

Take a Bite of the Reality Sandwich: Revisiting the Security of Progressive Message Authentication Codes

Eric Wagner
eric.wagner@fkie.fraunhofer.de
Fraunhofer FKIE
RWTH Aachen University

Jan Bauer
jan.bauer@fkie.fraunhofer.de
Fraunhofer FKIE

Martin Henze
henze@cs.rwth-aachen.de
RWTH Aachen University
Fraunhofer FKIE

ABSTRACT

Message authentication guarantees the integrity of messages exchanged over untrusted channels. However, to achieve this goal, message authentication considerably expands packet sizes, which is especially problematic in constrained wireless environments. To address this issue, *progressive* message authentication provides initially reduced integrity protection that is often sufficient to process messages upon reception. This reduced security is then successively improved with subsequent messages to uphold the strong guarantees of traditional integrity protection. However, contrary to previous claims, we show in this paper that existing progressive message authentication schemes are highly susceptible to packet loss induced by poor channel conditions or jamming attacks. Thus, we consider it imperative to rethink how authentication tags depend on the successful reception of surrounding packets. To this end, we propose R2-D2, which uses randomized dependencies with parameterized security guarantees to increase the resilience of progressive authentication against packet loss. To deploy our approach to resource-constrained devices, we introduce SP-MAC, which implements R2-D2 using efficient XOR operations. Our evaluation shows that SP-MAC is resilient to sophisticated network-level attacks and operates as resources-conscious and fast as existing, yet insecure, progressive message authentication schemes.

CCS CONCEPTS

• **Security and privacy → Cryptanalysis and other attacks**; **Hash functions and message authentication codes**.

KEYWORDS

ProMACs, Progressive Authentication, Cyber-Physical Systems

ACM Reference Format:
Eric Wagner, Jan Bauer, and Martin Henze. 2022. Take a Bite of the Reality Sandwich: Revisiting the Security of Progressive Message Authentication Codes. In *Proceedings of the 15th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec '22), May 16–19, 2022, San Antonio, TX, USA.* ACM, New York, NY, USA, 16 pages. https://doi.org/10.1145/3507657.3528539

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
WiSec '22, May 16–19, 2022, San Antonio, TX, USA
© 2022 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-9216-7/22/05…$15.00
https://doi.org/10.1145/3507657.3528539

1 INTRODUCTION

Message authentication enables a recipient to verify that a message stems from the claimed sender [9]. The most prominent and widely-used approaches for such verification are message authentication codes (MACs), which are used to add authentication tags (short: *tags*) to each message [23]. Naturally, these tags expand packets: To achieve the minimum security level of 128 bit recommended by NIST [46], 16 bytes have to be added to a message. In resource-constrained environments, *e.g.*, industrial control systems [19, 29, 55]with messages as small as a single byte [20], a large fraction of a packet is thus consumed by this tag. In these environments, the expansion of packets is problematic because of (i) payloads that are only a few byte long [34], (ii) bandwidth limitations [22], (iii) energy restrictions [13], and (iv) reliability requirements [55].

To overcome this obstacle, early work proposed to reduce tag sizes by truncating tags [50, 54, 59, 66] or aggregating tags of multiple messages [10, 17, 21, 32, 35]. These reductions, however come at the cost of reduced security or intolerable delays. *Progressive* message authentication codes (ProMACs) [5, 36, 37, 53] address these drawbacks by immediately providing reduced integrity protection upon reception of a message, which is then progressively reinforced by subsequent messages, eventually achieving "full" security. Often, the initial protection suffices to optimistically process messages since the system could recover from the unlikely scenario of a retrospectively (within seconds) detected attack [11, 41, 59, 61].

While ProMACs promise to provide strong and low-latency integrity protection even over wireless and other lossy channels, we show that this is not the case for current ProMAC schemes: Random transmission failures or a network-level adversary, *i.e.*, an attacker with the ability to drop or alter packets, can deliberately remove the integrity protection from a complete sequence of messages by interfering with only two carefully chosen packets. Thus far, ProMACs do not consider the collateral damage of packet loss, *i.e.*, the impact of lost packets on the verifiability of surrounding tags and attacks emerging from it. To address resulting vulnerabilities, it is imperative to decouple the dependency of tags on a sequence of directly subsequent messages to prevent attackers from voiding the integrity of messages by interfering with a few selected packets.

Contributions. To enable the *secure* utilization of progressive message authentication, we make the following contributions:

- We show that current ProMAC schemes are prone to a *sandwich attack*, where an adversary selectively attacks the two messages surrounding a message sequence to remove integrity protection of the complete sequence (Section 3).
- To increase ProMACs' resilience to such attacks as well as transmission failures, we propose *R2-D2*, our generic solution to decouple the dependency of tags on a direct message sequence. R2-D2

builds on the properties of Golomb Rulers to achieve optimal verification delays for predefined security guarantees, which it pairs with randomness and immediate protection bits to achieve *randomized and resilient dependency distribution* (Section 4).
- We design and implement *SP-MAC*, a ProMAC scheme for resource-constrained devices that realizes R2-D2's mitigations using efficient XOR operations. Our evaluation shows that SP-MAC effectively protects against network-level interference while operating as resource-conscious as current ProMACs (Section 5).

Availability Statement. The source code underlying this paper is available at: https://github.com/fkie-cad/spmac

2 PROGRESSIVE AUTHENTICATION

A major challenge of secure communication in resource-constrained scenarios stems from the overhead of integrity protection: Even the tiniest message requires a tag of several bytes (*e.g.*, 16 bytes for 128-bit security), thus significantly increasing messages sizes. In this section, we motivate the core idea of ProMACs, a recent proposal to address this issue (Section 2.1), together with a motivating example (Section 2.2). Afterward, we formally introduce ProMACs (Section 2.3) and present three practical implementations (Section 2.4).

2.1 Core Idea and Benefits of ProMACs

Traditional MACs (*e.g.*, HMAC [8]) occupy large parts of the total payload for short messages. To partly mitigate this issue, tags can be truncated at the cost of reduced security [50]. To provide short tags with strong security guarantees, the core idea behind ProMACs is to partially offload integrity protection into the future. Therefore, each message is initially only protected with a reduced security level, similar to truncated MACs, but subsequent messages quickly increase this protection to an adequate level (*e.g.*, 128 bit). Since tags aggregate the protection of multiple messages at once, ProMACs realize short tags, while enabling passive resynchronization if packets get lost. This passive resynchronizability is in stark contrast to previous proposals for short tags, such as aggregated MACs, which jointly authenticate multiple messages with a single tag, and stateful MACs, which continuously reinforce the integrity of all previously sent messages but cannot cope with packet loss.

Consequently, ProMACs are proposed for various (wireless) scenarios such as vehicular communication [5, 36, 37, 45, 53], (industrial) IoT [5, 36, 37], drone control [5, 7], and internal communication within hardware components (*e.g.*, Intel SGX or SoCs) [5, 26, 42]. To cope with the low latency requirements of those scenarios, ProMACs rely on optimistic security, which (partly) defers security processing into the future and allows a system to continue under the assumption that all traffic is benign. In the unlikely event that an attack is detected retrospectively, the system recovers from already processed malicious messages. Optimistic security is especially attractive in isolated networks where attacks are relatively rare and the potential damage in a short time frame is comparable to that of less advanced attacks (*e.g.*, denial of service) [11, 41, 58–61, 72].

2.2 A Motivating Example for ProMACs

To illustrate how ProMACs' benefits manifest themselves in practice, we consider a comprehensive example from an Industrial Control System (ICS). In particular, we envision a closed-loop motion controller that reacts to continuously updated sensor readings [2]. Especially for moving systems, wireless, and thus unreliable, communication is used to avoid error-prone and expensive cable management [2]. While such a controller is resilient to the immediate impact of individual faulty sensor readings, one or multiple maliciously crafted messages can, over time, bring the system into an equipment-damaging or even life-threatening state. Meanwhile, communication channels between sensors and controllers are often constrained due to *e.g.*, a high number of network participants.

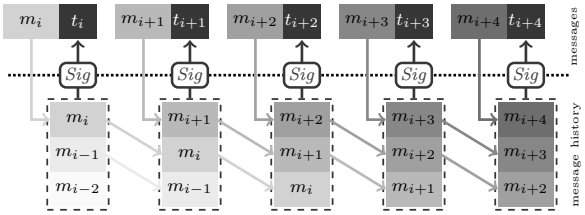


Figure 1: ProMACs store a history of recent messages, which is used to derive tags, effectively aggregating integrity protection of multiple messages to reduce tag sizes.

While bandwidth constraints prevent the traditional protection of each message with a 16 byte long tag, using a truncated (*i.e.*, less secure) tag hampers the reliable detection of manipulations. Therefore, an attacker could manipulate messages with long-term impact, *e.g.*, a scaling factor for speed adjustments, and thus bring the system into a critical state. While aggregated MAC schemes could eventually ensure integrity with high confidence, they would give an attacker the opportunity to manipulate multiple messages before any authenticity is verified. ProMACs mitigate this weakness by providing, albeit reduced, immediate security. ProMACs thus protect against the immediate impact of manipulated messages, while also protecting against manipulations with long-term impact.

2.3 Formal Definition of ProMACs

To explain how ProMACs realize efficient protection, we formally introduce them based on traditional MACs.

Traditional MACs. A MAC scheme allows two communication partners to authenticate exchanged messages using a pre-shared secret k. To authenticate a message m, the sender uses the tag generation algorithm $Sig_k(m)$ to generate the corresponding authenticity tag t. Upon reception of a message, the verification algorithm $Vrfy_k(m, t)$ enables the recipient to evaluate whether the received tag is valid. This verification is done by computing the tag for the received message m and comparing it to the received tag t. A MAC scheme is considered secure if it is computationally infeasible to generate a (m,t)-pair that would be accepted by $Vrfy_k$, without knowing the secret k. This requirement can, *e.g.*, be achieved by using keyed hash functions such as `HMAC-SHA256` to compute t.

ProMACs. ProMACs extend traditional MACs by additionally giving recent historical messages as input to $Sig_k(m_1, \ldots, m_n)$ and $Vrfy_k(m_1, \ldots, m_n, t)$. As shown in Figure 1, the generation of tags is then based on *multiple* messages. The tag t_{i+2}, for instance, is computed from messages m_{i+2}, m_{i+1}, and m_i. Likewise, the integrity of message m_{i+2} is protected by tags t_{i+2}, t_{i+3}, and t_{i+4}. Thus, ProMACs protect each message with multiple tags, which means that

each tag is only responsible for providing a fraction of the overall targeted security level. Since each tag aggregates partial integrity protection for multiple messages, progressive integrity protection results in shorter tags. Meanwhile, a valid first tag (to the degree it can be verified) is considered sufficient to optimistically process a message, while a recovery mechanism is triggered if an attack is detected within subsequent tags. In this context, the dependencies $\mathcal{D}$ ($\{0\} \subseteq \mathcal{D} \subset \mathbb{N}_0$) describe how the reception of one message influences the authenticity of surrounding messages. We say that a ProMAC instance has the dependencies $\mathcal{D}$, if the generation and verification of tag t_i require knowledge of $\{m_{i-d} | d \in \mathcal{D}\}$. Consequently, a message m_i blends into all tags $\{t_{i+d} | d \in \mathcal{D}\}$ and a tag t_i protects the integrity of all messages $\{m_{i-d} | d \in \mathcal{D}\}$.

2.4 Existing ProMAC Schemes

Three distinct approaches realize the theoretical notion of ProMACs: *Whips* [5], CuMAC [36, 37], and *Mini-MAC* [53]. Our discussion of these approaches focuses on their selection of identical dependencies $\mathcal{D}$, as those describe how the failure to receive one message influences the verifiability of neighboring tags.

Whips *(CCS'20).* Whips [5] was proposed alongside the formal introduction of ProMACs. It provides a fixed security level of 128 bit with a constant memory overhead per message stream. To this end, Whips tracks the message history via an internal state s, used to derive tags t, that is composed of a counter c (for replay protection) and a fixed number n of substates $\tilde{s}$. The number of substates is inversely proportional to the targeted tag lengths, such that if smaller tags are used, a message's integrity is protected by more tags. Each substate $\tilde{s}_i$ corresponds to exactly one message and is computed as $\tilde{s}_i = trunc(\mathsf{HMAC}_k(m_i))$. The size of $\tilde{s}$ depends on the targeted security level (*e.g.*, $\tilde{s}$ has to be at least 32 byte long for 128-bit security [5]). To generate a new tag t_i for a message m_i, the state s_i is first updated by (i) incrementing the counter c, (ii) appending the substate $\tilde{s}_i$ to s, and (iii) removing the substate $\tilde{s}_{i-n}$ from s. The tag t_i for message m_i is then computed as $trunc(\mathsf{HMAC}_k(s_i))$. Since a tag thus depends on the last $n-1$ messages, Whips relies on sliding window-based dependencies, *i.e.*, $\mathcal{D} = \{0, \ldots, n-1\}$.

