AFRL-RI-RS-TR-2020-077

BLOCKCHAIN TECHNOLOGY WITH TERNARY CRYPTOGRAPHY

NORTHERN ARIZONA UNIVERSITY

MAY 2020

FINAL TECHNICAL REPORT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

STINFO COPY

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE

■ **AIR FORCE MATERIEL COMMAND** ■ **UNITED STATES AIR FORCE** ■ **ROME, NY 13441**

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09. This report is available to the general public, including foreign nations. Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-RI-RS-TR-2020-077 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE CHIEF ENGINEER:

/ S /
TODD N. CUSHMAN
Work Unit Manager

/ S /
JAMES S. PERRETTA
Deputy Chief, Information
Exploitation & Operations Division
Information Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
MAY 2020	FINAL TECHNICAL REPORT	MAR 2019 - MAR 2020

4. TITLE AND SUBTITLE

BLOCKCHAIN TECHNOLOGY WITH TERNARY CRYPTOGRAPHY

5a. CONTRACT NUMBER
N/A

5b. GRANT NUMBER
FA8750-19-1-0024

5c. PROGRAM ELEMENT NUMBER
62788F

6. AUTHOR(S)
Bertrand Cambou
Michael Gowanlock
Julie Heynssens
Saloni Jain
Duane Booher
Ian Burke
Jack Garrard
Christopher Philabaum

5d. PROJECT NUMBER
BC2S

5e. TASK NUMBER
BC

5f. WORK UNIT NUMBER
TC

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Northern Arizona University
1295 South Knoles Drive
Flagstaff, Arizona, 86011

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Research Laboratory/RIGA
525 Brooks Road
Rome NY 13441-4505

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/RI

11. SPONSOR/MONITOR'S REPORT NUMBER
AFRL-RI-RS-TR-2020-077

12. DISTRIBUTION AVAILABILITY STATEMENT
Approved for Public Release; Distribution Unlimited. This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09

13. SUPPLEMENTARY NOTES

14. ABSTRACT

We are proposing end-to-end schemes by inserting tamper resistant devices in the hardware of the peripheral devices, and with the use of ternary cryptography. The tamper resistant devices, which are designed with nanomaterials, act as Physical Unclonable Functions to generate secret cryptographic keys. One-time use public-private key pairs are generated for each transaction with cryptographic schemes incorporating a third logic state to mitigate man-in-the-middle at-tacks. The generation of these pairs is compatible with post quantum cryptography. Finally, we are proposing the use of noise injection techniques with high performance computing to in-crease the security of the system. We present prototypes to demonstrate the feasibility of these schemes, and to quantify the relevant parameters. We conclude by presenting the value of blockchains to secure the logistics of strategic manufacturing operations.

15. SUBJECT TERMS

Physical unclonable functions, Blockchain, ternary cryptography

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			**TODD N. CUSHMAN**
U	U	U	UU	34	19b. TELEPHONE NUMBER (*Include area code*) **315-330-4895**

Table of Contents

LIST OF FIGURES

LIST OF TABLES

1.0 SUMMARY

Blockchain technology is a game changer enhancing security for the supply chain of smart manufacturing. This technology enables the tracking and recording of the history of each transaction in a ledger stored in the cloud that cannot be altered, and when combined with digital signatures, verifies the identity of the participants with its non-repudiation capabilities. One of the weaknesses of this technology is the difficulty preventing malicious participants from gaining access to public-private key pairs. Groups of opponents often interact freely with the network, in particular when cloud-based methods manage the key pairs. Therefore, we are proposing end-to-end schemes by inserting tamper resistant devices in the hardware of the peripheral devices, and with the use of ternary cryptography. The tamper resistant devices, which are designed with nanomaterials, act as Physical Unclonable Functions to generate secret cryptographic keys. One-time use public-private key pairs are generated for each transaction with cryptographic schemes incorporating a third logic state to mitigate man-in-the-middle attacks. The generation of these pairs is compatible with post quantum cryptography. Finally, we are proposing the use of noise injection techniques with high performance computing to increase the security of the system. We present prototypes to demonstrate the feasibility of these schemes, and to quantify the relevant parameters. We conclude by presenting the value of blockchains to secure the logistics of strategic manufacturing operations.

2.0 INTRODUCTION

The purpose of the introduction is to bring relevant background information related to blockchain technology, smart manufacturing, and ternary Physical Unclonable Functions (PUFs). Blockchain technology is based on one-way cryptographic functions, the hash functions that are considered as extremely safe. The potential limitations of blockchain is the digital signature, and public key distribution, which are important to ensure that only legitimate users are participating in the network. The research question, the core of this paper, is to demonstrate that it is possible to generate a new public key for each transaction with acceptable latencies. Therefore, the leakage of the key is less critical considering that the same key is never used twice. In this paper, we define "smart manufacturing" as the process to use a network of subcontractors and suppliers that interact through the cloud, and open internet communications. This is a field of use where blockchain technology brings value: non-alterable ledgers, and non-repudiable transactions. Mainstream manufacturing operations, which actually manufacture products, are not considered here, nor are the prevention of Trojans and counterfeits in the manufacturing of semiconductor devices. The objective of the work presented in this paper is to develop and characterize an end-to-end cryptographic system, hardware and software, that uses a network of PUFs inserted in distributed client devices. At the server level, the algorithms are based on ternary cryptographic schemes in which the third state includes instable states, as well as perfectly stable states that increase the level of randomness, i.e. entropy; the logic used at the client level is binary. This work is agnostic regarding the type of PUF needed; we selected SRAM-based PUFs with commercial components because they are well known, and easy to use. The server-client device asymmetry of the protocol enables the use of extremely powerful computers at the server level, High Performance Computing (HPC), while keeping low power embedded microcontrollers at the client level.

2.1 Blockchain technology

In 2008 the paper published under the name Satoshi Nakamoto "Bitcoin: a peer-to-peer electronic cash system" [1] was no less than a revolution in the financial world. Two relatively mature technologies, the hash functions with Merkle trees, and Digital Signature Algorithms (DSA) [2-9] were successfully integrated in the architecture to track all transactions in a virtual public leger that is non-alterable after emission, and non repudiable. The innovation behind Bitcoin included a trust management scheme using aspects of game theory to allow the generation, i.e. mining, of additional cryptocurrencies. The "peer-to-peer" aspect of the scheme can be controversial for some, due to its inability to prevent suspicious users to participate. However, many believe that the blockchain technology has the potential to evolve and gain as much importance as existing internet technology.

Blockchain technology with a hashing function such as SHA-2 can protect the data flow needed to track transactions for applications such as personal information, finance, transportation, logistic and smart manufacturing, see Fig.1. The digital signature, securing Bitcoins is based on Elliptic Curve Cryptography (ECC) [10-11], which also has the potential to secure the Internet of Things (IoTs) infrastructure. The underlying assumption is that the entire infrastructure of IoTs is homogeneous, with each node being protected by a crypto processor handling the hashing, and having a secure non-volatile memory to store the cryptographic keys. The DSA can be based, but not limited to, an extended finite field ECC (also called Galois Fields ECC: GF-ECC), which operates at lower power than the older finite field ECC, and Rivest Shamir Adelman (RSA) [12]. Malicious side channel attacks, and physical hijacking of IoT nodes, can expose the private keys,

thereby compromising the security of the infrastructure. The distribution of public-private key pairs in such an environment can be risky. Without the reliable protection of private keys, the DSA of the blockchains is vulnerable, and the technology loses its value. ECC is also not quantum computing resistant [13] and should be replaced by alternate DSA methods such as the ones offered by hash-based, lattice, code, and multivariate cryptography [14-16].

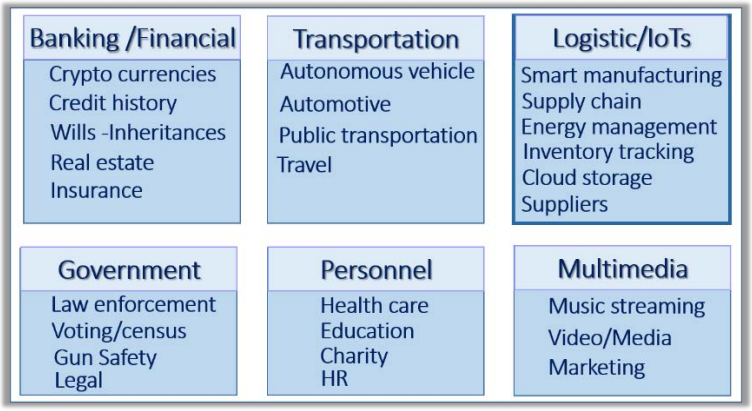

Figure 1: Example of fields that could benefit from the blockchain technology.

2.2 Smart Manufacturing

Manufacturing operations are outsourcing an increasingly large proportion of the supply chain through networks of interconnected multi-echelon subcontractors that interact remotely with the worldwide web [17-20]. The risks of cyberattacks are rapidly accelerating, in addition to the emergence of counterfeit products, poor quality elements of unknown sources, and the insertion of hardware Trojans, malwares, worms, and viruses in electronic components. The traditional centralized mechanism cannot control the full supply chain due to a conflicting interest with certain suppliers, and the vulnerability of heterogeneous Information Technology Systems (ITS) to malicious entities. The ITS supporting a manufacturing operation is usually different from the ITS of the suppliers, which in turn are relying on subcontractors having their own ITS.

Blockchain technology with DSA is bringing fully transparent possibilities [21], with traceable, non-alterable, and non-repudiable transactions, and decentralized governance allowing multiple layers of suppliers and subcontractors to participate. In order to avoid unwelcomed suppliers, the management of the pubic keys of the DSA can be tracked by Certificate Authorities (CA) that are trusted by the manufacturing entity [22], as shown in Fig.2. The list of the valid public keys is available in the cloud. Therefore, all trusted suppliers and subcontractors can share, and directly verify the transactions transmitted by their peers. The suppliers hash their transactions, sign them with their private key, and post the resulting information in the cloud. The tracking can incorporate information that is required by the manufacturing operation such as origin, quantity, quality, proof of sustainability [23-24], intellectual property [25], copyrights [26], and a list of subcontractors. It has been suggested that the blockchain technology can also track the identification of the spare parts with RFID's and tokens [27]. The use of blockchain to secure traditional manufacturing that actually manufacture products in not considered in this paper. The reader interested to secure integrated circuits are invited to read the excellent summary presented by Guin and DiMase [28].

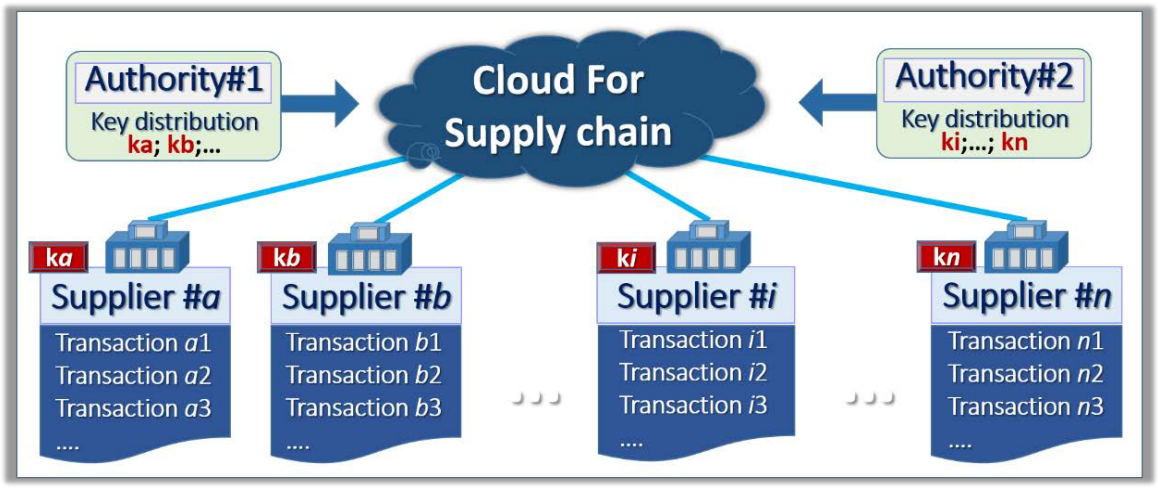

Figure 2: Example of certificate authorities distributing valid public keys to a constellation of suppliers.

