# Security Requirements and Solutions for Integrated Satellite-Terrestrial Information-Centric Networks

Nikos Fotiou, Yiannis Thomas, Vasilios A. Siris and George C. Polyzos
Mobile Multimedia Laboratory, Department of Informatics
School of Information Sciences and Technology
Athens University of Economics and Business
Patision 76, 104 34, Athens, Greece
Email:{fotiou, thomasi, vsiris, polyzos}@aueb.gr

***Abstract*—Information-Centric Networking (ICN) has been in the spotlight of recent research efforts. ICN architectures depart from the traditional host-centric (inter-)networking paradigm and leverage the role of information by placing it in the core of all networking functions. A target of ICN is to address the security shortcomings of the legacy host-centric paradigm. In this paper we discuss security requirements of an integrated satellite-terrestrial ICN architecture, we present some security solutions and we assess these solutions in our integrated testbed.**

## I. Introduction

Information-Centric Networking (ICN) is an emerging paradigm that has received increased attention by the research community. Various research efforts (see [1] for a survey on them) advocate that a network built around information will overcome various limitations of the current networking architectures, including inefficient mobility handling, lack of effective multicast, insecurity and distorted business environment.

Most networking architectures are designed to interconnect endpoints. Nevertheless, nowadays the Internet users are mainly interested in accessing information items rather than connecting to machines. On the other hand, the current network design principles have not been adapted to this requirements shift, therefore, even though a user might be interested, for example, in a particular YouTube video, the network regards the corresponding actions as a request to connect to the YouTube server and to transfer (a bunch of meaningless for the network) bytes. Although "just" working, this design causes unnecessary resource consumption and has led to the creation of an ecosystem of supplementary, vertically organized, "add-ons"–such as P2P networks or CDNs–that violate the current networking model in order to provide more efficient services.

ICN is envisioned to bring the functionality of those addons solutions to the core of the network, providing a common–horizontal–mean for efficiently distributing information. In an ICN based architecture each piece of information has a statistically unique name and applications can request the network to deliver named information, therefore the primary function of the network is to locate and deliver information rather than to locate hosts and arrange communications between them [2].

Moreover, the Internet was designed to operate in a completely trustworthy environment, hence user and data authentication, data integrity and user privacy were not a requirement. What is more, the Internet was designed to forward any traffic injected in the network. These characteristics allow spammers, hackers and attackers in general to launch Denial of Service (DoS). Unlike the current Internet, ICN architectures are interest-driven, i.e., there is no data flow unless a user has explicitly asked for a particular piece of information. This is expected to significantly reduce the amount of unwanted data transfers. Moreover, most ICN architectures add a point of indirection between users requesting a piece of information and users possessing this piece of information, decoupling the communication between these parties. This decoupling can be a step towards fighting DoS and protecting user privacy.

In [3] we applied PSI (Publish-Subscribe Internet), a Future Internet ICN architecture defined by the PURSUIT project [4], in an integrated satellite-terrestrial network and we demonstrated potential gains in various applications scenarios, using an integrated testbed. In this paper we extend our testbed to include security solutions for access control, content integrity, content authenticity, content provenance verification and subscriber privacy. Through measurements we evaluate these solutions and we propose recommendations.

## II. The PSI Architecture

The PSI architecture is built around the so called publish/subscribe paradigm, i.e., users interested in receiving a specific information item "subscribe" to it (hence they are referred to as the *subscribers*) and users wishing to disseminate an information item "advertise" and "publish" it (hence they are referred to as the *publishers*). A typical transaction includes the following steps i) information advertisement ii) information subscription and iii) information publication.

Information is the building block of the PSI architecture. Every information item is identified by a flat identifier known as the *Rendezvous Identifier* (RId). Moreover every information item belongs to–at least–one *scope*. Scopes serve a dual purpose: they give a hint for the information location and they group information items under a certain dissemination strategy, defined by the scope owner. Each scope is identified by the *Scope Indentifier* (SId). A path of scopes is defined