CuMAC *(CNS'20 & IoT-J'21).* Simultaneous to the formalization of ProMACs [5], CuMAC [37] proposed the similar concept of cumulative message authentication codes. In CuMAC, first, a traditional MAC σ is computed from a counter c and a message m. Then, σ is split into n fragments, *i.e.*, $\sigma = \sigma^0 || \ldots || \sigma^{n-1}$. Finally, the tag t_i for message m_i is computed by aggregating fragments of the MAC σ for the n past messages using XOR (one distinct fragment per message). More precisely, $t_i = \sigma_i^0 \oplus \sigma_{i-1}^1 \oplus \cdots \oplus \sigma_{i-n}^{n-1}$. Thus, CuMAC also relies on a sliding window of the n most recent messages ($\mathcal{D} = \{0, \ldots, n-1\}$). To further improve CuMAC, CuMAC/S [36] tries to predict future messages and pre-authenticates these messages to achieve immediate full authentication upon message reception. However, this does not change its dependencies $\mathcal{D}$.

Mini-MAC *(Veh. Comm.'17).* Mini-MAC [53] re-authenticates CAN bus messages within subsequent messages to address the problem of insufficient payload size. Although originally not designed as generally applicable, retrospectively Mini-MAC can be interpreted as ProMAC scheme if we ignore optional extensions. Mini-MAC derives a tag t_i (for message m_i) from a sliding window

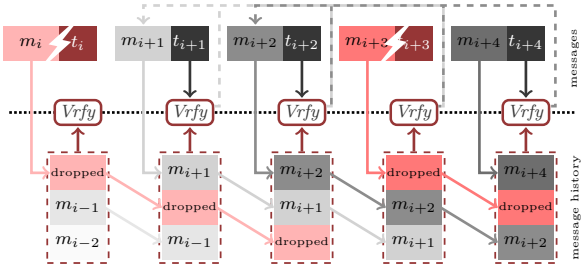


Figure 2: Sliding window-based dependencies, as used by all current state-of-the-art ProMACs schemes, allow an attacker to remove integrity protection from multiple messages by sandwiching them between dropped packets.

of the n most recent messages ($\mathcal{D} = \{0, \ldots, n-1\}$) and a counter c (for replay protection): $t_i = trunc(\mathsf{HMAC}_k(c \| m_{i-(n-1)} \| \ldots \| m_i))$. The size of the sliding window n is not fixed. A larger n results in higher eventual security, but also requires more computations and increases the impact of transmission failures. Additionally, t_i is truncated to the space remaining in the given packet. Consequently, Mini-MAC provides integrity protection in a best-effort manner.

3 SECURITY CONSIDERATION FOR PROMACS

Security of ProMACs so far centered around an attacker with the same goal and means as for traditional MACs, *i.e.*, attacking individual packets by guessing keys or forging tags [5, 36, 37, 53]. In this setting, ProMACs provide at least the same security as traditional MACs: For the latest message, the security of ProMACs is allegedly identical to traditional (truncated) MACs and becomes stronger with subsequent packets [5]. However, these security considerations ignore the impact of *dropped* packets, which influence the verifiability of neighboring tags. Hence, we extend ProMACs' threat model (Section 3.1) and show that this leads to novel attacks (Section 3.2) that severely limit the applicability of current ProMACs.

3.1 Extended Threat Model for ProMACs

To accommodate for ProMACs spreading authenticity over multiple packets, some of which may be lost due to a lossy (*e.g.*, wireless) channel, we extend the original threat model of ProMACs [5] in two ways. Firstly, we extend the attackers' capability beyond simply observing and querying message-tag pairs by giving them the additional capability of inducing and reacting to transmission failures. Secondly, we alter the attackers' goals to include not only the forging of valid tags for a previously unseen message but also the disruption of the communication channel by *e.g.*, amplifying a DoS attack by abusing the characteristics of ProMACs.

3.2 Sandwich Attack Against Current ProMACs

So far, ProMACs did not consider the effects of transmission failures, either caused by a lossy channel or active interference, in their (formal) security proofs [5, 36]. The *sandwich attack* against ProMACs presented in this paper leverages exactly this attack vector: If two transmission failures are less than the tracked message history apart, all messages "sandwiched" between these failures remain unauthenticable. Selective jamming to induce these failures

is, however, hardly distinguishable from random packet loss, such that these attacks are hard to detect as only a small number of dropped packets can have detrimental consequences.

Figure 2 explains the root cause of this attack using an example with a message history of length $n = 3$ (chosen short for illustration). We assume that message m_i is not received, either through a transmission failure or jamming. Then, because of the dependencies $\mathcal{D} = \{k|0 \leq k < n\}$ of current ProMACs, all future tags t_{i+k} $(k < n)$ cannot be verified. In itself, this is not a serious problem, as eventually, messages m_{i+k} $(k < n)$ will still receive (reduced) integrity protection through the tags t_{i+k+j} $(n-k+i<j<\text{n})$. Therefore, in the primarily envisioned scenarios for ProMACs (*cf.* Section 2.1), m_{i+1} and m_{i+2} would be processed optimistically under the assumption that messages stemming from an attacker would be detected before any real damage could occur. However, if another message m_j $(i+1 < j \leq n)$ is also not received (by chance or triggered by interference), then all tags $\{t_i, t_{i+1}, \ldots, t_{j+n-1}, t_{j+n}\}$ cannot be verified. Consequently, all messages m_l $(i < l < j)$, sent in between m_i and m_j, cannot be authenticated, as their integrity protection relies on the tags $\{t_{l+k}|k \in \mathcal{D}\} \subset \{t_i, t_{i+1}, \ldots, t_{j+n-1}, t_{j+n}\}$. Already for our selected short message history, the authenticity of m_{i+1} and m_{i+2} cannot be verified despite being received correctly.

Overall, the prospects of ProMACs to bandwidth-efficiently protect lossy communication are highly desirable. However, their susceptibility to network-level disturbances limits ProMACs' deployability. Looking at the practical consequences of the sandwich attack in Appendix A.1, we further see that the attack is more impactful for shorter tags due to their larger sliding windows. Consequently, those scenarios that benefit most from ProMACs' bandwidth savings are the most vulnerable to the presented sandwich attack.

4 R2-D2: A BASIS FOR SECURE PROMACS

The weakness of current ProMAC schemes stems from an inherent design choice: By spreading a message's integrity protection over *consecutive messages*, these schemes become susceptible to packet loss, where the (malicious) interference on a few packets invalidates the authenticity for multiple messages (*cf.* Section 3). To eliminate this attack vector, it is necessary to *interleave message dependencies* to better distribute the effects of dropped packets to prevent individually lost packets from reducing the protection of targeted messages to insecure levels. However, while the general idea of interleaving message dependencies seems promising, it requires finding the right trade-off between delay for full integrity protection and resilience to dropped packets. To achieve this goal, we propose our Randomized and Resilient Dependency Distribution (R2-D2) as a foundation for ProMACs that are resilient to network-level interference. As shown in Figure 3, R2-D2 is based on *optimally* interleaved dependencies (Section 4.1) and a generalization of this concept to achieve *parameterized security guarantees* (Section 4.2). R2-D2 enhances this foundation through *bit dependencies*, *randomization*, and *immediate protection bits* (Section 4.3).

4.1 Golomb Ruler-based Dependencies

Existing ProMAC schemes rely on a sliding window for their dependency distribution, where each tag's computation requires knowledge of the last n consecutive packets, *i.e.*, $\mathcal{D} = \{0, \ldots, n\}$. However,

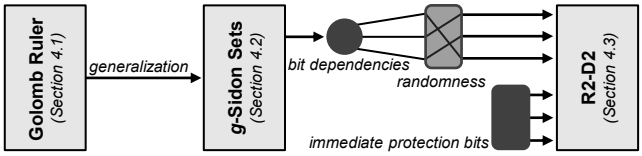


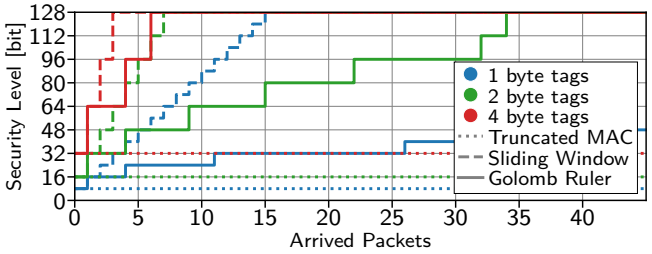
Figure 3: R2-D2 combines different theoretical building blocks to realize randomized and resilient dependency distribution and to thwart sandwich attacks against ProMACs.

exactly this property is abused by the sandwich attack to render the integrity protection of targeted messages void. To mitigate this weakness, ProMACs have to interleave dependencies such that the effects of dropped packets are cushioned by a large set of tags. To achieve this goal, we propose to use *Golomb Rulers* [6, 57], which are used, *e.g.*, in radio astronomy to determine optimal antenna placements [62], to minimize overlap between tags of different messages. Intuitively, a Golomb Ruler is a set of integer marks on a discrete ruler, placed such that the distance between any pair of marks is unique. Formally, a set S ($\{0\} \subseteq S \subset \mathbb{N}_0$) is a Golomb Ruler iff $\forall s_1, s_2, s_3, s_4 \in S$ with $s_1 \neq s_2$ and $s_3 \neq s_4$ it holds that $s_1 - s_2 = s_3 - s_4 \iff s_1 = s_3$ and $s_2 = s_4$. The length of a Golomb Ruler is defined as the value of its largest element.

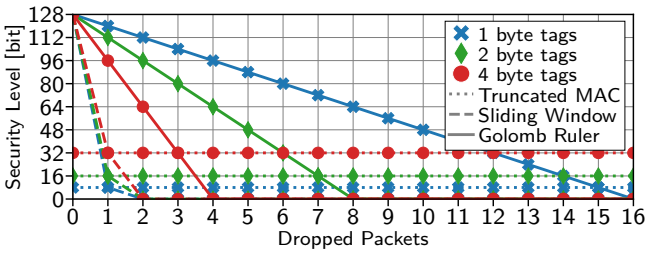
Golomb Rulers provide the theoretical foundation to realize message dependencies that minimize the overlap between tags of neighboring messages. As the distance between two tags protecting a specific message is unique, Golomb Rulers guarantee that a message's security level is reduced by at most the integrity protection provided by one tag for any dropped message (see proof in Appendix A.3.1). Exemplarily, using the Golomb Ruler $\{0, 1, 4, 6\}$ of length 6 ensures that any dropped message only invalidates at most one tag protecting the integrity of any other message, while providing full security guarantee after 6 subsequent messages have been received. Using the shortest Golomb Rulers for a given number of elements, *i.e.*, an optimal Golomb Ruler, as in the previous example, thus provides the mentioned security guarantees within the shortest possible delay until full authenticity is reached.

We now have to investigate what Golomb Ruler-based dependencies mean in general for the security of ProMACs, based on two core metrics: (i) the *delay* to reach full protection, and (ii) the *resilience* to targeted packet dropping. Here, delay is expressed as transmitted packets, as actual time depends on the communication pattern of the underlying application. Resilience is expressed by the minimum obtainable bit security of any message for a given number of lost packets. To express resilience in terms of bit security, we assume that a n-bit tag provides exactly n bits of security, *i.e.*, 128-bit security is realized by appending a 16-byte tag to a message.

In Figure 4, we show the delay and resilience of Golomb Rulers-based ProMACs by comparing them to truncated MACs and sliding windows-based ProMACs for tags that are 1, 2, and 4 byte long. Figure 4a shows how the protection develops over time in absence of an attacker. We observe a fast increase in security for sliding window-based ProMACs and no variation in the provided security over time for truncated MACs. Using optimal Golomb Rulers as dependencies, the delay until full security is reached is acceptable for tag sizes of 4 and 2 byte, whereas 1-byte tags only reach full security after receiving 177 additional messages.



(a) Golomb Ruler-based dependencies require the reception of more messages to achieve full 128-bit security than sliding window approaches, and thus increase the delay of integrity protection in a message stream, particularly for small tags (≤2 byte).



(b) Dropped packets have an obvious impact on the progressive nature of ProMACs. However, the resiliency to dropped messages is significantly increased by Golomb Ruler-based ProMACs compared to current state-of-the-art ProMACs.

