On Requirements & Concepts for TT&C Link Key Management

Christoph Bader
Airbus Defence & Space GmbH
christoph.bader@airbus.com

***Abstract*—Recent reports on the state of satellite security reveal that many satellite systems that are operational today do not implement sufficient protection against cyber-attacks. Most notably is the fact that many systems lack of cryptographic protection on their TT&C link. If COMSEC protection on the TT&C link is missing an attacker with access to the RF link can eavesdrop on the communication and, even worse, could be able to inject specially crafted messages that would be processed by the satellite.**

In this paper, we analyze needs and establish high level requirements for concepts aiming to secure TT&C link communication (with respect to confidentiality and authentication). The requirements cover key aspects of security and operations. We assess existing standards (SDLS and SDLS EP) against our requirements and determine that SDLS is suitable for traffic protection while SDLS EP does not meet all security requirements for key management (namely, it does not meet post compromise security). Finally, we discuss alternative key management approaches such as stateless authenticated key agreement and stateful authenticated key agreement (or key evolution protocols) and how they meet our requirements.

I. Introduction

After a long period in which security of satellite systems was not (publicly) studied, the topic has gained more and more attention recently [14]–[16], [20], [23], [25], [27]–[29]. In the past many satellite systems were ”protected” by obscurity and the belief that an attacker cannot access the RF link. In consequence, many operational satellites today do not implement cryptographic protection on the TT&C link [30]. It is consensus that COMSEC protection on the RF link is key for the protection of satellite systems [22], [23], [25], [29].

The issue of missing cryptographic protection on the TT&C link was recognized more than ten years ago by the *Consultative Committee for Space Data Systems* (CCSDS), a standardization body for space systems. To address the issue, CCSDS established the Space Data Link Security standard (SDLS) in 2015 [10] and, more recently, its extended procedures (SDLS EP) [11]. Since developing a satellite system takes a couple of years and the operational lifetime of a satellite system can go beyond 10 years, it cannot be expected that SDLS and/or SDLS EP are implemented in many operational satellite systems today as was found out (implicitly) by Willbold et al. [30].

While SDLS defines formats and procedures for traffic protection on the RF link, SDLS EP specifies procedures and data units to manage the onboard function that implements COMSEC protection (including the procedures defined by SDLS). When applied together, SDLS and SDLS EP provide a crypto and key management concept for satellite systems, covering crypto mechanisms for traffic protection but also management of keys - two aspects that go hand in hand for satellite systems. This is because TT&C link protection requires a property we denote as *all frame protection* (or all frame decryption). We define this property as the capability to be able to protect or decrypt all data units at any time throughout the mission (data units in the context of TT&C links are called *frames*, a TC frame has a maximum size of 1024 octets, cf. Section I-B); this applies in particular to the first data unit at the beginning of a contact. For this to work, the key management concept and the protocol for traffic protection need to be well coordinated.

SDLS and SDLS EP achieve all frame decryption by building only on symmetric cryptography. This solution requires little complexity in a closed satellite control system but comes along with some drawbacks with respect to security. In particular, after keys have been compromised it is not possible to reach a secure state again.

In this paper, we therefore address the following question related to COMSEC protection of TT&C links: 1) what are the key requirements needed from a crypto and key management concept for TT&C link protection; 2) how do SDLS and SDLS EP meet these requirements and 3) are other approaches available for key management of satellite systems.

A. Contribution

Our contribution is threefold:

1) We analyze and establish high level requirements for crypto/key management concepts for COMSEC protection of TT&C links. Here, we address different aspects such as requirements for traffic protection or key management but also take into account operational constraints; the key requirements we establish are ”all frame protection” and post-compromise security (detailed in Section II).
2) We analyze the ”compliance” of the SDLS protocol against the requirements that we have established for

Workshop on Security of Space and Satellite Systems (SpaceSec) 2024
1 March 2024, San Diego, CA, USA
ISBN 979-8-9894372-1-4
https://dx.doi.org/10.14722/spacesec.2024.23053
www.ndss-symposium.org

traffic protection and find out that SDLS meets them (if configured appropriately).

3) We assess several options for key management with focus on particular key update or key renewal: a) the key management approach by SDLS EP, b) key management based on stateless authenticated key agreement and c) key management based on stateful key agreement or key evolution protocols, like ratcheted key agreement. While stateless AKE protocols with strong security properties have been long studied in the crypto community [2]–[4], [9], [18], [21], ratcheted key exchange is a comparably new crypto primitive that was developed for mobile messaging in the early 2010s and was since then studied intensively [1], [8], [17], [26]. We find out that ratcheted key exchange (or stateful key agreement in general) is a very promising candidate for TT&C link key management.

a) Scope of this paper: The scope of this paper is on COMSEC protection of the TT&C link between the ground and a single satellite or a small set of satellites. Protection of payload management and of communication with the user segment are out of scope as these are more mission specific.

B. Background

A satellite usually is built from a generic *platform* and a mission specific *paylaod*. Both, platform and payload, are made of different hardware components, each implementing specific functions. The satellite platform comprises those components that are usually not mission specific, e.g., components to establish an RF link connection to ground for control of the satellite, or power and orbit control. The payload implements mission specific functions.

