Authentication Methods for UAV Communication

Mariana Rodrigues*, Jean Amaro†, Fernando Santos Osório† and Kalinka. R. L. J. C. Branco*
*Laboratory of Critical Embedded Systems
Universidade de São Paulo (USP), São Carlos, São Paulo, Brazil
Email: rodrigues.mariana@gmail.com, kalinka@icmc.usp.br
†Laboratory of Mobile Robotics
Universidade de São Paulo (USP), São Carlos, São Paulo, Brazil
Email: jean.amaro@usp.br, fosorio@icmc.usp.br

***Abstract*—Unmanned Aerial Systems (UASs) are being employed in many different applications, and with the increased usage security concerns become more accentuated. Communication links are core components of a UAS, and providing a secure communication channel is a necessity in which Authentication strategies play a major role. However, resource-constrained devices cannot adopt off-the-shelf security strategies that were not developed taking into account those devices' constraints. Because of that, new security strategies, including authentication, were purposed for these applications. In this paper, we analyze and compare two authentication protocols developed for WSNs and modified for UAV usage. Tests were performed analyzing time execution and CPU usage on security-specific operation such as hash tables and elliptic curve operations.**

***Index Terms*—Unmanned aerial vehicles, UAV, Security, Authentication**

I. Introduction

Unmanned Aerial Systems (UASs) are being employed in both civilian and military fields in applications such as infrastructure and environmental monitoring, disaster control and response, surveillance, among others. UASs are usually composed by one or more Unmanned Aerial Vehicles (UAVs), a Ground Control Station (GCS) and the communication links [1]. The use of multiple smaller UAVs in a swarm rather than a single UAV has been proving to be useful in civilian applications [2] and presents some advantages. The purchase and maintenance costs are lower; the missions are realized faster and have less chance of failure, since there are other UAVs to take on a task if a problem happens; and the communication with the GCS is broadened beyond the Line-of-Sight (LoS) if a ad hoc network is created [3].

In a multiple UAV system, communication becomes even more preponderant. UAS communication is particularly challenging due to (a) the high node mobility, (b) fluid topology, (c) long distance between the nodes, which can make the communication links intermittent, and (d) the power constraints, since UAVs are battery-equipped and have to save as much power as possible [4].

Security is always an issue in communicating systems and a great concern in UAV networks. The wireless medium is inherently insecure [5] and can be easily eavesdropped or suffer other attacks, like Denial-of-Service (DoS), Man-in-the-Middle (MitM), Replay attacks, or Sybil/Impersonation attacks.

DoS attacks target the system availability by hindering or blocking the access to a host or service by trying to exhaust some host resource such as processing power, memory, communication bandwidth or disk space [6]. In UASs, a jamming between the UAV and the GCS can make the system lose UAV control, which can crash or be taken over by an attacker [7]. In MitM attacks, a malicious node puts itself between two hosts in the system. In a passive MitM attack, the malicious node just relays every intercepted message to the intended recipient without modification, with the intention of eavesdropping. In an active MitM attack, the intercepted message can be modified to take control of the communication or inject false data in the system [8]. MitM attacks can violate confidentiality, integrity, and privacy of restricted data [9] and is considered one of the most dangerous attacks on UASs, since an adversary can intercept and alter communication and control packets or take control of a UAV by deauthenticating the original node [10]. Replay attacks are MitM attacks in which data packets are intercepted and replayed to the destination server at a later time so an unauthorized user can gain access to the system [11]. Commonly used against authentication protocols, it can be very effective if the timestamp of a request is not considered in the authentication process [8]. Sybil is an impersonation attack in which a malicious node claims several legitimate identities and impersonate them in the system. This attack enables false data injection and routing disturbance [8], [9].

Establishing secure communication channels is necessary for safe UAV operation. Not only the external communication links (between different UAVs or between a UAV and the infrastructure) but also the intra-vehicle communication also needs to be secure. The UAV must assure that only authorized parties access its resources, and that all internal modules are authenticated so built-in device security can be achieved.

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

We thank CNPq for the financial support to Jean Amaro of graduate program the University of São Paulo at São Carlos.

