Educational Nanosatellite Hacking

My-Loan Dang
Institut Supérieur de l'Aéronautique et de l'Espace (ISAE-SUPAERO), Université de Toulouse, 31055 Toulouse, FRANCE
firstname.lastname@student.isae-supaero.fr

Jérome Lacan
Institut Supérieur de l'Aéronautique et de l'Espace (ISAE-SUPAERO), Université de Toulouse, 31055 Toulouse, FRANCE
firstname.lastname@isae-supaero.fr

Doriane Perard
Institut Supérieur de l'Aéronautique et de l'Espace (ISAE-SUPAERO), Université de Toulouse, 31055 Toulouse, FRANCE
firstname.lastname@isae-supaero.fr

Thibault Gateau
Institut Supérieur de l'Aéronautique et de l'Espace (ISAE-SUPAERO), Université de Toulouse, 31055 Toulouse, FRANCE
firstname.lastname@isae-supaero.fr

***Abstract*— Security issue concerning telecommunication with educational nanosatellite missions is currently underestimated or not considered critical and often neglected. Loosing contact with a nanosatellite in the academic world is, until now, pretty common. Without thrusters, safety studies on LEO orbits usually guaranty their atmospheric re-entry. Even hacking a nanosatellite would have low impact concerning others spacecraft, apart for being a space debris doing random attitude changes. That may change. For example, electrical propulsion allows to significantly modify an orbit, and opens new possibilities, and therefore new threads.**

While classic space actors are taking a great care for securing telecommunications, academic teams usually lake of funding, expertise and knowledge. This paper aims at providing a warning for academic nanosatellite conceptors, and provide some recommendations and concrete methods. This work is relying on a concrete use-case, a space mission involving an academic nanosatellite: the NIMPH project. Platform is a 3U Cubesat. It is planned to be operated by radio-amateur on UHF/VHF bandwidth. We conducted a security analysis in order to identify the current and future threats for this kind of specific space mission (academics nanosatellite). We also recommend procedures that can be used to protect against such risks, with as low effort as possible for the development, and minimal cost.

This paper also describe symmetric key cryptography schemes. They are not yet deployedin the framework of academic projects to our knowledge, but they could be easily usable. Two cryptographic procedures have been especially identified to authenticate and/or encrypt telecommands on the up-link. As a proof of concept, Authenticated Encryption with Associated Data construction based on the Advanced Standards Algorithm in Galois Counter Mode (AES/GCM) has consequently been implemented. We simulate how a telecommand would be processed through this cryptographic procedure, from authenticated encryption on the ground to verification and decryption by the On-Board-Computer.

1. BACKGROUND

NIMPH Specifications

Over the past few decades, an increasing number of Cubesats have been sent on Lower-Earth Orbits (LEO) to conduct short-term space missions for educational purposes. As such, the Nanosatellite to Investigate Microwave Photonics Hardware (NIMPH)[2] is currently being designed and developed within the Centre Spatial Universitaire de Toulouse (CSUT), the Laboratoire d'Analyse et d'Architecture des Systèmes (LAAS) and Centre d'Etude pour la Recherche Nucléaire (CERN) in the framework of the JANUS project. NIMPH, which is to be launched by the end of 2020, is a nanosatellite that aims at studying the influence of space environment and radiations on optoelectronic components. This 3U Cubesat will be deployed with an optoelectronic payload for a mission of 2 years on LEO, which parameters can be found in table1.

Table 1. NIMPH orbit parameters [1]

Visibility per day	21 min
Time between two passages	12 h
Number of passages per day	3 - 4

During its operational phase, the nanosat will communicate with two types of ground stations.

- Based at the ISAE-SUPAERO, the dedicated ground station is designed to send telecommands (TC) on the uplink and receive telemetry data (TM) on the downlink. A back-up ground station at Université Paul Sabatier could also be used for the same purpose.
- A set of amateur radio ground stations will be able to receive and decode house keeping data from the satellite.

Communication between space and ground segments is supported by the AX.25 protocol on amateur radio frequencies (UHF/VHF), with a data flow of 9600 bps [1]. TCs and TMs packets structure is based on the Packet Utilization Standards (PUS) as defined by the European Cooperation for Space Standardization (ECSS), which is fully compliant with the Space Packet protocol issued by the Consultative Committee for Space Data Systems (CCSDS).

Legal and philosophical framework

Although security has always been a major issue for military, industrial, or scientific space missions [2], it has still to be addressed in the framework of academic projects. Indeed, it is usually considered that the the equipment and knowledge required to hack educational Cubesats are only concentrated within a limited academic and amateur community. Thus, the

[2]https://www.csut.eu/project/nanosatellite-to-investigate-microwave-photonics-hardware/

978-1-7281-2734-7/20/$31.00

probability that attacks against nanosatellites occur is usually seen as negligible, as those who would have the capacity to perform such attacks would have no interest in doing so. Furthermore, for space communication based on amateur radio frequencies, encryption has traditionally been prohibited, due to both legal and philosophical reasons. Enforcing the Article 25 of International Radio Regulations [3], the Radio-Amateur Satellite Corporation (AMSAT) indeed considers that data transmitted on radio-amateur frequencies should remain open source [4].

Nonetheless, security analysis for nanosatellites is becoming more and more important in the light of recent technological, legislative and philosophical evolutions. Today, the worst-case scenario following a malicious attack would be a Denial-Of-Service (DOS) that would lead to the permanent loss of the spacecraft. But even more concerning, the development of electric propulsion capabilities on future nanosatellites could turn them into weapons to target other spacecraft.

Taking into account the risks entailed by providing propulsion to nanosatellites, the Federal Communication Commission has recently issued a Notice of Proposed Rulemaking [5] to propose encryption on both the uplink and downlink on amateur radio frequencies. In a following comment, the AMSAT has agreed to the necessity of encrypting TCs [4].

2. PROBLEM STATEMENT

In this context, it has become of utmost importance to review the attacks that can be performed against space missions, as well as to outline the different security strategies that can be adopted to counter them. Which of those procedures are applicable to nanosatellites?