Figure 4: Golomb Ruler-based ProMACs (in comparison to sliding window-based ProMACs and truncated MACs) protect against network-level attacks, albeit with a stiff tag length-dependent trade-off between speed and security.

In contrast, Figure 4b shows the resilience of different schemes against network-level attacks. Here, the susceptibility to the sandwich attack of sliding window-based ProMACs can be seen again, as the provided protection of a message can be rendered void with just 2 dropped packets. Truncated MACs are, as expected, not susceptible to network-level attacks. When it comes to the resilience of Golomb Ruler-based ProMACs, we see an inverted behavior as in Figure 4a. 1 and 2 byte long tags provide significant resilience to network-level attacks, remaining well over the security level of truncated MAC, even if a high fraction of the relevant packets are dropped. However, longer tags remain susceptible to variants of the sandwich attack, *i.e.*, the integrity of message protected by 4 byte long ProMACs can be attacked by dropping 4 targeted messages.

Golomb Ruler-based dependencies provably minimize the delay until full security while ensuring that a dropped packet impacts at most one tag protecting any other message. However, the resulting inflexible trade-off between achievable delay and resilience might not match the requirements of specific use cases. We thus discuss how R2-D2 addresses this challenge via generalized Golomb Rulers.

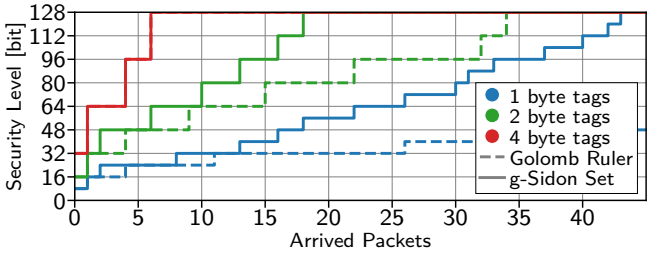
4.2 Tag Length-independent Security Levels

Message dependencies with minimal overlap based on Golomb Rulers directly couple the security loss from a dropped packet to the length (and thus bit security) of individual tags (*cf.* Section 4.1). Thus, while providing optimal dependencies w.r.t. the number of unverifiable tags in any given message, these dependencies may result in unacceptable verification delays for certain scenarios. To resolve this stiff trade-off, we propose *tag length-independent security levels* for providing message dependencies that enable a parametrization of the maximum security loss per dropped packet.

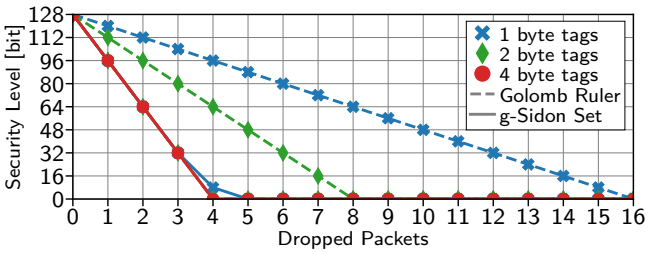
This parametrization enables to define tags of different sizes that each provide similar resilience to network-level attacks. The core idea to achieve tag length-independent security levels in R2-D2 is to give control over how many tags, protecting the integrity of a single message, become at most unverifiable through a dropped packet. To realize this idea, we use *g-Sidon Sets* [57], a generalization of Golomb Rulers. Intuitively, a g-Sidon Set is a set of integer marks on a discrete ruler, which are placed such that the distance between any pair of two marks occurs at most g times. Formally defined, a set $S \subset \mathbb{N}_0$ is a g-Sidon Set iff any pairwise difference between elements occurs at most g times, *i.e.*, S is a g-Sidon Sets iff there exist at most g distinct pairs ($s_0 \in S$, $s_1 \in S$) with $s_0 < s_1$ such that $s_0 - s_1 = k$, for all $k \in \mathbb{Z}^*$. Using g-Sidon Sets as ProMAC dependencies $\mathcal{D}$ guarantees that any message's security level is reduced by at most the integrity protection provided by g tags for any dropped message (see proof in Appendix A.3.2). Thus, Golomb Rulers are 1-Sidon Sets, as any difference between elements in a Golomb Ruler is unique. Overall, these provable and parameterized level of protection against network-level attacks is possible iff g-Sidon Sets are used as dependencies, while their optimality guarantees that full security is achieved in the fastest possible way.

For ProMACs, g-Sidon Sets thus promise to efficiently parameterize the maximum security loss for dropped packets in terms of bit security to decouple this property from the tag length and gain control over verification delays. To verify this claim, we compare verification delays and resilience to network-level attacks of Golomb Ruler-based dependencies to those based on g-Sidon Sets in Figure 5. We choose g such that a dropped message induces at most a 32-bit security loss, independent of the underlying tag size (*i.e.*, $g = 1$ for 4-byte tags, $g = 2$ for 2-byte tags, and $g = 4$ for 1-byte tags). Figure 5a shows the improvements to verification delays based on parameterization. By allowing dependencies to overlap twice, we can nearly half the verification delay of 2-byte tags, and the verification delay of 1-byte tags can be reduced from 177 to 43. However, this speedup also reduces the resilience to attacks, as can be seen in Figure 5b. Here, we show the advantage of g-Sidon Sets-based dependencies since all parameterizations lose security to a similar extent with the number of dropped packets, but this loss is bounded by the maximal security loss of 32 bits per dropped packet. Thus, a variation of the sandwich attack would require the targeted dropping of at least four packets to remove all authenticity.

Dependencies based on g-Sidon Sets thus achieve tag length-independent security levels and allow a flexible parameterization of the trade-off between verification delays and resilience to packet loss. We observe that full security can be provided significantly faster for smaller tags optimal g-Sidon Sets than by Golomb Rulers-based dependencies, which is counteracted by a reduction in resilience to packet loss. Additionally, using optimal g-Sidon Sets lets an attacker know the optimal strategy to remove integrity protection from a targeted message, whereas it would be advantageous



(a) Using g-Sidon Sets instead of Golomb Rulers can significantly reduce the delay until the full security level is reached.



(b) ProMACs based on g-Sidon Sets enable parameterized tag length-independent security loss resulting from dropped packets.

Figure 5: g-Sidon Sets, in contrast to Golomb Rulers, enable to control the security loss per dropped packet (here: 32 bits).

to hide this strategy. In the following, we see how R2-D2 addresses these remaining weaknesses through bit dependencies, hiding of the optimal attack strategy, and immediate protection bits.

4.3 Secure Dependencies through R2-D2

Both, message dependencies with minimal overlap (Section 4.1) and its more flexible generalization for tag length-independent security levels (Section 4.2), realize optimal and thus *deterministic* dependencies for their respective parametrizations. Consequently, while these improved dependencies considerably increase the number of necessary packet drops to disable integrity protection, an attack can still leverage this determinism to derive *which* packets to drop.

R2-D2 addresses this issue by *randomizing* dependencies to hide which messages have to be dropped. Further enhancing this approach, R2-D2 introduces *bit dependencies*, *i.e.*, each bit of a tag protects a different message set. Each R2-D2 instance is initialized with a *pseudorandom* set of dependencies $\mathfrak{D} = \{\mathcal{D}_0, \mathcal{D}_1, \dots\}$, where the number of dependencies equals the tag length, *i.e.*, $|\mathfrak{D}| = |t|$. Each dependency $\mathcal{D}_i$ is a g-Sidon Set, where its order, *i.e.*, number of elements, depends on the tag length $|t|$ such that the total number of dependencies equals the targeted security level, *e.g.*, $\sum_{0 \leq i < |t|} |\mathcal{D}_i| = 128$. The parametrization of g follows from the tolerable security loss (*cf.* Section 4.2). The set of dependencies $\mathfrak{D}$ is pseudorandomly sampled (using a shared key between sender and receiver) from the n precomputed most optimal g-Sidon Sets.

Instead of selecting one of n potential dependencies, the number of potential distributions increases to $\binom{n}{|t|}$ through randomized bit dependencies. This increased variety enables strong resilience with a relatively short n, *e.g.*, 64, meaning that verification delay remains low and that even constrained devices can store the set of potential bit dependencies. Additionally, by using dependencies of different orders, R2-D2 can achieve a specific security level, *e.g.*, 128 bits, even if the targeted tag length does not divide the security level, since the achieved security level amounts to $\sum_{0 \leq i < |t|} |\mathcal{D}_i|$ bits.

As an additional benefit, bit dependencies enable *immediate protection bits*. Those only depend on the current message ($\mathcal{D}_i = \{0\}$) and thus, like truncated MACs, are resilient to network-level interference. This ensures that, no matter how many packets are dropped by an attacker, the protection of a received message is never completely removed. To still reach the targeted bit security level when using immediate protection bits, the order of the remaining dependencies $\mathcal{D}_i \in \mathfrak{D}$ has to be increased accordingly.

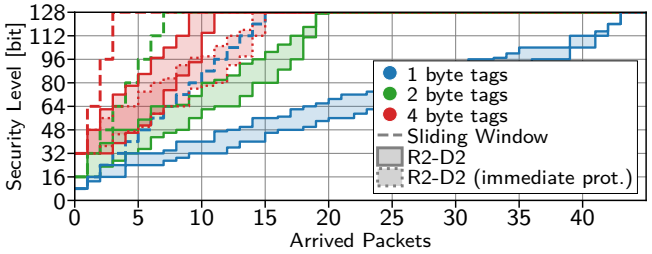
The concepts of randomized bit dependencies and immediate protection bits promise to increase the resilience to network-level attackers without significant impacts on the verification delay. To verify this claim, we again compare the verification delay and resilience of R2-D2 to sliding window-based dependencies, the current state-of-the-art in Figure 6. In addition to the 1, 2, and 4 byte tags with randomized bit dependencies, we also consider the case where half of a 4-byte tag is reserved for immediate protection bits. As before, all tag lengths are parameterized to allow a maximal security loss of 32 bits per dropped packet (further parameters are presented and discussed in Appenix A.4). All dependencies are selected from the 64 shortest ones for a given parametrization, *i.e.*, $n = 64$.

In Figure 6a, the verification delay of R2-D2 constitutes an area between the minimum and maximum delay depending on which dependencies are randomly selected. Overall, we observe similar delays as in Figure 5a. Additionally, in Figure 6b, we show the worst-case resilience of R2-D2 against a network-level attacker. Therefore, we assume that (i) the attacker knows the selected bit dependencies, and (ii) that the most vulnerable dependencies are selected for the most efficient variation of the sandwich attack. We observe that the resilience of R2-D2 increases even in this worst-case scenario through the introduction of bit dependencies, while in reality, the attack would require to drop even more packets as he cannot be sure which packets need to be dropped to execute a sandwich attack. In practice, this protection against targeted packet drops is even higher, since the randomly selected dependencies remain secret.

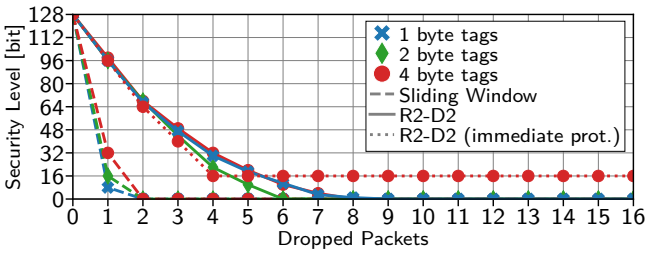
Considering the introduction of immediate protection bits to 4-byte tags, we see in Figure 6a that this addition results in a slight increase in the delay until full protection is achieved. The reason for this additional delay is that to still target 128-bit security, higher-order dependencies have to be chosen because fewer bits are available for progressive authentication. Looking back at Figure 6b, we see that this additional delay creates baseline protection that is not susceptible to network-level attacks, no matter how many packets an attacker can drop, similar to truncated MACs (*cf.* Figure 4b).

4.4 Security Properties of R2-D2

Overall, R2-D2 defines parametrizable dependencies which increase the resilience of ProMACs to packet drops, either through a bad channel or malicious activities (*i.e.*, sandwich attacks and variations thereof). In addition to inheriting the security guarantees of g-Sidon Set-based dependencies, *i.e.*, parameterizable bounds for the maximal security loss per dropped packet, R2-D2 additionally



(a) R2-D2 introduces randomness into its bit dependencies without significantly increasing the delay for full integrity protection.



(b) R2-D2 protects against sandwich attacks, even if the attacker learns the randomized bit dependencies (worst-case assumption).

Figure 6: Even in the worst-case (*i.e.*, the attacker somehow learns the secret bit dependencies), R2-D2 offers high resilience to network-level attacks without significant compromises in terms of verification delay.

hides the optimal attack strategy for an attacker. A big strength of R2-D2 is its flexibility: ProMACs can be adapted to different use cases by choosing trade-offs between the maximal security loss per dropped packet, tag lengths, and acceptable delays until full authenticity can be provided. Finally, R2-D2 achieves provably *optimal* authentication delays for given security parameters, such that we can understand where ProMACs may not be applicable.

