Architecting Information Security Services for Federated Satellite Systems

Marc Sanchez Net[a], Iñigo del Portillo[b] and Bruce Cameron[c] and Edward F. Crawley[d]
Massachusetts Institute of Technology, Cambridge, MA 02139, USA

This paper investigates the provision of information security services in Federated Satellite Systems. We initiate the discussion by describing possible threats that the system faces, as well as the specific security services that have to be provided in order to mitigate them. Next, we define a set of five primal security functions that a federated satellite system has to implement and propose the Interaction State Model, a functional model that characterizes the security state when two federates are interacting with each other. Differences in these security states are then used to define the Interaction State Machine, a transition state diagram that can be used to rapidly identify which security functionality has to be provided in order to securitize an unreliable interaction. Finally, we apply the Interaction State Model and Interaction State Machine in a multi-hop setting where information is relayed through multiple federated satellite system participants. Based on this discussion, we define the concept of best-effort vs. guaranteed services as applied to the context of federated satellite systems and information security services. Their usefulness in architecting information security services is finally demonstrated through an illustrative example.

[a] PhD Candidate, Department of Aeronautics and Astronautics, 77 Massachusetts Avenue, Room 33-409. 02139 MA, USA
[b] PhD Candidate, Department of Aeronautics and Astronautics, 77 Massachusetts Avenue, Room 33-409. 02139 MA, USA
[c] Director, System Architecture Lab, Department of Aeronautics and Astronautics, 77 Massachusetts Avenue, Room 31-413. 02139 MA, USA
[d] Professor, Department of Aeronautics and Astronautics, 77 Massachusetts Avenue, Room 33-413. 02139 MA, USA

Nomenclature

$Cert_A$ = Node A certificate

D = Message hash

F_i = FSS federate

H = Message header

M = Message payload plus header

N_i = FSS node

PrK_A = Node A private key

PuK_A = Node A public key

SyK = Symmetric key

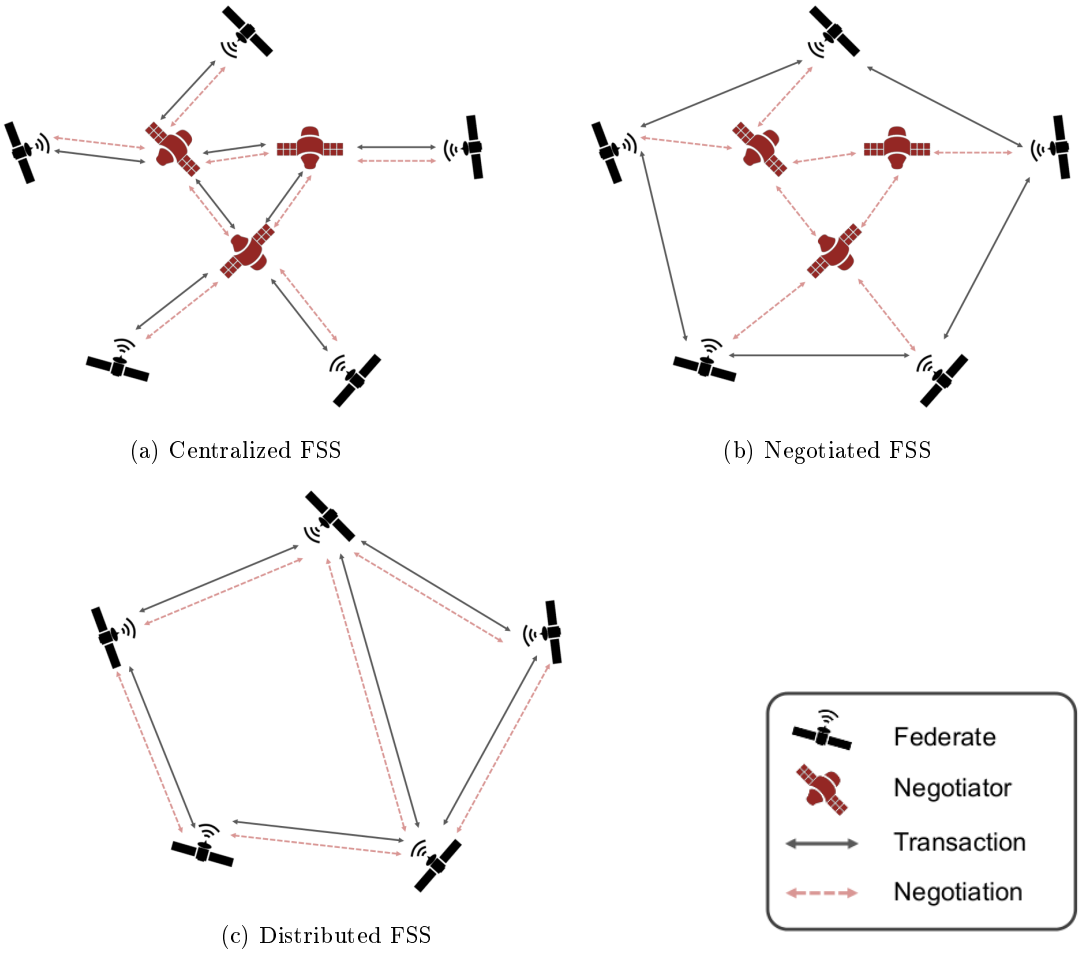
I. Introduction

A. Federated Satellite Systems

The concept of a federated satellite system (FSS) has been proposed in the literature as a distributed cloud-like architecture for improving the provision of communication and processing services to space-based assets [1]. Two main rationales justify this improvement: First, the FSS architecture aims at leveraging under-utilized capabilities from spacecraft that are in orbit [2]. Second, it provides the basis for a business-oriented network architecture through which service provision is distributively provided in exchange for monetary compensation [3].

References [2] and [4] provide a detailed description of the FSS concept. A satellite federation is defined as a set of "spacecraft that engage in opportunistic collaboration with each other during their mission lifecycle" [2]. The federates, i.e. the spacecraft participating in a federated system, can be suppliers, customers or both depending on whether they provide or utilize resources in the federation. Interoperability among them is based on two types of interactions, *transactions* and *negotiations*. The former refers to the mechanisms by which "federates exchange resources among each other" [4]. In contrast, *negotiation* refers to the ability of federates to efficiently allocate resources from suppliers to customers.

The functionality inherent to FSS transactions and negotiations can be allocated to different elements of the system in order to generate three canonical FSS architectures [4]: Centralized (Fig.



(a) Centralized FSS

(b) Negotiated FSS

(c) Distributed FSS

Fig. 1 FSS Architectures

1a), that is, both negotiating and transacting is done through an FSS-dedicated node, the *negotiator*. Negotiated (Fig. 1b), i.e. negotiator nodes are tasked with efficiently allocating supplier's resources to customer jobs, but the actual transaction is executed directly. Finally, fully distributed FSS architectures (Fig. 1c) require no negotiator nodes since both transaction and negotiation is directly carried out by the federates.

B. Research Goals

Given that the FSS paradigm aims at efficiently allocating space-based resources among different interoperable participants, it is clear that ensuring a secure and trustworthy environment is a fundamental requirement for such a system. As noted in the FSS literature, malicious federates could potentially participate in the federation, thus hindering both its usefulness and attractiveness [2]. Therefore, the fundamental goals of this paper are to:

1. Identify the different aspects of the FSS security architecture based on previous literature.

2. Characterize the security threats that the FSS has to address in order to provide a trustworthy environment where resources can be exchanged.

3. Identify the information security services that have to be provided in order to mitigate the previously characterized threats.

4. Identify security functionality that the FSS needs to implement in order to provide the aforementioned information security services.

5. Define the concept of best-effort vs. guaranteed information security services for multi-hop interactions in an FSS network.

Note that this objective is different from providing specific recommendations on which security suites to implement in a real FSS. Other standardization entities such as the Internet Research Task Force [5] or the Consultative Committee for Space Data Systems (CCSDS [6]) already provide valuable guidance on which standards to utilize in the space domain. Note also that, apart from the first research goal, we specifically center this discussion around information security services. Other types of security services such as physical, transmission and transactional security are considered outside the scope of this paper.

