Distributed Small Sat Location Verification

Kalabic, Uros; Weiss, Avishai; Chiu, Michael

TR2021-033 April 28, 2021

Abstract

We provide a scheme for verifying satellite locations, where the locations are reported by satellites themselves. In the scheme, satellites periodically report their locations to the network and these are included in a list. The locations are occasionally verified through a type of proof-of-location protocol. A proof of location is provided via the solution to a cryptographic puzzle and adherence to geometric constraints. The cryptographic puzzle is constructed as a layered data packet, where each layer can be removed only by a specific satellite. This by itself ensures that the challenge was completed in the requisite order. As part of the response to the challenge, each satellite reports its position and time. These reports are verified against the physical constraints of the speed of light and radio signal strength decay. The scheme secures location reports and proofs of location using a permissioned blockchain. Having a source of truth allows locations to be verified retrospectively, which is important due to asynchronous communication between satellites. Satellite reports can arrive out of order and updating past observations with more reliable information is essential to achieving consensus on the veracity of location reports. Although location verification could be implemented as a centralized scheme, decentralization enables scalability and trustless cooperation between satellite operators, potentially reducing costs in deployment and operation of satellite constellation.

Integrated Communications Navigation and Surveillance (ICNS) Conference

Mitsubishi Electric Research Laboratories, Inc.
201 Broadway, Cambridge, Massachusetts 02139

DISTRIBUTED SMALL SAT LOCATION VERIFICATION

Uroš Kalabić, Mitsubishi Electric Research Laboratories, Cambridge, Massachusetts
Avishai Weiss, Mitsubishi Electric Research Laboratories, Cambridge, Massachusetts
Michael Chiu, Department of Computer Science, University of Toronto, Canada

Abstract

We provide a scheme for verifying satellite locations, where the locations are reported by satellites themselves. In the scheme, satellites periodically report their locations to the network and these are included in a list. The locations are occasionally verified through a type of proof-of-location protocol. A proof of location is provided via the solution to a cryptographic puzzle and adherence to geometric constraints. The cryptographic puzzle is constructed as a layered data packet, where each layer can be removed only by a specific satellite. This by itself ensures that the challenge was completed in the requisite order. As part of the response to the challenge, each satellite reports its position and time. These reports are verified against the physical constraints of the speed of light and radio signal strength decay.

The scheme secures location reports and proofs of location using a permissioned blockchain. Having a source of truth allows locations to be verified retrospectively, which is important due to asynchronous communication between satellites. Satellite reports can arrive out of order and updating past observations with more reliable information is essential to achieving consensus on the veracity of location reports. Although location verification could be implemented as a centralized scheme, decentralization enables scalability and trustless cooperation between satellite operators, potentially reducing costs in deployment and operation of satellite constellation.

Introduction

The size of small sats is regulated by the Federal Communications Commission (FCC) to ensure that they are large enough to be tracked in orbit. The current streamlined FCC approval process [1] requires that satellites be no smaller than 10cm in any dimension; therefore, to be eligible, the smallest satellite can be no smaller than 1U. However, for some applications, it is possible and even desirable to reduce the size of satellites even further, as smaller satellites increase access to space for both nations and corporations [2]. The increased access to space is especially enabled when small sats are part of constellations, and are able to perform tasks that are conventionally done by fewer, larger satellites.

Constellations themselves, due to their numbers and regardless of their size, place an additional burden on maintaining space situational awareness. According to current plans, several constellations will become operational over the coming decade and substantially increase the number of satellites in Earth orbit, at the very least quintupling the total currently in orbit [3]. This expected, exponential growth of small objects in space underlies the importance of developing novel techniques for improving trackability of large amounts of small objects in space.

In this work, we propose a novel scheme for reducing reliance on ground-based tracking of satellites, in which satellites self-report their locations to a blockchain-secured network and these locations are periodically verified by other satellites using telemetry. Decentralized verification of location reporting is a recent technology, first deployed in 2019 by Helium Systems [4]. The Helium network is composed of ground-based routers providing wireless coverage to IoT devices. The deployment of the Helium network is decentralized and so routers need to periodically prove their location in order to verify coverage.

Our scheme is similar to that of Helium, with several modifications. The primary difference is the implementation of a prediction algorithm based on open-loop orbital mechanics, necessary because satellites move rapidly in orbit and therefore require occasional correction to their reported locations. Another difference is that our blockchain protocol is permissioned [5], allowing only authenticated parties to participate in the verification protocol. Permissioning and a closed network are preferred because the satellite operators are parties that are mostly trusted, do not require pseudonymity, and do not need to be explicitly rewarded for performing location reporting. Permissioning also allows us to explicitly prevent collusion between satellites having the same operator, something that is not done in a fully distributed network because, in this case, operators typically own only a small portion of devices and these networks are sufficiently heterogenous.

Concept

Satellites in Earth orbit are currently monitored using ground-based systems [1]. This would be unnecessary if their location could be verified, with high confidence, by some other means. We propose a blockchain-based, distributed system of verifiable location reporting, in which a satellite reports its location to the system, and the location is verified. A schematic of the concept is provided in Figure 1. The figure shows that location verification is a kind of predictor-corrector scheme, commonly used in estimation. Conceptually, the location report is an a-priori estimate; the verification provides an innovation that determines confidence in the a-posteriori estimate.