Furthermore, in the particular framework of the NIMPH project, what procedure could be designed and implemented to secure data transmission between space and ground segments?

3. STATE-OF-THE-ART

Attacks performed against satellites have been well documented by CCSDS Standards [2] and academic research, in the framework of industrial and military space missions. Threats against spacecraft are usually classified according to the three aspects of Security, namely: Confidentiality, Integrity and Availability (CIA).

1. Confidentiality aims at protecting data against unauthorized disclosure.
2. Integrity refers to protection against data corruption.
3. Availability is defined as the capability to access critical resources.

Attacks against Confidentiality

Data confidentiality can be breached if communications between ground stations and spacecraft are intercepted. An attacker could either tap transmitted data, or identify which entities are communicating. To counter eavesdropping attacks, cryptography based on symmetric ciphers is widely used for point-to-point space communications [6].

In the framework of nanosatellite communications, a protocol based on the One-Time-Pad (OTP) has been proposed for downlink communication encryption [7]. It highlights the perfect secrecy ensured by the One-Time-Pad (OTP), which consists in XORing the payload with a non-repeating and non-reusable key. Nevertheless, OTP only insures data confidentiality but not integrity, as the resulting ciphertext is malleable (due to the linearity of XOR operator). Alternatively, XTEA [8], the encryption algorithm supported by the Cubesat Space Protocol (CSP) is not either a relevant choice. In addition to its high computation cost, several attacks have been documented [9, 10]. Finally, it should always be remembered that encryption without authentication does not insure security, as data can be tampered with even when kept confidential.

Attacks against Integrity

Insuring the integrity of the uplink and downlink communications is critical. Sending malicious telecommands to a satellite could prompt maneuvres that would modify its orbit after launch, potentially allowing a malicious attacker to target other spacecraft. Alternatively, an attacker could impersonate the spacecraft to send forged housekeeping data, inciting the ground operating team to send telecommand that would not be adapted to the real satellite state. Procedures to detect data modification during transmission are thus necessary. Nowadays, cyclic redundancy checks (CRC) computed at different levels of data encapsulation enable the detection of accidental data corruption, but do not protect against malicious modification. Indeed, as CRC is linear, it can be easily modified in order to fit corrupted data. As a consequence, a non linear integrity check should be designed and implemented. Similarly, authentication of both transmitting ends should be provided to insure that TCs and TMs have been issued by authorized entities. Integrity schemes recommended by the CCSDS Standards [11] are further developed in the next sections. For nanosatellites, the lightweight solution "CubeSec and GndSec" [12] has been developed, proposing a hardware implementation of AES/GCM on the ATXMega128 microcontroller.

Data integrity should also consider replay attacks, which refer to the retransmission of authentic data that could have been previously tapped and recorded by a hacker. Resending a valid telecommand to a satellite could result in unexpected maneuvres, as said before. Furthermore, continuously replaying the same command (such as Reset command) could also provoke excessive energy consumption. To prevent those attacks, timestamps can be added to transmitted data. Nonetheless, a procedure authenticating and verifying timestamps would require clock synchronization between spacecraft and ground stations, as well as take into account the delay due to space propagation. Session tokens, Sequence Numbers (SN), one-time passwords can also be used to assess if a data has been previously transmitted [2]. For NUTS development [13], it has been chosen to use randomly generated SN, to be sent along with the message in order to insure uniqueness of each packet. Upon receival, the recipient should check if the SN has been received in a previous message. If the packet is authentic, the SN is stored in memory and the data is further processed. Sequence Numbers have been designed to be generated independently so that an attacker that has access to one or more SN can not derive the previous or subsequent SN. Finally, challenge-response protocol could also be used to authenticate the emitting entity when any doubts on identity arise [14].

Corrupted software or tainted hardware components also represent threats to spacecraft data integrity. Forbes et al. [15] developed a scenario of supply chain attack in off the shelf components. A malware injected in the Operating System

before launch could be triggered by a communication packet sent by a malicious ground station. This malware could, for instance, reconfigure flight software. Alternatively, it could overwrite cryptographic keys used to secure communications so as to lock out the authorized operating ground teams and take control of the spacecraft. A Secure Cyber-Physical System [15] has been proposed to prevent malicious binaries from running and to detect anomaly through Machine Learning.

Attacks against Availability

Loss of availability of satellite resources is most often a consequence of integrity attacks, resulting in Denial-of-Service (DOS). For instance, the satellite orientation could be altered following a malicious TC, thus putting the payload or communication devices out of service. DOS might also occur if flight software is hacked and security thresholds are modified, unexpectedly triggering transition to safe mode. Cristini [16] suggests to implement Fault Detection, Isolation and Recovery procedure on autonomous satellites, so as to enable reconfiguration of onboard software.

Jamming communication between satellite and ground station is another possible attack against data availability. This attack could be executed through network flooding that would disrupt or obstruct data transmission as the satellite passes over an authorized ground station. Those attacks can be particularly dangerous when security procedures require data synchronization [17]. When using session token or sequence numbers, desynchronization between satellite and ground station would imply failure of authentication requests. Solutions such as frequency hopping, spread spectrum or multiple link paths are used to counter those attacks [2]. More lightweight solutions for LEO satellites have been developed, such as resynchronization challenge protocols [17].

4. OBJECTIVE

As malicious communication with spacecraft seems to be the most accessible means to hack nanosatellites, the purpose of this research is to design a procedure to secure data transmission between a spacecraft and its dedicated ground station. For the particular case of NIMPH development, a focus on cryptographic schemes has been chosen to protect communication between the Cubesat and the ISAE-SUPAERO ground station. The objective is to design, implement and test first an authentication and/or encryption algorithm to protect TCs on the uplink. Developing an authentication-only procedure for TMs on the downlink would further enhance data transmission security.