In order to address these research goals, this paper is structured as follows: Section II provides an overview of advancements in information security in the context of space-based systems. We focus our attention on areas especially relevant to the FSS network such as security mechanisms in peer-to-peer or ad-hoc networks that are subject to high latency connectivity. Section III focuses on identification of information security threats vis-a-vis the FSS. These are treated as basic requirements that the information security architecture of the FSS has to address. Then, section IV presents the Interaction State Model as a basic tool to understand quality of service in information security of an FSS architecture. Finally, section V summarizes the findings of this paper and identifies areas of future work.

II. Background and Related literature

A. Security Services in Space Networks

The CCSDS defines, in reference [7], three complimentary viewpoints that should be considered when architecting a secure space-based network: The physical view, the informational view and the transmission view. The physical view treats an FSS as a pool of distributed independent software and hardware resources. These include space assets participating in the federation, as well as any ground assets that are used in order to provide end-to-end connectivity to a spacecraft. Risks inherent to the physical view are mostly related to malfunction and/or loss of a resource and/or asset. Next, the transmission viewpoint "provides mechanisms for hiding the presence of the communications link and/or preventing the link from being jammed" [7]. In that sense, the transmission viewpoint largely influences the system's physical layer and, consequently, is tightly connected to the physical viewpoint. On the other hand, the information viewpoint treats FSSs as a set of nodes that exchange and process information flows. Security risks are generally associated with the validation, corruption and/or loss of information during the execution of a service in the federated system. Finally, and not included in the original CCSDS categorization, we argue that FSSs should also provide transactional security services that ensure a trustworthy environment in the marketplace of suppliers and customers. These security services are analogous, for instance, to the functionality provided by the 3-D Secure protocol currently used for credit card transactions (see for instance [8]).

Fig. 2 provides a schematic representation of the different security aspects that should be considered for a FSS network. Observe that the provided categorization has been deliberately layered in order to indicate the need to provide security services of layer L as basic infrastructure for supporting layer $L+1$. Observe also that information security has been highlighted as it is the primary concern of this paper (see section I B).

B. Information Security Services

Information security (InfoSec) services are provided in a network in order to mitigates risks inherent to the informational view of the security architecture. They are broadly defined as the set of practices that a system has to provide in order to protect information from unauthorized and

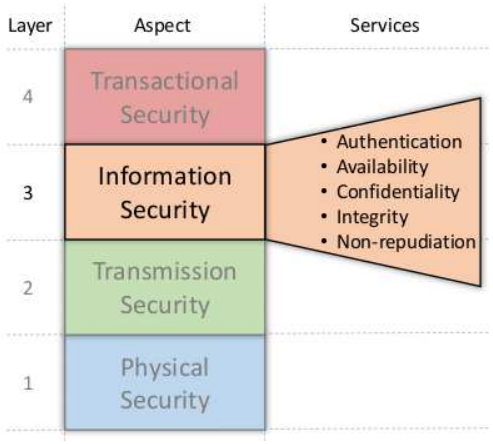


Fig. 2 Layered FSS security architecture

malicious access [7], [9]. Information security typically encompasses five fundamental services (see Fig. 2), the definition of which is adapted to the FSS context:

- *Authentication*: Set of functions that allow a federate to verify the identity of another federate.
- *Availability*: Set of functions that ensure resources from an FSS supplier cannot be disrupted or disabled through an attack from any internal or external element to the FSS.
- *Confidentiality*: Set of functions that allow information in the FSS network to be disclosed and accessed only by parties that are properly authorized to do so.
- *Integrity*: Set of functions that allow a federate to verify that the information being received has not been modified by any member of the federation.
- *Non-repudiation*: Set of functions that allow a federate to identify and prove, without any doubt, the authorship of a stream of information (or any changes it might have undergone).

C. Implementing InfoSec Services in Space Networks

Multiple mechanisms have been developed in order to guarantee InfoSec services in communication networks. For instance, reference [10] discusses several alternatives to provide InfoSec services to small near Earth satellites. Based on a similar categorization of InfoSec services, it identifies candidate algorithms that can be used to provide integrity, authentication and confidentiality services using well-known security mechanisms such as hashing functions, symmetric and asymmetric cryptography. In fact, real security systems typically combine these mechanisms in order to exploit their

respective advantages and drawbacks [11]. For instance, symmetric key cryptography is known for having better computational performance but requires complex mechanisms to securely exchange keys between two entities [12] [13]. Furthermore, it is not scalable since a unique private key has to be shared by any two entities, and it does not provide non-repudiation services since no key is uniquely assigned to a user [13].

Multiple authors have investigated the implementation of public-key cryptography in satellite communications systems. For instance, Cruickshank describes an efficient Public Key Infrastructure (PKI) security system for data and voice services over satellite communications [14]. He uses public key mechanisms to perform session set-up and authentication between a satellite network and a mobile user, while information exchange sessions are then ciphered using symmetric-key. Similarly, Ji et al. describe the "Dynamic Key Management model for Satellite Networks" which present a set of key management policies for satellite networks [15]. Their protocol is based on a hierarchical model with three different keys per satellite: A public key, a primary key and a session key. The PKI is used to update the satellite's primary key (long term symmetric key), which in turn is used to share session keys among pairs of satellites.

Key management is a popular topic in satellite network security research. The lack of permanent connectivity between a satellite and a certificate authority (or even between two satellites), introduces a set of challenges in the key management and certificate verification processes as compared to terrestrial key management. Several authors have proposed different key management schemes for two main types of space communication networks: multicast networks and delay-tolerant networks (DTN). In the multicast area, most of the proposed key management solutions are based on Logical Key Hierarchy schemes [16] [17] [18]. On the other hand, key management in DTN continues to be an active area of research (see references [19] [20] for instance) as both the CCSDS [21] and the DTN Research Group [22] acknowledge that it remains an open issue to be resolved.

Finally, several authors have raised concerns with respect to the use of PKI in satellite networks. In [12], the authors claim that PKI is too computationally expensive to be applied in satellite networks and propose a simplified symmetric authentication algorithm. Reference [23] assures that PKI requires a highly complex public-key management system. In addition, PKI cannot preserve

Table 1 InfoSec Threats and Attacks

Section	*Threat*	*InfoSec Service*	*Attack*		*Description*
			Name	*Type*	
III A	Identity theft	Authentication Confidentiality	Eavesdropping	Passive	A federate steals the identity of another federate
III A	Identity theft	Authentication Confidentiality	Impersonation	Active	A federate impersonates part of another federate system, thus gaining access to the rest of it
III A	Identity theft	Authentication Confidentiality Non-repudiation	Impersonation Denial of service	Active	A federate sends malicious messages through the FSS network under a false stolen identity
III B	Data theft	Authentication Confidentiality	Eavesdropping	Passive	A federate copies information content from another federate while relaying it
III C	Data corruption	Integrity Non-repudiation	Substituting	Active	A federate modifies the information stream that it is processing and/or relaying
III D	Selective dropping	Confidentiality Non-repudiation	Denial of service	Active	A federate does not relay an information stream, thus destroying the information
III E	Data replay	Integrity Non-repudiation	Replaying	Active	A federate records and then re-sends the same information multiple times
III F	Supplier disruption	Availability Non-repudiation	Denial of service Traffic Analysis	Active	A federate wastes supplier resources by submitting useless or malicious jobs

the user anonymity as identities are interchanged in the authentication process.

III. Security Threats and Attacks in the FSS Network

In this section we consider possible security threats and attacks relevant to a FSS. For the present discussion, a security *threat* is defined as "the expressed potential for the occurrence of a harmful event such as an attack" [24]. In turn, a security *attack* is defined as "the action taken against a target with the intention of doing harm". Attacks can be active, that is, they aim at modifying the system resources or affect their operation, or passive, i.e. their main purpose is to discover and use the information being sent through the system [25].

Table 1 summarizes the InfoSec threats identified for FSSs, along with the attacks federates could potentially utilize in order to exploit the system vulnerabilities. Similarly, sections III A to III F provide a detailed discussion for each of them.

A. Identity Theft

Identity theft would occur in FSSs if a supplier were to steal the identification and authentication credentials from a customer through an eavesdropping attack. Note that this attack is passive since

the supplier is not disrupting other services in the FSS network, nor is it hindering the operations of the customer spacecraft.