Practical Authentication Delays. As ProMACs provide only reduced initial security, messages are processed optimistically and may retrospectively be detected as malicious. Related work on optimistic security [11, 45, 58, 60, 61] puts resulting delays into perspective to understand where ProMACs are applicable. For intra-vehicular communication, Szilagyi *et al.*, *e.g.*, demonstrate that authentication delays of up to 100 messages are acceptable even for throttling control due to high sampling rates and physical inertia [58, 61]. Similarly, Nilsson *et al.* conclude that delays of up to 16 messages are easy to recover from [45]. For ICSs, Castellanos *et al.* show that the optimistic processing of over 100 messages does not significantly impact the systems' state under attack [11] and Szilagyi *et al.* argue that many ICSs can handle malicious packets if detected within 30 messages [60]. While R2-D2 can be parameterized to individual needs, 2 and 4 byte long tags (latter with 16 immediate protection bits) with a maximum security loss of 32 bit are attractive in many real-world deployments: Packets are adequately protected against network-level attacks and reach full security within acceptable delays. However, R2-D2's optimal delays show that shorter tags or strong security guarantees with ProMACs, in general, can only be provided if longer delays are acceptable.

ProMACs on High-Error Rate Channels. R2-D2 lowers the number of ProMAC-protected messages with unverifiable authenticity due to normal packet loss. Yet, we still measure unverifiable authenticity for around 10% of messages (down from 73 % for sliding window-based dependencies as seen in Appendix A.1) for 1-byte tags on a channel with a packet loss of 9.1%. Thus, for adequate security, the use of R2-D2's immediate security bits and slightly longer tags are crucial to realize secure ProMACs for such channels. Meanwhile, for the same 1-byte tags, we did not observe a single message that lost all authenticity over tens of millions of transmitted packets with a packet loss of 0.9%. Thus, for higher reliability channels, R2-D2 makes it unlikely that a processed message has to be reverted because it could not be retroactively authenticated without malicious interference.

5 SP-MAC: A SECURE PROMAC SCHEME

R2-D2 provides the necessary building blocks for efficient ProMACs schemes that are resilient to network-level interference. One of its core features to achieve this protection is a shift from message dependencies to bit dependencies. As this shift fundamentally changes the interplay between tag aggregation and cryptographic operations, current ProMAC schemes (Whips [5], CuMAC [36, 37], and Mini-MAC [53]) cannot easily be retrofitted with R2-D2's improvements. Therefore, we propose a novel ProMAC scheme for *staggered* progressive message authentication codes (SP-MAC) that leverages traditional and secure MACs (*e.g.*, `HMAC-SHA256` [8]) and aggregates these based on R2-D2's secure dependency distribution using efficient XOR operations. As a result, SP-MAC does not only provide built-in protection against network-level attacks but even achieves this as resource-consciously as existing ProMAC schemes. In the following, we introduce SP-MAC (Section 5.1), discuss its security (Section 5.2), and evaluate its performance (Section 5.3).

5.1 Staggered Progressive MACs

Orienting ourselves on the good performance results of CuMAC [36, 37], SP-MAC, in contrast to Whips [5] and Mini-MAC [53], computes tags using an aggregation procedure for traditional MACs instead of defining how tags are directly derived from the recent message history. In a nutshell, SP-MAC thus operates as outlined in Figure 7, where exemplarily 6-bit traditional MACs are first computed and then compressed into 2-bit ProMACs tags.

In more detail, to generate t_i for a message m_i, SP-MAC first computes σ_i as a traditional MAC (*e.g.*, `HMAC-SHA-256-128` [8]) over m_i and a counter ctr, *i.e.*, $\sigma_i = HMAC(m_i, ctr)$. The counter is initialized with 0 and incremented after each message to protect against replay attacks. SP-MAC computes each bit of the tag t_i individually since each bit depends on a unique set of past messages (*cf.* Section 4.3). For pseudorandomly selecting these bit dependencies $\mathfrak{D}$, a pre-shared secret is used. Subsequently, SP-MAC derives t_i from σ_i and from the MACs of messages in the recent past as follows.

In the following, we refer to the j-th bit of t_i as t_i^j, *i.e.*, $t_i = t_i^0 || \dots || t_i^{|t_i|-1}$, where $|t_i|$ denotes the bit-length of t_i. SP-MAC also splits σ_i into its individual bits, by first spitting σ_i into $\lceil |\sigma_i|/|t_i| \rceil$ parts. We then denote $\sigma_i^{a,b}$ as the b-th bit of the a-th part of σ_i, *i.e.*, $\sigma_i^{a,b}$ is the bit at position $(a \cdot |t_i| + b)$ of σ_i. SP-MAC then

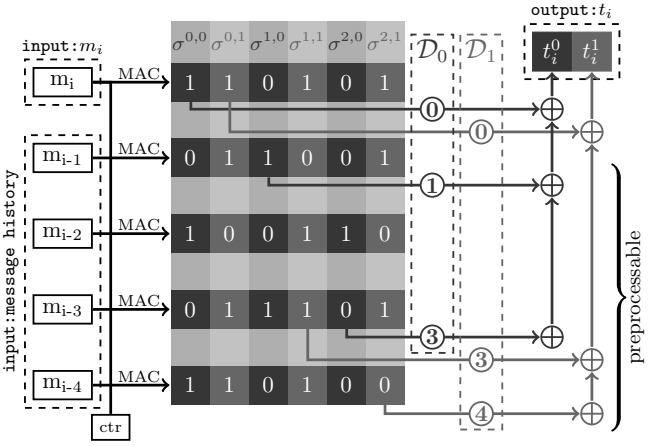


Figure 7: SP-MAC computes traditional MACs (only 6 bits shown here) for a message and derives aggregated and compressed tags through efficient XOR operations.

computes each bit t_i^j of the final tag t_i of message m_i using the bit dependencies $\mathcal{D}_j$ for this bit as follows:

$$t_i^j = \bigoplus_{0 \le n < |\mathcal{D}_j|} \sigma_{i-\mathcal{D}_j[n]}^{n,j}$$

with $\mathcal{D}_j[n]$ representing the n-th entry of $\mathcal{D}_j$. At the start of a message stream, missing values are initialized with 0. Consequently, each bit t_i^j of t_i depends exactly on those messages defined in the corresponding bit dependencies $\mathcal{D}_j \in \mathfrak{D}$ and each bit $\sigma_i^{a,b}$ is included in exactly one tag.

To speed up calculations, SP-MAC partially caches previous messages' tags until they are fully depleted, *i.e.*, all bits have been incorporated into tag computations of subsequent messages. Furthermore, to reduce latency when computing the tag of a certain message, all but one XOR operation (incorporating the bit dependencies of this message into all tag bits at once) can be preprocessed since this processing only relies on bits from previous messages' tags. Using efficient XOR operations, which are mostly precomputable in idle time, SP-MAC is particularly suitable for resource-constrained environments, which are the prime profiteer of ProMACs.

5.2 Security Discussion

The security of SP-MAC follows from its resilience to key recovery attacks and the unforgeability of tags.

5.2.1 Resilience to Key Recovery Attacks. By overhearing the communication, an attacker learns strictly less information from channels that use SP-MAC for integrity protection than channels that rely on SP-MAC's underlying MAC scheme, as the tags computed by them are needed to compute the SP-MAC tag. Hence, a key recovery attack against SP-MAC is at least as hard as against the underlying MAC scheme. Thus, the key used to compute the tags cannot be recovered as long as the underlying MAC scheme does not expose a key recovery attack. The rationale behind this claim is the following: Given a stream protected with traditional MACs, an adversary can choose arbitrary bit dependencies (without needing access to the key) to derive the tags that would have been sent by SP-MAC, similar to the illustration in Figure 7. Thus, any key recovery attack against SP-MAC also attacks the underlying MAC protocol, as an adversary only needs to transform the underlying MAC into SP-MAC's representation before launching the attack.

5.2.2 Unforgeability of Integrity Protection. Security of (traditional) MACs relies on the unforgeability of tags, *i.e.*, attackers can neither directly forge tags nor guess the secret key (*cf.* Section 3.1). When considering ProMACs, the integrity of a single message is secured by multiple tags. At the same time, a single tag protects multiple messages. Other than traditional MACs, ProMACs, therefore, have to define their security based on the (computational) infeasibility of circumventing the integrity protection of a single message.

We assume that the underlying MAC scheme provides a security level of n bits using an n-bit tag, *i.e.*, the probability of guessing a tag is not better than 2^{-n}. This assumption is expected to hold for common MAC schemes, *e.g.*, `HMAC-SHA-256-128` [8], and eases discussions on SP-MAC's security in face of dropped packets. To show the security of SP-MAC, we first look at traditional MACs and think of an n-bit tag as n individual 1-bit MACs. Each of these 1-bit MACs provides 1 bit of security, *i.e.*, the probability that an attacker guesses it correctly is 2^{-1}. A message protected by a traditional MAC is transmitted with its n 1-bit MACs, and if it is not altered, all 1-bit MACs can be verified. For SP-MAC, this procedure changes as the 1-bit MACs are distributed over multiple packets and aggregated using XORs.

This aggregation of multiple MACs itself does not impact security, as XOR-ing multiple MACs still leads to a secure MAC scheme [10]. However, combining multiple MACs introduces dependencies on the successful reception of other messages, as a MAC can only be verified if *all* messages protected by the XOR-ed MACs were received unaltered. Here, R2-D2 ensures that these dependencies are staggered, and thus prevents the sandwich attack introduced earlier (*cf.* Section 3). Consequently, a significant number of messages have to be dropped to render a large number of MAC fragments covering a single message unverifiable. To illustrate this issue, in case R2-D2 is parameterized for a maximum security loss of 16 bits per dropped message for a maximum security level of 128 bit and 4 targeted messages could be dropped, SP-MAC still achieves a security level of at least 64 bit for the targeted message. However, SP-MAC's selected bit dependencies $\mathfrak{D}$ are derived from a pre-shared secret, thus hiding the strategy to achieve this worst-case attack from third parties. Consequently, an adversary needs to drop a suspiciously high number of transmissions [5] to even attempt to circumvent the integrity protection of a single message.

By providing resilience to key recovery attacks and ensuring the unforgeability of integrity protection, SP-MAC is able to realize secure progressive message authentication. Most notably, SP-MAC is the first ProMAC scheme that offers protection against network-level attacks, while still quickly achieving full integrity protection.

5.3 Performance Evaluation

ProMACs are specifically designed for resource-constrained environments, especially for wireless scenarios with high-frequency

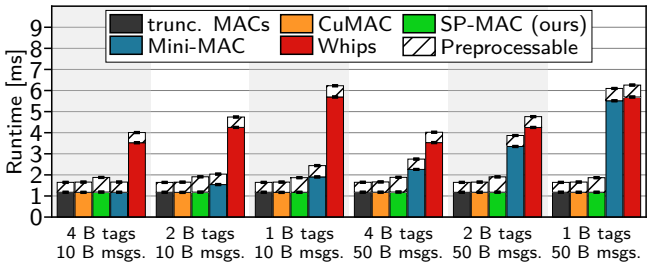


Figure 8: SP-MAC's tag computation adds only marginal (preprocessable) overhead compared to traditional MACs while twarting network-level attacks against ProMACs.

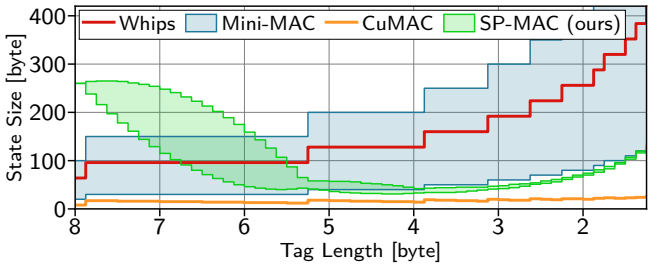


Figure 9: Despite tracking longer histories to thwart attacks, SP-MAC's memory footprint is in the same order as vulnerable ProMAC schemes. Note the decreasing x-axis that represents increasing space savings by ProMACs.

communication (*cf.* Section 2.2). In this context, energy consumption is generally a key metric for battery-powered devices. However, since the bulk of energy of those devices is consumed by the transmission and reception of wireless communication [56], the difference between ProMACs schemes is negligible. Nevertheless, for this reason, the overall energy cost of ProMACs is significantly lower compared to traditional MACs as they require longer tags and thus have a higher transmission overhead. Consequently, our evaluation of SP-MAC focuses on the two most important aspects of these environments: the computational overhead of tag generation and the memory overhead necessary to track message histories.

