An authentication framework for a hybrid satellite network with resource-constrained nodes

Ayan Roy-Chowdhury[a], John S. Baras[b] and Michael Hadjitheodosiou[c]

[a,b]Electrical and Computer Engineering and Institute for Systems Research, University of Maryland, College Park, USA;

[c]Institute for Systems Research, University of Maryland, College Park, USA

ABSTRACT

The new phase of space exploration involves a growing number of human and robotic space missions to remote planets with varying communication and service requirements. Due to the critical nature of the missions, security is a very important requirement that needs to be addressed. Among primary security requirements are user authentication and message integrity that are needed to ensure that the data in the network is transmitted without unauthorized modifications between the source and destinations, and that data from only authorized network nodes are accepted by other nodes. In this paper we focus on the issue of user authentication and data integrity for a specific space network architecture supporting lunar exploration. We consider a hybrid network consisting of a terrestrial network on Earth, a network on the lunar surface, and a satellite constellation that connects the two surface networks. The lunar network comprises sensor nodes serviced by stationary gateways and mobile robotic vehicles with sensing capability, while the network on Earth is envisioned as a combination of private and public networks. The problem of authentication in this network is complex due to the presence of nodes with varying capabilities in terms of computation strength, storage and energy. The nodes on Earth and the gateways on the lunar surface would have higher computation and energy capabilities compared to the satellites and the sensor nodes. In this situation, an authentication protocol that is optimized to the strengths and limitations of the different classes of nodes would be most suited. We focus on a solution that will operate under the constraints of the space environment (delay, limited energy, limited processing capability at remote nodes). We present a framework for user authentication and data integrity based on an authentication algorithm that makes use of symmetric certificates and hash chains of keys used to compute Message Authentication Codes, to provide asymmetric authentication capabilities to the network nodes, nodes with more resources. We give a detailed description of the authentication protocol we develop for this network and provide an analysis of the security of the protocol by considering various types of passive and active attacks. We also highlight the savings incurred in terms of processing, storage and network bandwidth, which we get in using the proposed protocol in comparison to standard public-key authentication protocols.

Keywords: Space mission network, sensor nodes, satellite broadcast, user authentication, message integrity, public-key cryptography, symmetric cryptography, hash chains.

1. INTRODUCTION

The future of space exploration envisions missions to remote planets to establish permanent outposts that would be connected to networks on Earth. The resulting network would be a hierarchical hybrid mesh, comprising networks on the remote planetary surface, connected to networks on Earth by high-speed satellite backbones that would act as "information highways" to transfer mission telemetry and control information from command centers on Earth, and also relay data from the planetary networks to nodes on Earth.[1,2] Such a network would have sensor nodes, humans, fixed and mobile robotic vehicles in the planetary network, while the network on

Further author information: (Send correspondence to Ayan Roy-Chowdhury)

Ayan Roy-Chowdhury: E-mail: ayan@umd.edu, Telephone: 1 301 405 6561

John S. Baras: E-mail: baras@isr.umd.edu, Telephone: 1 301 405 6606

Michael Hadjitheodosiou: E-mail: michalis@isr.umd.edu, Telephone: 1 301 405 7904

International Conference on Space Information Technology, edited by Cheng Wang, Shan Zhong, Xiulin Hu, Proc. of SPIE Vol. 5985, 59855R, (2005) · 0277-786X/05/$15 · doi: 10.1117/12.659403

Earth would be a combination of private mission control networks and users connected through public Internet-like networks to the satellite gateways. The space component would include satellite constellations that might be interconnected to form a network in space. The capabilities of the planetary nodes would vary widely - from very resource-limited sensor nodes to the base stations and robotic vehicles with higher processing power, storage and energy. The terrestrial nodes would also be similarly varied, but on average would have higher capabilities than the planetary nodes, and easier to control and to adapt to changes in the network environment.

Security is a major component of any network and in this case is a critical and complex requirement. Only the mission control center on Earth and other authorized terrestrial users should be able to send messages to the remote network, and the collected data from the planetary network should be accessible only to mission control (and possibly to other involved scientists in external networks), and no other entity. Therefore suitable security mechanisms should be in place to ensure that (a) the satellites and/or the remote network do not accept spurious command and control messages from unauthorized entities on Earth, and (b) the data sent by the planetary network is accessible only to authorized entities on Earth. This requires that the nodes in the network be able to authenticate the source of command messages, and verify the integrity of the messages to ensure they are not modified in transit. The traffic should also be encrypted so that unauthorized entities cannot read anything meaningful from the satellite transmissions.

The authentication, message integrity and encryption algorithms implemented in the network should be fine-tuned for the peculiar characteristics of the network. Due to the differences in the capabilities of the network nodes, not all would be able to execute similar security algorithms with the same performance. Standard security protocols employed for end-to-end communication in terrestrial networks would fare poorly in the space setting. For example, the terrestrial gateways and the remote planetary gateways would be able to process public-key cryptographic algorithms much more efficiently than the sensor nodes and the satellites, given their superior computation capabilities. It is therefore important to develop for such a network security protocols that operate within the constraints of space environment (limited power/computational ability of the nodes). The security architecture should allow different algorithms and protocols to co-exist in different segments of the network, and to inter-operate seamlessly to ensure efficient end-to-end performance.