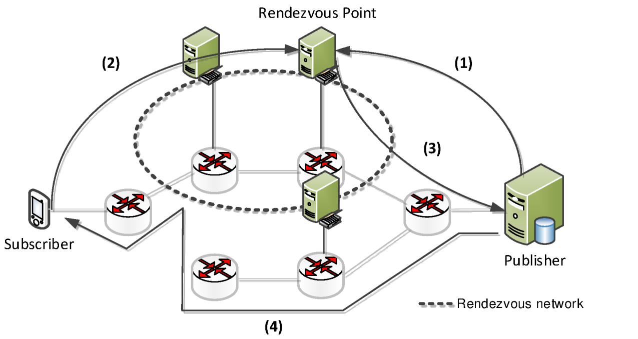

Fig. 1. PSI Architecture overview. (1) information advertisement, (2) information subscription, (3) advertisement/subscription match notification, (4) information publication

as the sequence of the SIds of the scopes that form this path. This sequence combined with an RId identifies uniquely an information item. Every scope is managed by a set of specialized network nodes called *Rendezvous Nodes* (RNs). All RNs form the *Rendezvous Network* of the architecture. The rendezvous network is a lookup service which routes a request for an information item (subscription) to a specific RN which "knows" (at least) one publisher of that item. This RN is also known as the *Rendezvous Point* (RP) for that information item.

Information availability is advertised by the *Publishers*. The advertisement process involves the announcement of a specific information identifier (RId) in one or more scopes. With this announcement the information identifier is stored in the RN that manage these scopes (and now become the RP of the advertised information item). After this step this information item becomes available to potential subscribers. Subscribers express their interest in an item by sending a subscription message to an RP. Upon receiving a subscription message for a specific information item and provided that at least a publisher exists, the RP instructs a *Topology Manager* (TM) to create a forwarding path from the publisher to the subscriber.

Fig. 1 illustrates the basic operations of the PSI architecture. In this figure the rendezvous network consists of a three nodes (which in this example form an overlay network). A publisher, that wishes to advertise an information item, locates the appropriate RN and sends the advertisement message (message 1). Now this RN becomes the RP for this information item. A subscriber on the other hand, wants to access this item, so she sends a subscription message to the RP (message 2). The RP instructs the TM to calculate a path between the publisher and the subscriber and notifies the publisher, that a publication/subscription match has taken place and a path from him towards the subscriber has been created (message 3). The publisher finally publishes the information item which is forwarded to the subscriber (message 4).

## III. Security requirements and solutions

### A. Access control

The need of access control in ICN is stronger than in current IP networks. The reason is that in ICN, information items do not always lie in the administrative realm of their owner, due to caching and content replication (which is facilitated by ICN). It must be possible therefore to enforce access control policies for information items scattered around the network. This requirement raises some new challenges, such as enabling caches and content replication points to enforce access control policies without compromising user privacy, as well as, without revealing user credentials, or even the access control policy itself. An access control scheme should be scalable and flexible, and it should provide reusable access control policies, as well as, user privacy protection.

Multicast environments (e.g. for the case of satellite IPTV service where a stream is multicasted to many receivers) specifically impose additional requirements regarding access control. It is desirable to implement access control mechanisms close to the leafs of the multicast tree, so every time a new node joins the tree, access control signaling does not have to be propagated to the root of the tree–which for the case of an integrated satellite-terrestrial network this usually means that signaling has to traverse the satellite link. Moreover the removal of a node from a multicast tree, should take place as smoothly as possible, without excessive signaling.

As an access control solution we have chosen the solution proposed by Fotiou et al. [9] This solution enables access control enforcement by any entity in the network, providing, at the same time, user credentials protection . With this solution the owner of an information item creates an access control policy and stores it in an Access Control Provider (ACP). ACP is a reliable entity which is trusted by the publisher to enforce an access control policy. Subsequently, he attaches a pointer to that policy (e.g., a URL) in every advertisement of items that are protected by that access control policy. Any 3rd party can request subscribers to authenticate themselves against an access control policy, at the location indicated by the pointer (i.e., at the ACP that holds the actual definition of the policy). Upon a successful authentication, 3rd parties are notified by the ACP.

### B. Content Integrity, Provenance and Authentication

A common security requirement for ICN is the protection of content integrity. Content integrity guarantees that a piece of content has not been modified during its transmission. Traditionally integrity has been provided by mechanisms that are based on a network of trusted third parties. However this model has many weaknesses and hence a depart from third party based solutions is highly desirable. Content integrity mechanisms in ICN should rely on the content name, rather than on certificates. This property significantly facilitates content replication, caching, and multihoming. Moreover a content integrity mechanism should not prohibit the usage of mutable or/and human readable content names. When multicast is used, all users receive the same piece of content, therefore integrity mechanisms can be distributed to all of the receivers (e.g., if all users download a big file, the integrity of each file chunk can be verified by different groups of users); such a distribution of integrity mechanisms not only enables faster integrity checks, but also prevents integrity-based DoS attacks, e.g., faulty requests for content retransmissions. However, trust between the nodes performing the integrity checks has to be ensured.

