Proof of Security of Mutual Authentication and Key Exchange in the L-band Digital Aeronautical Communication System LDACS

Abstract. Communication is the invisible backbone of the modern air transportation system. Aeronautical data links enable ground crews to communicate with air crews and guide them safely and efficiently to any destination in the world. However, since air traffic is growing rapidly, currently deployed systems experience capacity problems and need modernization. Several new aeronautical data links with different deployment areas shall therefore be introduced. The L-band Digital Aeronautical Communications System (LDACS) is foreseen for terrestrial communication in continental air spaces. The implementation of state-of-the-art cybersecurity is an integral part of this modernization and an LDACS security architecture has consequently been developed. However, as of now, its correctness has not been proven. The contribution of this paper is the formal proof of the security of the Mutual Authentication and Key Exchange (MAKE) protocol of LDACS using the symbolic model checker Tamarin. Our results prove, that the suggested MAKE procedure for LDACS is secure in the standard model.

Keywords: LDACS, Security, Authentication, Key Exchange, Symbolic Model Checker, Tamarin

1 Introduction

Civil air traffic is growing fast [17] and despite the COVID-19 pandemic, which has temporarily reduced the world air traffic passenger numbers at some points by 35% compared to the pre-COVID-19 level, the International Civil Aviation Organization (ICAO) anticipates air traffic passenger numbers to grow from 2022 onward again [23]. In the long term, the International Air Traffic Association (IATA) estimates air traffic passenger numbers to double to 8.2 billion in 2037 compared to 4 billion in 2018 [22].

However, most aeronautical communication systems in operation today suffer already from capacity issues [43]. The rise of new entrants such as drones, UAVs, single piloted or autonomous aircraft [20] and ultimately the lack of spectrum, require therefore a paradigm shift in aeronautical communications.

In order to overcome the capacity constraints of the legacy analogue systems, digitalization of aeronautical communications is necessary [41] and currently underway [43]. To allow this process to occur smoothly, confidentiality, integrity and availability of data are key. Safety and security are strongly interrelated in aviation [41,42]. Cybersecurity becomes thus a key enabler for the modernization of civil aviation [20,30].

Unfortunately, cybersecurity for Communication, Navigation and Surveillance (CNS) systems is not realized in most deployed aeronautical systems [12, 33, 44]. To facilitate change and support the digitalization process, there are several initiatives e.g. the Single European Sky ATM Research (SESAR)[1] program in the EU, and NextGEN[2] in the US. These initiatives envision a Future Communications Infrastructure (FCI) utilizing multiple secure digital aeronautical data links. Candidate data link technologies for the FCI are the L-band Digital Aeronautical Communication System (LDACS) for long-range terrestrial aeronautical communications [39], Iris satellite communications for oceanic, polar and remote areas, and the Aeronautical Mobile Airport Communication System (AeroMACS) for airport communications. Of these candidate technologies AeroMACS has the most mature cybersecurity architecture. Originating from Worldwide Interoperability for Microwave Access (WiMAX) with a dedicated security layer [18] and a Public Key Infrastructure (PKI) in place [15], AeroMACS may serve as the blueprint for other systems. As LDACS is next in line with SESAR's FCI strategy, the development of a strong cybersecurity architecture for LDACS is paramount.

A cybersecurity architecture for LDACS has been published in [27–30]. However, a formal prove of the security of the Mutual Authentication and Key Exchange (MAKE) procedure that is fundamental to the security of the overall system has not been published yet.

The objective of this paper is thus to provide a formal analysis of the proposed mutual authentication and key agreement protocols of LDACS and to prove their security.

In section 2 we introduce the L-band Digital Aeronautical Communication System (LDACS). In section 3 we introduce our method for symbolic modeling and analysis of security protocols. In section 4 we discuss the LDACS MAKE protocol and its security objectives. We derive the lemmata to be proved by our method and state our assumptions. In section 5, we present the results from the formal security proof and discuss them in section 6. Finally, we conclude with key findings, recommendations and future work in section 7.

2 Background

LDACS is a ground-based cellular digital aeronautical communications system for flight guidance and communications related to the safety and regularity of flight. LDACS has been developed in Europe and is currently under standardization in ICAO. It supports Air Traffic Services (ATS) and Aeronautical Operational Control (AOC) applications. It has been designed with future applications in mind, such as 4D trajectory management, and offers therefore at least 10 times more net capacity than the currently used terrestrial link i.e. the VHF Digital Link (VDL) Mode 2 system [19]. Instead of kilobits per second, LDACS offers up

[1] `https://www.sesarju.eu/`, last access June 20, 2020
[2] `https://www.faa.gov/nextgen/`, last access June 20, 2020

to 2 Mbps. By enabling not only communication but also navigation and surveillance through the use of the same radio signals, it is the world's first integrated CNS system [39].

The LDACS network is comprised of several radio cells and is controlled by one Ground Station Controller (GSC). Each radio cell has a transmission site, called a Ground Station (GS) and can serve up to 512 Aircraft Stations (AS). This is illustrated in Figure 1.

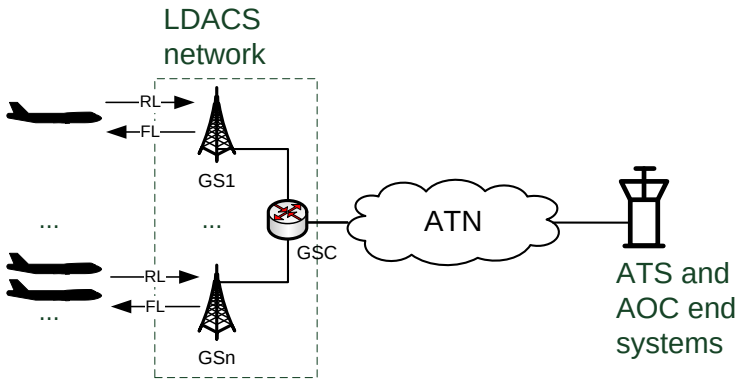

Fig. 1. LDACS network architecture: Aircraft systems (AS) connect to LDACS ground-stations (GS) controlled by a ground-station controller (GSC). The GSC connects the LDACS sub-network to the global aeronautical telecommunications network (ATN) to which the corresponding ATS and AOC end systems are attached.