In this paper we focus on the problem of user authentication and message integrity for a lunar mission network that can be considered representative of future space networks. We propose that user authentication and message integrity for the sensor nodes in the lunar surface, the satellites and similar devices with resource constraints should be secure but lightweight. The algorithms should minimize the energy expenditure and the computation power required of the nodes. In parallel, stronger cryptographic algorithms (for example, public key cryptography) can be used for nodes with higher resources. Therefore the end-to- end authentication and message integrity protocols should allow different algorithms to co-exist and inter-operate in different segments of the network. We have proposed an algorithm for authentication and message integrity in resource-constrained devices that is ideally suited for the sensor network in our proposed topology.[3] Named *extended TESLA certificates*, the algorithm is based on authentication using TESLA key hash chains[4,5] and its extension to a certificate infrastructure.[6,7] Our algorithm makes use of public-key cryptography on a limited scale to perform initial bootstrapping of the nodes. Authentication and message integrity at the nodes is done using symmetric cryptography-based certificates which are computation and energy-friendly. The algorithm requires a certificate authority with higher capabilities reachable by all the users in the network. The algorithm can be implemented also in a hierarchical manner, with one infrastructure at the level of the "weakest" nodes, and a second infrastructure at a higher level involving the nodes with more powerful resources. Based on this algorithm, we describe an instantiation of end-to-end unicast user authentication and message integrity protocols for the lunar space network that we consider.

The rest of the paper is organized as follows. In section 2 we give a brief description of the lunar network we consider. The TESLA and TESLA certificate algorithms are reviewed in section 3. We give a description of our authentication protocol for the lunar network architecture in section 4. Security and performance analyses of the authentication protocol are in section 5. We conclude the paper in section 6 with a discussion of our current and future work on this topic.

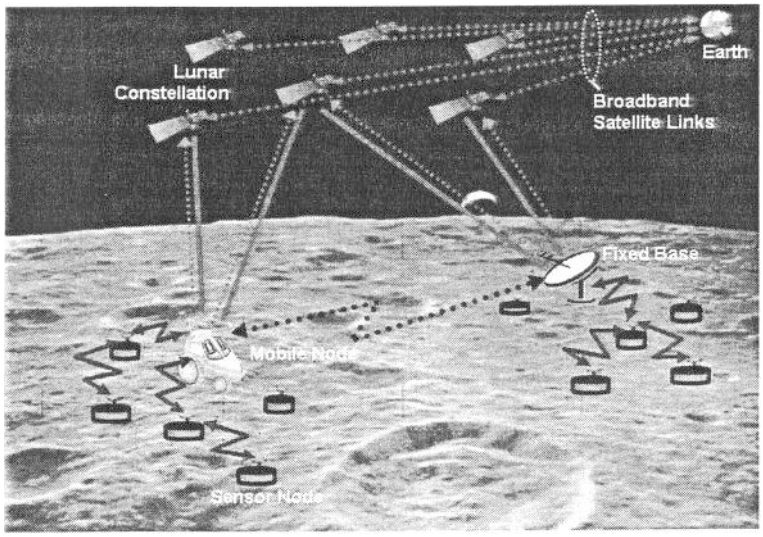

Figure 1. Schematic of lunar surface network

2. DESCRIPTION OF THE NETWORK TOPOLOGY

We give a brief description of the lunar network topology since the type of security protocols to implement depend largely on the characteristics of the network. We make certain assumptions about the network characteristics because the mission requirements are not yet widely known. We divide the network into three segments - the lunar surface network, the space satellite component, and the terrestrial network, and we follow a modular "bottom-up" approach, specifying the network starting with the topology on the lunar surface, and then extending it to Earth.

2.1. The lunar network

We design the lunar surface network to be comprised of sensor nodes serviced by stationary gateways and mobile robotic vehicles with sensing capability. The sensors are grouped into clusters based on their geographical location and radio range proximity to one another. Each group of sensor nodes has a satellite gateway/base station (BS) that aggregates the data collected by the sensor nodes in its range. There might be multiple base stations that can communicate wirelessly with one another. The base stations can also communicate with a lunar satellite constellation orbiting the moon. The satellite constellation relays the data collected from the base stations to networks on Earth. The network is managed from a dedicated control center on Earth that can send remote commands via satellite uplink. A schematic of the lunar network is given in Fig. 1.

Each sensor node has limited processing power and storage, to perform basic sensing applications and store several megabytes of data. The energy of each sensor node is renewable, based on solar sources. We make the important assumption that the network supports IP protocol in our model. Therefore, an IP address is associated with each node, and each also supports ad hoc routing protocols. We assume that the sensor nodes can support different security functions. However, due to the limitations on computation power and storage, public-key cryptography is not suitable since it makes heavy demands on computation, energy and memory to store keys, for resource constrained devices. Therefore we assume that the sensor nodes support public-key cryptography on a limited scale, primarily for bootstrapping security functions. Otherwise, for all security applications, the sensor nodes support symmetric cryptographic algorithms for encryption, authentication and data integrity, which are much less computation and energy intensive. The security algorithms are encoded in software and hardware in the sensor nodes and such functionality is re-configurable by downloading new software from the base stations. The important parameters in the function of a sensor node are: lifetime (i.e., energy), maximizing data collection, and maximizing the data transfer.