2.3 Ternary Physical Unclonable Functions

PUF technology exploits the variations created during fabrication to authenticate each device from any other device, acting as hardware "fingerprint" [29-34]. Solutions based on PUFs embedded in the hardware of each supplier node, see section 2.0, can mitigate the risk of an opponent reading the keys stored in the non-volatile memory, when the keys for the DSA of the blockchains are generated on demand. Authentication protocols based on PUFs, embedded in each IoT node [35-36], are effective with i) intra-PUF stability, ii) inter-PUF randomness, and iii) small enough drifts of the PUF characteristics over time. Nanotechnology Memory structures [37], SRAM [38], DRAM [39], non-volatile memories and Flash [40-41], ReRAM [42-44], and MRAM [45], are suitable to generate strong PUFs. In the protocols selected in this work, the initial readings of the PUFs, also called the "initial responses", are the result of computations, and statistical analysis to sort out the cells that are solidly identified as logical "0" or "1", and the unstable fuzzy cells that are identified with an additional third state "X". During the enrollment cycle of PUFs, the three potential states (0,1,X) of the PUF's cells are downloaded in a database or look up table in the server. This operation has to be done once in a secure environment. The PUF "responses" are the data streams generated by the PUFs during the life of the client devices. In the protocols developed in this paper the client devices read the PUFs only once, therefore only handle binary states (0,1). During authentication cycles in which the PUFs are "challenged": the "initial responses", which are stored in the server and have binary states, are compared with the "responses" read from the client device. This results in matching "Challenge-Response-Pairs" (CRP), when the "responses" are the same as the "initial responses".

The PUFs can age, and be subject to environmental drift, electro-magnetic interferences, and aging. When the CRP error rates are below 10%, the false rejection rate (FRR), and false acceptance rate (FAR) are usually acceptable, and the PUFs can be used as part of authentication protocols to protect cyber physical systems. The use of PUFs to generate cryptographic keys from the responses, a focus of this work, is more challenging than generating responses for authentication. A single-bit mismatch in a cryptographic key is not acceptable for most encryption protocols. The use of error correcting methods[46-48], helper data [49-50], and fuzzy extractors[51-52] is therefore needed to achieve the zero error level. Error correcting schemes burden client devices, as they consume additional computing power to run fuzzy extraction, and error

correcting codes. Such protocols also increase the vulnerability to differential power analysis, leaking information to the opponents. With ternary PUFs [53-54], when the fuzzy "X" states are blanked, CRP error rates are typically reduced by two orders of magnitudes to the 10^{-3} range, which greatly simplifies the entire error correcting protocols. Conversely, when the "X" are selected, CRP error rates are higher, which can be turned as a feature when HPC are used to handle the erratic keys as presented below.

2.4 Outline of the presentation

Shown in section "Methods, Assumptions and Procedures", is an overview of the proposed architecture, with the objective to enhance the cybersecurity of Smart Manufacturing. PUFs are inserted in the hardware of each supplier that interacts with the cloud, to authenticate the suppliers, and to generate the one-time public-private key pairs. Section "results and discussion", is describing the experimental work, in which a prototype was developed to validate the concept. This section includes the characterization of the important parameters of the protocol, including latencies and error rates. The limitations in term of security of the protocol is presented. The potential value of High-Performance Computing (HPC) to strengthen the protocols is analyzed experimentally. The considerations used to deploy such a blockchain technology to the supply chains of strategic smart manufacturing operations is presented at the end of the section "results and discussion". The SRAM-based PUF technology is potentially sensitive to side channel analysis, and the leakages of the keys. However, this is an excellent technology in terms of entropy, with relatively low CRP error rates and stability, which enabled the authors to develop and characterize the end-to-end cryptographic protocols presented here.

3.0 METHODS, ASSUMPTIONS AND PROCEDURES

3.1 Overview

The blockchain-based architecture presented above, has the potential to enhance the cybersecurity of Smart Manufacturing; however, it can also create a sense of false security when the public-private key pairs of certain suppliers are compromised. As shown in Table 1, the security of the blockchains is due to the combination of several technologies [55-57].

- The first layer of security of blockchain technology is coming from hash pointers, and Merkle trees to generate non-alterable public ledgers. As existing hash algorithms such as SHA-2, and SHA-3 are considered safe, no further improvements are suggested.
- The second layer of Table 1, digital signatures, storing and handling of the public-private key pairs, can become a major liability for Smart Manufacturing. The prime objective of the work presented in this paper is to enhance security in this area.
- The third layer of security shown in Table 1, which is based on trust mechanisms relying on peer-based groups, is a vibrant field of research [58-62]. However, it is highly questionable that the manufacturing of strategic assets such as weapons, planes, and satellites will rely on peer-based trust mechanism.

Table 1: Layers of security offered by the Blockchain technology

1- Hash Pointers and Merkel Trees ➔ Blockchains Chain of messages with non alterable public ledgers
2- Digital signatures with public/private key pairs Identification of the users & non repudiation
3- Trust mechanisms Majority rules against small participants Maximize revenues by following the rules

The overall architecture proposed is shown in Fig.3, includes the following protections:

- The Ternary Addressable Public Key Infrastructure (TAPKI) for the generation of a new private key from the PUFs for each transaction, During enrollment, the image of the PUF, the challenges, are stored in look up table of the CA. New keys are generated at each transaction by the TAPKI.
- The generation of public keys with asymmetrical cryptography, and a path toward post quantum cryptography;
- Response Based Cryptographic (RBC) scheme to verify that the public keys generated from the private keys and the TAPKI are valid, with acceptable error rates;
- A cryptographic scheme that use noise injection in the PUF, and High-Performance Computing to mitigate attacks from opponents that do not have access to similar computing power. The RBC verifies the validity of the public keys and post them in a public ledger.

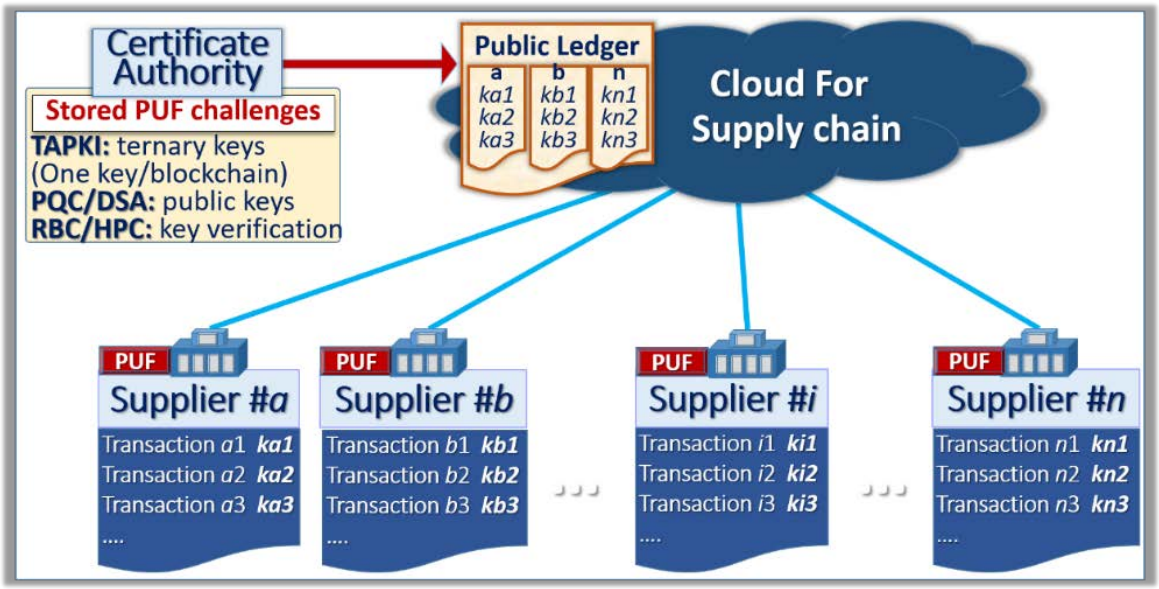

Figure 3: PUFs distribution, and ternary cryptography, to generate new keys at each transaction.

3.2 Ternary Addressable Public Key Infrastructure (TAPKI)

The term Public Key Infrastructure (PKI), has been used to describe an environment where each communicating party is equipped with two keys, the private key is secret, and the public key is public information. With asymmetrical cryptographic schemes, the messages encrypted with one of the two keys is decrypted with the second key. In the case of the DSA for blockchains, the author of a blockchain encrypts the signatures with the private key, in such a way that anyone can verify the signature with the public key. These private keys can be stolen by various methods, during the generation, distribution, and storage of the keys, as well as during encryption/decryption cycles. The objective of the TAPKI [63], see Fig 4, is to provide additional security to PKI by generating a new private key for each blockchain transaction from distributed ternary PUFs.

To sign a new blockchain, the server transmits the information needed by the supplier to generate a new private key from the Ternary PUF. The information is shared, with a communication channel that is assumed insecure. The random number ***T*** generated at each transaction by the TAPKI concurrently feeds two hashing elements at the server and the supplier levels. The number ***T*** is concatenated with the password **PW**, and additional multifactor schemes, to generate the message digest $\mathbf{A_i}$. The message digest is turned into a particular address $\{\mathbf{X_i}, \mathbf{Y_j}\}$ of the PUF array, see Fig.4. For example, if the PUF array contains 1024x1024 cells for a one Mega bit memory, 10 digits of the message digest are used for $\mathbf{X_i}$, and 10 digits for $\mathbf{Y_i}$.

Only the server with the appropriate look up table, and the client device with its PUF can independently generate the same private key for the TA-PKI protocol; a third party without the same look up table cannot find the same address $\mathbf{\{X_i, Y_j\}}$. At which address, the server extracts a ternary stream $\mathbf{C_i}$ from the initial responses stored in the look up table, and the supplier reads a binary stream $\mathbf{C_i'}$ from the PUF. Both streams should be similar, with some errors present in $\mathbf{C_i'}$ due to the natural physical variations of the PUF.

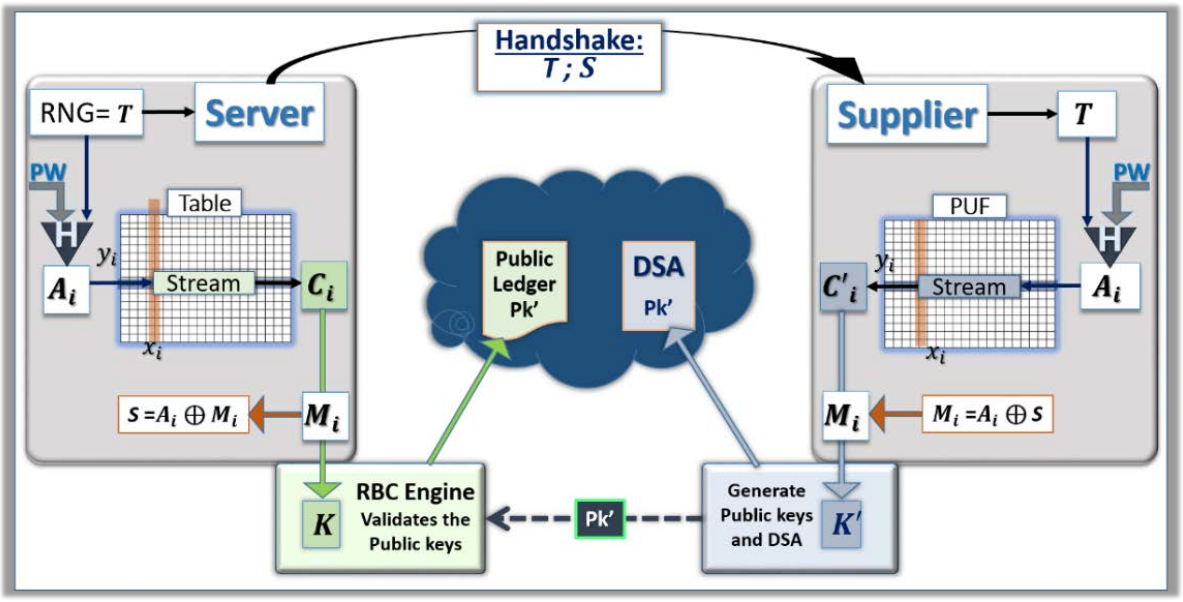

Figure 4: Generation and verification of public keys with TAPKI and RBC.

