A Certificateless Group Authenticated Key Agreement Protocol for Secure Communication in Untrusted UAV Networks

Benjamin Semal, Konstantinos Markantonakis, and Raja Naeem Akram
Information Security Group, Royal Holloway, University of London, Egham, United Kingdom
Email: benjamin.semal.2018@live.rhul.ac.uk, {k.markantonakis, r.n.akram}@rhul.ac.uk

***Abstract*—Smart cities are increasingly relying on information and communication technologies to cope with today's challenges, including increasing population, heterogeneous economic growth, and environmental legislation hardening. The use of Unmanned Aerial Vehicles (UAVs) generates a growing interest in coping with such challenges, along with new business opportunities. As a result, it is expected to see the number of drone-based applications to rise at a very fast pace, entailing new cybersecurity risks to emerge. In this paper, we propose to tackle the problem of secure communication among untrusting parties with a certificateless-group authenticated key agreement (CL-GAKA) scheme. The intent is to enable confidentiality, message integrity, and authenticity in UAV-to-UAV communication. Provisioning untrusted UAV networks with trusted communication will provide ground for further research and applications, such as collaborative cybersecurity deterrence, network extension using trusted relay nodes, collaborative mission exercise in time-critical applications, or anonymous communication for user privacy. Moreover, in order to fill the gap of technology awareness, we provide an implementation and demonstrate that the generation of a session key between two users can be performed in approximately 600ms on a Raspberry Pi 3 Model B+ system-on-chip (1.4GHz Cortex-A53). Finally, the protocol is formally evaluated with the Scyther tool.**

I. Introduction

As cities grow in population, they have to rely more and more on information and communication technologies in order to provide efficient services at reduced cost. Among these key technologies, Unmanned Aerial Vehicles (UAVs) are increasingly playing their part with major companies stepping in. For instance, Amazon and Google are currently competing in the development of a drone delivery service, i.e. Amazon Prime Air and Project Wing respectively. AT&T has started to automate its cell tower inspections by using drones. In 2017, the city of Dubai has launched a flying taxi service, bringing drones to the public transportation arena. Within smart cities, drones will operate in a wide range of applications. This trend is expected to grow at a (very) fast pace, resulting in more and more organizations operating drones in urban areas.

Together with the many benefits of the use of UAVs in smart cities, comes the need for regulation and security. One of the main security concerns is secure communication, as drones are vulnerable to cyber threats [1], [2]. With the proliferation of drone-based services, it is becoming essential to devise a protocol for secure channel establishment. These channels are required for collision avoidance, data relay, identification, mission planning, and so on. The use of UAV fleets is also generating a great interest for operations such as network extension, surveillance, and monitoring and mapping [3].

Secure channel protocols usually assume a prior relationship between communicating entities. For example, they would be preloaded with a common symmetric cryptographic key, a password, or a pool of digital certificates. A multi-stakeholder environment, such as a smart city, offers the potential for a multiplicity of drone-based applications, causing as many organizations to interact with each other. In this paper, we address the problem of secure communication between untrusting parties. Our protocol allows UAVs belonging to different owners to interact and communicate securely with each other. There it should encompass the following three key requirements. First, as UAVs operate in open environments, mutual authentication is indispensable. Secondly, different organizations may be involved, hence the necessity for joint key control. Finally, the key escrow problem must be eliminated, in order to prevent the third party from having full knowledge of users' secret keys.

In this paper, we propose a certificateless-group authenticated key agreement (CL-GAKA) protocol, which provides the above mentioned requirements. The intent is also to include additional security attributes, such as revocation, non-repudiation, denial of service prevention, and conditional privacy. The contributions of this paper are summarized as follows:

1) Propose a certificateless group authenticated key agreement protocol dedicated to untrusted UAV networks, based on the theoretical work of Teng *et al.* [4];
2) Implement and analyze the performance of the key agreement protocol with a set of Raspberry Pi 3 Model B+ system-on-chip (1.4GHz Cortex-A53).

Section 2 provides a background on UAV networks and key agreement protocols. Section 3 describes the proposed CL-GAKA scheme, and lists the desired security and operational requirements. Finally, section 4 relates the implementation details, the formal analysis, and the performance evaluation.

II. Related Work and Rationale

This section outlines the requirements for secure UAV-to-UAV communications. Several approach to authenticated key

Table I: Comparative description of WSNs, VANETs, and UANETs

	WSN	VANET	UANET
Description	Wireless network consisting of spatially distributed sensors	Self-organizing wireless network consisting of spatially distributed ground vehicles	Self-organizing wireless network consisting of spatially distributed airborne drones
Node Category	Low-end	Medium-end	Medium-end (civil)
Topology	Star, mesh	Mesh	Star, mesh
Node Mobility	Static	High-speed, typically 0-30 m/s, in a 2-dimensions space	Very high-speed, typically 0-100 m/s, in a 3-dimensions space
Ownership	Nodes belong to the same system	Nodes are manufactured by different organizations and used by different owners	Nodes are manufactured by different organizations and potentially used by different owners
Exchanged information	Physical and environmental conditions	Physical conditions	Physical and environmental conditions, operational data, network traffic
Examples of applications	Environmental monitoring, structural monitoring, industrial machine monitoring, process monitoring	Platooning, traffic information system, road transportation emergency services, electronic brake light, on-the-road services	Environmental monitoring, network extension, delivery service, structural monitoring, search & rescue, law enforcement

agreement are evaluated. Finally, related work is presented and a rationale for the paper is provided.

