A Step Toward Working with Untrusted Ground Stations

Anonymized Author(s)

ABSTRACT

We are witnessing a shift toward outsourcing satellite and ground station services to third-party commercial entities. As with any enterprise, these third parties can be vulnerable to cyber compromise, including image tampering and deepfake injection. The multimedia community is beginning to establish standards and technology to enable authenticity verification of multimedia created and edited by others. While appealing to the remote sensing domain, the nature of raw satellite imagery is incompatible with the proposed change verification tools, resulting in the need for a means to validate updates made to image products. We present a simple method for verifying a specific class of algorithms. Our inverse processing approach eliminates the need to see the original image as the reversed data can be checked against an original digital signature. We demonstrate our approach on basic image restoration routines and conclude with a discussion on open challenges and next steps.

CCS CONCEPTS

• **Security and privacy** → *Symmetric cryptography and hash functions*; **Authentication**.

KEYWORDS

Image Authentication, Remote Sensing, Ground Processing, Ground Station as a Service

ACM Reference Format:
Anonymized Author(s). 2022. A Step Toward Working with Untrusted Ground Stations. In *IH&MMSec '22: ACM Workshop on Information Hiding and Multimedia Security, June 27–29, 2022, Santa Barbara, CA.* ACM, New York, NY, USA, 6 pages. https://doi.org/NotAvailable/NotAvailable

1 INTRODUCTION

In the remote sensing domain, end users typically receive and use post-processed data products rather than raw sensor data. These imaging sensors produce images with imperfections that are corrected by adjusting the raw data using calibration factors. Furthermore, images are enhanced to reduce atmospheric effects and artifacts of off-nadir view angles. Much of this post processing is performed on the ground after the data is downlinked. Historically, this processing was performed in-house, but with the advent of Ground Station as a Service [1, 2, 7] and other third-party offerings, this post-processing may be performed off premise by untrusted or semi-trusted third-parties.

Trust issues regarding data product authenticity reduce the value of the data. Third-party processing systems can be vulnerable to processing errors, man-in-the-middle attacks, and system compromise. Furthermore, a third-party could harbor malicious activities such as data injection or data tampering. Recent technology advancements are raising the sophistication levels of possible attacks. Satellite imagery is susceptible to image compromising attacks and new machine learning-based deepfake geography generators [21, 25, 28] are making it harder for humans and machines to detect fake data.

A variety of methods have been proposed to enable integrity checking of post-processed data products. While cryptographic hashes become invalid when even a single bit changes, perceptive hash functions have shown utility in checking image similarity. Some have applied these methods to the remote sensing domain [12, 14, 22, 27]. The Coalition for Content Provenance and Authenticity (C2PA) [4] is standardizing approaches for securely establishing the pedigree of multimedia. The basic premise is to compute a digital signature when the image is created, and then when the multimedia is updated, the changes are added to the history and the multimedia is re-signed by the editor. Downstream users can then verify that the data was created and edited by the said parties and that no changes were made elsewhere.

These approaches have an important place, but ultimately leave the checking of the edits to the end verifier. With perceptive hash functions, there could be a mismatch between what the hash function allows and what the end user needs. With C2PA, the changes made may appear to match the provenance record, but a malicious or compromised editor could make additional changes that are hard for the end user to detect. For example, the addition of noise could cause a downstream classifier to perform poorly [16]. This problem becomes much harder when the post-processing system does not provide the original data, as is common with some satellite imagery providers today [11].

Given the necessity of post-processing, the remote sensing community needs a way to verify that actions taken by third-parties were performed correctly. We assert that the output of certain classes of algorithms can be checked for correctness by an end user that has not seen the original data. Given post-processed data, a cryptographic hash of the original data, and post-processing details, the end user can revert the data and then check it against the original cryptographic hash. Our technique is limited to invertible functions, which can be used to perform aspects of radiometric correction for image restoration [20]. This technique can also be used to undo a sequence of algorithms, in the reverse order of their original execution. Additional techniques will be required for other types of algorithms, including those that incur loss.