Research was also sponsored by the Army Research Office and was accomplished under Grant Number W911NF-18-1-0012. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office or the U.S. Government. The U.S. Government is authorised to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

978-1-7281-2999-0/19/$31.00 ©2019 IEEE

Authorized licensed use limited to: University of Exeter. Downloaded on May 06,2020 at 10:21:17 UTC from IEEE Xplore. Restrictions apply.

Since UASs operate without human intervention, the majority of authentication operations will be device-to-device. It is important that all modules are authenticated before the UAV gain access to the GCS. However, UAV limited computational and power resources hinder the application of off-the-shelf security solutions [12].

Authentication and encryption are fundamental security requirements to create a secure communication channel. Although slower, public key cryptography schemes are often used during the initial phases of authentication and key agreement, in which a session key can be established between the devices and used for a lighter, faster symmetric cryptography. Mutual authentication prevents MitM attacks and together with encryption helps preventing DoS attacks in resource-constrained environments [13].

In this paper, two authentication schemes for Wireless Sensor Networks (WSNs) are investigated to be applied in UAVs. The goal is to provide mutual authentication and key agreement so secure communication channels can be created between vehicles. The advantage of using WSN or IoT protocols is that UAVs can be easily integrated to these systems with minimum effort.

The paper is organized as follows: Section II presents some related work regarding authentication in resource-constrained devices; Section III presents the authentication strategies evaluated in Section IV; finally, Section V concludes the paper.

II. Related Work

There are many authentication strategies for resource-constrained environments, such as those found in surveys such as [11] and [14]. In this work, the objective of authentication is to prevent impersonation (by providing mutual authentication) and replay attacks (by the use of timestamps or nonces in the authentication scheme). Some related work is presented in this section.

In [15], it is proposed the use of a secured channel for improving communication security between UAV and the GCS by continuous authentication. In a setup phase, the GCS and the UAV exchange an array of random numbers. During operation, an array index is used as challenge to provide authentication.

Authors from [16] propose a preliminary architecture for UAV ID-based authentication. In the scheme, the UAV is equipped with a RFID tag that provides unique identification. In the authorization process, a temporary UAV identification is provided to preserve privacy, and both IDs are used to generate the cryptography keys. However, there is no mutual authentication, which leave the system vulnerable to man-in-the-middle and impersonation attacks. In fact, establishing a secure communication between drones and the GCS is cited as a future development.

A similar strategy of composing the private key is proposed in [17]. An authentication strategy for Vehicle Ad hoc Networks (VANETs), it models a scenario in which vehicles equipped with On-Board Units (OBUs) authenticate itself to the Roadside Units (RSUs) that compose the network infrastructure. A fully trusted Private Key Generator (PKG), which can be located in the Cloud, is responsible for producing secret and public keys. The scheme divides the vehicle's private key into two parts which are updated periodically, and the vehicle's authentication is done by the RSU. The scheme do not use bilinear pairing; according to the authors, a pure-ECC approach has better efficiency and is feasible for VANET environment.

The work in [18] presents an authentication and key agreement for smart grid scenarios. The scenario is composed by home smart meters (SMs) and a utility server in a wireless mesh network. The utility server, called security associate (SA), is responsible for the key management, assuming the duties of a Certified Authority (CA). All nodes have a unique serial number which is applied to a cryptography function to provide the node's public key. The SA has the information of all nodes stored in a local database, and is responsible for providing the SMs' private key. When a new SM wishes to enter the network, it chooses any of other SM as Authentication Agent (AG), which will function as a mediator between SM and SA mutual authentication. If the authentication process is successful, the SM can query its private key from SA. In order to maintain the security level of the system, the SA periodically generates and broadcasts a new cryptography function that is used to refresh the system's cryptography keys. Since those keys are ID-based, each node can calculate its own private key and its peers' public keys once the new function is received.

The strategy proposed on [19] allows a random user to connect with a Wireless Sensor Network (WSN) and configure a secure communication with it. The scenario is a WSN with small, resource-constrained sensor nodes (S_j), some gateway nodes (GNs) with more resources, and users who want to connect to a sensor and receive data from it. Each sensor node is loaded with its own ID and a secure password key X_{GN-S_j}, while the gateway node is loaded with a list of all passwords shared with sensor nodes that needs to be updated when a new node enters the network. Both users and sensors are registered in the network, and the mutual authentication occurs between user and sensor, sensor and GN, and user and GN.