5. STANDARDS ALGORITHMS FOR SYMMETRIC KEY CRYPTOGRAPHY

First, it should be remembered that focusing on protecting data transmission between space and ground segment relies on implicit security hypothesis. It is indeed assumed that the integrity of the software and hardware components of the spacecraft, both on the ground and on flight, is insured. Similarly, the integrity of the authorized ground station is supposedly protected. Finally, the operating team on the ground is considered as a Trusted Party.

To secure space communications, symmetric key cryptography is traditionally considered the most efficient scheme [6]. In this case, cryptographic operations are based on a shared secret key known by both satellite and ground station prior to communications. Given the limitation of nanosatellites missions - low memory space, low bandwith environment, high transmission error - the cost of public key cryptography would exceed the benefits provided by such a scheme. Indeed, for point-to-point communication, a public key cryptography scheme, also referred to as asymmetric cryptography, is not necessary. Thus, as NIMPH is designed to receive TCs from a single ground station based at ISAE-SUPAERO, symmetric key cryptography is the most relevant framework to develop authentication and/or encryption procedures.

In-depth insight of up-to-date cryptography can be found in the open-source Graduate Course offered by Dan Boneh [18]. Cryptographic algorithms shall be selected among the standards issued by the National Institute of Standards and Technology (NIST)[3], and their implementation shall be compliant with the Federal Information Processing Standards Publication 140-2 (FIPS PUB 140-2)[4] as well as with the standards issued by the Agence Nationale de la Sécurité des Systèmes d'Information[5] (ANSSI).

In symmetric key cryptography, integrity schemes are based on Message Authentication Code (MAC). A MAC is a fixed-length tag, computed by applying cryptographic functions to payload data with a secret key. The tag, also called digest, is unique for each message and can only be produced by entities that share the secret key. It can be used as a security header to be transmitted along with the authenticated data by the emitting end. Upon receival, the MAC of the payload data shall be computed by the recipient and compared with the received MAC. If the received MAC and the computed MAC match, the data is authenticated and may be further processed. If not, the packet is considered as inauthentic and is discarded. Two symmetric key cryptographic procedures are identified by the CCSDS Standards [11] to protect data integrity.

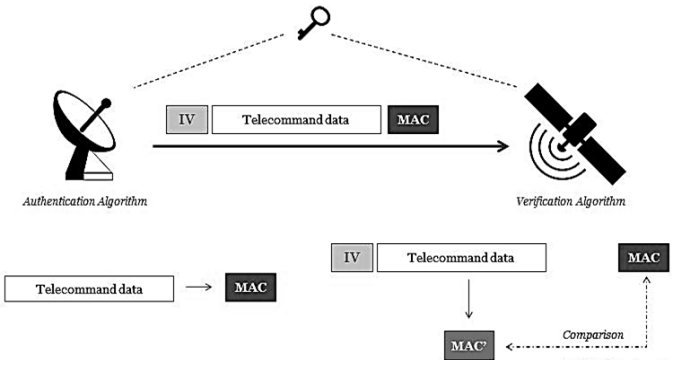

Figure 1. Authenticating TCs through symmetric key cryptography

When only authentication is required, which is the case for traditional nanosatellites, or for TMs sent on the downlink, a hashing function can be used to compute Message Authentication Codes (HMAC). CCSDS Standards [11] as well as NIST recommendations [19] enhance the relevance of using Secure Hash Algorithm-2 with a 256 bit-key (SHA-256) as the underlying hash function.

[3]https://www.nist.gov/
[4]https://csrc.nist.gov/projects/cryptographic-module-validation-program
[5]https://www.ssi.gouv.fr/

For data that should be both encrypted and authenticated, such as TCs sent to nanosatellites provided with propulsion, Authenticated Encryption with Associated Data (AEAD) is highly recommended. AEAD schemes use an encryption algorithm rather than a hashing function to produce the MAC. In particular, using the Advanced Encryption Standard in Galois Counter Mode (AES/GCM) [20] would be of interest when data need to be partially encrypted but fully authenticated.

Finally, for space missions that would require secure communication between several entities, like for a constellation of different satellites communicating together or with a network of more than one ground station, the Rivest-Shamir-Adleman (RSA) protocol can be used to generate Digital Signature to identify each communicating entity. This public/private key cryptography scheme will not be further detailed in this paper, but extensive literature can be found in CCSDS and NIST standards.

Authentication based on HMAC/SHA-256

HMAC is a Merkle–Damgård construction, which iteratively applies the same hashing function to the different block of data to be authenticated, as shown in figure2.

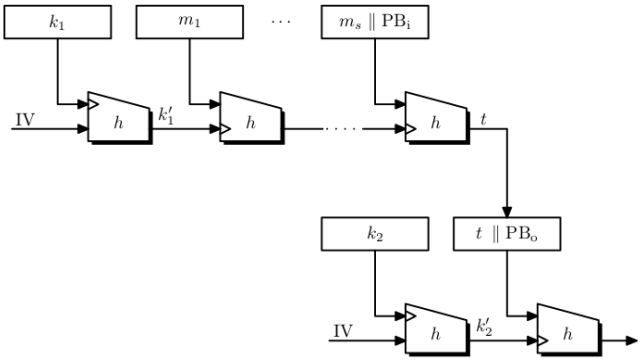

Figure 2. Hash Message Authentication Code [18]

Two secret keys, k1 and k2, are used to insure the authenticity of the procedure. These two keys are both derived from the same secret key which is XORed with two different pads, the ipad and the opad. The first key is prepended to the data i.e. used as the first block data. The second key is hashed and used at the last step of the procedure. The Initialization Vector (IV), supposed to be unique for each message, shall be transmitted along with authenticated data and MAC tag. If not repeated for two different messages, the IV protects against replay attacks.

