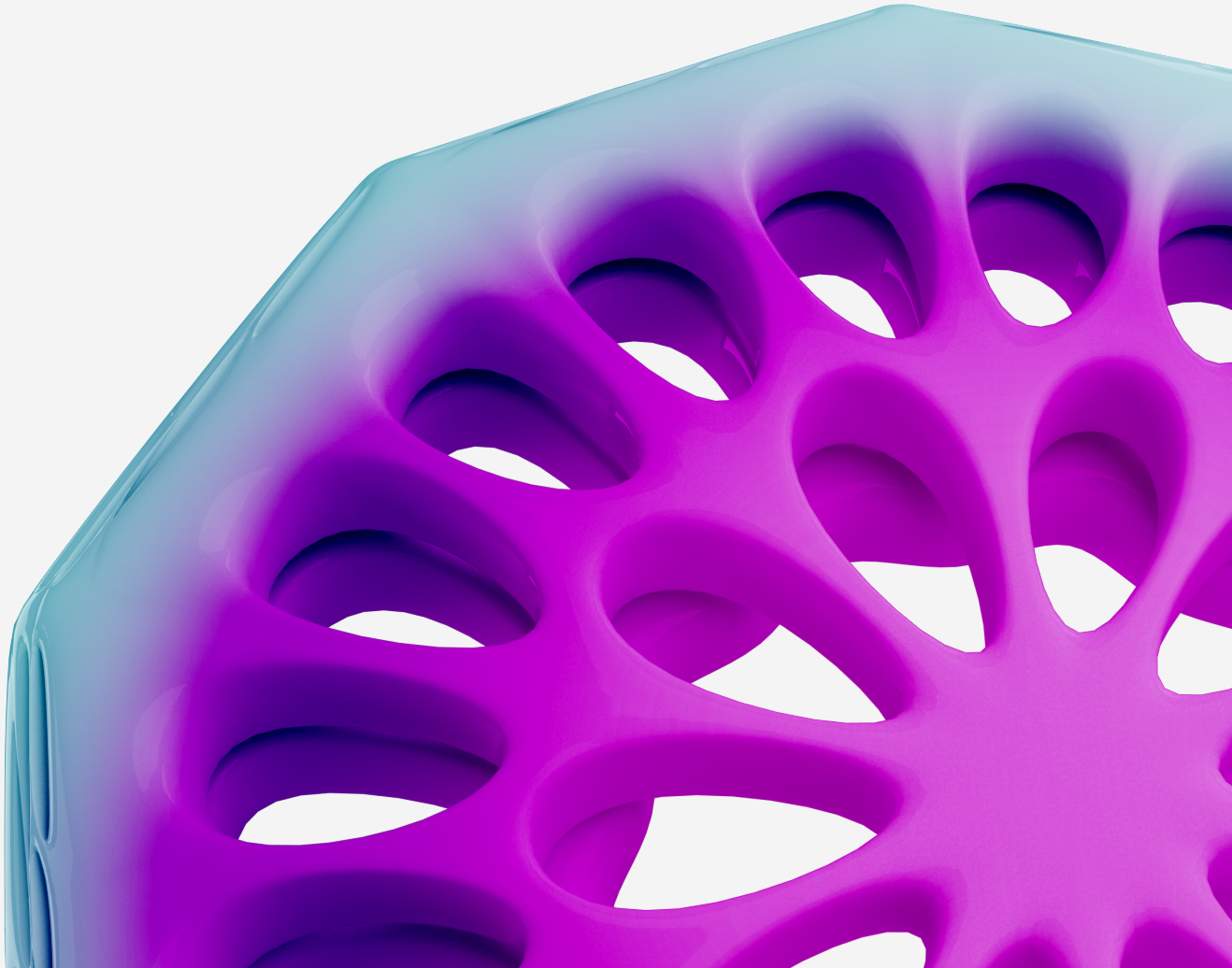
CRYPTOSAT
TRUST IN SPACE

March 2023

SpaceVDF

Verifiable Delay Functions using Cryptographic Satellites

Yonatan Winetraub, Elad Sagi, Yan Michalevsky (Cryptosat)
Chhi'mèd Künzang, Jonathan Gross (Protocol Labs)



Introduction

Native blockchains don't have an inherent way to measure passage of time, nor provide a guarantee that some time has passed between two events. Some protocols and smart contracts require incorporating time delays as part of their execution. For example, if two parties agree to exchange goods over future payment they may commit a smart contract in which the funds are deposited at the beginning of the contract but are released after a certain amount of time. As financial systems depend more and more on these types of contracts, the incentive to "cheat" increases. For this reason, blockchain users require a more robust guarantee for the passage of time.

Variable Delay Functions (VDF) or time lock puzzles are used to guarantee passage of time computationally. A VDF allows calculation of a pseudorandom output value which depends on its input and cannot be calculated faster than some minimum time. However, because this function allows an efficiently verifiable proof, it requires much less time to verify that the calculation was performed correctly. When a VDF is used to guarantee the passage of time, we must assume an attacker can evaluate the function on the fastest theoretically possible hardware.

On Earth, it is very difficult to guarantee a specific VDF must conclude within x amount of time (but not faster). A determined attacker with might build a faster computer (or ASIC) and attempt to evaluate the VDF faster, to gain financial incentives. Therefore, there is a need to build a VDF system that has some physical limitations on how quickly it can compute. Since physics cannot be broken, there is a guarantee that an attacker will not be able to evaluate the VDF faster.