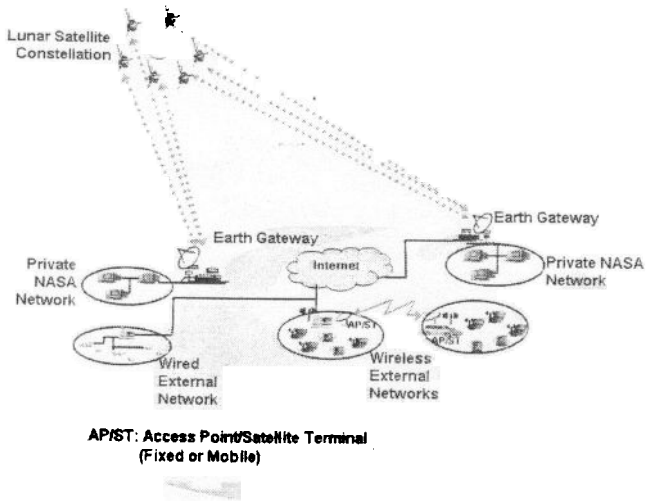

Figure 2. Schematic of the terrestrial network

Each base station can also act as a sensor and collect data itself, which it sends to the orbiting constellation. We assume that the base stations have higher processing power, more storage and higher energy compared to ordinary sensor nodes. Each base station is IP-addressable and supports ad hoc routing protocols. The base stations can be either fixed or mobile. The fixed base stations are mostly similar to fixed satellite gateways. The mobile base stations are robotic vehicles with movement patterns determined by mission control on Earth. A base station may service multiple clusters. Each base station is capable of content caching, and can store data locally, to be transmitted at a later time to the sensor nodes or to the satellite. The base stations support both public key cryptographic operations and symmetric cryptographic operations.

2.2. The space network

We assume a constellation of six satellites in orbit around the moon that provides total coverage to the lunar surface network. The satellites collect data from the base stations, and relay the collected data directly to the gateways on Earth. The satellites also relay command and control data from Earth to the base stations, and subsequently downloaded to the sensor nodes as needed. Each lunar satellite supports multiple spot-beams, and has a switch for onboard processing of the data. Each satellite is associated with an IP address, is capable of supporting security functionalities for both public key and symmetric cryptography and is also capable of content caching.

2.3. The terrestrial network

A schematic of the terrestrial network is given in Fig. 2. There are multiple satellite gateways on Earth connected to the lunar constellation. In the terrestrial segment, the gateways connect to the mission control center and the associated private network of the mission operators. The private network is connected to the open Internet through high-speed terrestrial links, with suitable protection by network firewalls. External wired or wireless LANs can receive authorized mission data by connecting via the Internet. The user nodes in the wireless LANs are typically mobile devices with processing power, storage and energy limited in comparison to nodes in wired LANs or the private mission networks. All the access points and mobile nodes have IP addresses and support ad hoc routing protocols.

The access points are capable of public key and symmetric key security operations and have no constraints on computation, storage or energy. The user nodes might have limited computation power, storage capacity and energy (for example, PDAs). We assume these nodes are also capable of both public key and symmetric key operations, though to preserve energy and for efficient computation, symmetric cryptographic operations are preferred.

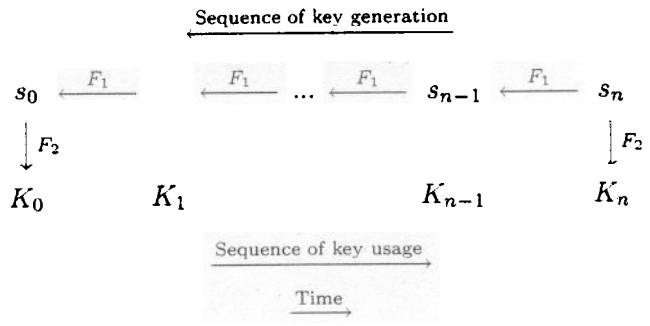

Figure 3. TESLA key generation

3. REVIEW OF TESLA AND TESLA CERTIFICATE

3.1. TESLA Broadcast Authentication Protocol

The TESLA broadcast authentication protocol[4,5] represents a fundamental paradigm shift in source authentication in a group setting. TESLA achieves asymmetric authentication between a source and receivers through the use of symmetric cryptographic Message Authentication Code (MAC) functions. The asymmetry is obtained through the *delayed disclosure* of the authentication keys. We give a brief description of TESLA in the following paragraphs.

TESLA divides the time of transmission by the source into n intervals of equal duration. The source generates a random key seed s_n for interval n, and computes a one-way hash chain by repeatedly applying a one-way function F_1 to s_n. The number of elements of the hash chain correspond to the number of intervals that the source transmits. The source computes the MAC computation key for each time interval by applying a second one-way function F_2 to each element of the hash chain. The functions F_1, F_2 are publicly-available and known to all the receivers. The algorithm is illustrated in fig. 3.