A. Communication in UAV Networks

In many aspects, UAV ad-hoc networks (UANETs) are similar to wireless sensor networks (WSNs) and vehicular ad-hoc networks (VANETs). Table I compares singularities among WSNs, VANETs, and UANETs. On the one hand, it is observed that UAVs have PC-like computational power, which allows more complex calculations than for sensors. On the other hand, the autonomy of a UAV is more limited than a ground vehicle, causing any computation (e.g. navigation) to impact the drone's resources. Furthermore, UAVs have unique mobility capabilities, resulting in network connectivity and communication efficiency issues. Communication among UAVs is an imperative for mission exercise, although it does not always occur among entities from the same organization. Any operation taking place in urban areas is likely to require exchange of sensitive data among untrusting entities. Furthermore, this collaboration can benefit time-critical operations, and allow secure peer-to-peer network extension.

B. Secure Communication in UANETs

UAVs communicate over an open network, which is controlled by an adversary. A communication protocol is secured if it guarantees that an adversary cannot infiltrate the protocol. There it should encompass at least three security properties: confidentiality, integrity, and authenticity. Table II proposes various security levels for communication protocols.

a) Confidentiality: privacy ensures that information is not made available to unauthorized individuals or entities. In other words, confidentiality protects the communication against eavesdropping. An eavesdropper aims at either reading the content of a message, or obtaining the origin and destination of the message. Security levels 4 and 6-8 are vulnerable to such malicious behaviour. In order to enforce confidentiality, the communicating entities must encrypt the message. For performance considerations, it is preferable to use cryptographic algorithms with symmetric ciphers. As a result, the entities must share a symmetric cryptographic key to encrypt and decrypt messages. If the communicating entities do not share a secret symmetric key, they must perform a key agreement procedure.

b) Integrity: message integrity provides the assurance that information cannot be tampered with by unauthorized individuals without being detected. Message integrity protects against replay attacks if it is added with session cookies (unique session identifier) and message authentication codes (MACs). In a replay attack, an adversary records the network traffic and repeats the messages to its target. Thus it aims at fooling an entity into a valid communication. A MAC is a tag appended to the message, which contains the message in ciphertext. Upon reception of a message, an entity computes its own MAC and compares it against the one in the message. Security levels 3, 5, and 7-8 are vulnerable to replay attacks. The use of MAC also requires the communicating entities to share a symmetric key, hence the necessity for a key establishment protocol.

c) Authenticity: authentication refers to the process of verifying the identity of a user, here a UAV. User authentication differs from message authentication, which is the act of verifying the integrity of an information. Authenticity provides ground for non-repudiation and trust mechanisms. Non-repudiation forbids a user from denying having performed a communication session. Because non-repudiation requires both integrity and user authenticity, security levels 2-3, and 5-8 are not appropriate. Additionally, authenticity precludes spoofing attacks. A spoofing attack is a type of man-in-the-middle (MiM) attack, where an adversary intercepts the network traffic between the communicating entities, and impersonates the sender to the receiver and vice versa. Thus, the middle-man creates its own (secure) channels with the receiver and the sender, allowing unrestricted modification of

Table II: Levels of security

	Confidentiality	Integrity	Authenticity
Level 1	✓	✓	✓
Level 2	✓	✓	✗
Level 3	✓	✗	✓
Level 4	✗	✓	✓
Level 5	✓	✗	✗
Level 6	✗	✓	✗
Level 7	✗	✗	✓
Level 8	✗	✗	✗

the network traffic in transit. Security levels 2, 5-6, and 8 are vulnerable to these attacks.

We have established that in order to provide confidentiality and integrity to the communication, entities must perform a key agreement procedure beforehand. In the remainder of this paper, the key agreement procedure will be referred to as the protocol, not the subsequent communication. Furthermore, the protocol is extended to an authenticated key agreement (AKA) one. As a result, entities will be able to generate a session key and authenticate each other at the same time. Thus, the AKA protocol will enable security level 1 in subsequent communication.

C. Authenticated Key Agreement

One of the first public-key protocol for key agreement was proposed in 1976 by W. Diffie and M. Hellman [5]. The Diffie-Hellman (DH) key exchange method allows two (or more) users to securely derive a secret key via unsecured channels. The security of the DH method is based on the difficulty of calculating discrete logarithms in a finite field, while its efficiency is based on the ease of calculating exponentiation in the same field. In other words, an adversary cannot recover the session key even if it eavesdrops the protocol execution. However, the DH method is vulnerable to MiM attacks, due to the absence of user authentication. Subsequent proposals to the DH method vary in the way mutual authentication is performed. We will briefly describe four AKA categories: password-based, certificate-based, identity-based, and certificateless-based.