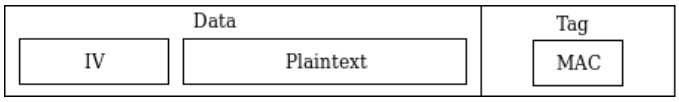

Figure 3. Data packet authenticated by HMAC

In order to build a secure authentication scheme, the underlying hashing algorithm should be a one-way compression function. It should have high preimage resistance, meaning that an attacker that has access to the output should not be available to determine the input of the function. Collision resistance should also be considered when choosing a hashing function, to insure that two different inputs can not produce the same output. Over the past few decades, NIST has successively selected 3 different Secure Hash Algorithms (SHA), renewing standards when security issues arose. SHA-1 can not be used anymore as collisions have been demonstrated [21]. CCSDS Standards [11] recommend any version of SHA-2 using output length of 256, 224 or 512 bits. For all SHA-2 algorithms, the collision resistance is equal to half the length of the output block in bits, whereas the preimage resistance is equal to the whole length of the output block [19]. As collision resistance is always weaker than preimage resistance, the first criteria should be the strongest security requirement.

For nanosat application, Edon-R256, one of the competing algorithms for SHA-3, has been investigated to authenticate uplink communications for the Norwegian University Test Satellite (NUTS) [13]. As compared to the other algorithms reviewed in that study, Edon-R256 has been shown to be the most efficient in terms of computing time and code size.

Authenticated Encryption with Associated Data using AES/GCM

The AEAD schemes such as AES/GCM enable the splitting of the message into two different parts, the associated data, which usually encompasses headers, and the plaintext. Two different cryptographic operations are performed depending on the data type: the associated data are authenticated but sent in clear, while the plaintext is both authenticated an encrypted. The data packet to be transmitted is composed of the IV, which is used to initialize the algorithm, the associated data, the ciphertext and the MAC tag, as presented in Fig. 4.

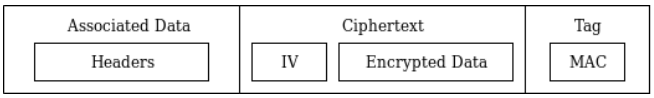

Figure 4. Authenticated Encryption with Associated Data

AES specifications. The Rijndael algorithm designed by Joan Daemen and Vincent Rijmen is the block cipher that was selected by NIST to become the Advanced Encryption Standards (AES) in 2001 [22]. This is the underlying cryptographic function used in GCM to encrypt the plaintext.

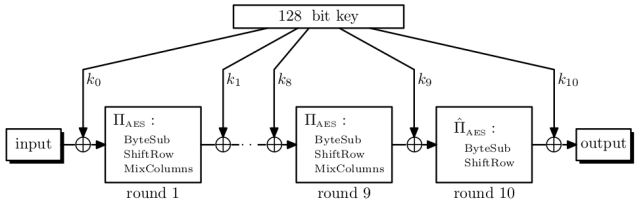

Figure 5. Advanced Encryption Standards with 128-bit key [18]

AES takes as input a 128-bit block as 4x4 matrix of bytes, and a secret key that can be of length 128, 192 or 256 bits. First, the secret key is expanded to derive different keys, to be used successively between each rounds of operations. Operations of substitutions, row shifting and column mixing are iterated on the byte matrix, as shown in figure5. All those operations are invertible, as AES is a two-way cryptographic function: if a ciphertext can be computed from a plaintext, the ciphertext should be decrypted into the plaintext. For AES using a 128-bit key (AES-128), the block is processed through 10 rounds.

AES in Counter Mode. For a message which length exceeds 128 bits, the data is split in 128-bit blocks to be encrypted by

AES. AES can then be performed in different modes, that can either be sequential or parallelizable. The sequential modes of encryption such as Cipher Block Chaining or Nested MAC have been extensively studied [18] but will not be further detailed in this paper. The latest modes, also called Counter Modes, are based on the independent encryption and decryption of each message block.

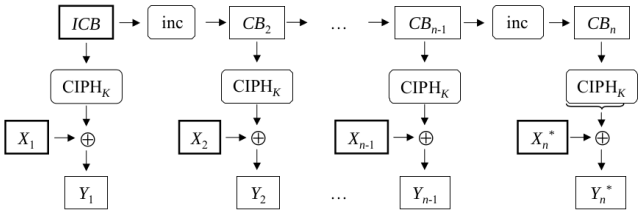

Figure 6. AES in Counter Mode [20]

For GCM, a message comprising more than one 128-bit block is encrypted as presented in figure6. A counter is initialized with an IV to produce the initial counter block (ICB). ICB is thus split into two parts: a random nonce or IV that identifies the message, and a counter that is incremented for each message block. For each message block, the counter is encrypted using AES and the resulting output is XORed with the current message block (Xi in figure6). The counter is then incremented and the same step is performed to encrypt the following message blocks.

AES/GCM specifications. Designed by David McGrew and John Viega, AES/GCM is an Encrypt-then-MAC procedure, as presented in Fig. 7. First, the plaintext (P) is encrypted using AES in Counter Mode with a secret key K. The resulting ciphertext (C) and the associated data (A) are then authenticated [20].

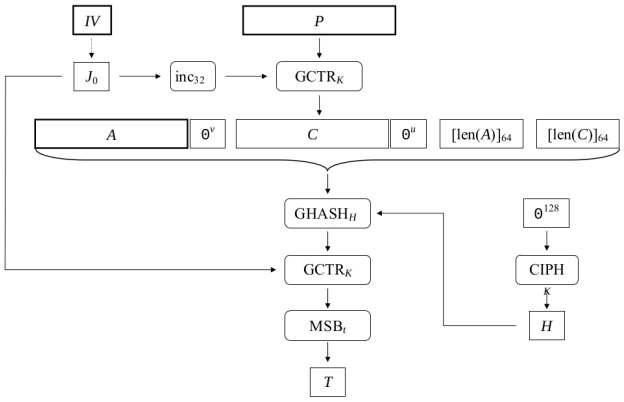

Figure 7. Authenticated Encryption with AES/GCM [20]

The authentication is based on the Galois Hash Function (GHASH). The input to the GHASH function comprises the associated data and the ciphertext padded with zeros, concatenated with the 64-bit representation of the length of the associated data and the ciphertext. The hash subkey H is derived from the AES encryption of a 128-bit "zero" block using key K. It is then used by the GHASH function to perform binary Galois Field multiplications on the input data. The resulting block is encrypted and then truncated to produce the final tag (T).