The strategy presented on [20] follows the same scenario as Strategy 2: a WSN composed by both sensor and gateway nodes, and users who wants to connect with it. The *registration* phase is applied to both sensors and users. Sensors are registered based on their ID and users with a ID/password pair, used in the *login* phase to connect to the network (as with Strategy 2, users can change their password at a later time). In the *authentication* phase, the user request the authentication to the GN, which verifies the user authenticity and forward it to a sensor node. The sensor node will authenticate the GN and answer the authentication accordingly; upon receiving a successful indication and verifying the sensor's authenticity, the GN forward the successful authentication request to the user, who in turn verifies the authenticity of the GN.

In [21], the authors propose a lightweight identity-based key establishment protocol called $NIKE^+$ for an Advanced

Authorized licensed use limited to: University of Exeter. Downloaded on May 06,2020 at 10:21:17 UTC from IEEE Xplore. Restrictions apply.

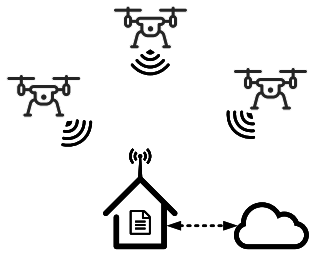

Fig. 1: Considered scenario for authentication strategies.

Metering Infrastructure (AMI) based on Elliptic Curves. The system is composed by Smart Meters (SMs), a AMI Head-End (AHE, a central node in the operation center) and a Trusted Authority (TA). The method has three phases: Setup, installation and Key Agreement. In the *Setup* phase, the TA chooses a master key x and other systems parameters related to the Elliptic Curve the system will use. In the *installation* phase, the TA uses the parameters defined on Setup phase and the master key x to define and share the element's private keys through a secure channel. For key agreement, the SM and AHE uses nonces and the keys received in installation phase to authenticate each other and agree on a session key.

III. Authentication Strategies Evaluation

In this Section, the evaluation of two authentication strategies are analyzed for mutual UAV authentication. On the first strategy, based on [19], the first UAV connects itself directly to the other UAV, while on the second strategy, based on [20] the first UAV requests a connection through the GCS.

These two strategies were selected given the fact that, according to [11], present support against common UAV attacks such as MitM and Replay attacks, and also provide partial support to Sybil and DoS attacks, as shown by TABLE I.

Paper	Attacks				
	DoS	MitM	Replay	Impersonation	Sybil
[19]	F	F	F	P	P
[20]	P	F	F	F	P

TABLE I: Support against selected attacks for chosen authentication strategies according to [11]. F means full support and P partial support against the attack.

The authentication process is divided into two phases: *registration* and *key agreement*. Registration phase deals with the connection between the UAV and the GCS, while the key agreement phase deals with the symmetric key exchange between two UAVs in the system. The considered scenario is illustrated by Fig. 1. The GCS has the identification information of all UAVs in the UAS and acts like the UAS central unit and the point of contact with outside networks. Each UAV also has the information of other UAVs it needs to communicate. Next, the two strategies for UAV authentication are described and illustrated.

Strategy 1: UAV to UAV Authentication Request

In this scenario, a UAV requests a connection directly to another UAV. This strategy is based on the one presented in [19], but the key agreement protocol was modified to be applied to M2M communication rather than a user with smart card accessing the WSN. The registration phase was not modified from the original proposal. Each UAV has an unique identification and a shared password with the CGS, which in turn stores all (ID, Password) pairs of the system. Password loading process is done offline by a network administrator. By the end of registration phase, described by Fig. 2 with notations from TABLE II, each UAV$_j$ will have stored a credential x_j and a shared key $H(K_{CGS}\|1)$. These values are then used for the key agreement protocol shown in Fig. 3.

Notation	Desciption
$UAV_{i,j}$	UAVs in the system
K_{CGS}	Secure password known only to the GCS
$K_{CGS-UAV_j}$	Secured password shared with the UAV$_j$
r_j	Secure random nonces
T_x	Timestamps
$\oplus$, $\|$ and $H()$	XOR, concatenation and hash function
$X =? Y$	Tests if $X == Y$

TABLE II: Notation used on Strategy 1.

