BLIC: A Blockchain Protocol for Manufacturing and Supply Chain Management of ICs

Sumanta Bose,* Mayank Raikwar,* Sourav Sen Gupta,*
Debdeep Mukhopadhyay,† Anupam Chattopadhyay,* Kwok-Yan Lam*
*School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798
Email: {sumanta001, mraikwar, sg.sourav, anupam, kwokyan.lam}@ntu.edu.sg
†Department of Computer Science and Engineering, Indian Institute of Technology, Kharagpur, India 721302
Email: debdeep@cse.iitkgp.ernet.in

***Abstract*—Blockchain technology has brought a huge paradigm shift in multiple industries, by integrating distributed ledger, smart contracts and consensus protocol under the same roof. Notable applications of blockchain include cryptocurrencies and large-scale multi-party transaction management systems. The latter fits very well into the domain of manufacturing and supply chain management for Integrated Circuits (IC), which, despite several advanced technologies, is vulnerable to malicious practices, such as overproduction, IP piracy and deleterious design modification to gain unfair advantages. To combat these threats, researchers have proposed several ideas like hardware metering, design obfuscation, split manufacturing and watermarking. In this paper, we show, how these issues can be complementarily dealt with using blockchain technology coupled with identity-based encryption and physical unclonable functions, for improved resilience against certain adversarial motives. As part of our proposed blockchain protocol, titled 'BLIC', we propose an authentication mechanism to secure both active and passive IC transactions, and a hybrid consensus protocol designed for IC supply chains. We present preliminary studies on the security, scalability, storage, privacy and anonymity analysis of BLIC.**

I. Introduction

Since the invention of semiconductors in the 1950s, it has played a pivotal role in the ongoing Information Technology (IT) revolution. The increasing density of components in an integrated circuit (IC), in conjunction with the skyrocketing costs of IC fabrication – have catapulted large 'fabless' design houses that partially or even completely rely on third-party manufacturers. Furthermore, to handle the complex System-on-Chip (SoC) designs, which consists of a multitude of components, system designers rely on a set of original equipment manufacturers (OEM), thereby establishing a complicated design flow across multiple abstractions. Naturally, such design flows are vulnerable and can be exploited by malicious suppliers, designers and manufacturers, who may resort to a number of techniques to gain unfair advantage. In Sec. II we discuss in detail about these vulnerabilities, current state-of-the art solutions, and the shortcomings therein. Then we introduce a {PUF + IBE} enabled blockchain protocol, titled 'BLIC' as an *alternate enabling technology* to deal with these vulnerabilities and restrict malicious activities. A detailed account of the proposed BLIC protocol for IC manufacturing and supply chain management is furnished in Sec. III. There we identify the components and processes in the blockchain framework, propose an authentication mechanism for secure IC transaction and a hybrid consensus protocol for blockchain validation in the supply chain network to maintain a distributed transaction ledger. In Sec. IV we analyze the security, scalability, storage, privacy and anonymity issues of the BLIC protocol. Finally in Sec. V we summarize and conclude the paper.

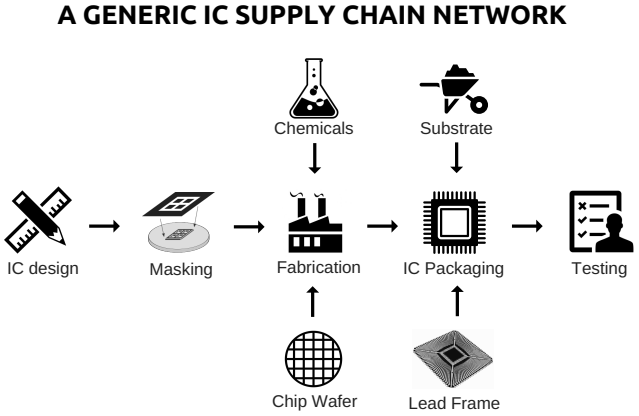


Fig. 1: A generic IC supply chain network. Source: Ref. [1] based on Taiwan's IC industry.

II. Background

Integrated circuit (IC) is the product of semiconductor manufacturing and its development consists of five major phases: (*i*) IC design (also called chip design), (*ii*) wafer fabrication, (*iii*) wafer probe, (*iv*) IC packaging (assembly) and (*v*) final quality test [1], [2]. Each phase is a major sub-industry of the whole semiconductor industry. In addition to these, a complete IC supply chain also includes mask, raw wafer, chemicals, substrate and lead frame. Fig. 1 shows a generic IC supply chain [1].

A. IC Supply Chain Vulnerabilities and Current State-of-the-art Solutions

The worldwide IC supply chain, however, is highly distributed and disaggregated. Semiconductor intellectual property (IP) for a typical design is procured from multiple global suppliers. Specialty services such as physical layout, reliability

engineering, package design, etc. are often involved, also from many disparate locations. And finally the fabrication-ready design is shipped to a remote wafer fabrication unit [3].

But the IC supply chain comes with its share of challenges, for which, it's important to understand today's IC-supply-chain ecosystem – from Electronic Design Automation (EDA) tools and IP reuse to manufacturing and packaging processes. Fabless companies, which comprise most of the IC design space, rely on IP cores, libraries, design services, software, and embedded operating systems. Once produced, most ICs (e.g., ASICs, FPGAs, etc.) are sold to system manufacturers to become part of a larger ecosystem before entering the end market as a complete product [4].

In the entire IC supply chain network, the design is exposed to a number of parties – designs that may be present in ICs of a range of devices such as smartphones, medical devices, autonomous vehicles, defense systems, etc. But once these devices are deployed, they might have certain rogue hardware or malicious software passively leaking critical information or actively sabotaging the device operation [5]. Currently, there are a number of vulnerabilities infiltrating the IC manufacturing and supply chain industry which include [6]:

- Piracy: An attacker can steal and claim the ownership of the IP, resulting in IP piracy [7].
- Trojan insertion: Malicious circuitry (Hardware Trojans) [8] might be inserted into the design by the attacker which can modify, control, disable, or monitor the communication and content of the circuit.
- Overproduction: An untrusted foundry/assembly having access to design IPs, may overbuild ICs and sell them illegally.
- Recycled: In today's IC supply chain, recycling is the most common vulnerability, accounting to over 80% of all counterfeits [6]. In recycling, ICs from outdated systems are recycled, i.e. they are taken from the outdated systems, repackaged and remarked and then sold in the market as new ICs [9].
- Remarked: Old marking on the package is removed and replaced with forged information [5], [6], [10].
- Cloned: Cloning is used to reduce to large development cost by copying the component without a legal IP. Cloning can be done either by reverse engineering or by obtaining IPs illegally [5], [6], [10].
- Out-of-spec/defective: An untrusted foundry/assembly may sell the out of spec or rejected components on open market, posing serious threats.
- Forged documentation: It includes fake certifications of compliance for some programs and forged log of a component.
- Side Channel Attack: Side channels in ICs such as timing and power information can be exploited to leak confidential data, which compromises the hardware implementation [11].