The rest of this paper is organized as follows. Section 2 discusses related work in greater depth. Section 3 explains our objectives and requirements. Section 4 provides an end-to-end workflow for signing, processsing, reversing, and checking an image. Finally, in Section 5 we provide our conclusions and path forward.

IH&MMSec '22, June 27–29, 2022, Santa Barbara, CA
© 2022 Association for Computing Machinery.
ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/NotAvailable/NotAvailable

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

2 RELATED WORK

Researchers have proposed many methods for ensuring or checking image authenticity, including digital signatures, digital watermarks, forensic analysis, and so forth [18]. We have sought solutions that enable verification of post-processed imagery without sacrificing security.

Digital signatures consist of a hash of the content that has been encrypted using asymmetric encryption. Holders of the public key can check the signature by recomputing the hash and then comparing it against the decrypted signature. The hash can be a cryptographic hash or a robust hash. Robust hash functions, including perceptive hash functions, are designed to allow for minor modifications, such as adjusting brightness of the image, without affecting the hash value. While the ability to do some level of post processing is a desirable property, it remains a challenge to form a hash algorithm that supports only the desired modifications. This leads to a trade-off between security and robustness [18]. As such, we have opted to depend on cryptographic hashes, which leverage the Avalanche Effect, where a small change to the input has a significant effect on the output.

Digital watermarking is another active method for image protection, with the added benefit of highlighting tampered areas in images. Once again, there are trade-offs between flexibility and security. Forensic analysis is a continual cat and mouse game between attackers and defenders [19].

Several organizations are working to establish standards and means for enabling authentication of imagery [4, 5, 9, 10] for combating disinformation. The C2PA specification does not try to establish that the provenance data is ’true’, but enables verification of its association with the asset, correct formation, and tamper-free status [3]. Furthermore, there is a focus on enabling verifiers to visually inspect the changes made to the original data, which doesn’t work well for imagery that is not easily checked by the human eye. We seek to fill, in part, the gap left by the standard to enable verification of the changes made by an editor.

A growing community is developing techniques for secure outsourced computation [26]. Some of these approaches have been applied to third-party image processing [29]. Our use case differs from the norm in that the verifier may not have access to the original data. While secure outsourced computation has challenges, such as an increased computational complexity, we believe future research could yield improvements vital for our use case.

Another approach for ensuring third-party correctness is to investigate the third-party’s infrastructure and establishment. For example, Kinser et al. proposed a methodology for establishing cyber trust scores for space enterprises [17]. While these accreditation processes are good, there can be a gap between perceived state and actual implementation.

3 APPROACH

In this section, we explain our approach to enabling cryptographic verification of some types of post-processing algorithms.

Our objective is to enable cryptographic verification of post-processed satellite imagery. The approach [4] of signing the imagery at the source and keeping a provenance record for the imagery throughout its lifecycle can also be applied to the remote sensing domain. The satellite payload would compute a cryptographic hash for a captured image and then encrypt it with an onboard private key, thus producing the digital signature. The data and this digital signature is downlinked and can be verified by any system or user with access to the public key. Post-processing systems would first check the data against the digital signature, perform processing, record actions taken, then compute a new digital signature. Receiving end users could then verify that the post-processed data was properly signed by the post-processing system. This process is depicted in Figure 1.

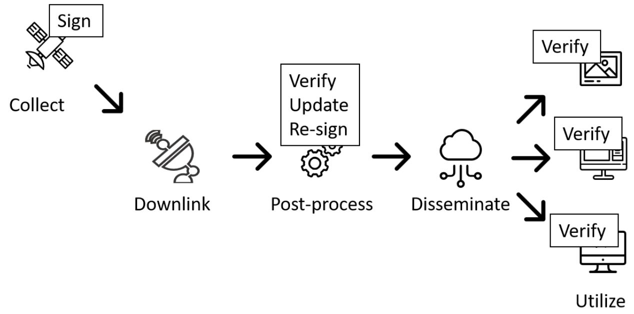

Figure 1: An example of how digital signatures and data provenance could be applied to a remote sensing workflow