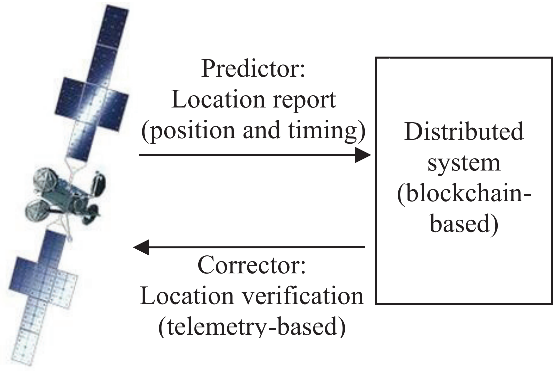

Figure 1. Schematic of Distributed Location Reporting and Verification

The location report consists of four variables: a three-dimensional position vector and a time stamp. The method of verification is tasked with verifying these variables, to some level of confidence. This is done by requiring satellites to periodically send telemetry signals between each other which, through the geometry of bilateration, can be used to determine the veracity of a location report. In effect, the proposed scheme makes satellites police themselves.

More specifically, the ones doing the policing are the satellite operators who continually challenge each other to ensure that all operators are being truthful. For this reason, in this work we require that there be at least three satellite operators so that no operator is simultaneously performing the task of providing a challenge, verifying position, and verifying the challenge and subsequent verification. The operators themselves may operate any number of satellites in constellations.

This trustless framework utilizes a blockchain as a source of trust. The use of a blockchain serves as a record of verified positions, which is made immutable through the implementation of a consensus protocol. In Figure 2, we provide a schematic of the information contained in the blockchain, which serves as the source of trust. In the figure, blocks contain a ledger of reported locations and challenges, which are cryptographic puzzles that verify location reports.

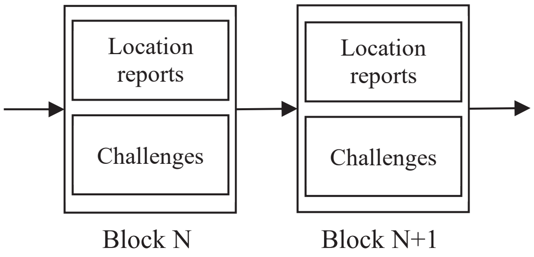

Figure 2. Schematic of Blockchain Ledger

Although it may be adequate to pass the responsibility of verification to a central entity, similar to the current, centralized practice of optical tracking, a distributed protocol gives the advantage of robustness and scalability:

- Robustness: There is no single point of failure; for example, the centralized system cannot give unfair advantages to favored participants.
- Scalability: Computing power scales proportionally to the size of the network; a distributed design enables more effective cooperation between participants, not needing to rely on a central service.

We have thus described the concept behind our distributed location verification scheme. In the following, we provide a survey of current technology along with a few additional considerations that are outside the scope of the present work. The rest of the paper describes our scheme in detail.

Technology Survey

This work is generally in the field of distributed systems applied to physical processes. Distributed systems are typically more reliable and scalable than centralized systems but introduce the need to provably achieve agreement, which is done through consensus protocols [6]. Conventionally, distributed systems are not directly connected to a physical process, although this has changed in recent years with the advent of blockchain technology.

Since the invention of the cryptocurrency Bitcoin [7], which was the first application of blockchain, many applications have been proposed in which the blockchain is used to connect a physical process to the digital domain. Bitcoin itself does this through the proof-of-work protocol, which requires the solution to a cryptographic puzzle, solvable only by expending computing resources. Not all applications of blockchain are tied to physical processes, but it is this aspect of blockchain and consensus protocols that is of interest to designers of large-scale distributed, physical systems.

Particularly interesting is the development of useful-work consensus algorithms [8]. Useful-work algorithms are so called in order to make a distinction from the nature of Bitcoin's proof-of-work algorithm, which is perceived as not being useful because it consumes a large amount of electric power. The idea behind useful work is to utilize an underlying physical process to serve within a consensus framework, i.e., to take advantage of work that is already being performed for some purpose other than consensus. The definition of "useful" is relative, as the Bitcoin algorithm might be perceived as useful to some. Primecoin [9] was the first to implement an alternative proof-of-work, in which it used the work required to compute prime numbers to secure the blockchain. Later, Filecoin [10] implemented an algorithm that serves as a proof of computer storage. Since then, there have been more proposals, the most successful one of which has been that of Helium Systems and its proof-of-location protocol, which secures what is currently the world's largest contiguous IoT network.

Helium uses proof of location to determine the veracity of location reports of routers within its network. This is necessary in order to prevent spoofing, since routers are paid in cryptocurrency and bad actors have an incentive to make a monetary gain through false reporting. In the Helium network, routers are stationary and only report their position to the network at setup. These positions are regularly verified by the network through cryptographic puzzles that can only be solved by sending a signal between routers in some required order. The puzzle is called a guided tour [11], [12], and the time taken to complete the tour should, under truthful reporting, be comparable to the tour length divided by the speed of radio. In the atmosphere, radio signals travel almost at the speed of light and their strength diminishes inversely proportional to about the square of the distance [13]. The proof of location is therefore able to take advantage of physical certainties to construct a proof, avoiding the need to perform additional work. This useful, physical work is then used as a basis of trust in the blockchain, upon which Helium runs its IoT network.

It is commonly understood in the aerospace community how multilateration can be used to determine position and timing as this is the basic principle behind Global Navigation Satellite Systems (GNSS) [14], such as the Global Positioning System (GPS). Other work, which uses ground-based routers for bilateration, includes [15], [16].

This work is also generally part of a growing body of work that seeks to apply blockchain to space applications [17], [18]. Such works include the work of Hyland-Wood et al. [19], [20], which take a broad perspective of blockchain in space and situate the usefulness of blockchain for space applications; some works [21], [22] discuss the use of blockchain for space asset tracking and improving space situational awareness, but do this in an ordinary permissioned framework, lacking verifiability; other works considering space include [23], [24]; the work of [25] considers the application of blockchain to the space economy and it is expanded upon in [20]; an innovative, non-academic approach, is that of SpaceChain [26], which seeks to be a space-based computing platform; our preliminary work in small sat location verification is presented in [27].