Risks associated with successful identity theft can be categorized based on the severity of the consequences they entail. At the lowest level, identity theft can be used as a prelude to monitor the customer system (spacecraft, ground segment and operations). In other words, the goal of the attack is to provide the sufficient authentication information to successfully identify and track the different parts of the customer system (e.g. tracking of the platform, monitoring the contacts and data generation). As an example, the attacker could use the authentication information to obtain access and monitor the customer mission operations center (MOC) and related data systems. The second level of severity encompasses all issues that arise when a malicious federate impersonates parts of the customer's system. In that case, the security attack becomes active since the operations of the customer platform could be potentially disrupted by, for instance, sending erroneous tracking and telemetry information to the ground systems. Finally, the third level of risk severity arises when identity theft is combined with identity concealment. In this case, the target of the security attack is no longer the customer spacecraft but the entire FSS. The malicious federate can execute other security attacks to any elements of the system while concealing its identity, thus making it impossible for the system to properly identify the origin of the security breach. Furthermore, it can also obtain services from the FSS network without being properly billed.

B. Data Theft

Data theft threats in the FSS network would be related to the ability of a federate to obtain proprietary information about another federate in the system. Compared to identity theft, the goal of data theft is not to supplant the customer federate but to steal valuable information from it. For instance, a supplier could steal data from a customer by eavesdropping on a normal FSS service. In that case, the FSS would negotiate the resource allocation between the supplier and the customer, and then the former would take advantage of it by copying the information he would be relaying and/or processing. Preventing this type of data theft attacks could be accomplished by (1) implementing authentication mechanisms that ensure malicious relays cannot claim stolen information as their's, (2) providing confidentiality services that hinder the eavesdropping process,

and (3) ensuring the anonymity of the FSS customer in the service negotiation phase so that the supplier cannot target a specific customer directly. Note that the latter is the only InfoSec service not listed for the data theft threat in table 1 since it is mostly related to the transactional view of the FSS security architecture.

C. Data Corruption

Data corruption intends to maliciously modify data sent or processed through the FSS network in order to render its content useless (and therefore valueless). Data corruption attacks are typically triggered by FSS suppliers and are categorized as active since they can potentially disrupt the operations of the customer spacecraft.

Data corruption can generally lead to two outcomes: Basic corruption entails introducing a sequence of bit errors in the data stream being relayed or processed. The goal of the attacker is solely to destroy the information content from the data stream. On the other hand, data corruption can also be combined with identity theft for impersonation purposes (see section III A). In this case, the attacker corrupts the data by generating valid messages that replace the original information content. This type of data corruption can therefore be undetectable by the customer's system since the semantics of the message are preserved during the corruption process.

Data corruption can be prevented in FSSs by ensuring integrity and non-repudiation services. The former provides an efficient mechanism to detect and deter undesired modifications on a data stream, while the latter ensures that, if data corruption occurs, the responsible party will be properly identified.

D. Selective Dropping

Selective dropping can be considered a specific instance of data corruption in which the FSS supplier intentionally denies service to a customer after it has been successfully negotiated. In other words, data sent to the supplier for relaying or processing purposes is discarded and therefore lost during the execution of the transaction.

Selective dropping threats arise mostly due to possible competition among agents in the FSS. For instance, consider two competing Earth imagery services that adhere to the FSS network in

order to downlink information to the ground. Then, it is conceivable that both companies, despite their willingness to cooperate in the FSS, do not want to provide services to one another.

Mitigation of selective dropping vulnerabilities could be achieved through both anonymity and non-repudiation mechanisms. The former would render the exchanged data stream confidential to the supplier who, as a result, would not be able to detect a potential conflict of interest in the data he would be serving. Alternatively, the latter would ensure that Denial of Service (DoS) attacks could be traced back to the appropriate FSS party. Finally, selective dropping from a customer to a supplier could also possible by inserting malware into the exchanged information. These attacks are analogous to phishing for data theft purposes and are therefore considered part of the physical FSS security architecture.

E. Data Replay

Data replay threats would potentially originate at malicious federates that would record and replay messages through the FSS network after the original message had reached its destination. They are categorized as active attacks since they can easily disrupt the operations of a spacecraft or result in total loss. For instance, assume two satellites are communicating through the FSS network. The supplier replays a message received without knowing its content, thus inadvertently triggering two thrust activations in the customer spacecraft and consequently placing it in the wrong orbit.

Reducing data replay threats could be achieved by ensuring proper synchronization of messages through the FSS network. This, in turn, would require integrity and non-repudiation services from the security architecture. Integrity would be used to ensure that a malicious federate could not compromise the time stamp of a message in order to disrupt the service synchronicity. On the other hand, non-repudiation would ensure that the source of replay messages could be successfully identified and removed from the FSS network.

F. Supplier Disruption

Supplier disruption consists of preventing legitimate users from accessing resources they might access under normal circumstances. The US Computer Emergency Readiness Team classifies DoS attacks in three categories [26]:

1. Consumption of scarce, limited or non-renewable resources. Attacks in this category are those that attempt to exhaust network connectivity, CPU time, available bandwidth or disk space. A typical example of an attack in this category is a SYN attack [27], [28].

2. Destruction or alteration of configuration information within an insecure computer. As the CPU time of a federate might be shared with external users, an attacker might inject malicious code to modify configuration parameters of the supplier federate. These types of attack are closely related with FSS physical view security and their analysis and mitigation remains out of the scope of this paper.

3. Physical destruction or alteration of network components. This threat belongs to the physical view of the FSS security architecture and will therefore not be considered in this paper.

Supplier disruption could be prevented by guaranteeing availability and non-repudiation services in the FSS network. Availability services would provide mechanisms to ensure that FSS suppliers can monitor and reject jobs should they detect the presence of a DoS attack. In turn, non-repudiation would allow the system to identify the source of a DoS.

IV. Security Architecture for the FSS Network

A. InfoSec Service Functionality

As previously mentioned, the FSS is highly heterogeneous and opportunistic. This fact allows the system to efficiently share resources among its federates, but it also facilitates security breaches in the different interactions between participants. This dichotomy is also present in current ground-based systems like the Internet and therefore we begin our analysis of the FSS InfoSec architecture by reviewing the mechanisms that have already been developed and implemented. Fig. 3 presents a high level OPM-based [29] description of a reference security architecture that combines well-known security mechanisms such as hashing, private and public key cryptography [30], coding and spread-spectrum techniques [31].

On the left-hand side of Fig. 3, InfoSec attributes are provided to a message M through 4 main processing steps: hashing, signing, encrypting and encoding. These processes represent the system's value delivery functions since they modify the state of the original message and progressively