Unlike regular digital photography, Electro-Optical (EO) remote sensing imagery is subject to significant imperfections due to extreme distances, sensor optics, atmospheric effects, sensor movement, and off-nadir look angles. Data processing is needed in remote sensing in order to generate consistent and reproducible measurements of the earth’s atmospheric and surface properties. Numerous textbooks are written on the subject of remote sensing data processing. Entire journal publications are dedicated to the subject and novel methods and algorithm applications are actively being developed. These algorithms have been organized by the community into a loosely-held categorization of processing levels [6, 8, 20]. For example, what NASA has identified as a Level 1A processed product may have undergone slightly different processing operation than another organization.

Variations and defects in the sensor’s readings are corrected by adjusting pixel values using calibration based correction methods [15]. Under some conditions, sensors may malfunction and require data replacement through smoothing or other estimation techniques. Sensor corrections can be performed onboard or on the ground. Atmospheric conditions, the angle of the sun, and the movement/angles of the satellite relative to earth cause distortions in the captured raw data that need to be corrected. Satellite imagery can also be processed to make the image useful to the human eye or transformed into indexes in order to measure earth/atmospheric conditions including moisture or vegetation levels. The types of processing concerned with correction and calibration of images in order to achieve an accurate representation of the earth surface are know as restoration and enhancement processing operations.[15] The intent is to remove surface, atmospheric, and instrument variations so an object’s reflectance at different periods in time across

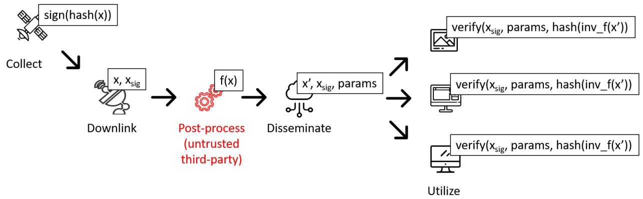
sign(hash(x))
Collect
x, x_{sig}
Downlink
f(x)
Post-process (untrusted third-party)
x', x_{sig}, params
Disseminate
verify(x_{sig}, params, hash(inv_f(x'))
verify(x_{sig}, params, hash(inv_f(x'))
verify(x_{sig}, params, hash(inv_f(x'))
Utilize

Figure 2: Overview of how a post-processed satellite data product can be reverted and checked against the original signature.

different sensors would produce the smallest variation in data values.

The transforms made in satellite imagery post-processing pipelines may not be suitable for verification by the human eye. Remote sensors may perform multi-spectral collects where some of the bands are not in the visible spectrum. Furthermore, the systems performing this processing may be untrusted by the end user because of lack of oversight or risk of compromise.

Our investigation of authenticity approaches confirms that perceptive hash functions offer too much flexibility as to the types of processing allowed in order to retain strong security guarantees [18]. A stronger verification mechanism is to have the end verifier re-run the algorithms on the original data and verify the results. For our scenario, this is infeasible as the end verifier does not receive the original data. However, some algorithms can be inverted, thus restoring the data to its original state. Furthermore, using a digital signature of the original data as a proxy for the original data, the resulting data can be checked against the signature for correctness. While only some algorithms can be inverted, we show that the primary algorithms used in basic image restoration, such as those commonly found in Level 1A processing, are invertible. An end-to-end example is offered in Figure 2. In Section 4 we describe some of these radiometric correction functions. Invertible functions are stackable in that they can be successfully inverted in the reverse order in which they were originally executed.

In order for this scheme to work, the post-processing system must provide information about the executed algorithms, their parameters, and their execution order. The responsibility of checking the correctness of the parameters falls to the verifier and additional research is required on this front. Some of these parameters, such as calibration frames, are reused for products from the same sensor. As such, the verifier need only check the parameter once. Ensuring that the same parameter is used across multiple products will cripple an adversary's ability to tamper with the data through use of custom masks. An example workflow is depicted in Figure 3.