In the key agreement protocol, a UAV$_i$ requests a connection to a UAV$_j$ by sending its credentials obtained in the registration phase. UAV$_j$ will then request the GCS for identity confirmation. GCS will authenticate the identities of both UAV$_i$ and UAV$_j$ and provide the necessary data for creating a session symmetric key. In turn, both UAV$_i$ and UAV$_j$ authenticate the GCS and calculate the symmetric key to communicate securely.

Strategy 2: UAV to CGS Authentication Request

In this scenario, a UAV requests a connection to another UAV through the GCS. This strategy is based on the one presented in [20] also with the key agreement protocol being modified. In the original protocol, a user with a smart card was connected to any sensor. In this scenario, the requesting UAV knows the identity of the UAV it wants to communicate with. This strategy is based on Elliptic Curve Cryptography, which has been demonstrated to have better performance than RSA [22].

Notation	Desciption
$UAV_{i,j}$	UAVs in the system
$K_{CGS-UAV}$	Master secret key only known to GCS
$y = xP$	CGS' Public key
DID_x	Disguised ID of UAS component
TC_x	Credential for UAV$_x$
T_x	Timestamps
$\oplus$, $\|$ and $H()$	XOR, concatenation and hash function
$X =? Y$	Tests if $X == Y$

TABLE III: Notation used on Strategy 2.

The GCS has a private key x and its corresponding public key xP. The GCS also creates a master secret key $K_{CGS-UAV}$.

Authorized licensed use limited to: University of Exeter. Downloaded on May 06,2020 at 10:21:17 UTC from IEEE Xplore. Restrictions apply.

UAV$_j$	**CGS**
Has UID_j	Knows master key K_{GCS}
Has $K_{GCS-UAV_j}$	Has $K_{GCS-UAV_j}$ for every UID_j
Selects random r_j	
$MP_j = H(K_{GCS-UAV_j} \| r_j \| UID_j \| T_1)$	
$MN_j = r_j \oplus K_{GCS-UAV_j}$	
$\xrightarrow{\{UID_j, MP_j, MN_j, T_1\}}$	
	Check $\lvert T_1 - T_c \rvert < \Delta T$
	$r'_j = MN_j \oplus K_{GCS-UAV_j}$
	$MP_j =? H(K_{GCS-UAV_j} \| r'_j \| UID_j \| T_1)$
	$x_j = H(UID_j \| K_{GCS})$
	$e_j = x_j \oplus K_{GCS-UAV_j}$
	$g_j = H(K_{GCS} \| 1)$
	$d_j = g_j \oplus H(K_{GCS-UAV_j} \| T_2)$
	$f_j = H(x_j \| d_j \| K_{GCS-UAV_j} \| T_2)$
$\xleftarrow{\{e_j, f_j, d_j, T_2\}}$	
Check $\lvert T_2 - T_c \rvert < \Delta T$	
$x_j = e_j \oplus K_{GCS-UAV_j}$	
$f_j =? H(x_j \| d_j \| K_{GCS-UAV_j} \| T_2)$	
$g_j = d_j \oplus H(K_{GCS-UAV_j} \| T_2)$	
Stores x_j and $g_j = H(K_{GCS} \| 1)$	
Deletes $K_{GCS-UAV_j}$	
$\xrightarrow{\text{Confirmation}}$	
	Deletes UID_j and $K_{GCS-UAV_j}$

Fig. 2: UAV registration from [19].

In order to perform the registration, UAV$_j$ send its identification UID$_j$ to the GCS through a secure channel. The GCS computes $TC_j = H(K_{CGS-UAV} || UID_j)$ as the UAV's credential, which is sent back through the secure channel and stored by the vehicle.

Since UAV$_i$ intends to communicate with UAV$_j$, it needs the ID information of UAV$_j$ to be sent to the GCS, and this point is the main modification on the key agreement protocol exposed in Fig. 4 using the notations from TABLE III.

Time constrains are very important in a UAS, given its critical nature. Its inherent resource constraint also impacts greatly on security and must always be considered. Any proposed security solution has to investigate its performance and impact on UAV operation [13].

IV. Performance evaluation

In order to perform an evaluation, the previously detailed authentication strategies were implemented and profiled regarding CPU usage and execution time. The implementation was done in C++ using Microsoft Visual C++ 2017 v14.1 compiler.