The sender uses the keys in the reverse order of their generation, that is, starting with K_1 in interval 1, followed by K_2 in interval 2, and so on. Owing to the one-way property of F_1 and F_2, it is computationally infeasible for any node to generate s_i knowing K_i, or to generate s_{i+1} knowing s_i. The sender bootstraps the hash chain by broadcasting to all the receivers the anchor element of the chain, for example s_0, signed with its private key (in case of public-key based bootstrapping), or by encrypting s_0 with the secret key it shares with each receiver in the network (for symmetric-key based bootstrapping).

For each packet generated in time slot i, the source uses the authentication key K_i to compute a MAC on the packet. The MAC is then appended to the packet, which is transmitted to the receiver(s). When a node receives a packet, it first checks whether the packet is *fresh*, that is, it was sent in a time interval whose corresponding TESLA key has not been disclosed. This is the fundamental security criterion in TESLA. Each receiver discards any packet that does not meet the security criterion, and buffers only the packets that satisfy the freshness condition. The receiver cannot authenticate the packets immediately since it does not know the corresponding key K_i. The sender discloses the key K_i at a later instant in time by broadcasting the corresponding key seed s_i. Upon receiving s_i, each receiver first verifies the authenticity of s_i by checking $s_i \xrightarrow{F_1} s_{i-1}$ (and therefore ultimately verifying against the anchor element s_0 which has already been authenticated). If s_i verifies correctly, each receiver can compute K_i: $s_i \xrightarrow{F_2} K_i$ and subsequently use the computed K_i to verify the MAC on the packets received during interval i.

Once s_i is disclosed, any node with knowledge of s_i can compute K_i and attempt to masquerade as the sender by forging MACs using K_i. Therefore, K_i is used to compute MACs on packets generated only during the interval i, other time intervals use different keys to compute the MACs. The key seed s_i is disclosed only d time slots after i so that no malicious node can compute K_i and forge packets in the intervening period. d is computed based on the maximum network delay from the source to all the receivers. This is the principle of delayed disclosure of keys.

The major advantage of TESLA in this regard is that it allows similar authentication through the use of computationally efficient MAC functions, and is therefore very attractive for authentication in devices of limited capabilities.

3.2. TESLA Certificates

The idea of certificates based on TESLA was proposed in.[5] The idea has been formalized to form a TESLA-based PKI in.[6] In the algorithm described in,[6] there is a certificate authority CA who creates certificates for an entity B. A low-powered device D contacts B to use its service. The CA and B initially share a secret key $K_{CA,B}$. During time slot n, the CA generates authentication key aK_{B_n} for B to use to compute the MAC on its messages in that interval. The CA creates a certificate $Cert_{CA_n}(B)$ to bind aK_{B_n} to B for interval n. The CA uses its TESLA key tK_{CA_n} to encrypt aK_{B_n} in the certificate, and uses the same key to compute a MAC on the different fields in the certificate.

$$Cert_{CA_n}(B) = \left(ID_B, \{aK_{B_n}\}_{tK_{CA_n}}, n+d, MAC_{tK_{CA_n}}(..)\right) \tag{1}$$

Equation 1 represents the TESLA certificate for node B. aK_{B_n} is known only to the CA and B during period n, while tK_{CA_n} is known only to the CA. $n+d$ indicates the time at which the CA will disclose tK_{CA_n} to the nodes, that is, it is the expiration time of the certificate. The CA sends $Cert_{CA_n}(B)$ to B alongwith aK_{B_n}, which is encrypted with $K_{CA,B}$.

In the time interval $\langle n, n+d\rangle$, D sends a request to B for using B's service:

$$D \rightarrow B: \ (request) \tag{2}$$

To authenticate itself to D, B sends an authentication packet containing its certificate and a MAC on the request, computed with aK_{B_n}.

$$B \rightarrow D: \ \left(Cert_{CA_n}(B), MAC_{aK_{B_n}}(request)\right) \tag{3}$$

When D receives the authentication message, it checks the timestamp of $Cert_{CA_n}(B)$ to make sure it has arrived before time $n+d$, when the CA discloses tK_{CA_n}. If the certificate is "fresh", D buffers the authentication packet. At time $n+D$, the CA discloses its TESLA key tK_{CA_n}. Upon receiving the key, D verifies $Cert_{CA_n}(B)$ by checking the MAC in the certificate using tK_{CA_n}. If the MAC verifies correctly, D obtains B's authentication key aK_{B_n} from the certificate by decrypting with tK_{CA_n}. Subsequently, D checks $MAC_{aK_{B_n}}(request)$ to verify the authenticity of B. Therefore, D is able to verify the identity of B only if it receives $Cert_{CA_n}(B)$ before $n+d$. Once the CA discloses its TESLA key tK_{CA_n}, any node could forge a certificate for the time interval n.

A TESLA certificate allows a node B to add authentication to packets for a single period in time. As the authors mention in,[6] the lifetime of the certificate is short. Therefore, a source node B that transmits for multiple time intervals will need several TESLA certificates from the CA. If there are many sources that send data over long intervals, this can add up to a substantial overhead.