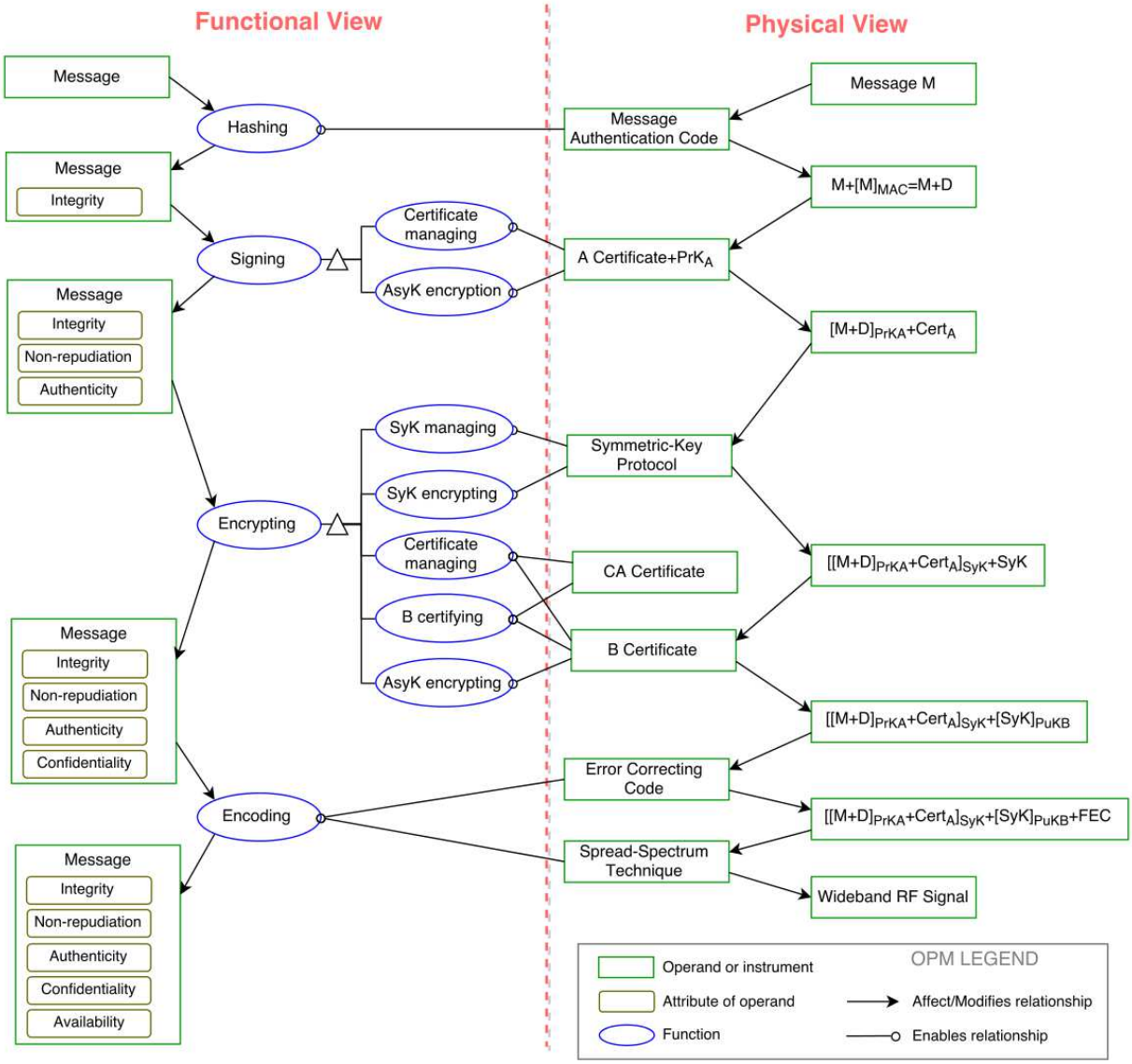


Fig. 3 Reference security architecture

provide the necessary InfoSec attributes needed to secure it. In contrast, the right-hand side of Fig. 3 provides a physical view for the reference architecture. It presents an overview of the different mechanisms necessary to secure the message, as well as a basic protocol narration (see reference [32]) of how the message is progressively modified. For instance, in order to ensure that a message has not been modified (i.e. it has integrity) the transmitting node A would compute a hash D by applying a hashing function $[\cdot]_{MAC}$ to the original message M. Then, he would transmit the original message and append the hash bits at the end of it. In turn, the receiver B could check that the message has not been modified by recomputing D at his end and verifying that it equals the received

hash. Similarly, if authentication and non-repudiation services are required, the transmitting node would take the message plus hash bits and apply his private key PrK_A. Then, he would transmit the encrypted message plus his public key $Cert_A$, all encrypted using the receiver public key PuK_B. Unfortunately, the encryption process in the PKI infrastructure is computationally expensive and inefficient for large file transfers. Instead, it is common to use a symmetric key SyK to encrypt most of the message, and just sign it by encrypting the SyK with the receiver's public key PuK_B. Note that this improved scheme is notionally depicted in Fig. 3.

Two important conclusions can be reached based on the previous discussion: First, providing all security attributes for an FSS interaction is in general possible if PKI is available for the system. However, PKI might not be necessary for certain FSS architectures in which federates only interact with trusted parties (e.g. centralized FSS). Second, providing all InfoSec services for every interaction in the FSS can result in large computational and bandwidth inefficiencies, which in turn hinders the ability of the system to satisfy the stringent margins that missions grant for FSS service execution.

B. End-to-end InfoSec in the FSS Network

We define end-to-end InfoSec services as the set of mechanisms that allow secure transmission and processing of a message (or part of a message) that is only relevant to the original sender and final addressee. For clarity and simplicity purposes, assume that the message exchanged between nodes F_1 and F_3 in Fig. 4 can be decomposed into a header H and a payload M. The payload M is constant and only relevant to F_1 and F_3, while the header H is progressively updated as the message is moved across the FSS network. Then, we explicitly divide the FSS InfoSec services into end-to-end and hop-to-hop services. The former refer to mechanisms available to protect the payload M, while the latter refer to services that secure the header H.

To further simplify the problem, we first assume that end-to-end InfoSec services will generally be provided by the original sending federate and final receiving federate independently from the FSS network. For instance, consider two nodes interacting through the FSS that belong to the same organization. Then, it is reasonable to assume that they share a set of pre-programmed symmetric keys that are put on the on-orbit platform prior to its launch. These symmetric keys could then

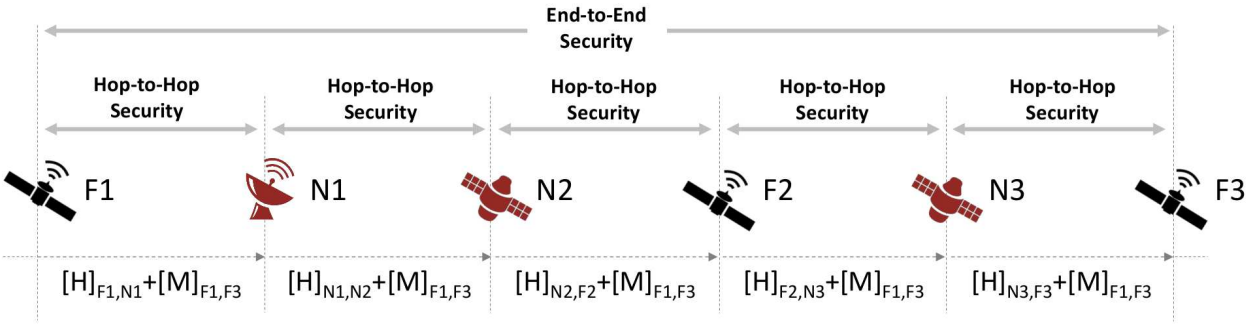


Fig. 4 End-to-end Information Security

be used to secure any messages' payloads by encrypting them prior to a transmission. On the other hand, we can also envision a situation in which two federates of different organization want to cooperate once they have been put in orbit. In that case, the exchange of a common symmetric key between both organizations could occur over a direct Earth-based communication channel (e.g. the Internet) rather the FSS network. This key would then be uploaded to the two federates independently by each organization using their respective symmetric keys. Therefore, once again the transmission between F_1 and F_3 would have its payload encrypted from the start, thus reducing the amount of data that needs to be secured by the FSS network to just the message headers H.

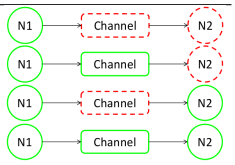
C. Modeling the FSS InfoSec Architecture

Based on the previous discussion, it is now clear that the FSS InfoSec architecture has to provide services to secure the headers of messages exchanged in the network. In this section we present the Interaction State Model and its extension to the Interaction State Machine. They provide a functional model to understand the architecture of the FSS InfoSec services for these headers in any interaction between two federates. They also serve as a foundation for the discussion of InfoSec services across the entire FSS.

1. The Interaction State Model

The goal of the hop-to-hop FSS InfoSec architecture is to provide the required mechanisms to ensure a secure environment for interaction between FSS participants, while guaranteeing that no extra redundancy or inefficiency is introduced in the system. As we previously mentioned, resources

Table 2 Interaction Security States

Node 1	*Channel*	*Node 2*	*Acronym*	*Symbol*
Trusted	Not Trusted	Not Trusted	TNN	
Trusted	Trusted	Not Trusted	TTN	
Trusted	Not Trusted	Trusted	TNT	
Trusted	Trusted	Trusted	TTT	

in space are limited and therefore efficiency becomes a critical requirement for the FSS InfoSec architecture. With that premise, we initiate the discussion by proposing a state-based model of FSSs in which participants engage in individual peer-to-peer interactions through a channel. Interactions are supported by a physical communication channel and are characterized by a security state, which in turn depends on the trustworthiness of the two FSS participants and the channel between them. Note that in this definition we neither prescribe the type of participant (federate or negotiator), nor the type of channel used for interacting (e.g. wireless, wired), thus allowing us to capture the wide heterogeneity present in FSSs.