Candidate algorithms or techniques for verification through inverse processing include:

- **Gain and offset correction** - a method of adjusting for detector bias. This is a part of radiometric correction and is sometimes called offset subtraction.
- **Filling of dead pixels** - using a dead pixel mask, produce a value for dead pixels, often by averaging the values of neighboring pixels.

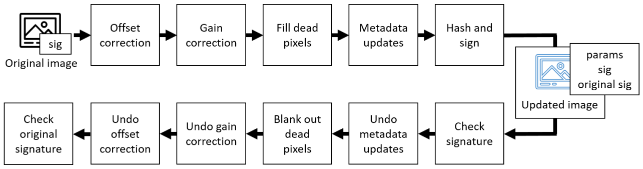

Figure 3: An example of how a sequence of algorithms used in image restoration can be reverted and checked against the original signature.

- **Append metadata** - include additional metadata (e.g. TIFF headers, EXIF, etc.) for collection or processing details, such as timestamps, equipment information, geolocation, etc.
- **Lossless compression** - reduce the size of the file without loosing information required for a full-fidelity reconstruction.
- **Scalar operations** - basic matrix addition, subtraction, division, and multiplication.

Algorithms that involve stretching, blending, or other non-reversible operations are not compatible with this technique. This limitation rules out higher level processing, such as lossy compression and orthorectification.

4 DEMONSTRATION

Here we present some simple post processing algorithms and their results for inverse processing. As a proof of concept, we start with an image (see Figure 4) which we presuppose to be a raw data which was captured from a remote satellite. The image is from Planet Labs' SkySat samples page[24]. While this is an already processed image, we will treat it as a raw data capture in order to demonstrate some common processing operations.

Step 1: Compute the digital signature (satellite payload). To simulate the onboard signing process of the satellite payload, we compute a digital signature for the original image. In our proof of concept, the image is rewritten as a 16-bit TIFF file with the TIFF headers removed.

Some formats, such as TIFF, are flexible in that they allow the same image content and metadata to be represented in different byte layouts within the file. Differences in write order or writer implementation can lead to different organizations of the data within the file. For this reason, we read-in the image and write it back out using our client library before generating the signature. This ensures that we can later bring the content back to our *original* state. The alternative is to port the content and metadata to some deterministic representation anytime before a signature is to be generated or checked. We generate the signature with OpenSSL:

```
$ openssl dgst -sha384 -sign <private key> <image> \
  > sig.sha384
```

OpenSSL hashes the file with SHA-384 then encrypts the hash with the private key. The signature is then written to a new file.

Step 2: Perform radiometric correction (third-party processor). Generally, radiometric correction for spectral radiance are applied to a raw data capture in the form of gain and offset corrections. Gain and offset algorithms effectively consist of basic matrix arithmetic [13]:

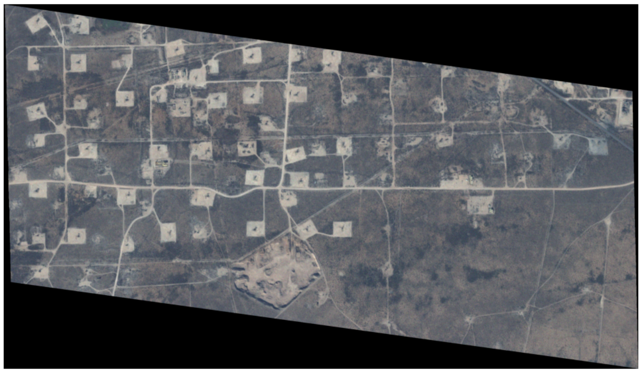

Figure 4: Presupposed raw sensor capture (RGB visual)