The server generates a mask M_i to blank the ternary cells, and then generates a key K that contains only the solid cells with "0"s and "1"s. The mask is XORed with the message digest A_i to generate the stream S, which is communicated to the supplier, as part of the handshake. This XOR operation is a way to encrypt the mask M_i; such encryption method is also called one-time pad because both the mask and the message digest are only used once during each handshake. The knowhow of S will not disclose either M_i or A_i. The supplier can XOR again the stream S with the message digest A_i to recover the mask. Both the server with the look up table, and the client device with the PUF will explore the same portion of the array with A_i, and mask the same cells with M_i to independently generate the keys K (server), and K' which should be close to each other's when the CRP error rates are low.

Shown in Fig.5 is an example of the sequential scheme used to generate a new public key ***k4j*** for the blockchain ***Block4j***, with a random number ***RN4j***, and ***Mask4j***. This figure simplifies the protocol and does not include the protection of the mask with the XOR presented above.

It is noticeable that the fuzzy cells of the ternary PUF, and associated ternary states, offer a protection against man-in-the-middle attacks sending their own handshakes. When an opponent sends random streams T_f and S_f to the supplier, the errors of the key K' generated from data stream C_i' will contain high error rates, because an invalid mask does not blank the fuzzy cells anymore. The client needs to have the right password PW to retrieve the right message digest A_i from a random number T_f. The man-in-the middle does not know the mask, PW, and A_i, so the stream S_f will be random, and $S_f \oplus A_i$ will be an invalid mask. Therefore, the erratic keys generated by the supplier under such handshake will not be recognizable by the server.

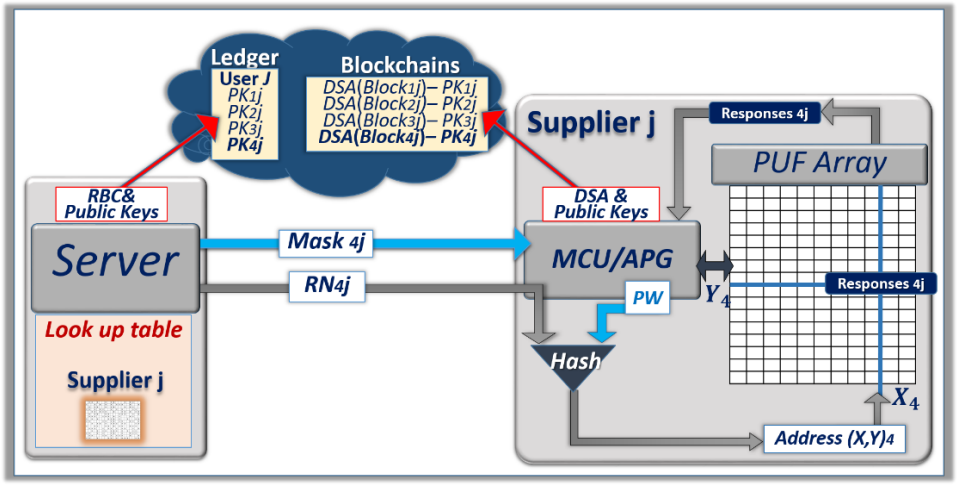

Figure 5: Architecture of the client device. Sequential scheme to generate new public keys for each blockchain.

3.3 Generation of the public keys – PQC considerations

In most PKI schemes, in which blockchains are secured with DSA, the public keys are generated by private keys with ECC. These private keys are natural numbers, typically 256-bits long. The primitive element of their cyclic group of the elliptic curve is multiplied by these natural numbers to find the public key. The reverse computation, finding private key from public keys, requires enormous processing power to independently uncover the private keys, and to prevent a third party from breaking the encryption method. In the proposed protocol, the TAPKI is acting as a key exchange mechanism for the private keys, which uses the ternary PUFs, while the public keys are generated by an asymmetrical cryptographic scheme such as ECC. The TAPKI is a generic method, we used ECC to design the prototype described in section 2.2, and to validate the overall architecture. With ECC, a single bit mismatch between the private key **K'** generated from the PUF, and the private key **K** generated from the look up table will result in entirely different public keys. The RBC scheme mitigates this problem, as presented below in section 2.4. The natural mismatch of the private keys **K'** and **K** is considered a feature in this work, see section 2.5, that is leveraged to enhance the security of the network of suppliers for Smart Manufacturing.

It is now anticipated that quantum computers (QC) will be able to break ECC, when the technology to master enough quantum nodes becomes available. Plans to replace ECC by PQC schemes have been developed for the DSA of blockchain technology, even if the timing of the availability of powerful QC is highly speculative [64-67]. The efforts required to implement the blockchain technology for Smart Manufacturing is such that a plan to migrate to PQC DSA is needed, even if non-PQC schemes are used at first. The project driven by the National Institute of Standards and Technology (NIST) has pre-selected nine potential PQC-DSA candidates during the round-2 phase of the program [68]: SPHINCS, and PICNIC with hash-based cryptography [69-73]; CRYSTALS, FALCON, and qTESLA with lattice cryptography [74-77]; GeMSS, LUOV, MQDSS, and Rainbow with multivariate cryptography [78-81]. The TAPKI protocol will be applicable to these PQC DSA schemes if it is possible to generate the private keys from the ternary PUFs, and generate the public keys from these private keys, along with the information shared openly during the handshake between server and suppliers.

Hash based PQC DSA (SPHINCS, PICNIC). The PQC DSA algorithm SPHINCS+ relies on well-known hash-based signature schemes, Winternitz One Time Signature (WOTS), Forest of Random Subsets (FORS), and a set of Merkle trees called hyper-trees [69]. The size of these hyper-trees are such that an almost infinite number of signatures can be generated with the same tree. The sizes of the keys are relatively small (256 to 512 bits). PICNIC uses zero-knowledge algorithms [70]. The disadvantage of hash-based cryptography is the high latencies to sign and verify, due to the need to perform large quantities of hashing, so the size of the signatures could be very large. The algorithms are straightforward to implement using TAPKI for Smart Manufacturing. The private keys are generated from the ternary PUF, as done with ECC, and the public keys are generated from the private keys by hashing them multiple times. At the client device side i.e. the supplier side, the latencies can be reduced with the hardware implementation of the hashing functions. At the server level, which can have access to parallel computing architectures and graphic processor units, the latencies issues need be mitigated to an acceptable level.

Lattice based PQC DSA (CRYSTAL, qTESLA, and FALCON). Lattice based algorithms exploit the hardness to resolve problems such as the Closest Vector Problem (CVP), and Learning With Error (LWE) algorithms, and share some similarities with the knack-pack cryptographic problem.

- The public-private key pair generation of CRYSTAL is based on polynomial computation in a lattice ring [74]. Matrix **A**, and the two vectors $\mathbf{s_1}$ and $\mathbf{s_2}$ are generated randomly, **t** is computed as $\mathbf{t} = \mathbf{A}\,\mathbf{s_1} + \mathbf{s_2}$; both **A** and **t** become the public key, while $\mathbf{s_1}$ and $\mathbf{s_2}$ become the private key. One method to implement CRYSTAL with the TAPKI protocol is to have the handshake pointing at three addresses in the PUF to generate **A**, $\mathbf{s_1}$ and $\mathbf{s_2}$, then to compute **t**. The DSAs are signed using the private keys and verified using the public keys. An alternate method to implement CRYSTAL is to use the handshake to send **A**, and the addresses to find $\mathbf{s_1}$ and $\mathbf{s_2}$, then compute **t**. This second method is not as secure, but does not have to mitigate potential errors due to the PUFs in the generation of matrix **A**.
- The key pair generation of qTESLA is similar to that used by CRYSTAL, however the signature/verification algorithms are different [76].
- FALCON, which uses NTRU ($\boldsymbol{N^{th}}$ degree of **TRU**ncated polynomial ring) arithmetic [75] is based on methods to generate public-private key pairs that can be implemented with TAPKI schemes. The PUFs can replace random number generators to find private keys; however, the resulting polynomial elements are not always usable, as they are subject to some pre-conditions. The server will need to try several possible TAPKI handshakes, and select the ones giving acceptable private keys. The generation of the public keys from the private keys is based on inverse modulo computation.

Multivariate PQC DSA (GeMSS, LUOV, MQDSS, and Rainbow). The private keys for multivariate-based PQC DSA algorithms are generated with random numbers forming invertible matrix, and polynomials. The TAPKI and ternary PUFs can replace the random numbers generators. The public keys are derived from the private keys, the signature of the DSA uses the private keys, and the verification uses the public keys. These multivariate methods have been known for a long time, and the size of their signature can be small, however it is still unknown if the performance, and size of the keys will be competitive with other methods.

The list of recommended PQC DSA should be reduced by NIST in the next two years, and the final recommendations are expected to be announced in the 2023-2025 window. The TAPKI will be easier to implement with PQC DSA algorithms based on relatively small key pairs; we anticipate that NIST will selects DSA algorithms with small keys.

3.4 Response-Based Cryptography for public key verification

In the scheme described in Fig.4, the secret key **K'** generated by the client device with TAPKI is slightly different from the key **K** generated by the server due to the errors caused by the drift of the physical parameters of the PUF. The RBC is the important scheme needed to validate the public key **PK'** used in the DSA scheme of the blockchain [82-83]. The RBC is a search engine that finds the uncorrected responses of the PUF, i.e. the private key **K'**. As shown in Fig.6, the starting point of the search is the reference key **K** stored in the look up table of the server.

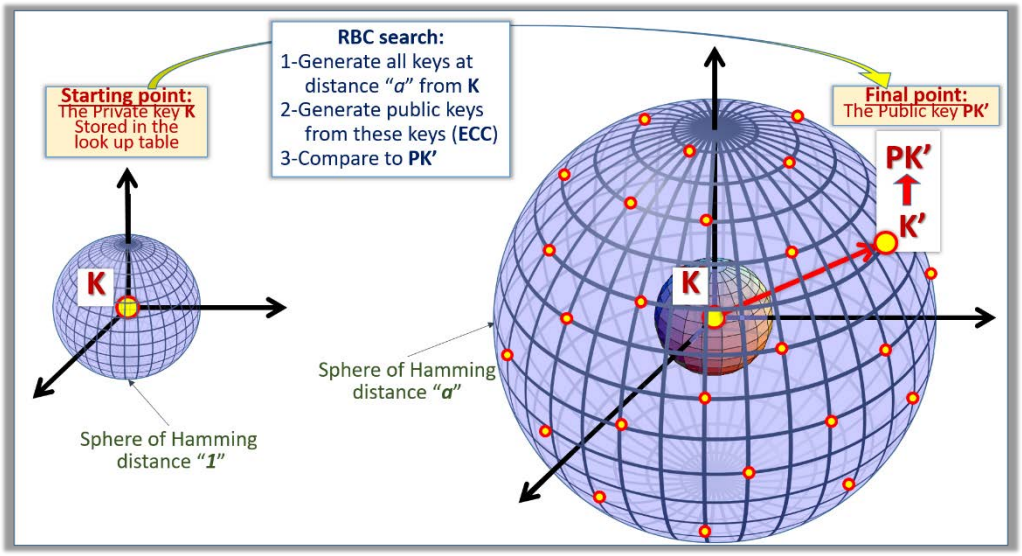

Figure 6: Verification of the validity of the public key K with RBC.