Another important security requirement is the content provenance verification, i.e., the ability to verify the sender of a piece of content. In a secure ICN architecture, it should be impossible for a (malicious) user to impersonate the owner of a piece of content. Proper content provenance mechanisms enable the deployment of effective accountability solutions and prevent man-in-the-middle attacks. The broadcast nature of satellite communications can enhance content provenance verification mechanisms, as it facilitates the distribution of information that can be useful for these mechanisms, such as black lists with malicious publishers, or lists of revocated encryption keys.

Every piece of content is generally composed of two parts: its name and its representation (e.g., consider a movie encoded in an mpeg-2 file named "Movie X.avi", the name of this piece of content is "Movie X.avi", whereas its representation is the bytes that form the mpeg-2 file). In an ICN architecture it should be possible to verify the binding of these two parts, i.e., given a content name and a representation, it should be possible to verify if they belong to the same content item or not. Content authentication is orthogonal to content integrity: if a user requests "Movie X" and receives "Virus A", the content integrity check will simply verify that "Virus A" has not been modified during transmission whereas, the authentication check will reveal that the content item the user received is not what he requested. In multicast applications, a piece of content can reach many users, therefore a successful phishing or spamming attack will have greater impact. For this reason, the content authentication requirement becomes even stronger.

All these security requirements can be satisfied using Identity Based Signatures [10]. Identity Based Signatures are digital signatures that can be verified using an identity, and some known public system parameters. A trusted entity, named Private Key Generator (PKG) is responsible for generating the public parameters, as well as, for generating private keys that correspond to identities. The identity of both the publisher and the content is used to produce the key for the signature generation: a successful verification of a signature proves the content item provenance (due to the publisher part of the key), as well as, the content authenticity (due to the content part of the key). The signature is applied over the hash of the content data therefore integrity checks can be easily done.

### C. Forwarding Plane Availability

The availability of the forwarding plane is of significant importance. The forwarding plane should be resistant to various attacks, which may lead to service interruption and Denial of Service. Forwarding availability may as well be affected by in-network content verification mechanisms: an attacker may request a big amount of content items from various sources–therefore protected with different security keys–saturating the resources of the forwarding nodes by making them verifying unnecessary data. Content verification combined with bad transport decisions may also affect the availability of the forwarding plane [11]. If content items are transmitted in small chunks, forwarding nodes will be kept busy performing many verifications, on the other hand if big chunks are used they will be fragmented in order to fit to layer 2 MTU. If a single fragment is lost then the verification of the chunk will fail and the whole chunk will be re-requested. Finally, forwarding plane availability is greatly supported by the caching system, therefore, attacks on caches may degrade forwarding plane availability. Broadcasting and use of satellite links for redundancy can help address this attack. On the other hand, satellite infrastructure and especially Gateways, can possibly be a single point of failure, hence appropriate filtering at ground stations may be necessary to avoid DoS attacks to satellites.

Forwarding plane availability may be supported by exploiting alternative routes and sources for content dissemination. It is not rare to have multiple paths among two users or to store the same content in multiple locations. Therefore, it would be a waste of resources not to use them either for performance enhancement or for resilience to node and path failures. In order to enhance performance through bandwidth aggregation and to bypass performance bottlenecks (due to congested links) many applications apply concurrent multipath and multisource transmission, where distinct content parts are requested from different sources and via different paths. Additionally, in order to boost resilience to network or host failures, a transfer can select back-up, disjoint or at least only semi-overlapping dissemination routes, in addition to the basic path used in a single path transmission. Consequently, if the basic path fails the transfer can be resumed via the back-up path. ICN constitutes a promising ground for applying such sophisticated transmissions schemes, as it combines source routing and end-hosts loose coupling. Source routing allows subscribers to completely specify the path that data will traverse, hence they can on-the-fly perform data-flow redirections and other relevant traffic engineering mechanisms, such as on and off path caching, for increasing the robustness of the delivery. Furthermore, loose coupling between publishers and subscribers implies that the transmission of a content item is not restricted or connected to a single publisher.

### D. Subscriber privacy

Preserving subscriber privacy is a challenging requirement. Subscriber privacy preservation usually refers to one (or more) of the following sub-requirements: anonymity, unlinkability, and unobservability [12].