In existing literature, several techniques have been reported to detect and prevent the aforementioned vulnerabilities. Some of the current state-of-the-art preventive measures include:

- Hardware watermarking: It was introduced to protect the sensitive copyright information of ICs. It is achieved by embedding the designer's signature in the IC during design phase [12], [13]. At any later stage, the designer can reveal the watermark and verify the IC/IP. There are several techniques to construct watermark on IC such as FSM watermarking, constraint based watermarking etc.
- Hardware metering: It is a set of methodologies or security protocols, which are used to track the produced ICs [14]. Each IC and its functionality is tagged in a unique way. The metering process can be of two types: Passive or Active. In passive metering, ICs are registered using challenge response pairs and later matched against their record in the database. On the other hand, in active metering, some IC functionalities are locked by the designer and in later stage unlocked to track the IC and to prevent overproduction [15].
- Physically Unclonable function (PUF): PUF is a digital fingerprint used to uniquely identify an IC [16]. PUF is a suitable candidate for IC identification and authentication and it has received much attention recently from cryptography and hardware security communities [17]. In PUF, unique IDs of genuine ICs are stored in a secure database which is used to detect overproduced and cloned ICs in later stage.
- Hardware obfuscation: It is used to protect the IP by making it unintelligible [18], [19]. In this technique, structure of the hardware is modified to hide the functionality and implementation of hardware by inserting additional gates in the design which makes it significantly more difficult to reverse-engineer. It can be categorized into passive and active hardware obfuscation. Passive obfuscation does not change the functionality of the system while active obfuscation does.

However, these preventive techniques fall short of being able to tackle multiple vulnerabilities and the problem still remains open. IC counterfeiting and piracy stills continues to be a real issue.[1]

B. Motivation

Today, 90% of goods in the global trade market are at some point transported by ocean shipping containers [22], and this includes ICs and IC manufacturing raw materials. A complex series of interactions takes place for these shipments to get from one point to the next. Manual paper-based processes are still common, and information about the status of goods is locked away in organizational silos. Such hand-offs can lead to missing and sometime incomplete information, especially because it is compiled and transmitted through highly manual processes. The supply chain is also slowed down by the complexity and sheer volume of point-to-point communication

[1]It was estimated that the cost of counterfeiting and piracy for G20 nations was $450 to $650 billion in 2008, $1.2 to $1.7 trillion in 2015 and it is projected that it could reach $4.2 trillion by 2022, while putting 5.4 million legitimate jobs at risk [20], [21].

across a loosely coupled web of land transportation providers, freight forwarders, customs, brokers, governments, ports and ocean carriers. The financial overhead for processing documents and information for a shipment is estimated to cost more than twice that of the actual physical transportation [23].

Such problems, at large, can be addressed with a distributed permissioned platform accessible by the supply chain ecosystem designed to exchange event data and handle document workflows. Blockchain technology can be applied to create a global tamper-proof system for digitizing IC trade workflow and tracking end-to-end shipments, thereby eliminating costly point-to-point communications and trade disputes, if any. This has the potential to track millions of container journeys per year and integrate with customs authorities on selected trade lanes. Implementing blockchain technology would result in trust, security, transparency and efficiency in global IC manufacturing and supply chains by digitizing and simplifying trade. It would serve as a medium for secure data exchange and a tamper-proof repository of documents and events in the IC supply chain. This can significantly reduce shipment delays and frauds and generate sizable financial assets.[2]

But the application of blockchain for supply chain management is not new and has been in the limelight recently [25], [26]. However, thus far, to the best of the author's knowledge, blockchain-enabled supply chain solutions has been proposed for the automotive, mining, agricultural, retail industry, etc [26], [27], [28], [29]; but not for the IC manufacturing and supply chain industry. Most supply chains are susceptible to a range of certain similar problems, which can be dealt with 'off-the-shelf' blockchain solutions such as the IBM Hyperledger fabric [30]. But every supply chain network has its specific set of problems pertaining to the kind of goods in circulation and participating entities involved, such as those discussed in Sec. II-A specific to the IC manufacturing and supply chain industry. However, not all of them can be successfully tackled by the current state-of-the-art preventive measures. Under such a premise, we propose 'BLIC' – a BLockchain protocol for IC manufacturing and supply chain management. In this work, we shall discuss how our proposed BLIC protocol tackles the IC supply chain problems, and how it fares in comparison with existing preventive measures.

C. Related Work

Recently a number of emerging technologies and companies have employed blockchain as an alternate approach to induce a paradigm shift in seemingly traditional industries, such as:

- (Crypto)-currency: Bitcoin, Litecoin, Ripple, etc.
- (Smart)-contract: Ethereum, Codius, etc.
- Diamond certification: Everledger.
- Insurance: AIA, AXA.
- File and email stamping: Stampery.
- Notary service: Viacoin, Block Notary.
- Authorship certification: Ascribe.
- Decentralized storage: Storj, MetaDisk.
- Decentralized IoT: Filament, Telehash, IBM + Samsung (ADEPT).
- Anti-Counterfeit Solutions: BlockVerify, ChainLink.
- Decentralized DNS: Namecoin.
- Music Industry: FairMusic, PeerTracks, Ujo.
- Global Supply Chain: ShipChain, IBM + Maersk.

Among all the aforementioned blockchain applications, efforts by ShipChain or IBM and Maersk in global supply chain management come closest to the problem we're attempting to solve in this work. However, as discussed in Sec. II-B, no blockchain protocol has been reported yet that is capable of tackling the IC supply chain problems. Now, in Sec. III, we formally propose our BLIC protocol.

III. Proposed BLIC Protocol

A blockchain is essentially a distributed database of records or public ledger of all transactions or digital events that have been executed and shared among participating parties. Each transaction in the public ledger is verified by consensus of a majority of the participants in the system. The blockchain contains a certain and verifiable record of every single transaction ever made. A blockchain is characterized by the following three defining properties.

- Provenance: There is a single source of origin for all transactions, and there cannot be a conflict of transactions.
- Consensus: If and only if all (or a majority) of the participants agree of the validity of a transactions, it is saved into the blockchain.
- Immutability: Once a transaction is saved into the blockchain, it cannot be changed or deleted at a later time.

The main hypothesis is that the blockchain establishes a system of creating a distributed consensus in the digital online world. This allows participating entities to know for certain that an event (or a transaction) with an unique digital signature has occurred by creating an *irrefutable* record in a distributed public ledger. It opens the door for developing a democratic, open and scalable digital ecosystem from a centralized one.

In this paper, we describe how our proposed BLIC blockchain framework can be applied as the backbone for managing IC manufacturing and IC supply chain – it can save into the blockchain every valid IC transaction between the involved parties. This is in several ways similar to the notion of 'smart contracts' between the involved parties. Smart contracts are computer programs that can automate the execution of the terms of a contract. When a pre-decided condition is met in a smart contract among the participating entities, then the contractual agreement between the parties involved is automatically executed and payments are made as per the contract in a transparent manner. Except that in the case of managing the IC supply chain, each IC in the supply chain network is a *unique active device* in itself and can participate in validating the transaction and achieving consensus. This

[2]According to a 2013 World Economic Forum report [24], reducing supply chain barriers to trade could result in a worldwide GDP increase of 5% and a total trade volume increase of 15%. This can be facilitated by the penetration of blockchain technology into major global supply chains.

can be termed as 'smart property' [31]. The concept of smart property is related to the idea of controlling the ownership of a property or asset via blockchain using smart contracts. The property can be physical such as car, house, smartphone etc. or it can be non-physical such as shares of a company. In our case these properties are ICs.