UAV$_i$	**UAV**$_j$	**GCS**
Has UID_i	Has UID_j	
Stores x_i	Stores x_j	
Store $H(K_{GCS} \| 1)$		Knows K_{GCS}
$M_1 = UID_i \oplus H(H(K_{GCS} \| 1) \| T_1)$		
Selects random nonce K_i		
$M_2 = H(x_i \| T_1) \oplus K_i$		
$M_3 = H(M_1 \| M_2 \| K_i \| T_1)$		
$\xrightarrow{\{M_1, M_2, M_3, T_1\}}$		
	Checks $\lvert T_1 - T_c \rvert < \Delta T$	
	$EUID_j = UID_j \oplus H(H(K_{GCS} \| 1) \| T_2)$	
	Selects random nonce K_j	
	$M_4 = H(x_j \| T_1 \| T_2) \oplus K_j$	
	$M_5 = H(UID_j \| M_4 \| T_1 \| T_2 \| K_j)$	
	$\xrightarrow{\{M_1, M_2, M_3, T_1, T_2, EUID_j, M_4, M_5\}}$	
		Checks $\lvert T_2 - T_c \rvert < \Delta T$
		$UID'_j = EUID_j \oplus H(H(K_{GCS} \| 1) \| T_2)$
		$x'_j = H(UID'_j \| K_{GCS})$
		$K'_j = M_4 \oplus H(x'_j \| T_1 \| T_2)$
		$UID'_i = M_1 \oplus H(H(K_{GCS} \| 1) \| T_1)$
		$x'_i = H(UID'_i \| K_{GCS})$
		$K'_i = M_2 \oplus H(x'_i \| T_1)$
		$M_3 =? H(M_1 \| M_2 \| K'_i \| T_1)$
		$M_5 =? H(UID'_j \| M_4 \| T_1 \| T_2 \| K'_j)$
		$M_6 = K'_j \oplus H(x'_i \| T_3)$
		$M_7 = K'_i \oplus H(x'_j \| T_3)$
		$M_8 = H(M_6 \| x'_i \| T_3)$
		$M_9 = H(M_7 \| x'_j \| T_3)$
	$\xleftarrow{\{M_6, M_7, M_8, M_9, T_3\}}$	
	Checks $\lvert T_3 - T_c \rvert < \Delta T$	
	$M_9 =? H(M_7 \| x_j \| T_3)$	
	$K''_i = M_7 \oplus H(x_j \| T_3)$	
	$SK_j = SK_i = H(K''_i \oplus K_j)$	
	$M_{10} = H(SK_j \| M_6 \| M_8 \| T_3 \| T_4)$	
$\xleftarrow{\{M_6, M_8, M_{10}, T_3, T_4\}}$		
Checks $\lvert T_4 - T_c \rvert < \Delta T$		
$M_8 =? H(M_6 \| x_i \| T_3)$		
$K''_j = M_6 \oplus H(x_i \| T_3)$		
$SK_i = SK_j = H(K''_j \oplus K_i)$		
$M_{10} =? H(SK_i \| M_6 \| M_8 \| T_3 \| T_4)$		

Fig. 3: UAV key agreement for Strategy 1 adapted from [19].

Authorized licensed use limited to: University of Exeter. Downloaded on May 06,2020 at 10:21:17 UTC from IEEE Xplore. Restrictions apply.

$\mathbf{UAV}_i$	**GCS**	$\mathbf{UAV}_j$
Has UID_i		Has UID_j
Stores TC_i		Stores TC_j
Knows UID_j	Knows $K_{GCS-UAV}$	

Generates $a \in \mathbb{Z}_{p-1}^*$
$A_i = aP$
$D_i = ay = axP$
$DID_i = UID_i \oplus H(A_i \| D_i)$
$IID_i = UID_j \oplus H(A_i \| UID_i)$
$C_i = H(UID_i \| IID_i \| T_1 \| D_i \| A_i \| TC_i)$