Anonymity is the ability to hide a user identity within a set (the anonymity set), i.e., every subscriber should not be identifiable within the anonymity set. The anonymity set is the finest grained information a malicious entity can learn about a subscriber. As an example, in an IPTV application, the set of the clients of an IPTV service can be considered as a possible anonymity set; a malicious user cannot learn any information about a subscriber, apart from the fact that he is a client of a particular provider. The bigger the anonymity set is, the better the privacy for the user.

Unlinkability of two (or more) items of a system, means that by observing the system it is not possible to learn any information about their relation. In the context of ICN this means that by observing messages (e.g., subscriptions and publications) it should not be possible to correlate these messages to each other, nor to associate these messages to a subscriber (or publisher) identity.

Unobservability means that the object requested by a particular subscriber cannot be distinguished from a set of objects, i.e., given a subscriber identity, a subscription message that originated from that subscriber and a set of information items, it should be not possible to determine the specific item in the set that is requested in the subscription. Unobservability and anonymity are two different properties; when unobservability is used the identity of the subscriber is not necessarily hidden. Consider again the example of an IPTV service; unobservability in that case, means that a third entity (attacker) cannot discover which channel a particular subscriber, has requested to view.

An aspect of subscribers privacy that is receiving increasing attention is that of Decisional Interference, i.e., the ability of an attacker to filter (censor) the information that a subscriber can access (e.g., prevent one or more IPTV subscribers from watching a specific program). ICN architectures should provide subscribers the means for circumventing malicious censorship. A characteristic of ICN architectures that affects subscribers privacy is that the content name, included in a subscription message, reveals more information about subscriber preferences than an IP address in a connection request in a legacy IP network.

In addition to names, caching, which is a key functionality provided by ICN, can also be a threat to subscriber unlinkability. Lauinger et al. [13], use response times in order to estimate if a piece of content is cached close to the attacker location: a smaller response time means that the requested content is cached close to the attacker. The fact that a piece of content is cached close to the attacker is an indication that another user, located close to the attacker, has requested the same content. Satellite broadcasting is expected to further facilitate similar timing-based attacks: due to the latency that is imposed by the satellite link, it should be easy for an attacker to determine if he received a piece of content from a node across the satellite link, or from a cache close to him. On the other hand, satellite broadcasting is resistant to decisional interference: unless an attacker owns the satellite network, it is very hard for him to censor (block) the transmitted information. Moreover in a broadcasting environment, it is very hard to determine the intended recipients of a transmitted piece of content, which is a significant advantage regarding subscriber anonymity. In multicast applications, e.g., IPTV, the strengths of an attacker are greatly determined by his position in the network; an attacker close to the source of a multicast tree can collect information about more subscribers compared to an attacker close to a leaf of a multicast tree. On the other hand, in order for an attacker to perform decisional interference on specific subscribers, without affecting the information that is received by the rest of the subscribers that belong to the multicast tree, he has to be located close to the attacked subscribers.

A promising solution for providing user privacy is the homomorphic encryption.

## IV. Integrated testbed

In this work we use the testbed implemented in [3]. The testbed includes nodes running Blackadder [5], which is an open-source prototype PSI implementation, and also emulated satellite components using the OpenSAND [6] open-source satellite network emulator.

Blackadder is a PSI prototype built using the Click modular router [7]. It provides a basic implementation of the architecture's core functions (i.e., advertising, subscribing, publishing). Blackadder supports a single RN and fixed length RIds and SIds. An API [8] is provided in order to enable applications to invoke the implemented functions. Blackadder runs in Linux and can operate either in user space or in kernel space offering two modes of operation: it can either communicate through the exchange of raw Ethernet frames over a LAN or operate in overlay mode on top of IP. For reasons of ease of deployment and due to testbed constraints, we operate Blackadder in overlay mode.

OpenSAND, is a tool which implements real satellite DVB encapsulation and emulates lower layer protocols, such as MPEG-2 Transport Streams (MPEG-2 TS) or ATM. OpenSAND supports three types of nodes: Satellite Terminal, Satellite Emulator, and Gateway. Satellite Terminals transmit/receive traffic to/from the emulated satellite. The Satellite Emulator emulates a transparent or regenerative satellite link including adding a preconfigured propagation delay. Finally, the Gateway acts as the central access point for Satellite Terminals and as the satellite NCC (Network Control Center), which monitors and controls the satellite network, and performs real-time time/frequency resource allocation. OpenSAND can also emulate satellite link unavailability. OpenSAND runs in Linux, which facilitates its integration with Blackadder.