5.3.1 Computational Overhead of Tag Generation. To evaluate the computational overhead of tag generation in SP-MAC, we implemented a prototype in C for the Contiki-NG [16] platform, which is widely used for low-power embedded systems in resource-constrained scenarios [29–31, 52]. Our prototype relies on the `HMAC-SHA-256-128` implementation of tinyDTLS as an underlying MAC scheme and uses the same R2-D2 parametrization as in Section 4.3 (maximal security loss of 32 bits per dropped packet and 16-bit immediate protection for 4-byte tags).

To compare the performance of SP-MAC with state-of-the-art ProMAC schemes, we additionally re-implemented[1] Whips [5], CuMAC [36, 37], and Mini-MAC [53] for the same platform and underlying MAC scheme. Furthermore, we use a `HMAC-SHA-256-128` MAC as baseline reference (we truncate the MAC for a fair comparison, although the full MAC needs to be computed anyway). To encourage further research, we will make all of our ProMAC implementations for Contiki-NG available to the research community.

With our implementations, we measure the time required to generate one tag of length 1, 2, and 4 byte, respectively, for 10 and 50 byte long messages on a Zolertia RE-Mote embedded device (ARM Cortex-M3 @ 32MHz, 16 kB RAM). All MAC schemes are parameterized for 128-bit security, except for the truncated MAC (baseline). We performed each measurement 30 times and report on the mean over these runs with 99% confidence intervals.

The measurements presented in Figure 8 establish a baseline of 1.17 ms for computing a traditional (truncated) MAC, irrespective of tag size and message length [2]. In contrast, the runtimes of Whips and Mini-MAC depend on the size of their internal state. For Mini-MAC, this state increases with growing message sizes and shrinking tags, ranging from 1.18 ms (4-byte long tag and 10-byte long message) to 5.52 ms (1-byte long tag and 50-byte long message). While Whips' processing overhead is independent of message sizes, it also increases for shrinking tags from 3.52 ms for 4-byte long tags to 5.69 ms for 1-byte long tags. Whips starts with a higher processing overhead, as it calls the underlying MAC function twice, once to compute a new substate and once to derive the actual tag.

On top of the baseline for truncated MACs, CuMAC introduces one additional non-preprocessable XOR operation, resulting in nearly identical runtime as truncated MACs. SP-MAC has the same online performance but adds a marginal 0.23 ms preprocessing overhead to realize R2-D2 (*cf.* Section 5.1) and protect against network-level attacks. As expected, neither SP-MAC's nor CuMAC's runtime is noticeably influenced by tag sizes or message length.

These results show that SP-MAC not only efficiently protects against network-level attacks but even operates at least as resource-conscious as existing ProMACs. Notably, SP-MAC's performance closely aligns with traditional MACs, showing that the benefits of ProMACs (*cf.* Section 2.1) can be realized without increased latency and with only a minor increase in consumed processing power.

5.3.2 Memory Overhead for Keeping Integrity State. In contrast to traditional (truncated) MACs, ProMACs inherently have to keep state on past messages for the computation of future tags (*cf.* Section 2.3). To evaluate the impact of this state keeping on the memory consumption of resource-constrained devices, we evaluate the size of SP-MAC's state in relation to the tag lengths and, again, compare it to Whips, CuMAC, and Mini-Mac. To avoid potential bias due to implementation decisions, we conduct a theoretical memory analysis that is independent of our re-implementations of current ProMAC schemes. We parametrize all schemes to provide at least 128-bit security, with security loss limited by tag sizes, which is the worst case for SP-MAC w.r.t. memory overhead.

The results of our analysis are visualized in Figure 9. First, the memory overhead of Whips depends on the number of tags required for the desired security level (security level divided by tag length). Its overhead ranges from 64 bytes for 8-byte long tags to 416 byte for 10-bit long tags. As the overhead of Mini-MAC additionally depends on the message length, we study the overhead for messages with lengths between 10 and 50 byte and find that the memory

[1] Re-implementation was necessary as no source code was available.
[2] This baseline can further be improved if deemed necessary, *e.g.*, with hardware acceleration. Resulting performance savings carry over to all four ProMAC schemes.

overhead of Mini-MAC follows the same trend as Whips. However, Mini-MAC's overhead depends on the message length, such that the resulting overhead spans an area around Whips' overhead. Exemplary, for a tag length of 4 byte, the overhead of Mini-MAC ranges from 40 byte to 200 byte. In contrast, CuMAC has a low and relatively constant memory overhead since the state stored per message decreases proportionally to the growing number of messages that are aggregated within one tag as tags vary in length.

Similar to CuMAC and Whips, the memory overhead of SP-MAC does not depend on the message length. However, it is influenced by the *random* selection of bit dependencies (*cf.* Section 4.3), as well as the interfering effects of tag length reductions (fewer bits per tag have to be precomputed) and the exponentially growing size of Golomb Rulers of increasing orders (tag precomputation has to start earlier). We thus display the range between the best and worst possible bit dependencies w.r.t. their memory overhead. As shown in Figure 9, this range is negligible for small tags and increasing for larger tags because the number of short Golomb Rulers with two elements (used in tags longer than 42 bit) is limited.

Overall, the memory footprint of SP-MAC is well-manageable regardless of tag lengths, even for devices with scarce resources (*cf.* Section 5.3.1). Moreover, even in the worst-case scenario for SP-MAC in terms of memory overhead, its memory footprint is in the same order as current ProMAC schemes, all of which are vulnerable to the presented sandwich attack (*cf.* Section 3).

6 RELATED WORK

This paper identifies an inherent vulnerability of existing ProMAC schemes and proposes an efficient, flexible, and secure way to achieve strong integrity protection in resource-constrained environments. Besides the directly related prior work on progressive message authentication (covered in Section 2), different streams of research tackle this challenge. Competing technologies to reduce authentication tag sizes are truncated MACs [50, 54, 59, 66], stateful MACs [5], or aggregated MACs [10, 17, 21, 32, 35], which, however, only achieve reduced security or cannot cope with lossy channels. Note that our approach can also be used *conservatively* as an alternative to aggregated MACs as discussed in the Appendix A.2.

Beyond ProMACs, several other message authentication schemes using optimistic security have been proposed in similar resource-constrained environments as those considered in this paper. Only reacting to multiple invalid short tags within a short time span [58, 59, 61] does, however, not protect against the manipulation of a few selected messages. The optimistic and selective use of aggregated MACs [11] does require reliable transmissions and leads to high authentication delays, two limitations that ProMACs address. Relying on external devices to verify broadcasted information [45] is only applicable with costly asymmetric cryptography that leads to unacceptable delays and bandwidth overhead in many scenarios.

Targeting to avoid all communication overhead for authentication, different approaches propose sender identification based on unique physical characteristics of the transmission signal [14, 33, 34, 44] or host behavior [1, 18]. In contrast to the reliable and deterministic nature of MACs, these approaches are, however, restricted to single-hop transmissions and often produce a significant number of false positives. Similarly, approaches to transparently retrofit integrity protection into legacy protocols hardly elongate packets, but therefore result in reduced security (*e.g.*, by using truncated MACs) [51, 54, 59, 64, 69]. ProMACs can replace truncated MACs in these protocols and thus improve security guarantees.

Specifically focusing on *multicast* message authentication in constrained environments, related work addresses the cost of asymmetric cryptography by splitting integrity verification over multiple packets [24] or by adapting symmetric cryptography to the multicast setting [12, 38, 47–49, 70]. For example, TESLA [38, 47, 48] achieves this through time-delayed key disclosure. BECAN [39], in contrast, improves bandwidth consumption for cooperative authentication scenarios where multiple devices must authenticate a single message. ProMACs address deficits in unicast message authentication, but their loss-tolerance shows potential to also improve multicast authentication based on symmetric cryptography.

To improve the performance of symmetric cryptography on resource-constrained devices, different approaches propose to leverage special lightweight ciphers [25, 40, 43], use hardware acceleration [29, 31, 71], or preprocess cryptographic operations [3, 29, 65]. As our work is agnostic to the underlying MAC scheme, these performance improvements conceptually carry over to SP-MAC.

7 CONCLUSION

Progressive message authentication codes (ProMACs) promise the compression of authentication tags while preserving the strong security of traditional MACs by partly offloading integrity protection into the near future. Contrary to prior beliefs, we show that ProMACs cannot cope particularly well with lossy channels, preventing their deployment in many wireless scenarios: The generation and verification of tags depend on a sliding window of past messages, providing the foundation for our *sandwich attack*, in which the integrity protection of a whole message sequence is rendered void if merely two packets are dropped. Therefore, we consider it imperative to rethink how transmission failures influence the integrity protection of neighboring messages. With this in mind, we propose randomized and resilient dependency distributions (R2-D2), which takes advantage of (i) optimal message dependencies, (ii) parameterized security guarantees, (iii) randomized bit dependencies, and (iv) optional immediate protection bits. Our evaluation shows that R2-D2 significantly increases the resilience of ProMACs to lossy channels to unleash their full potential. At the same time, R2-D2 achieves full integrity protection with comparable delays to current ProMAC schemes. To take advantage of R2-D2 and realize a secure and resource-conscious ProMAC scheme, we propose SP-MAC that builds upon the proven security provided by traditional MACs, aggregating and distributing those across multiple messages using efficient XOR operations. SP-MAC is thus not only resilient to lossy channels and sophisticated network-level attacks but also operates as resource-conscious as state-of-the-art ProMAC schemes.

ACKNOWLEDGMENTS

Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC-2023 Internet of Production – 390621612. We thank Misha Lavrov for pointing us to Golomb Rulers as well as the anonymous reviewers and our shepherd Mridula Singh for their fruitful comments.

REFERENCES

[1] Chuadhry Mujeeb Ahmed, Martin Ochoa, Jianying Zhou, and Aditya Mathur. 2021. Scanning the Cycle: Timing-Based Authentication on PLCs. In *Proceedings of the ACM Asia Conference on Computer and Communications Security (ASIA CCS)*. 886—-900. https://doi.org/10.1145/3433210.3453102

[2] Ismet Aktas, Alexander Bentkus, Florian Bonanati, et al. 2017. *Position Paper: Wireless Technologies for Industrie 4.0.* Technical Report. VDE.

[3] Ralph Ankele, Florian Böhl, and Simon Friedberger. 2018. MergeMAC: A MAC for Authentication with Strict Time Constraints and Limited Bandwidth. In *Proceedings of the International Conference on Applied Cryptography and Network Security (ACNS)*. 381–399. https://doi.org/10.1007/978-3-319-93387-0_20

[4] Emekcan Aras, Nicolas Small, Gowri Sankar Ramachandran, Stéphane Delbruel, Wouter Joosen, and Danny Hughes. 2017. Selective Jamming of LoRaWAN using Commodity Hardware. In *Proceedings of the 14th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*. https://doi.org/10.1145/3144457.3144478

[5] Frederik Armknecht, Paul Walther, Gene Tsudik, Martin Beck, and Thorsten Strufe. 2020. ProMACs: Progressive and Resynchronizing MACs for Continuous Efficient Authentication of Message Streams. In *Proceedings of the Conference on Computer and Communications Security (CCS)*. 211–223. https://doi.org/10.1145/3372297.3423349

[6] Wallace C. Babcock. 1953. Intermodulation interference in radio systems frequency of occurrence and control by channel selection. *The Bell System Technical Journal* 32, 1 (1953). https://doi.org/10.1002/j.1538-7305.1953.tb01422.x

[7] Christoph Bachhuber, Eckehard Steinbach, Martin Freundl, and Martin Reisslein. 2017. On the Minimization of Glass-to-Glass and Glass-to-Algorithm Delay in Video Communication. *IEEE Transactions on Multimedia* 20, 1 (2017). https://doi.org/10.1109/TMM.2017.2726189

[8] Mihir Bellare. 2006. New Proofs for NMAC and HMAC: Security Without Collision-Resistance. In *Proceedings of the Annual International Cryptology Conference (CRYPTO)*. Springer. https://doi.org/10.1007/11818175_36

[9] Mihir Bellare, Ran Canetti, and Hugo Krawczyk. 1996. Keying Hash Functions for Message Authentication. In *Proceedings of the Annual International Cryptology Conference (CRYPTO)*. https://doi.org/10.1007/3-540-68697-5_1

[10] Mihir Bellare, Roch Guérin, and Phillip Rogaway. 1995. XOR MACs: New methods for message authentication using finite pseudorandom functions. In *Proceedings of the Annual International Cryptology Conference (CRYPTO)*. Springer. https://doi.org/10.1007/3-540-44750-4_2