In this document we aim to evaluate how VDF algorithms based on physical limits can be implemented in satellites and which physical properties / or roles of physics we can utilize to guarantee the passage of time. The goal of this study is to perform principal system analysis, identify main issues and risks, propose a path for derisking and come up with a budget and timeline for a suitable satellite (or satellite constellation).

Speed of light

Using verifiable communication distance as a bound to the passage of time.

Orbital mechanics

Using the finite speed of a satellite revolving around the Earth as a way to verify the passage of time.

Thermodynamics

Using the second law of thermodynamics to provide a bound on how quickly a computation can happen based on the power available to a small satellite and the heat dissipation availability of the satellite.

Key Terms

This document explores limitations and opportunities of several physical mechanisms to guarantee passage of time for time-lock puzzles.

We explore the use of sequential computation along with off the shelf satellite components and dedicated hardware. Key parameters of the system in this study are:

VDF “Tick” delay

The minimal basic time that the system guarantee passed. Goal: 1 sec.

A_max (Attacker’s Maximum Advantage)

Bounding the possible advance an attacker could have due to randomness or system vulnerability. Goal: A_max <1.001.

Availability

What is the solution scale to get to 100% availability of the service?

Ease of implementation and cost

Are there specialized technologies required to be developed? What are the launch opportunities and costs?

In this document we explore the relationship between VDF Tick delay (μ), A_max, and the uncertainties in the system Δ.

$$\mu = \frac{\Delta}{A_{max} - 1}$$

Using commercial off the shelf satellite components, we estimate $\Delta \approx 1sec$. This doesn’t allow us to satisfy A_max and VDF Tick delay (μ) together.

- $A\_max = 1.001$ *yields* $\mu = 1000sec$
- $\mu = 1sec$ *yields* $A\_max = 2$

This document will explore different implementations of a VDF as well as options to improve Δ

Time of Flight / Speed of Light Design

In this design, the VDF is based on the time passing between two sequentially signing parties. The basic building block of that transaction is described by the diagram below:

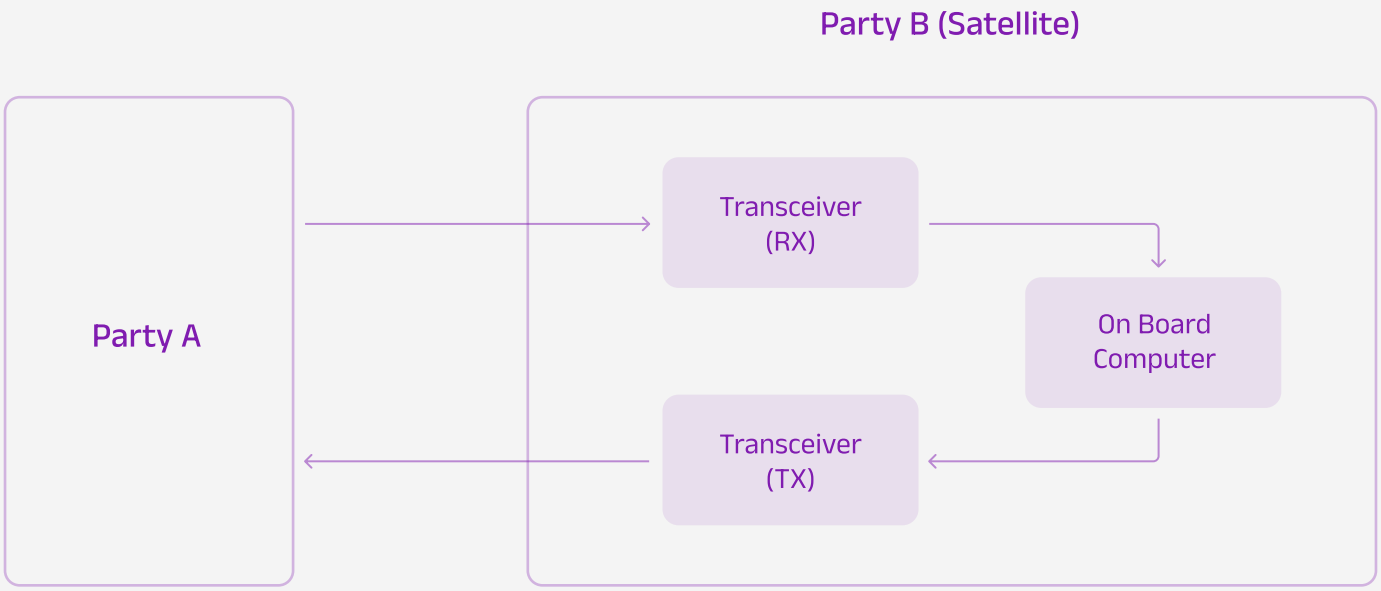


Figure 1: Satellite built-in delays

Points on the Graph	Delay Description	Expected Delay Value
1→2 $(D_{1\to2})$	**Light propagation through space.** Measured from the time radio signal leaves party A's antenna until radio signal reaches party B's antenna. $D_{1\to2}$ can be broken to two components: intended delay μ, and an error component σ due to uncertainties in satellite position. $D_{1\to2} = \mu + \sigma$	μ is the main delay used to ensure the VDF and will be discussed at length later. σ is due to the error in satellite position estimation. A standard TLE component published by NORAD has an average position error of ~15km which corresponds to $50\mu sec$.
2→3 $(D_{2\to3})$	**Transceiver RX delay.** Delay between when radio signals "touch" the antenna until the bit stream comes out of the transceiver. This delay is highly impacted by the transceiver implementation, the radio signals encoding method, and any on board buffers.	We measured both transceiver TX and RX delay using a common transceiver by placing it in loopback mode and measuring the time it took from the moment a signal was sent until it was received. The measurement was done at 9600bps uplink and downlink. Transmission of ~500bits each way requires ~120ms of the loop-back measurement time (with overhead).
6→7 $(D_{6\to7})$	**Transceiver TX delay.** Opposite of 2→3. Delay between when bytes are received in transceiver until radio signals are transmitted	
3→4 5→6 $D_{3\to4}$ $D_{5\to6}$	**Satellite Bus.** Different components at the satellite communicate using I2C bus or UART communication with limited bit rates	I2C standard speed is 40kbit/sec. UART can support similar speeds. Assuming a signature message is ~500bits, this delay is at the order of 25msec total.
4→5 $(D_{4\to5})$	**Signature Calculation Delay.** Once transmission is received, the OBC will sign the data package as a high priority process and send it back to transmission.	Using a standard CPU named iOBC, we signed a message using standard PEM-encoded ECDSA key and measured signature time to be ~600ms.

For the purpose of this study, we can assume that an attacker could take advantage of delays between 2→5 or σ. Examples include: attacker speeding up calculation or using natural variability in calculation times to gain an advantage over an honest party. The overall delay that can be used by an attacker is then

$$\Delta = \sigma + D_{2\to3} + D_{3\to4} + D_{4\to5} + D_{5\to6} + D_{6\to7} \approx 1sec$$

We can therefore tie the A_max requirement with the delays of the system:

$$A_{max} = \frac{\mu+\Delta}{\mu} \leq 1.001$$

$$\mu \geq 1000 \cdot \Delta$$

Using delay values from the table, we can see that a single light propagation delay step should exceed ~1000 seconds, or ~800 times the distance to the moon traveling at the speed of light.

Relaxing μ Requirement (What Can be Done?)

In the table above we can see the main contributors to Δ (uncertainty in the delay) are signature computation time and transceiver TX,RX delays. In order to further reduce tick time μ closer to the requirement range we will need to also resolve more minor contributors to Δ such as bus delays. Below are several proposals for how to do so. It is worth mentioning that μ requirement cannot be made "easier" if we allow for multiple transmission event between satellites since each transmission event adds Δ uncertainty in the delay for each transmission.

We can, however consider the following improvements:

Switch to a more capable on board computer ($D_{4\to5}$)

IOBC is a 400MHz ARM9 processor which is space hardened for prolonged operation in space. Most modern space computers have similar performances. We can however utilize Raspberry Pi zero (1GHz) architecture for faster computation bringing delay down to $D_{4\to5} \approx 5msec$ (see benchmark for 4GHz CPU). Note that the modified computer will not be radiation hardened, and therefore unlikely to operate at altitudes higher than Low Earth Orbit. Meaning GEO or Lunar orbits are unlikely to be feasible using this architecture.

Redesign the Transceiver ($D_{2\to3}$, $D_{6\to7}$)

This process is expected to be expensive. We may need to have a much faster bit rate.

Switch to faster bus ($D_{3\to4}$, $D_{5\to6}$)

If both on board computer and transceiver allows, we can utilize I2C fast mode which will yield speeds up to 400,000 bits per second gaining $D_{3\to4} = D_{5\to6} = 1.5ms$.
For faster speeds we will need to utilize space wire or SATCAN which can operate at faster speeds without being vulnerable to radiation bit flips. These protocols however are not used by Raspberry Pi and other modern processors natively.

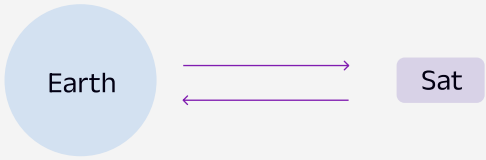
Implementing the above improvements could (optimistically) provide us with $\sigma \approx 10msec$ yielding:

$\mu \geq 10sec$, **about ten times the distance to the moon**

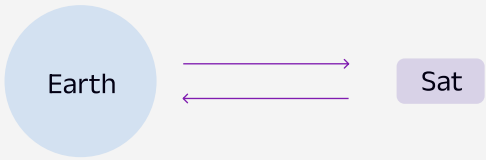
In an more aggressive scenario $\sigma \approx 1msec$, which provides $\Delta \geq 1sec$, about the distance to the moon.