The decryption/verification, as shown in Fig. 8, is symmetric

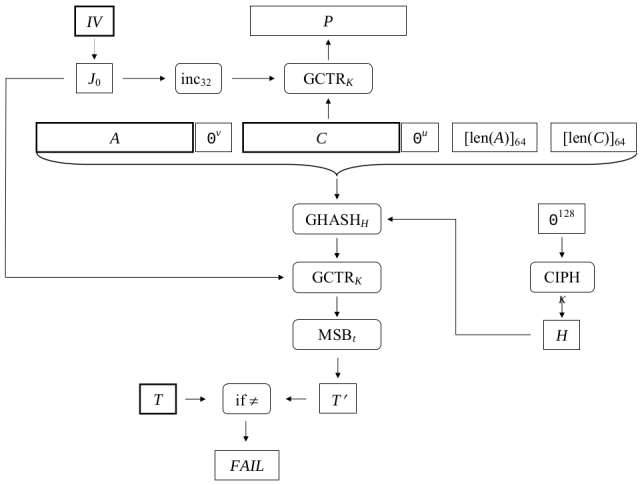

Figure 8. Authenticated Decryption with AES/GCM [20]

to the encryption procedure. First, the length of each input, supposed to be known by both transmitting ends, is verified. If the verification succeeds, the associated data and ciphertext are compressed, encrypted and truncated following the authentication steps previously presented. If the recomputed tag is similar to the received tag, the ciphertext is decrypted and further processed.

6. SECURITY PARAMETERS MANAGEMENT

The security strength of a procedure not only relies on the algorithm, but also on other security parameters such as the secret key, the Initialization Vector and the length of the MAC tag.

Key generation, sharing and storage

Cryptoperiod—Defining the length of the operational lifetime of a given key is necessary to reduce the risk of cryptanalytic attacks that can be performed against data protected by that key. It limits the time available to conduct such attacks, and the period within which the key is exposed to unauthorized disclosure. Thus, the system vulnerability increases with longer cryptoperiods, which can be measured either as the maximum amount of data that can be authenticated or encrypted, or in time units. For a symmetric authentication and encryption key, the cryptoperiod shall not exceed 2 years [23]. As nanosatellites missions usually last 2 years or less, using a single key throughout the mission would be sufficient.

In terms of data, the cryptoperiod is limited by the maximum number of invocations of the algorithm with the same key. The total number of blocks of plaintext and associated data protected by the same key shall not exceed 2^64 [20]. The total number of TCs to be encrypted and authenticated during NIMPH mission is estimated to 50 000 per year. As the AX.25 TC Transfert Frame Information Field comprises at most 2048 bits, the maximum number of 128-bit blocks in a TC amounts to 16. Thus, it is estimated that a total number of 1 600 000 data packets are to be authenticated and encrypted on the uplink throughout the mission. Using a single cryptographic key would meet the requirement of the maximum cryptoperiod. If the CCSDS Standards [6] recommend the use of two different type of keys, the Master Keys and the Session Keys, which lifecycle should be split

into different phases and states, this model is considered in this research as not adapted to the limited resources and requirements of nanosatellites.

Key generation— As the secret lies in the keys, they shall be generated so as to be unpredictable and thus, indistinguishable from random. Two different procedures are recommended to generate keys [24].

- Pseudo Random Bit Generator (PRBG), as validated by NIST [25] can be used. The security of the generated keys depend on the seed from which the PRBG is initialized. The seed shall be kept secret and shall comprise enough entropy to provide sufficient randomness to the system.
- Key Derivation Functions (KDF) [26] can generate keys from other secret pre-shared keys, referred to as Master Keys. Based on a HMAC construction, KDF implementation shall take into account the security of the key-derivation-key algorithm, the preimage resistance of the underlying hash function and the length of the derived keys, also called Sessions Keys.

Key storage—Keys shall be securely stored on-board and in the ground station in a non-volatile memory, as the risk of losing them following a power incident can not be afforded. Those keys can be generated using either PRBG or KDF on the ground, and uploaded on the On-Board-Computer (OBC) prior to flight. If the on-board memory is not sufficient to store all the keys that are to be used over the nanosatellite lifetime, another possibility could be to generate keys on the ground during the mission and transmit them to the spacecraft through the Over-The-Air-Rekeying (OTAR) [6] protocol. Alternatively, only the secret seeds or the Master Keys can be uploaded on the spacecraft prior to the flight, and used during the mission to generate or derive new Session Keys all along the lifetime of the spacecraft.

As for short-term missions supported by nanosatellites, it would be relevant to generate keys on the ground using a PRBG from a FIPS-140 approved module and upload them on the OBC prior to launch. For NIMPH, as a single key could be used during the nanosatellite lifetime, it can be considered that the available on-board memory, which amounts to 100Gbits, exceeds the volume required to store this key.

Uniqueness requirement on the pair (IV, key)

As stated before, for a given key, the Initialization Vectors used in the authenticated encryption algorithm should be unique for each data packet. Quantitatively, the probability that the same pair (IV, key) is used to authenticate and encrypt two distinct data packet shall not exceed $2^{-}32$ [20]. Using a single key to authenticate and encrypt traffic on the uplink, with an estimated amount of 100 000 TCs for a 2-year mission, it could be considered that an IV of length 96 bits is sufficient.

IVs of length 96 bits can be generated either by a deterministic or randomized procedure [20]. If the procedure is deterministic, the IV is composed of two parts: a fixed field known by both emitting ends, identifying the context or device, and a counter that is incremented for each invocation. As the IVs are, in this case, supposed to be known by each transmitting party, it does not necessarily have to be transmitted along with the data. Nonetheless, given the high probability of data packet loss during space communication, the risk of counter desynchronization between spacecraft and ground station would be heavy to be afforded.