$\xrightarrow{\{DID_i, A_i, IID_i, T_1, C_i\}}$

Checks $|T_1 - T_c| < \Delta T$
$D_i' = xA_i = xaP$
$UID_i' = DID_i \oplus H(A_i \| D_i')$
$UID_j' = IID_i \oplus H(A_i \| UID_i')$
$TC_i = H(K_{GCS-UAV} \| UID_i')$
$C_i =? H(UID_i' \| IID_i \| T_1 \| D_i' \| A_i \| TC_i)$
$TC_j = H(K_{GCS-UAV} \| UID_j')$
$DID_{GCS} = UID_i' \oplus H(DID_i \| TC_j \| T_2)$
$C_{GCS} = H(UID_i' \| TC_j \| A_i \| T_2)$

$\xrightarrow{\{T_2, DID_i, DID_{GCS}, C_{GCS}, A_i\}}$

Checks $|T_2 - T_c| < \Delta T$
$UID_i'' = DID_{GCS} \oplus H(DID_i \| TC_j \| T_2)$
$C_{GCS} =? H(UID_i'' \| TC_j \| A_i \| T_2)$
Generates $b \in \mathbb{Z}_{p-1}^*$
$B_j = bP$
$D_j = by = bxP$
$SK_{ij} = H(bA_i) = H(abP)$
$C_j = H(TC_j \| UID_i'' \| UID_j \| B_j \| T_3)$

$\xleftarrow{\{B_j, T_3, C_j\}}$

Checks $|T_3 - T_c| < \Delta T$
$D_j' = xB_j = xbP$
$C_j =? H(TC_j \| UID_i' \| UID_j' \| B_j \| T_3)$
$E_{GCS} = H(UID_i' \| TC_i \| D_i' \| UID_j' \| B_j \| T_4)$

$\xleftarrow{\{B_j, T_4, E_{GCS}\}}$

Checks $|T_4 - T_c| < \Delta T$
$E_{GCS} =? H(UID_i \| TC_i \| D_i \| UID_j \| B_j \| T_4)$
$SK_{ij} = H(aB_j) = H(abP)$

Fig. 4: UAV key agreement for Strategy 2 adapted from [20].

The tests were run on a Intel Core i5-7300HQ desktop with DDR4-2400 / PC4-19200 DDR4 SDRAM SO-DIMM memory running Windows 10 x64 - Education Edition. Crypto++ 8.0.0 library[1] was used for elliptic curve and hash (SHA-256 [23]) operations. Transmission and public key encryption operations are discarded from the analysis.

TABLE IV brings the mean execution time for each execution block for both strategies. An execution block is defined as the totality of operations executed by each agent before or after exchanging information. Based on this, both strategies have five execution blocks. Every strategy was run 30 times to guarantee statistical viability.

Block	Execution time (μs)	
	STRATEGY 1	STRATEGY 2
1	2,36	1461,24
2	2,63	728,26
3	7,64	2186,02
4	2,49	3,60
5	16,59	729,26
Total	31,71	5108,38

TABLE IV: Execution time by blocks in both authentication strategies.

The table shows that Strategy 2 has a greater execution time than Strategy 1. This is caused by elliptic curve operations adopted. Tables TABLE V and VI bring the estimated execution time on cryptographic operations based on CPU usage profiling done in Visual Studio for both strategies. In order to get measurable CPU usage, the profiling was run 10 times, with Strategy 1 executing one hundred thousand and Strategy 2 one thousand authentication operations.

STRATEGY 1			
Agent	Time (μs)	SHA-256 operations	
		Time (μs)	%
UAV_i	18,95	2,35	12,40
UAV_j	5,12	2,56	50,00
GCS	7,64	5,29	69,24

TABLE V: Estimated time spent on operations based on CPU usage profiling for Strategy 1.

TABLE V details the estimated time spent on SHA-256 operations in Strategy 1 from Crypto++ library based on CPU usage profiling. The high execution time on UAV_i is explained by the system initialization, which spends more time than other agents. TABLE VI details the estimated time spent on both SHA-256 and elliptic curve operations in Strategy 2. The scalar multiplication operation on a elliptic curve was separated from other operations in order to emphasize the high percentage of processing spent on this operation.

Overall, Strategy 2 showed higher mean execution time, mostly because of costly elliptic curve operations. Since its registration phase also includes sending information over a secure channel and therefore using EC operations more times to encrypt and decrypt messages, the total time for executing the operation will be higher than Strategy 1. On the other

[1] https://www.cryptopp.com

Authorized licensed use limited to: University of Exeter. Downloaded on May 06,2020 at 10:21:17 UTC from IEEE Xplore. Restrictions apply.