Possible Orbital Architectures

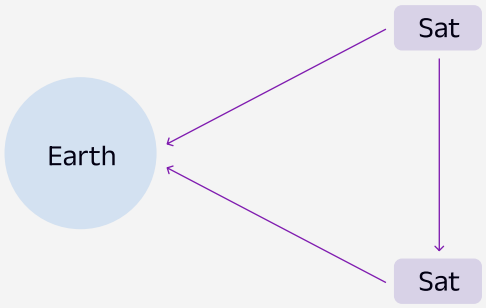
Option A: 1 Satellite at Lunar Orbit / Close to Lunar Orbit

Description & Orbit	One satellite placed in L4 or L5 or lunar orbit, receives messages from Earth, signs them and returns the message back	
Basic Tick Delay Unit	About 1-2 seconds for a round trip transmission	
A_max	Better than 1.001 at simple implementation	
Availability	12 hours a day using one ground antenna, 24 hours with a world wide ground station network	
Cost	Satellite complexity is similar to Chandrian-2, or Beresheet-2 orbiter at cost of ~$70M. Probably much cheaper if using a shared satellite with a commercial lunar orbiter. In order to communicate with satellite we will probably need to rent bandwidth from NASA's deep space network.	

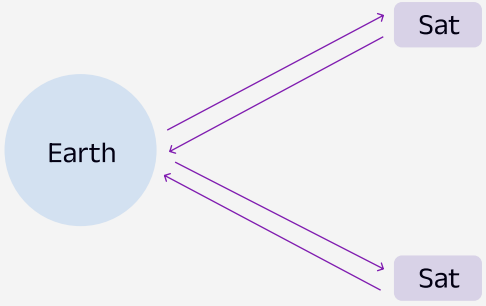
Option B1: 1 Satellite at Geostationary Orbit (*)

Description & Orbit	One satellite placed in GEO orbit, receives messages from Earth, signs and returns the message back	
Basic Tick Delay Unit	About 200msec for a round trip transmission	
A_max	1.05 to 1.005 depending on implementation	
Availability	24 hours a day using one ground station	
Cost	Main cost factor is technology development required for a faster computer and low latency transceiver. Component development is high (TBD) due to radiation hardening limitations.	

Option B2: 2 Satellites at Geostationary Orbit, Direct Communication

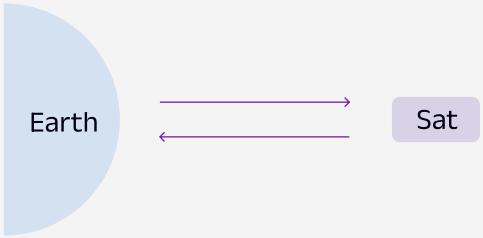
Description & Orbit	Two satellites are placed in GEO, distance of TBD km from each other	
Basic Tick Delay Unit	About 200msec to 500msec for a round trip transmission, depending on satellite to satellite distance	
A_max	1.02 to 1.002 depending on implementation	
Availability	24 hours a day using two ground stations one for transmit and one for receive	
Cost	Main cost factor is technology development required for a faster computer and low latency transceiver. We will also need to develop a satellite to satellite communication terminal which is not as common for GEO satellites. Component development is TBD, but should be high due to radiation hardening limitations. This option is probably more expensive when compared with Options B1 and B3.	

Option B3: 2 Satellites at Geostationary Orbit, Communication Through a Ground Station

Description & Orbit	Very similar to option B2 but no need for a direct satellite to satellite communication. In this case the ground station provides a simple communication relay that doesn't impact security as messages are signed directly by satellites.	
Basic Tick Delay Unit	About 200msec for a round trip transmission	
A_max	1.02 to 1.002 depending on implementation	
Availability	24 hours a day using two ground stations	
Cost	NRE cost very similar to B1, launch of two satellites.	

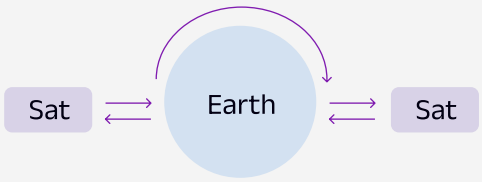
Option C1: 1 Satellite at LEO

Description & Orbit	One satellite placed in LEO orbit, receives messages from Earth, signs and returns the message back. This is the simplest solution to implement
Basic Tick Delay Unit	About 2-15msec for a round trip transmission
A_max	100 to 2 depending on implementation
Availability	About one hour a day using the ground station. 24 hours a day using a ground station network and multiple independent satellites.
Cost	This is the simplest solution.



Option C2: 2 Satellites at LEO

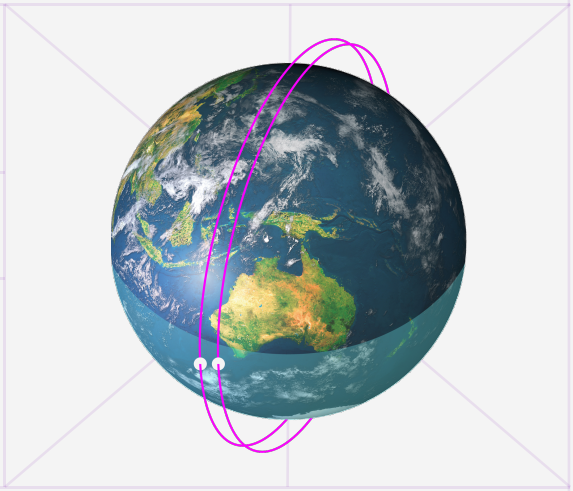
Description & Orbit	Satellites orbit the Earth on opposite sides, one satellite is over the US while the other is over Australia. Satellites don't have a direct line of sight from each other, but can each communicate with a ground station. Ground stations extend information via the internet. Most of the delay is due to the signal traveling across the world via an internet connection (half of Earth's circumference). The VDF guarantee comes from the satellite's known location over the ground station.
Basic Tick Delay Unit	60-70 msec
A_max	20 to 1.1 depending on implementation
Availability	About 20 minutes per day using one pair of satellites. 24 hours a day using multiple satellites
Cost	Main cost factor is technology development required for a faster computer and low latency transceiver. However this is a very simple solution similar to C1



