ARID: Anonymous Remote IDentification of Unmanned Aerial Vehicles

Pietro Tedeschi
ptedeschi@hbku.edu.qa
Division of Information and Computing Technology (ICT), College of Science and Engineering (CSE), Hamad Bin Khalifa University (HBKU)
Doha, Qatar

Savio Sciancalepore
s.sciancalepore@tue.nl
Eindhoven University of Technology (TU/e), Department of Mathematics and Computer Science
Eindhoven, The Netherlands

Roberto Di Pietro
rdipietro@hbku.edu.qa
Division of Information and Computing Technology (ICT), College of Science and Engineering (CSE), Hamad Bin Khalifa University (HBKU)
Doha, Qatar

ABSTRACT

To enable enhanced accountability of Unmanned Aerial Vehicles (UAVs) operations, the US-based Federal Avionics Administration (FAA) recently published a new dedicated regulation, namely *RemoteID*, requiring UAV operators to broadcast messages reporting their identity and location. The enforcement of such a rule, mandatory by 2022, generated significant concerns on UAV operators, primarily because of privacy issues derived by the indiscriminate broadcast of the plain-text identity of the UAV on the wireless channel.

In this paper, we propose ARID, a solution enabling *RemoteID*-compliant Anonymous Remote Identification of UAVs. The adoption of ARID allows UAVs to broadcast *RemoteID*-compliant messages using ephemeral pseudonyms that only a Trusted Authority, such as the FAA, can link to the long-term identifier of the UAV and its operator. Moreover, ARID also enforces UAV message authenticity, to protect UAVs against impersonation and spoofed reporting, while requiring an overall minimal toll on the battery budget. Furthermore, ARID generates negligible overhead on the Trusted Authority, not requiring the secure maintenance of any private database.

While the security properties of ARID are thoroughly discussed and formally verified with *ProVerif*, we also implemented a prototype of ARID on a real UAV, i.e., the 3DR-Solo drone, integrating our solution within the popular Poky Operating System, on top of the widespread MAVLink protocol. Our experimental performance evaluation shows that the most demanding configuration of ARID takes only ≈ 11.23 ms to generate a message and requires a mere 4.72 mJ of energy. Finally, we also released the source code of ARID to foster further investigations and development by Academia, Industry, and practitioners.

KEYWORDS

Unmanned Aerial Vehicles, Anonymity, Remote Identification, Authentication, Prototyping

ACSAC '21, December 6–10, 2021, Virtual Event, USA

ACM Reference Format:
Pietro Tedeschi, Savio Sciancalepore, and Roberto Di Pietro. 2021. ARID: Anonymous Remote IDentification of Unmanned Aerial Vehicles. In *Annual Computer Security Applications Conference (ACSAC '21), December 6–10, 2021, Virtual Event, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3485832.3485834

1 INTRODUCTION

Unmanned Aerial Vehicles (UAVs), also known as *drones*, have raised significant attention from Academia and Industry over the last years, thanks to the great benefits they can bring to several application domains, such as Transportation, Health, Entertainment, and the Military, to name a few [36],[28], [48].

Nonetheless, the widespread adoption of UAVs also generated severe privacy and safety concerns [35], [45],[47]. Indeed, several Critical Infrastructures (CIs) operators, such as the ones in airports and military areas, recently reported invasions and unauthorized accesses by amateur UAV, creating serious security, privacy, and safety issues [44],[53],[8].

To enable accountability of UAV operations and identify malicious operators, regulatory authorities such as the US-based Federal Avionic Administrator (FAA) recently introduced a dedicated regulation, namely *Remote Identification* (*RemoteID*) [23], and also Europe is taking similar actions [16]. In brief, *RemoteID* forces all UAVs, including amateur and toy ones, to emit periodic broadcast messages reporting their identity, location, and information about the ground station (see Section 3 for further details). *RemoteID* regulations became effective on the 21st of April, 2021, and UAV operators need to comply with this rule from September 2022 [24].

While meeting the requests of CI operators, the *RemoteID* rule created significant concerns for the UAV community [4]. In particular, UAV operators in critical sectors such as retail, transportation, and health assistance raised issues related to the privacy of their operations, warning that the *RemoteID* rule could enable straightforward identification of an UAV and its operator, as well as uncontrolled leakage of private and sensitive information, such as storage centres location and customer information [20]. Recently, UAV amateur communities also filed a case to the FAA for the privacy issues generated by the mandatory adoption of *RemoteID* [19]. To partially meet privacy requests by UAV operators, the latest *RemoteID* rule provides the possibility to replace the UAV long-term identity with a *Session Identifier*, i.e., a *pseudonym*, hiding the identity of an UAV while still enabling unique identification from the FAA. However, the *RemoteID* rules neither specify how to generate such identifiers, nor provide guidelines to operators for their design.

Pseudonyms generation and management issues have been investigated by a few contributions in the last years, especially in the context of Vehicular Ad-Hoc Networks (VANETs) [12]. Also, specific standards are available on the topic, including the ones published by ETSI [21]. However, as discussed in Section 2, such solutions either require the support of dedicated infrastructure elements, such as Internet-connected Road-Side Units (RSUs), or online Authorities, or they assume the presence of multiple collaborating peers. Conversely, commercial amateur UAVs often do not feature a (persistent) Internet connection, and they operate in an ad-hoc setup without any chance to interact with other peers. The cited requirements make previously published schemes unsuitable for the problem, and call for new domain-specific solutions.

Contribution. In this paper, we propose an Anonymous Remote IDentification solution (*ARID*) to tackle the cited challenges. In particular, *ARID* is a lightweight and *RemoteID*-compliant solution enabling any UAVs to generate *RemoteID* messages that can be verified only by legitimate authorities, being otherwise anonymous. *ARID* achieves the cited objectives independently from the presence of a persistent Internet connection and the presence of multiple collaborating peers, fulfilling all the requirements of amateur UAVs. Adopting *ARID*, only a trusted authority (e.g., the FAA) can obtain the UAV's long-term identity. At the same time, we also provide message authentication to protect drones from false reporting, by allowing the authority to verify (and discard) spoofed messages that could be generated by malicious parties.

While the security and privacy feature of *ARID* have been discussed and formally verified, we also showed the deployability and ease of adoption of our solution. Indeed, we implemented a prototype of *ARID* on a real UAV, i.e., the 3DR-Solo, integrating it with the open-source Poky OS (a reference distribution of the Yocto Project) and the Micro Air Vehicle Link (MAVLink) protocol. Our extensive performance evaluation shows that, assuming the highest level of security, *ARID* messages can be generated and delivered with a single broadcast message in only ≈ 11.23 ms, requiring just 4.72 mJ ($\approx 1.67 \cdot 10^{-6}$% of the UAV battery).

Our prototype implementation, whose source code has been released [17], leverages popular open-source libraries and tools, supported by the large variety of commercial UAVs. These features enhance the impact of *ARID*, demonstrating its deployability and ease of adoption, and paving the way for further research in the field.

Roadmap. The rest of this paper is organized as follows. Section 2 reviews related work, Section 3 introduces the *RemoteID* rule, Section 4 highlights the system and adversary models, Section 5 provides the details of *ARID*, Section 6 analyzes the security of our solution, while Section 7 reports the details of the implementation of the prototype and the performance assessment of *ARID*. Finally, Section 8 tightens conclusions and draws future works.

2 RELATED WORK

Only a few previous scientific contributions considered UAVs anonymity, with connection to authentication functionalities and unlinkability. For instance, the authors in [50] proposed a privacy-preserving authentication framework for Internet-connected drones, leveraging the emerging *Edge Computing* network architecture. Each drone interacts with an edge node for pseudonym generation, and the edge node maintains a translation map, allowing to switch from pseudonyms to real identities. Although being a feasible option for drones equipped with an Internet connection, the cited solution is not generalizable also for remote identification of amateur drones, which are likely not Internet-connected. In addition, such a solution requires assistance from dedicated infrastructure elements, not available everywhere.

Some contributions investigated anonymity through pseudonymity in the context of VANETs. For instance, the authors in [14] first tackled the problem of pseudonym-based authentication between two vehicles in a VANET. They proposed a hybrid scheme, combining locally-generating pseudonyms with group-based signatures, and the secrecy of the identity of a node mainly depends on the size of the group. However, such a proposal hardly fits with the scenario of amateur drones, where operators usually fly a drone without any coordination with other amateurs.

The authors in [42] proposed the adoption of hierarchical privacy-preserving pseudonyms, to be used in authentication procedures by a smart car. The initial pseudonym is released by a trusted authority, maintaining a database of the credentials assigned to the vehicles. When the vehicle needs to operate in a given area, it interacts with the local RSU to obtain new area-based pseudonyms that can be traced back to the original identity. The authors in [41] used a similar approach but focusing on efficient revocation mechanisms. The above schemes always require assistance from the infrastructure (RSU), being not applicable to our problem.