[11] John Henry Castellanos, Daniele Antonioli, Nils Ole Tippenhauer, and Martín Ochoa. 2017. Legacy-Compliant Data Authentication for Industrial Control System Traffic. In *In Proceedings of the International Conference on Applied Cryptography and Network Security (ANCS)*. Springer, 665–685. https://doi.org/10.1007/978-3-319-61204-1_33

[12] Yacine Challal, Abdelmadjid Bouabdallah, and Yoann Hinard. 2005. RLH: receiver driven layered hash-chaining for multicast data origin authentication. *Computer Communications* 28, 7 (2005). https://doi.org/10.1016/j.comcom.2004.10.009

[13] Zhao Cheng, Mark Perillo, and Wendi B Heinzelman. 2008. General Network Lifetime and Cost Models for Evaluating Sensor Network Deployment Strategies. *IEEE Transactions on Mobile Computing* 7, 4 (2008). https://doi.org/10.1109/TMC.2007.70784

[14] Wonsuk Choi, Hyo Jin Jo, Samuel Woo, Ji Young Chun, Jooyoung Park, and Dong Hoon Lee. 2018. Identifying ECUs Using Inimitable Characteristics of Signals in Controller Area Networks. *IEEE Transactions on Vehicular Technology* 67, 6 (2018). https://doi.org/10.1109/TVT.2018.2810232

[15] Carlos Alexandre Gouvea Da Silva and Carlos Marcelo Pedroso. 2019. MAC-Layer Packet Loss Models for Wi-Fi Networks: A Survey. *IEEE Access* 7 (2019). https://doi.org/MAC-LayerPacketLossModelsforWi-FiNetworks:ASurvey

[16] Adam Dunkels, Bjorn Gronvall, and Thiemo Voigt. 2004. Contiki - a lightweight and flexible operating system for tiny networked sensors. In *Proceedings of the 29th IEEE international Conference on Local Computer Networks (LCN)*. IEEE, 455–462. https://doi.org/10.1109/LCN.2004.38

[17] Oliver Eikemeier, Marc Fischlin, Jens-Fabian Götzmann, Anja Lehmann, Dominique Schröder, Peter Schröder, and Daniel Wagner. 2010. History-Free Aggregate Message Authentication Codes. In *Proceedings of the International Conference on Security and Cryptography for Networks (SCN)*. Springer. https://doi.org/10.1007/978-3-642-15317-4_20

[18] David Formby and Raheem Beyah. 2020. Temporal Execution Behavior for Host Anomaly Detection in Programmable Logic Controllers. *IEEE Transactions on Information Forensics and Security* 15 (2020), 1455–1469. https://doi.org/10.1109/TIFS.2019.2940890

[19] Andreas Frotzscher, Ulf Wetzker, Matthias Bauer, Markus Rentschler, Matthias Beyer, Stefan Elspass, and Henrik Klessig. 2014. Requirements and current solutions of wireless communication in industrial automation. In *Proceedings of the IEEE International Conference on Communications Workshops (ICC)*. IEEE. https://doi.org/10.1109/ICCW.2014.6881174

[20] Brendan Galloway and Gerhard P Hancke. 2012. Introduction to Industrial Control Networks. *IEEE Communications surveys & tutorials* 15, 2 (2012). https://doi.org/10.1109/SURV.2012.071812.00124

[21] Rosario Gennaro and Pankaj Rohatgi. 1997. How to sign digital streams. In *Proceedings of the Annual International Cryptology Conference (CRYPTO)*. Springer. https://doi.org/10.1007/BFb0052235

[22] René Glebke, Martin Henze, Klaus Wehrle, Philipp Niemietz, Daniel Trauth, Patrick Mattfeld, and Thomas Bergs. 2019. A Case for Integrated Data Processing in Large-Scale Cyber-Physical Systems. In *Proceedings of the 52nd Hawaii International Conference on System Sciences (HICSS)*.

[23] Oded Goldreich. 2009. *Foundations of Cryptography II: Basic Applications*. Cambridge University Press.

[24] Philippe Golle and Nagendra Modadugu. 2001. Authenticating Streamed Data in the Presence of Random Packet Loss. In *Proceedings of the Network and Distributed System Security Symposium (NDSS)*.

[25] Zheng Gong, Pieter Hartel, Svetla Nikova, Shao-Hua Tang, and Bo Zhu. 2014. TuLP: A Family of Lightweight Message Authentication Codes for Body Sensor Networks. *Journal of computer science and technology* 29, 1 (2014). https://doi.org/10.1007/s11390-013-1411-8

[26] Shay Gueron. 2016. Memory Encryption for General-Purpose Processors. *IEEE Security & Privacy* 14, 6 (2016). https://doi.org/10.1109/MSP.2016.124

[27] Taieb Hamza, Georges Kaddoum, Aref Meddeb, and Georges Matar. 2016. A survey on intelligent MAC layer jamming attacks and countermeasures in WSNs. In *Proceedings of the 84th Vehicular Technology Conference (VTC-Fall)*. IEEE. https://doi.org/10.1109/VTCFall.2016.7880885

[28] Thomas Hänel, Leonhard Brüggemann, Felix Loske, and Nils Aschenbruck. 2021. Long-Term Wireless Sensor Network Deployments in Industry and Office Scenarios. In *2021 IEEE 22nd International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*. https://doi.org/10.1109/WoWMoM51794.2021.00024

[29] Jens Hiller, Martin Henze, Martin Serror, Eric Wagner, Jan Niklas Richter, and Klaus Wehrle. 2018. Secure Low Latency Communication for Constrained Industrial IoT Scenarios. In *Proceedings of the 43rd Conference on Local Computer Networks (LCN)*. IEEE. https://doi.org/10.1109/LCN.2018.8638027

[30] Jens Hiller, Jan Pennekamp, Markus Dahlmanns, Martin Henze, Andriy Panchenko, and Klaus Wehrle. 2019. Tailoring Onion Routing to the Internet of Things: Security and Privacy in Untrusted Environments. In *Proceedings of the 27th IEEE International Conference on Network Protocols (ICNP)*. IEEE. https://doi.org/10.1109/ICNP.2019.8888033

[31] Sotirios Katsikeas, Konstantinos Fysarakis, Andreas Miaoudakis, Amaury Van Bemten, Ioannis Askoxylakis, Ioannis Papaefstathiou, and Anargyros Plemenos. 2017. Lightweight & Secure Industrial IoT Communications via the MQ Telemetry Transport Protocol. In *Proceedings of the IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 1193–1200. https://doi.org/10.1109/ISCC.2017.8024687

[32] Jonathan Katz and Andrew Y Lindell. 2008. Aggregate Message Authentication Codes. In *Proceedings of the Cryptographers' Track at the RSA Conference (CT-RSA)*. Springer. https://doi.org/10.1007/978-3-540-79263-5_10

[33] Marcel Kneib and Christopher Huth. 2018. Scission: Signal Characteristic-Based Sender Identification and Intrusion Detection in Automotive Networks. In *Proceedings of the Conference on Computer and Communications Security (CCS)*. ACM. https://doi.org/10.1145/3243734.3243751

[34] Marcel Kneib, Oleg Schell, and Christopher Huth. 2020. EASI: Edge-based sender identification on resource-constrained platforms for automotive networks. In *Proceedings of the Network and Distributed System Security Symposium (NDSS)*. https://doi.org/10.14722/ndss.2020.24025

[35] Vladimir Kolesnikov and Wonsuck Lee. 2012. MAC aggregation protocols resilient to DoS attacks. *International Journal of Security and Networks* 7, 2 (2012). https://doi.org/10.1504/IJSN.2012.050028

[36] He Li, Vireshwar Kumar, Jung-Min Park, and Yaling Yang. 2021. Cumulative Message Authentication Codes for Resource-Constrained IoT Networks. *IEEE Internet of Things Journal* (2021). https://doi.org/10.1109/JIOT.2021.3074054

[37] He Li, Vireshwar Kumar, Jung-Min Jerry Park, and Yaling Yang. 2020. Cumulative Message Authentication Codes for Resource-Constrained Networks. In *Proceedings of the IEEE Conference on Communications and Network Security (CNS)*. IEEE. https://doi.org/10.1109/CNS48642.2020.9162217

[38] Donggang Liu and Peng Ning. 2004. Multilevel μTESLA: Broadcast authentication for distributed sensor networks. *ACM Transactions on Embedded Computing Systems (TECS)* 3, 4 (2004). https://doi.org/10.1145/1027794.1027800

[39] Rongxing Lu, Xiaodong Lin, Haojin Zhu, Xiaohui Liang, and Xuemin Shen. 2012. BECAN: A Bandwidth-Efficient Cooperative Authentication Scheme for Filtering Injected False Data in Wireless Sensor Networks. *IEEE Transactions on Parallel and Distributed Systems* 23, 1 (2012). https://doi.org/10.1109/TPDS.2011.95

[40] Atul Luykx, Bart Preneel, Elmar Tischhauser, and Kan Yasuda. 2016. A MAC Mode for Lightweight Block Ciphers. In *Proceedings of the International Conference on Fast Software Encryption (FSE)*. Springer. https://doi.org/10.1007/978-3-662-52993-5_3

[41] Daisuke Mashima, Ramkumar Rajendran, Toby Zhou, Binbin Chen, and Biplab Sikdar. 2019. On optimization of command-delaying for advanced command authentication in smart grid systems. In *2019 IEEE Innovative Smart Grid Technologies-Asia (ISGT Asia)*. IEEE.