In [27] the initial LDACS's cybersecurity architecture was analyzed, leading to an updated cybersecurity architecture and integration of new algorithms as published in [28,30]. The approach was successfully evaluated in [29] and provides the baseline for this work.

3 Symbolic Modeling and Analysis of Security Protocols

In order to prove the security of the authentication and key exchange protocols of LDACS we use software based model checking tools.

Model checking is a formal technique for systems verification developed since the 1980's [10] and has been successfully used e.g. for checking the correctness of hardware designs of microchips [8] but also for checking concurrent (software) systems and communication protocols [21].

The main idea is to specify the target system as a state-transition graph (or Kripke structure) which can be analyzed easier than a full implementa-

tion in e.g. Very High Speed Integrated Circuit Hardware Description Language (VHDL) or C-code and to verify that a given specification or property of this system holds within its complete state-space. These properties are often safety-properties like deadlock freedom, or liveliness-properties like termination- or response-guarantees and are usually formalized with a temporal-logic formula (LTL/CTL).

If successful, the model checker can show, that a certain state of the model (e.g. a "Crash") cannot be reached, or, that a state (e.g. "Stop") will be reached if certain conditions are met. Otherwise, the model checker outputs a counterexample that witnesses the violation of the specification by the behavior of the modeled system [9].

To be able to prove security properties of e.g. communication protocols, one has to additionally model the attacker, since he or she (possibly) plays an active part in each run of the protocol. Following the works of Dolev and Yao [16], the ideal, most powerful attacker is assumed, who can create, intercept or modify any message in the network, spoof any identity and even compromise long term keys. Often, this is implemented by sending every message to the attacker first, who then blocks or relays them unmodified to the honest agents at will.

In general, this can be done manually using a general-purpose model checker like SPIN [26], but more specialized cryptographic protocol checkers have the advantage of a built-in algebraic attacker model and predefined security properties like secrecy and agreement [4].

3.1 Choice of Model Checker

Shinde et al [40] provides an overview of tools for the specification and verification of cryptographic protocols.

We evaluated three of the most recent and actively developed tools of this list, namely: Scyther [14], ProVerif [6] and Tamarin [38]. The first and the last are (co-)developed by Cas Cremers, who also gives a comparison of the features of his tools in [13]. First, we tried Scyther which has good tool support and a concise syntax. Unfortunately, we could not model the MAKE protocol with it, since our authentication scheme uses exponentiation, which is not natively supported by Scyther. After this, we tried ProVerif which has support for this and achieved good results. But we had difficulties in proving the correctness of the protocol without having an active attacker. Finally, we chose Tamarin which is very similar to ProVerif in terms of features and capabilities but has the advantage of more flexibility in the protocol description due to its more general syntax which allows loops to occur and the possibility to prove arbitrary properties of the protocol. Tamarin provides two different ways of constructing proofs: a fully automated mode that guides proof search and an interactive mode. The termination of the automated process returns either a proof of correctness or an attack scenario. The interactive mode enables the user to inspect the attack graphs, proof states and combine the manual proof guidance with the automated search. From a user's perspective, Tamarin model checker allows for the direct specification of the security properties to be proven. The flexible

modeling framework and expressive language property makes Tamarin fit for analyzing a wide range of security properties. Tamarin model checker has shown strong results in related works to analyze many authenticated key exchange protocols [38]. In [32] Simon presented the symbolic analysis of security protocols using Tamarin prover. The AR-PKI protocol is presented and verified using Tamarin prover in [3] and the automated verification of Group Key Agreement protocols was presented in [37] using the Tamarin prover as well.

We chose to use Tamarin for this work. We see the advantage of the Tamarin model checking approach in its application in parallel to the protocol design, since changes in the protocol can easily be transferred to the model and vice versa. It delivers reproducible security proofs while maintaining flexibility towards further improvements. It can be used akin to a "unit test" in software development. Each change in the protocol can be applied to the model and automatically tested for defined security goals.

3.2 Method

Tamarin models are written in a domain-specific multiset rewriting formalism which defines a labeled transition system. The systems initial state is an empty multiset of facts, which gets filled by the creation of facts during the transitions of the system. All possible transitions of the system are defined by *rules*. Each rule has a left-hand side giving a sequence of required *facts* to describe the preconditions of the transition to execute. Also, each rule has a right-hand side, given as a sequence of the new facts created after the transition has been executed. Between those two, one or more *action facts* or *transition labels* can be defined. Each protocol run will then generate a trace consisting of this action facts. To reason about the behavior of the protocol, the behavior can be formalized as a trace property, which is a set of traces using first-order logic formulas over action facts and timepoints. These trace properties are called 'lemmata' and can be proven.

While modeling the LDACS MAKE protocol with Tamarin, we tried to follow the standards and recommendations from the Tamarin documentation as much as possible [1]. As there is no built-in *role* type in Tamarin, but only rules and facts, roles must be carefully modeled using *state* facts which link multiple rules together by carrying the state of a role from rule to rule. This way we modeled the roles of the aircraft station (AS) and the ground station controller (GSC). As recommended in the documentation, each role has an initial *creation* rule which also assigns a unique session-ID to distinguish multiple instances of the roles. The other rules for AS and GSC directly reflect their message exchange over an untrusted network as pictured above, by using the Tamarin built-in facts *In()* and *Out()*. The rule for the public key infrastructure basically allows anyone to register its public key in a public database so that anyone can verify e.g. a given signature and an ID from an incoming signed message. To make the verification task easier we added the restriction, that each actor can register only one key-pair per ID. To later check the protocol for perfect forward secrecy, we also added a rule for modeling the compromise of private keys to the attacker.

4 Mutual Authentication and Key Exchange in LDACS

Mutual authentication and key exchange (MAKE) in LDACS are deeply linked to the cell entry procedure.