Many papers also investigated privacy-preserving pseudonym change strategies in VANETs, through *mix-zones*. Specifically, the approaches leveraging *mix-zones* use to change the pseudonym of a vehicle only when the number of vehicles in the neighbourhood is significantly high, so to confuse the attacker about new assignments. For instance, the authors in [11] proposed a scheme leveraging the RSUs at the road intersections to swap the pseudonyms of two vehicles. The same authors extended this concept in [13] considering Vehicular Location Privacy Zones (VPLZ), where vehicles access for service and exit in an order different than the entrance one. Similarly, the authors in [9] used the concepts of crowd areas and syntactic obfuscation jointly to confuse the attacker. Similarly, the authors in [34] proposed to swap pseudonyms in a group in a way to provide ϵ-differential privacy in the set of features of the vehicles. However, schemes of this type always require assistance from the infrastructure.

Despite sharing some features, the research challenges tackled in this paper are different from anonymous communications in VANETs. Indeed, privacy-preserving and secure remote identification for UAVs implies not broadcasting the long-term identity of the UAV indiscriminately on the wireless channel, as well as protecting the UAV from false invasion reports (see Section 4.2). Conversely, in the case of mutual anonymous authentication in VANETs, a vehicle receives a specific request from another entity and can decide if sharing its identity with the requesting vehicle on a case-by-case basis. Therefore, our scenario and adversary model do not consider anonymous mutual authentication, which is a different research problem.

Anonymous identification has also been considered in other contexts. For instance, in the avionic context, the authors in [6] proposed a mechanism for securely generating aircraft pseudonyms. They introduced a dedicated entity, namely the Trusted Registration Authority (TRA), in charge of assisting vehicles in generating time-bounded pseudonyms and able to trace back a pseudonym to its identity. However, their method requires continuous interaction with the authority for pseudonym generation. In addition, in case of leakages on the CA, the correspondence between the long-term identity and the pseudonyms is disclosed.

In the maritime context, the authors in [25] proposed the usage of a pseudonyms set generated by a trusted authority for each vessel, used once for every time slot. Such an approach avoids infrastructure assistance but requires a persistent connection from the vessel to the authority. The authors in [26] proposed to integrate IEEE P1609.02 pseudonymous generation and authentication features within the maritime domain. However, the IEEE P1609.02 standard mainly refers to unicast transactions, while the *Remote Identification* rule involves broadcast-only communications. Also, note that approaches such as [54] for secure and anonymous broadcast do not map to our problem, as they assume a secure communication channel among a set of parties.

We also highlight that our solution cannot be replaced by a signcryption protocol [55]. Although signcryption schemes allow decreasing the computational and communication overhead compared to sign-then-encrypt and hybrid solutions, their anonymous versions require heavy pairing operations [51], or polynomial interpolation [40], or proxies [31], not feasible on commercial UAVs, or using ring signatures, requiring a set of trusted parties [39],[15]. Therefore, they cannot be contextualized directly to our problem.

To sum up, the discussion above confirms that anonymous remote identification for UAVs is a different research problem, characterized by specific technology-dependent constraints. Such constraints make previous solutions unsuitable for this problem and call for new domain-specific solutions.

3 FAA REMOTE IDENTIFICATION RULE

With the widespread commercialization of autonomous and remotely-piloted UAVs, even more news of intentional and unintentional private-area invasion attacks are appearing. Indeed, UAVs equipped with a camera can record audio and video of sensitive areas. At the same time, when operated close to sensitive areas (e.g., airports and critical infrastructures), also small UAVs typically sold as toys for children, devoid of cameras and microphones, could cause collisions and create severe safety risks.

The cited incidents motivated avionics authorities to regulate the operation of UAVs in several ways. In this context, the US-based FAA has been the first to announce the publication of a *Remote Identification* (*RemoteID*) regulation, published in its final version in April, 2021 [23]. The scope of the *RemoteID* regulation is to integrate amateur and remotely-piloted drones into the local National Airspace System (NAS), to provide operators with an easy way to timely identify the presence of a UAV, its location, and the location of the related control station.

According to the RemoteID specification, UAVs must periodically broadcast messages containing at least the following information.

- A unique identifier of the identity of the drone.
- An indication of the drone's current location, expressed in terms of latitude, longitude, geometric altitude, and speed.
- The indication of the current location of the control station piloting the drone, expressed in terms of latitude, longitude, and geometric altitude.
- A timestamp of the message.
- An indicator of the emergency status of the drone.

The discussed requirements apply from the take-off to shutdown. UAVs should broadcast the messages on an unlicensed radio frequency (e.g., the worldwide ISM frequency band [2.4 – 2.5] GHz), with a transmission rate of at least 1 message per second and maximum allowed latency of 1 second from the location acquisition.

The specification applies independently from the weight of the UAV and the presence of an Internet connection on board or on the control station (exceptions exist [22]). Overall, the operators can assure the compliance of their UAVs to the *RemoteID* specification either through a built-in module or through a *remote ID broadcast module*, integrated after deployment on an UAV via a software update. Operation without a remote identification strategy in compliance with the *RemoteID* specification is possible only at specific areas, namely FAA-Recognized Identification Areas (FRIAs), governed by community-based organizations or educational institutions.

We notice that although the specification aims to enhance the safety and security of UAVs, the *RemoteID* specification does not take cybersecurity into account. Indeed, the specification aims to provide minimum performance requirement for the operation of UAVs, while the implementation of dedicated security and privacy strategies is left to UAV operators. Therefore, the specification does not mandate the usage of any message authentication techniques but only imposes that the FAA must be able to trace back the long-term identity of the UAV and its owner through the unique identifier broadcasted in the *RemoteID* messages.

According to the final rule, the unique identifier can be either the serial number of the UAV, generated by the manufacturer, or a session identification number (session ID). However, the session IDs must allow the FAA and authorized entities to go back to the long-term identity of the UAV. Note that the rule does not provide any further detail on the deployment of session identifiers.

The rule just became effective in April 2021 [43]. However, the UAV manufacturers will be required to comply with the direction starting from September 2022, while this requirement will apply to the pilots beginning from September 2023 [24].

4 SCENARIO AND ADVERSARIAL MODEL

This section describes the scenario and the adversary model considered in our work. Specifically, Section 4.1 illustrates the system model, while Section 4.2 describes the adversary model.

4.1 System Model and Assumptions

The scenario assumed in this work is depicted in Figure 1.

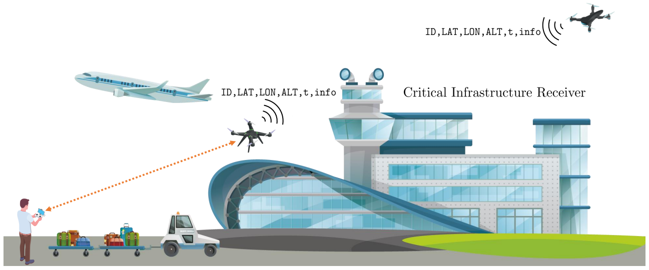

Figure 1: Scenario assumed in this work.

We assume a generic UAV d_n, that can be either an autonomous vehicle or a Remotely-Piloted Aircraft System (RPAS), piloted via a controller. We assume that d_n would like to be compliant with the latest specifications of the FAA on the remote identification. Therefore, d_n broadcasts information about its location, speed, location of the remote controller, time mark, emergency status, and a unique identifier, as presented in Section 3. However, d_n would like to maintain *anonymity*, i.e., it would like not to share publicly any information that would lead to the immediate and uncontrolled identification of its serial number, MAC address, and owner information. A solution to this issue, is for d_n to transmit a pseudonym that would allow only the authorized entities to disclose its real identity.

We also assume that d_n features enough processing capabilities to support the execution of symmetric and asymmetric encryption operations. This latter is a realistic assumption, since all the commercial drones available on the market feature CPUs able to control the motors and the flight through dedicated algorithms, much more expensive and complex than traditional symmetric and asymmetric encryption schemes, often available also through dedicated hardware support [38].

Without loss of generality, we assume that d_n does not feature an Internet connection and cannot communicate with any other infrastructure element than its remote controller. At the same time, in line with the FAA *RemoteID* specification, d_n uses one of the WiFi channels on the unlicensed frequency band $B = [2.4 - 2.5]$ GHz to communicate with the remote controller. In addition, we assume that d_n is fully compliant with the *RemoteID* specification. Therefore, it does not shut down the *RemoteID* module, and it does not falsify its location intentionally. To comply with the anonymity requirement, we also assume that any *RemoteID* packet delivered by d_n contain an anonymized (e.g., nullified) MAC address.

Our scenario also assumes the presence of multiple CI operators, e.g., the ones controlling airports, nuclear plants, military infrastructures, and other CIs. Such operators are interested in monitoring the nearby of their physical premises, looking for UAVs approaching and invading *sensitive* areas. To this aim, such entities leverage multiple receiving radios, tuned on the unlicensed frequency band $B = [2.4 - 2.5]$ GHz, to listen for *RemoteID* messages. Whenever any UAVs enter the monitored area, the legitimate receivers would like to detect the event and identify the owner of the UAV. The identification of the owner can occur through communication between the CI operator and a Trusted Authority (TA).

The TA is responsible for regulation and accountability of UAVs activities. Before flying time, manufacturers and UAV operators register the unique identity and additional information of their UAVs with the TA. If an invasion attack occurs, the CI operators can report the incident to the TA, providing the packets emitted from the invading UAV as evidence of the invasion. The TA is the only entity that can unveil the long-term identity of the UAVs carrying out the invasion, in a way that the UAVs and their operators can be accountable for the event.