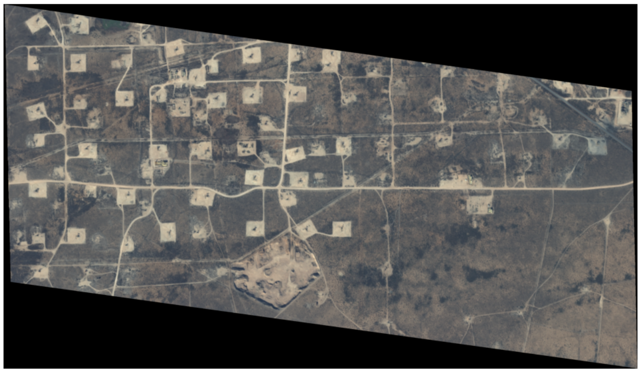

Figure 5: Corrected image (RGB scaled)

$$SR = DN \times Gain + Offset$$

SR is the resulting spectral radiance matrix, *DN* is the digital number, a matrix with the numerical values for the frame, *Gain* is the gain coefficient, and *Offset* is the offset matrix.

Radiometric calibration coefficients can be derived using estimation methods to standardize spectral radiance sensor response. Typically these calibration coefficients are considered to be stable and require infrequent updating [23]. The calibration coefficients are generated by the satellite operator and can be digitally signed for downstream verification. The coefficients can be distributed through a data stream or through a trusted public distribution service, such as a blockchain. The calibration metadata and frame files should not change dynamically for every data capture. End users are responsible for checking the calibration coefficients prior to performing image verification routines.

While correction frames and coefficients do provide an opportunity to insert/introduce malicious data into a processing pipeline, it is practically ineffective. These values could be signed once and validated many times. The frame will be used for validation across essentially all data captures from an instrument. This is due to the static nature of the calibration frame and coefficient metadata.

In our proof of concept, the offset was a frame base correction applied across pixels and spectral bands and was simulated using a random sample from a power distribution. The gain constant simulates a conversion of a DN to a spectral radiance. The offset represents a correction across focal plane sensors to account for variation in the sensor data read.

As an example, a band from the image file would be corrected for gain and offset as follows:

$$\begin{bmatrix} 1 & 2 \\ 3 & \dots \end{bmatrix} * 10 + \begin{bmatrix} 5 & 1 \\ 7 & \dots \end{bmatrix} = \begin{bmatrix} 15 & 21 \\ 37 & \dots \end{bmatrix}$$

These operations are executed on each band within the file (red, green, blue, and near-infrared). The resulting spectral radiance matrix for each band is then written to a new image file (See Figure 5), which we will call the *corrected image*. Figure 6 demonstrates gain and offset correction using simulated coefficients for gain.

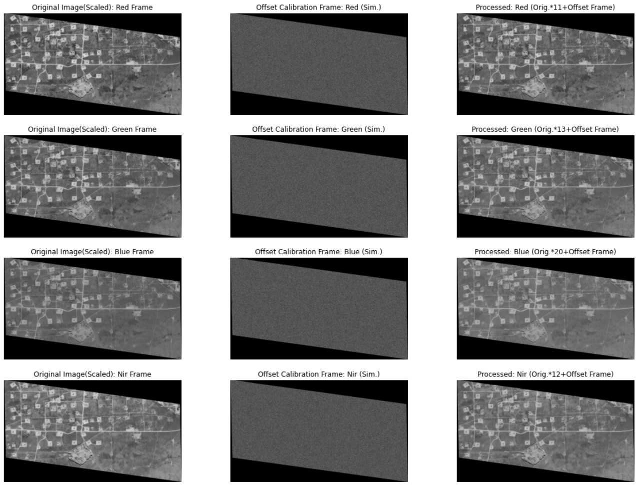
Original Image(Scaled): Red Frame
Offset Calibration Frame: Red (Sim.)
Processed: Red (Orig.*11+Offset Frame)
Original Image(Scaled): Green Frame
Offset Calibration Frame: Green (Sim.)
Processed: Green (Orig.*13+Offset Frame)
Original Image(Scaled): Blue Frame
Offset Calibration Frame: Blue (Sim.)
Processed: Blue (Orig.*20+Offset Frame)
Original Image(Scaled): Nir Frame
Offset Calibration Frame: Nir (Sim.)
Processed: Nir (Orig.*12+Offset Frame)

Figure 6: Frame-based gain and offset processing

Step 3: Show that signature no longer matches. At this point, the corrected image does not check against the original digital signature. Signature checking is also performed with OpenSSL:

```
$ openssl dgst -sha384 -verify <public key> \
  -signature sig.sha384 <corrected image>
```

This check against the corrected file produces a `Verification Failure` message.