The LDACS specification [19] provides a detailed overview on the LDACS cell entry procedure. Once a GS is securely connected to the aeronautical ground network via the GSC, it starts sending a broadcast message, the System Identification Broadcast (SIB) message, containing relevant information such as network identification, physical parameters such as channel frequencies and more. When an Aircraft Station (AS) enters the cell served by a GS, it receives the SIB and sends a CELL_RQST message in reply. The CELL_RQST message contains a "unique address identifying the LDACS radio" [19]. When the GS receives the CELL_RQST message, a CELL_RESP message is sent back to the AS, informing the AS about its Subscriber Access Code (SAC). The SAC is a local and temporary address for the AS in the cell. After this exchange of control channel messages, both communication parties are informed about LDACS specific addresses, timing, frequency, and power values and can start the user data communication. This is illustrated in step 1 of figure 2.

Up until now, no security information has been exchanged, however, previous threat and risk analyses [27, 28, 45] for LDACS have identified several safety-critical applications, that require security. In particular, applications supporting air traffic services and safety-related aeronautical operational control communications are of concern [29, 30].

Bilzhause et. al identified five objectives to secure LDACS [5]. These five objectives were later extended to nine objectives in the LDACS Standards and Recommended Practises (SARPS) agreed upon by ICAO [24].

Based on these objectives and results from threat and risk analysis, a first cybersecurity architectural draft was presented in [30]. Security protocols and algorithms were proposed and evaluated in [28] and a performance analysis of the LDACS MAKE protocol was published in [29]. In [31] we investigated the optimal choice of Diffie-Hellman key exchange procedures and improved the previously presented MAKE procedure substantially. The contribution of this paper is the formal proof of the security of this protocol.

The objective of the LDACS MAKE protocol is to establish a shared session key K between any two parties AS and GSC, in which they can have "mutual belief", following the definition of Boyd [7]: "Mutual belief in the key K is provided for B only if K is a good key for use with A, an A wishes to communicate wit B using key K which A believes is good for that purpose." Following the hierarchy of authentication and key establishment goals of Boyd, this mutual belief goal can be split up into the sub-goals *entity authentication*, *key confirmation* and *good key*. Additionally, we want to address the issue of compromised long-term keys. We summarize the objectives of the LDACS MAKE protocol therefore as such:

Mutual Authentication: Both parties can be sure of the identity of the other and that both actually participated in this interaction.

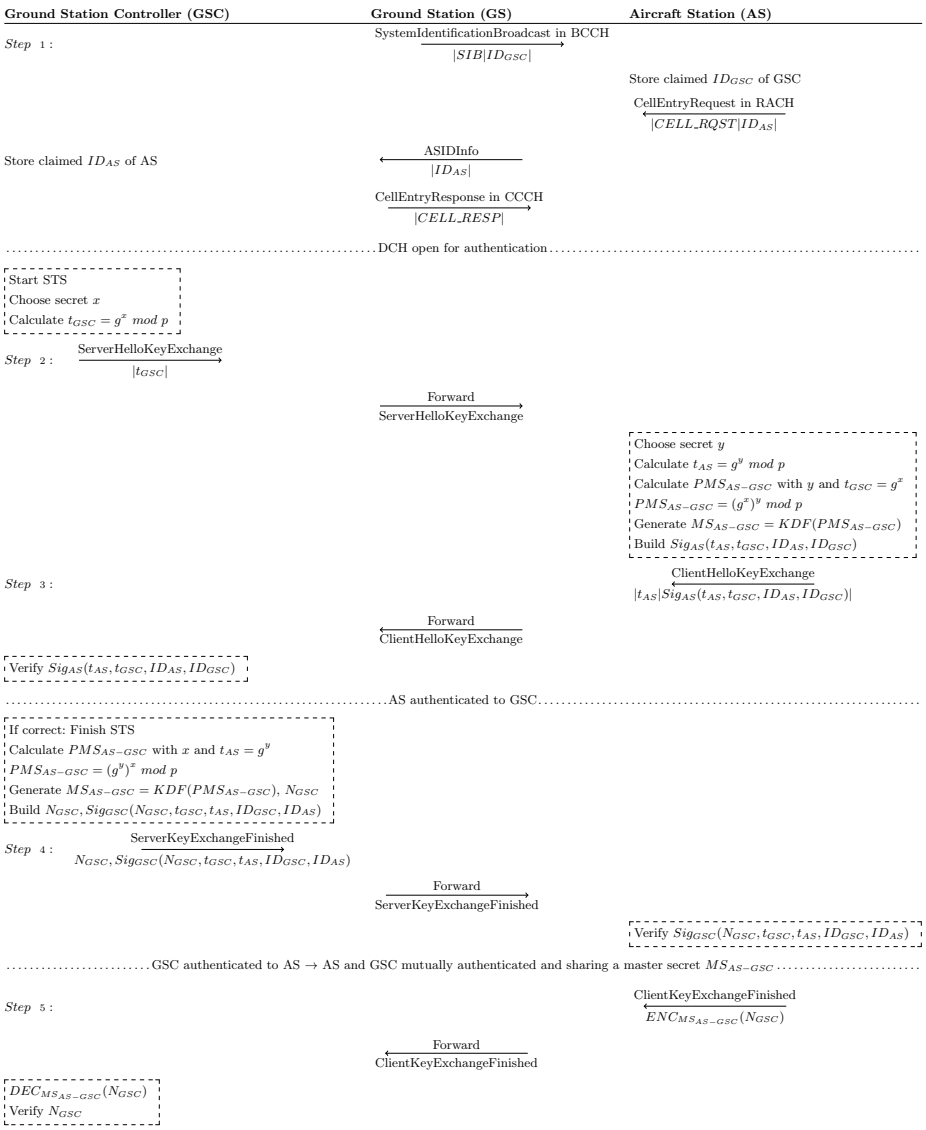

Fig. 2. LDACS MAKE Protocol. The protocol uses the Broadcast Control Channel (BCCH), Random Access Channel (RACH), Common Control Channel (CCCH), and Data Channel (DCH) logical channels defined in [19].

Secure Key Agreement: Both parties have established a shared session key, which means both parties know this key and know that they can use it for a secure communication with the other party for the duration of this session. The key must have never been used before in a session and only the two parties can know it.