Orbital Mechanics Delay

Option D: Satellite meet at the poles

Description & Orbit	MVP: Two satellites orbit the Earth on a sun synchronous orbit (SSO) at different longitudes of ascending nodes (difference of degrees) see orbits on Figure 2. The two satellites are in phase such that they will meet at the two poles. While at the poles, the satellites can exchange signatures using short range communication. Each exchange corresponds to half orbit time or about 45 minutes
Basic Tick Delay Unit	10 minutes to 45 minutes depending on the location of the ground station used for upload
A_max	<1.001
Availability	About 1 hour per day for one pair of satellites. 24 hours a day using multiple satellites (at order of 20 satellites) and ground stations
Cost	A relatively simple solution. It would require development and validation of the close range communication system and a protocol that would ensure that only one step of the protocol is performed as part of each rendezvous of the two satellites



Sequential computing scheme options

Option #1 (~10 minutes delay)

- ✓ VDF input is uploaded to satellite A by a ground station in North America using long range communication (VHF/UHF)
- ✓ Satellite A signs the input
- ✓ At the pole, satellite A transmits it to satellite B using short range communication
- ✓ Satellite B signes the input
- ✓ Satellite B transmits VDF to a ground station using long range communication (VHF/UHF)

The time delay is determined at the time of upload.

Option #2 (45 minutes delay)

- ✓ VDF input is uploaded to satellite A by a ground station using long range communication (VHF/UHF)
- ✓ Satellite A signs the input
- ✓ At the pole, satellite A transmits it to satellite B using short range communication
- ✓ Satellite B signs the input
- ✓ At the next pole, satellite B transmits the input back to satellite A
- ✓ Satellite A signs the input again
- ✓ Satellite A transmits VDF to a ground station using long range communication (VHF/UHF)

The time delay is at least 45 minutes guaranteed using orbital mechanics.

In both cases A_max <1.001 as communication overhead is significantly smaller compared to the delay.

Guarantee of Short Range Communication

UHF/VHF long range communication system supports ranges of 600km to 5,000km. At the pole, we expect satellites to use a short range communication system suitable for ranges of up to 50km.

We can suppress an attacker from listening to satellite A's output and transmit it to satellite B before arriving at the pole thus getting a VDF advantage using the following methods:

Reduced power

Transmission to 50km requires 100 times less power than transmission to Earth, therefore by reducing the transceiver's power we can make it more difficult for an Earth attacker to hear the signals

Note that since NORAD monitors orbiting objects, we could see if a space based attacker came close to one of the satellites away from the poles and issue a warning that VDF for that particular time point is invalid. It is virtually impossible to "tailgate" a satellite without NORAD triggering a warning.

Also there are alternatives to NORAD:

- European version: EUSST.eu
- Russian version: The 821st Main Center for Reconnaissance of Situation in Space
- Chinese version

Non governmental alternatives:

- Leolabs, exoanalytics

Reduce the signal received on Earth

There are two techniques to prevent signals in space reaching the ground (each depends on the wavelength of the communication:

1. Atmospheric absorption (IR communication)

This technique is more relevant to optical infra-red communication. We would like to transmit signals at the IR absorption lines of the atmosphere to prevent any signals from reaching the ground. One drawback of IR communication is it requires directional signaling

2. Ionosphere bouncing (RF communication)

The Ionosphere has been used for many years to help amature radio enthusiasts to communicate over long distances by reflecting the radio waves off the free ions in the ionosphere. A satellite traveling above the ionosphere is expected to experience a similar effect, however instead of bouncing towards the Earth, the signals will be reflected back to space. This will significantly reduce the probability of an Earth based attacker receiving the signal

Key Challenges & Cost

Cost of such a system is relatively low as it utilizes mostly off the shelf components with minor developments. Key challenges are:

Synchronizing

Two polar orbiting satellites such that they meet twice per orbit. This can be achieved by existing differential drag technology which is a simple proven algorithm but requires implementation in our current system

Short range communication

Solutions exist on the market, we will need to tailor a specific solution and demonstrate signal attenuation over long distances either by showing atmospheric absorption or ionosphere bouncing

Thermodynamics Limit of Computation Speed

A 1U CubeSat at a size of 10 by 10 by 10cm has a physical limit on the amount of solar energy it can capture ($EMax_{in} = 3.5W$, see calculation below) as well as the amount of heat it can dissipate to space via radiation ($Max_{out} = 32W$, see calculation below). Notably, this energy bound depends on the satellite's area, and therefore we need a way to verify the size of the satellite once in orbit (link) or trust that the satellite is indeed 1U CubeSat.

On the other side of the equation, we use power to compute the VDF. There is a thermodynamic bound to the power consumption of the calculation (Lanauer's principle link). Notably, computers today are very far from that limit, however processor power efficiency (performance per watt) doubles every 2.6 years (Koomey's Law, link, see some benchmarks) so advancements are being made in that field all the time.

