2022 NDIA MICHIGAN CHAPTER
GROUND VEHICLE SYSTEMS ENGINEERING
AND TECHNOLOGY SYMPOSIUM
CYBERSECURITY OF GROUND SYSTEMS TECHNICAL SESSION
AUGUST 16-18, 2022 - NOVI, MICHIGAN

BLOCKCHAIN INTEGRITY CHECKS FOR SOFTWARE (BICS)

George G Fortney
SAIC, Sterling Heights, MI

ABSTRACT

Currently there is no method to ensure that the software loaded on a vehicle has been compromised at the software level. Common practice is to use physical port security to secure all network and data bus connection points with physical devices requiring tool, keys, or damage to tamper evident devices to prevent, inhibit, or discourage unauthorized connection; turn off access to the ports in the BIOS and password protect the BIOS. As well as give non-admin access to user accounts and password protect the operating systems. All these countermeasures help to prevent access but there is no way to tell if the software was compromised if not detected by these methods. Blockchain technology ensures that the software has not been compromised by comparing a hash generated at start up and comparing it to the distributed ledger. This technology helps to bring Warfighter technology into the future.

1. INTRODUCTION

Blockchain methodology is a decentralized, distributed and digital ledger that is used to record transactions across many computer systems so that the record cannot be altered retroactively without the alteration of all subsequent blocks and the consensus of the network. It is a two-part system. It uses an encryption algorithm to create a unique hash and writing this hash to the ledger. The algorithm detects even the slightest change in the input as illustrated in figure 1 below [1].

This methodology ensures that no one entity can change the ledger as they would have to control at least 51% of all the ledgers. The appending of blocks rather than updating ensures a chain of custody and makes it even more difficult to compromise the Blockchain.

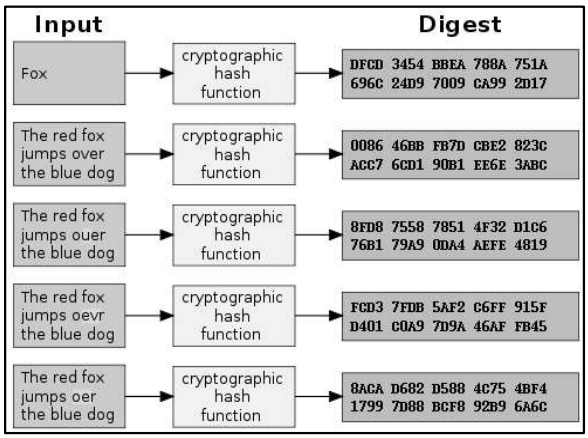

Figure 1 Blockchain Hashes reveal even the slightest changes

A hash is a one way function. This means that one cannot take a hash and reverse engineer it to reveal the un-hashed data.

2. DISTRIBUTED LEDGER TECHNOLOGY

In Blockchain usage, a distributed ledger is a database that is consensually shared and synchronized across networks spread across multiple sites, institutions or geographies used for recording the transaction of assets in which the transactions and their details are recorded in multiple places at the same time. Unlike traditional databases, distributed ledgers have no central data store or administration functionality. This enables the security and validation as ledgers can be compared to see if any one or more have been compromised.

3. MERKLE TREE AND MERKLE ROOT

The Merkle tree concept was named after Ralph Merkle who patented it in 1979. A hash tree or Merkle tree is a tree in which every leaf node is labeled with the hash of a data block and every non-leaf node is labeled with the cryptographic hash of the labels of its child nodes. Hash trees allow efficient and secure verification of the contents of large data structures. A Merkle tree is recursively defined as a binary tree of hash lists where the parent node is the hash of its children, and the leaf nodes are hashes of the original data blocks (figure 2)

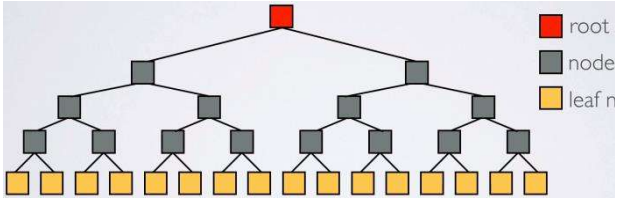

Figure 2 Binary Merkle Tree Example [2]

Demonstrating that a leaf node is a part of a given binary hash tree requires computing a number of hashes proportional to the logarithm of the number of leaf nodes of the tree. This differs with hash lists, where the number is proportional to the number of leaf nodes itself.