In this context, the *ARID* solution we describe in this manuscript aims at ensuring anonymity and message authenticity for UAVs, protecting their long-term identity while operating in the wild. Moreover, *ARID* enables CI operators to report eavesdropped packets to the TA, to identify UAVs invading protected areas.

Table 1 reports the notation used throughout the manuscript.

Table 1: Notation used throughout the paper.

Notat.	Description
d_n	Generic *RemoteID*-compliant UAV.
B	Operation bandwidth of the UAV d_n.
$\mathcal{A}$	Adversary.
r	Generic Receiver of a CI operator.
$Auth$	Trusted Authority.
sk_n, pk_n	Private and Public Key of d_n.
C_n	Public-key certificate of d_n.
sk_A, pk_A	Private and Public Key of $Auth$.
C_A	Public-key certificate of $Auth$.
p	Prime number defining the size of the ECC field.
a, b	Parameters of the elliptic curve.
$\mathcal{G}$	Cyclic Group of the elliptic curve.
G	Generator point of the elliptic curve.
n	Order of the elliptic curve.
γ	Co-factor of the elliptic curve.
H	Hashing function.
S	Symmetric encryption algorithm.
E	Public-key encryption algorithm.
D	Public-key decryption algorithm.
$sign$	ECC public-key signature algorithm.
$verify$	ECC public-key signature algorithm.
T_i	*ARID* messages inter-arrival time.
$lat_{n,t}$, $lon_{n,t}$, $alt_{n,t}$	Latitude, longitude, and altitude of d_n at the time t.
$info_{n,t}$	Additional *RemoteID* information (speed of d_n, position of the control station, emergency status).
$h_{n,t}$	Digest generated by d_n at the time t.
$v_{n,t}$	Nonce generated by d_n at the time t.
$\delta_{n,t}$	Location signature generated by d_n at the time t.
$K_{n,t}$	Ephemeral key generated by d_n at the time t.
$c_{n,t}$	Ephemeral pseudonym generated by d_n at the time t.
$\rho_{n,t}$	Encrypted key generated by d_n at the time t.

4.2 Adversary and Threat Models

The adversary assumed in our work, namely $\mathcal{A}$ features both passive and active features. On the one hand, $\mathcal{A}$ is a global eavesdropper on the bandwidth $B = [2.4 - 2.5]$ GHz. Thus, $\mathcal{A}$ can detect and decode any message sent by UAVs on any of the channels in B to identify and track a specific UAV. On the other hand, $\mathcal{A}$ also features active capabilities. For instance, it can replay packets previously listened to and transmit rogue packets on the wireless communication channel, trying to impersonate a specific drone and falsifying its location reports. The combination of the passive and active features described above contribute to defining $\mathcal{A}$ as an adversary following the well-known Dolev-Yao attacker model [18].

Overall, the goal of the adversary is three-fold. First, the adversary would like to obtain the long-term identity of a specific UAV by simply listening to the broadcast *RemoteID* packets. Second, the adversary would like to track the drone passively. Third, the adversary would like to cheat the whole *RemoteID* system by making a specific UAV appear in a given sensitive location.

5 ANONYMOUS REMOTE IDENTIFICATION FRAMEWORK

This section provides all the details of the *ARID* scheme. Section 5.1 introduces the actors involved in *ARID*, while Section 5.2, Section 5.3, and Section 5.4 illustrate the Registration, Online, and Reporting Phase of *ARID*, respectively.

5.1 Actors

ARID involves the following actors.

- *UAV* d_n. It is a generic UAV, compliant with the *RemoteID* rule. While turned on, it emits on the bandwidth B anonymous *RemoteID* packets, i.e., messages not directly leading to its long-term identity, owner, and manufacturer.
- *CI Operator* r. It is a generic wireless receiver, deployed by the operator of a CI, such as an airport. Its role is to listen to the bandwidth B for wireless packets emitted by *RemoteID*-compliant UAVs to timely detect any invasion. When an invasion is detected, it reports such an event to the Authority for follow-up investigation.
- *Authority* $Auth$. It is a Trusted Third-Party (TTP), such as the Federal Avionic Administrator. Its role includes: (i) storing the information on UAVs compliant with *ARID*, (ii) providing cryptography materials for the correct operations of *ARID*, and finally, (iii) analyzing reports provided by CI operators on invasions of sensitive areas by UAVs, in a way to identify the authenticity of the provided evidence and trace back the long-term identity and owner of the UAV. Note that the Authority does not need to communicate with the UAV d_n after the Registration, while it is supposed to host an online service for reports submitted by CI operators.

5.2 Registration Phase

The aim of the *Registration Phase* of *ARID* is to register an UAV with the Authority, in a way to enable unique identification. At the same time, the UAV receives the cryptography materials necessary to run *ARID*. Figure 2 shows the sequence diagram of the *Registration Phase* of *ARID*. Note that the interactions between the UAV (or its owner, or its manufacturer) and the Authority occur through a regular Internet connection, e.g., secured via the well-known Transport Layer Security (TLS) protocol.

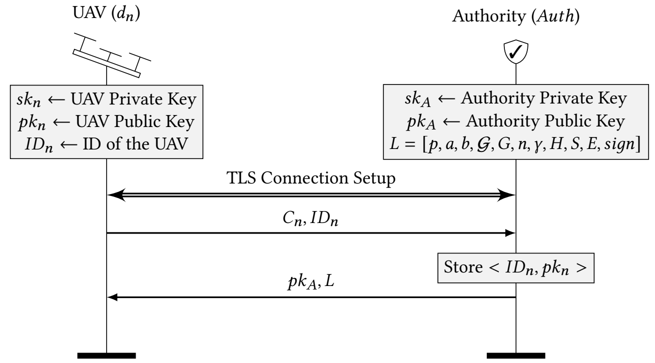
UAV (d_n)
Authority ($Auth$)
$sk_n \leftarrow$ UAV Private Key
$pk_n \leftarrow$ UAV Public Key
$ID_n \leftarrow$ ID of the UAV
$sk_A \leftarrow$ Authority Private Key
$pk_A \leftarrow$ Authority Public Key
$L = [p, a, b, \mathcal{G}, G, n, \gamma, H, S, E, sign]$
TLS Connection Setup
C_n, ID_n
Store $< ID_n, pk_n >$
pk_A, L

Figure 2: Sequence Diagram of the *Registration Phase* of *ARID*.

The operations in this phase are described below.

- Assume that the UAV d_n has a private-public key pair (sk_n, pk_n), and a public-key certificate C_n, signed by a TTP (e.g. Verizon [52]). After the establishment of the TLS connection, d_n provides to $Auth$ the long-term identity ID_n and the public key certificate C_n (including the public key pk_n).
- Assume that $Auth$ has a private-public key pair (sk_A, pk_A), and a public-key certificate C_A, signed by a Certification Authority (CA). At the reception of the information from d_n, $Auth$ stores in a local *Registration Table* the entry for D_n, containing the tuple $< ID_n, pk_n >$. Then, it provides to d_n the cryptography materials necessary to run *ARID*. These materials include the public key pk_A and the set of public parameters of the cryptography system L, including, in turn, the prime number p, the parameters a and b of the selected elliptic curve, the generator point G of the elliptic curve, the cyclic group $\mathcal{G}$ of the elliptic curve, the order n of the elliptic curve, and the co-factor γ. Moreover, L also includes the definition of the cryptography algorithms necessary to run *ARID*, i.e., the symmetric encryption scheme S, the public-key encryption scheme E (implying the corresponding decryption scheme D), the hashing function H, and the signature generation scheme $sign$ (implying the corresponding verification scheme $verify$).
- At the reception of the message from $Auth$, d_n locally stores the public key pk_A and the parameters L for use in the following *Online Phase*.

After the completion of this phase, d_n does not need any further communication with the Authority. When an UAV cannot connect to the Internet, the owner/operator can execute this phase on behalf of the UAV, e.g., on a secure Internet-connected terminal. However, the values of P_A and L have to be stored manually on d_n before any operations.

Moreover, note that the *Registration Table* hosted on the Authority can also be publicly available. Indeed, the information in the table do not allow tampering or de-anonymization of *ARID* messages. We will formally verify this property of *ARID* in Section 6.

5.3 Online Phase

During the *Online Phase* of *ARID*, the UAV d_n generates and emits *RemoteID*-compliant messages, enabling operators to identify their locations, while still preserving UAV anonymity. The sequence diagram of the operations executed by d_n every T_i seconds is depicted in Figure 3. Note that T_i is not fixed, and can vary randomly in the interval $\left[T_{i,MIN}, 1\right]$ seconds, with $T_{i,MIN}$ to be defined through real experiments in Section 7.