The objective of the search is to find **K'** that is a stream with "***a***" errors, i.e. the Hamming distance between both streams is "a". The search algorithm is an iterative process:

Step 0: A public key **PK** is generated from **K**, and compared with **PK'**, which is known. If equal the search stops;

Step 1: All keys at a Hamming distance of one from **K** are generated with their associated public keys. If one public key matches **PK'** the search stops;

Step ***a***: All keys at a Hamming distance of "***a***" from **K** are generated with their associated public keys. If one public key matches **PK'** the search stops;

Step ***a***+1: When the RBC search is positive, the public key **PK'** is posted in the public ledger as valid.

The RBC method is effective when the error rate is low enough, if not the latencies are prohibitive. For example, with 256-bit keys, the RBC is able to find keys having 3 to 4 errors, which correspond to an error rate of 1.5%. Ternary PUFs characterized in the experimental section have error rates below 0.1%, which is well within the search capabilities of the RBC scheme. Conversely, the typical error rates of the PUFs without ternary states and the blanking of fuzzy states is in the 5 to 10% range. The average latency $\mathbf{A}_{(\lambda,N)}$ of the RBC search for N-bit long PUFs with an average number of erratic bits $\boldsymbol{\lambda}$ is given by:

$$\mathbf{A}_{(\lambda,N)} = \boldsymbol{\tau}_0 \sum_{X=0}^{X=N} \mathbf{P}_{\lambda}(X) \left[\sum_{i=0}^{i=X} \binom{N}{i} - \frac{1}{2}\binom{N}{X}\right] \approx \boldsymbol{\tau}_0 \frac{1}{2}\binom{N}{L} \quad (1)$$

- $\mathbf{P}_{\lambda}(X)$ is the probability to have **X** erratic bits in the **N**-bit long keys, with $\boldsymbol{\lambda}$ erratic bits;

- τ_0 is the average latency to generate a public key from a private key, and to compare it to **PK'**;
- **L** is the integer number greater than λ: **L**-1< $\lambda \leq$ **L** (the approximation correct when λ is large);

The use of PQC DSA schemes with slow**er** key pair generation could results in high latencies with lower efficiencies of the RBC search, which could be a limiting factor of the scheme. In order to be able to use PUFs with higher rate of errors, and PQC DSA with higher latency we recommend implementing a scheme with key fragmentation [84]. The general concept behind this operation is summarized in Fig.7. The keys are fragmented into **k** segments, and padding is used to keep the resulting sub-keys at the same length. In the development described in the experimental section, the error free padding information is shared as part of the handshake. Public keys are generated from the **k** sub-keys to feed the RBC search engine. When **k** is an integer number dividing **N**; **N/k** has to be an integer number as well. The average latency $\mathbf{A}_{k(\lambda,N)}$ of the RBC search with fragmentation by **k** is given by:

$$\mathbf{A}_{k(\lambda,N)} = \mathbf{k} \cdot \mathbf{A}_{(\lambda/k,N/k)} \tag{2}$$

$$\mathbf{A}_{k(\lambda,N)} = \mathbf{k}\,\tau_0 \sum_{X=0}^{X=N} \mathbf{P}_{\lambda/k}(X) \left[\sum_{i=0}^{i=X} \binom{N/k}{i} - \frac{1}{2}\binom{N/k}{X}\right] \approx \mathbf{k}\,\tau_0\,\frac{1}{2}\binom{N/k}{L/k} \tag{3}$$

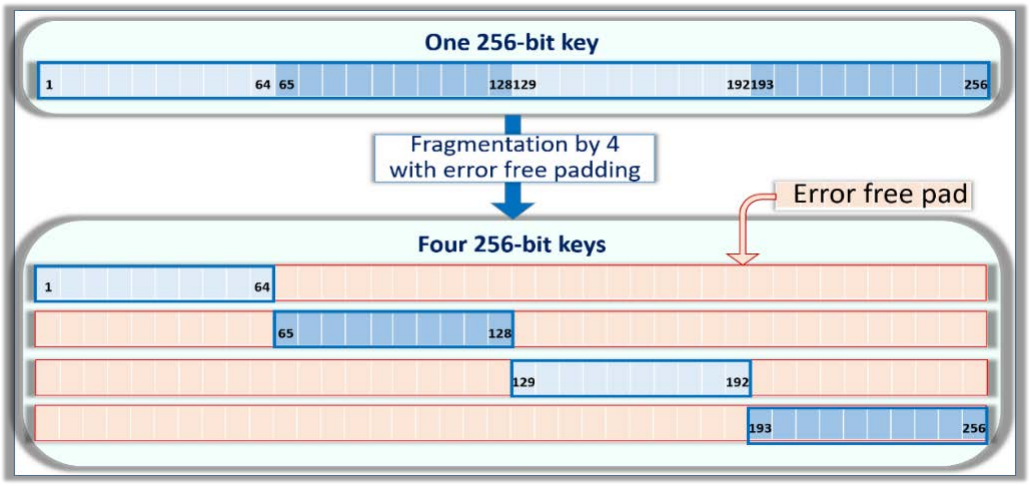

Figure 7: Example of fragmentation by 4 of a 256-bit long key.

- $\mathbf{A}_{(\lambda/k,N/k)}$ is the av. latency of the search with **N/k** bit long keys, and λ/**k** av. erratic bits;
- **L/k** is the integer greater than λ/**k**: (**L/k**)-1< λ/**k**≤**L/k** (approximation correct when λ is large);

With fragmentation, the RBC search latencies are greatly reduced. For example, when **N**=256, λ=16, **k**=4 the ratio between the latencies without and with fragmentation is:

$$\mathbf{A}_{(16,256)} / \mathbf{A}_{4(16,256)} \approx ¼ \binom{256}{16} / \binom{64}{4} = ¼\ (1.0\ 10^{25}) / (6.4\ 10^{5}) = 3.9\ 10^{18} \tag{4}$$

It is desirable to minimize the levels of fragmentation to reduce electronic power at the supplier level. During the experimental work, based on a 200MHz MIPS RISC microcontroller, we measured that one cycle of public key generation with ECC took less than 100 µs; a fragmentation by 8 can be done well within 1 ms. This latency is reduced by two orders of magnitude when the supplier operates with a commercial 4GHz quad core PC, which is mainstream in Smart Manufacturing. A PQC DSA technology that operates with public key generation that is one hundred thousand times slower than ECC will be still acceptable on a PC with latencies around one second.

3.5 Noise injection and HPC

The verification of the validity of public keys by the CA is critical for a network of suppliers involved in Smart Manufacturing, conversely this could become a target for the opponent. The objective of this work is to develop a scheme based on HPC to bring additional security for Smart Manufacturing. The HPC used in this work has an excess of 2,000 effective cores, and the defect density of the ternary PUFs can be adjusted from 0.01% to 10% by changing the masking of the fuzzy cells. As presented above, when the error rates of the PUFs are low enough, approximately lower than 1.0%, the computing power of commercially available PCs is enough for the RBC search to quickly verify a public key. The concept presented is to inject noise in the PUF to generate highly noisy keys, in such a way that only CA's equipped with HPC can be effective in the public key generation, thereby restricting access to opponents with inferior computing power. An example of sequence that was developed based on (3) is shown in table 2. The ternary PUF based on commercially available SRAMs was set up in such a way that the challenge-response-pair error rates averaged 0.05%. This was done by submitting the SRAMs to one hundred repetitive Power-off-on cycles and keeping only the cells awaking as solid "0" or "1" states. About 20% of the SRAM cells were blanked, and the resulting mapping stored in the look up table of the server. The noise is injected in 256-bit long keys by randomly flipping 36 bits, representing an approximate 14% error rate. It was experimentally characterized that, with a fragmentation by 4, the HPC can verify a public key in 1.2 seconds. The estimated latency of a commercial PC is approximately 1.4 days. Thereby, if the maximum acceptable time to verify a public key by the CA is set around 5.0 seconds, only powerful HPCs are needed to reduce FRR.

Table 2: Example of sequence to protect a network with HPC

1- Ternary PUF	Error rates ≈ 0.05%
2- Noise injection in the PUF	14 % in 256-bit long keys
3- Use fragmentation by 4	
4- Use HPC for RBC search	2,000 cores
5- Average RBC latency with HPC	1.2 seconds
6- Average RBC latency with PC	1.4 days
7- Maximum acceptable latency:	5.0 seconds

Importance of the elimination of the fuzzy cells. The use of ternary PUFs that mask fuzzy cells, enhance the stability of the scheme. With ternary PUF error rates in the 0.05% range, and subject to a normal distribution, the natural variations is such that the probability to have three bad bits or more on 256-bit long key is $3.18\ 10^{-4}$, which makes the minimization of the False Rejection Rates (FRR) of the search with HPC relatively easy. Conversely, a PUF having 4% error rates will face the natural variations shown in Fig.8, from 2 to 20 errors, which makes the protocol with HPC described in Table 2, hard to implement.

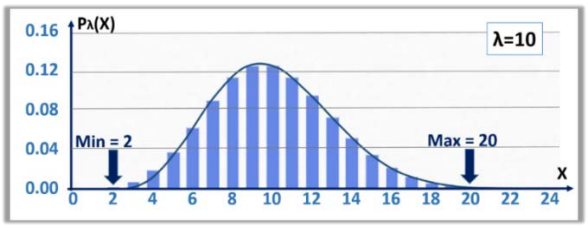

Figure 8: Normal distribution of the number of errors X for a 256-bit long key, and λ=10.

Assuming that the noise injector adds 10% bad bits of a 256-bit long key, the HPC will not be able to find the erratic keys when the errors due to the PUFs are on the high end of the distribution, thereby resulting in FRR. When the errors are at the low end of the normal distribution, a regular PC is anticipated to be able to find the erratic keys, which defeats the purpose of the scheme. In summary, the injection of noise to discriminate HPCs versus PCs is only effective when the PUFs have low error rates.

Fragmentation to widen the window of operation. As presented in section 2.4, key fragmentation allows the use of PUFs with higher error rates. This fragmentation method can also widen the window of operation of the schemes, based on noise injection, and HPC, see Fig.9. Without fragmentation, an injection of approximately 1.5% bad bits into 256-bit long keys, differentiates the use of HPCs from regular PCs, the PCs are not powerful enough. However, the addition of few bad bits would increase FRR to non-acceptable levels, even with HPC-based search. With key fragmentation by 4, the injection of 7% to 15% bad bits into 256-bit long differentiate well the use of HPCs from PCs. This represents a wide window of operation in which the FRR of HPCs could be set very low. The sequence proposed in table 2 is set at the high end of the window, i.e. 14%, to prevent the effectiveness of more powerful PCs.