a) Password-based: a trivial approach to mutual authentication is to share a secret password among users. The security of password-based protocols relies on the difficulty of guessing (brute-force) the password. One example is the Encrypted Key Exchange (EKE) scheme [6], where users rely on a secret to authenticate each other, and to generate a random public key. Because the password encrypts the public key, it it is impossible to guess it without cracking the public key algorithm. The EKE protocol however fails to provide joint-key control, as one user arbitrarily chooses the session key. More generally, the main drawback of this approach is that users need to agree on a password a priori. Because it must remain secret, this is suitable only for private networks. Other password-based key exchange protocols include, among others, Fortified Key Negotiation [7], SPEKE [8], AuthA [9], PAK/PPK [10], and Dragonfly [11].

b) Certificate-based: another simple solution is to combine the DH method with a digital signature scheme. This approach relies on the use of digital certificates. Each user is in possession of private/public key pair, of which the public key is signed by a trusted third party. Each user must obtain a certificate with the other user's public key. They verify the certificate with the third party's public key, and proceed with the key exchange process. The Station-To-Station (STS) protocol [12] is a variant of the DH method, where users demonstrate knowledge of the shared key by encrypting their signatures, making authentication straightforward. This method is also resilient to MiM attacks. However, the length of exchanged messages is greatly increased as data needs to be encrypted and/or signed, augmenting the performance overhead. Furthermore, certificate-based schemes are dependent on a Public Key Infrastructure (PKI). Certificate management includes creating, distributing, storing, exchanging, verifying, and revoking certificates. These processes are time-consuming, error-prone, and their complexity often leads its users to prefer a plain communication. Other certificate-based AKA protocols have been proposed, such as MQV (revised by Law *et al.*) [13], YAK [14], and ISAKMP [15].

c) Identity-based: in 1984, A. Shamir presented Identity-Based Encryption (IBE) [16] as an alternative to PKI. Its main advantage is that it eliminates the need for certificate management. Later on, identity-based AKA protocols were proposed [17], [18] based on bilinear pairings. In the initialization phase, a trusted third party, called Key Generation Center (KGC), is responsible for issuing private/public key pairs to every user. During the key exchange, users exchange ephemeral keys, and compute a unique session key, as in the DH method. This method enables, among other security attributes, mutual authentication and key confirmation. In addition, the PKI is eliminated. On the other hand, the KGC is in possession of users' secret key. Thus, a malicious KGC is capable of recovering every session key and performing standard MiM attacks. This is referred as the key escrow problem. Other influential identity-based AKA protocols have been proposed by Chen and Kudla [19], Okamoto *et al.* [20], McCullagh *et al.* [21], Ring *et al.* [22], and others.

d) Certificateless-based: in 2003, Al-Riyami *et al.* published a seminal work on CertificateLess-Public Key Cryptography (CL-PKC) [23]. It included, among other security primitives, a CertificateLess-Authenticated Key Agreement (CL-AKA) protocol. In this approach, the trusted third party delivers a partial private key to each user. As a result, their CL-AKA protocol offers the same advantages as in identity-based AKA, while eliminating the key escrow problem. During the key agreement phase, entities exchange ephemeral keys and compute the session key using a bilinear pairing. Therefore, the certificateless approach has the advantage to eliminate the need for pre-established secrets, certificate management, and key escrow. Therefore, the certificateless approach seems to be the most advantageous for security and performance

considerations.

D. Group-based Authenticated Key Agreement

While the majority of CL-AKA protocols have been proposed for two-party key agreement, fewer address the problem of group key agreement [4], [24]–[27], i.e. more than two entities. A group-based protocol is of significant interest for swarm-based applications, as a single key is required for an entire fleet. In 2012, Teng *et al.* [4] proposed a group key agreement with constant rounds, meaning that the number of rounds is independent from the number of users. Considering that a set of messages must be broadcast after each round, this feature is of particular interest in UANET applications. Furthermore, it includes the majority of the desired security attributes, apart from revocation and conditional privacy. Finally, our scheme makes use of asymmetric bilinear pairing, for performance consideration. Therefore, we will use [4] as a basis for the proposed CL-GAKA scheme.

E. Rationale

We have compared in section II.A the characteristics among UANETs, VANETs, and WSNs. It has been shown that UANETs deal with outstanding constrains including autonomy, node mobility, network connectivity, and communication efficiency. Furthermore, we have examined the security properties of a communication protocol in open environments, and concluded that it must capture at least confidentiality, integrity, and authenticity. Finally, several approaches to the entity (UAV) authentication problem have been studied, and the certificateless approach has shown to be the most suitable one.

III. Proposed Protocol

This section describes the desired security and operational security attributes. Presented heretofore is the CL-GAKA protocol.

A. Scope of the Paper

This paper addresses UAV-to-UAV secure channel establishment. UAV-to-Infrastructure communication, as well as the routing problem, are out of the scope of this paper. Additionally, by enforcing entity authentication, it provides ground for the implementation of a trust mechanism. The trust mechanism itself is beyond this proposal. Finally, the group key establishment protocol is designed for static groups. The problem of dynamic UAV inclusion and delisting will be the subject of future research.