Now we will discuss in detail about the components and processes in our BLIC protocol, followed by its workflow of authenticating secure IC transaction, consensus protocol and distributed transaction ledger.

A. Components of the BLIC Protocol

The framework of our proposed BLIC protocol for IC manufacturing and supply chain management consists of the following components.

- A set of N nodes representing the parties/entities involved in the BLIC blockchain IC supply chain network. These nodes are connected in a distributed network topology.
- A distributed ledger, i.e. the blockchain BC that logs execution results of all IC transactions between any two parties, maintained independently in all the nodes.
- A database (optionally encrypted) DB that maintains the IC supply chain transaction results of all the nodes. It is stored in a (*key*, *value*) format.
- A sender (*A*) and a recipient (*B*) executing a transaction of an IC from A to B. A and B can be any two chosen nodes among the N nodes, with $A \neq B$ (For transaction protocol see Sec. III-C).
- A transaction log T_l, generated and signed by both sender and recipient of the IC transaction (A and B) and broadcast to the network. T_l contains all the necessary details of that particular transaction.
- A set of validators V, peer nodes that validate the IC transaction log T_l, cast their vote and partake in achieving consensus on T_l.
- A transaction block T_b, which is a collection of several transaction logs encapsulated together based on a count cap or time-out. The transaction blocks are stored in the distributed ledgers of every node i.e. committed to the blockchain. (See Sec. III-D).
- A leader L, who keeps a chronological record of all the validated IC transaction blocks and commits them to its own BC. Note that $L \in V$, i.e. the leader belongs to the set of validators.
- A set of followers F, peer nodes who validate the successful commitment of the transaction block T with the leader L's ledger (blockchain). In case of conflict, the ledger of L is assumed to be the *truth*, and the F-member replicates L's ledger into its own.

Based on the aforementioned components, the IC supply chain blockchain framework consists of the following processes.

- Election mechanism based on the concept of Proof of Elapsed Time (PoET) utilizing *election timeout* as the criteria to determine the leader of the next term [32], [33]. To realize an efficient distributed consensus, PoET implements a fair lottery function characterized by:
  - Fairness: The leader election is distributed across the widest possible range of participating node population.
  - Investment: The cost incurred from the process of leader election is proportional to the offered value that is gained.
  - Verification: It is computationally simple to verify the legitimacy of the elected leader by all participating nodes.
- Cryptographic algorithmic authentication of sender, receiver and transacted IC.
- Cryptographic algorithmic access of data to validating peers for achieving consensus and committing transactions to the blockchain.

B. Workflow of the BLIC Protocol

The sequential workflow of the BLIC protocol is enumerated below.

1) Authenticating secure IC transaction (both active and passive IC), ensuring legitimate sender, receiver and IC.
2) Broadcasting the IC transaction details to all participants in the BLIC network.
3) Achieving consensus on the IC transaction from a majority of participants in the BLIC network.
4) Creating transaction records, and committing the transaction to an immutable distributed ledger i.e. to the BLIC blockchain.

In the sections to follow, each of the step will be discussed and analyzed in fair depth: Step 1 in Sec. III-C, and Steps 2 to 4 in Sec. III-D.

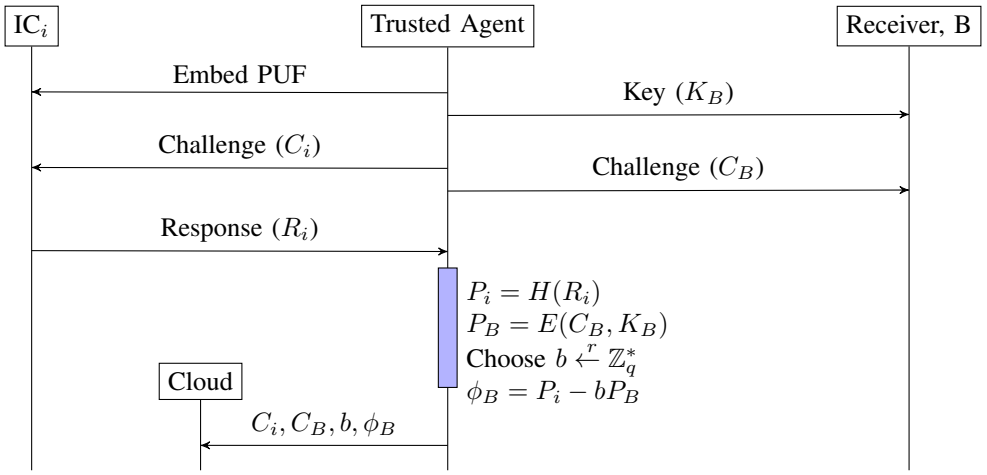
C. PUF with IBE Model for Authenticating Secure IC Transaction

In this section, we consider the scenario when an IC indexed i is ideally supposed to move from party A to party B. The following two arguments are imperative.

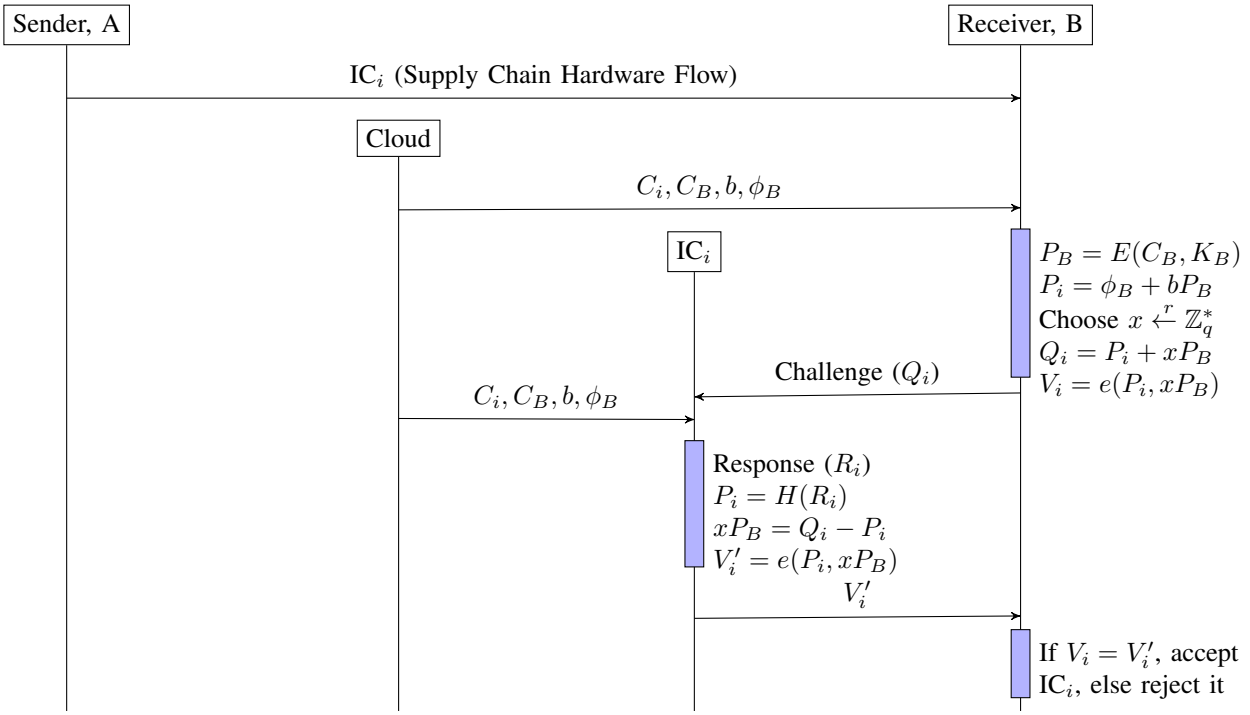
1) Party B is interested to determine that it is indeed interacting with (i.e. receiving) the IC indexed i, from the legitimate sender Party A.
2) A successful authentication between the IC and party B would ensure that both the IC and party B are legitimate.