OpenSAND emulator by default operates upon three LAN networks: the satellite network, the terminal network and the gateway network. The satellite network includes the three essential entities of openSAND, i.e., the Terminal, the Emulator and the Gateway. As illustrated in Fig. 2 each entity is deployed on a physically distinct machine and a router is used for connecting the three nodes. The satellite network emulates the two satellite links (from gateway to terminal and reverse), by applying the proper packet encapsulation, error concealment and latency.

The other two (terrestrial) networks are used for attaching application users at the edges of the satellite testbed. In our

| | |
|---|---|
| Maximum bandwidth on terrestrial wired links | 100 Mbps |
| Blackadder execution level | User space |
| Blackadder running mode | IP overlay |
| openSAND payload type | Transparent |
| openSAND ST Return Link Standard | DVB-RCS |
| openSAND ROHC | Disabled |
| openSAND Forward link Encapsulation Scheme | ULE, MPEG2-TS |
| openSAND Return link Encapsulation Scheme | ATM/AAL5 |
| openSAND delay | 500ms |

TABLE I
TESTBED PARAMETERIZATION.

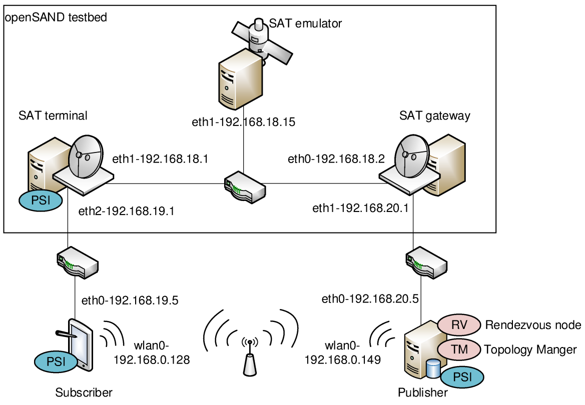

Fig. 2. Testbed overview

case we added two Blackadder-enabled nodes at the terminal and the gateway network, each of which is able to host multiple ICN applications.

The configuration parameters of both Blackadder-enabled nodes and openSAND network emulator are presented in Table I.

## V. EVALUATION

### A. Access Control

In order to evaluate the chosen solution we assume a Content Distribution Network (CDN) and an Access Control Provider (ACP). The publisher has stored her files in the CDN. The publisher creates an access control policy and stores it in the ACP. Finally, the publisher informs the CDN about the information items that are protected by that access control policy. The subscriber subscribes to a protected item. This subscription initiates the subscriber authentication process. Upon a successful subscription the ACP notifies the CDN. This notification means that the subscriber has provided the correct username and password, therefore he is eligible to access protected item.

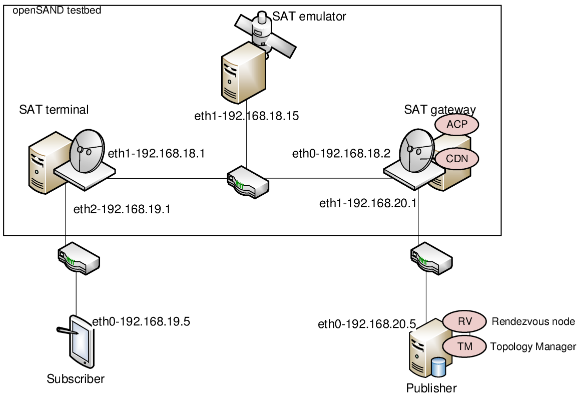

Fig. 3. Access control system

The implemented solution adds some additional messages, therefore, the performance of the system is greatly affected by the placement of the various entities. It should be noted here that the blackadder prototype does not support subscriptions (and publications) with arguments–e.g., subscribe and the same time provide credentials. However this kind of messages are required in this scenario. In order to cope with this limitation “operations with arguments” have been implemented as higher layer functions, which introduce more (lower-layer) messages.

If the subscriber and the ACP are located in opposite sides of the satellite link (as in Fig. 3), at least two messages have to be sent over the satellite link in order to implement the desired functionality. In that case, 2.9sec are required in order for the first chunk of the file to arrive whereas in the case in which subscriber and ACP are located in the same side, only 0.015sec are required since the satellite link is never used during the subscription process.