4. AUTHENTICATION PROTOCOL FOR UNICAST COMMUNICATION

Our objective is to design an end-to-end user authentication protocol that allows any receiver in the network to securely authenticate messages from a sender node with limited expenditure of processing power and energy. In particular, we consider the receiver to be located in the lunar network, for example a sensor node, which receives a certain command from an authorized node located in the terrestrial mission network. We assume that the nodes do not have any pre-existing security information about one another. However, all the nodes have access to an online certificate authority, and all the nodes are loosely time-synchronized with the CA. The CA can communicate with the entire network simultaneously through wireless broadcast channels. The wireless transmission channels are assumed to be error-free, so that control messages or data packets do not get lost. We also assume that appropriate policies are in place to allow each node to securely identify itself to the CA during the initial bootstrapping phase, and each node A shares a unique secret key $K_{CA,A}$ with the CA.

The receiver does not need to trust the source or have any prior information about the source; the only requirement, as stated above, is that the receiver trust the CA.We assume that one-way functions F_1 and F_2, derived from pseudo-random function (PRF) families, are publicly available.

In designing the authentication protocol, we need to keep in mind the following design constraints and requirements:

- We assume the sensor nodes would have significantly lower processing power than any terrestrial node or the lunar base stations/satellite gateways.
- The energy available to the lunar nodes, either sensor nodes or the base stations, is limited at any point in time.
- There is a significant propagation delay (of the order of 1.4 seconds[2]) in one-way transmission over the space segment.
- The security algorithms should be adaptable, i.e., they should be designed such that they can be upgraded in the future.

We design our authentication algorithm based on TESLA certificates that would allow the sensor nodes to do user authentication and message integrity efficiently while maintaining strong security. In context of the constraints discussed above, we modify the TESLA certificate algorithm to allow each certificate have a longer lifetime, by incorporating hash chains into TESLA certificates. We also introduce *two* CAs - a terrestrial CA (CA_e) being located at the mission control center on Earth, and a lunar CA (CA_m) located at a satellite gateway on the Moon. The nodes in the terrestrial network share long-term secrets with CA_e, while the nodes in the lunar network share long-term secrets with CA_m. The two CAs are connected to one another via a secure, encrypted control channel over the satellite links with key K_{CA_e,CA_m}, and they share all security information with one another. When a terrestrial node wants a TESLA certificate, it requests CA_e to generate one. Similarly, when a lunar node wants a TESLA certificate, it requests CA_m.

In the following sections, we describe in brief the protocol operations when a terrestrial node A wants to send authenticated messages to a lunar node B. We start with a description of how the TESLA certificate for A is generated by CA_e, with the hash chain extension that extends the lifetime of the certificate.

4.1. Bootstrapping of the Source Node and the Certificate Authority

We make use of the TESLA key chain generation described in.[4,5] Initially, the source node A generates a random seed $s_{A,n}$ and applies one-way function F_1 to $s_{A,n}$ to form a *hash chain*:

$$s_{A,0} \xleftarrow{F_1} s_{A,1} \xleftarrow{F_1} \cdots \xleftarrow{F_1} s_{A,n-1} \xleftarrow{F_1} s_{A,n} \tag{4}$$

The value n depends on the number of time intervals in which A expects to be a source. If the duration of each time interval is Δ, and the total time of A's transmission is T, we have $n = \frac{T}{\Delta}$. A subsequently applies F_2 to each key $s_{A,i}$ generated above and obtains the output $s'_{A,i}$.

$$\begin{array}{ccccccc} s_{A,0} & \xleftarrow{F_1} & s_{A,1} & \xleftarrow{F_1} & \cdots & \xleftarrow{F_1} & s_{A,n-1} \xleftarrow{F_1} s_{A,n} \\ \downarrow F_2 & & \downarrow F_2 & & & & \downarrow F_2 \quad \downarrow F_2 \\ s'_{A,0} & & s'_{A,1} & & \cdots & & s'_{A,n-1} \quad s'_{A,n} \end{array} \tag{5}$$

In time period t_0, A sends $s_{A,0}$ to CA_e (encrypted with the long-term key $K_{CA,A}$ that A shares with the CA) for obtaining a TESLA certificate.

$$A \rightarrow CA_e : \{s'_{A,0}\}_{K_{CA,A}} \tag{6}$$

On successful verification of A's identity, CA_e generates a TESLA certificate for A:

$$Cert_{CA}(A) = \left(ID_A, \{s_{A,0}\}_{tK_{CA_0}}, t_0 + d, MAC_{tK_{CA_0}}(..)\right) \quad (7)$$

Here d is the key disclosure delay for the CA TESLA signature key, and $tK_{CA,0}$ is the CA MAC key for the time period $\langle t_0, t_0 + d\rangle$. The key disclosure delay and the time interval for using each key is computed taking into account the large propagation delay over the satellite links. The propagation delay being approximately constant, that can just be added as an offset in the computation of the disclosure delay.