To achieve this requirement, here we propose to employ an authentication and key exchange protocol called {PUF + IBE}, from [34], by combining the concepts of Identity Based Encryption (IBE) [35], [36] using Cryptographic Pairing with Physically Unclonable Functions (PUFs). This combination can help to do away with the requirement of explicitly storing the secret challenge-response pairs (CRPs). The proposed protocol is based on two security assumptions: (*i*) physical and mathematical unclonability of the constituent PUFs, and, (*ii*) computational intractability of the Elliptic Curve Discrete Logarithm Problem (ECDLP) [37].

Physically Unclonable Functions (PUFs) promise to be a critical hardware primitive to provide unique identities to



(a) Phase 1: Initial Enrollment (Identity-based Encryption)



(b) Phase 2: Authenticating Secure IC Transaction

Fig. 2: (a) Initial Enrollment Phase and (b) IC Authentication Phase in the PUF with IBE Model for Secure IC Transaction

billions of connected devices in the Internet of Things. In traditional authentication protocols a user presents a set of credentials with an accompanying proof such as password or digital certificate. However, for the IC supply chain management we need more evolved methods as these classical techniques are challenged by issues such as password dependency and inability to bind access requests to the ICs from which they originate in the supply chain. Additionally, the protocols need to be lightweight and heterogeneous. Although PUFs seem promising to address such problems, its unclonability property puts forward an open problem of how to develop such a mechanism without needing to store the challenge-response pair (CRP) explicitly at the verifier's end.

Here we will describe the protocol, but for this, first we will define some notations and conventions used in this protocol, as itemized below.

- $E(\cdot)$ is an encryption scheme such as AES.
- $H(\cdot)$ is a cryptographically secure hash function.
- $e(\cdot)$ is a standard bilinear pairing function [38].

The protocol has an initial enrollment phase, when a trusted

agent[3] embeds a PUF into the IC and also provides a secret key K_B to the party B, with which the IC has to interact in the future.

During this period, the trusted agent asks a challenge C_i to the IC indexed i and obtains a response, say R_i from its PUF. Let $P_i = H(R_i)$, where H is a cryptographically secure hash function. Likewise, for a challenge C_B the trusted agent computes $P_B = E(C_B, K_B)$, where E is some encryption like AES. Now, for the sake of convenience assume that the hash function and the encryption's domain belongs to an Elliptic curve group (wlog. we can always assume so as this can be achieved by suitable hash functions). Now the trusted agent choose randomly a scalar $b \in Z_q^*$, where q is a prime number. He estimates $\phi_B = P_i - bP_B$. Then the information, (C_i, C_B, b, ϕ_B) is stored in the cloud. It may be emphasized that the cloud can be public and hence the secrecy of this data is not required.

After this initialization phase, we are set to describe the authentication protocol between the sending party A and receiving party B for the IC indexed i in the IC supply chain. The authenticity of sending party A is determined by using standard public key cryptographic system, such as digital signature or public key encryption, which we outline later. Upon authentication, the IC is dispatched from party A to party B. It reveals its identity as i^{th} IC, and accesses the information (C_i, C_B, b, ϕ_B) from the cloud. Now party B uses its secret K_B and computes, $P_B = E(C_B, K_B)$. Likewise using b and ϕ_B it can compute $P_i = bP_B + \phi_B$. Now it creates a challenge which only the legitimate IC (indexed i with the corresponding PUF) can respond. It generates a random $x \in Z_q^*$, and computes $Q_i = P_i + xP_B$, and $V_i = e(P_i, xP_B)$, where e is a standard cryptographic bilinear pairing. Now, it challenges the IC with Q_i, and asks it to respond with V_i.

It is easy to see that the IC with the correct corresponding PUF can respond to the challenge: It also accesses the cloud database, and retrieves the values of (C_i, C_B, b, ϕ_B). It computes using its PUF, $P_i = H(R_i)$, where R_i is the PUF response for C_i. Then it computes $xP_B = Q_i - P_i$, and thus it can correctly compute $V_i' = e(P_i, xP_B)$. It responds with the value of V_i', and party B accepts only of $V_i = V_i'$.

The exact security arguments are given in [34], but broadly the uniqueness of PUF, and the hardness of Elliptic Curve Discrete Logarithmic Problem (ECDLP) are the most important factors responsible for the security of the protocol. The authentication will only pass if the IC indexed i is meant for party B.

Passive Device Transaction:: In a generic IC supply chain, the IC is not active until it reaches the fabrication phase, therefore to ensure a secure IC transaction before that, the IC should be encrypted first and then transferred to the legitimate recipient. On receiving the encrypted IC, the recipient should be able to decrypt the encrypted IC. So in a passive IC transaction, to ensure security, standard public key encryption (asymmetric encryption) scheme can be used where each party has a pair of public and private key, which can be used to encrypt and decrypt the IC. This technique ensures the confidentiality of the IC but it doesn't authenticate the sender of the IC. So to achieve the confidentiality as well as sender authentication, the IC must be signed first using the sender's private key and then encrypted using recipient's public key. This technique provides both confidentiality of IC and authenticity of sender but there is still a shortcoming with this approach. This technique doesn't convey much about the party who encrypted the IC. For example IC_i is transacted from sender A to recipient B. Then according to above sign/encrypt technique, the transaction of IC_i from sender A to recipient B will occur as :

$$A \rightarrow B : \{\{\text{IC}_i\}_{SK_A}\}_{PK_B}$$

Here IC_i is some plaintext containing it's identification and shipment information. SK_A and PK_B are the secret and public keys of the sender A and recipient B respectively. Here, the problem arises when B is malicious. B can simply decrypt and can encrypt using some third-party C's public key PK_C and create a transaction as:

$$A \rightarrow C : \{\{\text{IC}_i\}_{SK_A}\}_{PK_C}$$

Now it appears to recipient C that the transaction was initiated from A, while in reality, it actually wasn't. To tackle this problem, the sign/encrypt/sign or encrypt/sign/encrypt techniques can be used. These techniques provide the authentication of sender as well as confidentiality of transacted IC and protect against the aforementioned kind of attack. In case of a passive IC transaction, if we consider sign/encrypt/sign technique then the transaction of IC_i from sender A to recipient B will occur as :

$$A \rightarrow B : \{\{\{\text{IC}_i\}_{SK_A}\}_{PK_B}, \#PK_B\}_{SK_A}$$

$\#PK_B$ indicates that sender A hashes recipient B's key.
In the above transaction, first A signs IC_i with its secret key SK_A and then encrypts the signed IC_i using B's public key PK_B, and then A signs the ciphertext too using PK_B. The message received by B has following significance:
– Inner Signature: Ensures A is the sender of IC_i.
– Encryption: Ensures B can only decrypt the IC_i.
– Outer Signature: Ensures A used B's key to encrypt the IC_i.
In this way, we can ensure the authenticity of A and confidentiality of IC_i.

D. Proposed Hybrid Consensus Protocol

Consensus protocol is a mechanism of establishing an agreement or an unison over the validity of a shared piece of information, among a group of mutually distrusting partakers. In a blockchain network, it keep the ledger transactions synchronized by ensuring that ledgers only gets updated when transactions are approved by the appropriate participants, and that when ledgers do update, they update with the same transactions in the same order.

[3]A trusted agent could be a regulatory authority or a consortium of companies [39].

There are many different algorithms for building consensus based on requirements related to performance, scalability, consistency, threat model, and failure model. While most distributed ledgers operate with an assumption of Byzantine failures (malicious attacker), other properties are largely determined by application requirements. For example, ledgers used to record financial and supply chain transactions (such as our case of IC supply chain) often require fast transaction rates with relatively few participants and immediate finality of commitment. Consumer markets, in contrast, require substantial aggregate throughput across a large number of participants; however, short-term finality is less important.

Algorithms for achieving consensus with arbitrary faults generally require some form of voting among a known set of participants. Following two general approaches are common in this regard.

1) The first approach is based on traditional Byzantine Fault Tolerance (BFT) algorithms and uses multiple rounds of **explicit votes** to achieve consensus. Examples of this approach include the *Ripple* and *Stellar*, that work on consensus protocols that extend traditional BFT for open participation [40], [41].
2) The second approach is often referred to as 'Nakamoto consensus', wherein a leader is elected through some form of 'lottery'. The leader then proposes a block that can be added to a chain of previously committed blocks. The elected leader broadcasts the new block to the rest of the participants who **implicitly vote** to accept the block by adding the block to a chain of accepted blocks and proposing subsequent transaction blocks that build on that chain. An example of this approach is *Bitcoin*, where the first participant to successfully solve a cryptographic puzzle wins the leader-election lottery [42].

For the case of achieving consensus in our IC supply chain distributed network, we propose a hybrid model utilizing both of the above approaches. Our consensus protocol operates at two level – (*i*) The first level is a Byzantine Fault Tolerance (BFT) algorithm approach where participating nodes engage in explicit voting to achieve consensus on a valid transaction. (*ii*) The second level is a leader based approach established upon the concept of Proof of Elapsed Time (PoET). PoET is a Nakamoto-style consensus algorithm. It is designed to be a production-grade protocol capable of supporting large network populations.

Leader Election:

PoET essentially works as follows. Every participating node generates a random `electionTimeout` wait time $t \in \{T, 2T\}$ for a pre-decided network-wide value of T. The node waits for the duration t and declares itself a candidate for the leader election. The participating node with the shortest `electionTimeout` wait time is therefore the first to declare candidacy for the leader election, and is elected as the leader. We follow the leader election procedure as detailed in the RAFT consensus protocol [33].

The PoET leader election algorithm meets the criteria for a good lottery algorithm. It randomly distributes leadership election across the entire population of participating nodes with distribution that is similar to what is provided by other lottery algorithms. The probability of election is proportional to the resources contributed (in this case, resources are general purpose processors with a trusted execution environment). An attestation of execution provides information for verifying that the participating node waited for the `electionTimeout` wait time. Further, the low cost of participation increases the likelihood that the population of participating nodes will be large, increasing the robustness of the consensus protocol.

Consensus Protocol Mechanism:

Here we shall describe the steps in our hybrid consensus protocol used in the BLIC blockchain.

- Let us consider that a leader has been elected based on the concept of PoET described in the preceding subsection.
- For our hybrid consensus protocol, we will first use the BFT model to achieve a consensus based on **explicit voting**.
- In case of any conflict, we will mitigate it and achieve agreement based on the leader-follower model (Nakamoto-style consensus). The ledger of the leader is assumed to be the *truth*, and the follower **implicitly votes** by accepting the leader's block by adding it into its own blockchain.

Here we will describe the intermediate steps to achieve consensus among the participating nodes in the IC supply chain network. For this, first we will define some notations and conventions used in this protocol, as itemized below.

- $A \rightarrow B : Z$, denotes that A is the sender, B is the recipient and Z is the transfered entity.
- $a \| b$ denotes the concatenation of strings a and b.
- $ENC(\cdot)$ denotes encryption function.
- $DEC(\cdot)$ denotes decryption function.
- $SIG(\cdot)$ denotes signature generation function.
- $VER(\cdot)$ denotes verification function.
- Both party A (IC sender) and party B (IC recipient) have a set of public key and secret key: (PK_A, SK_A) for A and (PK_B, SK_B) for B respectively.

Now, we define our consensus protocol formally, executed in the following order.

1) After the transaction $[A \rightarrow B : i^{th}\ IC$ (**IC hardware**)] has been authenticated by both A and B, A generates a *transaction log* $T_{l,i}$ containing all the details of the i^{th} IC transaction. The log $T_{l,i}$ can have the fields described in Sec III-E.
2) A computes $T_A = T_{l,i} \| SIG\,(T_i, SK_A, PK_A)$
3) $A \rightarrow B : CT_A = ENC(T_A, PK_B)$
4) B performs the following steps:
   a) $DEC(CT_A, SK_B, PK_B) = T_A$ and extract $T_{l,i}$ from T_A.
   b) $VER(T_{l,i}, SIG(T_{l,i}, SK_A, PK_A), PK_A) \stackrel{?}{=} true$
   c) If step (b) is not true, B requests A to send the transaction ciphertext CT_A again, else continue.