Perfect Forward Secrecy: The established session key remains secret, even when the private signing keys of the involved parties have been compromised after this session.

4.1 LDACS MAKE Protocol

The LDACS MAKE protocol is illustrated in figure 2. It has five steps:

Step 1 – Cell Entry Procedure: First the AS connects to the LDACS network via the cell entry procedure described in section 4. During this procedure the AS is told a claimed identity of the network ID_{GSC} and the network is told a claimed identity of the AS, ID_{AS}. Once that procedure is done, the GSC chooses a secret x and calculates its public key $t_{GSC} = g^x$.

Step 2 – Server Hello Key Exchange: The GSC sends the $ServerHelloKeyExchange$ message to the AS. The AS chooses a secret y and calculates its public key t_{AS}. It then calculates the Pre-Master Secret (PMS) $PMS_{AS-GSC} = (g^x)^y \mod p$ and the Master Secret (MS) $MS_{AS-GSC} = KDF(PMS_{AS-GSC})$ via a predefined Key Derivation Function (KDF) and creates its own signature $Sig_{AS}(t_{AS}, t_{GSC}, ID_{AS}, ID_{GSC})$.

Step 3: – Client Hello Key Exchange: The AS sends now its public key t_{AS} and its signature to the GSC in the $ClientHelloKeyExchange$ message. The GSC verifies the AS signature $Sig_{AS}(t_{AS}, t_{GSC}, ID_{AS}, ID_{GSC})$. If that verification passes, at this point the AS is authenticated to the GSC. The GSC proceeds to generate $PMS_{AS-GSC} = (g^y)^y \mod p$ and $MS_{AS-GSC} = KDF(PMS_{AS-GSC})$ via a predefined KDF and builds another signature tag $Sig_{GSC}(t_{GSC}, t_{AS}, ID_{GSC}, ID_{AS})$.

Step 4: – Server Key Exchange Finished: The GSC sends its nonce N_{GSC} and signature $Sig_{GSC}(N_{GSC}, t_{GSC}, t_{AS}, ID_{GSC}, ID_{AS})$ in the $ServerKeyExchangeFinished$ message to the AS. There the AS verifies the GSC signature $Sig_{GSC}(N_{GSC}, t_{GSC}, t_{AS}, ID_{GSC}, ID_{AS})$. If that verification passes, at this point the GSC is authenticated to the AS

Step 5: – Client Key Exchange Finished: To attain key confirmation, the AS encrypts the nonce N_{GSC}, $ENC_{MS_{AS-GSC}}(N_{GSC})$ and sends that in the $ClientKeyExchangeFinished$ message to the GSC. At the GSC, the N_{GSC} nonce is decrypted and verified. If that verification step is successful, key confirmation of the key MS_{AS-GSC} is achieved.

4.2 Simplifications and Assumptions

It is foreseen that prior to any flight, AS, GS, GSC have entity-specific certificates and have agreed on common values for a Diffie-Hellman Key Exchange

variation (e.g. Diffie-Hellman Key Exchange (DHKE), Elliptic Curve Diffie-Hellman (ECDH), Supersingular Isogeny Diffie–Hellman (SIDH)). For simplification we therefore assume: (1) AS and GSC generate a fresh public key for each protocol run (t_{AS}, t_{GSC} serve as a nonce placeholder). (2) AS, GS and GSC have unique identifiers. (3) AS, GS and GSC have a signed certificate (handled by an LDACS PKI) proving their identity claim. (4) AS, GS and GSC have all necessary public keys of respective communication entities to verify signatures and identity claims. (5) AS, GS, GSC have agreed on necessary Diffie-Hellman parameters (e.g. p, g). (6) GS and GSC have established a secured communication channel prior to the LDACS MAKE procedure.

For our model checking process, we had to make some further assumptions about the building blocks in the LDACS MAKE procedure: We assume (1) a given PKI with a (2) trustworthy CA, which (3) binds uniquely identifiable station-IDs to public signing keys (e.g. certificates like X.509) of all possible parties (aircraft and ground stations). In our model, (4) stations can register their public keys autonomously, but we (5) limit the storage to one key per station-ID. Further, we assume (6) that an unauthorized key registration with a spoofed station-ID is not possible. We assume, (7) that the private signing keys are only known to their respective stations and are initially not known to the adversary. To verify the claimed forward secrecy, we model a key corruption after the protocol run. As we employ symbolic model checking, the modeled cryptosystems are treated as black boxes meaning that (8) signature and (9) encryption mechanisms are assumed secure as long as the appropriate keys are unknown to the adversary. Therefore, it is assumed, (10) that the adversary can not learn anything from messages he cannot decrypt. Also, (11) a guaranteed freshness is assumed for all generated values like session IDs or nonces. We do not model any physical properties of the communication - e.g. timing-issues of the message exchange - and therefore (12) we can not capture so called side channel attacks. A feature of Tamarin is (13) the unbound number of stations and therefore parallel or interleaved protocol runs by default, which allows to identify the most complex attacks on the protocol. This may not be possible in the real world due to limited resources like computing power and bandwidth restrictions.

4.3 Lemmata to be Proofed

In the property specification-part, we first formulated a lemma for sanity checking of our model. We wanted to verify that the protocol can execute from the creation of the stations until the successful key-exchange: **Lemma "Executable"**: There exists a trace where A in role AS gets created and B in role GSC gets created, A is requesting B for cell entry, both are starting the protocol by exchanging t_{AS} and t_{GSC}, and finally both commit by having the shared session key.

To prove that a secure key exchange is possible, meaning that both parties have a secret shared key after a protocol run, we added another lemma to prove

that: **Lemma "Secure Key Exchange"**: If A finishes a run with B, it can be sure, that it has a fresh key P and that B also has this key for use with A (mutual understanding), and this key has not been exchanged before implicating that also no other agent knows it. The only exclusion is when the private key of an honest agent has been corrupted before.

Then we have a lemma to prove the perfect forward secrecy of the exchanged session key and hence all secrets which will be encrypted with it. We used the recommended secrecy-lemma from the Tamarin documentation for this: **Lemma "Perfect Forward Secrecy"**: The exchanged session key (MS) can not be known by the attacker, even when he acquires the private key of one or both parties later on.