Step 4: Inverse processing (end user). The next two steps show the end user's responsibility to verify the data. First, the user must revert the corrected image back to its original (DN) format, by subtracting the offset matrix and dividing by the gain. This process is used to ensure that no other processing happened to the file.

$$DN = (SR - Offset)/Gain$$

Continuing the example above, this would look as follows:

$$(\begin{bmatrix} 15 & 21 \\ 37 & \dots \end{bmatrix} - \begin{bmatrix} 5 & 1 \\ 7 & \dots \end{bmatrix})/10 = \begin{bmatrix} 1 & 2 \\ 3 & \dots \end{bmatrix}$$

Once again, this operation is performed for each band within the image file and a new image file is created with the resulting

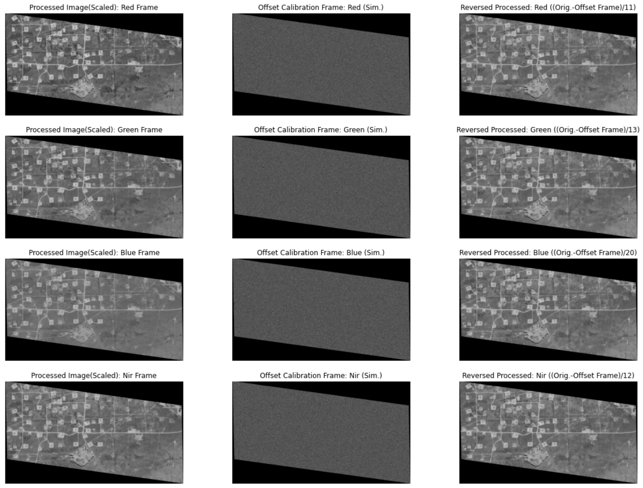
Processed Image(Scaled): Red Frame
Offset Calibration Frame: Red (Sim.)
Reversed Processed: Red ((Orig.-Offset Frame)/11)
Processed Image(Scaled): Green Frame
Offset Calibration Frame: Green (Sim.)
Reversed Processed: Green ((Orig.-Offset Frame)/13)
Processed Image(Scaled): Blue Frame
Offset Calibration Frame: Blue (Sim.)
Reversed Processed: Blue ((Orig.-Offset Frame)/20)
Processed Image(Scaled): Nir Frame
Offset Calibration Frame: Nir (Sim.)
Reversed Processed: Nir ((Orig.-Offset Frame)/12)

Figure 7: Frame-based gain and offset inverse processing

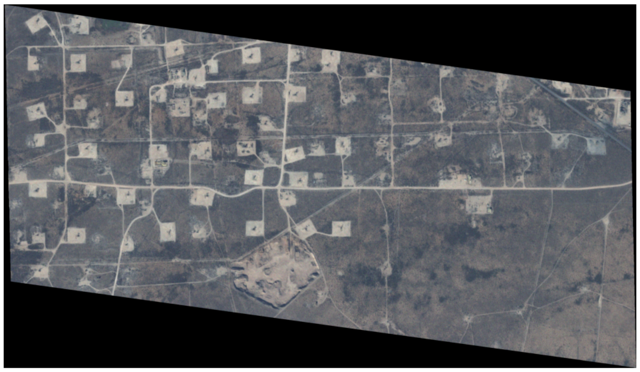

Figure 8: Inversed image (RGB scaled)

DN frames. This inverse processing must be performed using the same coefficients as were used to correct the file in the first place. This reverse processing is demonstrated in Figure 7. In our proof of concept, to ensure congruence with the original file, we use the same TIFF writer and ensure that metadata is not included in the file. We call this file the reverted file and it is depicted in Figure 8.