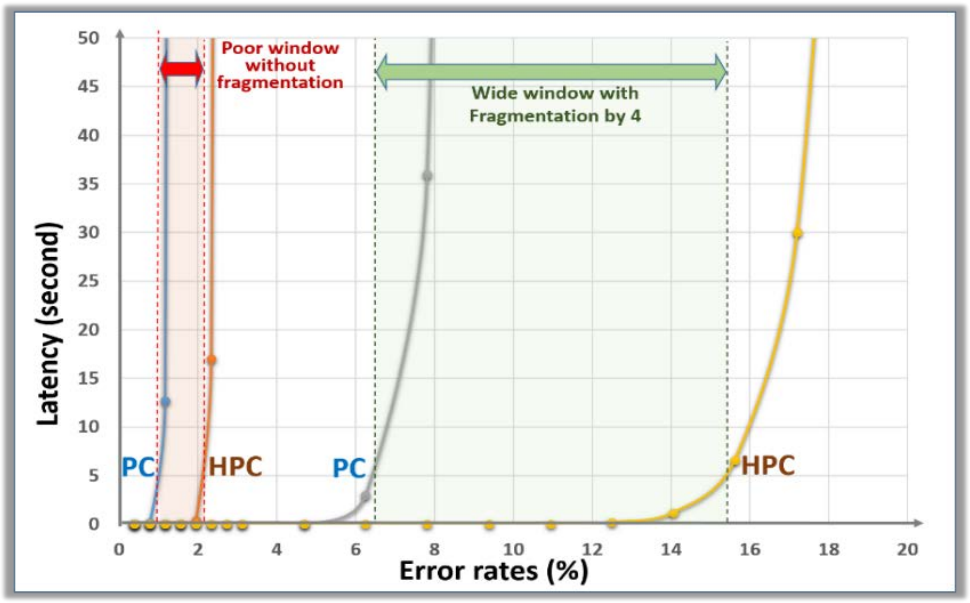

Figure 9: Modelling the latencies of RBC searches for 256-bit long keys.

4.0 RESULTS AND DISCUSSION

4.1 Experimental validation

Prototype. Commercially available components, such as SRAMs, SHA-512, and ECC, have been selected to validate the protocol securing Smart Manufacturing with blockchains. The ternary PUFs were designed with the SRAM, and the private key generation used TAPKI schemes. One of the challenges of this development was the public key matching algorithm with RBC, which allows the server to independently recognize the public keys generated by the ternary PUFs of each client device. On the client device, the objective was to implement ECC key exchange and DSA protocol as part of TAPKI in a microcontroller environment with relatively low computing power. On the server side, the objective was to implement ECC key exchange as part of the RBC search algorithm, executable on both PCs and the HPC. One of the complexities of the overall project was to develop a software stack working in such heterogeneous computing environment, from low end microcontrollers, window-based PCs, and HPCs. A representation of the prototype that was designed for this work is summarized in Fig.10. The two client devices are driven by the microcontrollers WiFire fabricated by Digilent. The custom daughter cards handle the SRAM PUFs, and the wireless connectivity. The tablets are used to enter messages, and display the message digests, digital signatures, and public keys. The protocol is the following:

- Step-1: Alice enters the message in the tablet in plain text;
- Step-2: Generation of the private keys with TAPKI, and the handshake between the CA (i.e. the PC) and the microcontroller board;
- Step-3: The microcontroller hashes the message, signs it with the private key and generates a public key with ECC. The resulting information is displayed on the screen of the tablet;
- Step-4: The same information is transmitted to Bob's microcontroller board;
- Step-5: Bob's microcontroller board verifies with the CA that the public key is valid, verifies that the signature is valid, and displays the information to the screen of the tablet. The RBC search is performed on the PC, with fragmentations by four, to validate the public key.

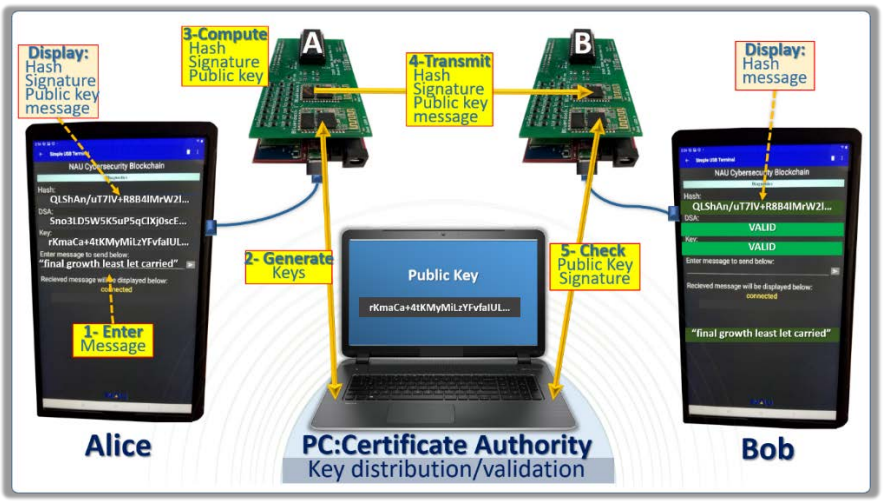

Figure 10: Overall representation of the prototype developed to demonstrate the architecture.

Client devices. To interface commercially available SRAM with the ChipKit WiFire microcontrollers, a custom daughter-card or shield is needed. This is a custom PCB that allows additional

hardware components to be placed on top of the ChipKit microcontroller. Before using the shields, breadboards and jumper wires were used to connect to the SRAM. The breadboard setup worked slowly and had room for many errors. One of the biggest issues faced, was figuring out how to power down the SRAM without completely powering down the entire microcontroller and project setup. To avoid this issue and create a smaller, more compact hardware package, the design of a custom PCB was implemented. With the shield design being the next step for the project, research was done on which components we can use to manage the SRAM's power, inputs, and outputs. Along with SRAM power management, a way to incorporate wireless peer-to-peer communication needed to be included in the hardware. After much prototyping, the components that were decided upon were 26 analog switches for SRAM power, I/O management, and two HC-06 Bluetooth modules.

For most of the prototyping phase desktop workstations or laptops were used to interface with the microcontrollers. The interfacing entailed a simple way to read out diagnostics, message data, and verifications through a computer terminal. Moving forward using a desktop or even multiple laptops for a demonstration is not ideal, so to make the demonstration more appealing and portable we moved to android tablets. We chose android tablets because we could easily implement the information read out through a simple app developed through android studio along with using open source apps to assist in data management. The Samsung 10.1" Tab A tablet was chosen for this. They meet the power standards for providing power to the microcontroller and shield components, along with being able to handle serial communication for interfacing. Fig.11 is showing the app layout that displays required diagnostics, verification checks and messages.

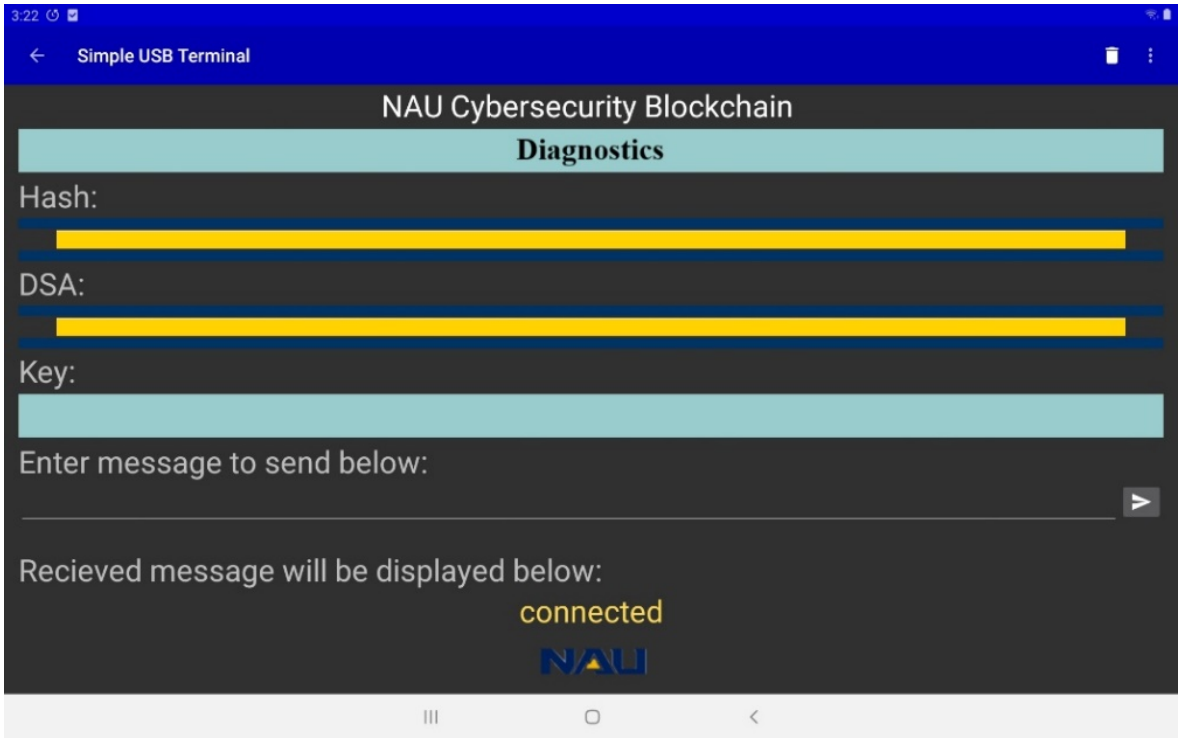

Figure 11: App screenshot of the tablet to display information.

An example of sequence demonstrated in the prototype is shown below:

Step-1: Message randomly generated by Alice:
'final growth least let carried'
(0x66696e616c2067726f777468206c65617374206c65742063617272696564)
Step-2: Key generated by the TAPKI, and Alice's WireFire Chipkit:
a) Random number exchanged during the handshake:
39b5b15badd904619ea98424a5545e49bd725ffa9d959bcd604d3232a2f471945a69
6994ce98a2568b49dfeec698cb001daff100c629fc46090456a292c4b1e7
b) Private key:
e2e4e4dbf34cba3177425cd7df5d21b20ae0c2660316cd396f0608e5e7a1fc7b296893
dbbb3
a369a9839a64063aee6606dfcbedf496a4bdfbdee123cd2a0472
c) Public key generated with ECC:
rKmaCa+4tKMyMiLzYFvfaIULIdemLQBBvPocyMi6iaxRV7NNbyvyR9Wb2sbT-
aBoG5ayHIS1sQHFYvtPn1s1gHA==
Step-3: Computing by Alice's WiFire Chipkit
a) Hashing of the message with SHA512:
QLShAn/uT7lV+R8B4lMrW2XClETs8/tlzxaPmDAs1hiv1dYSOhxs7JduzU-
MuZzrZpUWBHhjKuW0Gx7skfEAe7g==
b) Digital signature of the message digest with the private key and ECC:
Sno3LD5W5K5uP5qClXj0scEuCH+6bFyCqsT4MQbcwQ4tZF08raCHHMJ51pd
vecBTmTns7ZqGz9/DNsGGupSsgg==
Step-4: Information transmited to Bob's WiFire Chipkit: message, message digest, signature, and
public key. This information is posted on the screen of both tablets;
Step-5: Verification by Bob's WiFire Chipkit:
a) The PC verifies the validity of the public key with RBC;
b) Bob's chipkit hashes Alice's message with SHA-5125 to check the message digest;
c) Bob's chipkit verifies the validity of the signature with the public key and ECC.

Characterization of the performance. After the manual entry of plain texts in the first tablet having variable lengths, the latencies of the entire protocol lasted less than one second: the second tablet displays the plain text, its message digest, and the validation of the DSA within 500ms. The generation of the 256-bit long public keys from the private keys with ECC takes 10,000 clock cycles; the generation of the 256-bit long private keys from the PUF takes 800 clock cycles. Other PKI protocols like RSA are much slower: the key pair generation with RSA takes 500,000 clock cycles for 1500-bit long keys, which have the same cryptographic strength as the 256-bit long keys for ECC. In both cases, ECC or RSA, the latency of the generation of the private keys from the PUFs is negligible. The CRP error rates of the SRAM-based PUFs with the masking of the unstable cells used in this design were characterized in the sub 10^{-4} range. No false reject of the RBC search were observed over thousands of cycles, and several months of repetitive testing. We also tested the protocol with SRAM PUFs without masking unstable cells, showing CRP error rates in the 5% range. With fragmentation by 8, we were able to get similar results: latencies around 500ms, and no observable false rejects. To the best of our knowledge, no other protocols have been published that generate one-time use public-private keys pairs from PUFs, to secure the DSA of blockchain technology with such latencies, and no observable FRR of the keys.