Other Considerations

The scheme we propose has additional requirements not covered by location reporting and its verification. Specifically, because a location must be determined before it can be reported, it requires a method for determining position, navigation, and timing (PNT); because verification is an active process, it requires that the process be online and that the satellites stay powered, i.e., there cannot be a total loss in communication. Although these requirements are outside of the scope of our work, we provide a discussion of how they may be addressed in the following.

PNT

The obvious and most convenient choice of obtaining PNT is the use of GNSS. GNSS is an accurate provider of PNT which generally works by quadrilateralizing telemetry signals and running error correction. The telemetry-based operation of GNSS suggests that our verification method could be implemented on top of GNSS-based methods. While this topic may be worthy of consideration, we do not explore it here.

Active Communication

The main advantage of optical tracking is that it is a method of passive surveillance, where the target does not need to cooperate. The scheme that we propose requires cooperation from the target, however, and this implies that it is not a suitable solution to all modes of operation, such as monitoring of space debris or noncooperative satellites. We therefore suggest that our scheme provide support to current space monitoring systems.

The need for active cooperation also implies that distributed monitoring of a satellite's location is not possible after a total failure. To increase reliability and minimize the possibility of total failure, we propose that our scheme be implemented on ruggedized, specialized hardware that runs independently. We expect that the additional costs incurred in implementation would be more than offset if its use would allow satellite operators to launch and operate larger numbers of smaller satellites.

Verified Location Reporting

As shown in Figure 1, location verification follows a predictor-corrector format, where the location report is offered as a first (a-priori) estimate, and the verification results in a corrected (a-posteriori) estimate. In this section, we describe this dynamic.

Prediction

Regardless of verification, satellite locations must obey the laws of physics. In this case, orbital dynamics governing the motion of a satellite i in Earth orbit are given by the ordinary differential equation [28],

$$\ddot{x}^i(t) + \frac{\mu}{\|x^i(t)\|^3} x^i(t) = a^i\big(x^i(t), t\big), \tag{1}$$

where $x^i(t)$ is the position in an inertial frame of reference, t is time, the independent variable, and $\mu = 3.97 \cdot 10^5$ is Earth's gravitational parameter; the forcing function a^i includes all accelerations acting on the satellite due to orbital perturbations or satellite thrust. Assuming that $a^i = 0$, the satellite trajectory traces out a conic section (a circle, ellipse, parabola, or hyperbola). In reality, perturbation forces can be assumed to be small over short time horizons and, with frequent enough correction to any estimate, Equation 1 should adequately predict the motion of the satellite when $a^i \approx 0$. We therefore disregard the acceleration component in the development of our scheme.

We discretize Equation 1 using the central difference method and time step h to obtain the difference equation,

$$x^i_{k+1} - 2x^i_k + x^i_{k-1} + \frac{h^2\mu}{\|x^i_k\|^3} x^i_k = 0, \tag{2}$$

where $x^i_k \triangleq x^i(t_k)$ is the position $x^i(t)$ at time $t = t_k$, where the sequence t_k satisfies $t_{k+1} = t_k + h$. This discretization preserves geometric properties of the continuous time dynamics of Equation 1 [29] and we use it with the expectation that it will remain relatively accurate over longer time horizons h. Note that different numerical methods may be used to reduce prediction error.

Rearranging Equation 2, we can form an a priori estimate for satellite position. That is, given the two positions x_k^i and x_{k-1}^i, the orbital dynamics predict the position at the following time instant x_{k+1}^i to be,

$$\hat{x}_{k+1}^i \triangleq \left(2 - \frac{h^2\mu}{\|x_k^i\|^3}\right) x_k^i - x_{k-1}^i. \tag{3}$$

The estimated position $\hat{x}_{k+1}^i$ can be compared to a self-reported location provided by the satellite itself. We should expect that a report of position $\bar{x}_{k+1}^i$ be close to the corresponding estimate $\hat{x}_{k+1}^i$. However, we cannot fully rely on self-reporting to correct the estimate because the dynamics are known and an operator could always set $\bar{x}_{k+1}^i = \hat{x}_{k+1}^i$. For this reason, any location report must be verified.

Verification

The network is required to verify self-reported positions to check for errors and false reporting. A verification is performed by confirming the position to neighboring satellites.

Bilateration

Given a target satellite i_0, the network determines two verifier satellites i_- and i_+ within line of sight (LOS) of the target. An LOS exists if there exists a chord connecting i_0 and $i_\pm$. This is true as long as the length of the average of the inertial positions $x_k^{i_0}$ and $x_k^{i_\pm}$ is greater than the Earth's radius r_E, i.e., $\left\|x_k^{i_0} + x_k^{i_\pm}\right\| > 2r_E$.

The verification is performed using bilateration. The first verifier i_- broadcasts a two-layer encrypted message containing its position and time $\left(y_c^{i_-}, t_c^{i_-}\right)$ to the target, whose first layer only the target can decrypt. When the target receives the message, it records its own position and time $\left(\hat{y}_c^{i_0}, \hat{t}_c^{i_0}\right)$ and the signal strength $\hat{m}_0$ and decrypts the first layer; it then passes along the remainder of the encrypted message to the second verifier i_+ along with its own position and time $\left(y_c^{i_0}, t_c^{i_0}\right)$. When the verifier receives the message, it records its own position and time $\left(\hat{y}_c^{i_+}, \hat{t}_c^{i_+}\right)$ and the signal strength $\hat{m}_+$, and decrypts the remainder.

