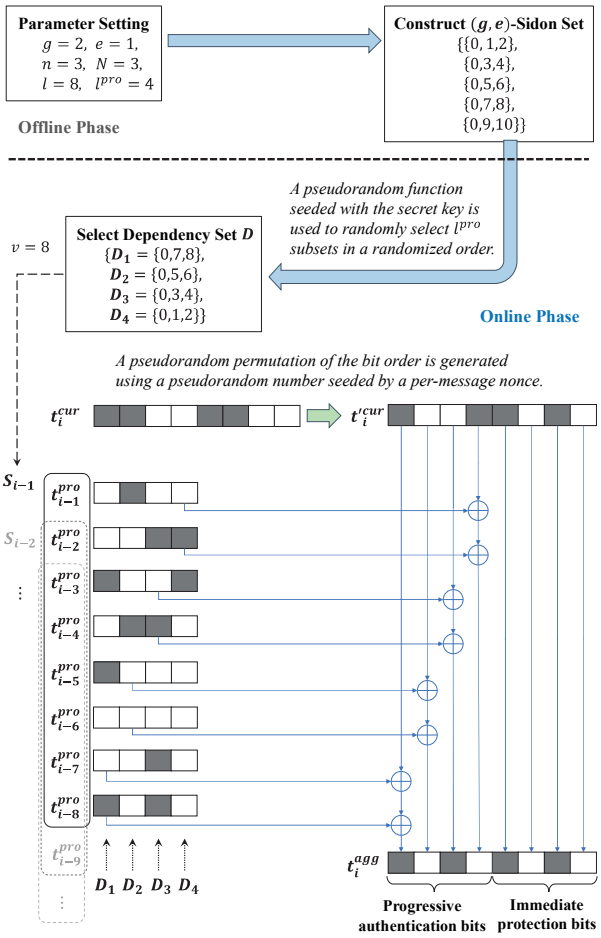

Fig. 2. Example of Progressive MAC construction using (g,e)-Sidon-based dependencies and accumulated authentication states.

Here, $[X]_a$ denotes the single bit at position a. In t_i^{agg}, the first l^{pro} bits correspond to progressive authentication, while the remaining bits provide immediate protection.

5. Verification under Partial Authentication: Before evaluating Eq. (3), any dependency on invalid entries flagged in $\boldsymbol{S}_{i-1}$ is excluded from computation. The corresponding bit positions in t_i^{agg} are treated as 'don't care' and excluded from tag verification. The security threshold is then adjusted to reflect the reduced number of contributing authentication bits.

Fig. 2 illustrates the tag aggregation process of out proposed Progressive MAC scheme. It depicts the overall procedure of generating an aggregated tag, including the construction of the (g,e)-Sidon-set-based on parameter setting, the selection of the dependency set, and the determination of the accumulated state length. It also shows how the permuted current tag and progressive tags from selected prior messages are combined to produce the final aggregated tag.

IV. Conclusions

This paper presented a progressive message authentication scheme tailored for real-time, short-length control data in constrained wireless environments such as drone RC systems. The proposed method extends existing MAC aggregation approaches by employing a (g,e)-Sidon-based dependency structure that enables control over both inter-bit redundancy and message index overlap. The scheme separates authentication bits into immediate and progressive components, enabling authentication to be progressively reinforced over time while preserving minimal verification capability even under message loss.

Future work will involve implementing the proposed scheme in practical communication environments and evaluating its performance under realistic wireless conditions. Particular attention will be given to evaluating the trade-offs among security, latency, and resource efficiency in the context of drone control systems.

Acknowledgment

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.RS-2023-00225201, Development of Control Rights Protection Technology to Prevent Reverse Use of Military Unmanned Vehicles).

References

[1] J. Yun, S. Yoon, B. Kim, and Y. Kang, "Applying lightweight cryptography to enhance drone RC security," in Proc. 15th IEEE Int. Conf. Information and Communication Technology Convergence (ICTC), pp. 1367-1368, Oct. 2024.

[2] E. Wagner, J. Bauer, and M. Henze, "Take a bite of the reality sandwich: revisiting the security of progressive message authentication codes," in Proc. 15th ACM Conf. Security and Privacy in Wireless and Mobile Networks (WiSec), pp. 207-221, May 2022.

[3] D. K. Nilsson, U. E. Larson, and E. Jonsson, "Efficient in-vehicle delayed data authentication based on compound message authentication codes," in Proc. 68th IEEE Vehicular Technology Conference (VTC), pp. 1-5. Sep. 2008.

[4] F. Armknecht, P. Walther, G. Tsudik, M. Beck, and T. Strufe, "ProMACs: progressive and resynchronizing MACs for continuous efficient authentication of message streams," in Proc. 2020 ACM SIGSAC Conf. Computer and Communications Security (CCS), pp. 211-223, Nov. 2020.