Step 5: Signature check (end user). If the data has been reconstructed bit for bit against our original capture and was in fact free of malicious processing activities (like data tampering and data injection) it will successfully verify. Signature checking is also performed with OpenSSL:

```
$ openssl dgst -sha384 -verify <public key> \
  -signature sig.sha384 <corrected image>
```

Our proof of concept produces a `Verification OK` message. The in-depth and cryptographically sound checking of the correction processing was successful.

Dead pixel filling. Dead pixel filling was not explicitly demonstrated in this proof of concept. However, the operation is essentially data replacement of an interpolation operation. Operations like averaging the surrounding pixels of a dead pixel to derive its value. In application this could be handled by maintaining an unusable data map and could be approached in a similar manner to a frame base correction. Prior to signing the raw data file the unusable data map could be used to flag non-sensor reads and have all DN values set to zero. The inverse processing would also need to set all unusable data pixel values to zero prior to cryptographic signing. A similar process is effectively demonstrated in our proof of concept by the 'black' borders surrounding the data capture. These regions of the file are defined as unusable data and found in the unusable data map file provided by Planet Labs.[24]

5 CONCLUSION

Third-party processing of satellite imagery is becoming more common place and we need solutions for checking the work of these untrusted or semi-trusted third parties. Our confidence in the correctness of the processing performed must match or exceed the importance of the problem we are solving with the data. In this work, we have applied concepts from recent advances in image authentication [4] to the remote sensing domain and highlighted two gaps: the lack of verification of third-party changes and raw satellite data not being amenable to proposed the difference-checking tools.

We have presented a simple mechanism to check the results of invertible algorithms against the digital signature from the original data. This method is cryptographically secure and allows no leeway for third-parties to deviate from their assigned processing. The primary open challenge is the verification of more complex algorithms, specifically those that are not invertible. We look toward the field of secure outsourced computation to provide cryptographic verification mechanisms for more advanced algorithms. Other open challenges include the verification of algorithm parameters originating from an untrusted party and the establishment of file formats that have the same binary representation for the same content independent of the library or platform that wrote the file.

REFERENCES

[1] [n. d.]. AWS Ground Station. https://aws.amazon.com/ground-station/. ([n. d.]). Accessed: 2022-02-11.

[2] [n. d.]. Azure Orbital | Microsoft Azure. https://azure.microsoft.com/en-us/services/orbital/. ([n. d.]). Accessed: 2022-02-11.

[3] [n. d.]. C2PA Explainer. https://c2pa.org/specifications/specifications/1.0/explainer/Explainer.html. ([n. d.]). Accessed: 2022-02-14.

[4] [n. d.]. The Coalition for Content Provenance and Authenticity (C2PA). https://c2pa.org. ([n. d.]). Accessed: 2022-02-11.

[5] [n. d.]. Content Authenticity Initiative. https://contentauthenticity.org. ([n. d.]). Accessed: 2022-02-16.

[6] [n. d.]. Data Processing Levels. https://earthdata.nasa.gov/collaborate/open-data-services-and-software/data-information-policy/data-levels. ([n. d.]). Accessed: 2022-02-16.

[7] [n. d.]. Ground Network Services - KSAT - Kongsberg Satellite Services. https://www.ksat.no/ground-network-services/. ([n. d.]). Accessed: 2022-02-11.

[8] [n. d.]. Preprocessing Levels and Location Accuracy. https://www.intelligence-airbusds.com/en/8721-preprocessing-levels-and-location-accuracy. ([n. d.]). Accessed: 2022-02-16.

[9] [n. d.]. Project Origin - Protecting Trusted Media. https://www.originproject.info/. ([n. d.]). Accessed: 2022-02-16.

[10] [n. d.]. Three challenges of our digital age. https://www.starlinglab.org/challenges/. ([n. d.]). Accessed: 2022-02-16.