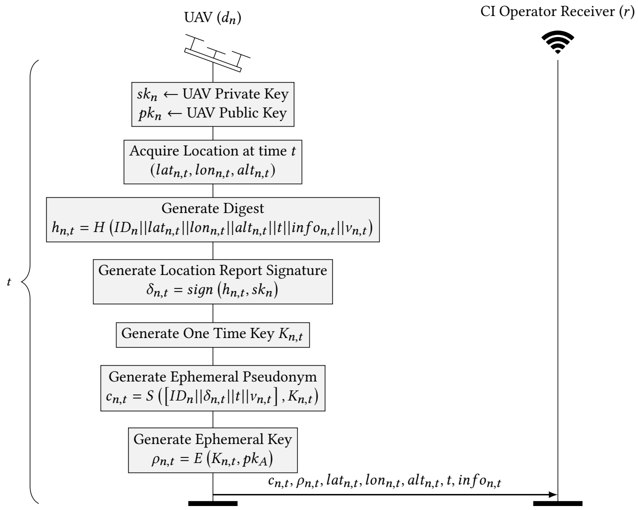
UAV (d_n)
CI Operator Receiver (r)
$sk_n \leftarrow$ UAV Private Key
$pk_n \leftarrow$ UAV Public Key
Acquire Location at time t
$(lat_{n,t}, lon_{n,t}, alt_{n,t})$
Generate Digest
$h_{n,t} = H\left(ID_n||lat_{n,t}||lon_{n,t}||alt_{n,t}||t||info_{n,t}||v_{n,t}\right)$
Generate Location Report Signature
$\delta_{n,t} = sign\left(h_{n,t}, sk_n\right)$
Generate One Time Key $K_{n,t}$
Generate Ephemeral Pseudonym
$c_{n,t} = S\left(\left[ID_n||\delta_{n,t}||t||v_{n,t}\right], K_{n,t}\right)$
Generate Ephemeral Key
$\rho_{n,t} = E\left(K_{n,t}, pk_A\right)$
$c_{n,t}, \rho_{n,t}, lat_{n,t}, lon_{n,t}, alt_{n,t}, t, info_{n,t}$
t

Figure 3: Sequence Diagram of the *Online Phase* of *ARID*.

Specifically, the UAV d_n executes the following operations.

- Assume that at the time t d_n acquires via Global Positioning System (GPS) the own location $\left[lat_{n,t}, lon_{n,t}, alt_{n,t}\right]$, in terms of latitude, longitude, and altitude. d_n first generates a digest $h_{n,t}$, according to Eq. 1.

$$h_{n,t} = H\left(ID_n||lat_{n,t}||lon_{n,t}||alt_{n,t}||t||info_{n,t}||v_{n,t}\right), \tag{1}$$

  where H refers to a generic secure hashing function, $info_{n,t}$ refers to the additional information included by d_n in the *RemoteID* packet (such as the speed and the position of the control station), $v_{n,t}$ is a nonce, and the operator || refers to the string concatenation.
- Then, d_n generates a *location report signature* $\delta_{n,t}$ as per Eq. 2.

$$\delta_{n,t} = sign\left(h_{n,t}, sk_n\right), \tag{2}$$

  being *sign* an ECC public-key signature algorithm (e.g., Elliptic Curve Digital Signature Algorithm (ECDSA)).
- Then, d_n generates a one-time ephemeral key $K_{n,t}$. Using such a key, d_n generates the ephemeral pseudonym $c_{n,t}$, as per Eq. 3.

$$c_{n,t} = S\left(\left[ID_n||\delta_{n,t}||t||v_{n,t}\right], K_{n,t}\right), \tag{3}$$

  where S refers to a generic symmetric encryption algorithm.
- Then, d_n generates the encrypted one-time key $\rho_{n,t}$, according to Eq. 4.

$$\rho_{n,t} = E\left(K_{n,t}, pk_A\right), \tag{4}$$

  being E a generic public-key encryption operation and pk_A the public-key of the Authority $Auth$.
- Finally, d_n delivers a broadcast *RemoteID* packet containing the ephemeral pseudonym $c_{n,t}$, the encrypted one-time key $\rho_{n,t}$, and all the mandatory *RemoteID* information, i.e., its latitude $lat_{n,t}$, longitude $lon_{n,t}$, altitude $alt_{n,t}$, the timestamp t, and the additional information $info_{n,t}$.
- The generic CI operator r continuously listens on the wireless channel. If r identifies a *RemoteID* packet (e.g., through analyzing network traffic via tools like *Wireshark*), it looks at the reported location of the UAV. If such a location is outside the protected area, r can simply discard the packet. Otherwise, r stores the packet locally and lately triggers the *Reporting Phase* (see Section 5.4).

5.4 Reporting Phase

The *Reporting Phase* is triggered exclusively by a CI operator, when it detects an invasion of the protected area by an UAV. Figure 4 shows the sequence diagram of the involved operations. Note that all the communications occur via a regular Internet connection, secured via the well-known TLS protocol.

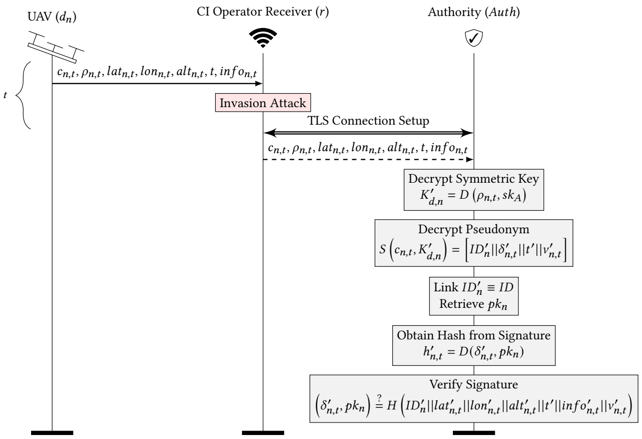
UAV (d_n)
CI Operator Receiver (r)
Authority ($Auth$)
$c_{n,t}, \rho_{n,t}, lat_{n,t}, lon_{n,t}, alt_{n,t}, t, info_{n,t}$
t
Invasion Attack
TLS Connection Setup
$c_{n,t}, \rho_{n,t}, lat_{n,t}, lon_{n,t}, alt_{n,t}, t, info_{n,t}$
Decrypt Symmetric Key
$K'_{d,n} = D\left(\rho_{n,t}, sk_A\right)$
Decrypt Pseudonym
$S\left(c_{n,t}, K'_{d,n}\right) = \left[ID'_n||\delta'_{n,t}||t'||v'_{n,t}\right]$
Link $ID'_n \equiv ID$
Retrieve pk_n
Obtain Hash from Signature
$h'_{n,t} = D(\delta'_{n,t}, pk_n)$
Verify Signature
$\left(\delta'_{n,t}, pk_n\right) \overset{?}{=} H\left(ID'_n||lat'_{n,t}||lon'_{n,t}||alt'_{n,t}||t'||info'_{n,t}||v'_{n,t}\right)$

Figure 4: Sequence Diagram of the *Reporting Phase* of *ARID*.

The operations executed in this phase are detailed below.

- Assume that at the time t the UAV d_n broadcasts a *RemoteID* message consistent with the format presented in Section 5.3, including the ephemeral pseudonym $c_{n,t}$, the encrypted one-time key $\rho_{n,t}$, the latitude $lat_{n,t}$, longitude $lon_{n,t}$, altitude $alt_{n,t}$ of the UAV, the timestamp t, and the additional information $info_{n,t}$.
- Assume also that the CI operator r receives the *RemoteID* message and verifies that the location of the UAV, in terms of latitude, longitude, and altitude, is reporting a position *inside* its restricted area, determining an *invasion.* Then, the CI operator r establishes a secure connection with the Authority $Auth$, and it reports the details of the message detected on

the communication channel, together with any additional local information.

- At the reception of the report from r, $Auth$ first obtains the reconstructed ephemeral key $K'_{d,n}$ of d_n, by decrypting the encrypted one-time key, as in Eq. 5.

$$K'_{d,n} = D\left(\rho_{n,t}, sk_A\right), \tag{5}$$

  being D the public-key decryption algorithm dual of the public-key encryption algorithm used by d_n. If the decryption is successful, $Auth$ proceeds further; otherwise, it discards the message.
- Then, using the reconstructed ephemeral key $K'_{d,n}$, $Auth$ obtains the reconstructed ephemeral pseudonym of d_n, by applying the operations in Eq. 6.

$$S\left(c_{n,t}, K'_{d,n}\right) = \left[ID'_n || \delta'_{n,t} || t' || v'_{n,t}\right] \tag{6}$$

  where S refers to the same symmetric encryption algorithms used by d_n, while the values ID'_n, $\delta'_{n,t}$, t', and $v'_{n,t}$ refer to the reconstructed values of the long-term identity of d_n, the location report signature, the generation timestamp, and the random nonce.
- $Auth$ verifies the consistency of the information retrieved from the ephemeral pseudonym. In particular, $Auth$ checks that the reconstructed timestamp t' matches precisely the value of t delivered in the report by the CI operator r. If they match, $Auth$ proceeds further; otherwise, it discards the message.
- Then, $Auth$ looks into the local *Registration Table* an entry for the UAV with long-term identity ID'_n. If a match is found, $Auth$ retrieves the corresponding registered public key pk_n; otherwise, it discards the message.
- Using the public key pk_n just retrieved, $Auth$ verifies the signature $\delta'_{n,t}$, by applying the check in Eq. 7.

$$verify\left(\delta'_{n,t}, pk_n\right) \stackrel{?}{=} H\left(ID'_n || lat'_{n,t} || lon'_{n,t} || alt'_{n,t} || t' || info'_{n,t} || v'_{n,t}\right) \tag{7}$$

  where $verify$ refers to the public-key signature verification algorithm dual of the public-key signature generation algorithm used by d_n. If $Auth$ verifies the signature, the report by r is considered *authentic*, and the UAV with long-term identity ID_n is deemed to be accountable for the invasion of the *restricted access* protected area (the owner can be contacted, charged, and blacklisted, based on the specific intrusion). Otherwise, the message is discarded as *not authentic*.