Thus, using a randomized generation of IVs seems more adapted. The IV, considered as a critical security parameter, should be kept secret before invocation. When used to encrypt and authenticate data, it should be transmitted in the packet with the data. On the emitting side, i.e. for the ground station, designing a randomized IV generation procedure should insure against IV repetition. In particular, the loss of power or reset of modules shall not imply violation of the IV freshness requirement. Against this risk, different procedures can be considered. A fresh key can be established so as to make sure that the IV repetition does not imply the repetition of the pair (key, IV). Alternatively, the IVs and keys can be stored in a non-volatile memory. For RBG construction, one or more values of IVs ahead of the value currently used can be stored. Finally, the RBGs can also be re-initialized with a fresh seed that either would be provided by the operating team, or would come from a physical non-deterministic source of entropy.

On the recipient side, the IVs transmitted along with the data should be stored in a non-volatile memory. In order to authenticate data against replay attacks, the IVs can be used as Sequence Numbers to identify each TC and assess if they have been previously received. Assuming that a total number of 100 000 TCs are received by NIMPH during its lifetime, with each TC being identified by a 96-bit IV, memory space required for all IVs would reach at most 10Mbits. As the available on-board memory is estimated to 100Gbits, storing all IVs to protect against replay attacks would be feasible for NIMPH if only TCs are encrypted and authenticated. To authenticate TM against replay attacks, on the downlink, additionnal memory space should be available both on the ground and on board.

Tag truncation

For space communication with critical bandwidth environment, it could be considered to reduce the length of the MAC tag to be transmitted with the data. NIST Standards [19] present the rules that should be applied to MAC truncation.

First, the left most bits shall be selected. For HMAC construction, the truncated length shall be at least twice the required collision resistance, and can not be less than 32 bits. The truncated length is also a function of the number of failed verifications that can be performed with a fixed given key, as well as the acceptable probability to authenticate forged data [19]. When using the AES/GCM scheme, the MAC tag length is a function of the number of input blocks to the GHASH functions and the acceptable probability of ciphertext forgery. NIST Standards [20] detail the maximum combined length of ciphertext and associated data that can be accepted in a single packet for 32 and 64 bits MAC tag. Usually, the length of truncated tag is 64 bits.

For the nanosatellite missions, two symmetric key cryptography schemes represent relevant choices for secure communication. First, for authentication only, the HMAC construction based on SHA-256, as compared to other SHA-2 and SHA-3 algorithms, would be the most efficient in terms of cycle per bytes[6]. This option would thus optimize processing time and power, while still providing sufficient security. Nevertheless, for NIMPH development, it would be more interesting to implement the AES/GCM procedure. The main advantage that AES/GCM offers over HMAC/SHA-256 is that it provides partial encryption in addition to full authentication. It can also be used as a standalone authentication procedure,

[6]https://en.wikipedia.org/wiki/Template:Comparison_of_SHA_functions

if encryption is not required. It could thus be used, at first, to authenticate both TCs on the uplink and TMs on the downlink. If, within a few years, legislation evolves so as to allow TC encryption on amateur radio frequencies, it would not be necessary to design and implement a new procedure as authentication and encryption are independently performed. AES/GCM could then be used to authenticate and encrypt TCs on the uplink while authenticating the TMs on the downlink. It is also important to notice that, on either side of the communication, GCM only uses AES encryption function, and not the decryption function. This reduces the code footprint. Furthermore, as only the counter is processed by AES, this encryption step can be computed prior to the use of the algorithm and stored in hardware, optimizing processing time and power on board. Another motivation for using AES/GCM is that the the computation can be parallelized and pipelined. In addition to high processing speed, this reduces the propagation of errors throughout the produced ciphertext [7]. Finally, as AES/GCM does not require padding, unlike HMAC/SHA-256, it reduces the additional overhead, which is important to consider in low bandwidth environment.

7. IMPLEMENTATION OF AN AUTHENTICATED ENCRYPTION PROCEDURE

The objective of this research was to develop and test a cryptographic scheme that could be used on both ends of the communication. As the AES/GCM cryptographic scheme is considered as more suitable for NIMPH mission, it was chosen to be implemented. For the ground segment, a software implementation of AES/GCM would enable encryption and authentication of a telecommand. Because of the limited resources available on-board, a hardware implementation of the AES/GCM decryption and verification functions has been considered.

In the framework of this research, implementation was based on open-source libraries or modules that are not FIPS 140-2 certified. For actual implementation of cryptography on NIMPH spacecraft, libraries and proprietary modules that have been certified by NIST shall be selected.

NIMPH Telecommunication Specification

To determine at which level of data encapsulation it would be possible to encrypt and authenticate TC data, NIMPH communication protocols shall be reviewed.

Communication between space and ground segments will be supported by the AX.25 protocol on amateur radio frequencies. Inside the Transfer Frame, the Information Field comprising 256 bytes is shown in Fig. 9 for telecommands.

Packet Primary Header (48)			Packet Data Field (0-1992)			
Packet ID (16)	Packet Seq. Control (16)	Packet Length (16)	Secondary Header (10 * 8)	Data		Packet Error Control
Associated Data			Ciphertext (Encrypted Telecommand)	Initialization Vector	Message Authentication Code	
128			0-1784	96	32	16

Figure 9. NIMPH data packet as encrypted and authenticated with AES/GCM

The Data packet of 2040 bits is structured on the Space Packet model [27] for the Primary Header and the Packet Utilization Standards [28] for the Secondary Header. For TCs, using AES/GCM would enable authenticated encryption of the data inside the Packet Data Field. A 96-bit IV is added for decryption, as well as the 32-bit MAC tag for verification. The IV and MAC can either be prepended as a security header or appended as a security tail. The primary and secondary headers are processed as authenticated associated data. It can also be considered to add 2 control bits in the Packet Data Field, one assessing if the packet is encrypted, the other one indicated if it is authenticated. In particular, this would enable to discriminate between TCs sent from the ISAE-Supaero ground station and other packets sent from amateur radio ground stations.