B. Security & Operational Requirements

The proposed key exchange protocol uses a distributed collaboration model. In other words, each entity has the same amount of responsibilities toward other nodes in the network. Security requirements include,

- **Mutual authentication** allows entities to authenticate each other, preventing an adversary from perpetrating spoofing attacks.
- **Key escrow elimination** protects entities against a malicious third party, as the third party only holds a portion of users' secret keys.
- **Mutual key agreement** guarantees that every communicating entity takes part in the key generation process (unlike in a key distribution scheme).
- **Joint key control** prevents one entity from influencing the outcome of the key generation process, e.g. by predetermining a portion of the session key.
- **Key freshness** ensures that a new session key is generated for every new communication session.
- **Entity revocation** prevents an entity from establishing new sessions if its certificate has been revoked. The third party delivers partial private keys that have a validity period, e.g. per flight.
- **Non-repudiation** precludes an entity from repudiating the establishment of a previous session (thus preventing exhaustion attacks).
- **Conditional privacy** allows entities to communicate anonymously by using pseudo-identities. Only the trusted third-party is able to recover any entity's identity from its pseudo-identity.
- **Known-key security** prevents an adversary from learning long-term secrets or session keys (future and past), if it is in possession of a particular session's key.
- **Perfect forward secrecy** precludes an adversary from learning previous session keys, if it is in possession of long-term secrets of other entities.

C. Definitions & Notations

This section briefly describes notions related to the certificateless scheme. Table III provides the list of notations used in the proposed protocol.

a) Admissible pairings: given a cyclic additive group $\mathbb{G}_1$ of prime order q, and a cyclic multiplicative group $\mathbb{G}_2$ of the same order, an admissible pairing is a map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ satisfying the following properties:

- Bilinear: $e(aP, bQ) = e(P, Q)^{ab}$ for all $P, Q \in \mathbb{G}_1$ and $a, b \in \mathbb{Z}_q^*$.
- Non-degenerate: $e(P, Q) \neq 1$.
- Computable: $e(P, Q)$ can be computed in polynomial time for all $P, Q \in \mathbb{G}_1$.

b) Bilinear Diffie-Hellman (BDH) assumption: the BDH problem forms the basis of security of our scheme. It is defined as follows: given a tuple $\langle P, aP, bP, cP \rangle$ with $a, b, c \in \mathbb{Z}_q^*$ uniformly randomly chosen, compute $e(P, Q)^{abc}$. By using the BDH assumption, it is considered that the BDH is infeasible, where there is no polynomial time algorithm to solve the BDH problem with non-negligible probability.

D. Proposed Protocol

The initialization phase consists of five algorithms. It is performed offline and prepares entities for the online key agreement procedure. The proposed initialization differs from [4] in the way the user's partial private key is created. The partial-private-key-extract algorithm has been modified

in order to enforce entity revocation and conditional privacy. Additionally, the original protocol uses a symmetric pairing while ours makes the use of an asymmetric one, allowing greater computational efficiency.

a) Initialization:

- **Setup** is run by the KGC. It outputs a set of public parameters,
  - Given three groups $\mathbb{G}_1$, $\mathbb{G}_2$, and $\mathbb{G}_T$ of some prime order q, choose a pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$.
  - Choose two random generators $P \in \mathbb{G}_1$, $Q \in \mathbb{G}_2$, and compute $g = e(P, Q) \in \mathbb{G}_T$.
  - Select a secret key $s \in \mathbb{Z}_q^*$ and compute the public key $P_0 = sP \in \mathbb{G}_1$.
  - Choose three cryptographic hash functions $H_1 : \{0,1\}^* \to \mathbb{Z}_q^*$, $H_2 : \mathbb{G}_2 \times \{0,1\}^* \to \{0,1\}^k$, and $H_3 : \{0,1\}^* \to \{0,1\}^l$.
  - Publish parameters *params* as $\langle \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P, Q, P_0, g, H_1, H_2, H_3 \rangle$.
- **Partial-private-key-extract** is run by the KGC. It takes as input *params*, the user's temporary identity TID_i, a time stamp t_i, and a random string m_i. It outputs the user's partial private key $D_i = (q_i + s)^{-1}Q \in \mathbb{G}_2$, where $q_i = H_1(TID_i||t_i)$ and $TID_i = H_3(m_i) \oplus ID_i$.
- **Set-secret-value** is run by the user. It outputs a random integer $x_i \in \mathbb{Z}_q^*$.
- **Set-private-key** is run by the user. It takes as input *params* and the partial private key D_i. It outputs the user's private key $\langle x_i, D_i \rangle$.
- **Set-public-key** is run by the user. It takes as input *params* and x_i. It outputs the user's public key $\langle q_i, P_i \rangle$, with $P_i = g^{x_i} \in \mathbb{G}_T$.

b) Group Key Agreement:

- **Setup.** Each user u_i sends a key establishment request, containing its temporary identity TID_i, partial public key q_i, and expiration date/time t_i of its partial public key.
- **Round 1.** Upon reception of the request, each user u_i verifies that t_j is not out-of-date. If the verification is invalid, u_i terminates the protocol, otherwise u_i chooses a random $r_i \in \mathbb{Z}_q^*$, $k_i \in \{0,1\}^k$, and generates a set of ephemeral keys $P_{i,j} = r_i(q_j P + P_0) = r_i(q_j + s)P$ for $1 \leq j \leq n$ and $j \neq i$. Each user u_i then broadcasts the set of $P_{i,j}$ along with $H_3(k_i)$.
- **Round 2.** Upon reception of $H_3(k_j)$ and $P_{j,i}$, each user u_i computes $sid_i^w = H_3(k_1)||...||H_3(k_n)$. Each user u_i then generates the set of $t_{j,i} = e(P_{j,i}, D_i)^{x_i} P_j^{r_i} = g^{r_j x_i + r_i x_j}$, $V_{j,i} = H_2(t_{j,i}||sid_i^w)$, and $K_{j,i} = V_{j,i} \oplus k_i$. The set of $K_{j,i}$ is broadcast.
- **Key generation.** Upon reception of $K_{i,j}$, each user u_i computes $\tilde{k}_j = V_{j,i} \oplus K_{i,j}$ and checks whether $H_3(\tilde{k}_j) = H_3(k_j)$ is valid. Upon successful verification, each user u_i generates the session key $sk_i^w = H_3(k_1||...||k_n||sid_i^w||pid_i^w)$.

Table III: List of notations in the proposed protocol

Notation	Description
q	Large prime number
e	Bilinear map, $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$
P	Generator point of $\mathbb{G}_1$
Q	Generator point of $\mathbb{G}_2$
s	KGC's private key
P_0	KGC's public key, $P_0 = sP$
H_1	Hash function, $H_1 : \{0,1\}^* \to \mathbb{Z}_q^*$
H_2	Hash function, $H_2 : \mathbb{G}_2 \times \{0,1\}^* \to \{0,1\}^k$
H_3	Hash function, $H_3 : \{0,1\}^* \to \{0,1\}^l$
u_i	i-th user $(1 \leq i \leq n)$
ID_i	Identity of u_i, of bit-length l
m_i	Random string
TID_i	Temporary identity of u_i, $TID_i = H_3(m_i) \oplus ID_i$
t_i	Time stamp of u_i's public key validity
q_i	Partial public key of u_i, $q_i = H_1(TID_i||t_i)$
D_i	Partial private key of u_i, $D_i = (q_i + s)^{-1}P$
x_i	Secret value randomly chosen from $\mathbb{Z}_q^*$ by u_i
P_i	Public value generated by u_i, $P_i = g^{x_i}$
sid_i^w	Session identifier for instance i of user w
pid_i^w	Partner identifier for instance i of user w
$\|\|$	Concatenation operator
$\oplus$	XOR operator

E. Comparison Against Existing Protocols

Table IV provides a comparative analysis among several CL-GAKA protocols [4], [24], [25], [27]–[31], and Won *et al.*'s proposal [32]. It shows that all CL-GAKA protocols encompass mutual key agreement, key escrow elimination, mutual key agreement, joint key control, key freshness, and known-key security. However, very few address entity revocation, non-repudiation, and conditional privacy. Among the closest proposals to the proposed protocol, [31] fails to provide forward secrecy. [32] doesn't account for joint key control or mutual key agreement, and their proposal is not suitable with group key establishment. Though it is worth mentioning that this protocol addresses two-party UAV-to-smart object key agreement. While [27] proposes a pairing-free anonymous scheme, it requires the KGC to be involved in the online key agreement phase. Because the KGC might not be available to each entity in operation, [4] is preferred as the basis for our UAV-to-UAV CL-GAKA scheme.

IV. Protocol Evaluation

This section provides the implementation details along with a performance evaluation. The results of the formal analysis are then outlined.

A. Implementation & Performance Evaluation

The testbed is described in figure 1. It consists of two Raspberry Pi 3 Model B+, and a wireless router. Each Raspberry Pi is supplied with a 1.4GHz 64-bit quad-core ARMv8 processor, as well as a Wi-Fi dongle.

The protocol has been implemented in C language, and makes use of the PBC library [33]. Because the initialization

Table IV: Comparative analysis among several certificateless protocols for authenticated key agreement

	[24]	[28]	[25]	[29]	[30]	[31]	[32]	[27]	[4]	Proposed protocol
Mutual authentication	●	●	●	●	●	●	●	●	●	●
Key escrow elimination	●	●	●	●	●	●	●	●	●	●
Mutual key agreement	●	●	●	●	●	●	-	●	●	●
Joint key control	●	●	●	●	●	●	-	●	●	●
Key freshness	●	●	●	●	●	●	●	●	●	●
Entity revocation	-	-	-	-	-	-	●	-	-	●
Non-repudiation	-	-	-	-	-	●	●	-	●	●
Conditional privacy	-	-	-	-	-	-	-	●	-	●
Known-key security	●	●	●	●	●	●	●	●	●	●
Forward secrecy	-	●	●	●	●	-	●	●	●	●

phase is performed offline, it is not measured. Figure 2 shows the performance results for the online key agreement phase, including the communication overhead. It is observed that, over two hundred simulations, and after removing outliers, the average elapsed time lies between 580ms and 620ms. Figure 3 shows further detailed performance results. The key agreement is first performed between two Raspberry Pi nodes, each communicating via its wireless local area network (WLAN) interface (figure 1). The total elapsed time is measured as 598ms, while the first and second round last respectively 13ms and 417ms. In order to measure the communication overhead, the same implementation is then executed over the loopback interface. Thus a single node is emulating both client and server. The total elapsed time is evaluated as 471ms, while the first and second round last respectively 9ms and 345ms. Table V describes the number of operations performed in each round. It outlines that the second round includes $n-1$ bilinear pairing operations, where n is the number of entities (e.g. for two entities, each must perform one bilinear pairing). Because such an operation is computationally very expensive, the second round is responsible for 71% of the total elapsed time.