6 SECURITY ANALYSIS

This section discusses the security features offered by *ARID*. Specifically, Section 6.1 illustrates the security properties of *ARID*, while Section 6.2 provides the automated formal proof through *ProVerif*.

6.1 Security Considerations

Overall, *ARID* provides the following security properties.

UAV Anonymity. *ARID* ensures the complete anonymity of the UAVs while maintaining full compatibility with *RemoteID* regulations. Indeed, each message emitted by d_n at the time t in the *Online Phase* includes an ephemeral pseudonym $c_{n,t}$, that is linked to the long-term identity ID_n but changes for any emitted message, due to the dependence from the timestamp t (included in the signature $\delta_{n,t}$) and the nonce $v_{n,t}$. The continuous renewal of the pseudonym and the usage of the public-key cryptography scheme E also provide *message unlinkability*, i.e., an adversary intercepting any two messages cannot distinguish if they have been emitted by the same UAV or by different UAVs. Note that, differently from other schemes available in the literature, *ARID* achieves anonymity without any assistance from external infrastructure elements, and neither it requires a continuous connection with the Authority (see Section 7.3 for more details). Moreover, we emphasize that this property still holds even when the *Registration Table* hosted on the Authority is public. Indeed, the only entity that can unveil the long-term identity of d_n is the Authority, using its private key sk_A. The *Anonymity* property of *ARID* has been also verified in *ProVerif* (see Section 6.2).

UAV Message Authenticity. *ARID* provides message authenticity to *RemoteID* messages, protecting against impersonation and message manipulation. Message authenticity is provided through the inclusion of the *location report signature* $\delta_{n,t}$, generated starting from: (i) the long-term identity of the UAV ID_n; (ii) all the information publicly-disclosed by the UAV in the *RemoteID* message; and, (iii) the private key sk_n. Given that d_n is the only entity possessing sk_n, only d_n can generate the location report signature $\delta_{n,t}$ associated to a particular *RemoteID* message, assuring message authenticity. Indeed, let us assume that the attacker modifies any plain-text information (location of the drone, speed, timestamp, ground station location, or emergency status). In the cited case, the verification of the signature $\delta_{n,t}$ will fail, leading to attack rejection. Similarly to the previous property, this feature holds even if the *Registration Table* is public, as formally verified in Section 6.2.

Protection against Replay Attacks. Being *RemoteID* messages broadcast, *ARID* cannot provide formal protection against replay attacks. Indeed, there is no interaction between the UAV and other entities that could ensure the messages' freshness. To identify replayed messages, *ARID* leverages the consistency among the timestamp in the broadcast message and the current time. Indeed, most UAVs feature a GPS receiver, used to obtain global synchronization. If a CI operator receives a message with a timestamp whose difference to the UTC time is higher than a threshold τ, such message is discarded, failing a freshness check. Note that an attacker could modify the timestamp t included in a broadcast message previously recorded to match it to the actual time, making a UAV appear at an old location at the current time. However, when such a message is reported to the Authority $Auth$, the Authority can easily verify the manipulation of the message, as the verification of the signature $\delta_{n,t}$ will fail.

Partial Protection against UAV tracking. From the security perspective, *ARID* also provides theoretical protection against UAV tracking. Indeed, not only an UAV never reveals the long-term identity, but it also uses an ephemeral pseudonym only once, not allowing to link two *RemoteID* messages. At the same time, we notice that a control station could also control many UAVs at the same time. Therefore, an adversary might not track a specific UAV with 100% accuracy by checking the location of the control station. This is evident when the adversary does not have any additional knowledge of the scenario, i.e., it does not know how many UAVs

are operating in the area.
Although these remarkable features, many contributions framed in the context of VANET pointed out the possibility to distinguish vehicles and track them based on the peculiar characteristics of their trajectories, with different degrees of accuracy [46], [30]. Recognizing the likely partial effectiveness of the mentioned tracking techniques, we define the protection against tracking offered by *ARID* as *partial*. However, anti-tracking is not the scope of *ARID*, and its applicability to tracking avoidance is left for future work.

6.2 Formal Verification — ProVerif

The security properties provided by *ARID*, i.e., UAV anonymity and message authenticity, have been formally verified via *ProVerif* [10]. *ProVerif* is an automated verification tool widely adopted in the recent literature to formally verify the security properties achieved by cryptographic protocols [49], [32], [5].

Specifically, *ProVerif* assumes the Dolev-Yao attacker model, i.e., the attacker can read, modify, delete, and forge new packets to be delivered on the communication channel. Under the cited assumptions, *ProVerif* checks if the attacker can break the security goals of the protocol defined by the user. In case an attack is found, *ProVerif* also provides a step-by-step description of the attack.

We implemented *ARID* in *ProVerif* to verify two main properties: (i) the secrecy of the long-term identity of the UAV; and, (ii) the authenticity of the messages emitted by an UAV. Therefore, according to the logic of the *ProVerif* tool, we defined two main events.

(1) *acceptUAV(id)*: Indicating that the UAV with long-term identity ID_n is running *ARID*.
(2) *termAuth(id)*: Denoting that the Authority has terminated *ARID* and verified that the UAV with the long-term identity ID_n generated the message.

In line with the logic of *ProVerif*, we verified the UAV message authenticity property through verifying security properties such as *sender authentication* and *impersonation resistance*. To this aim, we checked that *event(acceptUAV(id))* cannot be executed after the execution of *event(termAuth(id))*. Moreover, we verified the strong secrecy of the long-term identity of ID_n, by verifying that the attacker is unable to distinguish when the secret changes, and that the attacker cannot obtain ID_n from the messages exchanged on the wireless communication channel.

The following output messages are provided by *ProVerif* to identify the fulfillment of the security properties of our interest.

- *event(last_event ()) ==> event(previous_event ()) is true*: meaning that the function *last_event* is executed only when another function, namely *previous_event*, is really executed;
- *not attacker(elem[]) is true*: meaning that the attacker is not in possession of the value of *elem*;
- *Non-interference elem[] is true*: meaning that an attacker cannot deduce any information about the value of *elem* from the eavesdropped messages.

The excerpt of the output of the *ProVerif* tool when ID_n is not public (regular condition) is shown in Figure 5.

The fulfilment of the query in Figure 5 demonstrates that the Authority always verifies message authenticity, i.e., when it detects that a message has been generated by ID_n, this is always true. Note that the correspondence in the query is not injective, because the attacker could have replayed the message. We handled replays *artificially* in *ProVerif*, through the verification of the freshness of the timestamp. Also, note that the *Non-interference* query is verified, i.e., an attacker cannot deduce any information about ID_n from the eavesdropped messages.

```
Verification summary:
Query not ID_n[] is true.
Query event(termAuth(ID_1)) ==> event(acceptUAV(ID_1)) is true.
Non-interference ID_n is true.
```

Figure 5: Excerpt of the output provided by the *ProVerif* tool when ID_n is not public.

As a side-property of *ARID*, we also checked if message authenticity still holds when the information provided to the Authority are public, i.e., the public key of the UAV pk_n and its identity ID_n. The excerpt of the output of the *ProVerif* tool in this case is shown in Figure 6.

```
Verification summary:
Query not ID_n[] is false.
Query event(termAuth(ID_1)) ==> event(acceptUAV(ID_1)) is true.
```

Figure 6: Excerpt of the output provided by the *ProVerif* tool when ID_n is public.

Note that, even when ID_n becomes public (e.g., due to the publication of the *Registration Table*), still message authenticity holds, meaning that the information in the *Registration Table* is not valuable for the attacker to impersonate any UAVs.

We also released the generated *ProVerif* source code, to allow interested readers to verify our claims and further re-use our code.

7 PERFORMANCE EVALUATION

This section provides the performance evaluation of *ARID*, both in qualitative and in quantitative terms. Section 7.1 reports the implementation details of the proof-of-concept, Section 7.2 illustrates the performance of *ARID*, while Section 7.3 qualitatively compares *ARID* with the approaches discussed in Section 2.

7.1 Implementation Details

We implemented a prototype of *ARID* on the 3DR-Solo commercial drone [3]. The 3DR-Solo hardware platform features a CPU *i.MX6 Solo* manufactured by *Freescale System*, connected to a *Pixhawk autopilot*. It also includes a single-core processor ARM Cortex A9 running at 1.00 GHz, and it is equipped with 7, 948 MB of ROM and 512 MB of RAM. The 3DR-Solo drone features a Cryptographic Acceleration and Assurance Module (CAAM), as well as True and Pseudo-Random Number Generator modules (certified by NIST), useful to execute Elliptic Curve Cryptography (ECC) primitives efficiently. As for the Operative System (OS), the 3DR-Solo runs the *3DR Poky* OS, based on the popular Linux Project Yocto [1].