A satellite is operated by its ground segment via an RF link connection. The RF link that is used to control the satellite (platform) is called TT&C (telemetry, tracking and command) link. A data unit transmitted over the TT&C link is called frame. A frame is addressed to a single satellite and may contain one or more packets addressing onboard components or functions. After reception onboard, the packets are sent to the respective recipient. Frames that are sent to the satellite are called TC frames. The generic format of a TC frame is depicted in Figure 1. Frames sent from the satellite to ground are called TM frames.

A frame is an OSI layer 2 data unit. It is noted that in satellite systems, the layer 2 is split into two sub-layers which are the data link protocol sublayer and the synchronization and channel coding sublayer [12]. The data link protocol sublayer is usually extended from the ground station (i.e., the antenna that provides RF link connection to the satellite) to the ground segment. I.e., the frames are not (dis-) assembled by the groundstation but by the ground segment.

The ground segment can communicate with the satellite (via the ground station) only while it is in sight of a ground station. This is called ground contact or contact. Except for geo-stationary satellites, satellites usually do not have permanent contact with a ground station on their TT&C link.

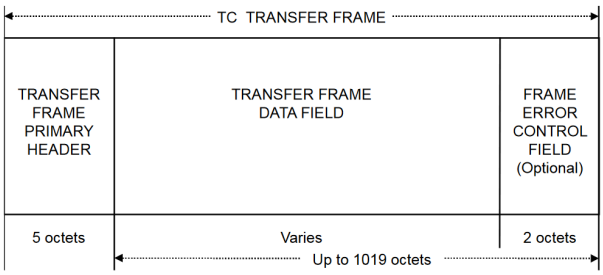

Fig. 1. Generic TC Transfer Frame format. The (max.) length of the data field is determined by a (10 bit) length field in the frame header defining the length of the actual frame (bounded by 1024 octets). Figure taken from [12].

Therefore, contact time on the TT&C link is expensive. This is complemented by the fact that - depending on the orbit - data rates can be very low (in the order of kilobits/second).

Once a contact and a communication link is established, the following high level functions are implemented by the satellite for TC frame reception: demodulation, decoding of data units and decryption/authentication (if applicable). Only after these steps have been performed, the data units are provided to the onboard data handling system. In TM direction, the onboard data handling system selects data to be sent through the downlink; the following steps are implemented on the downlink path: encryption/authentication, encoding and modulation.

In the remainder of this paper we denote by *protection function* (or security function) the onboard unit, module or element that implements TC/TM frame protection. I.e., in TC direction, it receives as input demodulated and decoded data and outputs it to the onboard data handling system after processing. In TM direction, the protection function receives TM frames from the onboard data handling system and protects them before forwarding to channel encoding and modulation.

II. What is required - and what is not

In this section we establish high level requirements that are needed from a crypto concept aiming to provide COMSEC protection on TT&C links. We also discuss properties that are not needed for TT&C link protection (as these distinguish TT&C link protection from other real world use cases of cryptographic mechanisms).

A. What is required

1) All Frames Protection: For COMSEC protection on TT&C links, we require that the traffic is protected with respect to confidentiality and authenticity and is protected against replay. I.e., an attacker that is able to eavesdrop on frames, or drop, alter or relay frames, or even inject crafted frames in the TT&C link communication shall not be able break the confidentiality or authenticity of the communication.

As described above, contact time is expensive. It is therefore required to be able to communicate with a satellite at the beginning of each contact without the need to ”set up” the protection function beforehand in the same contact. The

onboard security function is required to be able to decrypt and authenticate each frame immediately. This applies in particular to the first frame in every contact or even the first frame after launch. This is key in order to be able to establish a connection with the satellite and get TM quickly, e.g., in case of an unplanned contact or in an emergency situation. This means that the protection function needs to know which key(s) to apply for authentication/decryption of the first frame or - more general - which crypto parameters to apply.

2) Long-term Security: The time from design of a satellite to end of mission can easily exceed 20 years with an operational lifetime of more than 10 years. For reasons of robustness, the cryptographic mechanisms protecting the TT&C link are often implemented directly in hardware[1] (ASIC or FPGA) and are therefore hardly updateable; though FPGAs exist that support to update the firmware in space [24], [31], it is risky to carry out an update in space and - in case a failure happens during the update process - recovery can be difficult. ASICs are not updateable at all. For this reason, it is needed to implement cryptographic mechanisms in satellite systems that are considered to be secure until at least end of mission. Due to the long operational lifetime of a satellite system this means:

- Security against Quantum Attackers: It is hard to estimate the progress on quantum computing within the next 20 years; therefore, taking a conservative approach, security against quantum attackers is required. This means on the one hand side that the cryptographic mechanisms providing confidentiality and authenticity need to resist against quantum attackers. On the other hand side this also means that the mechanisms to be used for key upload or key establishment are required to provide protection against quantum attackers.
- Post Compromise Security: It is hard to keep keys secret for 20 years. Attacks on the ground segment are considered a major threat for satellite systems [25], [30] and it has to be assumed that during the lifetime of a satellite, its ground segment will be attacked. A compromise of the cryptographic keys cannot be excluded. Therefore, satellite systems are required to re-establish a secure state after keys have been compromised. Following [13], we define two notions of post compromise security which are *weak* and *strong* post compromise security.