Finally, we have one lemma to prove the mutual authentication of both parties by an injective agreement over exchanged messages (t_{AS} and t_{GSC}), which can be seen as the strongest definition of authentication according to Lowe [25]. Again, this lemma has been formulated very close to the recommendations and examples from the Tamarin documentation [1]: **Lemma "Mutual Authentication via Injective Agreement"**: If A finishes a run with B by exchanging y, it can be sure, B also ran the protocol with A and y has not been exchanged before in any other run. The only exclusion is when the private key of an honest agent has been compromised before.

5 Results

We ran the Tamarin prover version 1.4.1 in automatic mode to prove the 4 lemmata we presented above. The verification took 20,158s on a Ubuntu 18.04 Laptop with an Intel(R) Core(TM) i7-8650U CPU and 16GB of RAM. All 4 lemmata could be verified without interaction. The source code of the Tamarin model is available for download at GitHub[3].

In Table 1 we present the Tamarin output for each lemma. The scope column states which type of proof has been done: 'exists-trace'-proofs verify, that the given property or lemma holds at least for one trace of the protocol; 'all-traces'-proofs respectively verify that the property holds for all protocol traces. The last column gives the number of verification steps that were executed by Tamarin to verify the appropriate lemma. As can be seen, the "Secure Key Exchange"-lemma took substantially more steps than the others to be verified.

In Figure 3 we present the visualization of the proof of the Executable-lemma. It shows a complete trace of the protocol run with both stations AS and GSC engaged. It can be seen, that all foreseen messages are being sent and that both stations commit with the same session key.

[3] https://github.com/RohanKrishnamurthy/L-DACS

Table 1. Verification results for the LDACS MAKE protocol

Lemma	Scope	Result	Steps
Executable	exists-trace	verified	21
Secure Key Exchange	all-traces	verified	615
Perfect Forward Secrecy	all-traces	verified	19
Mutual Authentication	all-traces	verified	29

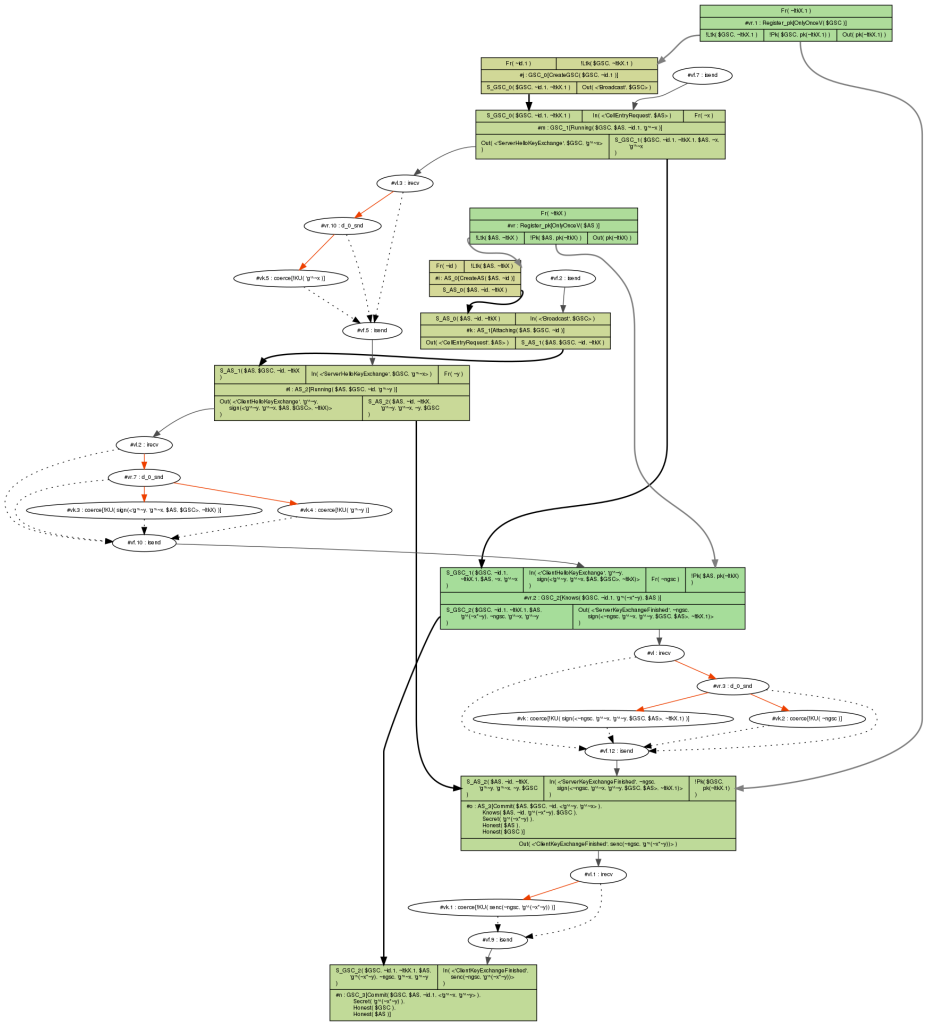

Fig. 3. Tamarin proof visualization of a successful LDACS MAKE protocol run

6 Discussion

Our results show that the LDACS MAKE protocol – which is a modified Station to Station (STS) protocol – is secure in fulfilling (1) mutual authentication of communication parties, (2) secure key agreement among communication parties and (3) perfect forward secrecy for subsequent key material.

6.1 Limitations of the Proof

However, our proof is valid only under the assumptions of Section 4.2. For real-world applications, these assumptions do not necessarily hold in general. Apart from the limitations of the assumptions themself, there are limitations originating from the method of symbolic model checking, too.