We implemented *ARID* in *C*, and we integrated it within the stock *3DR Poky* OS, version 1.5.1. In particular, *ARID* runs on top of the popular protocol MAVLink 1.0 [33] and on UDP, using the lightweight Micro Air Vehicle Message Marshalling Library [2], i.e., a highly optimized library for resource-constrained systems. We recall that MAVLink is a lightweight messaging protocol, supported by the majority of commercially available UAVs, enabling communication among UAVs and between a UAV and its on-board components. We also recall that MAVLink over UDP requires the exchange of frames characterized by a Maximum Transmission Unit (MTU) of 263 bytes. We report the structure of the MAVLink frame (customized to include the data of *ARID* as payload) in Figure 9 of Annex 8, while Table 3 (Annex 8) provides additional configuration details. To implement *ARID* in a fully standard-compliant fashion, we extended MAVLink with a dedicated Message ID (`0xDE`). Moreover, each UAV features a static ID of 4 bytes provided from the manufacturer, spanning in the range [`0x00000000` − `0xFFFFFFFF`].

We used El-Gamal elliptic curves (with point compression) for encryption/decryption operations, and the ECDSA algorithm for signature generation/verification [27]. We integrated the cited algorithms through the OpenSSL library ver. 1.0.0 [37].

For the experimental evaluation, as well as to allow a complete customization of the offered security services, we selected four elliptic curves, i.e., *secp160r1, secp192k1, secp224k1* and *secp256k1*, providing security levels equivalent to 80, 96, 112 and 128 symmetric key bits, according to the most recent NIST guidelines [7] (Table 4 (Annex 8) provides additional configuration details). Moreover, we adopted the *SHA-256* hashing function and a cryptographic Pseudo Random Number Generator (PRNG) (*/dev/urandom*) seeded with 2,048 bits. We selected the cited four curves because they provide an adequate level of security for different scenarios, while also allowing not to exceed the MAVLink MTU of 263 bytes. Additional *larger* curves could be used but at the cost of message fragmentation (not desirable). Finally, we implemented the generic receiver r and the Authority $Auth$ as separated processes on a regular laptop.

Our implementation on the 3DR-Solo requires 1,559.168 KB of Flash Memory (with a static linking of the adopted libraries) and 92.184 KB of RAM. We also released the source code of *ARID* to allow interested Researchers and Industry to verify our claims and possibly extend *ARID* with additional features [17].

Finally, we remark that our implementation leverages popular open-source tools, such as the Poky OS, MAVLink, and OpenSSL, supported by a large variety of commercial UAVs. The availability of the code contributes to enhance the impact of *ARID*, demonstrating its deployability, and fostering further research in the domain.

7.2 Performance Assessment

In this section, we report a few experimental tests performed using the implementation discussed in Section 7.1, aimed at measuring the cost of *ARID* on a real UAV in terms of time and energy.

We first measured the time needed to generate and transmit a *ARID* packet on the 3DR-Solo, by considering the four elliptic curves cited in Section 7.1. We report in Figure 7 the average time required to execute *ARID* over 1,000 tests (with 95% confidence intervals), considering the separate contribution of the processing (packet generation, cryptography operations) and radio operations. Note that the measured time spans from the GPS location acquisition to the packet delivery (both included). In the worst case (curve *secp256k1*), *ARID* requires 11.23 ms on average, ≈ 2 orders of magnitude less than the maximum interarrival time $T = 1$ s recommended by the *RemoteID* rule, confirming its suitability for integration in real UAVs.

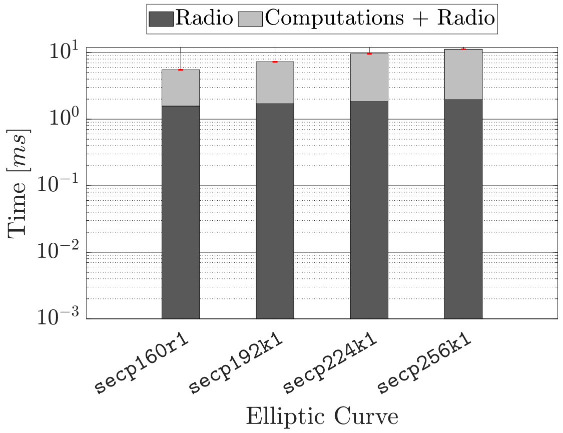

Figure 7: Time required to execute *ARID* on the 3DR-Solo drone, considering different elliptic curves.

To measure the energy consumption of each instance of *ARID*, we used the telemetry data conveyed by the 3DR-Solo to the remote controller through the MAVLink protocol. In detail, we measured the difference in the electrical current drained by the drone between two different states: (i) at rest; and, (ii) during the execution of *ARID*. We computed an average difference of ≈ 20 mA in the electric current drained by the drone over 1,000 runs.

To estimate the energy consumption of the radio operations, we considered that the radio chip on-board of the 3DR-Solo drone is a chip of the family *AR9300*, working with an input voltage of 3.3 V, consuming 296.970 mA in TX mode and 187.879 mA in RX mode with the IEEE 802.11b protocol [29]. We also assumed that a packet is modulated through the standard Direct Sequence Spread Spectrum (DSSS) modulation using Differential Binary Phase-Shift Keying (DBPSK), a Transmission Rate of 1.0 Mbps on the 22 MHz channel bandwidth, and a Short Guard Interval of 800 ns. We computed the contributions of the processing and radio chip to the overall energy consumption of *ARID* through Eq. 8.

$$E[mJ] = V \cdot \int_0^T i(t)dt, \tag{8}$$

being V the input voltage (15.11 V for the UAV's battery and 3.3 V for the radio chip) and $i(t)$ the instantaneous drained current (additional 20 mA required by *ARID* on the UAV's battery and 296.970 mA for the radio chip).

Figure 8 reports the average results of our experiments, together with the 95% confidence interval, computed over 1,000 tests. Table 4 in Appendix B also reports in a tabular form the values of time and energy consumption of Figure 7 and Figure 8.

Taking as a reference the worst-case configuration (curve *secp256k1*), *ARID* consumes only ≈ 4.72 mJ per instance (i.e., delivered *ARID*

Table 2: Comparison between *ARID* and anonymity solutions available in the literature.

Ref.	No Online Authority	No Infrastructure Assistance	No Group Reliance	No Revocation Procedures	No Pairing Operations	Robustness to TA DB Leakages
[50]	✓	–	✓	–	✓	–
[14]	✓	✓	–	–	–	–
[42]	✓	–	✓	–	–	–
[11]	–	✓	–	–	–	–
[34]	✓	–	–	–	✓	–
[13]	✓	–	–	–	✓	–
[9]	✓	–	✓	–	✓	–
[41]	–	–	✓	–	–	–
[6]	–	✓	✓	–	✓	–
[25]	–	✓	✓	–	✓	–
ARID	✓	✓	✓	✓	✓	✓

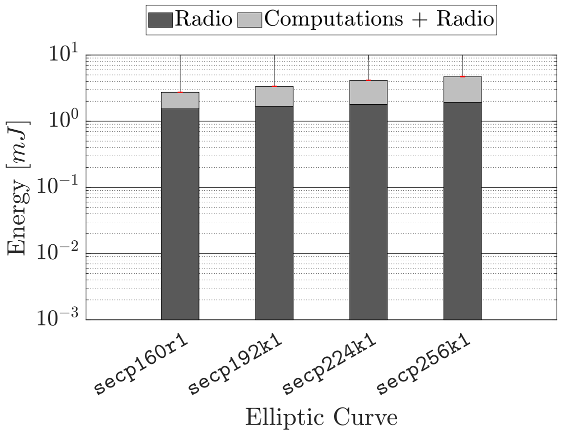

Figure 8: Energy required to execute *ARID* on the 3DR-Solo drone, considering different elliptic curves.

packet), confirming once more its little impact on the UAV's operating life. Indeed, given that the overall capacity of the battery powering the 3DR-Solo drone is $282,860$ J ($5,200$ mAh), assuming to adopt the curve *secp256k1*, *ARID* consumes on average only $\approx 1.67 \cdot 10^{-6}\%$ of the battery of the drone for each instance.

Finally, we evaluated the impact of *ARID* on the battery lifetime. We experimentally verified that the most energy-consuming configuration of *ARID* (*secp256k1*) reduces the lifetime of the 3DR-Solo by only 1.05% compared to the default (non-anonymous) *RemoteID* configuration, further demonstrating its limited overhead.

7.3 Comparison

In this section, we compare *ARID* against current approaches for anonymization as per what discussed in Section 2. Note that such a comparison can only be qualitative, i.e., along reference system requirements. Indeed, any quantitative comparison does not apply to our case, as all the cited solutions require assumptions that cannot be satisfied in our scenario, such as the presence of online authorities and additional infrastructure elements or peers.

Table 2 summarizes the discussion in Section 2, and compares the cited contributions along reference system requirements.

We notice that previously published approaches are not complaint with the requirements for anonymous remote UAVs identification. Indeed, most of them provide anonymity by assuming the continued availability of either an online authority or an element of the infrastructure (such as the RSUs in VANETs) or the presence of multiple peers in the network (the other vehicles in a VANET). These assumptions are not realistic for commercial UAVs, often piloted by independent pilots. At the same time, the approaches previously proposed can require complex pairing operations, hardly supported by small UAVs, as well as pseudonyms revocation procedures in case of misbehaviour, that would require connection to additional infrastructure elements. Moreover, note that all previous approaches leveraged a list stored on the TA, used to translate pseudonyms into long-term identities. Therefore, in case of a leakage/publication of the list, these solutions cannot further provide the anonymity of the participating entities.