$tK_{CA,0}$ is generated by the CA using the one-way chain algorithm. The CA starts with an initial seed $s_{CA,n}$ and generates $tK_{CA,0}$ as follows:

$$s_{CA,0} \xleftarrow{F_1} \cdots \qquad s_{CA,n-1} \xleftarrow{F_1} s_{CA,n}$$

$$tK_{CA,0} \qquad \cdots \qquad tK_{CA,n-1} \qquad tK_{CA,n}$$

It is to be noted that from the perspective of the user node A or B, there is no difference in whether the certificate is generated by CA_e or CA_m since both are equally trusted (the two CAs can be looked upon as replicated copies of one another to address the problem of long propagation delay). CA_e sends $Cert_{CA}(A)$ to A, and at the same time securely transmits to CA_m over the dedicated secure channel the certificate for A alongwith $tK_{CA,0}$.

$$CA_e \rightarrow Ca_m : \{Cert_{CA}(A), tK_{CA,0}\}_{K_{CA_e,CA_m}} \quad (9)$$

$$CA_e \rightarrow A : \{Cert_{CA}(A)\}_{K_{CA,A}} \quad (10)$$

4.2. Message Transmission from Source to Receiver

Let terrestrial node A send messages to lunar sensor node B starting in the time interval $\langle t_0, t_0 + d\rangle$. A computes a MAC over the message m_0 using $s'_{A,0}$ and includes its TESLA certificate $Cert_{CA}(A)$ with the message it sends to B:

$$A \rightarrow B : \{M_0 | M_0 : \left(m_0, MAC_{s'_{A,0}}(m_0), Cert_{CA}(A)\right)\} \quad (11)$$

The message is actually transmitted over the satellite link to the lunar satellite gateway, which forwards the message to sensor node B. B checks the freshness of the certificate by checking the timestamp of $Cert_{CA}(A)$ to make sure it has arrived before time $t_0 + d$. If $Cert_{CA}(A)$ has arrived within $\langle t_0, t_0 + d\rangle$, B stores M_0 in its buffer, else B discards the message.

Checking the timestamp on $Cert_{CA}(A)$ is critical for the security of our algorithm. Once the CA discloses $s_{CA,0}$ at time $t_0 + d$, any node in the network can create a fake certificate with timestamp $t_0 + d$, allegedly generated by the CA. Therefore receivers will only accept certificates for which the CA TESLA key has not been disclosed at the time of receiving the certificate.

4.3. Message Authentication at Receiver

At time $t_0 + d$, CA_m broadcasts the key $tK_{CA,0}$ to the lunar network. Since our objective is to avoid public-key cryptographic operations for the sensor nodes, to authenticate the broadcast to each receiver, CA_m encrypts $tK_{CA,0}$ with the long-term secret key it shares with each sensor node.

$$CA_m \rightarrow i : \{\langle t_0, t_0 + d\rangle, s_{CA,0}\}_{K_{CA,i}}, \forall i \quad (12)$$

where i stands for a sensor node in the lunar network.

Receiver B checks the authenticity of the CA broadcast by verifying that the message has been encrypted using $K_{CA,B}$. If verification is successful, B checks the MAC on $Cert_{CA}(A)$ using $tK_{CA,0}$, which is derived from

$s_{CA,0}$ that is obtained from (12). If the MAC is correct, B obtains $s_{A,0}$ from $Cert_{CA}(A)$ by decrypting with $tK_{CA,0}$. B then obtains $s'_{A,0}$ from $s_{A,0}$:

$$s_{A,0} \xrightarrow{F_2} : s'_{A,0} \tag{13}$$

and subsequently B checks $MAC_{s'_{A,0}}(m_0)$ using $s'_{A,0}$ and accepts m_0 if the MAC verifies correctly. B also stores in memory the CA key broadcast message (and therefore $s_{CA,0}$), $Cert_{CA}(A)$ and the initial key $s_{A,0}$ of A's hash chain.

Messages from A to B in subsequent time intervals use the corresponding key of A's key chain to compute the MAC. A does not have to include its TESLA certificate in messages subsequent to M_0, under the assumption that every receiver has received M_0 correctly. For example, in the period $\langle t_i, t_i + \Delta \rangle$, message M_i from A to B would look like:

$$A \rightarrow B: \{M_i | M_i: \left(m_i, MAC_{s'_{A,i}}(m_i)\right)\} \tag{14}$$

At time $t_i + d$. A transmits $s_{A,i}$ to B. B can check the correctness of $s_{A,i}$ immediately by verifying $s_{A,i} \xrightarrow{F_1} s_{A,i-1} \xrightarrow{F_1} \ldots \xrightarrow{F_1} s_{A,0}$. Since $s_{A,0}$ has already been verified, and F_1 is a secure one-way function, the above check will verify that $s_{A,i}$ belongs to A's key chain. However, if B wants to be additionally careful, it can verify $s_{A,i}$ going through all the steps outlined above, using the CA key broadcast message and $Cert_{CA}(A)$.

Thus all the messages from A to B can be authenticated using low-computation symmetric MACs. A and B do not need to perform clock synchronization directly with one another (their clocks can be synchronized with their respective CAs), thereby saving on additional delay and protocol complexity (and possibly also on the cyclical dependency between authentication and clock synchronization).

The CAs need not be on-line all the time and do not need to broadcast frequent key disclosure messages. However, if the security policy demands so, the CA can periodically generate new TESLA certificates for a source, and broadcast periodic key disclosure messages. After the initial key disclosure message from the CA signed with the stored long-term shared secrets, subsequent key disclosure messages from the CA can be authenticated using one-way chains. For example, CA discloses the key $s_{CA,i}$ in period $\langle t_i, t_i + d \rangle$. Receiver B can verify that $s_{CA,i}$ belongs to CA's one-way chain: $s_{CA,i} \xrightarrow{F_1} s_{CA,i-1} \xrightarrow{F_1} \ldots \xrightarrow{F_1} s_{CA,0}$, where $s_{CA,0}$ has been verified previously from equation 12.