Limitations of Symbolic Modeling and Analysis: As indicated in Section 4.2 there is no guarantee for computational soundness of the protocol since we can not make assertions about the used cryptographic algorithms. For example, the used Diffie-Hellman exponentiation is generally accepted as "secure" due to the difficulty of factoring big numbers. In the symbolic model, we can not verify or falsify this so-called "Diffie-Hellman-assumption" - we assume it is right. The same holds true for the other used cryptographic primitives as signatures or encryption schemes. These (e.g. AES, RSA) are not specified in our model, but modeled as symbolic functions transforming data in an unspecified way. Also, because of the symbolic level we are operating, we can not make any assertions regarding possible vulnerabilities of an actual software implementation of the protocol, e.g. because of buffer overflows. These kinds of flaws are out of scope of this work. Also, side-channel attacks because of e.g. timing issues can not be found in this way, since they are also implementation-specific.

Implications of Assumptions: In Section 4.2 we mentioned 19 assumptions, which don't necessarily hold true in real-world implementations. Many of the mentioned assumptions rely on the secure setup and operation of the underlying LDACS PKI. For instance, the assumed impossibility of unauthorized key registration with a spoofed ID, the privacy of private signing keys, storage limits and an autonomous registration of station public keys are all dependent on the way the PKI is set up and how keys are transported. In addition we assumed that the LDACS PKI would be similar to the AeroMACS PKI [15]. As this is a requirement for the proposed LDACS MAKE procedure to work as intended, due to the corroboration of identity and public keys and trust established within the chain-of-trust model that the PKI uses, it is reasonable to assume such a PKI to be built for LDACS. However, already the second assumption, that of a trustworthy Certificate Authority (CA) proves difficult, as there are many examples, where intermediate sub-CAs were compromised in the past [36]. As there is no such incident reported for the AeroMACS PKI as of yet and with ICAO's efforts to establish their own aviation PKI [34], the underlying PKI infrastructure will likely become more robust in the future. The unique bind of an ID and public signing key can be broken in principle [11], thus there are additional measures necessary to enforce this assumption, such as regular system and chain-of-trust

integrity checks, secure logging procedures and more. Finally, the overall implementation of security functions such as signature mechanisms or encryption algorithms is not guaranteed to be correct and work as intended [2].

6.2 Recommendations for LDACS the MAKE Protocol

We derived several recommendations for the further improvement of the LDACS MAKE protocol from our results.

Replay attacks: There are several strategies to further protect against replay attacks such as authenticated ranges, so that the LDACS AS and GS radio measure the properties of the radio channel and estimate the origin of the signal source. Thus with all GS positions known to all AS, rogue GS can be easier detected by AS. Furthermore the time between AS requests and GS responses can be measured up until a certain response $\triangle$. With this margin of error, all responses taking longer than the lower estimated boundary might be messages that were intercepted and replayed by a possible attacker. Another way to rule out attacks is to use the frequency plan, including all frequencies for Forward Link (FL) and Reverse Link (RL) and the respective expected Equivalent Isotropically Radiated Power (EIRP) to allocate all information together: frequencies, IDs, EIRPs at certain times, key material and so forth. With this additional information attacking the radio link becomes increasingly difficult.

Denial of service attacks: One idea to strengthen the LDACS link against Denial of Service (DoS) attacks is to include server cookies in the SIB that are to be returned by honest ASs. As there are three Broadcast (BC) slots in the Broadcast Control Channel (BCCH) channel of LDACS, up to 1000 Bits of additional information can still be fit into these slots.

Age of security material: One possibility to shorten the lifetime of overall used cryptographic material on the moving nodes, thus the aircraft, is to hand out smartcards alongside the flight plan with pre-stored cryptographic material for every flight. Thus the LDACS radio itself does not possess any cryptographic keys but they are applied for each flight only and revoked after the flight has been successfully completed. One way to approach this is the use of Identity Based Signature (IBS) [35], where keys can be revoked mathematically after a certain timeframe and the public keys are bound by default to a respective entity.

7 Conclusion

In this paper, we investigated the LDACS Mutual Authentication and Key Exchange (MAKE) procedure for establishing a secure LDACS communication link between aircraft and ground station.

The contribution of this paper is the formal proof of the security of this protocol. Using the symbolic model checker Tamarin, we built a mathematical, formal model of the LDACS MAKE procedure and following several security

objectives that this procedure must fulfill, we derived several provable lemmata for Tamarin. Tamarin finally proved that the LDACS MAKE procedure is secure in the standard model and is proven to have no design flaws in its architecture. This constitutes an important step for the development of the general LDACS cybersecurity architecture since authentication and key establishment are the most crucial steps in establishing secure wireless communication.

Discussing our security proof, we elaborated the limitations of symbolic model checking and their consequences for a real world implementation for the LDACS MAKE procedure. We finally suggested several security improvements for a real world application of the proposed LDACS MAKE protocol, such as the estimation of radio signal origins of AS and GS radio signals to harden against possible rogue senders.

In future research, we will investigate the agreement and exchange of Diffie-Hellman parameters, as well as control channel security for the control channels of LDACS.

AeroMACS	Aeronautical Mobile Airport Communication System
AS	Aircraft Station
BC	Broadcast
BCCH	Broadcast Control Channel
CA	Certificate Authority
CCCH	Common Control Channel
CNS	Communication, Navigation and Surveillance
DCH	Data Channel
DHKE	Diffie-Hellman Key Exchange
DoS	Denial of Service
ECDH	Elliptic Curve Diffie-Hellman
EIRP	Equivalent Isotropically Radiated Power
FCI	Future Communications Infrastructure
FL	Forward Link
GS	Ground Station
GSC	Ground Station Controller
IATA	International Air Traffic Association
IBS	Identity Based Signature
ICAO	International Civil Aviation Organization
KDF	Key Derivation Function
LDACS	L-band Digital Aeronautical Communication System
MAKE	Mutual Authentication and Key Exchange
MS	Master Secret
PKI	Public Key Infrastructure
PMS	Pre-Master Secret
RACH	Random Access Channel
RL	Reverse Link
SAC	Subscriber Access Code
SARPS	Standards and Recommended Practises
SESAR	Single European Sky ATM Research

SIB	System Identification Broadcast
SIDH	Supersingular Isogeny Diffie–Hellman
STS	Station to Station
VHDL	Very High Speed Integrated Circuit Hardware Description Language
WiMAX	Worldwide Interoperability for Microwave Access