4.2 Analysis of the levels of security

In this architecture, the tablets, the communication between the tablets and the WiFire Chipkits, the wireless communication between the two Chipkits, and the communication Chipkits to PC are assumed to be vulnerable, and non-secure. The purpose of the tablets is to display non-secure publicly available information: messages, message digests, digital signatures, and public keys. This publicly available information is freely transmitted tablet to Chipkit, and Chipkit to Chipkit. The TAPKI handshake PC to ChipKit is also publicly available information, which is protected by multifactor authentications of the Chipkit such as passwords, pin codes, biometric prints, and PUF challenges. The most vulnerable link of the architecture is the Chipkit, and the daughterboard with the SRAM PUF. Examples of vulnerabilities include:

- Loss of the Chipkits to the opponents, who will directly attack the SRAM PUFs, read the mapping of the responses, and generate a look-up table, similar to the one stored by the CA, or a clone of the client device to fool the CA;
- Side channel analysis to extract the private keys during generation from the PUF, or during the public key generation from the private keys, and during the digital signature cycles also from the private keys. Examples of analysis include differential power analysis (DPA), fault injections, and the use of sensing elements of the electromagnetic interferences generated by the Chipkit;
- Generic software attacks between the client device and the CA such as man-in-the-middle with fake client devices to generate malicious blockchains from un-authorized users. The users could then interface with fake CA, pretending to be legitimate;
- Neutralization of certain client devices with malware injection, Denied of Service (DoS) attacks, confusion of the PUF with thermal, or EMI attacks;
- Attacks directed at the CA to steal the look up tables of the PUFs and develop fake CAs handling the constellation of client devices.

The design of the WiFire Chipkit uses generic components, which are by definition non-secure. The implementation of this proposed scheme will require a set of improvements such as the following:

- Replacement of the SRAM by tamper resistant components. When lost to the opponent, the responses of the SRAM PUFs are relatively easy to extract. Advanced memory devices such as Resistive RAM, and Magnetic RAM can be used to design lower power PUFs, which are more difficult to break [41-44];
- Use encryption and protection schemes to generate PUF responses that prevent wide leakages of the content of the PUF;
- Design of a custom secure microcontroller chip integrating the PUF, the crypto-processor, and RISC processor, with hardware implementation of the cryptographic protocols. Commercial SIM and banking cards are currently leveraging powerful secure microcontroller chips with wireless connectivity that could replace the WiFire Chip set, and interact directly with a tablet, or other terminal device. Commercial secure microcontrollers are equipped with counter measures against side channel analysis, DPA, and physical attacks;
- Implement multi-factor authentication of the CA to mitigate man-in-the-middle attacks, and the entry of malicious CA's. The look up table of the CA, which the store the PUF challenges, and the initial responses can provide one of these factors.

In a smart manufacturing environment, the entities managing their suppliers usually have a stringent process to qualify their suppliers. The delivery of a secure microcontroller with a PUF for each is rather simple from a logistical standpoint. Before delivery, the managing entities will capture the image of the PUF and store it in a look up table of their server, which can be handled in a highly secure environment. The responsibility of the managing entities will be to implement a process to protect their servers from the opponent, and to act as a CA for the constellation of supplier. The manufacturing of strategic assets usually have access to powerful servers, and HPC, such as the one analyzed below in section 5.0.

4.3 Impact of High-Performance Computing for RBC

Description of the schemes driving the HPC. We implement the response-based cryptography protocol described in Sections 2.4 and 2.5. Our implementation is written in C and is parallelized using MPI [MPI] and Pthreads [Pthreads]. Let **KS(a, k)** be the total key space, containing all of the **N=**256-bit keys that need to be searched using the starting key **K** (known by the server), with a hamming distance of **a,** using fragmentation **k**. Since *on average,* a key will be found half-way through the search at hamming distance **a,** the total number of keys searched, with fragmentation **k**, is as follows:

$$|\mathbf{KS}(\mathbf{a},\mathbf{k})| = \sum_{\mathbf{i=0}}^{\mathbf{a-1}} \binom{\mathbf{256/k}}{\mathbf{i}} + \frac{\mathbf{1}}{\mathbf{2}}\binom{\mathbf{256/k}}{\mathbf{a}} \quad (5)$$

Given the total key space, we assign $|\mathbf{KS}(\mathbf{a},\mathbf{k})|/\mathbf{p}$ keys to search to each MPI process rank, where there are **p** physical cores on our platform**.** Without the loss of generality, we assume **p** evenly divides $|\mathbf{KS}(\mathbf{a},\mathbf{k})|$**.** When one rank finds the correct key, **PK',** the search needs to terminate**.** There are several methods that could be employed to terminate the search; however, some methods lead to unacceptable overhead. We briefly describe our search termination procedure as follows. Each MPI rank creates two threads (implemented using Pthreads). One thread performs the search for the correct key, while the other thread performs communication between ranks. If a rank finds the correct key, then this information is sent to all other process ranks, and their respective communication threads terminate the search at each rank. To ensure that each communication thread consumes few computational resources, which would otherwise be used by the search thread, we use the Iprobe functionality in MPI that performs a non-blocking check for the message that indicates that the search needs to be terminated. We adjust a parameter that determines how often we check for this message, to reach a trade-off between message checking overhead and the number of wasted searches, where wasted searches refer to those searches that are performed after the key has been found.

Statistical analysis with HPC. All code is written in C. In all of our experiments, we use 256-bit keys, and average our response times over 10 trials. All code is compiled using the -O3 compiler optimization flag and is compiled using the GNU compiler v.6.2.0. As described in Section 2.4, a PUF will have an error rate that follows a distribution. In our experiments, we select a single Hamming distance that does not vary as a function of distribution (e.g., the distribution in Figure 8). Since the algorithm response time will be impacted based on when the key is found within the search space, we elect to fix the key to be found in the middle of the key space at hamming distance **a**, such that we achieve the *average case* response time. This average case is outlined in Sections 2.4. and 2.5. All experiments are carried out on the Monsoon cluster at NAU. In our experiments,

we use two dedicated compute nodes. Each node has 2x 2.6 GHz Intel Xeon Gold 6132 processors with 2x14=28 physical cores. All experiments were performed on 64 physical cores across three nodes. Regarding the experimental results, we note the following caveat: our implementation may contain remaining errors in the timing procedure. Therefore, the response times reported in this section may not be accurate in absolute term. The experiments in this section are thereby preliminary and may change in future implementations. We will be conducting a detailed performance evaluation of RBC for ECC as part of future research work. Despite this, we find that the reported measurements are in general agreement with the expected latencies derived by the model. Fig. 12 plots the measured response time vs. Hamming distance for (a) **k**=1, no fragmentation; (b) **k**=2; (c) **k**=4; and (d) **k**=8.

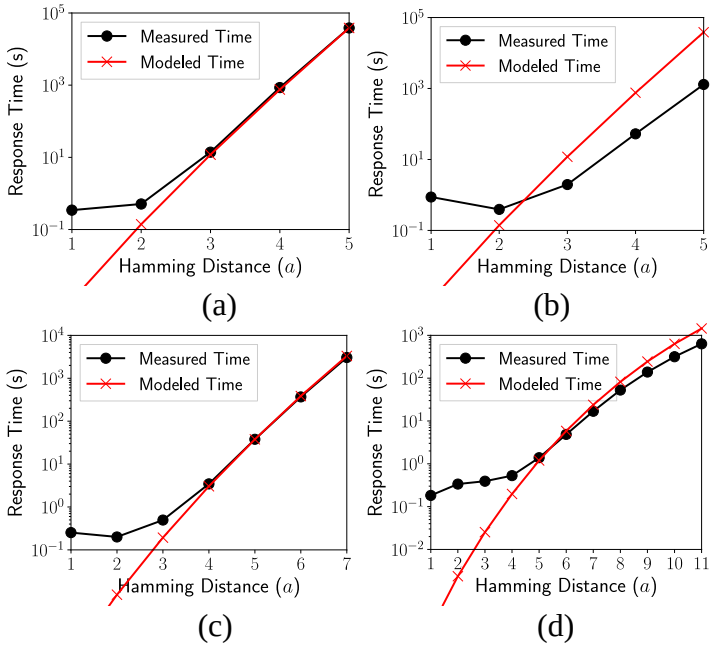

Figure 12: Measured and modeled latencies for (a) k=1, (b) k=2, (c) k=4; (d) k=8.

In all experiments, we use **p**=64 physical cores. Since increasing the Hamming distance exponentially increases the search space, we plot the response time on a log scale. In Figure 9(a) with **k**=0 (no fragmentation), we find that at Hamming distance **a**=3-5, and using **p**=64 cores, we cannot find the key in a reasonable amount of time, where only **a**<3 is practical for the search. For example, at **a**=5, the key is found in 38570 seconds. At the other extreme, Figure 9(d) plots **k**=8, where the key can be found within 1.39 seconds at **a**=5. This shows that the use of fragmentation increases the range of practical hamming distances, **a.** Consequently, when implementing RBC in practice, the values of **k** and **a** can be carefully selected based on **p** to achieve the desired key authentication throughput.

Although we limited **p**=64 in this evaluation, our implementation is expected to achieve good scalability on larger core counts. We model the response time of the search to determine whether the expected performance is impacted by any of the search parameters. We first measure the constant $\boldsymbol{\tau_0}$, which is the time to perform one ECC calculation. Since $\boldsymbol{\tau_0}$ is implementation-dependent, it must be experimentally derived. We find that $\boldsymbol{\tau_0}$= 8.417x10^{-6} seconds on our platform using **p**=64 cores. Using the number of keys, $|\mathbf{KS}(\mathbf{a}, \mathbf{k})|$, as a function of the Hamming distance, **a**, and

fragmentation **k**, and the value of τ_0, our model is simply $\tau_0|\mathbf{KS}(\mathbf{a}, \mathbf{k})|$. Fig. 12 compares the measured and modeled algorithm response time. On the smaller workloads (low Hamming distance) we find that the model underestimates the total response time. This is because there are overheads associated with the implementation that are amortized on the larger workloads but are not amortized on the smaller workloads. Overall, we find that our model is able to capture the performance behavior of the search.

4.4 Discussion: use of the method for smart manufacturing

Additive Manufacturing (AM) creates an object by adding layers of material from three-dimensional data. By comparison, traditional, or subtractive, manufacturing processes is where the product is created by cutting away material from a larger piece [83]. Due to the numerous technical and economic advantages that this technology promises, AM is expected to become a dominant manufacturing technology in both industrial and home settings. The National Defense Authorization Act for FY 2017 Senate Report "strongly encouraged" the DoD to more aggressively pursue AM capabilities to improve readiness and enable the Military Services to be more self-sustainable. The Office of the Deputy Assistant Secretary of Defense for Manufacturing and Industrial Base Policy is the DoD AM lead that oversees the implementation of AM and reports to the Under Secretary of Defense for Research and Engineering. [85] The growing penetration of AM at manufacturers across the world and the dependence of this technology on computerization have already raised security concerns, some of which have been proven experimentally [85]. The parts themselves have now become new targets for cyber criminals. More specifically, the parts "digital twin", the digital file that contains the parts' specs and manufacturing instructions now becomes a vulnerability. This is due to the dependency of the effectiveness of AM almost entirely on the integrity of digital files to instruct the 3D printing mechanism [86]. To ensure the integrity and traceability of digital files and assure their secure delivery at each stage in the supply chain, ranging from the file developer all the way to the end user, more companies are turning to blockchain. Blockchain functions like a distributed database that maintains a continuously growing list of ordered records. Blockchain works by storing information, in this case design files, across each phase of the digital supply chain. The phases would include design, distribution, manufacturing, and in-field on any participating nodes. If an additive manufacturing supply chains implemented blockchain at these transactional node levels, it has the potential to assure that all assets were traceable, and their provenance known. Users would have the capability to see and trace the full lifecycle of the part.