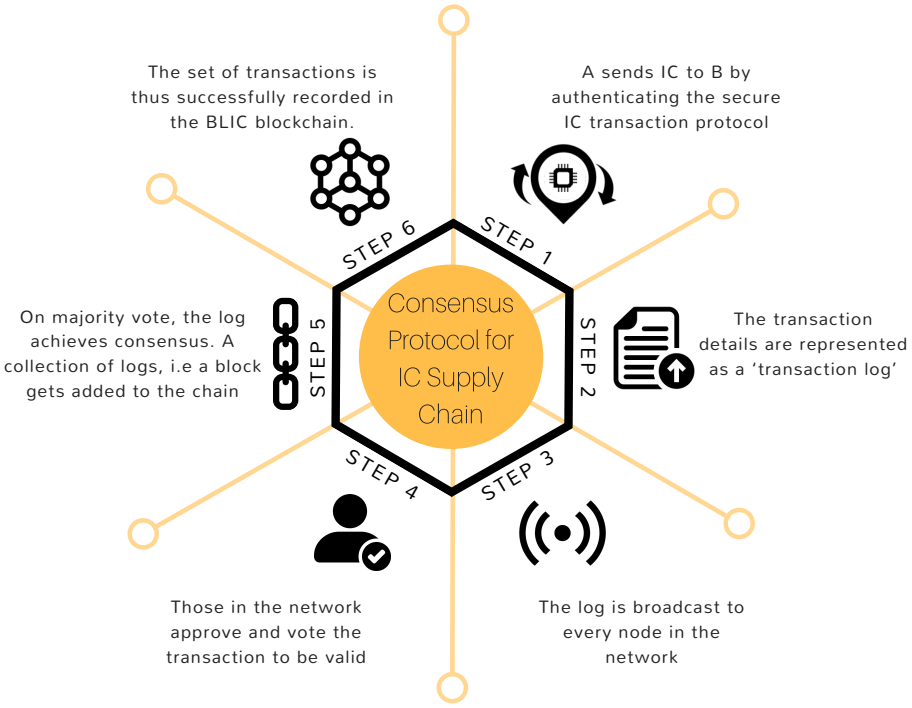


Fig. 3: Workflow of our proposed BLIC blockchain consensus protocol for IC manufacturing and supply chain network.

d) If contents of $T_{l,i}$ are incorrect, B notifies A and requests to send the transaction ciphertext CT_A again, else continue.

5) B computes $T_{A,B} = T_A \| SIG\,(T_A, SK_B, PK_B)$
6) $B \rightarrow$ *All nodes in network* : $CT_{A,B} = ENC(T_{A,B}, PK_M)$, where PK_M is the public key of the M^{th} node in the network $\forall M \in \{1, 2, \dots N\}$ if there are a total of N nodes in the network.
7) M^{th} node in the network $\forall M \in \{1, 2, \dots N\}$ performs the following steps:
   a) $DEC(CT_{A,B}, SK_M, PK_M) = T_{A,B}$ and extract T_A from $T_{A,B}$, and $T_{l,i}$ from T_A.
   b) $VER(T_A, SIG(T_A, SK_B, PK_B), PK_B) \stackrel{?}{=} true$
   c) If step (b) is not true, M^{th} node requests B to send the transaction ciphertext $CT_{A,B}$ again, else continue.
   d) $VER(T_{l,i}, SIG(T_{l,i}, SK_A, PK_A), PK_A) \stackrel{?}{=} true$
   e) If step (d) is not true, M^{th} node requests A to send the transaction ciphertext CT_A to B again, else continue.
8) At this stage the IC transaction $[A \rightarrow B : i^{th}\ IC$ (**IC hardware**)] which had been authenticated by both A and B, has been successfully validated by the M^{th} node in the network $\forall M \in \{1, 2, \dots N\}$. Every nodes broadcasts to every other node in the network including itself about its status (vote) on the *transaction log* $T_{l,i}$. Every follower node F has to ensure that it's vote has been recorded by at least the leader L (L acknowledges the vote receipt to every follower).
9) Every node counts the successful validation vote corresponding to the *transaction log* $T_{l,i}$ from all the nodes in the network. If $T_{l,i}$ receives majority vote (validated by a majority of nodes), then *consensus* has been achieved for *transaction log* $T_{l,i}$, else consensus has not been achieved.
10) A collection of several transaction logs are encapsulated together to construct a *transaction block* T_b. The transaction block T_b is then stored in the distributed ledgers of every node i.e. committed to the BLIC blockchain, BC.
11) Every follower node F, validates the successful commitment of the *transaction block* T_b with the leader L's ledger. In case of conflict, the ledger of L is assumed to be the *truth*, and the F-member replicates L's ledger into its own.

Following the consensus protocol steps, described above, Fig. 3 summarized as a simplified six-step process, how two parties A and B having a transaction of an IC would achieve consensus on the transaction and commit it to the blockchain. The steps are:

(i) A sends IC to B by authenticating the secure IC transaction protocol described in Sec. III-C.
(ii) The transaction details are represented as a 'transaction log'.
(iii) The log is broadcast to every node in the network.
(iv) Those in the network approve and vote the transaction log to be valid.
(v) On majority vote, the log achieves consensus. A collection of logs, i.e a block gets added to the chain, providing an immutable and transparent record of transactions.
(vi) The set of transactions is thus successfully recorded in

the BLIC blockchain, BC.

E. Distributed Transaction Ledger

Each IC transaction is represented as a *transaction log* T_l consisting of transaction information such as: serial no. sender, recipient, item description, IC ID, date, time, credit, debit, balance, memo, etc. The transaction log is broadcast to every node in the BLIC blockchain network and is subjected to verification by voting by peer nodes. After successful verification by a majority of nodes based on the consensus protocol described in Sec. III-D, a set of transaction logs can be encapsulated as a *transaction block* T_b and recorded in the distributed public ledger (i.e. committed to the BLIC blockchain, BC). The encapsulation is triggered by a pre-decided count of transaction logs or a T_b-creation-timeout, whichever happens earlier.

Every single transaction needs to be verified for validity before it is recorded in the public ledger. A verifying node needs to ensure three things before verifying any transaction:

(i) Spender owns the IC – digital signature verification of the sender on the transaction.
(ii) Spender has spent the correct IC – digital signature verification of the IC on the transaction.
(iii) Spender has sufficient IC in his/her account: checking every transaction against spender's account ('public key') in the ledger to make sure that he/she has sufficient IC balance in his/her account.

However, there is a need to maintain the order of these transactions that are broadcast to every other node in the IC supply chain network to ensure that double-spending of the ICs do not occur. Considering that the ICs are passed node by node through the network, there has to be a guarantee that the order in which they are received at a node is the same order in which the transactions are generated. This requirement of ordering the transactions is achieved by placing them in groups i.e. in transaction blocks and then linking them with a timestamp. The transactions in a particular block are considered to have occurred at the same time. These blocks are linked to each-other in a linear and chronological order with every block containing the hash of the previous block. Moreover, transactions are paired in a Merkle tree fashion, which allows us to verify them as needed and not include the body of every transaction in the block header, while still providing a way to verify the entire blockchain. The Merkle root enables us to verifies the integrity of all of the transactions. If any transaction is added/removed or modified, it will change the hash of its parent and induce a domino effect resulting in the Merkle root's hash changing as well, enabling us to detect any forgery.