References

1. Tamarin Prover Manual (2020 (accessed Jul 28,2020)), https://tamarin-prover.github.io/manual/index.html
2. Aumasson, J.P.: Serious cryptography: a practical introduction to modern encryption. No Starch Press (2017)
3. Basin, D., Cremers, C., Kim, T.H., Perrig, A., Sasse, R., Szalachowski, P.: Design, analysis, and implementation of arpki: An attack-resilient public-key infrastructure. IEEE Transactions on Dependable and Secure Computing **15**(3), 393–408 (2018)
4. Basin, D., Cremers, C., Meadow, C.: Model Checking Security Protocols, chap. 4, p. 100. Springer (2015), http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.708.9043&rep=rep1&type=pdf
5. Bilzhause, A., Belgacem, B., Mostafa, M., Gräupl, T.: Datalink Security in the L-band Digital Aeronautical Communications System (LDACS) for Air Traffic Management. Aerospace and Electronic Systems Magazine **32**(11), 22–33 (November 2017)
6. Blanchet, B.: An Efficient Cryptographic Protocol Verifier Based on Prolog Rules. In: 14th IEEE Computer Security Foundations Workshop (CSFW-14). pp. 82–96. IEEE Computer Society, Cape Breton, Nova Scotia, Canada (Jun 2001)
7. Boyd, C., Mathuria, A., Stebila, D.: Protocols for Authentication and Key Establishment. Springer (2020)
8. Cimatti, A., Clarke, E., Giunchiglia, F., Roveri, M.: NUSMV: A New Symbolic Model Verifier. pp. 495–499 (01 1999)
9. Clarke: Handbook of model checking, vol. 31. Springer International Publishing AG (2018). https://doi.org/10.1007/s00165-019-00486-z, https://doi.org/10.1007/s00165-019-00486-z
10. Clarke, E.M., Emerson, E.A., Sistla, A.P.: Automatic Verification of Finite State Concurrent System Using Temporal Logic Specifications: A Practical Approach. In: Proceedings of the 10th ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages. p. 117–126. POPL '83, Association for Computing Machinery, New York, NY, USA (1983). https://doi.org/10.1145/567067.567080, https://doi.org/10.1145/567067.567080
11. Cohn-Gordon, K., Cremers, C., Garratt, L.: On post-compromise security. In: 2016 IEEE 29th Computer Security Foundations Symposium (CSF). pp. 164–178. IEEE (2016)
12. Costin, A., Francillon, A.: Ghost in the Air(Traffic): On Insecurity of ADS-B protocol and Practical Attacks onADS-B Devices. Black Hat USA pp. 1–10 (August 2012)
13. Cremers, C.: Security protocol analysis tools (2020 (accessed July 08, 2020)), https://people.cispa.io/cas.cremers/tools/index.html

14. Cremers, C.: The Scyther Tool: Verification, Falsification, and Analysis of Security Protocols. In: Computer Aided Verification, 20th International Conference, CAV 2008, Princeton, USA, Proc. Lecture Notes in Computer Science, vol. 5123/2008, pp. 414–418. Springer (2008). https://doi.org/10.1007/978-3-540-70545-1_38
15. Crowe, B.: Proposed AeroMACS PKI Specification is a Model for Global and National Aeronautical PKI Deployments. In: WiMAX Forum at 16th Integrated Communications, Navigation and Surveillance Conference (ICNS). pp. 1–19. IEEE, New York, NY, USA (April 2016)
16. Dolev, D., Yao, A.C.: On the Security of Public Key Protocols (Extended Abstract). In: 22nd Annual Symposium on Foundations of Computer Science, Nashville, Tennessee, USA, 28-30 October 1981. pp. 350–357. IEEE Computer Society (1981). https://doi.org/10.1109/SFCS.1981.32, https://doi.org/10.1109/SFCS.1981.32
17. EUROCONTROL Statistics and Forecast Service: European Aviation in 2040 - Challenges of Growth. Tech. Rep. 2, EUROCONTROL, Brussels, Belgium (2018 (accessed July 5, 2019)), https://www.eurocontrol.int/sites/default/files/content/documents/official-documents/reports/challenges-of-growth-2018.pdf
18. Giraudon, N., Iannes, M., Tamalet, S., Lehmann, M., Ben Mahmoud, S., Larrieu, N., Correas, A., Fasetta, S.: Part 1 - AeroMACS Safety and Security Analysis, Part 2 - AeroMACS Security Analysis (December 2014 (accessed July 9, 2019)), https://www.icao.int/safety/acp/ACPWGF/ACP-WG-S-5/IP09%20-%20SESAR%20AeroMACS%20Safety%20and%20Security%20Analysis.pdf
19. Gräupl, T., Rihacek, C., Haindl, B.: LDACS A/G Specification. Sesar2020 pj14-02-01 d3.3.030, German Aerospace Center (DLR), Oberpfaffenhofen, Germany (August 2019)
20. Hall, A., Wingfield, J., De Moura, G., Tiscareno, K.: Advancing Cyber Resilience in Aviation: An Industry Analysis. World Economic Forum pp. 1–28 (2020)
21. Holzmann, G.: The Design and Validation of Computer Protocols (01 1991)
22. IATA: IATA Forecast Predicts 8.2 billion Air Travelers in 2037 (October 2020 (accessed Jun 20, 2020)), https://www.iata.org/en/pressroom/pr/2018-10-24-02/
23. ICAO: Effects of Novel Coronavirus (COVID-19) on Civil Aviation:Economic Impact Analysis. https://www.icao.int/sustainability/Documents/COVID-19/ICAO_Coronavirus_Econ_Impact.pdf (May 2020 (accessed May 13, 2020))
24. (ICAO), I.C.A.O.: Finalization of LDACS Draft SARPs - Working Paper WP05 including Appendix. Tech. rep., ICAO, Montreal, Canada (October 2018)
25. Lowe, G.: A Hierarchy of Authentication Specification. In: 10th Computer Security Foundations Workshop (CSFW '97), June 10-12, 1997, Rockport, Massachusetts, USA. pp. 31–44 (1997). https://doi.org/10.1109/CSFW.1997.596782, https://doi.org/10.1109/CSFW.1997.596782
26. Maggi, P., Sisto, R.: Using SPIN to Verify Security Properties of Cryptographic Protocols. In: Bosnacki, D., Leue, S. (eds.) Model Checking of Software, 9th International SPIN Workshop, Grenoble, France, April 11-13, 2002, Proceedings. Lecture Notes in Computer Science, vol. 2318, pp. 187–204. Springer (2002). https://doi.org/10.1007/3-540-46017-9_14, https://doi.org/10.1007/3-540-46017-9_14
27. Mäurer, N., Bilzhause, A.: Paving the Way for an IT Security Architecture for LDACS: A Datalink Security Threat and Risk Analysis. In: 18th Integrated Communications, Navigation and Surveillance Conference (ICNS). pp. 1A2/1–1A2–11. IEEE, New York, NY, USA (April 2018)