Having a secure blockchain architecture becomes a cornerstone towards securing AM capabilities. The current weakness of blockchain technology is the private keys. When stored in the non-volatile memory, or when they are too weak, this become a vulnerability. A paper presented by Independent Security Evaluators (ISE) discovered that funds from weak-key addresses are being pilfered and sent to a destination address belonging to an individual or group that is running active campaigns to compromise/gather private keys and obtain these funds. On January 13, 2018, this "blockchainbandit" held a balance of 37,926 ETH valued at $54,343,407. [87] The work presented in this paper demonstrates a solution based on PUFs embedded in the hardware of each supplier node as an effective mitigation to the private key weakness of blockchain technology. This will increase the reliability and resilience of the AM process.

5.0 CONCLUSIONS

The authors recognize that one of the most impressive aspect of the technology behind Bitcoin, the elimination of a central authority in favor of peer-to-peer trust mechanism, is not included in the proposed architecture. We argue that the smart manufacturing of strategic assets with networks of suppliers can benefit from certificate authorities restricting the list of suppliers, monitoring public key infrastructures, and the validity of the digital signatures. Such a restrictive environment can still benefit from non-alterable, non-repudiable ledgers, resulting from hash functions, and digital signatures. The prototype developed in this research work demonstrates that commercially available SRAM-based PUFs with ternary cryptographic schemes can generate exceptionally reliable one-time use public-private key pairs for the digital signature of each blockchain. We experimentally verified that the latencies to generate keys, hash the messages and sign them are in the 500ms range; the false reject rates, due to erratic ternary PUF responses, are extremely low. The commercially available WiFire chipkits with custom daughter-cards are relatively low power, and it is expected that they can be replaced by custom secure integrated circuits with complexity similar to mainstream SIM cards.

Adding HPCs, and noise injection to the private key generation is going one step further in the direction of establishing strong CAs that monitor the key distribution to known suppliers. The very preliminary data generated experimentally by our HPC seems to validate the models proposed to optimize RBC search latencies. For example, with masked SRAM-based PUFs having low error rates, the injection of about 14% bad bits into 256-bit long keys, and RBC search using fragmentation by four, our HPC can verify the validity of public keys within seconds, while regular PCs are not powerful enough to perform such verification. The scheme is anticipated to increase the cost to break the supplier based smart manufacturing environment using blockchain technology.

The future work envisioned by the authors includes:

- ***Replacement of the SRAM-based PUF by tamper resistant components.*** Two memory technologies are considered, Resistive RAM and Magnetic RAM. The architecture suggested in this paper is agnostic on the type of PUF selected, as long as the defect density is low enough. It has to be noticed that the masking methodology proposed is effective to reduce the defect density of the SRAM PUFs from 5% to 10^{-5}. The effort needed to get similar results with the ReRAMs and the MRAMs is not under-estimated.
- ***Replacement of the Elliptic Curve Digital Signature by quantum resistant DSA***. Both hash and lattice based cryptographic schemes that are currently under consideration by the NIST driven PQC program are excellent candidates. We intend to take an early look at SHINCS, CRYSTAL, and qTESLA, which seems to be compatible with PUF-based private key generation. The main figure of merits of the novel PQC DSAs that will be characterized are the latencies to generate public keys from the private keys, and false public key generation. The RBC search includes large quantities of public key generation, therefore excessive latencies will be prohibitive. We will investigate if HPC/GPU technology can reduce these latencies. The second important figure of merit is the size of the private keys. Long keys will be overly sensitive to errors in the PUF responses,
- ***Optimization of the HPC/GPU.*** The work presented in this paper is preliminary and will require a lengthy investigation. Several parameters of the RBC search can be optimized to enhance the efficiency of HPC's and GPU's, namely the levels of fragmentation, the type of noise injection, the use of DSA algorithms, and the ways to concurrently feed parallel computing units;

- ***Enhancing the levels of security of the architecture.*** We presented in section 3.2, we see the need to implement some remedies to mitigate the potential vulnerabilities. We also intend to involve third-party to highlight additional potential weaknesses.

In conclusion, the proposed architecture, which uses distributed PUFs and ternary cryptographic schemes, has the potential to enhance security of the blockchain technology when applied to the logistic of smart manufacturing. The prototype developed is encouraging; however, the implementation will require significant additional resources, and third-party assessment.

6.0 REFERENCES

1. Nakamoto, S. Bitcoin: a Peer to Peer Electronic Cash System. www.biticoin.org; 2008;
2. Croman, K.; Decker, C.; Eyal, I.; Gencer, A.E.; Juels, A.; Kosba, A.; Miller, A. On scaling decentralized blockchains. Int. Conf. on Financial Cryptography and Data Security, 2016;
3. Luu, L.; Narayanan, V.; Zheng, C.; Baweja, K.; Gilbert S.; Saxena, P. A secure sharing protocol for open blockchains. ACM SIGSAC Conf. on Comp. and Comm. Security, 2016;
4. Eyal, I., Gencer, A. E., Sirer, E. G., Renesse, R. V. Bitcoin-NG: A Scalable Blockchain Protocol. NSDI, 2016;
5. Dorri, A., Kanhere, S. S., Jurdak, R. Blockchain in internet of things: challenges and solutions. arXiv preprint arXiv: 1608.05187, 2016;
6. Gervais, A., Karame, G. O., Wüst, K., Glykantzis, V., Ritzdorf H., Capkun, S. On the security and performance of proof of work blockchains. In ACM SIGSAC, 2016;
7. Zheng, Z., Xie, S., Dai, H.-N., Wang, H. Blockchain challenges and opportunities: A survey. International Journal of Web and Grid Services, pp. 1-25, 2016;
8. Dua, S.; Du, X. Data Mining and Machine Learning in Cybersecurity; CRC Press of Taylor & Francis Group: Boca Raton, FL, USA, 2016;
9. Buczak, A.; Guven, E. A Survey of Data Mining and Machine Learning Methods for Cybersecurity Intrusion Detection. IEEE Commun. Surv. Tutor. 2016, 18, 1153–1176;
10. Paar, C.; Pezl, J. Understanding Cryptography. Springer: New York, NY, USA, 2011;
11. Pfleeger, C.P.; Pfleeger, S.L.; Margulies, J. Security in Computing. 5th ed.; Prentice Hall: Upper Saddle River NJ, USA, 2015;
12. Rivest, R.; Shamir, A.; Adleman, L. A method for obtaining digital signatures and public-key cryptosystems; Commun. ACM 1978, 21, 120–126;
13. Shor, P. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. J. Soc. Ind. Appl. Math. 1999, 41, 303–332.
14. Alagic, G.; Alperin-Sheriff, J.; Apon, D.; Cooper, D.; Dang, Q.; Liu, Y-K; Miller, C.; Moody, D.; Peralta, R.; Perlner, R.; Robinson, A.; Smith-Tone, D. Status Report on the First Round of the NIST Post-Quantum Cryptography Standardization Process. https://doi.org/10.6028/NIST.IR.8240, 2019;
15. Tera, H. Introduction to Post-Quantum Cryptography in scope of NIST's Post-Quantum Competition. Thesis University of Tartu, Institute of Computer Science, Computer Science Curriculum, 2019;
16. Nejatollahi, H.; Dutt, N.; Ray, S.; Regazzoni, F.; Banerjee, I.; Cammarota, R. Post-quantum Lattice-based Cryptography Implementations: A Survey. ACM, https://doi.org/10.1145/3292548, 2018;
17. Chen, S.; Shi, R.; Ren, Z.; Yan, J.; Shi, Y.; Zhang, J. A Blockchain-based Supply Chain Quality Management Framework. 14th IEEE Int. Conf. on e-Business Engineering, 2017;
18. Banerjee, A. Blockchain Technology: Supply Chain Insights from ERP; Advances in Computers, ISSN 0065-2458, 2018;
19. Zhang, Y.; Xu, X.; Liu, A.; Lu, Q.; Xu, L.; Tao, F. Blockchain-Based Trust Mechanism for IoT-Based Smart Manufacturing System; IEEE Trans. on Computational Social Systems, 2019;
20. Bahga, A.; Madisetti, V. Blockchain Platform for Industrial Internet of Things. Journal of Software Engineering and Applications, 9, 533-546, 2016;
21. Francisco, K.; Swanson, D. The Supply Chain Has No Clothes: Technology Adoption of Blockchain for Supply Chain Transparency. MDPI, Logistics, 2018;

22. Abeyratne, S.; Monfared, R. Blockchain Ready Manufacturing Supply Chain Using Distributed Ledger. Int. Journal in Engineering and Technology, IJRET, Vol5, 2016;
23. Saberi, S.; Kouhizadeh, M.; Sarkis, J.; Shen, L. Blockchain technology and its relationships to sustainable supply chain management. Int. Journal of Production Research, 2018;
24. Tijan, E.; Aksentijevic, S.; Ivanic, K.; Jardas, M. Blockchain Technology Implementation in Logistics. MDPI, Sustainability, 2019;
25. Holland, M.; Stjepandic, J.; Nigischer, C. Intellectual Property Protection of 3D Print Supply Chain with Blockchain Technology. IEEE Int. Conf. on Engineering Technology and Innovation, 2018;
26. Holland, M.; Nigischer, C.; Stjepandic, J. Copyright Protection in Additive Manufacturing with Blockchain Approach. Open Access by IOS Press, doi:10.3233/978-1-61499-779-5-914, 2017;
27. Westerkamp, M.; Victor, F.; Ktipper ,A. Blockchain-based Supply Chain Traceability:Token Recipes model Manufacturing Processes. arXiv:1810.09843v1[cs.CY], 2018;
28. Guin U.; DiMase D. Counterfeit Integrated Circuits: Detection Avoidance, and the Challenges Ahead. J. Electron Test, DOI 10.1007/s10836-013-5430-8, 2013.
29. Pappu, R.; Recht, B.; Taylor J.; Gershenfield, N. Physical one-way functions; Science. Vol 297 No5589 pp2026-2030; 20 Sept 2002;
30. Gassend, B.; Clarke, D.E.; van Dijk, M.; Devadas, S. Silicon Physical Random Functions. In: Atluri, V. (ed.) Proceedings of the 9th ACM Conf. on Computer and Comm. Security, CCS 2002, pp. 148–160. 2002;
31. Delavar, M.; Mirzakuchaki, S.; Ameri, M.H.; Mohajeri, J. PUF based solution for secure communication in advanced metering infrastructure. ACR publication, 2014;
32. Herder, C.; Yu, M-D.; Koushanfar, F.; Devadas, S. Physical Unclonable Functions and Applications: A Tutorial. Proceedings of the IEEE, vol. 102, no. 8, pp. 1126-1141, 2014;
33. Maes, R.; Verbauwhede, I. Physically unclonable functions: a study on the state of the art and future research directions. In Towards Hardware-Intrinsic Security, 2010;
34. Jin, Y. Introduction to hardware security; Electronics 2015, 4, 763-784; doi:10.3390/electronics4040763;
35. Flikkema, P.G.; Cambou, B. Adapting Processor Architectures for the Periphery of the IoT Nervous System. In Proceeding of the IEEE WF-IoT, Reston, VA, USA, 12–14 December 2016;
36. Cambou, B. Enhancing Secure Elements—Technology and Architecture. In Foundations of Hardware IP Protection. Springer Int. Publishing: New York, NY, USA, 2017;
37. Gao, Y.; Ranasinghe, D.C.; Al-Sarawi, S.F.; Kavehei, O.; Abbott, D. Emerging Physical Unclonable Functions with nanotechnologies. IEEE, DOI:10.1109/ACCESS.2015.2503432;
38. D. E. Holcomb, W. P. Burleson, K. Fu; Power-up SRAM state as an Identifying Fingerprint and Source of TRN; IEEE Trans. on Comp., vol 57, No 11; Nov 2008;
39. Christensen, T. A.; Sheets, J. E. Implementing PUF utilizing EDRAM memory cell capacitance variation. Patent No.: US 8,300,450 B2; Oct. 30, 2012;
40. Prabhu, P.; Akel, A.; Grupp, L. M.; Yu, W-K S.; Suh, G. E.; Kan, E.; Swanson, S. Extracting Device Fingerprints from Flash Memory by Exploiting Physical Variations. 4th Int. Conf. on Trust and Trustworthy Computing; June 2011;
41. JPlusquellic, J.; Swarup, B. Systems and methods for generating PUF's from non-volatile cells. US patent 10,216,965; 2019;