We could hypothesize a scheme where a space based VDF is a simple ASIC calculating a simple non reversible cryptographic puzzle. The puzzle doesn't need to be a sequential problem and can be easily parallelizable like verifying many elliptic curve signatures. The idea is we can find a simple implementation of VDF that can be easily implemented on state of the art hardware. However, due to the available input power and heat dissipation constraints and the date of the launch, there is an upper limit on how fast the calculation can run on the hardware that was available at the time of launch.

This solution's Amax depends on the exact implementation of ASIC technology and how well we can bound the solar input of the satellite. We estimate A_max to be around 2.

Computing Maximum Energy Input and Output

In order to compute the max power input on a cubesat we use

$$EMax_{in} = f \cdot \eta \cdot A \cdot O = 3.5W$$

Where

- $f = 1000[W/m_2]$ is the solar flux around the Earth.
- η is the solar panel efficiency. The best commercial products offer η=40%.
- Where A is the maximum surface area the Cubesat can turn to the sun.

$$A = 1.7 \times 10cm \times 10cm = 0.017\,[m^2]$$

- O is the percent of the orbit in which the satellite is in darkness. Here we assume the satellite provides continued service and in SSO orbit we can assume 50% of the time the satellite is at an eclipse and therefore during the day the available power is 50% lower as batteries require charging.

In order to compute the max power that can be dissipated to space by a 1U cubesat we use the heat radiation equation:

$$EMax_{out} = \varepsilon \cdot \sigma \cdot A \cdot T^4 = 32W$$

Where

- ε = 95% is the satellite's emissivity, its ability to radiate heat
- σ is constant $5.67 \times 10^{-8}\,[W/m^2K^4]$
- A is the surface area of the satellite (6 faces of 10x10cm each) $A = 0.06\,[m^2]$
- T is the temperature of the satellite, since our satellite hosts a battery, temperature cannot be constantly above 42C/315K without losing significant performances

Measuring Satellite's Surface Area

In order to violate the input or output energy consumption constraints, the cubesat could deploy additional solar panels and radiators significantly increasing the available power.

Here we discuss a few methods to verify a satellite's surface area post launch. These methods can be done by multiple independent observers that can attest that the satellite's surface area is as identified in the specification and thus providing a hard limit on the power provided.

Direct imaging

A 10cm satellite at a typical distance of 500km requires 0.2 microradians or 0.04 arcsec to be directly imaged. This is a significant requirement, comparable with Hubble space telescope's performances.

Radar & Optical Cross Section

By shining a beam of light or RADAR and measuring the reflected power, we can estimate the crosssection of the satellite. However this method could be skewed if the satellite has a "stealth" design aimed towards reflecting the beams away thus appearing less bright when measured.

Dimming of Background Stars

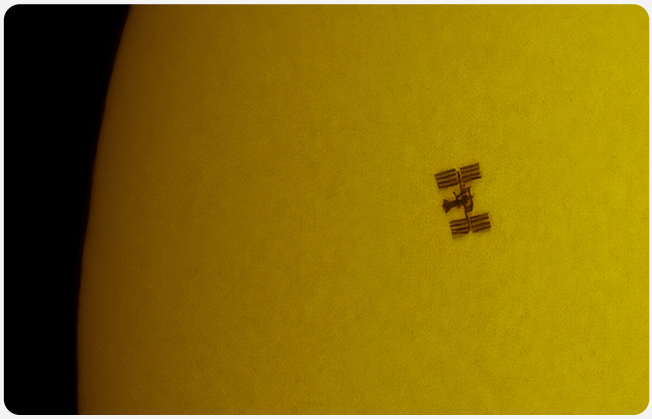
When the satellite crosses in front of a background star, the light of the star will dim according to the cross section area of the satellite. This method, called transient photometry, is used to detect planets in other solar systems.

The method is as follows:

- The satellite measures its location using an on board GPS and publishes that information
- A ground telescope (diameter of 30 cm) predicts when the satellite will pass in front of a random star in the sky
- The ground telescope measures the start intensity right before and during the pass, by comparing the two measurements along with the telescope's diameter we can estimate the satellite's cross section.

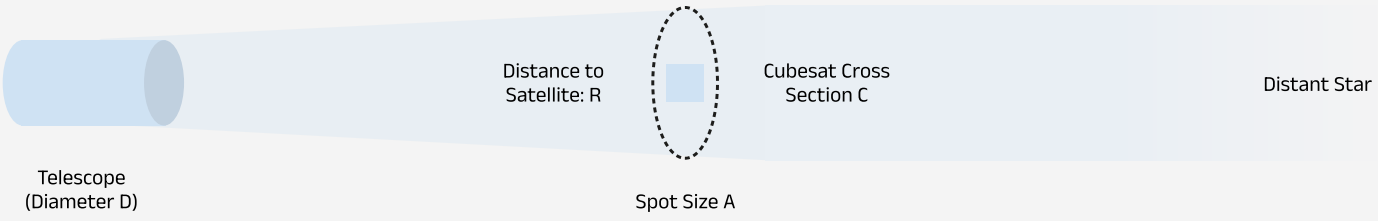
Using the diffraction limit, we can estimate the spot size of a telescope as a function of the distance to the satellite. We estimate the dimming effect C/A to be at the order of 10%-50% for diffraction limited small telescopes (after correcting for atmospheric aberrations).