Furthermore, by comparing the results between the first and second setup (figure 3), the communication overhead is calculated as 21% of the total elapsed time. The remaining overhead is caused by the key agreement setup, key generation,

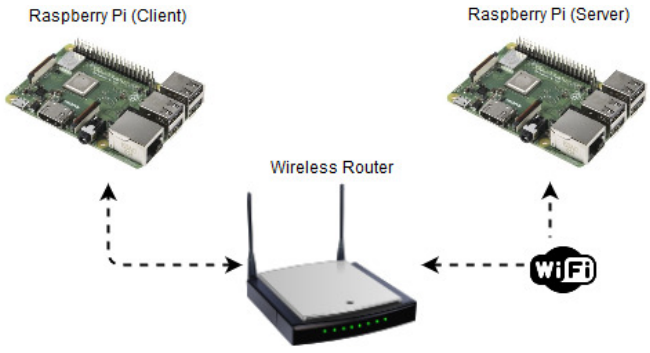

Figure 1: Testbed

Table V: Complexity analysis

	Round 1	Round 2
Bilinear pairing	-	$n-1$
Modular exponentiation	-	$2(n-1)$
Elliptic curve point multiplication	$2(n-1)$	$n-1$
Elliptic curve point addition	$n-1$	-

and further code execution.

B. Formal Analysis

The Scyther tool has been chosen to formally analyze the protocol. Scyther is used to find attack paths in security protocols under the perfect cryptography assumption. It takes as input a high-level description of the protocol, and evaluates the desired security properties, i.e. security claims. The role of each communicating entity is defined as a set of events, such as sending and receiving messages. Internal computations have been simplified as one-way functions, and a key confirmation round has been added for the sake of the evaluation. Our protocol description is given in Appendix A, with Alice being the initiator, and Bob the responder. Scyther succeeded in proving the following security claims,

- Secrecy of the session key.
- Aliveness.
- Weak agreement.
- Non-injective agreement.
- Non-injective synchronisation.

V. Conclusion & Further Work

In this paper, we have outlined the requirements for secure UAV-to-UAV communication over open networks. In order to meet these requirements, a certificateless-authenticated key agreement scheme has been proposed. This approach enables the elimination of the certificate management and the key-escrow problems. As a result, the key exchange protocol is highly suitable for resource-constrained devices. Furthermore, the proposed protocol accounts for groups of entities,

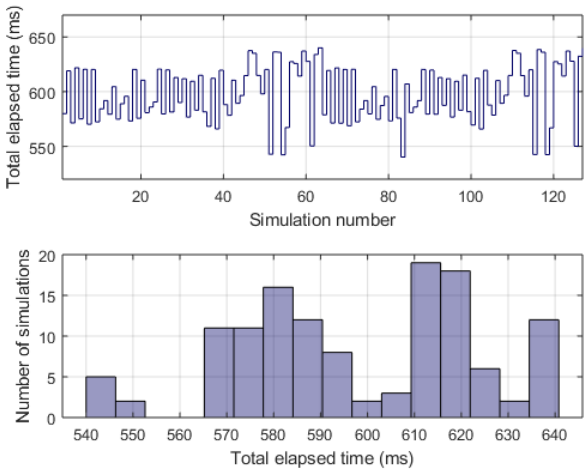

Figure 2: Online key agreement performance results

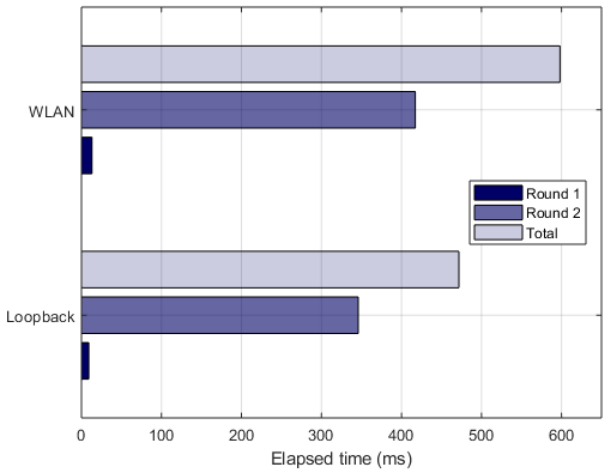

Figure 3: Detailed online key agreement performance results

allowing more than two entities to generate a symmetric cryptographic key at once. Several theoretical works have been studied against the desired security properties. In this paper, we proposed a certificateless group authenticated key agreement protocol, based on [4]. The implementation shows that the online key agreement phase can be performed in approximately 600ms between two entities, on a Raspberry Pi 3 Model B+ system-on-chip (1.4GHz Cortex-A53). Finally, the proposed scheme has been evaluated with the Scyther tool, showing that our protocol is resilient against several attacks.