28. Mäurer, N., Schmitt, C.: Towards Successful Realization of the LDACS Cybersecurity Architecture: An Updated Datalink Security Threat- and Risk Analysis. In: 19th Integrated Communications, Navigation and Surveillance Conference (ICNS). pp. 1A2/1–1A2–13. IEEE, New York, NY, USA (April 2019)
29. Mäurer, N., Gräupl, T., Schmitt, C.: Evaluation of the ldacs cybersecurity implementation. In: 38th Digital Avionics Systems Conference (DASC). pp. 1–10. IEEE (September 2019)
30. Mäurer, N. and Bilzhause, A.: A Cybersecurity Architecture for the L-band Digital Aeronautical Communications System (LDACS). In: 37th Digital Avionics Systems Conference (DASC). pp. 1–10. IEEE, New York, NY, USA (September 2018)
31. Mäurer, N., Gräupl, T. and Schmitt, C.: Comparing Different Diffie-Hellman Key Exchange Flavors for LDACS. In: 39th Digital Avionics Systems Conference (DASC). pp. 1–10. IEEE, New York, NY, USA (October 2020)
32. Meier, S., Schmidt, B., Cremers, C., Basin, D.: The tamarin prover for the symbolic analysis of security protocols. In: Proceedings of the 25th International Conference on Computer Aided Verification - Volume 8044. p. 696–701. CAV 2013, Springer-Verlag, Berlin, Heidelberg (2013)
33. Niraula, M., Graefe, J., Dlouhy, R., Layton, M., Stevenson, M.: ATN/IPS Security Approach: Two-way Mutual Authentication, Data Integrity and Privacy. In: Integrated Communications, Navigation, Surveillance Conference. pp. 1–17. IEEE (2018)
34. Patel, V.: ICAO air-ground security standards status ICNS conference 2016. In: 2016 Integrated Communications Navigation and Surveillance (ICNS). pp. 1–31. IEEE (2016)
35. Paterson, K.G., Schuldt, J.C.: Efficient identity-based signatures secure in the standard model. In: Australasian Conference on Information Security and Privacy. pp. 207–222. Springer (2006)
36. Roosa, S.B., Schultze, S.: Trust darknet: Control and compromise in the internet's certificate authority model. IEEE Internet Computing **17**(3), 18–25 (2013)
37. Schmidt, B., Sasse, R., Cremers, C., Basin, D.: Automated verification of group key agreement protocols. In: 2014 IEEE Symposium on Security and Privacy. pp. 179–194 (2014)
38. Schmidt, B., Meier, S., Cremers, C.J.F., Basin, D.A.: Automated Analysis of Diffie-Hellman Protocols and Advanced Security Properties. In: Chong, S. (ed.) 25th IEEE Computer Security Foundations Symposium, CSF 2012, Cambridge, MA, USA, June 25-27, 2012. pp. 78–94. IEEE Computer Society (2012). https://doi.org/10.1109/CSF.2012.25, `https://doi.org/10.1109/CSF.2012.25`
39. Schnell, M.: Update on LDACS - The FCI Terrestrial Data Link. In: 19th Integrated Communications, Navigation and Surveillance Conference (ICNS). pp. 1–10. IEEE, New York, NY, USA (April 2019)
40. Shinde, A.H., Umbarkar, A., Pillai, N.: Cryptographic protocols specification and verification - a survey. ICTACT Journal on Communication Technology **8**(2) (2017)
41. Slim, M., Mahmoud, B., Pirovano, A., Larrieu, N.: Aeronautical Communication Transition From Analog to Digital Data: A Network Security Survey. Computer Science Review **11-12**, 1–29 (May 2014)
42. Standar, M.: Next generation of CNS services and the enabling infrastructure "Anything that can be connected, will be connected". In: 18th Integrated Communications, Navigation and Surveillance Conference (ICNS). pp. 1–9. IEEE, New York, NY, USA (April 2018)

43. Thomas Boegl (Rohde & Schwarz), Mathias Rautenberg (Rohde & Schwarz), Bernhard Haindl (FrequentisAG), Christoph Rihacek (Frequentis AG), Josef Meser (Frequentis AG), Pierluigi Fantappie (Leonardo), Noppadol Pringvanich (IATA), John Micallef (SITA), Klauspeter Hauf (DFS), John MacBride (UK NATS), Philippe Sacre (Eurocontrol), Bart van den Einden (Eurocontrol), Thomas Gräupl (DLR), and Michael Schnell (DLR): LDACS White Paper–A Roll-out Scenario. Tech. rep., ICAO, Montreal, Canada (October 2019 (accessed June 6, 2020)), https://www.ldacs.com/wp-content/uploads/2013/12/ACP-DCIWG-IP01-LDACS-White-Paper.pdf
44. Yang, H., Zhou, Q., Yao, M., Lu, R., Li, H., Zhang, X.: A Practical and Compatible Cryptographic Solution to ADS-B Security. IEEE Internet of Things Journal **6**(2), 3322–3334 (2018)
45. Zelkin, Natalie and Henriksen, Stephen: L-band digital aeronautical communications system engineering-initial safety and security risk assessment and mitigation. Nasa/cr—2011-216327 saa3–978–1, NASA, Herndon, Virginia, USA (January 2011)