Another issue will be the imaging speed. The satellite will cross over the satellite by an order of 100 microseconds (while traveling at 7km/sec). To achieve this precise timing we require the cubesat to transmit its location, and likely require using a PMD sensor for precise timing and low intensity measurement.



Example: International Space Station transient in front of the sun. Cubesat transient image won't be as sharp, but will show momentary dimming on the night stars background

Formulation

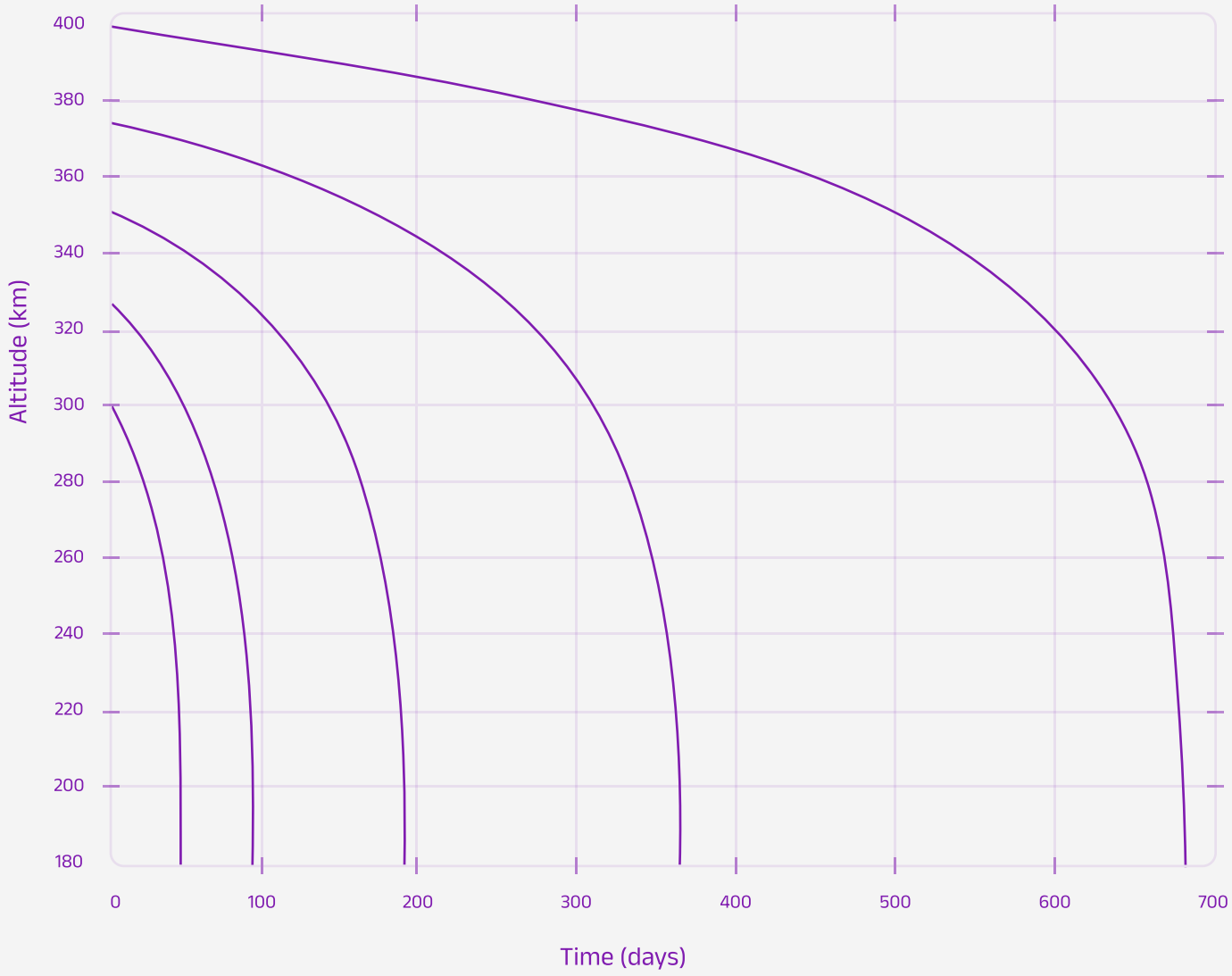


Orbital Decay

Surprisingly, there is no distinct Earth atmosphere / space boundary. The atmosphere becomes less and less dense at higher altitudes. In low Earth orbit, residual atmospheric particles collide satellites to slowly lower their orbit until they fall back to Earth (orbital decay). Although orbital decay may take years until it brings down a spacecraft (depending on altitude, see image below), it is measurable by NORAD and other ground tracking services.

Orbital decay speed is strongly associated with the surface area of the satellite (A), the larger the surface area, the more drag the spacecraft encounters which yields a faster decay. In order to validate a satellite's surface area, we can measure orbital decay rate and compare it to an atmospheric model. Alternatively, we can launch a second reference cubesat constructed by an independent party but estimated to have the same surface area and compare the orbital decay of both satellites, if one is significantly faster than the other, it can serve as indication for a larger surface area.