### B. Content Integrity, Provenance and Authentication

For the evaluation of the proposed solution we consider a scenario in which the publisher and the subscriber are pre-configured with the system parameters (public information for verifying an identity based encryption). The subscriber requests a content identified by RIdA. The subscriber knows the (human-readable) identifier of the publisher. The publisher signs the content using the private key that corresponds to the concatenation of his identifier and RIdA (digital signature). Anyone that knows the system parameters can verify the signature. The signature is applied for every transmitted chunk using the following process: A 160 bit hash of the chunk is generated using the SHA1 function, then the hash is encrypted.

The components of the system are deployed as shown in Fig. 4. As it can be seen, PKG and publisher are located in the same side of the satellite link, as every time the publisher wants to generate a key he has to contact the PKG.

The size of the generated signature is 58 bytes, therefore, it does not introduce significant communication overhead. Signature verification in an Intel Dual Core 3.4Ghz machine with 2GB of RAM, using PBC Sig library 0.0.8 [14], required 0.027s which is negligible for a single subscriber. However, the overall system performance can be affected if all (or multiple) intermediate nodes perform signature verification. Moreover for each identity, a different key has to be generated. Key generation requires 0.018s, in a machine with the same specifications as described above. The granularity of the identity

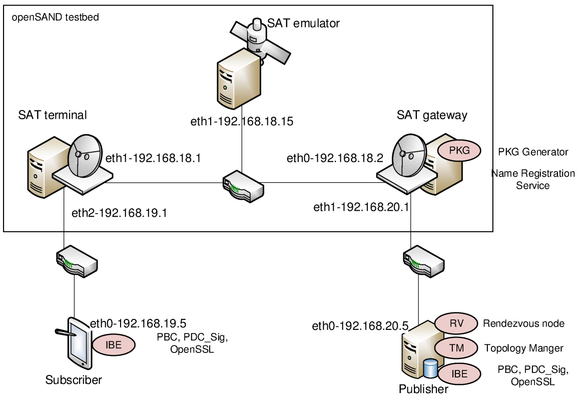

Fig. 4. IBE setup

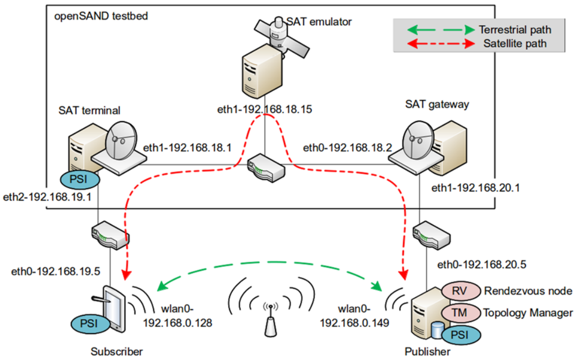

Fig. 5. Setup for the multipath based solution

also affects how often a private key has to be generated: if each chunk has its own identity then a private key has to be generated for every chunk, otherwise, if all chunks of the same piece of content have the same identity, then a private key has to be generated per content item. Since the time required to generate a signature is small (0.038s), it may be possible to generate them "on-line" during (the first) transmission.

### C. Forwarding Plane Availability

Using the topologies shown in Fig. 5 and Fig. 6, we design two experimental scenarios. The first one aims at examining the gains of utilizing multiple paths and the second the gains of utilizing multiple sources.

In the multipath scenario, we use the topology of Fig. 5, which provides two available transmission paths from the publisher to the subscriber. The first is a two-hop route composed of a satellite and a fixed Ethernet link (path 1), and the second path is a single hop terrestrial link (path 2), which in the testbed is wireless. According to this scenario the data flow will follow one path at any time, selecting always the fastest route in terms of delay, which in our testbed is path 2. In case path 2 fails, the transmission will be shifted to path 1 until content delivery is possible again via path 2. We implemented a script that runs at the subscriber node and periodically disables the terrestrial path for a period of 10 seconds and then re-enables it for 15 seconds. There results of this scenario are presented in Table II.

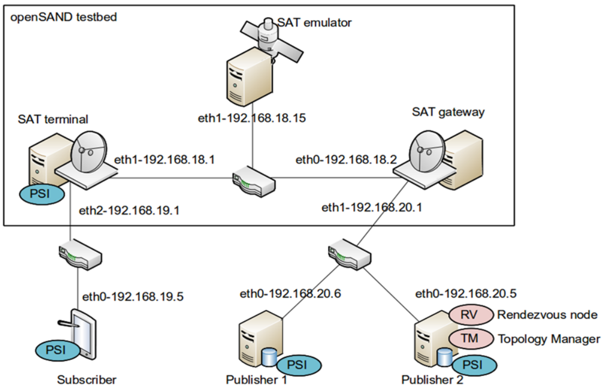

Fig. 6. Setup for the multisource based solution

| | Transfer rate (KBs) | Path-switching time (ms) |
|---|---|---|
| Singlepath without failure | 1164.4 | - |
| Singlepath with failure | 615.5 | - |
| Multipath with failure | 961.5 | 1.2 |

TABLE II
RESULTS OF MULTIPATH SCENARIO.