Assuming that participant N_1 is a legitimate FSS node, four canonical interaction security states are possible (see table 2). In the first one, the first node considers the channel and its peer to be untrustworthy, while the second one assumes only the peer to be unreliable. This is plausible in the FSS context since satellite-to-satellite connections through directional RF or optical communications might be hard to physically intercept or interrupt, thus hindering the possibility of eavesdropping or man-in-the-middle attacks. The third option assumes that the peer node is trustworthy, while the channel is not. This is representative of a situation in which a federate has authenticated a ground-based negotiator and can therefore view its peer node as a trusted authority, but the channel remains insecure since the space-to-ground signal is relatively easy to intercept. Finally, the fourth option models a situation in which both nodes and the channel are reliable.

Based on this categorization, we can now define the FSS hop-to-hop InfoSec architecture by identifying the security functions that participants have to implement in order to reach a security state for any given interaction. Since two types of interactions are possible, tables 3 and 4 summarize the results for negotiations and transactions. Note that four of the five functions listed are the primary value delivery functions from the reference security architecture in Fig. 3. That being

Table 3 Negotiation Security Architecture

FSS Architecture	*Interaction State*	*Hashing*	*Signing*	*Certifying*	*Encrypting*	*Encoding*
Distributed		✓	✓	✓	✗	✓
Distributed		✓	✓	✓	✗	✗
Negotiated or centralized		✓	✗	✗	✗	✓
Negotiated or centralized		✗	✗	✗	✗	✗

Table 4 Transaction Security Architecture

FSS Architecture	*Interaction state*	*Hashing*	*Signing*	*Certifying*	*Encrypting*	*Encoding*
Negotiated or distributed		✓	✓	✓	✓	✓
Negotiated or distributed		✓	✓	✓	✓	✗
Centralized		✓	✗	✗	✓	✓
Centralized		✗	✗	✗	✗	✗

said, in accordance with section IV B, encrypting only refers to functionality provided to ensure that *only* N_1 and N_2 can comprehend the headers, while end-to-end encryption on the message is assumed to be implemented by the original transmitting node. Furthermore, certifying is added as a primary function for the system's architecture. It encompasses all functions that a node has to perform in order to obtain the certificate of his interaction's peer and validate his identity. Particularly, it includes the definition of the proper certificate authorities (CA), as well as the distribution mechanisms required in order to ensure that a participant will have (or will be able to obtain) a peer's certificate at any point in time. While these functions do not directly modify the message, they are known to be challenging in space communication networks [21] and their impact should therefore be minimized.

We now turn our attention to table 3 which specifies the security functions required for obtaining a secure negotiation between two FSS participants. Since the system is assumed to be an auction-based marketplace, any participant can freely monitor the status of the bid-ask auction. As a result, there is no need for encryption. Similarly, in a centralized or negotiated FSS architecture the interaction is carried out between two nodes that trust each other, a federate and a negotiator (it is assumed that both federates exchange symmetric keys with the FSS central authority upon adhesion to the FSS and they are broadcasted to all negotiators). Therefore, no real-time signature and certification processes are necessary. On the other hand, hashing is included if either the channel

or peer node are untrustworthy. The former protects the interaction against man-in-the-middle attacks, while the latter ensures that the peer cannot freely modify a customer's request or supplier's demand, thus maliciously altering the bid-ask matching process. Finally, encoding functions protect the physical channel against the presence of attenuation and/or intentional jamming.

Next, we analyze table 4 which describes the security mechanisms present during the FSS transacting interactions. Note that, in this case, all interaction states with untrustworthy elements require encryption in order to shield the information headers against traffic analysis attacks. As previously mentioned, the preferred encryption mechanism will be symmetric-key based. Nevertheless, cases in which N_2 is untrustworthy will require negotiating a temporary hop-unique symmetric key which will in turn require an initial public-key secured exchange in order to guarantee non-repudiation services. Alternatively, a negotiated FSS architecture could also embed the symmetric key establishment between two untrustworthy nodes as part of the negotiation phase. This is advantageous since the negotiator is a trusted authority for the two interacting federates N_1 and N_2. However, it also increases the security mechanisms required during a negotiation interaction and works only if the three parties are in view of one another at the same time. Finally, since FSSs mix space and ground assets, we advocate security mechanisms that can be implemented on an isolated peer-to-peer (only N_1 and N_2) environment in order to facilitate delay-tolerant security.

2. The Interaction State Machine

Having described the different states that can occur in an FSS negotiation and transaction, we extend the State Interaction Model for FSS InfoSec architectures by defining the types of transitions that allow FSS participants to regulate the InfoSec services they obtain from the system. This extension results in the Interaction State Machine, a transition diagram that is specific for each type of interaction in FSSs and clearly specifies the InfoSec services that an FSS node can provision given his and his peer's node security architecture.

Fig. 5 presents a pictorial view of the InfoSec State Machine for a transacting interaction between two FSS nodes (note that based on tables 3 and tables 4, the only difference between the InfoSec State machine for transactions and negotiation interactions is the added presence of the

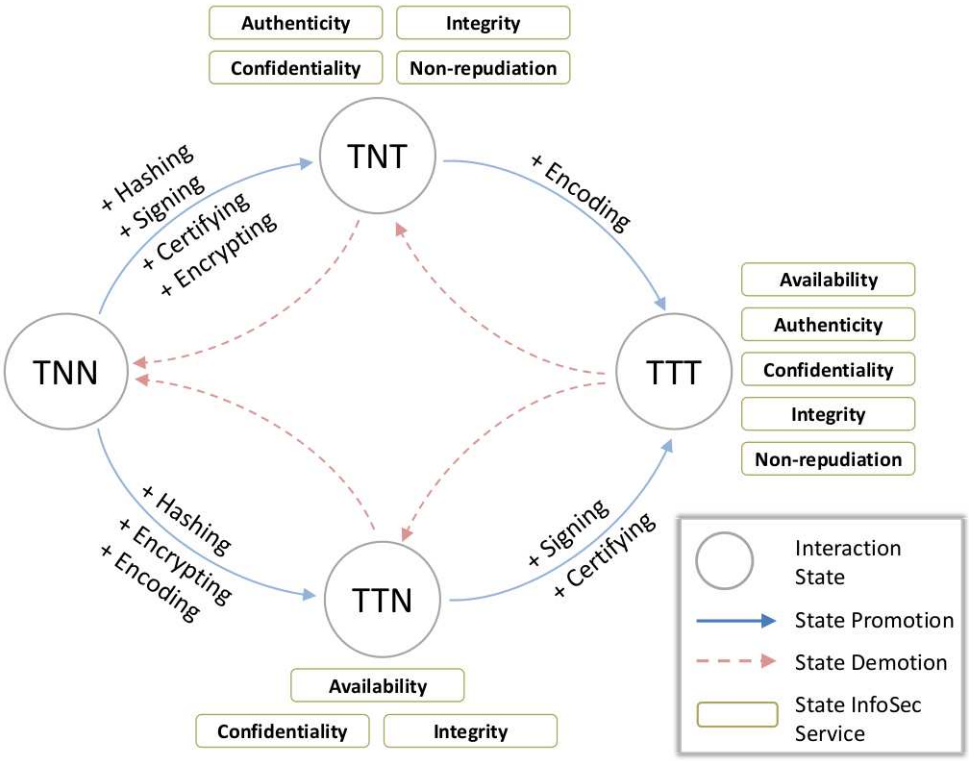


Fig. 5 InfoSec State Machine

encrypting functionality in the former). Two types of transitions have been defined, promotions and demotions. An InfoSec promotion occurs when the difference between the number of trustworthy elements in the final and original state is positive. In other words, the interaction becomes "more secure". A demotion captures the opposite transition in which the interaction becomes more susceptible to security attacks.