Cryptographic Puzzle

Bilateration requires a proof that the signal was sent and received by the required satellites. As mentioned above, the puzzle is constructed in layers, using the onion-like protocol that enables the Helium proof-of-location service [4]. In the protocol, the challenge c is constructed by the network as a data packet consisting of three layers, shown in Figure 3.

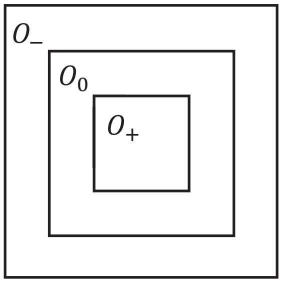

Figure 3. Layered Encrypted Challenge [4]

The layers are given as,

$$O_b \triangleq E_{\check{s}\kappa_b}(\nu_b, \tau_b, O_{b++}),\ b \in \{-,0,+\}, \tag{4}$$

where $E_{\check{s}\kappa_b}$ is the encryption function, encrypted with the shared key $\check{s}\kappa_b$, $\nu_b \triangleq E_{p\kappa_b}(S_b)$ is the nonce S_b, corresponding to the satellite i_b, encrypted with the satellite's public key $p\kappa_b$, and τ_b is the time at which the challenge is first supposed to be executed, so that $t_c^{i_-} \approx \tau_b$. Note that, for the final layer, O_{++} is empty.

The method of encryption is outside the scope of this work. We note that a standard set of encryption tools is available to achieve this purpose [30].

Challenge Receipt and Response

When a satellite i_b receives a challenge, it records the time and the signal strength at which it arrived and attempts to decrypt it with its private key. If the result is uninterpretable, then the satellite may still be a witness, described below.

If the result is interpretable, it means that the private key forms a pair with the public key with which the layer was encrypted, i.e., $(p\kappa_b, s\kappa_b)$ is a public-private key pair. In this case, i_b removes the layer and decrypts ν_b to discover the nonce S_b. The satellite then creates a receipt which consists of a hash of the nonce and other relevant information. See Figure 4 for an illustration.

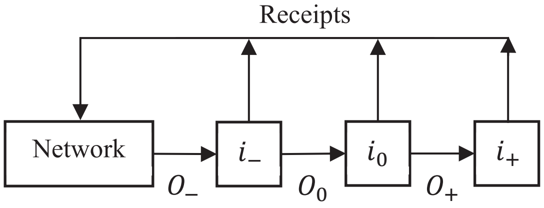

Figure 4. Schematic of Challenge Receipt and Response [4]

Physical Constraints

The physical properties of radio signals imply that,

$$\left\|\hat{y}_c^{i_0} - y_c^{i_-}\right\| \approx c\left|\hat{t}_c^{i_0} - t_c^{i_-}\right|, \tag{5a}$$

$$\left\|\hat{y}_c^{i_+} - y_c^{i_0}\right\| \approx c\left|\hat{t}_c^{i_+} - t_c^{i_0}\right|, \tag{5b}$$

since radio travels through vacuum at the speed of light c, and,

$$\frac{\widehat{m}_0}{\left\|\hat{y}_c^{i_0} - y_c^{i_-}\right\|^2} \approx \frac{m_*}{d_*^2}, \tag{6a}$$

$$\frac{\widehat{m}_+}{\left\|\hat{y}_c^{i_+} - y_c^{i_0}\right\|^2} \approx \frac{m_*}{d_*^2}, \tag{6b}$$

since radio signals decrease inversely proportional to the inverse square of distance in a vacuum [13]. The value m_*/d_*^2 is the strength of signal empirically determined at distance d_*.

Due to noise, the expressions in Equations 5 and 6 are approximations rather than strict equalities. Verification is not meant to be precise, but rather to measure veracity in reporting; therefore, some error is acceptable.

Trusting the output of the verifiers, Equations 5 and 6 give only four equations for nine unknowns: six in $\hat{y}_c^{i_0}$ and $y_c^{i_0}$, two in $\hat{t}_c^{i_0}$ and $t_c^{i_0}$, and one in $\widehat{m}_0$. To reduce the number of unknowns to five, we require that the target transmit the message almost instantaneously after receipt, which results in,

$$\hat{y}_c^{i_0} = y_c^{i_0}, \tag{7a}$$

$$\hat{t}_c^{i_0} = t_c^{i_0}. \tag{7b}$$

Note that Equation 7 is not approximate because this is enforced on the target.

The system of Equations 5-7 is underdeterminate by one equation. It can become determinate, or even overdetermined, by introducing witnesses to the verification protocol. We describe these in the following.

Witnesses

It is possible for any satellite within LOS of the target to help with verification of the target. For those that are not chosen to verify a target's location, they may still record receipt of messages transmitted from the target without having to decrypt the second layer. The encrypted message has only to be signed by a verifier, but more knowledge about position can be used to increase trust in, along with robustness of, location verification. Therefore, when a witness $i_w \notin \{i_0, i_\pm\}$ receives a message from the target, it can record its own position and time $\left(\hat{y}_c^{i_w}, \hat{t}_c^{i_w}\right)$ and the signal strength $\widehat{m}_w$ to yield two additional relationships,

$$\left\|\hat{y}_c^{i_w} - y_c^{i_0}\right\| \approx c\left|\hat{t}_c^{i_w} - t_c^{i_0}\right|, \tag{8a}$$

$$\frac{\widehat{m}_w}{\left\|\hat{y}_c^{i_w} - y_c^{i_0}\right\|^2} \approx \frac{m_*}{d_*^2}. \tag{8b}$$

Equation 8 holds for any witness i_w and, as long as there is at least one witness, the system of Equations 5-8 is overdetermined and can therefore be used to determine confidence in the location report $\left(y_c^{i_0}, t_c^{i_0}\right)$.