The work presented in this paper provides ground for applications such as collaborative cybersecurity deterrence, network extension using trusted relay nodes, collaborative mission exercise in time-critical applications, or anonymous communication for user privacy. However, further research is required. The presented protocol is currently suitable for static groups only, i.e. the number of communicating entities is constant during the session. In future work, the protocol must encompass dynamic groups, allowing dynamic inclusion and delisting of entities in the same session. Furthermore, the problem of misbehavior from authenticated entities has not been considered. Further work will address this challenge with reputation or trust-based mechanisms.

Acknowledgment

We thank Bertram Poettering for helpful comments and discussions on implementation decisions.

References

[1] S. Kamkar, "Skyjack," http://www.samy.pl/skyjack/, 2013, [Online].

[2] B. Chapman, "Build a wi-fi drone disabler with raspberry pi," https://makezine.com/projects/build-wi-fi-drone-disabler-with-raspberry-pi/, 2016, [Online].

[3] S. Hayat, E. Yanmaz, and R. Muzaffar, "Survey on unmanned aerial vehicle networks for civil applications: A communications viewpoint," *IEEE Communications Surveys Tutorials*, vol. 18, no. 4, pp. 2624–2661, 2016.

[4] J. Teng and C. Wu, "A provable authenticated certificateless group key agreement with constant rounds," *Journal of Communications and Networks*, vol. 14, no. 1, pp. 104–110, 2012.

[5] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[6] S. M. Bellovin and M. Merritt, "Encrypted key exchange: password-based protocols secure against dictionary attacks," in *Proceedings 1992 IEEE Computer Society Symposium on Research in Security and Privacy*, 1992, pp. 72–84.

[7] R. J. Anderson and T. M. A. Lomas, "Fortifying key negotiation schemes with poorly chosen passwords," *Electronics Letters*, vol. 30, no. 13, pp. 1040–1041, 1994.

[8] D. P. Jablon, "Strong password-only authenticated key exchange," *SIGCOMM Comput. Commun. Rev.*, vol. 26, no. 5, pp. 5–26, 1996.

[9] M. Bellare and P. Rogaway, "The autha protocol for password-based authenticated key exchange," 2000.

[10] V. Boyko, P. MacKenzie, and S. Patel, "Provably secure password-authenticated key exchange using diffie-hellman," in *Advances in Cryptology — EUROCRYPT 2000*, B. Preneel, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 156–171.

[11] D. Harkins, "Simultaneous authentication of equals: A secure, password-based key exchange for mesh networks," in *2008 Second International Conference on Sensor Technologies and Applications (sensorcomm 2008)*, 2008, pp. 839–844.

[12] W. Diffie, P. V. Oorschot, and M. Wiener, "Authentication and authenticated key exchanges," *Designs, Codes and Cryptography*, vol. 2, no. 2, pp. 107–125, 1992.

[13] L. Law, A. Menezes, M. Qu, J. Solinas, and S. Vanstone, "An efficient protocol for authenticated key agreement," *Designs, Codes and Cryptography*, vol. 28, no. 2, pp. 119–134, 2003.

[14] F. Hao, "On robust key agreement based on public key authentication," in *Financial Cryptography and Data Security*, R. Sion, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 383–390.

[15] D. Maughan, M. Schertler, M. Schneider, and J. Turner, "Internet security association and key management protocol (isakmp)," 1998.

[16] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Advances in Cryptology*, G. R. B. D. and Chaum, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1985, pp. 47–53.

[17] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," in *Advances in Cryptology — CRYPTO 2001*, J. Kilian, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 213–229.

[18] N. P. Smart, "Identity-based authenticated key agreement protocol based on weil pairing," *Electronics Letters*, vol. 38, no. 13, pp. 630–632, 2002.

[19] L. Chen and C. Kudla, "Identity based authenticated key agreement protocols from pairings," in *16th IEEE Computer Security Foundations Workshop, 2003. Proceedings.*, 2003, pp. 219–233.

[20] T. Okamoto, R. Tso, and E. Okamoto, "One-way and two-party authenticated id-based key agreement protocols using pairing," in *Modeling Decisions for Artificial Intelligence*, V. Torra, Y. Narukawa, and S. Miyamoto, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 122–133.