Assume two nodes N_1 and N_2 in the FSS network have been interacting uninterruptedly during the last ten minutes. Their interaction is in security state TTN, that is, they both have agreed upon a hashing mechanism and symmetric key (exchange of symmetric keys over insecure channels can be done through mechanisms that do not require public-key infrastructure - e.g. Diffie-Hellman key exchange algorithm [33] [34]), as well as the proper coding and modulation techniques to protect the physical communication channel. Assume also that node N_1 wants to now transmit highly sensitive information such as its own telemetry. He can promote the security state of this interaction by sending a signed message to N_2 in which he requests his peer's identity in the form of a public key. He can then use the appropriate certificate authority to authenticate N_2 and obtain a TTT

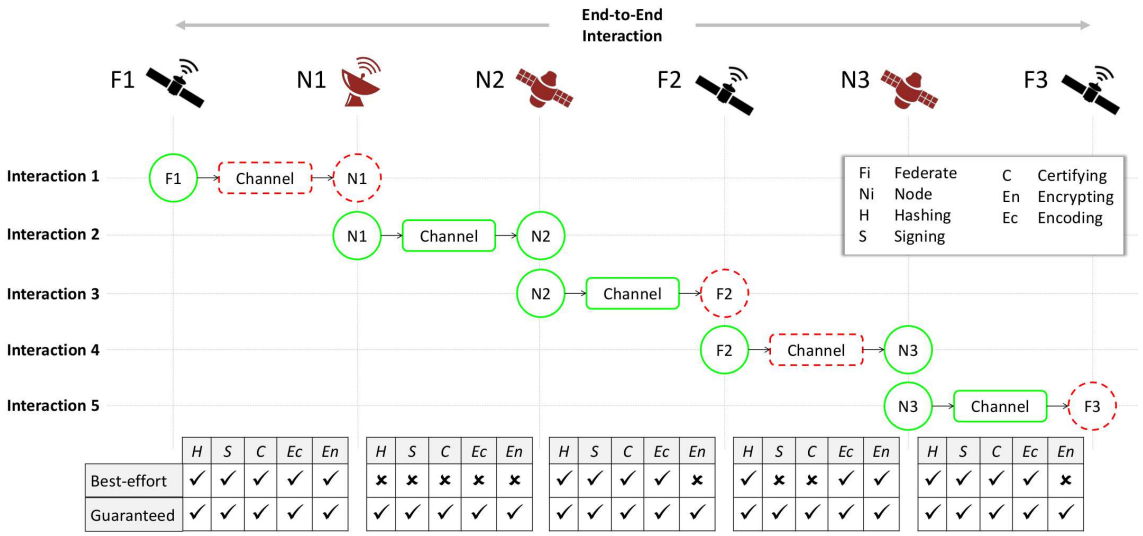


Fig. 6 Multi-Hop FSS Security Architecture

interaction. This results in extra communication and processing steps for both N_1 and N_2, but it also ensures authenticity and non-repudiation services in their interaction.

D. Extending the Interaction State Model to Multi-Hop Service Execution

Based on the definitions of the Interaction State Model, the goal of this section is to analyze how the model can be used to understand the FSS InfoSec architecture across the entire system. Although the discussion and examples herein presented refer mostly to transactions between FSS participants, analogous conclusions can be drawn by other types of interactions such as negotiations.

1. An Introductory Example

Fig. 6 provides a graphical representation of the security architecture for the multi-hop interaction between federates F_1 and F_3. The transaction is executed through a combination of ground or space-based federates and negotiators that are connected to one another through interactions at different security states. Note that these states are exclusive to each interaction and are therefore defined locally (on a hop-to-hop basis). In other words, interaction 5 is in state TTN because negotiator N_3 *thinks* that the channel is trustworthy but F_3 is not. This assessment is completely independent from F_1's opinion on F_3 (they belong to the same organization and therefore trust each other).

Based on Fig. 6, assume an FSS centralized architecture in which F_1 wants to transmit information to F_3. After the negotiation phase ends, it is granted FSS services through negotiator N_1, which will then relay the information to another FSS node (N_2 in our case). This node will then repeat the process until the data reaches its destination. At the start of the transacting interaction F_1 does not know the nodes that will transmit his information, nor the types of interactions that it will traverse. However, at a local level, it is cognizant of potential security issues when transmitting information to N_1 (it has never interacted with N_1 but it knows that N_1 is a ground station). As a result, it decides to initialize his interaction with N_1 in state TNN. Assume that F_1's information is highly sensitive and requires all InfoSec services. Therefore, it progressively executes security functionality to promote his interaction with N_1 to state TTT and then starts the transmission of the message. Once N_1 receives the information, it decides to route it through N_2. Once again, N_1 has to check the security state for his connection with N_2 and compare them with the security requirements of the information it has received. If his interaction with N_2 does not provide sufficient security mechanisms, it must then promote his interaction state in order to provide the quality of service negotiated with N_1. Luckily, in this case both N_1 and N_2 are negotiators directly controlled by the FSS central authority. They have successfully communicated several times and therefore know and trust each other. As a result, N_1 considers his interaction with N_2 to be already in state TTT and information is sent directly to the third node. This decision maximizes the efficiency of the transaction between N_1 and N_2, but also exposes F_1's information to potential security attacks that N_1 has not foreseen. Finally, the process to secure the information at each interaction is repeated for $N_2 - F_2$, $F_2 - N_3$ and $N_3 - F_3$ until F_1's data reaches the end destination.

2. Quality of Service for FSS Security Services

Section IV D 1 has demonstrated how to apply the Interaction State Model to understand the security architecture of a multi-hop transacting interaction. The fundamental premise in the presented mechanism is that FSS nodes make security decisions at the local level in order to provision multi-hop InfoSec services. This maximizes the efficiency of the system by avoiding redundancy in the security functions that each node implements. Nevertheless, it can also result in violating the

security requirements of an information stream from the point of view of the end-to-end nodes.

In the previous example N_1 sends F_1's information without any security guarantees because he locally considers his interaction with N_2 to be fully secure. Nevertheless, F_1 can have a different view on the state of $N_1 - N_2$ interaction, he might not trust N_1's judgment or he might simply not want the risk of having a third party making decisions about the security of his information. In any case, it is clearly desirable for the FSS security architecture to consider mechanisms that ensure the delivery of a given security level for the execution of a multi-hop interaction.

To that end, we distinguish between *best-effort* and *guaranteed* FSS security services. The former captures scenarios analogous to the example from section IV D 1, i.e. each node executes certain security functionality depending on the current state of their interaction with another node. In contrast, guaranteed FSS security services enforce the set of functionality that each node in the network has to implement in order to be able to transmit or process information (see Fig. 6). Note that this simple definition can be extended by defining different levels of InfoSec quality of service for a multi-hop FSS interaction. As an example, a mission might require all nodes to implement encryption mechanism while the certification process should not be repeated if two interaction peers have already implemented it.

Finally, we discuss possible implications of security QoS in FSS service negotiation and execution. On one hand, it has already been stated that requesting security QoS for a multi-hop interaction might result in unnecessary burden and inefficiency to the individual interactions executed by the different federates. Therefore, it seems natural to envision that FSSs will provide billing mechanisms that fairly compensate nodes willing to satisfy more stringent QoS requirements. In other words, the ability to market and provision multiple security QoS levels for a multi-hop interaction enriches the FSS auction-based negotiation mechanism by allowing participants to value both the execution and the perceived quality of an interaction. On the other hand, security QoS can also be introduced as an important factor in the routing mechanisms for the FSS network. In that sense, optimal routes will not only be related to classical measures such as number of hops or latency [35], but also to the ability of an interaction to easily satisfy the security requirements of an information stream. For instance, routes requiring fewer interaction promotions should be favored

as they maximize the efficiency of the system and therefore increase its overall capacity.

E. Example: Negotiated FSS in the Earth Domain

This section demonstrates the ability of the Interaction State Model and Interaction State Machine to structure and analyze the provision of InfoSec services in a federated satellite system. Consider a constellation of low Earth orbit small satellites that collect data on the composition and evolution of clouds. It belongs to a privately owned company headquartered (HQ) in California that sells the data to weather and climate scientists that utilize it to feed their prediction models. The system was initially designed so that each satellite would downlink the acquired data once per orbit through a proprietary ground station located in western part of the United States. Nevertheless, once the constellation has been deployed the system customers indicate that it would be highly valuable to have images available every 30 minutes (or three times per orbit), especially when atmospheric conditions in a region of interest are highly variable. Given that these only occur infrequently, the company management considers building new ground stations across the world as an unfeasible solution and would like to lease space communication services on demand.