Confidence

The confidence in a location report is given as the weighted mean-square error of the relationships given in Equations 5, 6, and 8,

$$\Gamma = q_d\left(e_d^2 + q_w e_{w,d}^2\right) + q_s\left(e_s^2 + q_w e_{w,s}^2\right), \tag{9}$$

where,

$$e_d^2 = \sum_{b=-,0}\left(\left\|\hat{y}_c^{i_{b++}} - y_c^{i_b}\right\|^2 - c^2\left|\hat{t}_c^{i_{b++}} - t_c^{i_b}\right|^2\right)^2,$$

$$e_s^2 = \sum_{b=-,0}\left(\widehat{m}_{b++}d_*^2 - m_*\left\|\hat{y}_c^{i_{b++}} - y_c^{i_b}\right\|^2\right)^2,$$

$$e_{w,d}^2 = \frac{1}{|W|}\sum_{w\in W}\left(\left\|\hat{y}_c^{i_w} - y_c^{i_0}\right\|^2 - c^2\left|\hat{t}_c^{i_w} - t_c^{i_0}\right|^2\right)^2,$$

$$e_{w,s}^2 = \frac{1}{|W|}\sum_{w\in W}\left(\widehat{m}_w d_*^2 - m_*\left\|\hat{y}_c^{i_w} - y_c^{i_0}\right\|^2\right)^2,$$

and q_d, q_s, and q_w are weights and W is the set of witnesses.

If the confidence Γ is within a predetermined bound Γ^*, the network accepts the target's location report $\left(y_c^{i_0}, t_c^{i_0}\right)$. If it is not, then the report is rejected.

Trust in Verification

The network logs reports provided by verifiers $\left(y_c^{i_-}, t_c^{i_-}\right)$, $\left(\hat{y}_c^{i_+}, \hat{t}_c^{i_+}\right)$ and witnesses $\left(\hat{y}_c^{i_w}, \hat{t}_c^{i_w}\right)$ as a bona fide location reports, which are themselves subject to verification. This avoids the need to blindly trust verifier and witness reports, and provides a source of location reporting.

Consensus in Location Reports

At every time-instant and for every satellite i, there are four possibilities. The location report was either 1) self-reported, in which case it was either 1a) verified, 1b) not verifiable, or 1c) not verified, or it was 2) not reported. The four cases are shown in Figure 5, in increasing order of trust.

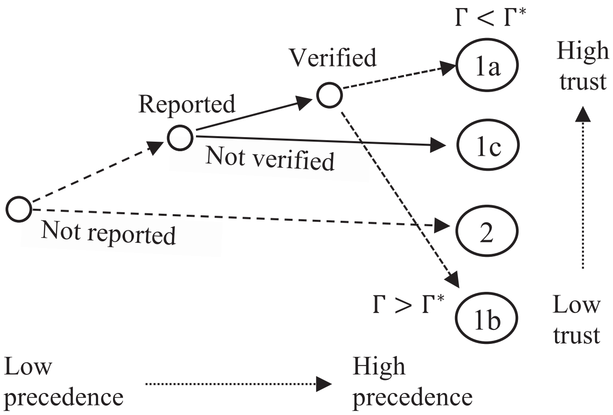

Figure 5. Trust and Precedence of Location Reporting

As the figure shows, the highest trust case is when a location report has been verified within the required level of confidence; the lowest trust case is when a location report has been verified to be outside the confidence bound. Therefore, the verification procedure is the main arbiter of trust in the network.

The network assigns a location to the satellite i differently depending on the case:

- Self-reported; verified

In this case, the network uses the challenge-verified location report. Given a location report $\left(y_c^i, t_c^i\right)$, given at $t_c^i \in (t_k, t_{k+1}]$ by target satellite i, the network determines the location $x_{k+1}^i = \check{x}_{k+1}^i$ at time t_{k+1} according to the weighted center difference formula,

$$\check{x}_{k+1}^i \triangleq \left(\frac{2}{1+\delta} - \frac{(1-\delta)h^2\mu}{\|X\|^3}\right)X - \frac{1-\delta}{1+\delta}x_{k-1}^i, \tag{9}$$

where $X = y_c^i$ and $\delta = (t_c^i - t_k)/h$.

- Self-reported; unverified

In this case, the network uses the self-reported estimate $\bar{x}_{k+1}^i$ from the satellite, which is provided along with a time $\bar{t}^i$. The update is then set to the position obtained using Equation 9 where $X = \bar{x}_{k+1}^i$ and $\delta = (\bar{t}^i - t_k)/h$.

- Unreported or unverifiable

In this case, the network either does not have a location report, or cannot trust the one that it has; therefore it sets $x_{k+1}^i = \hat{x}_{k+1}^i$, where the expression for $\hat{x}_{k+1}^i$ is given in Equation 3.

We have thus far described one step of the protocol for determining a consensus location for each satellite. This step relies on the correctness of the network state, which consists of all satellite positions at all times. In the following, we introduce the concept of trust, which will be used to ensure confidence in reported locations.

Trust Score

The trust score measures the closeness of a location report to that predicted by the network. Generally, closeness can be mathematically defined using any metric d, where $d(x,y) \leq d(x,z)$ implies x is closer to y than z. Since trust in location reporting is related to distance, the metric we use to measure trust is the common Euclidean metric, which measures Euclidean distance,

$$d_k^i = d\left(x_k^i, \hat{x}_k^i\right) = \left\|x_k^i - \hat{x}_k^i\right\|. \tag{10}$$

The goal of each satellite is then to minimize large and sustained deviations in the distance measure. A simple requirement enforcing this is that the distance metric not surpass some bound,

$$d_k^i \leq D^*, \tag{11}$$

at any time t_k.