  Strong Post Compromise Security: After an attacker learns all keys and crypto parameters currently used by the system (short and long-term)[2], it shall be possible to exclude the attacker from the communication at some point in the future with respect to both, confidentiality and authenticity. I.e., for strong post compromise security it is required that, after compromise (where the attacker learns all keys), it shall be possible within a finite time interval to establish new keys that provide sufficient entropy and cannot be distinguished from random keys - even by the attacker that compromised the system. We note that after compromise, an attacker could take over control of the satellite and, by that, try to prevent re-establishing a secure state. Once compromises are "allowed", this scenario cannot be prevented in general. In order to re-establish a secure state, the attacker has to be passive for some *finite* interval. This notion of post compromise security follows [5], [13] and it seems reasonable to make this assumption (that the attacker is passive for some time) for satellite systems as ad hoc communication to the satellite is not possible in general.

  Weak Post Compromise Security: A weaker notion, weak post compromise security, is discussed in [13]: weak post compromise security allows the adversary to temporarily access the long term key of a party; a secure protocol in this notion allows to establish a secure state after such access.

  For the purpose of this paper, we focus on compromise of the ground segment only; i.e., we do not consider a compromise of the onboard protection function and consider the keys of the onboard protection function to be secret over the mission lifetime (if not revealed through compromise on ground).
- As a past command may reveal sensitive information, we also require past sessions to be secure in case of compromise. This means that an attacker shall not be able to learn the content of "past" messages after compromise. This is a well established property for key negotiation schemes and called forward secrecy [9].

3) Summary: We summarize the requirements established above:

1) **All Frames Protection:** The protection function can protect any frame (and in particular, the first frame of a contact) without necessarily requiring additional set up during operations. Setup of the protection function prior to launch is accepted as we will argue below.
2) **Security against quantum attackers:** Crypto mechanisms need to provide protection of confidentiality and authenticity in the long-term; in particular, protection against quantum attackers is required.
3) **Post Compromise Security:** After a compromise of all cryptographic parameters and keys, it shall be possible to re-establish a secure state within a finite time interval.
4) **Forward Secrecy:** Past sessions are required to be secure in case of compromise.

We observe that these requirements address three aspects which are a) traffic protection (requirements 1, 2), b) key management (requirements 2, 3, 4) and c) a proper way to operate both together (requirement 1).

We discuss these aspects in Sections III and IV. Before we come to that, we briefly discuss properties that are desired but also properties that are not necessarily required for COMSEC protection on TT&C links.

[1] I.e., they are not implemented in SW running on a softcore on the FPGA, but as logical circuit in the FPGA.

[2] It is noted that at this point in time the attacker is able to eavesdrop on all communication

B. What is desired

The above list defines ”hard” requirements that a concept for traffic protection and key management needs to meet. In this section we elaborate on ”soft” requirements. These are properties that are desirable but are of less importance (from security point of view) compared to the requirements established above.

Low overhead for security management: The operational lifetime of a satellite can go well beyond 10 years. From security point of view, it is not desired to use only a single key set to protect the TT&C link for a mission lifetime of more than 10 years. This is complemented by the fact that memory and other hardware components degrade due to the harsh space environment. It follows that remote management of the onboard protection function is needed to handle multiple key sets (e.g., in order to check integrity of existing keys before selecting a new key set for operational use, or in order to establish a new key set)[3].

Following the argument above that contact time is expensive, it is desired from operational point of view that management of the protection function does not occupy much of the available bandwidth / contact time.

C. What is not necessarily needed

We briefly discuss requirements that are often considered as useful in cryptographic applications but that are not necessarily needed for COMSEC protection of TT&C links:

1) Open Systems: Satellite systems are usually closed systems (at least for what concerns platform control). For every mission, there is one (or more) ground segment(s) that control one or more satellite(s). Secure communication is required only in this closed system. The crypto mechanisms for TT&C link key management are therefore not required to work in open systems.

2) Statelessness: It is noted that usually in satellite systems the ground segment keeps track of the state of all relevant onboard parameters and variables and the satellite frequently reports its state via telemetry, e.g., temperature or power consumption of (critical) units. This has proven to be a good choice from operations point of view in the past. We therefore accept statefulness also for the cryptographic mechanisms for COMSEC protection of TT&C links.

III. Traffic Protection

The Space Data Link Security (SDLS) protocol that was standardized by CCSDS [10] specifies formats and procedures for data unit protection on frame level, i.e., for traffic protection on the TT&C link. As can be seen in Figure 2, SDLS defines a *Security Header*, containing, e.g. a replay counter or an IV to be used for encryption, and a *Security Trailer* carrying a MAC over (selected parts of) the frame. The definition and length of these fields is mission specific. However, SDLS

[3]The ”key check” serves as example to demonstrate the need for management of the onboard security function. Many more ”managed parameters” could be implemented.