For this example, assume initially that a negotiated FSS network already exists. It mixes both low Earth orbit (LEO) and geosynchronous (GEO) spacecraft with their respective ground systems. All of them are compatible from a communications perspective because they were originally designed using space communication standards such as the ones proposed by the CCSDS. Furthermore, assume that the implemented architecture for the FSS network is negotiated such that all "negotiators" are ground stations scattered across the world connected to a network operations center (NOC) that assigns resources in real-time. Then, we would like to understand if InfoSec services could be provided for the return of weather data using the opportunistic FSS network such that the latency constraint of 30 minutes is met and the security of the images collected is not compromised.

Fig. 7 depicts the different types of interactions that will occur in the problem at hand. On the left side, we plot the negotiation process by which F_1 contacts the FSS ground station F_2, who then relays the request to the NOC, which finally allocates satellite N_1 to support F_1 during the next part of this orbit. In turn, the right side plots the interactions present in the network when F_1 and N_1 are executing the transaction planned during the negotiation phase. Note that in this phase

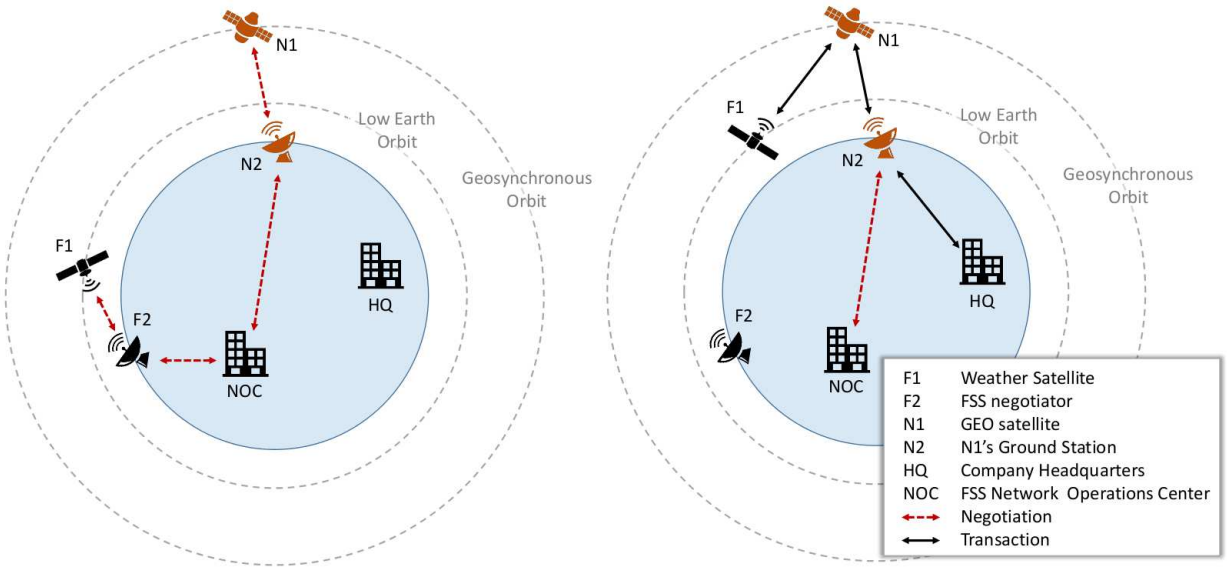


Fig. 7 Negotiated FSS for Weather Satellite

Table 5 Initial Security States

Phase	*Interaction*	*Interaction State*	*Rationale*
Negotiation	$F_1 - F_2$	TNT	Unsecure space-to-ground (SGL) channel, F_2 is an FSS node
Negotiation	$F_2 - NOC$	TTT	Secure internal FSS network
Negotiation	$N_1 - N_2$	TNT	Unsecure SGL channel, N_1 and N_2 belong to the same organization
Negotiation	$N_2 - NOC$	TNT	Unsecure terrestrial channel, NOC is FSS node
Transaction	$F_1 - N_1$	TNN	Unsecure space link, other node is unknown
Transaction	$N_1 - N_2$	TNT	Unsecure SGL, N_1 and N_2 belong to the same organization
Transaction	$N_2 - HQ$	TNN	Unsecure terrestrial link, other node is unknown
Transaction	$N_2 - NOC$	TNT	Unsecure terrestrial channel, NOC is FSS node

there is still a negotiation interaction between N_2 and the NOC that carries all billing information required so that FSS management can properly determine the amount of resources used by F_1 while utilizing the system.

Next, we exemplify the use of the Interaction State Model and the Interaction State Machine to analyze the required InfoSec architecture for the problem at hand. We start by using tables 3 and 4 of the Interaction State Model to assess the initial security states for the different interactions between federates. The results are summarized in table 5, which has been constructed keeping in mind that the current FSS has a negotiated architecture. We observe that only one interaction is in the TTT state from the beginning, thus indicating that InfoSec services will be required in order to ensure secure delivery of the weather data.

Based on this initial categorization of the possible FSS network interactions, we now study the set of functionality that the FSS InfoSec architecture will have to provide in order to mitigate security threats. To that end, we utilize the InfoSec State Machine to identify which interactions have to be promoted to a higher security state. In turn, this exercise provides qualitative insights into the amount of computational and communication burden incurred while securing the weather data.

Assume first that weather data requires only a best-effort type of service by which it is only required that no data is lost due to an untrustworthy FSS node. Then, only two interactions ($F_1 - N_1$ and $N_2 - HQ$) will have to promote their state to successfully provide the desired level of InfoSec services. Since both of them occur in the transaction phase, we can directly use Fig. 5 to determine the set of security functions that will have to be provided, namely hashing, signing, certifying and encrypting. In turn, no other interactions in the system will require additional security functions, therefore potentially saving a large amount of computational and communication resources. On the other hand, consider now that the weather data is sensitive enough to require a guaranteed service by which two nodes in the network can only transact if they are in a TTT state. Then, all interactions in the transaction phase will have to be upgraded, thus requiring all nodes to perform all security functionality. Note that this requirement largely penalizes the performance of the $N_1 - N_2$ interaction because the GEO satellite is now forced to check the identity of its own ground station before relaying F_1's information. The same inefficiency would also be observed in the $F_2 - NOC$ negotiation interaction, albeit its consequences in terms of computation and communication burden are significantly lower since it happens over a terrestrial line.

V. Conclusions

A. Summary

This paper studies the architecture of information security services in the context of federated satellite systems. Initially, the needs of the system vis-a-vis the InfoSec architecture are assessed based on a threat analysis. It details the different vulnerabilities that FSSs have to address in order to provide a trustworthy environment in which resources between FSS participants can be securely exchanged. Mitigation of these vulnerabilities is achieved by provisioning five types of InfoSec

services: Authentication, availability, confidentiality, integrity and non-repudiation. In particular, we note that, unlike other satellite communication systems, non-repudiation is a fundamental service in the FSS environment due to the heterogeneity and transaction-based nature of the system.

Based on the five types of InfoSec services, we identify five canonical functions that an FSS network has to implement in order to deliver them: Hashing, signing, certifying, encrypting and encoding. Next, we map these five functions into the two types of interactions present on FSSs, namely negotiations and transactions, and we identify the different security states that can occur when two federates exchange information: TNN, TTN, TNT, TTT. Finally, we identify which security functions have to be executed to ensure that an interaction between federates is in a given state under the different FSS architectures identified in the literature. As expected, negotiated and specially centralized FSS architectures require less and simpler security functionality in order to guarantee that trustworthiness in a peer-to-peer interaction.

Once the mapping between security functions and security states has been completed, we introduce the InfoSec State Machine that defines how federates can upgrade or degrade the security state of any interaction. We argue that this state machine can be used to understand both the level of security performance provided by the network at a local scale, as well as understanding the level of computation and communication burden incurred while securing the information. Indeed, obtaining a fully trustworthy channel with another federate requires a node to incur both processing and communication overhead, both of which are undesirable in the context of a space-based asset with limited power resources. In that sense, we provide several examples that demonstrate how information can be securely transmitted through the system by allowing each federate in the information path to make local decisions on which security functions to implement given his beliefs on the security state for the next hop. This realization leads us to differentiate between best-effort and guaranteed security services. The former allows federates to independently decide which security functionality to implement, while the latter is decided by the node where the information originates and forces all other nodes to provide a pre-specified set of security functionality. Finally, we argue that providing best-effort or guaranteed security services should be be analyzed as a parameter that affects the ask-bid negotiation process by which FSS nodes regulate the exchange of resources

through the network.