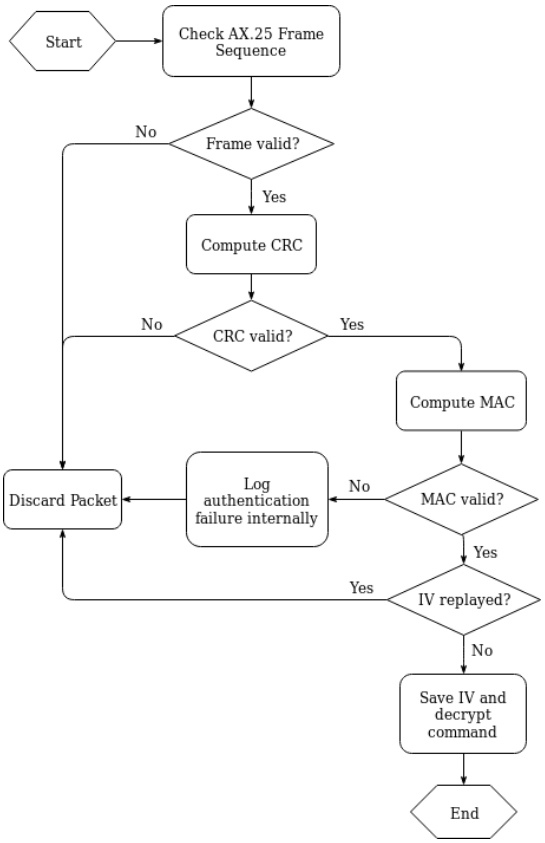

Figure 10. Processing received TC on-board

Upon reception, the packet is processed as presented in Fig. 10. In order to avoid cryptanalytic attacks, packets that fail authentication verification shall not be acknowledged. No error message shall be sent back to the emitter of the packet to indicate the failure. Data shall be silently discarded, but authentication errors shall be logged internally. Only after an excessive number of unsuccessful authentication, can the recipient terminate the connection and/or notify the ground station that recurrent security issues occurred.

Software implementation

All programming languages provide AES/GCM software implementation through open-source libraries. Cryptodome, an open-source Python library, has been chosen to conduct tests of AES/GCM implementation. It is indeed one of the most popular libraries in Python for educational purpose [18]. The algorithm utilizing Cryptodome defines an authenticated encryption method and a verification and decryption method. For each of this method, authentication and verification can be used independently from encryption/decryption. This software implementation can be used to test compatibility and interoperability with hardware implementation.

Hardware implementation

For the space segment, hardware implementation was based on the open-source programs provided by Tariq Bashir Ahmad [29] within the OpenCores community. Written in Verilog, the programs describe a full duplex block consisting in an encryption-authentication block and a decryption-verification block. Those programs were used to build an Intellectual Property (IP). A simulation using the test bench provided with the open-source program was successfully run on the encryption-authentication block. In order to implement this IP on a Xilinx FPGA, it was integrated into a hardware design on Vivado software. That design comprises interfaces between the Zynq microprocessor and the FPGA, modelized by IPs written in VHDL. It was then implemented onto a ZedBoard Zynq-7000 ARM/FPGA SoC.

The Initialization Vector, associated data, plaintext and secret key are provided by a C algorithm executed by the microprocessor. This algorithm also offers an interface to read the output data, i.e. the ciphertext and MAC tag, in order to test if the open-source AES/GCM performs proper data encryption and authentication. For testing, input data can be extracted from the test bench provided by the IP author. Another alternative would be to determine input data formatted according to NIMPH telecommunication specifications, as in Fig. 9. Those input data, along with an IV and secret key generated accordingly to security requirements mentioned in section 6, can be processed through the Python algorithm so as to get the corresponding ciphertext and MAC tag. Those resulting data can then be compared to the values obtained with the AES/GCM hardware implementation.

8. Conclusion

As part of NIMPH project, this paper offered an insight into security threats that are likely to be emphasized by the arrival of electric propulsion. Attacks against confidentiality, integrity and availability of spacecraft resources, as well as security procedures to counter them have been outlined. This literature review was necessary in order to assess if those strategies could be applicable to nanosatellites, and especially to NIMPH mission. As the most accessible way to hack nanosatellites would be to send malicious telecommands, a particular focus on securing uplink communication was adopted. The purpose of this research was thus to design, implement and test a procedure to secure data transmission from the ISAE-Supaero ground station to NIMPH spacecraft. This particularly represents a challenge as NIMPH space communication will be supported by amateur radio frequencies. If encryption has until now been prohibited on those frequencies, legislation is likely to change with the increasing use of electric propulsion on-board. Thus, a solution had to be determined taking those recent technological and legislative evolutions into account.

Two symmetric key cryptography schemes were investigated to authenticate and/or encrypt TCs. HMAC/SHA-126, a keyed hash algorithm, can be used if authentication only is required. Alternatively, AES/GCM would represent the most adapted security construction, since it provides partial encryption in addition to full authentication. More over, as it can be used as a standalone authentication algorithm, AES/GCM is compliant with the current legislation prohibiting encryption. If legislation were to evolve towards mandatory TC encryption for nanosatellites provided with propulsion, using AES/GCM would still represent a relevant choice.

To evaluate the cost of implementing AES/GCM in terms of additional overhead, computational power and processing time, a deeper outlook into NIMPH specification was taken. Encrypting the data field inside the PUS protocol, while processing Space Packet primary header and PUS secondary header as associated data could be an interesting option. Given this possible scheme, AES/GCM software implementation was conducted with the Python open-source library Cryptodome. Hardware implementation, that would be more suitable for on-board processing, was also performed using an open-source Intellectual Propery written in Verilog. This enabled the testing of compatibility and interoperability among different implementations.

Perspectives: comparing AES/GCM software and hardware implementation

This research can be further developed by testing software and hardware implementation that would be more adequate to NIMPH. For software implementation, the most relevant option to integrate cryptographic functions in NIMPH flight software would be OpenSSL. Although it has not been used in this research, this C library has been validated multiple times by NIST[7]. Considered as one of the most secure open-source cryptography library, its implementation can be found in various FIPS 140-2 validated modules. For hardware implementation, several proprietary modules that also are FIPS 140-2 certified can be purchased from hardware constructor[8].