Conversely, *ARID* provides anonymous remote identification for UAVs without any assistance, either from an online authority or from a dedicated network infrastructure, and it also applies to independent vehicles, with no other entities in the neighbourhood. In addition, *ARID* does not require time- and energy- consuming pairing operations. Furthermore, although the TA of *ARID* stores the long-term identities of the UAV and the related public keys, the long-term identities are always protected at run-time through the public key of the Authority. Therefore, even in case of a leakage on the TA, the anonymity of the registered drones is preserved, provided that the TA private key(s) are kept secret. The combination of all these features makes *ARID* the ideal solution for anonymous remote identification of amateur, remotely-piloted UAVs.

8 CONCLUSION

In this paper, we proposed *ARID*, a lightweight and low-cost protocol providing anonymous remote identification of Unmanned Aerial Vehicles. *ARID* has been carefully designed to be fully compliant with the latest *RemoteID* regulations by the FAA, while also providing a tunable level of security. Overall, *ARID* offers complete anonymity and unlinkability of UAVs broadcast messages, allowing only the Trusted Authority (e.g., the FAA) to unveil the long-term identity of the emitting UAV. At the same time, *ARID* does not require interactions between UAVs and other infrastructure elements or peers, and it can be provided as a simple software update.

While the security properties of *ARID* have been discussed and formally proved via *ProVerif*, we also implemented a prototype of *ARID* on a real 3DR-Solo drone, using the open-source Poky OS and well-known OpenSSL cryptography library. Our experimental performance evaluation shows that *ARID* requires at most only ≈ 11.23 ms to create and transmit anonymous *RemoteID* messages, while spending at most ≈ 4.72 mJ of energy ($\approx 1.67 \cdot 10^{-6}\%$ of the overall battery capacity).

We also released the source code of *ARID* [17], enabling the interested community to verify our findings, as well as to foster further research in the domain.

Future work include the extension of *ARID* to other domains, such as avionics and maritime, enriched with the capability to authenticate the *ARID* messages without compromising UAV's anonymity.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers, that helped improving the quality of the paper. This publication was partially supported by awards NPRP-S-11-0109-180242 from the QNRF-Qatar National Research Fund, a member of The Qatar Foundation, and NATO Science for Peace and Security Programme - MYP G5828 project "SeaSec: DronNets for Maritime Border and Port Security". This work has been partially supported also by the INTERSCT project, Grant No. NWA.1162.18.301, funded by Netherlands Organisation for Scientific Research (NWO). The findings reported herein are solely responsibility of the authors.

REFERENCES

[1] 3D Robotics. 2020. Yocto Linux. https://tinyurl.com/y2axm74b. (Accessed: 2021-06-21).

[2] 3DR Robotics. 2014. MAVlink Protocol Setup for Solo. https://github.com/3drobotics/mavlink-solo. (Accessed: 2021-06-21).

[3] 3DR Solo Website. 2020. 3DR Solo Website. https://3dr.com/solo-drone. Accessed: 2021-06-21.

[4] AIN Online. 2021. NBAA: Remote ID Drone Rule Raises Privacy Concerns. https://tinyurl.com/3vzzv39d. (Accessed: 2021-06-21).

[5] Thibaud Antignac, Mukelabai Mukelabai, and Gerardo Schneider. 2017. Specification, Design, and Verification of an Accountability-Aware Surveillance Protocol. In *Proceedings of the Symposium on Applied Computing* (Marrakech, Morocco) *(SAC '17)*. Association for Computing Machinery, New York, NY, USA, 1372–1378.

[6] Amirhossein Asari, Mahdi R. Alagheband, Majid Bayat, and Maryam Rajabzadeh Asaar. 2021. A new provable hierarchical anonymous certificateless authentication protocol with aggregate verification in ADS-B systems. *Computer Networks* 185 (2021), 107599.

[7] Elaine Barker. 2020. *Recommendation for key management: Part 1 - General.* Technical Report. NIST.

[8] BBC. 2019. Gatwick Airport: Drone attack grounds flights. http://www.bbc.co.uk/news/uk-england-sussex-4662375. (Accessed: 2021-06-21).

[9] L. Benarous, B. Kadri, and S. Boudjit. 2020. Alloyed Pseudonym Change Strategy for Location Privacy in VANETs. In *IEEE 17th Annual Consumer Communications Networking Conference (CCNC)*. 1–6.

[10] Bruno Blanchet. 2009. Automatic Verification of Correspondences for Security Protocols. *Journal of Computer Security* 17, 4 (2009), 363–434.

[11] A. Boualouache and S. Moussaoui. 2014. S2SI: A Practical Pseudonym Changing Strategy for Location Privacy in VANETs. In *International Conference on Advanced Networking Distributed Systems and Applications*. 70–75.

[12] Abdelwahab Boualouache, Sidi-Mohammed Senouci, and Samira Moussaoui. 2017. A Survey on Pseudonym Changing Strategies for Vehicular Ad-Hoc Networks. *IEEE Communications Surveys & Tutorials* 20, 1 (2017), 770–790.

[13] A. Boualouache, S. M. Senouci, and S. Moussaoui. 2020. PRIVANET: An Efficient Pseudonym Changing and Management Framework for Vehicular Ad-Hoc Networks. *IEEE Transactions on Intelligent Transportation Systems* 21, 8 (2020), 3209–3218.

[14] Giorgio Calandriello, Panos Papadimitratos, Jean-Pierre Hubaux, and Antonio Lioy. 2007. Efficient and Robust Pseudonymous Authentication in VANET. In *Proceedings of the Fourth ACM International Workshop on Vehicular Ad Hoc Networks* (Montreal, Quebec, Canada) *(VANET '07)*. 19–28.

[15] Yu Fang Chung, Zhen Yu Wu, and Tzer Shyong Chen. 2009. Ring signature scheme for ECC-based anonymous signcryption. *Computer Standards & Interfaces* 31, 4 (2009), 669–674.

[16] European Commission. 2021. *Detailed rules on unmanned aircrafts.* Technical Report. European Commission.

[17] Cybersecurity Research and Innovation Lab (CRI-LAB). 2021. Source code of ARID. https://github.com/pietrotedeschi/arid.

[18] Danny Dolev and Andrew Yao. 1983. On the security of public key protocols. *IEEE Transactions on information theory* 29, 2 (1983), 198–208.

[19] Drone DJ. 2021. Crowdfunded lawsuit filed against the FAA over Remote ID. https://tinyurl.com/euy342jr. (Accessed: 2021-06-21).

[20] Drone DJ. 2021. Letter to FAA: Please reconsider your Remote ID proposal. https://tinyurl.com/r7v24eva. (Accessed: 2021-06-21).

[21] ETSI. 2012. Intelligent Transport Systems (ITS); Security; Trust and Privacy Management, document TS 102 941, v1.1.1.

[22] FAA. 2021. Operations Over People General Overview. Available Online: https://www.faa.gov/uas/commercial_operators/operations_over_people/.

[23] FAA. 2021. Remote Identification of Unmanned Aircraft. Available Online: https://www.faa.gov/news/media/attachments/RemoteID_Final_Rule.pdf.

[24] FAA. 2021. UAS Remote Identification Overview. Available Online: https://www.faa.gov/uas/getting_started/remote_id/.

[25] Athanassios Goudossis and Sokratis K Katsikas. 2019. Towards a secure automatic identification system (AIS). *Journal of Marine Science and Technology* 24, 2 (2019), 410–423.

[26] John Hall, Jordan Lee, Joseph Benin, Christopher Armstrong, and Henry Owen. 2015. IEEE 1609 influenced automatic identification system (AIS). In *IEEE 81st Vehicular Technology Conference (VTC Spring)*. IEEE, 1–5.

[27] Darrel Hankerson, Alfred J Menezes, and Scott Vanstone. 2006. *Guide to Elliptic Curve Cryptography.* Springer Science & Business Media.

[28] Mostafa Hassanalian and Abdessattar Abdelkefi. 2017. Classifications, applications, and design challenges of drones: A review. *Progress in Aerospace Sciences* 91 (2017), 99–131.

[29] Stratos Keranidis, Giannis Kazdaridis, Nikos Makris, Thanasis Korakis, Iordanis Koutsopoulos, and Leandros Tassiulas. 2014. Experimental Evaluation and Comparative Study on Energy Efficiency of the Evolving IEEE 802.11 Standards. In *Proc. of Int. Conf. on Future Energy Systems*. 109–119.

[30] Sanaz Khakpour, Richard W Pazzi, and Khalil El-Khatib. 2017. Using clustering for target tracking in vehicular ad hoc networks. *Vehicular communications* 9 (2017), 83–96.

[31] Muhammad Asghar Khan, Habib Shah, Sajjad Ur Rehman, Neeraj Kumar, Rozaida Ghazali, Danish Shehzad, and Insaf Ullah. 2021. Securing Internet of Drones with Identity-based Proxy Signcryption. *IEEE Access* (2021), 1–1.