[11] 2022. Planet Imagery Product Specifications. https://assets.planet.com/docs/Planet_Combined_Imagery_Product_Specs_letter_screen.pdf. (Mar 2022). Accessed: 2022-02-15.

[12] Bruno Carpentieri, Arcangelo Castiglione, Alfredo De Santis, Francesco Palmieri, and Raffaele Pizzolante. 2019. One-pass lossless data hiding and compression of remote sensing data. *Future generation computer systems* 90 (2019), 222–239.

[13] Xuexia Chen, Lee Vierling, and Don Deering. 2005. A simple and effective radiometric correction method to improve landscape change detection across sensors and across time. *Remote Sensing of Environment* 98, 1 (2005), 63–79.

[14] Kaimeng Ding, Zedong Yang, Yingying Wang, and Yueming Liu. 2019. An improved perceptual hash algorithm based on u-net for the authentication of high-resolution remote sensing image. *Applied Sciences* 9, 15 (2019), 2972.

[15] Ronald Eastman. 1999. Guide to GIS and Image Processing Volume 2. 1 (01 1999), 27–39.

[16] Ian Goodfellow, Patrick McDaniel, and Nicolas Papernot. 2018. Making machine learning robust against adversarial inputs. *Commun. ACM* 61, 7 (2018), 56–66.

[17] Sean Kinser, Pete de Graaf, Matthew Stein, Frank Hughey, Rob Roller, David Voss, and Amanda Salmoiraghi. 2020. Scoring Trust Across Hybrid-Space: A Quantitative Framework Designed to Calculate Cybersecurity Ratings, Measures, and Metrics to Inform a Trust Score. In *The 34th Annual Small Satellite Conference*.

[18] Paweł Korus. 2017. Digital image integrity–a survey of protection and verification techniques. *Digital Signal Processing* 71 (2017), 1–26.

[19] ShiYue Lai and Rainer Böhme. 2011. Countering counter-forensics: The case of JPEG compression. In *International Workshop on Information Hiding*. Springer, 285–298.

[20] Joseph M Piwowar. 2001. Getting your imagery at the right level. *Cartouche* 41 (2001).

[21] Christopher X Ren, Amanda Ziemann, James Theiler, and Juston Moore. 2021. Deepfaking it: experiments in generative, adversarial multispectral remote sensing. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, Vol. 11727. 117270M.

[22] D Sal, Manuel Graña, and Alicia d'Anjou. 2006. A moga to place the watermark in an hyperspectral image. In *2006 IEEE International Symposium on Geoscience and Remote Sensing*. IEEE, 783–786.

[23] Robert A Schowengerdt. 2006. *Remote sensing: models and methods for image processing*. Elsevier.

[24] Planet Team. 2017–. Planet Application Program Interface: In Space for Life on Earth. (2017–). https://api.planet.com

[25] Abraham Noah Wu and Filip Biljecki. 2021. GANmapper: geographical content filling. *arXiv preprint arXiv:2108.04232* (2021).

[26] Yang Yang, Xindi Huang, Ximeng Liu, Hongju Cheng, Jian Weng, Xiangyang Luo, and Victor Chang. 2019. A comprehensive survey on secure outsourced computation and its applications. *IEEE Access* 7 (2019), 159426–159465.

[27] Xingang Zhang, Haowen Yan, Liming Zhang, and Hao Wang. 2020. High-resolution remote sensing image integrity authentication method considering both global and local features. *ISPRS International Journal of Geo-Information* 9, 4 (2020), 254.

[28] Bo Zhao, Shaozeng Zhang, Chunxue Xu, Yifan Sun, and Chengbin Deng. 2021. Deep fake geography? When geospatial data encounter Artificial Intelligence. *Cartography and Geographic Information Science* 48, 4 (2021), 338–352.

[29] M Tarek Ibn Ziad, Amr Alanwar, Moustafa Alzantot, and Mani Srivastava. 2016. Cryptoimg: Privacy preserving processing over encrypted images. In *2016 IEEE Conference on Communications and Network Security (CNS)*. IEEE, 570–575.