4.4. Revocation of TESLA Certificates

The CA might need to broadcast a certificate revocation message at any time circumstances warrant that the TESLA certificate of a node has to be revoked. Assume the CA revokes the TESLA certificate of node A in the time period $\langle t_i, t_i + d \rangle$. Then the CA broadcasts the following message to the network:

$$CA \rightarrow network: (\langle t_i, t_i + d \rangle, REVOKE(Cert_{CA}(A)), s_{CA,i}, MAC_{s_{CA,i+1}}(..)) \tag{15}$$

The receiver buffers the message and waits for the CA to disclose $s_{CA,i+1}$. The traffic received from A in the intermediate period is also buffered, awaiting the verification of the revocation message, due to the possibility that the revocation message might be a fake.

The CA discloses $s_{CA,i+1}$ with the next message it broadcasts to the network. The receiver can verify the authenticity of $s_{CA,i+1}$ and therefore the revocation message by verifying the correctness of the one-way chain:

$$s_{CA,i+1} \xrightarrow{F_1} s_{CA,i} \xrightarrow{F_1} .. \xrightarrow{F_1} s_{CA,0} \tag{16}$$

where $s_{CA,0}$ has been verified from the initial key disclosure broadcast message of the CA. If the revocation message is correctly verified, the receiver discards the buffered messages from A and adds the sender to the revoked users list.

The revocation message can be merged with the key disclosure message, the combined message can look like:

$$CA \rightarrow j: \{\langle t_i, t_i + d \rangle, REVOKE(..), s_{CA,i}, MAC_{s_{CA,i+1}}(..)\}_{K_{CA,j}} \forall j \tag{17}$$

where j is a receiver node, the REVOKE field will contain the TESLA certificates to be revoked, the MAC is computed on the revoked certificates and signing the message with the long-term secret $K_{CA,j}$ shared with node j verifies $s_{CA,i}$ for any node j that might need the verification.

Non-repudiation is not provided by the authentication algorithm we have described in this section. We are currently investigating efficient additions to the algorithm to provide this important feature.

5. SECURITY AND PERFORMANCE ANALYSIS

5.1. Security Analysis

The authentication protocol described in section 4 provides strong authentication and is resistant to active attacks by malicious nodes in the network. In the following, we discuss some attacks against the protocol. We assume that CA_e and CA_m are always secure, since compromise of the CA is a single point of failure in the network, and will nullify most security properties of our algorithm.

5.1.1. Malicious Node with Connectivity to Source and Receiver

We consider the case where a malicious node in the network attempts to create fake packets from a source to the receiver(s). We assume that the malicious node X can hear packet transmissions from the actual source A, and can also transmit to the receiver B. X can also receive the broadcast messages from the CA. Therefore, shortly after time $t_0 + d$, X has knowledge of $Cert_{CA}(A)$, message M_0 from A to B and $s_{CA,0}$ broadcast by CA_e. X can verify that $s'_{A,0}$ belongs to the authentication hash chain of A by performing the verification procedure of equation (13). Having obtained a verified element of A's authentication chain, X can attempt to spoof messages as coming from A, staring at time $t_0 + k\Delta$, where $\frac{d}{\Delta} = k$. To achieve this, X needs to generate $s_{A,k}$ from $s_{A,0}$ where $s_{A,k} = F_1^{-k}\{s_{A,0}\}$. Due to the one-way property of F_1, this is computationally infeasible for X, and is of complexity $O(2^K)$, where each element of the hash chain is K bits and K is assumed to be large. Without a valid $s_{A,k}$, it would be impossible for X to spoof a message that would be successfully authenticated by B.

X could also attempt to spoof packets from A at any time between $\langle t_0, t_0 + d\rangle$. This would require that X successfully generate an element of A's hash chain without knowledge of any legitimate element of the hash chain. This has the same computational complexity of $O(2^K)$ and is computationally infeasible for any X with finite resources.

Failing any attack on A's hash chain as above, X could attempt to masquerade as the CA and generate a fake certificate for A as in equation (7), and also generate fake CA key disclosure broadcast message similar to equation (12). However, unless X knows the long term secrets that CA_e shares with each node, it will not be able to correctly sign the fake $Cert_{CA}(A)$, and therefore the fake certificate will be rejected by A. Likewise, the fake CA broadcast message from X will be rejected by the receivers unless the message is signed using the long-term shared secret between the receiver and CA_m. As per our assumption of the security of the CA, the shared secret is known only to CA_m and the receiver, and therefore X would not be successful in this attack.

X could attempt to fake CA key disclosure messages subsequent to (12), but (a) the fake hash element $s_{CA_X,i}$ will not verify successfully to the anchor element $s_{CA,0}$ and (b) this still does not allow X to fake elements of A's hash chain. At best, X would be able to confuse the receivers temporarily and therefore launch a DoS attack for a limited time.

Once the CA has disclosed its TESLA key $s_{CA,i}$ for period $\langle t_i, t_i + d\rangle$, any node in the network can create a fake certificate, purportedly generated by the CA before $t_i + d$, by computing $tK_{CA,i}$ from $s_{CA,i}$. However, when the fake certificate is sent to any receiver, it will arrive after time $t_i + d$, and therefore be rejected, as per the security requirement in section 4.2.