Studying Table II we see that the path failure reduced the performance of the singleflow scenario by 41.2%. However, when exploiting both paths, the path failure decreased the average throughput of the transmission by 17.5%, which means that the service improved its performance by 23.7%. Thereupon, we argue that exploiting multiple paths significantly enhances forwarding plane availability. A second interesting result is that the path-switching time is 1.2 milliseconds. The fact that the window size of the inactive path is never closed (it is set to 3 packets), so as to keep probing the network's condition, minimizes the time period during which the transmission is idle and makes the path-switch transparent to the end-host.

In the multisource scenario we use the topology of Fig. 6, which provides two publishers attached to the GW and a subscriber at the ST. There is only one path amongst the PSI nodes, hence all data will be transmitted via the openSAND testbed. According to this scenario, the subscriber will pull data only from one publisher at the time, selecting always the fastest route in terms of delay. In order to illustrate source-switching clearer and to reduce protocol flappiness, we added 500ms delay to Publisher 2, making Publisher 1 the preferred source. Nevertheless, Publisher 1 is scheduled to stop responding at time $t$, when the subscriber will start requesting data from the "back up" publisher. We implemented a script that "shuts down" Publisher 1 for 8 seconds and

| | Transfer rate (KBs) | Source-switching time (ms) |
|---|---|---|
| Single-source without failure | 1040.4 | - |
| Single-source with failure | 617.8 | - |
| Multisource with failure | 905 | 1.2 |

TABLE III
RESULTS OF MULTISOURCE SCENARIO.

then re-enables it. The script is programmed to break the connection on the $20th$ second of the transfer. There results of this scenario are presented in Table III.

Similarly to the multipath scenario, Table III unveils the advantages of dynamic source selection. Obviously, shutting down the publisher while the single-source transmission is still active, creates an critical bottleneck. Thereafter, the throughput of the service is decreased by 40.7% compared to the "without failure" scenario. On the contrary, the black-out reduces the performance of the multisource transmission by 13%, which constitutes an improvement of 27.7% for the forwarding plane resilience. Even though this result is quite expected, it should be noted that this gain is achieved without the need for complex operations, such as statefull routers or increased signaling overhead, since the PSI architecture inherently allows source-routing and centralized path selection. Additionally, we believe that the application of concurrent multipath transmission [15] would provide even better results.

### D. Subscriber privacy

In this subsection we evaluate a privacy preserving solution that is based on the homomorphism property of the Paillier cryptosystem. This property allows operations over encrypted data by a 3rd party without revealing to that 3rd party any information associated with this data. The evaluated solution is based on a query/response model where a subscriber defines a linear equation over a set of information items and a publisher solves this equation. The result of this operation is the item in which the subscriber is really interested. Nevertheless, the publisher of the data is unable to interpret the result as it is encrypted with a key that is known only to the subscriber. In order to support the required cryptographic operation an open source library is used.

In this setup the subscriber, who has a public/private key pair, learns–through an out-of-band mechanism–an ordered list of the publications the publisher offers. Then, she creates a query over that list and sends it to the publisher. The publisher calculates the response, which is the desired item encrypted with the public key of the subscriber. It is assumed that the publisher has advertised 50 publications and each publication is 128 bytes.

A subscriber using an Intel Centrino single core and 1GB of RAM and the Pallier library V0.8 [16], needs 0.840s in order to create a query. However queries can be pre-computed therefore this time is considered negligible. The size of the subscriber's public key is 256 bytes. If the subscriber transmits her public key with every query then the size of the query is 13056 bytes.

In an Intel Dual Core 3.4Ghz machine with 2GB of RAM, it required 0.602s for the publisher to create a block of the response (i.e., to combine 128bytes of all 50 publications). Therefore it is clear that combining a large number of publications introduces significant delays. Moreover the ciphertexts are twice as big as plaintexts, i.e. in order to transmit a file of 1KB, 2KB are sent to the subscriber. The subscriber requires in average 0.060s to decrypt a block of the response.