[42] Sadia Moriam, Elke Franz, Paul Walther, Akash Kumar, Thorsten Strufe, and Gerhard Fettweis. 2018. Protecting Communication in Many-Core Systems against Active Attackers. In *Proceedings of the Great Lakes Symposium on VLSI (GLSVLSI)*. ACM. https://doi.org/10.1145/3194554.3194582
[43] Nicky Mouha, Bart Mennink, Anthony Van Herrewege, Dai Watanabe, Bart Preneel, and Ingrid Verbauwhede. 2014. Chaskey: an efficient MAC algorithm for 32-bit microcontrollers. In *Proceedings of the International Conference on Selected Areas in Cryptography (SAC)*. Springer.
[44] Pal-Stefan Murvay and Bogdan Groza. 2014. Source Identification Using Signal Characteristics in Controller Area Networks. *IEEE Signal Processing Letters* 21, 4 (2014). https://doi.org/10.1109/LSP.2014.2304139
[45] Dennis K Nilsson, Ulf E Larson, and Erland Jonsson. 2008. Efficient in-vehicle delayed data authentication based on compound message authentication codes. In *2008 IEEE 68th Vehicular Technology Conference*. IEEE.
[46] National Institute of Standards and Technology. 2020. Recommendation for Key Management: Part 1 – General. *NIST Special Publication 800-57 Part 1 Revision 5* (2020).
[47] Adrian Perrig, Ran Canetti, Dawn Song, and J. Doug Tygar. 2001. Efficient and Secure Source Authentication for Multicast. In *Proceedings of the Network and Distributed System Security Symposium (NDSS)*.
[48] Adrian Perrig, Ran Canetti, J. Doug Tygar, and Dawn Song. 2000. Efficient authentication and signing of multicast streams over lossy channels. In *Proceeding of the IEEE Symposium on Security and Privacy (S&P)*. IEEE. https://doi.org/10.1109/SECPRI.2000.848446
[49] Adrian Perrig, Robert Szewczyk, Justin Douglas Tygar, Victor Wen, and David E Culler. 2002. SPINS: Security protocols for sensor networks. *Wireless networks* 8, 5 (2002).
[50] Bart Preneel and Paul C. Van Oorschot. 1995. MDx-MAC and building fast MACs from hash functions. In *Proceedings of the Annual International Cryptology Conference (CRYPTO)*. Springer. https://doi.org/10.1007/3-540-44750-4_1
[51] Andreea-Ina Radu and Flavio D Garcia. 2016. LeiA: A lightweight authentication protocol for CAN. In *Proceedings of the European Symposium on Research in Computer Security (ESORICS)*. Springer.
[52] Rizwan Hamid Randhawa, Abdul Hameed, and Adnan Noor Mian. 2019. Energy efficient cross-layer approach for object security of CoAP for IoT devices. *Ad Hoc Networks* 92 (2019). https://doi.org/10.1016/j.adhoc.2018.09.006
[53] Jackson Schmandt, Alan T. Sherman, and Nilanjan Banerjee. 2017. Mini-MAC: Raising the bar for vehicular security with a lightweight message authentication protocol. *Vehicular Communications* 9 (2017).
[54] Hendrik Schweppe, Yves Roudier, Benjamin Weyl, Ludovic Apvrille, and Dirk Scheuermann. 2011. Car2X Communication: Securing the Last Meter - A Cost-Effective Approach for Ensuring Trust in Car2X Applications Using In-Vehicle Symmetric Cryptography. In *Proceedings of the IEEE Vehicular Technology Conference (VTC Fall)*. IEEE. https://doi.org/10.1109/VETECF.2011.6093081
[55] Martin Serror, Eric Wagner, René Glebke, and Klaus Wehrle. 2020. QWIN: Facilitating QoS in Wireless Industrial Networks Through Cooperation. In *Proceedings of the IFIP Networking Conference (Networking)*.
[56] Faisal Karim Shaikh and Sherali Zeadally. 2016. Energy harvesting in wireless sensor networks: A comprehensive review. *Renewable and Sustainable Energy Reviews* 55 (2016), 1041–1054. https://doi.org/10.1016/j.rser.2015.11.010
[57] Simon Sidon. 1932. Ein Satz über trigonometrische Polynome und seine Anwendung in der Theorie der Fourier-Reihen. *Math. Ann.* 106, 1 (1932). https://doi.org/10.1007/BF01455900
[58] Christopher Szilagyi. 2012. *Low cost multicast network authentication for embedded control systems*. Ph. D. Dissertation. Carnegie Mellon University.
[59] Christopher Szilagyi and Philip Koopman. 2008. A Flexible Approach to Embedded Network Multicast Authentication. In *Proceedings of the 2nd Workshop on Embedded Systems Security (WESS)*. https://doi.org/10.1184/R1/6620639.v1
[60] Christopher Szilagyi and Philip Koopman. 2009. Flexible multicast authentication for time-triggered embedded control network applications. In *Proceedings of the IEEE/IFIP International Conference on Dependable Systems & Networks (DSN)*.
[61] Christopher Szilagyi and Philip Koopman. 2010. Low Cost Multicast Authentication via Validity Voting in Time-Triggered Embedded Control Networks. In *Proceedings of the 5th Workshop on Embedded Systems Security (WESS)*.
[62] A. Richard Thompson, James M. Moran, and George W. Swenson Jr. 2017. *Interferometry and Synthesis in Radio Astronomy*. Springer Nature. https://doi.org/10.1007/978-3-319-44431-4
[63] Gilles Thonet, Patrick Allard-Jacquin, and Pierre Colle. 2008. ZigBee – WiFi Coexistence. *Schneider Electric White Paper and Test Report* 1 (2008).
[64] Anthony Van Herrewege, Dave Singelee, and Ingrid Verbauwhede. 2011. CANAuth - A Simple, Backward Compatible Broadcast Authentication Protocol for CAN bus. In *Proceedings of the ECRYPT Workshop on Lightweight Cryptography*, Vol. 2011.
[65] Eric Wagner, Martin Serror, Klaus Wehrle, and Martin Henze. 2022. BP-MAC: Fast Authentication for Short Messages. In *Proceedings of the 15th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*.
[66] Hai Wang and Abraham O Fapojuwo. 2017. A Survey of Enabling Technologies of Low Power and Long Range Machine-to-Machine Communications. *IEEE Communications Surveys & Tutorials* 19, 4 (2017). https://doi.org/10.1109/COMST.2017.2721379
[67] Yi-Hung Wei, Quan Leng, Song Han, Aloysius K. Mok, Wenlong Zhang, and Masayoshi Tomizuka. 2013. RT-WiFi: Real-time high-speed communication protocol for wireless cyber-physical control applications. In *Proceedings of the 34th Real-Time Systems Symposium (RTSS)*. IEEE. https://doi.org/10.1109/RTSS.2013.22
[68] Matthias Wilhelm, Ivan Martinovic, Jens B. Schmitt, and Vincent Lenders. 2011. Short paper: reactive jamming in wireless networks: how realistic is the threat?. In *Proceedings of the fourth ACM conference on Wireless network security(WiSec)*. ACM. https://doi.org/10.1145/1998412.1998422
[69] Andrew K. Wright, John A. Kinast, and Joe McCarty. 2004. Low-Latency Cryptographic Protection for SCADA Communications. In *Proceedings of the International Conference on Applied Cryptography and Network Security (ANCS)*. Springer, 263–277. https://doi.org/10.1007/978-3-540-24852-1_19
[70] Taojun Wu, Yi Cui, Brano Kusy, Akos Ledeczi, Janos Sallai, Nathan Skirvin, Jan Werner, and Yuan Xue. 2007. A Fast and Efficient Source Authentication Solution for Broadcasting in Wireless Sensor Networks. In *New Technologies, Mobility and Security*. Springer. https://doi.org/10.1007/978-1-4020-6270-4_5
[71] Kaiyuan Yang, David Blaauw, and Dennis Sylvester. 2017. Hardware Designs for Security in Ultra-Low-Power IoT Systems: An Overview and Survey. *IEEE Micro* 37, 6 (2017). https://doi.org/10.1109/MM.2017.4241357
[72] Chenxi Zhang, Xiaodong Lin, Rongxing Lu, and P-H Ho. 2008. RAISE: An efficient RSU-aided message authentication scheme in vehicular communication networks. In *Proceedings of the IEEE International Conference on Communications (ICC)*.
[73] Ting Zhu, Ziguo Zhong, Tian He, and Zhi-Li Zhang. 2010. Exploring Link Correlation for Efficient Flooding in Wireless Sensor Networks. In *Proceedings of the 7th USENIX conference on Networked systems design and implementation (NSDI)*.

A APPENDIX

A.1 Implications of the Sandwich Attack

As shown in this paper, current ProMAC schemes suffer from a common vulnerability inherent to their design: Since integrity protection is distributed among *consecutive* messages, the (forced) loss of two messages that are less than the message history apart already disables integrity protection for all messages in between. Thus, with minimal and unsuspicious interference, an attacker can covertly remove authenticity from multiple ProMAC-protected messages. Depending on how ProMACs are deployed, this vulnerability leads to effective denial-of-service attacks or even false data injections.

A.1.1 Reverting All Suspicious Traffic. ProMACs promise to enable the optimistic processing of messages upon reception based on reduced initial security. In the unlikely event of a retroactive detection that a message could not be authenticated, the effects of a potentially malicious message have to be reverted. If ProMACs operate on a high-reliability link with hardly any packet loss, this mode of operation is reasonable. However, in the presence of a jamming attack or a less reliable channel, current ProMACs schemes cannot ensure a stable operation due to frequent rollbacks. Consider a *selective jammer* that listens to the communication channel to identify "interesting" packets, *i.e.*, those that need to be suppressed to launch a sandwich attack, and then deliberately distorts these packets [4, 27, 68]. To implement such a jammer, an attacker has to control a device that (i) is in range of the targeted communication and (ii) can actively jam ongoing communication.

In Figure 10, we study how jamming capabilities translate to success rates for the sandwich attack. A successful attack means that a considered packet could not be authenticated, which in this deployment scenario forces the entire system to roll back to the point at which the targeted message has been optimistically processed. To this end, we assume that an attacker attempts to make exactly one message unauthenticable by jamming the neighboring packets influencing this message's integrity. We consider three tag sizes (1,

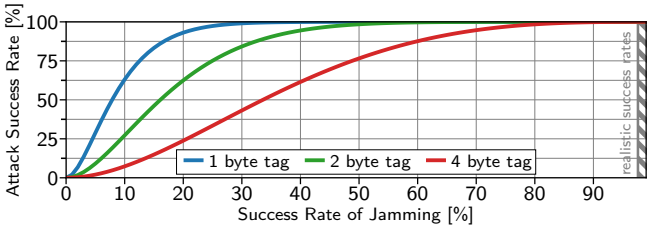


Figure 10: How well a selective jammer can execute the sandwich attack depends on the effectiveness of jamming. In realistic scenarios, the attack can be launched reliably even from an imperfect jammer.

2, and 4 bytes) and analytically compute the attack success rate for varying likelihoods of successful jamming. Our results show that even an imperfect jammer can execute the sandwich attack reliably. Furthermore, smaller tags are more susceptible to selective jammers (for identical security levels), as their larger sliding windows (which are inversely proportional to tag sizes) give an attacker more opportunities to jam packets.

To put our results into perspective, we highlight (in Figure 10) the practical likelihood of successful selective jamming [4, 68]. For IEEE 802.15.4, used in common wireless protocol stacks for constrained devices, selective jamming has proven effective in covertly dropping packets with success rates between 97.6% and 99.9% [68]. Similar numbers have been reported for LoRaWAN, used for energy-efficient long-range IoT communication, where selective jamming using commodity hardware shows success rates between 98.7% and 99.9% [4]. These success rates can further be improved if transmission times can be predicted, *e.g.*, when TDMA is used on the medium access layer [27]. Considering these numbers, our analysis shows that active jammers can execute the sandwich attack with success rates above 99.9% in realistic settings. Consequently, in this deployment, a jammer can cripple a ProMAC-protected communication link with selective, and thus stealthy, interference.

Furthermore, current ProMACs cannot be exposed to harsh environments, such as *e.g.*, ICS (*cf.* Section 2.2) with realistic error rates of 1 to 10 % [63, 67], without restrictions. To validate this claim, we simulate two wireless communication channels ("low-error" and "high-error") based on the Gilbert-Elliot (G-E) model, commonly used to simulate wireless channels based on a Markov chain with two states [15]. In the G-E model, the two states are used to encode a "good" and "bad" channel state, with corresponding packet drop probabilities $\texttt{err}_{\texttt{good}}$ and $\texttt{err}_{\texttt{bad}}$. The parameters `p` and `r` define the probability for switching from "good" to "bad" and vice versa. We parameterize the G-E models as summarized in Table 1 based on recommendations from the literature [28], to represent packet error rates of approximately 1% (low-error) and 10% (high-error), which are realistic for scenarios envisioned for ProMACs [63, 67].

Using these two channel models, we first investigate the number of unauthenticatable packets for varying tag sizes (1, 2, and 4 bytes). Figure 11 reports on the mean attack success rate over 30 Monte Carlo simulations covering 1000 attacks with 99% confidence intervals. A successful "attack" again means that the considered packet

Channel	Model Parameters (%) p	r	$\texttt{err}_{\texttt{good}}$	$\texttt{err}_{\texttt{bad}}$	resulting avg. PER
low-error	0.5	75.9	1.1	62.4	1.50%
high-error	3.2	83	6.8	78.8	9.47%

Table 1: Our parametrization of the Gilbert-Elliot models of a realistic low- and high-error link based on recent longitudinal measurements of industrial networks [28].

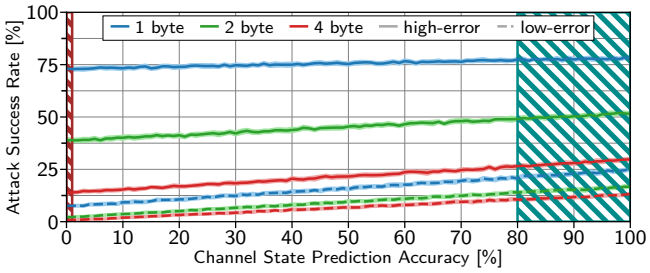


Figure 11: Even attackers without jamming capabilities can execute the sandwich attack, especially if they can predict the state of the communication channel well enough.

could not be authenticated. We found that the overall channel quality, as well as the used ProMAC tag length, influence the attack's success rate. In particular, we observe (in Figure 11) that between 0.6% or 73.1% of messages do not have any verifiable integrity protection, depending on the overall channel quality and tag length. These results indicate the number of overall unauthenticable packets for current ProMAC schemes over lossy channels.

A.1.2 Reacting Only to Explicitly Detected Attacks. Previous publications on ProMACs [5, 36, 37, 53] imply a second deployment scenario, where rollbacks only take place after a manipulation is *explicitly* detected. However, in this scenario, an attacker can inject malicious traffic into a data stream that is not detected as such by jamming the neighboring packets with a high success rate (*cf.* Figure 10). Alternatively, a less powerful attacker can abuse the naturally occurring unauthenticatable packets on a link with higher error rates. If attackers are even able to predict a bad channel state, they can increase the chances of their data injection not being detected. To illustrate this behavior, Figure 11 increase the channel state prediction accuracies (specifying the likelihood of the channel being in the "bad" state) for the attacker on the x-axis. After predicting a bad channel, the attacker waits for one additional transmission before injecting a forged message. The attack is successful if the two windows, starting and ending with the forged message, each contain at least one transmission failure.