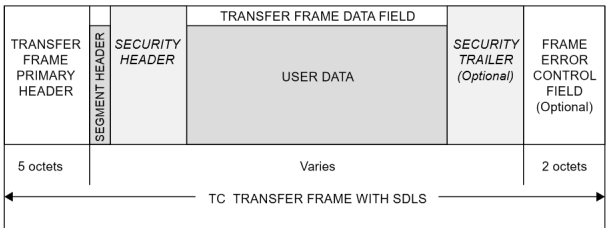

Fig. 2. Generic format of protected TC Transfer Frame. The frame now has a security header and a security trailer. Figure taken from [10].

generically defines where to put them in a frame and how to use them (conceptually).

Let us recall from Section II that the requirements on traffic protection are 1) protection against quantum attackers, and 2) all frames protection. We assess the suitability of SDLS to meet these requirements as follows:

1) Though the standard specifies a baseline implementation, the SDLS can be implemented with practically any symmetric encryption algorithm and/or authentication algorithm; It is therefore possible to select appropriate symmetric algorithms and use them with appropriate key lengths that are considered secure against quantum attackers. We argue therefore that SDLS allows to meet requirement 2 (i.e., protection against quantum attackers) if configured appropriately.
2) When protecting a frame with SDLS, the respective key(s) and crypto parameters used for frame protection are defined by a 16 bit *Security Parameter Index* (SPI) that is transmitted by the sender of a frame to its receiver as part of the security header. An SPI is bound to a so-called *virtual channel*. Multiple virtual channels may exist with a single satellite that can be protected individually and independently from other virtual channels. By transmitting the SPI as part of the security header, the SDLS protocol in general allows to determine the key(s) to use for frame protection on the receiving end on an ad-hoc basis per frame. This is necessary to meet requirement 1 (i.e., all frames protection).
   It is noted that, here, it is implicitly assumed that the keys are already available onboard (i.e., at the receiving end in case of TC frames) at the time when the frame arrives and that they are working well. With that assumption, it is required to carry out some ”initialization phase” already on ground to establish keys that the onboard security function can then use after launch to decrypt the first frame received. This assumption is reasonable for satellite systems, since before launch satellites are usually well protected. Once in nominal operation, key(s) can be updated. After a successful key renewal, the new keys can be used in future sessions.
   We highlight that with this approach implicitly defined by SDLS, cryptographic primitives or properties such as zero roundtrip time key exchange [19] or immediate decryption [1] are not needed in order to achieve all

frames protection. Instead, SDLS (though not explitictly stated in the standard) makes use of the fact that the a stateful is acceptable as is one that works only in closed system (as discussed in Section II).

From the above discussion we conclude that SDLS - if set up appropriately - is well suited for traffic protection on TT&C links. Therefore, when discussing the key management approaches in Section IV, we silently assume that they are used together with SDLS for traffic protection; in particular, for all approaches discussed we assume that keys or key sets have a unique SPI that can be used with SDLS.

As discussed before, when using SDLS generically, a secure initialization phase is needed on ground. Therefore, we put attention on this aspect when discussing key management approaches below; we note that once the protection functions have been initialized on ground and once keys have been established, the communication for key management (and in particular for key renewal) can go *in parallel* to nominal traffic (at times when occupation of the bandwidth for security management is acceptable from operations point of view).

IV. Key Management

In this section we will discuss several key management approaches for satellite systems and how to use them together with SDLS. We will focus on the mechanisms for key renewal or key update only as argued above.

A. SDLS EP

After release of the SDLS, CCSDS has published the SDLS Extended Procedures (SDLS EP). This standard specifies procedures for remote management of the onboard protection function [11]. These procedures build on symmetric cryptography only: a few symmetric *master keys* are injected into the onboard protection function (and its counterpart on ground) prior to launch. These master keys can be used to upload symmetric *session keys* which can then be assigned to a virtual channel for traffic protection under SDLS.

a) Secure Initialization: For secure initialization, it is required to inject a set of master keys but also some session keys to the protection function prior to launch. The session keys are needed in order to allow for all frame protection at times when no session key has been uploaded yet.

b) Key Renewal: With SDLS EP, keys can be uploaded to the onboard security function. Only session keys can be uploaded; the to-be-uploaded session key is protected by a master key during upload.

c) Compliance against requirements: We summarize the compliance of SDLS EP against the requirements established in the previous section.

All Frames Protection: It is recalled that this property is fulfilled as discussed in Section III assuming that the onboard and ground security functions are initialized as described above.

Protection against Quantum Attackers: SDLS EP rely only on symmetric cryptography for key upload. Therefore, it is - in general - suitable to provide protection against quantum attackers if configured correctly. Therefore, a concept that relies on SDLS EP can meet the requirement if the primitives are chosen appropriately.

Post Compromise Security: SDLS EP builds on pre-shared keys to provide confidentiality for key renewal. By that, it does not provide post-compromise security: we recall that new keys are introduced to the system by an upload that is protected by a master key. Once the master keys are known to an attacker, it is always able to decrypt the session keys that are newly uploaded. Therefore, after an attacker's compromise of all keys, it is not possible to exclude the attacker from the communication in the future. It is thus not possible to establish a secure state after a finite period in time following on a comromise. Neither weak nor strong post compromise security can be met.