STRATEGY 2							
Agent	Time (μs)	SHA-256 op		EC op			
				S. Multiplication		Other	
		Time (μs)	%	Time (μs)	%	Time (μs)	%
UAV_i	2190,50	4,19	0,19	2168,21	98,98	11,04	0,50
GCS	731,87	5,13	0,70	665,39	90,92	1,80	0,25
UAV_J	2186,02	4,31	0,20	2161,25	98,87	12,30	0,56

TABLE VI: Estimated time spent on operations based on CPU usage profiling for Strategy 1.

hand, according to TABLE I, Strategy 1 is not as effective for Impersonation attacks as Strategy 2.

V. CONCLUSION

Authentication is a major security requirement for establishing a secure communication channel. In resource-constrained applications such as those with UAVs, it is not feasible to apply off-the-shelf solutions, which demands the development of new authentication protocols that are at the same time efficient and effective. In this paper, two authentication protocols developed for WSNs were modified and evaluated for UAV mutual authentication and communication. The first strategy is based mainly on hash operations, while the second apply elliptic curve operations. Time execution and CPU usage results show that the hash-based strategy is more efficient, providing full support for most common UAV attacks analyzed.

Both strategies analyzed refers to the same scenario: UAVs communicating in a system with a GCS as a central authenticator. Future work includes the analysis of other strategies for decentralized scenarios (without a central unit) and in-vehicle authentication in order to provide a full UAV-suited authentication suite.

REFERENCES

[1] K. P. Valavanis and G. J. Vachtsevanos, *Handbook of Unmanned Aerial Vehicles*. Dordrecht: Springer Netherlands, 2015, ISBN: 9789048197064.

[2] L. Gupta, R. Jain, and G. Vaszkun, "Survey of Important Issues in UAV Communication Networks," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1123–1152, 2016. [Online]. Available: http://doi.org/10.1109/COMST.2015.2495297

[3] . Bekmezci, O. K. Sahingoz, and . Temel, "Flying Ad-Hoc Networks (FANETs): A survey," *Ad Hoc Networks*, vol. 11, no. 3, pp. 1254–1270, may 2013. [Online]. Available: http://dx.doi.org/10.1016/j.adhoc.2012.12.004

[4] N. Hossein Motlagh, T. Taleb, and O. Arouk, "Low-Altitude Unmanned Aerial Vehicles-Based Internet of Things Services: Comprehensive Survey and Future Perspectives," *IEEE Internet of Things Journal*, vol. 3, no. 6, pp. 899–922, dec 2016. [Online]. Available: http://doi.org/10.1109/JIOT.2016.2612119

[5] D. He, S. Chan, and M. Guizani, "Communication Security of Unmanned Aerial Vehicles," *IEEE Wireless Communications*, vol. 24, no. 4, pp. 134–139, aug 2017. [Online]. Available: http://dx.doi.org/10.1109/MWC.2016.1600073WC

[6] W. Stallings and L. Brown, *Computer Security: Principles and Practice, Global Edition*. Pearson, 2017.

[7] H. Sedjelmaci and S. M. Senouci, "Cyber security methods for aerial vehicle networks: taxonomy, challenges and solution," *The Journal of Supercomputing*, vol. 74, no. 10, pp. 4928–4944, oct 2018. [Online]. Available: https://doi.org/10.1007/s11227-018-2287-8

[8] J. Vacca, *Computer and Information Security Handbook*. Elsevier Science, 2017.