In practice, such prediction accuracies can be upwards of 80% if the attacker is in the vicinity of the receiver (< 1 m apart), as indicated by practical measurements [73]. Assuming an accuracy of 80%, an attacker can successfully launch a sandwich attack in 10.4% to 75.4% of attempts, depending on the link quality and tag lengths (in Figure 11). Overall, the prospects of ProMACs are extremely desirable (*cf.* Section 2.1), but even a moderate error rate on the transmission medium leads to significant risk of false data

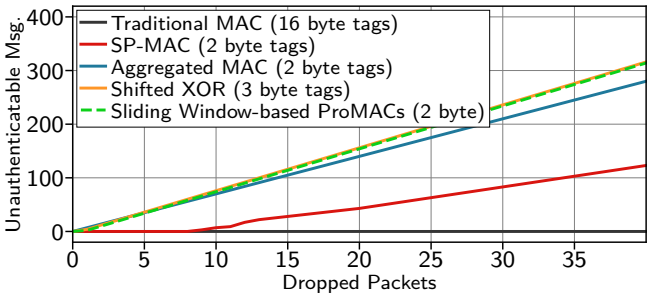


Figure 12: While traditional aggregated MAC schemes are immediately disrupted by dropped packets, SP-MAC, as aggregated MAC schemes, can tolerate some dropped packets and reduce the effects of ongoing DoS attacks in comparison to traditional schemes.

injections or crippled communication channels. Meanwhile, smaller tags with larger sliding windows favor the sandwich attack exactly for those scenarios that benefit the most from ProMACs.

A.2 SP-MAC as an Aggregated MAC Scheme

While this paper focuses on ProMACs that process messages *optimistically* upon reception, ProMACs can also be used *conservatively* as an alternative to aggregated MACs. In such a scenario, messages would only be processed once they are fully authenticated and discarded otherwise. This scenario, however, does not align with the low-latency properties attributed to ProMACs and has thus not been considered in the literature so far. Still, we demonstrate how our later improvements to ProMACs allow them to outperform aggregated MAC schemes in such deployment.

In an aggregate MAC scheme, the authentication of a batch of messages is aggregated into a single tag that is sent to the receiver after the last message of the batch has been transmitted. Those schemes only process messages once a certain bit level of security is achieved and thus do not enable the same low latency processing as ProMACs. Still, many scenarios do not require low latency, and current aggregated MAC schemes [10, 17, 21, 32, 35] are a well-proven method to reduce the overhead of traditional message authentication in such cases. However, similar to ProMACs, aggregated MAC schemes are susceptible to network-level interference: Attacks similar to the one presented against ProMAC schemes (*cf.* Section 3) allow for an attacker to drop a few select packets having a cascading effect on the verifiability of neighboring packets [35], which finally results in many discarded transmissions. In the following, we show how SP-MAC, our proposed ProMAC scheme, is also resilient to network-level interference when used as an aggregated MAC scheme. For our analysis, we compare different aggregated MAC schemes based on how many messages are discarded due to the lack of integrity protection in the presence of an attacker that drops selected packets. As a baseline, we use traditional MACs, where the receiver can verify the integrity of each received packet. As aggregated MAC scheme, we first use a traditional scheme [17, 32]. Additionally, we consider schemes that are aggregated based on shifted XORs [35], an aggregation mechanism that protects, at the cost of slightly longer tags, against DoS attacks where the attacker is only able to selectively drop certain packets. In a simulation, we compare these schemes against sliding window-based ProMAC schemes (Whips [5], CuMAC [36, 37], and Mini-MAC [53]) and our proposed ProMAC scheme (SP-MAC). We consider short authentication tags (2 bytes in most cases) and parameterize SP-MAC to a maximum security loss of 16 bit per dropped packet. For the five considered approaches, we let an attacker drop an increasing number of selected packets chosen to maximize the number of packets discarded by the receiver. We consider a message authentic, *i.e.*, it does not have to be discarded by the receiver if it reaches at least a security level of 32 bit.

The results of our analysis are shown in Figure 12. For traditional MACs, we see, as expected, that no message loses its integrity protection while being received. Aggregated MACs, on the other hand, aggravate the effect of an attack because a single dropped packet invalidates all messages authenticated in the corresponding batch (8 in this case). Even the Shifted XOR aggregation scheme does not improve resiliency, as it was specifically designed to protect against attacks that cannot drop all packets. In fact, the Shifted XOR aggregation scheme performs even worse than aggregated MACs for an attacker that can drop arbitrary packets. Sliding window-based ProMACs exhibit similarly poor results, as an attack can invalidate a sequence of messages by dropping the two packets at the edge of this sequence (*cf.* Section 3).

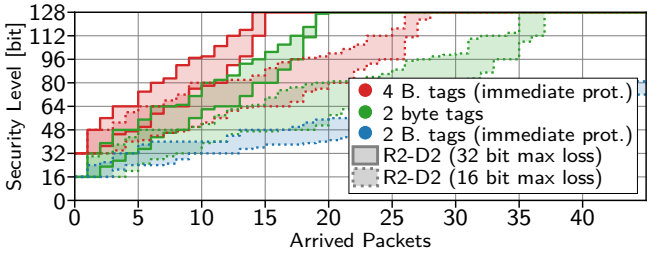
Regarding SP-MAC, we observe a completely different behavior than that of other aggregated MAC schemes. First, the initial packet drops do not invalidate the integrity protection of any messages transmitted in other packets. The security properties of R2-D2, on which SP-MAC is built, ensure that only a certain fraction of each message's integrity protection (16 bit in our parameterization) depends on the successful reception of any other packet. Furthermore, SP-MAC's optimized dependencies on surrounding packets also provide protection against ongoing attacks. While SP-MAC cannot ensure that all messages' integrity protection can be verified, it still cushions the effects of the DoS attack by limiting the number of unauthenticatable messages to less than half of what aggregated MACs achieve. Thus, while exact numbers still depend on SP-MAC's specific parameterization (*e.g.*, targeted security level or maximum security loss), our analysis shows that SP-MAC significantly outperforms competing aggregated MAC schemes in scenarios with packet loss caused by either lossy channels or an active attacker. Consequently, SP-MAC is not only the first ProMAC scheme that is resilient to sandwich attacks but also improves the state-of-the-art of aggregated MAC schemes.

A.3 Proofs of Security Properties

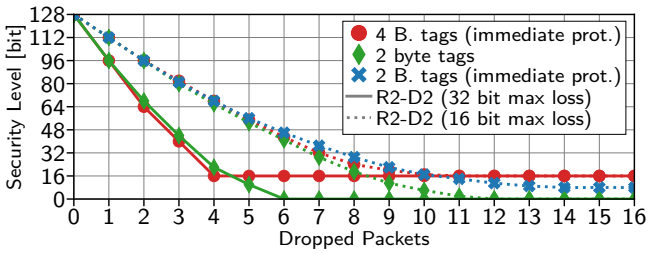
A.3.1 *Security Properties of Golomb Rulers-based dependencies.*

Proposition 1. By dropping a message m of a ProMAC-protected stream, any other message's integrity protection is reduced by at most the security provided by one tag iff the dependency $\mathcal{D}$ is a Golomb Ruler.

Proof. Without loss of generality, we investigate the authenticity of the message m_i. The integrity of m_i is protected by tags $\{t_{i+d} | d \in \mathcal{D}\} = \mathcal{P}$. We prove our claim by contradiction. Therefore, we assume that there exists $m_j (j \neq i)$, such that at least two tags from $\mathcal{P}$ become unverifiable if m_j is dropped. These two tags have



(a) Higher resilience to network-level attackers inherently increase the delay until full authenticity can be reached, but keeps them within a range that is still manageable in many scenarios.



(b) By reducing the maximally lost bit security per dropped packet, the resilience to network-level attacks increases accordingly as has been proven for R2-D2's dependency selection.

Figure 13: R2-D2's flexible parameterization enables a fine-granular and controllable trade-off between security, authentication delay, and message overhead.

the form $t_{i+d} \in \mathcal{P}$ and $t_{i+d'} \in \mathcal{P}$ $(d \neq d')$. If both of these tags become unverifiable because m_j is dropped, m_j has to be included in the intersection $\mathcal{I} = \{m_{i+d-\delta} | \delta \in \mathcal{D}\} \cap \{m_{i+d'-\delta'} | \delta' \in \mathcal{D}\}$ of the messages that are required to compute these tags. Since we assumed that $m_j \in \mathcal{I}$, this requires the existence of $d, d', \delta, \delta' \in \mathcal{D}$, such that $d - \delta = d' - \delta'$. However, exactly when $\mathcal{D}$ is a Golomb Ruler, this only holds if both differences equal 0, *i.e.*, $d = \delta$ and $d' = \delta'$. However, if $d - \delta = d' - \delta' = 0$, then it has to hold that $\mathcal{I} = \{m_i\}$, which means that $m_j (i \neq j) \notin \mathcal{I}$. Thus, there cannot exists any message m_j that, if dropped, invalidates multiple tags authenticating m_i. □

A.3.2 *Security Properties of g-Sidon Set-based dependencies.*

Proposition 2. Using g-Sidon Sets as ProMAC dependencies $\mathcal{D}$ guarantees that any message's security level is reduced by at most the integrity protection provided by g tags for any dropped message.

Proof. Without loss of generality, we investigate the authenticity of the message m_i. The integrity of m_i is protected by tags $\{t_{i+d} | d \in \mathcal{D}\} = \mathcal{P}$. We prove Proposition 2 by contradiction. Therefore, we assume that there exists $m_j (j \neq i)$, such that at least $g + 1$ distinct tags from $\mathcal{P}$ become unverifiable if m_j is dropped. These $g + 1$ tags have the form $t_{i+d} \in \mathcal{P}$, with a distinct d for each tag. If all of these $g + 1$ tags become unverifiable because m_j is dropped, m_j has to be included in the intersection $\mathcal{I} = \{m_{i+d-\delta} | \delta \in \mathcal{D}\} \cap \{m_{i+d'-\delta'} | \delta' \in \mathcal{D}\}$ of the messages that are required to compute these tags. Since we assumed that $m_j \in \mathcal{I}$, this requires the existence of $g+1$ distinct δ-δ' pairs, such that $d - \delta = d' - \delta'$, with $d, d', \delta, \delta' \in \mathcal{D}$. However, exactly when $\mathcal{D}$ is a g-Sidon Set, there exist by definition at most g distinct δ-δ' pairs (*cf.* Section 4.2). Thus, there cannot exist a message m_j that, if dropped, invalidates more than g tags that authenticate m_i. □

A.4 Looking at Additional R2-D2 Parameters

In this paper, we mainly considered a security loss of 32 bits per dropped packet as an acceptable security guarantee. Indeed, in Section 4.4, we saw how exactly such a parameterization provides adequate security in a range of practical examples while keeping the delay to achieve full authentication within an acceptable range. However, we also observe that these guarantees might not provide sufficient resilience to network-level attacks. Meanwhile, R2-D2 offers flexible parameterization, which enables finding an application-specific balance between security, authentication delay, and tag size. To further illustrate how these trade-offs for different parameters, we show three additional parameters sets that reduce the security loss per dropped packet to at most 16 bits.

Since 1-byte tags already lead to significant delays if security loss was parameterized to 32 bit (cf. Figure 6a), we consider 4 and 2 byte long tags. For those tags, we reserve half of them, *i.e.*, 16 and 8 bit respectively, for immediate protection bits, as those ensure that no network-layer attacker can remove all integrity protection for any single message. Additionally, we include a 2-byte tag that does not have any immediate security bits. Figure 13 reports on the results for these new parameters (dotted lines) as well as the R2-D2 parameters previously evaluated in Section 4.3 for the same tag lengths. In Figure 13a, we observe that delays to achieve full authentication are prolonged by reducing maximally accepted bit security loss per dropped packet. While the delays for tags with 16 progressive security bits are at most 37 messages, which is acceptable for many applications (*cf.* Section 4.4), delays for 2-byte tags with 8 bits of immediate security reach up to 75 messages. Thus, such short tags, with the additional benefit of never losing all authenticity due to a network-level attacker, introduce considerable delays. If these delays are not acceptable and ProMACs are still desirable, some other trade-offs must be made in such situations (*i.e.*, slightly longer tags or less resilience against network-layer attackers), since R2-D2's delays are provably optimal and can not be further improved without such trade-offs.

Finally, Figure 13b shows how the increased resilience to dropped packets manifests itself in practice. As expected, at the cost of longer authentication delays, R2-D2 can significantly improve ProMACs' resilience to packet loss. Additionally, this figure illustrates what has been previously proved for R2-D2 under its various parameters, *i.e.*, the achieved bit security never reduces by more than 16 bits per dropped packet for the newly evaluated parameters (dotted lines). All in all, we can conclude that the provably optimal dependency distributions of R2-D2 for its core parameters (maximum bit security per dropped packet, tag length, and immediate security bits) are thus flexibly adaptable to the needs of many realistic scenarios to realize ProMACs with adequate resilience to network-level attackers.