Forward Secrecy: Whether or not SDLS EP achieve forward secrecy depends on the implementation of the master and session key handling on ground. A conclusive statement cannot be made on conceptual level.

d) Compliance Summary: SDLS EP occupies the link to a minimum extent for key renewal and can provide protection against quantum attackers. However, it does not provide post compromise security.

We note that, following the argument above, any solution that builds only on pre-shared keys and symmetric crypto mechanisms for key renewal cannot meet post compromise security. Therefore those concepts are ruled out. Consequently, in the remainder, solutions building only on symmetric pre-shared keys are not further considered. We note that with this we have to accept an increase of link occupation for security management.

B. Stateless Authenticated Key Exchange

Authenticated key exchange protocols have been studied for long time by the cryptographic community [2]–[4], [9], [18], [21] as standalone primitive. Authenticated key agreement (AKE) schemes are used when two parties need to establish a common secret (from which they can then derive application specific symmetric traffic keys). If public keys are assumed to be authentic, key agreement schemes with a broad range of security guarantees exist, including weak post compromise security [13].

In this section we consider AKE in its generic stateless form. These AKE schemes are usually designed for use in open systems. They are designed to establish fresh and independent keys with every new key agreement session and each session is carried out independently from past sessions. This provides good robustness if, in an error scenario, no key is available. Conceptually, a simple authenticated key agreement protocol is depicted in Figure 3.

a) Secure Initialization: In order to establish an initial key set that allows for decryption of the first frame in the very first contact after launch it is required to establish trust between the (onboard and ground) security functions and, based on the established trust, to carry out the AKE protocol already on ground. When doing so, one or more symmetric key sets are

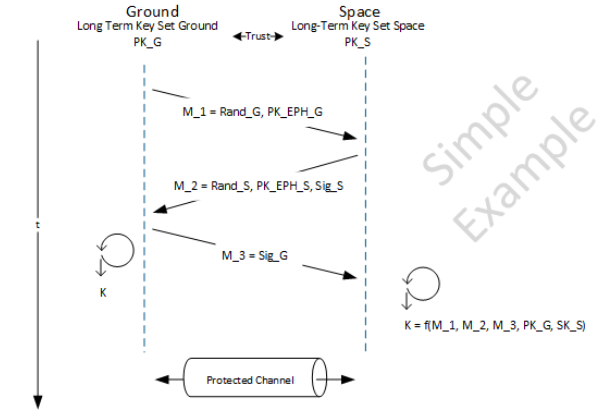

Fig. 3. Simple example of a key agreement protocol. The key that is derived at the end of the session depends only on the content of the messages and the long-term keys.

derived and one of these can be used to protect the first frame in the first contact.

We note that we use the AKE to generate a set of shared symmetric keys on ground that can be used for traffic protection. These keys are used only for traffic protection; they are not used to provide authentication or confidentiality for an upcoming AKE run.

b) Key Renewal: To establish a new key set, a new run of the AKE protocol is carried out. Once the protocol has been carried out successfully and new keys have been established, these can be used for protection of future frames: For this, the frame needs to carry the respective SPI.

c) Compliance against requirements: We now discuss the compliance against the requirements established in Section II of a key management approach building on AKE:

All Frames Protection: We recall that this property is fulfilled as discussed in Section III assuming that the security functions are initialized as described above.

Protection against Quantum Attackers: Stateless AKE protocols have been developed that are secure in quantum attacker models [7]. It is noted that such protocols make use of both, post-quantum secure public key encryption and post-quantum secure signatures (or comparable post quantum secure public key primitives providing authentication). Therefore, a concept based on stateless AKE allows to meet requirement 2 if the protocol is designed correctly with suitable primitives.

Post Compromise Security and Forward Secrecy: Forward secrecy is a well established security property for AKE protocols today [7], [9]. As forward secrecy relies on ephemeral key pairs generated for key agreement, we argue that carrying out a new run of such AKE protocol after compromise but without attacker intervention could be sufficient to establish new secrets unknown to the attacker and by that make the attacker blind of what is being communicated after compromise (here, we assume that new entropy can be fed to the function generating the ephemeral keys). Having that said, a concept based on stateless AKE can meet weak post compromise security.

However, strong post compromise security cannot be met by stateless AKE [13]: Once the attacker has knowledge of long term keys, it can *at any time in the future* carry out a new AKE run. I.e., even if the attacker is passive for some time, stateless AKE does not allow to re-establish a secure state.

d) Compliance Summary: From security point of view, we can state a concept based on stateless AKE can meet all requirements except for strong post compromise security. Instead, only weak post compromise secuity can be achieved.

As we stated above, in order to allow for post quantum security, the AKE protocol itself needs to be secure against quantum attackers. Ciphertext and public keys of PQ secure public key encryption schemes and signatures of PQ secure signature schemes are large with sizes in the order of many kB; also they are expensive to compute onboard. Therefore, also from link occupation point of view, stateless AKE is not an ideal choice.