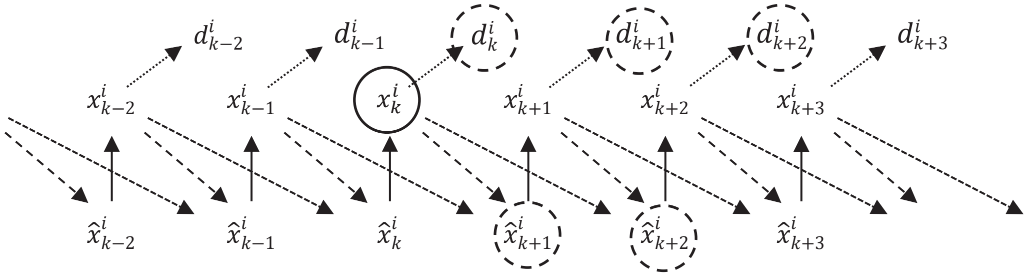

Figure 6. Challenges Acting as Anchor of Location List

Multiple Reports

The concept of trust score also allows us to create an ordering of location reports. To ensure robustness in reporting, we accept the location report with the lowest trust, i.e., the highest trust score.

Location List

Satellite positions x_k^i and the associated, a priori estimates $\hat{x}_k^i$ are published to a list, shown in Figure 6. The relationships between variables are given in Equations 3 and 10 and shown in Figure 5. Figure 6 shows that the estimate is a function of the prior position (m-dashed arrow) and the second to prior position (n-dashed arrow). The list logs position estimates and self-reported positions, comparing them to each other (solid arrow) and determining trust scores according to the norm of their difference (dotted arrow).

The positions are determined as one of the four possibilities shown in Figure 5. Specifically, since the reports are obtained asynchronously, a position x_k^i is logged according to the best information available. This position is overwritten if more credible information is received later. The order of preference of each case is given as:

- Reported, verified: 1a) $x_k^i = \check{x}_k^i$ with $(X, \delta) = (y_c^i, (t_c^i - t_k)/h)$
- Reported, unverifiable: 1b) $x_k^i = \hat{x}_k^i$
- Reported, unverified: 1c) $x_k^i = \check{x}_k^i$ with $(X, \delta) = \left(\bar{x}_k^i, (\bar{t}^i - t_k)/h\right)$
- Unreported: 2) $x_k^i = \hat{x}_k^i$

The precedence is shown in Figure 5. Note that in both cases 1b) and 2), the position is logged as the estimate $\hat{x}_k^i$; also note that an estimate is either 1a) verified or 1b) unverifiable, so there is no conflict in precedence between the two cases.

Therefore, at every update of the blockchain, performed at t_k, if no report has been received, the position is logged as the estimate $\hat{x}_k^i$. If a report is received, it overwrites the estimate. If the report is unverified, it is overwritten by receipt of a challenge, which certifies whether a report is verified or unverified. In this way, challenges are used to anchor trust in location reports as the protocol is retrospective and may uncover false reporting any time after the fact.

Although it is possible to give responsibility of state determination to a central entity, as discussed in the section describing the concept, distributed systems have advantages of robustness and scalability. In the following section, we describe the blockchain protocol that enables a distributed location reporting system.

Blockchain

The blockchain runs as a distributed computing protocol, where the responsibility of running the blockchain falls to the participants, i.e., the constellation operators. Each participant is also responsible for ensuring truthful location reporting through either reporting of their own satellites' locations, or through verifying others'. The protocol is designed to perform these tasks and distribute responsibility amongst participants.

The blockchain is used as a source of truth. Having a trustless source of satellite locations enables us to determine confidence in individual reports and assign a trust value to each satellite. The trust value we assign is a score that depends on the discrepancy between self-reported and network-verified locations; the protocol for doing so has been described in the previous section. The score is secured by cryptography and the physics of orbital dynamics and can be used to base the determination of network permissions and continued participation for individual satellites and their operators. The only way to increase a trust score is through regular, verifiable reporting of location and, since the trust score positively impacts network participation, truthful location reporting is the only way to ensure continued network participation. We begin by describing how trust is tracked, before describing the blockchain protocol.

Trust Measures

To ensure continued network participation, a satellite must report its location truthfully and participate in challenges. The location report score V_k^i for a satellite i is given as the count of violations of Equation 11 over a moving window of length N,

$$V_k^i = \sum_{\ell=0}^{N} \#\{d_{k-\ell}^i > D^*\}, \tag{12a}$$

with the window having been introduced to ensure forgetfulness. The score is therefore incremented every time a location report is deemed false by the network and decremented when the false report is forgotten. The score is upper-bounded as $V_k^i \leq V^*$, so that a violation of this upper bound results in censure and loss of privileges in the network.

Challenges and participation in them are also regulated by a count. This must be done to ensure that satellites are incentivized to participate in challenges because, without participating in challenges, the network defaults to assuming that a satellite's true location is equal to its estimate, which implies a perfect trust score. The protocol therefore keeps track of the following:

- Challenges failed
- Challenges witnessed
- Failures to participate in a challenge

The first is kept track of through a count,

$$F_k^i = \sum_{\ell=0}^{N_F} \#\{\Gamma > \Gamma^*\}, \tag{12b}$$

and the latter two are kept track of through a coupled count,

$$P_k^i = \sum_{\ell=0}^{N_P} \#\{\text{DNP}\} - a_w \#\{\text{witnessed}\}, \tag{12c}$$

where a weight $a_w \ll 1$ is assigned for acting as a witness, so that acting as a witness can counteract the negative effect of not participating in a challenge when instructed to by the network. The counts N_F and N_P are the windows corresponding to the number of challenges as target and the number of challenges as verifier, respectively.