B. Future Work and Open Issues

While this paper provides a generic approach to architecting InfoSec services for an FSS network, several areas of future work can be envisioned. On the one hand, this paper focuses its attention on the informational view of the FSS security architecture while recognizing that both the physical and transactional views should be analyzed. Therefore, a similar study could be performed for these complimentary views assuming information security to be a fundamental service provided by the FSS infrastructure. On the other hand, further research in InfoSec security is also required in order to fully define the architecture and design of the FSS security system. Three main areas of research are of particular interest: First, anonymity is a desirable service in marketable environments such as FSSs, since it avoids illicit disruptions due to competing federates. Second, the PKI assumed for non-repudiation purposes has to be adaptable to a delay-tolerant environment. As mentioned in the literature review, this is currently an active area of research in the community and therefore has no unique solution (e.g. [19] [20]). Finally, this paper does not provide any recommendations on which specific security mechanisms should be implemented in order to execute the different security functions from figure 3. Therefore, performance of different security mechanisms (e.g. RSA [36], Kerberos [37]) should be analyzed to understand the inefficiencies that arise when providing different quality of service levels in a secure end-to-end FSS interaction.

References

[1] Golkar, A., "Federated Satellite Systems: A Case Study on Sustainability Enhancement of Space Exploration Systems Architectures," *64th International Astronautical Congress, no. IAC-13-D3*, Vol. 4, 2013.

[2] Golkar, A., "Design Margin Utilization in Commercial Satellite Cloud Computing Systems," *65th International Astronautical Congress, no. IAC-14-D3*, 2014.

[3] Grogan, P. T., Golkar, A., Shirasaka, S., and de Weck, O. L., "Multi-stakeholder Interactive Simulation for Federated Satellite Systems," *Aerospace Conference, 2014 IEEE*, IEEE, 2014, pp. 1–15.

[4] Golkar, A. and Cruz, I. L., "The Federated Satellite Systems Paradigm: Concept and Business Case Evaluation," *Acta Astronautica*, Vol. 111, 2015, pp. 230–248.

[5] Symington, S., Farrell, S., Weiss, H., and Lovell, P., "Bundle Security Protocol Specification," Tech. rep., Internet Engineering Task Force, 2011.

[6] CCSDS, "Space Data Link Security Protocol," Tech. rep., The Consultative Committee for Space Data Systems, February 2012.

[7] CCSDS, "Security Architecture for Space Data Systems," Tech. rep., The Consultative Committee for Space Data Systems, November 2012.

[8] Jarupunphol, P. and Mitchell, C. J., "Measuring 3-D Secure and 3D SET Against E-commerce End-user Requirements," *Proceedings of the 8th Collaborative Electronic Commerce Technology and Research Conference*, Citeseer, 2003, pp. 51–64.

[9] Oxford Dictionary, June 2015.

[10] Vladimirova, T., Banu, R., and Sweeting, M., "On-board Security Services in Small Satellites," *Military and Aerospace Programmable Logic Device Proceedings*, 2005.

[11] Gupta, V., Gupta, S., Chang, S., and Stebila, D., "Performance Analysis of Elliptic Curve Cryptography for SSL," *Proceedings of the 1st ACM workshop on Wireless security*, ACM, 2002, pp. 87–94.

[12] Chang, Y.-F. and Chang, C.-C., "An Efficient Authentication Protocol for Mobile Satellite Communication Systems," *ACM SIGOPS Operating Systems Review*, Vol. 39, No. 1, 2005, pp. 70–84.

[13] Stamp, M., *Information Security: Principles and Practice*, John Wiley & Sons, 2011.

[14] Cruickshank, H., "A Security System for Satellite Networks," *Satellite Systems for Mobile Communications and Navigation, 1996., Fifth International Conference on*, IET, 1996, pp. 187–190.

[15] Ji, Y., Ma, H., and Zheng, G., "Analysis and Design on Key Updating Policies for Satellite Networks," *International Journal of Computers, Communications and Control*, Vol. 3, No. 4, 2008, pp. 343–352.

[16] Howarth, M. P., Iyengar, S., Sun, Z., and Cruickshank, H., "Dynamics of Key Management in Secure Satellite Multicast," *Selected Areas in Communications, IEEE Journal on*, Vol. 22, No. 2, 2004, pp. 308–319.

[17] Arslan, M. G. and Alagoz, F., "Security Issues and Performance Study of Key Management Techniques over Satellite Links," *Computer-Aided Modeling, Analysis and Design of Communication Links and Networks, 2006 11th International Workshop on*, IEEE, 2006, pp. 122–128.

[18] Wallner, D., Harder, E., and Agee, R., "Key Management for Multicast: Issues and Architectures," Tech. rep., RFC 2627, 1999.

[19] Zhou, J., Song, M., Song, J., Zhou, X.-w., and Sun, L., "Autonomic Group Key Management in Deep Space DTN," *Wireless personal communications*, Vol. 77, No. 1, 2014, pp. 269–287.

[20] Farrell, S. and Cahill, V., "Security Considerations in Space and Delay Tolerant Networks," *Space Mission Challenges for Information Technology, 2006. SMC-IT 2006. Second IEEE International Conference on*, IEEE, 2006, pp. 8–pp.

[21] Book, G., "Space Missions Key Management Concept," Tech. rep., The Consultative Committee for Space Data Systems, 2011.

[22] Farrell, S., Symington, S., Weiss, H., and Lovell, P., "Delay-tolerant Networking Security Overview," Tech. rep., Internet Research Task Force, 2009.

[23] Chen, T.-H., Lee, W.-B., and Chen, H.-B., "A Self-verification Authentication Mechanism for Mobile Satellite Communication systems," *Computers & Electrical Engineering*, Vol. 35, No. 1, 2009, pp. 41–48.

[24] Gregory, P., *CISSP Guide to Security Essentials*, Cengage Learning, 2009.

[25] Shirey, R. W., "Internet Security Glossary," Tech. rep., Internet Engineering Task Force, 2000.

[26] CERT Coordination Center, "Denial of Service Attacks," Tech. rep., Software Engineering Institute, 2001.

[27] CERT Coordination Center, "CERT Advisory CA-1996-21 TCP SYN Flooding and IP Spoofing Attacks," Tech. rep., Software Engineering Institute, 1996.

[28] Meadows, C., "A Formal Framework and Evaluation Method for Network Denial of Service," *Computer Security Foundations Workshop, 1999. Proceedings of the 12th IEEE*, IEEE, 1999, pp. 4–13.

[29] Soderborg, N. R., Crawley, E. F., and Dori, D., "System Function and Architecture: OPM-based Definitions and Operational Templates," *Communications of the ACM*, Vol. 46, No. 10, 2003, pp. 67–72.

[30] Inc., C. G., "Public Key Encryption and Digital Signature: How do they work?" Tech. rep., CGI Group Inc., 2004.

[31] Jo, K. Y., *Satellite Communications Network Design and Analysis*, Artech house, 2011.

[32] Briais, S. and Nestmann, U., "A Formal Semantics for Protocol Narrations," *Theoretical Computer Science*, Vol. 389, No. 3, 2007, pp. 484–511.

[33] Merkle, R. C., "Secure Communications over Insecure Channels," *Communications of the ACM*, Vol. 21, No. 4, 1978, pp. 294–299.

[34] Diffie, W. and Hellman, M. E., "New Directions in Cryptography," *Information Theory, IEEE Transactions on*, Vol. 22, No. 6, 1976, pp. 644–654.

[35] Lluch, I., Grogan, P. T., Pica, U., and Golkar, A., "Simulating a Proactive Ad-hoc Network Protocol for Federated Satellite Systems," *Aerospace Conference, 2015 IEEE*, IEEE, 2015, pp. 1–16.

[36] Rivest, R. L., Shamir, A., and Adleman, L., "A Method for Obtaining Digital Signatures and Public-key Cryptosystems," *Communications of the ACM*, Vol. 21, No. 2, 1978, pp. 120–126.

[37] Neuman, B. C. and Ts' O, T., "Kerberos: An authentication Service for Computer Networks," *Communications Magazine, IEEE*, Vol. 32, No. 9, 1994, pp. 33–38.