[21] N. McCullagh and P. S. L. M. Barreto, "A new two-party identity-based authenticated key agreement," in *Topics in Cryptology – CT-RSA 2005*, A. Menezes, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 262–274.
[22] J. W. Ring, K.-K. R. Cho, E. Foo, and M. H. Looi, "A new authentication mechanism and key agreement protocol for sip using identity-based cryptography," in *AusCERT Asia Pacific Information Technology Security Conference 2006*, A. Clark, M. McPherson, and G. Mohay, Eds. Gold Coast, Australia: AusCERT, 2006.
[23] S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *Advances in Cryptology - ASIACRYPT 2003*, C.-S. Laih, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 452–473.
[24] S. Heo, Z. Kim, and K. Kim, "Certificateless authenticated group key agreement protocol for dynamic groups," in *IEEE GLOBECOM 2007 - IEEE Global Telecommunications Conference*, 2007, pp. 464–468.
[25] M. Geng, F. Zhang, and M. Gao, "A secure certificateless authenticated group key agreement protocol," in *2009 International Conference on Multimedia Information Networking and Security*, vol. 1, 2009, pp. 342–346.
[26] S. Islam and A. Singh, "Provably secure one-round certificateless authenticated group key agreement protocol for secure communications," *Wireless Personal Communications*, vol. 85, no. 3, pp. 879–898, 2015.
[27] A. Kumar and S. Tripathi, "A pairing free anonymous certificateless group key agreement protocol for dynamic group," *Wireless Personal Communications*, vol. 82, no. 2, pp. 1027–1045, 2015.
[28] E. J. Lee, S. E. Lee, and K. Y. Yoo, "A certificateless authenticated group key agreement protocol providing forward secrecy," in *2008 International Symposium on Ubiquitous Multimedia Computing*, 2008, pp. 124–129.
[29] H. Sun, Q. Wen, H. Zhang, and Z. Jin, "A novel pairing-free certificateless authenticated key agreement protocol with provable security," *Frontiers of Computer Science*, vol. 7, no. 4, pp. 544–557, 2013.
[30] G. Xiaozhuo, X. Taizhong, Z. Weihua, and W. Yongming, "A pairing-free certificateless authenticated group key agreement protocol," in *2014 IEEE Intl Conf on High Performance Computing and Communications, 2014 IEEE 6th Intl Symp on Cyberspace Safety and Security, 2014 IEEE 11th Intl Conf on Embedded Software and Syst (HPCC,CSS,ICESS)*, 2014, pp. 510–513.
[31] C. Cao, J. Ma, and S. Moon, "Provable efficient certificateless group key exchange protocol," *Wuhan University Journal of Natural Sciences*, vol. 12, no. 1, pp. 41–45, 2007.
[32] J. Won, S.-H. Seo, and E. Bertino, "A secure communication protocol for drones and smart objects," in *Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security*, ser. ASIA CCS '15. New York, NY, USA: ACM, 2015, pp. 249–260.
[33] B. Lynn, "Pbc library - pairing-based cryptography - about," https://crypto.stanford.edu/pbc/, [Online].

APPENDIX A
SCYTHER SCRIPT

```
hashfunction H3, Pij, Kji, kj;
usertype SessionKey;

protocol clgaka(Alice, Bob)
{
  role Alice
  {
    fresh TIDA: Ticket;
    fresh qA: Ticket;
    fresh tA: Ticket;
    fresh DA: Ticket;
    fresh rA: Nonce;
    fresh kA: Nonce;

    var TIDB, qB, tB;
    var PBA;
    var hashOfkB;
    var KAB;
    var kB;
    var sk: SessionKey;

    send_1(Alice, Bob, TIDA, qA, tA);
    recv_2(Bob, Alice, TIDB, qB, tB);

    send_3(Alice, Bob, Pij(rA, qB), H3(kA));
    recv_4(Bob, Alice, PBA, hashOfkB);

    send_5(Alice, Bob, Kji(H3(kA), hashOfkB, PBA, DA
        , rA, kA));
    recv_6(Bob, Alice, KAB);

    match(kB, kj(H3(kA), hashOfkB, PBA, DA, rA, KAB)
        );
    match(sk, H3(kA, kB, H3(kA), hashOfkB, TIDA,
        TIDB));
    send_7(Alice, Bob, {TIDA, Pij(rA, qB), Kji(H3(kA
        ), hashOfkB, PBA, DA, rA, kA)}sk);
    recv_8(Bob, Alice, {TIDB, PBA, KAB}sk);

    claim(Alice, Secret, sk);
    claim(Alice, Alive);
    claim(Alice, Weakagree);
    claim(Alice, Nisynch);
    claim(Alice, Niagree);
  }

  role Bob
  {
    fresh TIDB: Ticket;
    fresh qB: Ticket;
    fresh tB: Ticket;
    fresh DB: Ticket;
    fresh rB: Nonce;
    fresh kB: Nonce;

    var TIDA, qA, tA;
    var PAB;
    var hashOfkA;
    var KBA;
    var kA;
    var sk: SessionKey;

    recv_1(Alice, Bob, TIDA, qA, tA);
    send_2(Bob, Alice, TIDB, qB, tB);

    recv_3(Alice, Bob, PAB, hashOfkA);
    send_4(Bob, Alice, Pij(rB, qA), H3(kB));

    recv_5(Alice, Bob, KBA);
    send_6(Bob, Alice, Kji(hashOfkA, H3(kB), PAB, DB
        , rB, kB));

    match(kA, kj(hashOfkA, H3(kB), PAB, DB, rB, KBA)
        );
    match(sk, H3(kA, kB, hashOfkA, H3(kB), TIDA,
        TIDB));
    recv_7(Alice, Bob, {TIDA, PAB, KBA}sk);
    send_8(Bob, Alice, {TIDB, Pij(rB, qA), Kji(
        hashOfkA, H3(kB), PAB, DB, rB, kB)}sk);

    claim(Bob, Secret, sk);
    claim(Bob, Alive);
    claim(Bob, Weakagree);
    claim(Bob, Nisynch);
    claim(Bob, Niagree);
  }
}
```