C. Stateful AKE and Key Evolution Protocols

In this section we discuss stateful AKE and key evolution protocols. These protocols are stateful in the sense that future protocol executions depend on the current state. As discussed in Section II, we accept a stateful protocol for TT&C link protection and its key management.

We briefly discuss two approaches of stateful AKE / key evolution protocols: Ratcheted Key Exchange and Stateful Key Agreement.

a) RKE: Ratcheted Key Exchange (RKE) is a relatively new cryptographic primitive. It has been (practically) developed in the context of secure mobile messaging [17] and has since then been studied also from a theoretical point of view [1], [6], [8], [26].

Conceptually, one can think of a ratcheted key agreement protocol as an ever ongoing key agreement protocol where - after a secure initialization process providing mutual authentication - new entropy is introduced from time to time to *update* existing keys (or key sets). In contrast to AKE, where a protocol run a) is defined in terms of a fixed number of messages and b) is stateless meaning that no state is carried over to the next execution of the AKE, RKE is continuously ongoing (i.e., the number of messages is not defined up front) and the keys are constantly *updated* with each new RKE message. The updated key depends on the new (entropy) message sent and the previous key in use but is still computationally indistinguishable from a truly random key.

The OTR protocol [6] was the first protocol deploying ratcheted key exchange. Conceptually, in this protocol, keys are derived continuously by making use of the Diffie Hellman Key Agreement and a standard key derivation function, cf. Figure 4

Whenever Alice receives a ”new” message from Bob (i.e., when the ”speaker” changes), Alice puts a new Diffie Hellman

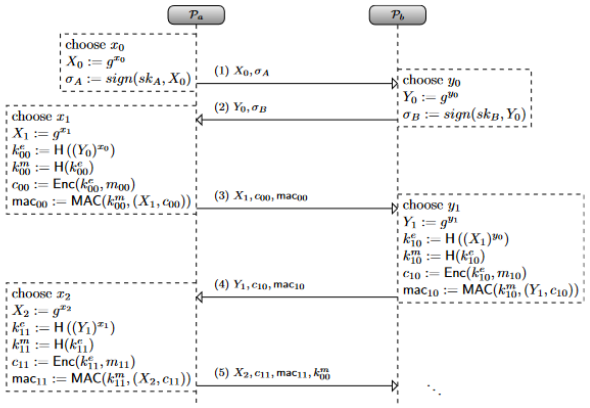

Fig. 4. OTR inititialization and key renewal. Figure taken from [17].

share in the next response to Bob (and vice versa). When receiving a new Diffie Hellman share, Alice uses the newly received share and the latest share she has sent to Bob to compute a new shared secret. This shared secret is then used as input for key derivation together with some secret established with the previous key update. By this, new entropy is fed constantly to the system; still, future keys depend on current keys. It is noted that forward secrecy and post compromise security are design goals of RKE and by this inherent features of secure RKE protocols [1].

We highlight that instead of exchanging Diffie Hellman shares, one could exchange post quantum secure KEM public keys or ciphertexts as well.

With the notable exeption of the first two message[4], all messages are protected against tampering by *symmetric* algorithms, i.e., by MACs; no digital signatures are used[5]. The MAC keys are derived from the evolving shared secret. This means that an attacker, after compromise, will be out of the system again after new entropy has been introduced to the system (as new keys, incl. authentication keys, depend on the newly introduced entropy).

The concept has recently been generalized to Secure Channel Establishment with Key Evolution (SCEKE) [5]. We highlight two key features of RKE/SCEKE when used to as mechanism to renew keys for TT&C link COMSEC protection:

1) No PQ signatures: after initial setup, no PQ sigantures are needed. We note that recent RKE/SCEKE protocols define a dedicated *intitialization procedure* which establishes the initial trust and does not make use of signatures at all [5], [17].
2) New entropy that is constantly fed to the system allows for both, forward secrecy and post compromise security.

[4]The first two messages in OTR are signed to provide peer authentication

[5]We note that the rationale for this design is to provide repudiation (since any message that is authenticated by a MAC could have been sent by either Alice or Bob); however, in the context of TT&C link key management, this approach allows to reduce overhead for key management

b) Stateful Key Agreement: We discussed stateless key agreement schemes in the previous section and came to the conclusion that it cannot achieve post-compromise secure key agreement. As for RKE schemes, stateful AKE schemes can achieve post-compromise security [13]. The protocol proposed in [13] achieves post-compromise security for AKE as follows: whenever an AKE session is carried out, the session keys are derived as usual; on top, a *token* is derived which is input to the KDF in the next AKE run. This allows to achieve post-compromise security. However, in contrast to RKE/SCEKE, where keys undergo evolution and past keys authenticate future keys, the protocol from [13] still builds on an AKE protocol (that need PQ signatures for authentication).

c) RKE vs Stateful AKE: As stated above, RKE requires to mutually authenticate initially. Once the protocol is set up, only symmetric primitives are used for message authentication. This is a benefit from our point of view as a) we accept initialization on ground in a secure environment and b) no PQ signature are needed in this case (having in mind that low overhead for security management is desired, cf. Section II). Therefore we prefer this approach against stateful secure AKE.