42. Chen, A. Comprehensive Assessment of RRAM-based PUF for Hardware Security Applications. 978-1-4673-9894-7/15/IEDM IEEE; 2015;
43. Cambou, B.; Afghah, F.; Sonderegger, D.; Taggart, J.; Barnaby, H.; Kozicki, M. Ag conductive bridge RAMs for PUFs. IEEE, HOST, 2017;
44. Korenda, A.; Afghah, F; Cambou, B. A Secret Key Generation Scheme for Internet of Things using Ternary-States ReRAM-based Physical Unclonable Functions. IWCMC 2018;
45. Vatajelu, E. I.; Di Natale, G.; Barbareschi, M.; Torres, L.; Indaco, M.; Prinetto, P. STT-MRAM-Based PUF Architecture exploiting MTJ Fabrication-Induced Variability, ACM trans.; July 2015;
46. Becker, G. T.; Wild A.; Güneysu, T. Security analysis of index-based syndrome coding for PUF-based key generation. IEEE, HOST, 2015;
47. Rahman, M. T.; Rahman, F.; Forte, D.; Tehranipoor, M. An Aging-Resistant RO-PUF for Reliable Key Generation. IEEE Trans. on Emerging Topics in Computing, vol. 4, no. 3, 2016;
48. Chen, T. I. B.; Willems, F. M.; Maes, R.; Sluis, E. v. d.; Selimis, G. A Robust SRAM-PUF Key Generation Scheme Based on Polar Codes. In arXiv:1701.07320 [cs.IT], 2017;
49. Maes, R.; Tuyls, P.; Verbauwhede, I. A Soft Decision Helper Data Algorithm for SRAM PUFs. In 2009 IEEE International Symposium on Information Theory, 2009;
50. Delvaux, J.; Gu, D.; Schellekens, D.; Verbauwhede, I. Helper Data Algorithms for PUF-Based Key Generation: Overview and Analysis. IEEE, CAD-ICS, 2015;
51. Taniguchi, M.; Shiozaki, M.; Kubo, H.; Fujino, T. A stable key generation from PUF responses with a Fuzzy Extractor for cryptographic authentications. IEEE,GCCE, 2013;
52. Kang, H.; Hori, Y.; Katashita, T.; Hagiwara, M.; Iwamura, K. Cryptographie key generation from PUF data using efficient fuzzy extractors. Int. Conf. on ACT, 2014;
53. Cambou, B.; Orlowski, M. Design of PUFs with ReRAM and ternary states. Proceedings of the Cyber and Information Security Research Conference, Oak Ridge, TN, USA, April 2016;
54. Cambou, B.; Flikkema, P.; Palmer, J.; Telesca, D.; Philabaum, C. Can Ternary Computing Improve Information Assurance?. Cryptography, MDPI, Feb 2018;
55. Niranjanamurthy, M.; Nithya1, B. N.; Jagannatha, S. Analysis of Blockchain technology: pros, cons and SWOT. Cluster Computing, https://doi.org/10.1007/s10586-018-2387-5, 2018;
56. Zhen, Z.; Xie, S.; Dai, H-N; Chen, X.; Wang, H. Blockchain challenges and opportunities: a survey. Int. J. Web and Grid Services, Vol. 14, No. 4, 2018;
57. Wu, J. Are blockchains immune to all malicious attacks?. Financial Innovation, Open Access, DOI 10.1186/s40854-016-0046-5; 2016;
58. Herbert, J.; Lichfield, A. A Novel Method for Decentralised Peer-to-Peer Software License Validation Using Cryptocurrency Blockchain Technology. Proceedings of the 38th Australasian Comp. Science Conf., 2015;
59. Harlev, M.; Yin, H.; Langenheldt, K.; Mukkamala, R.; Vatrapu, R. Breaking Bad: De-Anonymizing Entity Types on the Bitcoin Blockchain Using Supervised Machine Learning. Proceedings of the 51st Hawaii Int. Conf. on System Sciences, 2018;
60. Tosh, D.; Shetty, S.; Liang, X.; Kamhoua, C.; Kwiat, K.; Njilla, L. Security Implications of Blockchain Cloud with Analysis of Block Withholding Attack. 17th Int. Symp. Cluster, Cloud and Grid Comp., 2017;
61. Meng, W.; Tischhauser, E.; Wang, Q.; Wang, Y.; Han, J. When Intrusion Detection Meets Blockchain Technology: A Review. IEEE Access, 10.1109/ACCESS.2018.2799854, 2018;

62. Zheng, Z.; Xie, S.; Dai, H.; Chen, X.; Wang, H. An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends. IEEE 6th International Congress on Big Data, 2018;
63. Cambou, B.; Telesca, D. Ternary Computing to Strengthen Information Assurance, Development of Ternary State based public key exchange. IEEE, Comp. conf , July 2018;
64. Kiktenko, E.; Pozhar, N.; Anufriev, M.; Trushechkin, A.; Yunusov, R.; Kurochkin, Y.; Lvovsky, A.; Fedorov, A. Quantum Secured Blockchains. Open Source, arXiv:1705.09258v3 [quant-ph], 2018;
65. Semmouni, M.; Nitaj, A.; Belkasmi, M. Bitcoin Security with Post Quantum Cryptography. https://hal-normandie-univ.archives-ouvertes.fr/hal-02320898, 2019;
66. Kampanakisy, P.; Sikeridisz, D. Two Post-Quantum Signature Use-cases: Non-issues, Challenges and Potential Solutions. 7th ETSI/IQC Quantum Safe Cryptography Workshop; 2019;
67. Campbell, R. Evaluation of Post-Quantum Distributed Ledger Cryptography. Open Access, JBBA, Vol 2, https://doi.org/10.31585/jbba-2-1-(4)2019, 2019;
68. Gaj, K. Toward Efficient and Fair Software/Hardware Codesign and Benchmarking of Candidates in Round 2 of the NIST PQC Standardization Process. CryptArchi2019, 2019;
69. Andrzejzak, M. Accelerating Lattice Sieving in FPGAs. CryptArchi2019, 2019;
70. Ducas, L.; Kiltz, E.; Lepoint, T.; Lyubashevsky, V.; Schwabe, P.; Seiler, G.; Stehlé, D. CRYSTALS-Dilithium Algorithm Specifications and Supporting Documentation. https://pq-crystals.org/dilithium, 2019;
71. Buchanan, B. quantum robust hash-based signatures. medium.com/coinmonks. Jul 2018;
72. Kampanakis, P.; Fluhrer, S. LMS vs XMSS: comparison of two Hash-based Signature Standards. https://eprint.iacr.org/2017/349;
73. Becker, G. Merkle Signature Schemes, Merkle Trees. Seminar Bochum University, Jul. 2008.
74. Ducas, L.; Kiltz, E.; Lepoint, T.; Lyubashevsky, V.; Schwabe, P.; Seiler, G.; Stehlé, D. CRYSTALS-Dilithium Algorithm Specifications and Supporting Documentation; https://pq-crystals.org/dilithium, 2019;
75. Fouque, P-A.; Hoffstein, J.; Kirchner, P.; Lyubashevsky, V.; Pornin, T.; Prest, T.; Ricosset, T.; Seiler, G.; Whyte, W.; Zhang, Z. Falcon: Fast-Fourier Lattice-based Compact Signatures over NTRU. NIST PQC project round 2, documentation, 2019;
76. Bindel, N.; Akleylek, S.; Alkim, E.; Barreto, P.; Buchmann, J.; Eaton, E.; Gutoski, G.; Kramer, J.; Longa, P.; Polat, H.; Ricardini, J;. Zanon, G. Lattice-based digital signature scheme qTESLA. NIST PQC project round 2, documentation, 2019;
77. Andrzejzak, M. Accelerating Lattice Sieving in FPGAs. CryptArchi2019, 2019;
78. Casanova, A.; Faugere, J.-C.; Macario-Rat, G.; Patarin, J.; Perret, L.; Ryckeghem, J. GeMSS: A Great Multivariate Short Signature. NIST PQC project round 2, documentation, 2019;
79. Beullens, W.; Preneel, B.; Szepieniec, A.; Vercauteren, F. LUOV: Signature Scheme Proposal. NIST PQC project round 2, documentation, 2019;
80. Chen, M-S; Hulsing, A.; Rijneveld, J.; Samardjiska, S.; Schwabe, P. MQDSS specifications. NIST PQC project round 2, documentation, 2019;
81. Ding, J.; Chen, M-S; Petzoldt, A.; Schmidt, D.; Yang; B-Y. Rainbow. NIST PQC project round 2, documentation, 2019;
82. Cambou, B.; Philabaum, C.; Booher, D.; Telesca. D. Response-Based Cryptographic Methods with Ternary Physical Unclonable Functions. 2019 SAI FICC, IEEE; (accepted March 2019);

83. Cambou, B.; Philabaum, C.; Booher, D. Response-based Cryptography with PUFs. NAU case D2018-049, Jun 2018; (sponsored by AFRL);
84. Graves, L.; Lubell, J.; Yampolskiy, M.; King, W. Characteristic Aspects of Additive Manufacturing Security from Security Awareness Perspectives. IEEE Access Journal, Volume7, 29 July 2019.
85. Audit of the DoD's Use of Additive Manufacturing for Sustainment Parts DODIG-2020-003; Publicly released: October 21, 2019;
86. Ellis, D.; Schuster, F. Why additive manufacturing needs blockchain. Supply Chain Quarterly, Quarter 1 2019;
87. Independent Security Evaluators, "Ethercombing: Finding Secrets in Popular Places", https://www.ise.io/casestudies/ethercombing/, April 23, 2019;

LIST OF SYMBOLS, ABBREVIATIONS, AND ACRONYMS

AFRL	Air Force Research Laboratory
AM	Additive Manufacturing
APG	Addressable PUF Generator
CA	Certificate Authority
CRP	Challenge Response Pair
DoD	Department of Defense
DoS	Denial of Service
DRAM	Dynamic Random-Access Memory
DSA	Digital Signature Algorithm
ECC	Elliptic Curve Cryptography
EMI	Electro Magnetic Interference
FAA	False Acceptance Rate
FRR	False Reject Rate
FORS	Forest of Random Subset
GPU	Graphic Processor Unit
HPC	High Power Computer
ITS	Information Technology System
ISE	Independent Security Evaluator
LWE	Learning With Errors
MCU	Micro Controller Unit
MRAM	Magnetic Random-Access Memory
NIST	National Institute of Science and Technology
NTRU	Nth degree of TRUncated polynomial ring
PC	Personal Computer
PCB	Printed Circuit Board
PKI	Public Key Infrastructure
PUF	Physical Unclonable Function
PQC	Post Quantum Cryptography
QC	Quantum Computer
RBC	Response Based Cryptography
ReRAM	Resistive Random-Access Memory
RFID	Radio Frequency IDentification
RSA	Rivest Shamir Adleman
SHA	Standard Hash Algorithm
SRAM	Static Random-Access Memory
TAPKI	Ternary Addressable Public Key Infrastructure
WOTS	Winternitz One Time Signature
XOR	Exclusive OR