IV. Security, Scalability, Storage and Privacy/Anonymity Analysis

Here, we report an in-depth analysis of the security, scalability, storage and privacy/anonymity aspects of the BLIC protocol.

A. Security Analysis

Security of our BLIC protocol is based on following characteristics:

1) BLIC blockchain ledger is immutable.
2) BLIC consensus can not be breached.
3) For every transaction, sender, receiver and IC authenticity is enforced.

In Table I we provide a security analysis by comparison of traditional preventive measures vs. our proposed BLIC protocol against the common vulnerabilities infiltrating the IC manufacturing and supply chain industry. Now we shall discuss how the BLIC protocol fares against each of the vulnerabilities.

- Piracy: The BLIC protocol prevents an attacker from stealing an IP owing to IP confidentiality in each phase of the supply chain. That makes it harder for an attacker to steal and claim the IP. In BLIC, during active IC transaction confidentiality of IP is achieved using {PUF + IBE}, while during passive IC transaction it is achieved by sign/encrypt/sign or encrypt/sign/encrypt techniques.
- Trojan insertion: Trojans can be injected in an IC during the design or fabrication phase. Till either of these phases, the IC is not active. Trojan can be detected using standard Trojan detection techniques such as IC fingerprinting [44]. If a Trojan is detected, the BLIC protocol identifies the path it followed in the supply chain. Say, a Trojan is detected in some phase P_i – then from the ledger, a transaction path such as $P_1 \rightarrow P_2 \rightarrow \ldots \rightarrow P_i$ can be detected, where P_1, P_2 are some preceeding phases in the IC supply chain.
- Overproduction: Overproduced ICs can be detected under authentication of the ICs using the {PUF + IBE} model.
- Recycled: In the BLIC protocol, each IC transaction is stored in the blockchain which consists of IC ID and the transacting parties. So if an attacker tries to recycle an old IC_i using repackaging from transacting party X in the IC supply chain, then the BLIC blockchain would restrict this by denying transaction since that IC_i has already been processed by party X. This is done by checking the corresponding entry of IC_i authentication by party X in the BLIC blockchain.
- Remarked: Remarked ICs can be detected by the BLIC protocol during IC authentication. It is hard to forge the PUF by an attacker, thus during the IC authentication phase the forged ICs fail to produce the expected response.
- Cloned: In the BLIC protocol, we're using PUFs to detect cloned ICs, since unique IDs are generated from the randomness of every IC. Cloned ICs fail to generate those IDs and can be detected during the IC authentication phase.
- Out-of-spec/defective: In generic IC supply chain, defective ICs are identified using manual testing, wherein a set of test vectors is prepared and each IC tested against it. In the BLIC protocol, we can automate this process

Vulnerabilities \ Preventive measures	Hardware watermarking	Hardware metering	Physically Unclonable function (PUF)	Hardware obfuscation	BLIC protocol (this work)
Piracy	✓ [43]	✓ [15]	✓ [43]	✓ [43]	✓
Overproduction	✗ [5]	✓ [5], [6]	✓ [5], [6]	✓ [43]	✓
Recycled	✗ [5]	✗ [5]	✗ [5]	✗ [43]	✓
Remarked	✗ [5]	✗ [5]	✓ [6], [43]	✗ [43]	✓
Cloned	✓ [5]	✓ [5], [6]	✓ [5], [6]	✓ [43]	✓
Forged documentation	—	—	—	—	✓

TABLE I: A security analysis by comparison of traditional preventive measures vs. our proposed BLIC protocol against the common vulnerabilities infiltrating the IC manufacturing and supply chain industry. Here, '✓' indicates that the preventive measure is effective against the vulnerability, while '✗' indicates it is ineffective, and '—' indicates inconclusive data.

to avoid errors during manual testing. This can be done running a smart contract to execute the test vectors and identify defective ICs. If any IC does not produce the expected output as per the smart contract (test vectors), that IC considered as defective.

- Forged documentation: One of the primary objectives behind blockchain implementation is to prevent forgery. To prevent forged documentation, certification or change-logs in the IC supply chain, every IC transaction document (or its hash) can be stored in the blockchain. At any later stage, it can be used to verify the documentation related to any previous IC transaction between any two parties to prevent forgery.
- Side Channel Attack: Side channel attacks poses a serious threat to the security of cryptography modules. Our BLIC protocol does not prevent side channel attacks. These attacks can only be prevented by standard techniques such as power analysis attack prevention [45], [46].

B. Scalability Analysis

The solution to scalability in a traditional database system is to increase computational power (i.e. add more servers) to handle the added transactions. In a decentralized blockchain network, however, every transaction needs to be processed and validated by every node. Thus to get faster, we need to add more computational power to every node in the network.[4] With the growing size of a network, the computational power, storage and bandwidth requirements of the participating nodes increases. This may lead to the risk of centralization, because at some point, it would only be feasible for a subset of the total node population to process a block.

So, in order to scale the BLIC blockchain protocol, we must have a mechanism to limit the number of participating nodes necessary to validate each transaction, while still ensuring the BLIC network's trust on the validity of each transaction. In this context, the following three requirements are imperative.

[4]In permission-less (public) blockchains, we have little or no control over the public nodes in the network. Therefore, there is a trade-off between low transaction throughput and high degree of centralization in most permission-less blockchain decentralized consensus protocols. However, for permissioned (private) blockchain networks like BLIC, we have more control over the network nodes, ergo over the scalability.

- Since every node doesn't have the responsibility to validate every transaction, they must have a statistical and economic trust mechanism to ensure that other transactions (those not personally validated by them) have been securely validated.
- There must be some way to guarantee transaction data availability. In other words, even for a transaction valid from the perspective of a node that has not directly validated it, unavailability of that transaction data (due to malicious attack, node power loss, etc.) would lead to a deadlock, and no other node can validate or create new transactions.
- In order to achieve scalability, different nodes must have the capability of processing transactions in parallel. However, blockchain state transitioning also has several serial (non-parallelizable) components, so there are certain restrictions on transitioning blockchain state, while balancing both scalability and parallelizability.

Some possible approaches of scaling the BLIC blockchain protocol are discussed below.

1) Off-chain state channels: State channels are essentially a mechanism by which interactions that were normally meant to occur within the blockchain would instead occur off the blockchain. This is done in a cryptographically secure way without increasing the risk of any participant, while providing significant improvements in cost and speed. In future, state channels could be a critical part of scaling the BLIC blockchain to support higher levels of use, as recent reports have shown [47]. A state channel can be implemented as follows:

1) A part of the BLIC blockchain would be locked using multi-signature (or some form of smart-contract), which can only be updated if a specific set of participants agree to it.
2) The participants would make updates amongst themselves by constructing and cryptographically signing transactions without submitting it to the blockchain. Each new update overrides previous updates.
3) At some later point, participants would submit the state back to the BLIC blockchain, which would close the state channel and unlock the state again.