In order to choose between software and hardware implementation, criteria such as power consumption, processing time and memory footprint shall be considered. As on NIMPH, the flash memory so far exceeds the space required for current applications, power consumption should be the first parameter to take into account.

For this reason, the purpose of further research could be to implement AES/GCM both on software and hardware onto XTratum, the device that will be used as OBC for NIMPH. This would enable the measurement of power consumption so as to evaluate power profile to integrate to the overall NIMPH power budget.

9. Acknowledgement

Thanks as well to Federico Pace and Arnaud Dion from the ISAE-Supaero DEOS department for their advice and support regarding hardware implementation. Finally, a special shout out to Clément Chalumeau who carried out the hardware implementation on FPGA, from the packaging of the IP to the implementation on the Zedboard.

References

[1] T. G. J. L. François RAVEL, Ethan CHERKI, "Telecommunication protocol for nimph nanosatellite project," 2018.

[2] C. C. for Space Data Systems, *Security Threat Against Space Missions, Information Report, Green Book*, 2014.

[3] I. T. Union, "Article 25 on amateur services, wrc-03 final acts, rr25.2a," in *World Radiocommunication Conference*, 2003.

[7] https://csrc.nist.gov/projects/cryptographic-module-validation-program/Certificate/1747

[8] https://www.xilinx.com/content/xilinx/en/products/intellectual-property/

[4] R.-A. S. Corporation, "Comments of radio amateur satellite corporation on notice of proposed rulemaking, fcc 18-159, 84 fed. reg. 4742," 2019.

[5] D. . Federal Communication Commission Washington, "Notice of proposed rulemaking," *FCC 18-159, 84 Fed. Reg. 4742*, 2019.

[6] C. C. for Space Data Systems, *Symmetric Key Management, Draft Recommended Practice, Red Book*, 2018.

[7] C. I. Banuelos, "Development of information assurance protocol for low bandwidth nanosatellite communications," Naval Postgraduate School Naval Postgraduate School United States, Tech. Rep., 2017.

[8] D. J. Wheeler and R. M. Needham, "Tea, a tiny encryption algorithm," in *International Workshop on Fast Software Encryption*. Springer, 1994, pp. 363–366.

[9] J. Lu, "Related-key rectangle attack on 36 rounds of the xtea block cipher," *International Journal of Information Security*, vol. 8, no. 1, pp. 1–11, 2009.

[10] Y. Ko, S. Hong, W. Lee, S. Lee, and J.-S. Kang, "Related key differential attacks on 27 rounds of xtea and full-round gost," in *International Workshop on Fast Software Encryption*. Springer, 2004, pp. 299–316.

[11] C. C. for Space Data Systems, *Crytographic Algorithm, Informational Report, Green Book*, 2014.

[12] O. Challa, G. Bhat, and J. Mcnair, "Cubesec and gndsec: A lightweight security solution for cubesat communications," *Small Satellite Conference https://digitalcommons.usu.edu/smallsat/2012/all2012/25/*, 2012.

[13] S. Prasai, "Access control of nuts uplink," Master's thesis, Institut for telematik, 2012.

[14] T.-H. Chen, W.-B. Lee, and H.-B. Chen, "A self-verification authentication mechanism for mobile satellite communication systems," *Computers & Electrical Engineering*, vol. 35, no. 1, pp. 41–48, 2009.

[15] L. Forbes, H. Vu, B. Udrea, H. Hagar, X. D. Koutsoukos, and M. Yampolskiy, "Securecps: Defending a nanosatellite cyber-physical system," in *Sensors and Systems for Space Applications VII*, vol. 9085, 2014, p. 90850I.

[16] F. Cristini, "Amélioration de la résilience de systèmes spatiaux soumis à des menaces: vers des réseaux de satellites autonomes," Ph.D. dissertation, ISAE-Institut Supérieur de l'Aéronautique et de l'Espace, 2014.

[17] I. Lasc, R. Dojen, and T. Coffey, "Countering jamming attacks against an authentication and key agreement protocol for mobile satellite communications," *Computers & Electrical Engineering*, vol. 37, no. 2, pp. 160–168, 2011.

[18] V. S. Dan Boneh, *A Graduate Course in Applied Cryptography*. Stanford University, 2017.

[19] "The keyed-hash message authentication code (hmac)," National Institute of Standards and Technology, Tech. Rep., 2008.

[20] M. Dworkin, "Recommendation for block cipher modes of operation: Galois/counter mode (gcm) and gmac," National Institute of Standards and Technology, Tech. Rep., 2007.

[21] X. Wang, Y. L. Yin, and H. Yu, "Finding collisions in the full sha-1," in *Annual international cryptology conference*. Springer, 2005, pp. 17–36.

[22] V. R. Joan Daemen, "Advanced encryption standard, federal information processing standards publication 197," National Institute of Standards and Technology, Tech. Rep., 2001.

[23] E. Barker, "Recommendation for key management, nist special publication 800-57 part 1, revision 4," National Institute of Standards and Technology, Tech. Rep., 2016.

[24] A. R. Elaine Barker, "Recommendation for cryptographic key generation," National Institute of Standards and Technology, Tech. Rep., 2019.

[25] J. K. Elaine Barker, "Recommendation for random number generation using deterministic random bit generators, special publication 800-90a revision 1," National Institute of Standards and Technology, Tech. Rep., 2015.

[26] L. Chen, "Recommendation for key derivation using pseudorandom functions," National Institute of Standards and Technology, Tech. Rep., 2009.

[27] C. C. for Space Data Systems, *TC Space Data Protocol, Recommended Standard, Blue Book*, 2015.

[28] E. C. for Space Standardization, *Telemetry and telecommand packet utilization*, 2016.

[29] T. B. Ahmad, "Gcm-aes block specification."