One could just hash the entire datum and then hash the hashes to get a root hash but this has problems in the amount of effort needed to verify and validate. For example if one wanted to prove M6 is not tampered with then the M6 message and all 15 of the other hashed messages need to be sent for validation. To validate someone has to hash M6 and then hash the M6 hash with all 15 other hashes to arrive at a root hash to compare to the ledger (figure 3)

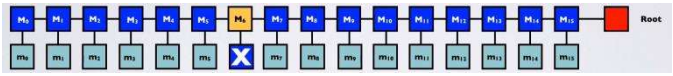

Figure 3 Non-Merkle Tree Hash [2]

A Merkle Tree only needs the M6 message and four other hashes (shown in blue) to validate the M6 message (figure 4). The M6 message is hashed and with the other four hashes creates the root hash to be compared in the distributed ledger. Because it is a binary tree structure if you have one of the two pairs you can hash right up to the root hash in far less steps than a non-Merkle Tree hash. Not only does this use less computing power but you can arrive at a conclusion far quicker.

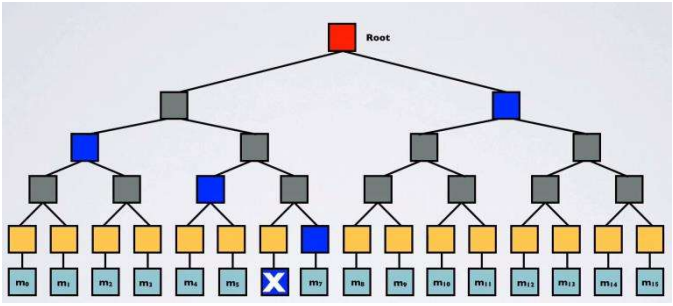

Figure 4 Merkle Tree Hash [2]

4. PROBLEM STATEMENT

Current methods of anti-tamper as stated in the abstract can be compromised in such a way that there is no evidence of tampering. If this occurs, malware, Trojan, key logger, false information or a destructive payload may enter the system unnoticed. These anti-tamper methods are reactive not proactive. A method to ensure executables and libraries are not compromised before they are used is needed to proactively ensure integrity of the software. The key to enabling this so that it is usable is the time it takes to check and then find the compromised software. The Merkel Tree's recursive nature and on board processing ensure that this can be an effective tool to cyber secure a vehicle at the software level.

5. SPECIFIC ISSUE #1

There needs to be a method of distributing the data in on-the-move geographically dispersed vehicles and data stores. This network needs to be secure and have the needed bandwidth and availability to support multiple ledgers and a means to synchronize them.

6. SPECIFIC ISSUE #2

On a vehicle platform (manned, un-manned, ground, air or sea) there can reside many software systems. There needs to be a way to quickly and securely check the hashes on the software and then the root node to ensure that the vehicle software system is not compromised. Proactively checking reduces the risk to the mission.

7. BLOCKCHAIN INTEGRITY CHECKS FOR SOFTWARE (BICS)

BICS is a proactive method to determine if software has been compromised before it is used. Together with current anti-tamper methods creates a security profile for the vehicle that can be validated against a distributed ledger to ensure the vehicle software has not been compromised.

By creating a secure military distributed ledger of these hashes which represent software on all vehicles one could compare the ledger hash to a hash generated upon start up to ensure the software has not been compromised. The processing of the hash(es) is done on the vehicle to arrive at the Parent Node hash. This comparison with the distributed ledger is a hash that is sent as text (this message can be encrypted for additional security). This creates a lightweight message that can transverse over a low bandwidth signal. Once a parent node has received the message it is compared to the read only ledger to verify and validate the hash. When all the vehicles has received and compared the entire Parent node hashes any discrepancy will be discovered. For example of hash length the Bitcoin blockchain uses a combination of SHA-256 (64-character strings) and SHA-512 (128-character strings) to encrypt data. The higher the SHA number generally, the more complex and secure the hash.

8. SOLUTION INTRODUCTION

When the vehicle software is installed in depot the encryption algorithm will be run on the executable software and libraries to create a unique hash. This hash will be written to read only media on the vehicle and in the vehicle borne ledger. These values will also become part of the distributed ledger that will reside on other vehicles and controlled data stores. The vehicle local ledger will contain these values as well as values for other vehicles.

Prior to mission start the vehicle will hash all executables and libraries. It will create a root hash and compare it to the root hash on the read-only media. If they match the

mission continues. If they do not then the mission commander is notified and further investigation can continue.