[32] Nadim Kobeissi, Georgio Nicolas, and Karthikeyan Bhargavan. 2019. Noise Explorer: Fully Automated Modeling and Verification for Arbitrary Noise Protocols. In *2019 IEEE European Symposium on Security and Privacy (EuroS&P)*. 356–370.

[33] Anis Koubâa, Azza Allouch, Maram Alajlan, Yasir Javed, Abdelfettah Belghith, and Mohamed Khalgui. 2019. Micro Air Vehicle Link (MAVlink) in a Nutshell: A Survey. *IEEE Access* 7 (2019), 87658–87680.

[34] X. Li, H. Zhang, Y. Ren, S. Ma, B. Luo, J. Weng, J. Ma, and X. Huang. 2020. PAPU: Pseudonym Swap With Provable Unlinkability Based on Differential Privacy in VANETs. *IEEE Internet of Things Journal* 7, 12 (2020), 11789–11802.

[35] Chao Lin, Debiao He, Neeraj Kumar, Kim-Kwang Raymond Choo, Alexey Vinel, and Xinyi Huang. 2018. Security and privacy for the internet of drones: Challenges and solutions. *IEEE Communications Magazine* 56, 1 (2018), 64–69.

[36] Mohammad Mozaffari, Walid Saad, Mehdi Bennis, Young-Han Nam, and Mérouane Debbah. 2019. A Tutorial on UAVs for Wireless Networks: Applications, Challenges, and Open Problems. *IEEE Communications Surveys & Tutorials* 21, 3 (2019).

[37] OpenSSL Found. 2021. OpenSSL - Cryptography and SSL/TLS Toolbox. https://www.openssl.org/. (Accessed: 2021-06-21).

[38] Muslum Ozgur Ozmen and Attila A. Yavuz. 2018. Dronecrypt - An Efficient Cryptographic Framework for Small Aerial Drones. In *MILCOM 2018 - 2018 IEEE Military Communications Conference (MILCOM)*. 1–6.

[39] Liaojun Pang and Huixian Li. 2013. nMIBAS: a novel multi-receiver ID-based anonymous signcryption with decryption fairness. *Computing and Informatics* 32, 3 (2013), 441–460.

[40] Liaojun Pang, Mengmeng Wei, and Huixian Li. 2019. Efficient and anonymous certificateless multi-message and multi-receiver signcryption scheme based on ECC. *IEEE Access* 7 (2019), 24511–24526.

[41] J. Qi and T. Gao. 2020. A Privacy-Preserving Authentication and Pseudonym Revocation Scheme for VANETs. *IEEE Access* 8 (2020), 177693–177707.

[42] U. Rajput, F. Abbas, and H. Oh. 2016. A Hierarchical Privacy Preserving Pseudonymous Authentication Protocol for VANET. *IEEE Access* 4 (2016), 7770–7784.

[43] Reuters. [n.d.]. https://tinyurl.com/tyda32k8. (Accessed: 2021-06-21).

[44] Reuters. 2021. Saudi-led coalition says it intercepts armed drone fired at Abha airport. https://tinyurl.com/2t9t9sf5. (Accessed: 2021-06-21).

[45] Savio Sciancalepore, Omar Adel Ibrahim, Gabriele Oligeri, and Roberto Di Pietro. 2019. Detecting Drones Status via Encrypted Traffic Analysis. In *Proceedings of the ACM Workshop on Wireless Security and Machine Learning* (Miami, FL, USA) *(WiseML 2019)*. Association for Computing Machinery, New York, NY, USA, 67–72.

[46] Mingshun Sun, Ming Li, and Ryan Gerdes. 2017. A data trust framework for VANETs enabling false data detection and secure vehicle tracking. In *IEEE Conference on Communications and Network Security (CNS)*. IEEE, 1–9.

[47] Pietro Tedeschi, Gabriele Oligeri, and Roberto Di Pietro. 2020. Leveraging Jamming to Help Drones Complete Their Mission. *IEEE Access* 8 (2020), 5049–5064.

[48] Pietro Tedeschi, Savio Sciancalepore, and Roberto Di Pietro. 2021. Modelling a Communication Channel under Jamming: Experimental Model and Applications. In *IEEE International Conference on Security, Privacy, and Anonymity in Computation, Communication, and Storage (IEEE SpaCCS)* (New York, USA).

[49] Pietro Tedeschi, Savio Sciancalepore, Areej Eliyan, and Roberto Di Pietro. 2020. LiKe: Lightweight certificateless key agreement for secure IoT communications. *IEEE Internet of Things J.* 7, 1 (2020), 621–638.

[50] Yifan Tian, Jiawei Yuan, and Houbing Song. 2019. Efficient privacy-preserving authentication framework for edge-assisted Internet of Drones. *Journal of Information Security and Applications* 48 (2019), 102354.

[51] Yuh-Min Tseng, Yi-Hung Huang, and Hui-Ju Chang. 2014. Privacy-preserving multireceiver ID-based encryption with provable security. *International Journal of Communication Systems* 27, 7 (2014), 1034–1050.

[52] Verizon Wireless. [n.d.]. https://tinyurl.com/9ztrt3fh. (Accessed: 2021-06-21).

[53] VOA News. 2021. Drone Attack Damages Hangar at US-Coalition Air Base in Iraq. https://tinyurl.com/33s7yber. (Accessed: 2021-06-21).

[54] Mahdi Zamani, Jared Saia, Mahnush Movahedi, and Joud Khoury. 2013. Towards Provably-Secure Scalable Anonymous Broadcast. In *3rd {USENIX} Workshop on Free and Open Communications on the Internet ({FOCI} 13)*.

[55] Yuliang Zheng and Hideki Imai. 1998. How to construct efficient signcryption schemes on elliptic curves. *Inform. Process. Lett.* 68, 5 (1998), 227–233.

ANNEX A: *ARID* PACKETS FORMAT

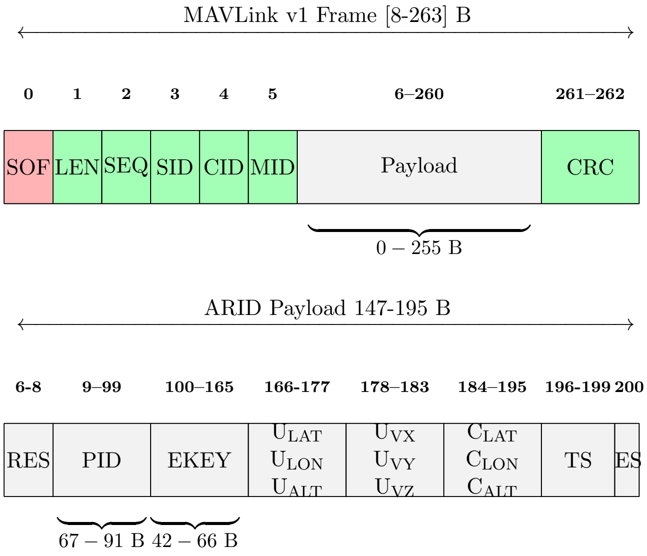

Figure 9: MavLink frame format and *ARID* payload content. Refer to Table 3 for related details.

Table 3: MAVLink Frame and ARID Payload Notation.

Acronym	Content/Size	Description
SOF	`0xFE`	MAVLink 1.0. Start of Frame
LEN	`0x00-0xFF`	Payload Length.
SEQ	`0x00-0xFF`	Sequence number. The value `0x00` represents the first message.
SID	`0x01-0xFF`	System Identification number of the UAV.
CID	`0x00-0xFF`	System Identification number of the component that is transmitting the message.
MID	`0x00-0xFF`	Message Type Identification number. Set to `0XDE` for *ARID*.
Payload	`0-255` B	*ARID* message.
CRC	2 B	Checksum for integrity check.
	***ARID* Payload**	
RES	3 B	Reserved Bytes (e.g. Network, System and Component ID).
PID	67-91 B	UAV Pseudonym $c_{n,t}$ (size based on selected curve).
EKEY	42-66 B	Encrypted one-time key $\rho_{n,t}$ (size based on selected curve).
U_{LAT},U_{LON},U_{ALT}	12 B	UAV Latitude, Longitude, and Altitude (4 bytes each).
U_{VX},U_{VY},U_{VZ}	6 B	UAV Speed x, y and z axis (2 bytes each).
C_{LAT},C_{LON},C_{ALT}	12 B	Latitude, Longitude, and Altitude of the Ground Station (4 bytes each).
TS	4 B	Message Timestamp.
ES	`0x00-0xFF`	UAV Emergency Status.

ANNEX B: DETAILED TIME AND ENERGY MEASUREMENTS OF *ARID*

Table 4: Avg. time and energy (with 95% confidence intervals) required to execute *ARID*, with different elliptic curves size.

Elliptic Curve	Radio Time (*ms*)	Comp. Time (*ms*)	Radio Energy (*mJ*)	Comp. Energy (*mJ*)
secp160r1	1.576	3.942 ± 0.0189	1.544	1.191 ± 0.00574
secp192k1	1.704	5.576 ± 0.0286	1.670	1.685 ± 0.00867
secp224k1	1.832	7.781 ± 0.0389	1.795	2.351 ± 0.01176
secp256k1	1.960	9.272 ± 0.0532	1.920	2.802 ± 0.01609