[9] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, "A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1125–1142, 2017. [Online]. Available: http://doi.org/10.1109/JIOT.2017.2683200

[10] R. M. Fouda, "Security Vulnerabilities of Cyberphysical Unmanned Aircraft Systems," *IEEE Aerospace and Electronic Systems Magazine*, vol. 33, no. 9, pp. 4–17, sep 2018. [Online]. Available: http://doi.org/10.1109/MAES.2018.170021

[11] M. A. Ferrag, L. A. Maglaras, H. Janicke, J. Jiang, and L. Shu, "Authentication Protocols for Internet of Things: A Comprehensive Survey," *Security and Communication Networks*, vol. 2017, pp. 1–41, 2017. [Online]. Available: http://doi.org/10.1155/2017/6562953

[12] L. Chen, S. Qian, M. Lim, and S. Wang, "An enhanced direct anonymous attestation scheme with mutual authentication for network-connected UAV communication systems," *China Communications*, vol. 15, no. 5, pp. 61–76, may 2018. [Online]. Available: http://doi.org/10.1109/CC.2018.8387987

[13] R. Altawy and A. M. Youssef, "Security, Privacy, and Safety Aspects of Civilian Drones," *ACM Transactions on Cyber-Physical Systems*, vol. 1, no. 2, pp. 1–25, nov 2016. [Online]. Available: http://doi.org/10.1145/3001836

[14] M. El-hajj, A. Fadlallah, M. Chamoun, and A. Serhrouchni, "A Survey of Internet of Things (IoT) Authentication Schemes," *Sensors*, vol. 19, no. 5, p. 1141, mar 2019. [Online]. Available: http://doi.org/10.3390/s19051141

[15] K. Yoon, D. Park, Y. Yim, K. Kim, S. K. Yang, and M. Robinson, "Security Authentication System Using Encrypted Channel on UAV Network," in *IEEE International Conference on Robotic Computing (IRC)*. Taichung, Taiwan: IEEE, apr 2017, pp. 393–398. [Online]. Available: http://doi.org/10.1109/IRC.2017.56

[16] B. Sana, T. Bayrem, and K. Ouajdi, "Privacy preservation and drone authentication using id-based signcryption," *Frontiers in Artificial Intelligence and Applications*, vol. 303, no. New Trends in Intelligent Software Methodologies, Tools and Techniques, pp. 226–239, 2018. [Online]. Available: http://doi.org/10.3233/978-1-61499-900-3-226

[17] Y. Zhou, S. Liu, M. Xiao, S. Deng, and X. Wang, "An Efficient V2I Authentication Scheme for VANETs," *Mobile Information Systems*, vol. 2018, pp. 1–11, 2018. [Online]. Available: http://doi.org/10.1155/2018/4070283

[18] H. Nicanfar, P. Jokar, and V. C. M. Leung, "Smart grid authentication and key management for unicast and multicast communications," in *IEEE PES Innovative Smart Grid Technologies*. Perth, Australia: IEEE, nov 2011, pp. 1–8. [Online]. Available: http://doi.org/10.1109/ISGT-Asia.2011.6167151

[19] M. S. Farash, M. Turkanović, S. Kumari, and M. Hölbl, "An efficient user authentication and key agreement scheme for heterogeneous wireless sensor network tailored for the Internet of Things environment," *Ad Hoc Networks*, vol. 36, pp. 152–176, jan 2016. [Online]. Available: http://doi.org/10.1016/j.adhoc.2015.05.014

[20] Q. Jiang, N. Kumar, J. Ma, J. Shen, D. He, and N. Chilamkurti, "A privacy-aware two-factor authentication protocol based on elliptic curve cryptography for wireless sensor networks," *International Journal of Network Management*, vol. 27, no. 3, p. e1937, may 2017. [Online]. Available: http://doi.org/10.1002/nem.1937

[21] A. Mohammadali, M. S. Haghighi, M. H. Tadayon, and A. Mohammadi-Nodooshan, "A Novel Identity-Based Key Establishment Method for Advanced Metering Infrastructure in Smart Grid," *IEEE Transactions on Smart Grid*, vol. 9, no. 4, pp. 2834–2842, 2018.

[22] D. Mahto and D. K. Yadav, "Performance Analysis of RSA and Elliptic Curve Cryptography," *International Journal of Network Security*, vol. 20, no. 4, pp. 625–635, 2018. [Online]. Available: http://doi.org/10.6633/IJNS.201807_20(4).04

[23] National Institute of Standards and Technology, "FIPS.180-4 — Secure Hash Standard," Gaithersburg, MD, Tech. Rep., jul 2015. [Online]. Available: http://doi.org/10.6028/NIST.FIPS.180-4

Authorized licensed use limited to: University of Exeter. Downloaded on May 06,2020 at 10:21:17 UTC from IEEE Xplore. Restrictions apply.