5.1.2. Attack on the CA Revocation Messages

A malicious node X in the network can attempt to broadcast fake revocation messages, similar to (15), and thereby attempt to disqualify legitimate sources in the network. To generate a fake revocation message that will be successfully accepted by the receivers, X should be able to compute a MAC on the fake revocation message using the key $s_{CA,i+1}$, with knowledge of at most the key $s_{CA,i}$. Using reasoning similar to the previous section,

owing to the one-way property of F_1, this has computational complexity $O(2^K)$ and is infeasible for X. At most, X can trick the receivers in buffering the fake revocation message, till the next message disclosure from the CA, when the MAC on the fake message will not verify correctly using the recently disclosed (correct) $s_{CA,i+1}$, and therefore be discarded.

Based on the security analysis above, the extended TESLA certificate approach is secure against message spoofing attacks by malicious nodes in the network.

5.2. Performance Analysis

The performance of the authentication protocol described in section 4 is equivalent to the original TESLA certificate approach. Compared to public-key authentication, TESLA certificates offer significant savings in power expenditure and the time required to generate authentication parameters for the messages, and to verify the authenticity of messages. As shown in,[6] it requires 46 milliseconds to perform a SHA-1[8] MAC computation on a 4096-bit message using a Pentium-4 2GHz machine, while 2048-bit RSA signing requires 2.26 seconds using the same platform. Therefore the savings is of the order of 49 times when using TESLA certificates as opposed to RSA public-key authentication.

In our algorithm, neither the source nor the receiver has to perform any public-key operation. All the messages from the source to the receiver are authenticated using MACs computed on the messages. Compared to authentication using digital signatures, this represents a substantial savings in computation power and delay. Each message size from the source to the receiver is also smaller in our algorithm compared to using digital signatures. For example, using SHA-1, the authentication MAC for packet p is 160 bits, while using 1024-bit RSA key, the signature would be 1024 bits. Therefore, for each message, the authentication field is 6.4 times smaller using TESLA certificates, representing a substantial savings in network bandwidth over a large number of messages.

6. CONCLUSION

In this paper we have addressed the problem of end-to-end user authentication for a space mission network. We have made the case as to why user authentication and message integrity are necessary, and why it is important to design algorithms that are efficient in terms of expenditure of node resources. We have proposed a protocol for user authentication and message integrity in a lunar mission network, for the case where the source and the destination are widely distributed in the terrestrial network and lunar surface network, and analyzed its security and performance.

The protocol described here is a preliminary step in the process of analyzing and implementing security algorithms for space mission networks. Much remains to done in this regard, and we believe the protocol can be improved in various ways. We are currently working on a software simulation of the protocol and on fine-tuning its parameters to obtain optimal performance. We are also investigating alternative efficient strategies for authentication algorithms, and looking at the issue of user authentication in group communication in the space network. We believe that this paper will serve as an useful contribution as further research is done for security algorithms in space networks.

ACKNOWLEDGMENTS

The research work reported here is partially supported by the National Aeronautics and Space Administration (NASA) Marshall Space Flight Center, under award number NCC8-235. The views expressed in this paper are solely the responsibility of the authors and do not reflect the views or position of NASA or any of its components.

The authors would like to thank Nicolas Rentz of INP Grenoble Telecom, France, for help with the network topology setup.

REFERENCES

1. A. Roy-Chowdhury, N. Rentz, M. Hadjitheodosiou, and J. Baras, "Modeling and design of a communication architecture supporting lunar exploration," in *to be published. 23rd AIAA International Communications Satellite Systems Conference And Exhibit 2005 (ICSSC 2005)*, AIAA, (Rome, Italy), September 2005.
2. A. Roy-Chowdhury, N. Rentz, M. Hadjitheodosiou, and J. Baras, "Hybrid networks with a space segment - topology design and security issues," in *to be published, IEEE Military Communications Conference (MILCOM) 2005*. IEEE, (Atlantic City, USA), October 2005.
3. A. Roy-Chowdhury and J. Baras, "A certificate-based light-weight authentication algorithm for resource-constrained devices." Tech. Rep. CSHCN TR 2005-4, Center for Satellite and Hybrid Communication Networks, University of Maryland College Park, 2005.
4. A. Perrig, R. Canetti, D. Song, and J. D. Tygar, "The tesla broadcast authentication protocol," *RSA Cryptobytes* , Summer 2002.
5. A. Perrig, R. Canetti, D. Song, and J. D. Tygar, "Efficient and secure source authentication for multicast," in *Proc. Network and Distributed System Security Symposium (NDSS)*, 2001.
6. M. Bohge and W. Trappe, "Tesla certificates: an authentication tool for networks of compute-constrained devices," in *Proc. of 6th international symposium on wirless personal multimedia communications (WPMC '03)*, (Yokosuka, Kanagawa, Japan), October 2003.
7. M. Bohge and W. Trappe, "An authentication framework for hierarchical ad hoc sensor networks," in *Proceedings of the 2003 ACM Workshop on Wireless Security (WiSE'03)*, pp. 79–87, ACM, (San Diego, USA) August 2003.
8. "Secure Hash Standard." `http://www.itl.nist.gov/fipspubs/fip180-1.htm`