Once the vehicle is underway and coms established the vehicle ledger will synchronize with the other distributed ledgers to validate the root hash. The root hash represents the total vehicle profile and requires only a single comparison to the ledger entries in the distributed ledger. This enables a very quick binary decision on the security status of the vehicle. This mitigates insider threat in case the read-only media has been compromised and replaced. The vehicle will have its root hash compared to the hash on file in the distributed ledger. If it does not match the root hash on file, notification will go to command and that vehicle will have additional investigation to determine what has been compromised and how.

To do root cause inspection one only has to check hashes at consecutive levels below the root hash. If the Level 1 hashes match the ledger then you move to the Level 2 hashes. If one does not match you only have to move down that side of the tree to find the hash that does not match. Very quickly you can determine what hash is compromised without having to hash the entire tree **(figure 5)**.

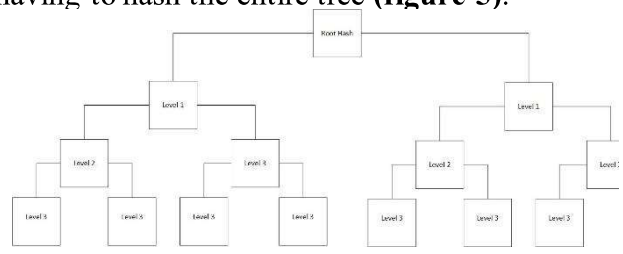

Figure 5 Hash levels of the Merkle Tree

9. REQUIRED ADDITIONAL HARDWARE

Hashing and comparing the software comes at a computational cost. Fortunately processing costs have been coming down over the years and with increased chip density and smaller size, industry is creating smaller faster processors. This technology can also be hosted in many of the current devices in use on modern vehicles like the Mounted Family of Computer Systems (MFoCS) displays or other small form factor headless boxes.

BICS lends itself to the Standardized A-Kit/Vehicle Envelope (SAVE) with standard mounting location and set of physical interfaces for integration of Command, Control, Computers, Communications, Cyber, Intelligence, Surveillance, and Reconnaissance (C5ISR) systems into Army ground vehicles. SAVE is intended to provide long term predictability and stability for vehicle and systems development. SAVE was initially developed for radios but also applies to computers and other systems. SAVE is a subset of the overall PEO GCS Common Infrastructure Architecture (GCIA). GCIA prescribes a modular open systems approach (MOSA), using Vehicular Integration for C4ISR/EW Interoperability (VICTORY) and other open interface standards for data sharing, enabling CMOSS and other MOSA benefits. SAVE is one of the physical portions of system integration within GCIA. SAVE also applies to PEO CS&CSS and other platforms that have no dependency on GCIA. Platforms like the JLTV recomplete which now has a Next Gen Vehicle Architecture (NGVA) requirement for a Vehicle Real-Time Embedded Computer Hub (VRTECH) can host this technology without the addition of another platform processing box.

10. USE ON LEGACY AS WELL AS NEW AND EMERGING PLATFORMS

As almost every vehicle contains some software this technology can be applied to legacy, new and emerging platforms. As we see older platforms receive technology upgrades to meet the MOSA requirement of

Title 10 Section 805 and as platforms become more software centric and tied into the tactical networks we see a need to cyber secure the platform software. The ability to do this proactively significantly reduces mission risk due to compromised software.

11. SUMMARY

The use of Blockchain and Merkle tree can be employed to proactively (not reactively) determine if a vehicle has any of its software compromised. This technology can be used on legacy, new and emerging platforms as well as data centers and robotic entities. It can be used to meet SAVE requirements. It can be put on an OpenVPX card in a CMOSS Mounted Form Factor (CMFF) chassis; it can be hosted in existing hardware or a small form factor processor box. As we rely more and more on Multi-Domain Operations (MDO) and the many sensor feeds of land, air, water and space the need to secure the message transports as well as the organic platform software is a necessity.

12. PRIOR ART

While the use of Blockchain for authentication and authorization has been socialized, the specific use of integrity checks of executables and libraries prior to running to validate authenticity has not. Blockchain technology is the basis and building block of BICS but it is not the invention itself, it is the enabling technology. The United States Patent and Trademark Office issued SAIC US patent 10,726,000 on Jul 28, 2020 with George Fortney as the inventor.

13. REFERENCES

[1] Curran, Brian, What is a Merkle Tree? Beginner's Guide to this Blockchain Component, July 9, 2018, from https://en.wikipedia.org/w/index.php?title=Cryptographic_hash_function&oldid=868055371

[2] Lie, R. (n.d.). Merkle Tree [PDF], IOTA. Retrieved December 30, 2018, from https://www.mobilefish.com/developer/iota/iota_quickguide_tutorial.html