All counts in Equation 12 are upper-bounded, by V^*, F^*, and P^*, respectively and a violation of this upper bound results in censure and loss of privileges in the network.

Challenge Construction

Challenges are issued randomly, with each challenge made repeatable by generating the random choice of satellite using verifiable entropy [31]. Challenges are issued so that a verifier and its target do not have the same operator. At the same time, and to keep verifiers honest, witness reports are accepted only from witnesses whose operators did not participate in the challenge. Challenges are issued so that, on average, all satellites will be visited at least once over the window N. Although this doesn't guarantee that a satellite would be visited at all, it does provide a framework in which one is able to calibrate parameters, such as the window N and weighting parameters, to improve performance of the protocol. The protocol parameters are listed in Table 1 along with their descriptions.

Table 1. Parameters

Parameter	Description
Γ^*	Position confidence bound
q_d, q_s, q_w	Confidence measure weights (Equation 9)
D^*	Trust score bound
V^*	Cumulative trust score bound
F^*	Bound on challenges failed
P^*	Bound on failure to participate
a_w	Witness participation incentive
N, N_F, N_P	Moving windows in counts

Protocol

The protocol is a permissioned blockchain run by constellation operators. Permissioned blockchains are appropriate for space because it is relatively straightforward and secure to register each operator and device for inclusion into the blockchain [19-22]. This is because, at present, satellites already go through a registration process, and it is also not possible to physically tamper with satellites without being detected. Nevertheless, the network must be protected from faulty or malicious communication, specifically Byzantine fault.

To provide Byzantine fault tolerance (BFT), we implement the conventional practical BFT (PBFT) [32] algorithm of Hyperledger Fabric [33]. The blockchain represents the state of the system, which consists of:

- Location reports
- Issued and completed challenges
- Witness reports

which have been described in the previous section. These are collected and included in blocks, using the Hyperledger Fabric ordering service. This ensures a source of truth for the procedures that run on top of the blockchain, including computation of trust scores and challenge construction.

Security

The blockchain is a trusted source of truth within the network. The network uses this information to assign a trust score to each satellite. In contrast to systems that are purely software-based, relying on dynamics makes continual equivocation, i.e., spoofing, almost impossible due to the requirement of eventual participation and the non-zero probability that faulty behavior will be identified. Moreover, satellites cannot actively, but only passively, increase their trust scores. Therefore, the use of verifiable entropy to determine the satellites in challenges ensures that continued equivocation will eventually be discovered.

The protocol itself is flexible and, by calibrating the parameters given in Table 1, the design is able to achieve the required level of trust. Note that the amount of precision is likely limited to some extent due to limitations in capabilities of sensors and telemetry.

Discussion and Future Work

Our protocol attempts to incentivize participation in consensus through trust. Specifically, access to space is controlled by the network, which distributes keys and registers assets onto the blockchain. Consistent maintenance of the trust score above a certain threshold and participation in challenges ensures good standing within the network. The possibility of access revocation should lead participants to adhere to the protocol of truthfully reporting their location, within some tolerance.

The main advantage of decentralization is the potential for scalability. By securing trust in location reporting, we provide a trusted platform upon which one may be able to base other protocols. As in Helium, one possibility that our location reporting protocol may enable is software-defined networking for satellites, where satellites, depending on their position, may function as routers or dedicated computational units; this would enable virtualized networks that are agnostic to which particular satellite acts as a packet router or local compute node. Furthermore, the blockchain can hold additional information such as proofs of various work done by the satellite, in addition to location reports, which would enable the commoditization of satellite constellations, since the responsibilities of running a satellite constellation and developing applications on top would be separated. This is particularly attractive, since launching satellites is very costly, and it may be better for operators to cooperate in a trustless manner than lose effort on competition.

Given the required computational power to run the blockchain and verify trust in location reporting, the main computational burden would likely be borne by ground-based servers. For this reason, we do not expect the potential size of a blockchain to be an impediment to the adoption of the underlying technology. If size is an issue, then we expect that one may be able to implement some of the novelty from projects like Mina [34], which uses zero-knowledge proofs to perform computations on the state transitions of the blockchain, ensuring that it stays constant in size.

As part of future work, we plan to implement and test the framework presented here in simulation.

Conclusion

In this work, we presented a distributed location verification protocol that uses proof of location to verify satellite positions and blockchain to provide a trustless source for reported locations. The protocol is retrospective and requires periodic participation. This gives a high probability that, with sufficient number of trustworthy participants, and given sufficient time, faulty behavior will be identified.

We make the point that, although the protocol could be run as a centralized scheme, the distributed nature of the blockchain allows for scalability. In particular, it can allow for trustless cooperation between satellite operators, lowering costs of deployment and operation of satellite networks in space.

References

[1] Federal Communications Commission, 2019, In the Matter of Streamlining Licensing Procedures for Small Satellites, FCC 19-81.

[2] Millan, Robyn M., et al., 2019, “Small Satellites for Space Science: A COSPAR Scientific Roadmap,” Adv. Space Res., vol. 64, no. 8, pp. 1466-1517.

[3] Union of Concerned Scientists, 2021, UCS Satellite Database.

[4] Haleem, Amir, Andrew Allen, Andrew Thompson, Marc Nijdam, Rahul Garg, 2018, Helium: A Decentralized Wireless Network.

[5] Polge, Julien, Jérémy Robert, Yves Le Traon, 2020, “Permissioned Blockchain Frameworks in the Industry: A Comparison,” ICT Express, to be published.

[6] Vukolić, Marko, 2016, Quorum Systems: With Applications to Storage and Consensus, Morgan & Claypool.

[7] Nakamoto, Satoshi, 2009, Bitcoin: A Peer-to-Peer Electronic Cash System.