For secure initialization and key renewal we focus on RKE, as initialization and key-renewal of stateful AKE follows the approach defined for stateless AKE described in the previous section.

d) Secure Initialization: Secure initialization is a key aspect for RKE/SCEKE protocls meaning that a secure initialization procedure is integral part of an RKE/SCEKE protocol. This procedure is to be carried out on ground.

e) Key Renewal: Also, key renewal is a key aspect for RKE/SCEKE protocls. I.e., an RKE/SCEKE protocol definition includes a mechanism for constant key update. This to be carried out for key renewal.

f) Compliance against requirements: We now discuss the compliance against the requirements established in Section II of a key management approach building on RKE:

All Frames Protection: We recall that this property is fulfilled as discussed in Section III assuming that the security functions are initialized as described above.

Protection against Quantum Attackers: RKE protocols have been developed that are secure against quantum attackers [6]. It is noted that such protocols make use of, post-quantum secure public key encryption but not necessarily of post-quantum secure signatures. We conclude that a concept that builds on RKE can meet the requirement if the protocol is designed correctly with suitable primitives.

Post Compromise Security and Forward Secrecy: Since both forward secrecy and post compromise security are design goals of RKE, we conclude that a concept that builds on RKE for key renewal can meet these requirements.

g) Compliance Summary: A concept building on steful AKE/RKE can meet all requirements including strong post compromise security. The overhead for key management is reduced compared to stateless AKE, as PQ signatures are not needed. This is another benefit compared to stateless AKE schemes.

TABLE I
SUMMARY: SECURITY REQUIREMENTS AGAINST KEY UPDATE / KEY RENEWAL CONCEPTS; HERE PQC = POST QUANTUM SECURITY, FS = FORWARD SECRECY, WPCS = WEAK POST COMPROMISE SECURITY, AND PCS = POST COMPROMISE SECURITY.

Concept	PQS	FS	WPCS	PCS
SDLS EP	✓	✓	✗	✗
Stateless AKE	✓	✓	✓	✗
Stateful AKE/RKE	✓	✓	✓	✓

We argue that - since we consider ground compromise only - to further reduce the key management overhead, the concept could be opptimized as follows: only the satellite has a long term (PQ secure) KEM public key; after initial set up on ground where the public key is exchanged and the first keys symmetric keys are derived (incl. authentication keys), ground regularly sends a new KEM ciphertext introducing fresh randomness to the system, i.e., no entropy is generated by the satellite after initial setup.

V. CONCLUSION ANND FUTURE WORK

As stated in Section III, a concept based on SDLS allows to meet post quantum security and all frames protection if complemented by an appropriate key management concept (providing in particular secure initialization and secure key update). Table I summarizes the key management concepts assessed in Section IV and whether they meet the requirements or not. We conclude that stateful AKE/ RKE are a promising candidate to provide secure key management for TT&C link protection. The combination of SDLS and stateful AKE/RKE allows to satisfy all our requirements.

We see several lines of research that require a follow up:

- This paper did consider only the TT&C link of a single satellite. Applicability (and extension) of the concepts discussed in this paper for constellations, including inter-satellite link communication, requires further analysis.
- Future work could formalize the security requirements that are needed for TT&C link COMSEC protection and its key management, define a security model and develop a suitable protocol that is provably secure in this model.
- We did consider only SDLS for traffic protection (where it is assumed that the good keys are already available on the receiving end of a channel when a frame is received). This allowed us to tackle traffic protection and key renewal independently. It is an interesting problem to find alternative solutions to all frames protection, e.g., protocols based on 0-RTT key exchange [19] that integrate with the constraints from the space domain (recall that a TC transfer frame is of max length 1024 octets, cf. Figure 1).
- It seems like strong post compromise security implies that - once ground and space are "out of sync", i.e., do not share the same key - communication with the satellite is lost and cannot be re-established. It is an open problem wether a solution exists that allows for both, strong post compromise security and re-establishment of a secure key after "loss of synchromnization".

REFERENCES

[1] Joël Alwen, Sandro Coretti, and Yevgeniy Dodis. The double ratchet: Security notions, proofs, and modularization for the signal protocol. In Yuval Ishai and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2019*. Springer International Publishing, 2019.

[2] Christoph Bader, Dennis Hofheinz, Tibor Jager, Eike Kiltz, and Yong Li. Tightly-secure authenticated key exchange. In Yevgeniy Dodis and Jesper Buus Nielsen, editors, *Theory of Cryptography*. Springer Berlin Heidelberg, 2015.

[3] Mihir Bellare and Phillip Rogaway. Entity authentication and key distribution. In Douglas R. Stinson, editor, *Advances in Cryptology — CRYPTO' 93*, 1994.

[4] Simon Blake-Wilson, Don Johnson, and Alfred Menezes. Key agreement protocols and their security analysis. In Michael Darnell, editor, *Crytography and Coding*. Springer Berlin Heidelberg, 1997.

[5] Olivier Blazy, Ioana Boureanu, Pascal Lafourcade, Cristina Onete, and Léo Robert. How fast do you heal? a taxonomy for post-compromise security in secure-channel establishment. In *32nd USENIX Security Symposium (USENIX Security 23)*. USENIX Association, 2023.