Vectorized Orbital Decay vs. Time $A^*C_D = 1$



Orbital Decay Example

Cryptographic instantiation

At large, the VDF or proof of sequential communication delay boils down to an ordered sequence of digital signatures computed by two or more parties to prove that a given number of communication rounds (with some minimum delay) took place. The signatures can later be verified by a verifier who has a verification key (or a set of verification keys).

We start by describing a simple cryptographic solution with two satellites. Each one of the satellites publishes a set of public keys corresponding to an EDDSA signature scheme. The size of the set depends on the maximum number of rounds we want to be able to attest to (and that sets the maximum delay we can prove). The computation is initiated by a user that submits a nonce to one of the satellites. The satellite signs it, and submits the signature to the other satellite which signs this signature in turn. After a sufficient number of rounds, the entire transcript is returned to the user who can verify the transcript using the set of public keys. Note that, while the transcript keeps growing, it does not need to be uploaded in its entirety to the satellite each time, as it is enough that only the hash of the transcript (computed after every update to it) is uploaded.

Below, we outline the algorithm more formally:

Procedure	Output
Setup(n) – where n is the number of rounds	
Satellite generates n key-pairs: $(sk_1,pk_1),(sk_2,pk_2),..,(sk_n,pk_n)$	Output $pk_1,pk_2,..,pk_n$ and broadcast to any groundstations that will participate in VDF calculations
$Round_1\ (challenge,\ sk_1)$ – where challenge is a random provided by the verifier	
Ground station submits the challenge to the satellite. The satellite signs the challenge: $\sigma_1 = sign_{sk_1}(challenge)$	Output σ_1 and send it to groundstation
$Round_{i+1}(\sigma_i,sk_{i+1})$	
Ground station submits the challenge to the satellite. The satellite signs the challenge: $\sigma_{i+1}=sign_{sk_{i+1}}(\sigma_i)$	Output σ_{i+1} and send it to groundstation
$Verify\ (challenge,\ \sigma_{i=\{1..n\}})$	
The verifier runs $Verify_{pk_1}(challenge,\ \sigma_1)$ $for\ i=2\ to\ n:\ Verify_{pk_i}(\sigma_i,\sigma_{i+1})$	*Accept* if all signature verifications passed. Otherwise *Reject*.

Instantiation using Ordered Multi-Signatures

While this scheme can be practical, its disadvantage is a linearly growing transcript, and linear verification time. To this end, we examined the option of using the recently developed CaSCaDE scheme by Baum and David [1]. The CaSCaDE construction relies on several assumptions, including standard cryptographic primitives such as hash functions and the existence of public-key infrastructure (PKI), as well as specific requirements such as the existence of a public bulletin board or ledger. Some features also require synchronized clocks, albeit this is optional and is only required for additional checks performed while executing the protocol.
At the heart of the construction is an Ordered Multi-Signature scheme (OMS). It enables to verify that a group of signers has signed a sequence of messages in a certain order, and the size of the aggregate signature is independent of the number of rounds.

We refer to the full CaSCaDE paper for a detailed description of the Ordered Multi-Signature scheme, and focus here on the key aspects that are relevant to the VDF implementation.

Performance and cryptographic operations and timing estimates

For a practical VDF deployment, it is important to validate that computation times are significantly smaller than signal travel time.

We instantiate the scheme using the Ed25519 curve as the group, which attains 128-bit security. One group exponentiation takes about 430 μsec. We ignore some additional arithmetic and hashing as this is expected to be the dominant operation. We provide estimates for Cortex A9 at 1200 MHz.
Setting the parameter l (output by the Setup alg. In CaSCaDE) to 2, key generation takes 430 μsec. The offline phase of the signing algorithm takes at most 1ms on a single core, and can be parallelized using multiple cores (Cortex A9 has 4 cores). Step 9 of the online phase is expected to be the computationally “expensive” part, taking 1ms for the left-hand side and 500μsec x n for the right-hand side. For n<10, we estimate the upper-bound for the online phase by 4ms.
The verification phase is one for which timing is less crucial, however it is lightweight enough to fit in about 600μsec. Using other curves such as NISTp256 will result in slower computations.

Given the numbers above, we conclude that a VDF implementation that uses the CaSCaDE scheme has to rely on two satellites placed at opposite points on an orbit to guarantee a minimal travel time sufficient to ensure that computation is negligible in comparison. This could result in VDF instantiations that are on the order of 10s or 100s of milliseconds, or more. Instatiations with orbital mechanics where two satellites can exchange messages even less frequently are possible as well, but result of course in significantly longer VDF computation timeframes.

On the other hand, an implementation that relies on simple EDDSA signatures can be deployed between a ground-station and single satellite, with the computation time being relatively small compared to the 40ms communication delay. As mentioned earlier, this comes at the cost of a growing transcript, which can be maintained on the ground.

References

1. CaSCaDE: Cryptography from Space Communications DElay.
Carsten Baum and Bernardo David, 2022.

2. Pereida García, Cesar, and Sampo Sovio. "Size, Speed, and Security: An Ed25519 Case Study."
Secure IT Systems: 26th Nordic Conference, NordSec 2021, Virtual Event, November 29–30, 2021,
Proceedings 26. Springer International Publishing, 2021.