[8] Miller, Andrew, Arvind Narayanan, 2014, “Alternative Mining Puzzles,” Lecture in Advanced Topics in Computer Science: Bitcoin and Cryptocurrency Technologies, Princeton University, New Jersey.

[9] King, Sunny, 2013, Primecoin: Cryptocurrency with Prime Number Proof-of-Work.

[10] Protocol Networks, 2017, Filecoin: A Decentralized Storage Network.

[11] Li, Xin, Bei Hua, Yi Shang, Yan Guo, LiHua Yue, 2007, “Bilateration: An Attack-resistant Localization Algorithm of Wireless Sensor Network,” Proc. Int. Conf. Embedded and Ubiquitous Comput., Taipei, pp. 321-332.

[12] Cota-Ruiz, Juan, Jose-Gerardo Rosiles, Ernesto Sifuentes, Pablo Rivas-Perea, 2012, “A low complexity geometric bilateration method for localization in wireless sensor networks and its comparison with least-squares methods,” Sensors, vol. 12, no. 1, pp. 839-862.

[13] Rappaport, Theodore S., 2002, Wireless Communications: Principles and Practice, 2nd ed., Prentice-Hall, Upper Saddle River, New Jersey.

[14] Groves, Paul D., 2008, Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems, Artech House, Boston.

[15] Abliz, Mehmud, Taieb Znati, 2009, “A Guided Tour Puzzle for Denial of Service Prevention,” Proc. Annu. Comput. Security Applicat. Conf., Honolulu, pp. 279-288.

[16] Ali, Isra Mohamed, Maurantonio Caprolu, Roberto Di Pietro, 2020, “Foundations, Properties, and Security Applications of Puzzles: A Survey,” ACM Comput. Surv., vol. 53, no. 4:72.

[17] Torky, Mohamed, Tarek Gaber, Aboul Ella Hassanien, 2020, Blockchain in Space Industry: Challenges and Solutions, arXiv:2002.12878.

[18] Jones, Karen L., 2020, Game Changer: Blockchains in the Space Sector, Center for Space Policy and Strategy, Aerospace Corp.

[19] Hyland-Wood, David, Peter Robinson, Roberto Saltini, Sandra Johnson, Christopher Hare, “Methods for Securing Spacecraft Tasking and Control via an Enterprise Ethereum Blockchain,” Proc. Int. Communications Satellite Systems Conf., Okinawa, pp. 669-684.

[20] Hyland-Wood, David, et al., 2020, “Blockchain Properties for Near-Planetary, Interplanetary, and Metaplanetary Space Domains,” J. Aerosp. Info. Syst., vol. 17, no. 10, pp. 554-561.

[21] Molesky, Mason, et al., 2018, “Blockchain Network for Space Object Location Gathering,” Proc. IEEE Annu. Info. Tech. Elec. Mobile Comm. Conf., Vancouver, pp. 1226-1232.

[22] Xu, Ronghua, Yu Chen, Erik Blasch, Genshe Chen, 2019, “Exploration of Blockchain-Enabled Decentralized Capability-Based Access Control Strategy for Space Situation Awareness,” Opt. Eng., vol. 58, no. 4:041609.

[23] Cheng, Shaochi, et al., 2018, “Blockchain Application in Space Information Network Security,” Proc. Int. Conf. Space Info. Netw., Changchun, China, pp. 3-9.

[24] de La Beaujardiere, Jack, Mital, Rohan, Mital, Rohit, 2019, “Blockchain Application within a Multi-Sensor Satellite Architecture,” Proc. IEEE Int. Geosci. and Remote Sensing Symp., Yokohama, pp. 5293-5296.

[25] Mandl, Dan, 2017, Bitcoin, Blockchains and Efficient Distributed Spacecraft Mission Control, Info. Sci. and Tech. Colloq., Goddard Space Flight Center, Code 581.

[26] SpaceChain, 2020, SpaceChain: Community-Based Space Platform.

[27] Kalabić, Uroš, Avishai Weiss, Michael Chiu, 2020, “Orbit Verification of Small Sat Constellations,” Proc. Int. Conf. Blockchain and Cryptocurrency, virtual.

[28] de Ruiter, Anton H. J., Christopher J. Damaren, and James R. Forbes, 2013, Spacecraft Dynamics and Control: An Introduction, Wiley, Chichester, United Kingdom.

[29] Lee, Taeyoung, Melvin Leok, N. Harris McClamroch, 2007, “Lie Group Variational Integrators for the Full Body Problem in Orbital Mechanics,” Celest. Mech. Dyn. Astron., vol. 98, pp. 121-144.

[30] Katz, Jonathan, Yehuda Lindell, 2015, Introduction to Modern Cryptography, 2nd Ed., Boca Raton, FL, CRC Press.

[31] Baignères, Thomas, et al., 2015, Trap Me if You Can: Million Dollar Curve, Cryptology ePrint Archive, Report 2015/1249.

[32] Castro, Miguel, Barbara Liskov, 1999, “Practical Byzantine Fault Tolerance,” Proc. Symp. Operating Syst. Des. and Implementation, New Orleans, pp. 173-186.

[33] Androulaki, Elli, et al., 2018, “Hyperledger Fabric: A Distributed Operating System for Permissioned Blockchains,” Proc. EuroSys Conf., Porto, Portugal, no. 30.

[34] Bonneau, Joseph, Izaak Meckler, Vanishree Rao, Evan Shapiro, 2020, Mina: Decentralized Cryptocurrency at Scale.

Acknowledgements

This work was supported by Mitsubishi Electric Research Laboratories (MERL). The authors acknowledge Dr. Kieran Parsons of MERL for technical discussions.

2021 Integrated Communications Navigation and Surveillance (ICNS) Conference
April 20-23, 2021