[6] Nikita Borisov, Ian Goldberg, and Eric Brewer. Off-the-record communication, or, why not to use pgp. WPES '04. Association for Computing Machinery, 2004.

[7] Joppe Bos, Leo Ducas, Eike Kiltz, T Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehle. Crystals - kyber: A cca-secure module-lattice-based kem. In *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*, 2018.

[8] Jacqueline Brendel, Rune Fiedler, Felix Günther, Christian Janson, and Douglas Stebila. Post-quantum asynchronous deniable key exchange and the signal handshake. In Goichiro Hanaoka, Junji Shikata, and Yohei Watanabe, editors, *Public-Key Cryptography – PKC 2022*. Springer International Publishing, 2022.

[9] Ran Canetti and Hugo Krawczyk. Analysis of key-exchange protocols and their use for building secure channels. In Birgit Pfitzmann, editor, *Advances in Cryptology — EUROCRYPT 2001*. Springer Berlin Heidelberg, 2001.

[10] CCSDS. Space data link security protocol, 2015.

[11] CCSDS. Space data link security protocol - extended procedures, 2020.

[12] CCSDS. Tc space data link protocol, 2021.

[13] Katriel Cohn-Gordon, Cas Cremers, and Luke Garratt. On post-compromise security. In *2016 IEEE 29th Computer Security Foundations Symposium (CSF)*, 2016.

[14] Andrei Costin, Syed Khandker, Hannu Turtiainen, and Timo Hämäläinen. Cybersecurity of cospas-sarsat and epirb: threat and attacker models, exploits, future research. In *Workshop on the Security of Space and Satellite Systems (SpaceSec23)*, 2023.

[15] James Curbo and Gregory Falco. A research agenda for space flight software security. In *2023 IEEE 9th International Conference on Space Mission Challenges for Information Technology (SMC-IT)*, 2023.

[16] Gregory Falco. The vacuum of space cyber security. 2018.

[17] Tilman Frosch, Christian Mainka, Christoph Bader, Florian Bergsma, Jörg Schwenk, and Thorsten Holz. How secure is textsecure? In *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 457–472, 2016.

[18] M. Choudary Gorantla, Colin Boyd, and Juan Manuel González Nieto. Modeling key compromise impersonation attacks on group key exchange protocols. In Stanisław Jarecki and Gene Tsudik, editors, *Public Key Cryptography – PKC 2009*. Springer Berlin Heidelberg, 2009.

[19] Felix Günther, Britta Hale, Tibor Jager, and Sebastian Lauer. 0-rtt key exchange with full forward secrecy. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology – EUROCRYPT 2017*. Springer International Publishing, 2017.

[20] Julian Huwyler, James Pavur, Giorgio Tresoldi, and Martin Strohmeier. Qpep in the real world: A testbed for secure satellite communication performance. In *Workshop on the Security of Space and Satellite Systems (SpaceSec23)*, 2023.

[21] Tibor Jager, Eike Kiltz, Doreen Riepel, and Sven Schäge. Tightly-secure authenticated key exchange, revisited. In Anne Canteaut and François-Xavier Standaert, editors, *Advances in Cryptology – EUROCRYPT 2021*. Springer International Publishing, 2021.

[22] Minghao Lin, Minghao Cheng, Dongsheng Luo, and Yueqi Chen. Clextract: Recovering highly corrupted dvb/gse satellite stream with contrastive learning. In *Workshop on the Security of Space and Satellite Systems (SpaceSec23)*, 2023.
[23] Mark Manulis, C. Bridges, R. Harrison, V. Sekar, and A. Davis. Cyber security in new space: Analysis of threats, key enabling technologies and challenges. *International Journal of Information Security*, 2021.
[24] Microchip. Rtg4 fpga datasheet, 2022.
[25] James Pavur and Ivan Martinovic. Building a launchpad for satellite cyber-security research: lessons from 60 years of spaceflight. *Journal of Cybersecurity*, 8(1), 2022.
[26] Bertram Poettering and Paul Rösler. Towards bidirectional ratcheted key exchange. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology – CRYPTO 2018*. Springer International Publishing, 2018.
[27] Frederick Rawlins, Richard Baker, and Ivan Martinovic. Death by a thousand cots: Disrupting satellite communications using low earth orbit constellations. In *Workshop on the Security of Space and Satellite Systems (SpaceSec23)*, 2023.
[28] Tobias Scharnowski, Felix Buchmann, Simon Wörner, and Thorsten Holz. A case study on fuzzing satellite firmware. In *Workshop on the Security of Space and Satellite Systems (SpaceSec23)*, 01 2023.
[29] Pietro Tedeschi, Savio Sciancalepore, and Roberto Di Pietro. Satellite-based communications security: A survey of threats, solutions, and research challenges. *Computer Networks*, 216, 2022.
[30] Johannes Willbold, Moritz Schloegel, Manuel Vögele, Maximilian J Gerhardt, Thorsten Holz, and Ali Reza Abbasi. Space odyssey: An experimental software security analysis of satellites. In *SP*, 2023.
[31] Xilinx. Virtex family datasheet, 2015.