## VI. CONCLUSIONS

In this work we discussed some security requirements for Integrated Satellite-Terrestrial Information-Centric Networks and we evaluated security solutions for: access control, content integrity, content authenticity, content provenance verification and subscriber privacy. These solutions were evaluated in an integrated testbed using an ICN prototype and a satcom network emulator. Our findings show that these solution offer added value, providing that their components are deployed in such a way that do not over-utilize the satellite link.

## ACKNOWLEDGMENT

This work has been performed under the ESA ARTES 1 project $\phi$SAT: The Role of Satellite in Future Internet Services (ESA/ESTEC Contr. No.: 4000103360/11/NL/NR). Responsibility for the content resides with the authors.

## REFERENCES

[1] G. Xylomenos, C. Ververidis, V. Siris, N. Fotiou, C. Tsilopoulos, X. Vasilakos, K. Katsaros, and G. C. Polyzos, "A survey of information-centric networking research," *Communications Surveys Tutorials, IEEE*, vol. PP, no. 99, pp. 1–26, 2013.

[2] D. Trossen, M. Sarela, and K. Sollins, "Arguments for an information-centric internetworking architecture," *SIGCOMM Comput. Commun. Rev.*, vol. 40, no. 2, pp. 26–33, 2010.

[3] C. Ververidis, P. Frangoudis, Y. Thomas, V. Siris, G. Polyzos, I. Andrikopoulos, F. Arnal, C. Baudoin, and M. Guta, "Experimenting with services over an information-centric integrated satellite-terrestrial network," in *Future Network and Mobile Summit (FutureNetworkSummit), 2013*, July 2013, pp. 1–10.

[4] FP7 PURSUIT project. [Online]. Available: http://www.fp7-pursuit.eu/PursuitWeb/

[5] G. Parisis, D. Trossen, and D. Syrivelis, "Implementation and evaluation of an information-centric network," in *IFIP Networking Conference, 2013*, 2013, pp. 1–9.

[6] OpenSAND. [Online]. Available: http://www.opensand.org/

[7] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek, "The click modular router," *ACM Trans. Comput. Syst.*, vol. 18, no. 3, pp. 263–297, Aug. 2000. [Online]. Available: http://doi.acm.org/10.1145/354871.354874

[8] D. Trossen and G. Parisis, "Designing and realizing an information-centric internet," *Communications Magazine, IEEE*, vol. 50, no. 7, pp. 60–67, 2012.

[9] N. Fotiou, G. F. Marias, and G. C. Polyzos, "Access control enforcement delegation for information-centric networking architectures," *SIGCOMM Comput. Commun. Rev.*, vol. 42, no. 4, pp. 497–502, Sep. 2012. [Online]. Available: http://doi.acm.org/10.1145/2377677.2377773

[10] J. Choon and J. Hee Cheon, "An identity-based signature from gap diffie-hellman groups," in *Public Key Cryptography PKC 2003*, ser. Lecture Notes in Computer Science, Y. Desmedt, Ed. Springer Berlin Heidelberg, 2002, vol. 2567, pp. 18–30.

[11] S. Salsano, A. Detti, M. Cancellieri, M. Pomposini, and N. Blefari-Melazzi, "Transport-layer issues in information centric networks," in *Proceedings of the Second Edition of the ICN Workshop on Information-centric Networking*, ser. ICN '12. New York, NY, USA: ACM, 2012, pp. 19–24.

[12] A. Pfitzmann and M. Khntopp, "Anonymity, unobservability, and pseudonymity a proposal for terminology," in *Designing Privacy Enhancing Technologies*, ser. Lecture Notes in Computer Science, H. Federrath, Ed. Springer Berlin Heidelberg, 2001, vol. 2009, pp. 1–9.

[13] T. Lauinger, N. Laoutaris, P. Rodriguez, T. Strufe, E. Biersack, and E. Kirda, "Privacy risks in named data networking: What is the cost of performance?" *SIGCOMM Comput. Commun. Rev.*, vol. 42, no. 5, pp. 54–57, Sep. 2012.

[14] PBC sig library. [Online]. Available: http://crypto.stanford.edu/pbc/sig/

[15] Y. Thomas, C. Tsilopoulos, G. Xylomenos, and G. C. Polyzos, "Accelerating file downloads in publish subscribe internetworking with multisource and multipath transfers," in *Proceedings of the World Telecommunications Congress*, 2014.

[16] Pallier library. [Online]. Available: http://acsc.cs.utexas.edu/libpaillier