Steps 1 and 3 involve blockchain operations which are

published to the network, pay fees and wait for confirmations. However, Step 2 does not involve the blockchain at all. It can contain an unlimited number of updates and can remain open indefinitely. In this sense, the BLIC blockchain can be used purely as a settlement layer to process the final IC transactions of a series of interactions for the final settlement, which helps lifts the burden from the underlying blockchain network. Not only can the transactional capacity be increased with state channels, but they can also provide two other very important benefits: increased speed and lower overhead.

2) Plasma: Plasma is a recently introduced technology and is a promising solution for scaling blockchain computation [48]. It can be visualized as a series of contracts running on top of a root blockchain. The validity of the state is enforced by the root blockchain in the Plasma chains using something called 'fraud proofs', which is a mechanism by which nodes can determine if a block is invalid. To implement this, the BLIC blockchains must be structured as a tree hierarchy, where each branch must be treated as a blockchain that has its own blockchain history and computations that are map-reducable. The child chains can be called 'Plasma blockchains', each of which are a chain within the main BLIC blockchain. The Plasma blockchain does not disclose the contents of the blockchain on the root chain. Instead, only the blockheader hashes are submitted on the root chain, which is enough to determine validity of the block. If there is proof of fraud submitted on the root chain, then the block is rolled back and the block creator is penalized, i.e. data is only submitted to the root chain in Byzantine conditions. As a result, the root blockchain would process only a tiny amount of commitments from child blockchains, which in turn decreases the amount of data passed onto the root blockchain and allows for a much larger number of computations. In addition, data is only propagated to those who wish to validate a particular state. This makes IC transaction validation more scalable by eliminating the need for every node to watch every chain. Instead, they only watch the ones they are impacted by in order to enforce correct behavior and penalize fraud. Fraud proofs allows any node to enforce invalid blocks and ensure that all state transitions are validated. Additionally, if there is an attack on a particular chain, participants can rapidly and cheaply do a mass-exit from the corrupt child chain.

Plasma might seem similar to state channels implementations that handle transactions off chain. The main difference between state channels and Plasma is that with Plasma, not all participants need to be online to update state. Moreover, the participants do not need to submit data to the root blockchain in order to participate and confirm transactions. An advantage of Plasma is that, state channel type solutions could be the main interface layer for rapid IC transactions on top of Plasma, while Plasma could update the state with minimal root chain state commitments.

3) Off-chain computations: Off-chain computations can be an enabler in scaling transactions in blockchain networks (Ex.:TrueBit for smart contracts [49]). Essentially, just like state channels, this approach would use a layer outside the blockchain to do the heavy lifting. It is a system that verifiably executes computations off-chain that would be otherwise computationally expensive to execute on-chain. This is how it can be implemented for the BLIC blockchain: Instead of every node participating, specific participants in the network, called 'Solvers', perform the computations made by IC transaction smart contracts and submit the transaction log, along with a virtual deposit. If the Solver's log is correct, then the solver is rewarded and the deposit returned. Otherwise, if the Solver cheated, the deposit is forfeited and any dispute is resolved on the BLIC blockchain using the 'Verification Game'. The Verification Games goes like this: There is a set of participants in the network called 'Verifiers' who check the Solvers' work off the blockchain. The solution is accepted by the system, if no error is reported by any Verifier. If a Verifier does dispute the correctness of the Solver's solution, the game proceeds in a series of rounds to settle the dispute on the blockchain, where 'Judges' in the network with limited computational power adjudicate all disputes. The system is built to ensure that in a modest round of interactions the Judges can settle the dispute with a relatively small amount of work compared to that required for performing the actual task off the blockchain. At the end of this game, if the Solver was in fact cheating, it will be discovered and penalized. If not, then the challenger (challenging Verifier) will pay for the resources consumed by the false alarm.

Overall, the protocol allows any node to post an IC transaction task, and any other node to receive a reward for completing it, while the system's incentive structure guarantees the correctness of transaction log. And by moving the computations and verification process off the blockchain into a separate protocol, it can scale to large numbers of transactions per unit time without significant constraints.

4) Sharding: Sharding in the blockchain world is similar to database sharding in traditional software systems. With traditional databases, a shard is a horizontal partition of the data in a database, where each shard is stored on a separate database server instance. This helps spread the load across different servers. Similarly, with blockchain sharding, the overall state of the blockchain is separated into different shards, and each part of the state would be stored by different nodes in the network [50]. Transactions that occur on the network are directed to different nodes depending on which shards they affect. Each shard only processes a small part of the state and does so in parallel. In order to communicate between shards, there needs to be some message-passing mechanism. Overall, to implement sharding in our BLIC blockchain, it would require us to create a network where every node only processes a small portion of all transactions, while still maintaining high security.

C. Storage Analysis

In the BLIC IC authentication protocol, a sizeable amount of data is stored in the secure cloud. To be specific, for each IC_i and for each receiving party of IC_i, some information is stored in the cloud during initial enrollment phase, which

is used to authenticate the secure IC transaction (See Fig. 2). Mathematically, the amount of data is proportional to the product of number of ICs and number of IC recipient nodes in the network. In an IC supply chain, a single IC moves through several parties, so the data for that particular IC and corresponding recipients is stored in the cloud. However, note that after the authentication of the secure IC transaction is completed by the receiving party, there is no use of the corresponding data (used in the IC authentication phase). So to save cloud-space, that data can be purged from the cloud after the specific IC authentication is completed. Data purging from the cloud can be triggered after a pre-decided timeout following the IC authentication completion.

D. Privacy and Anonymity Analysis

In the BLIC protocol, anonymity can be achieved using several techniques. Access control can be employed to make some parts of the IC supply chain anonymous for selected users i.e. every participating node has restrictive access over certain parts of data in the blockchain, which they're authorized to access. The choice of access control type could be either attribute based or role based depending on the application. Additionally, alternate techniques such as transaction mixer/tumbler [51] and joint transaction [52] can be used to achieve anonymity and privacy.

Note that, it is during the transfer of an IC from an current owner to the future owner when anonymity and privacy must be ensured judiciously. In a case of ownership transfer of IC_i, three parties are involved: (*i*) the IC_i whose ownership is being transferred, (*ii*) the current owner who's responsible for initiating the ownership transfer protocol, and (*iii*) the future owner who would take control of the ownership once the protocol concludes. The ownership transfer protocol must be equipped with security requirements that protect the privacy and anonymity of the involved parties. During the process, the following requirements must be met: (*i*) the privacy of the current owner and future owner should be preserved, (*ii*) an adversary must not be able to learn about the previous chain of owners, and (*iii*) an adversary must not be able to breach current ownership statements.

The privacy and anonymity of the current and future owner can be ensured using pseudonyms. Regarding transacted ICs, the privacy and anonymity can be achieved by (*i*) {PUF + IBE} for active ICs and (*ii*) sign/encrypt/sign or encrypt/sign/encrypt techniques for passive ICs, as described in Sec. III-C.

Additionally, there are various other techniques to achieve privacy and anonymity of ICs and transacting parties. Signature schemes such as threshold signature, ring signature, blind signature can be used to achieve this goal. In a supply chain, some parties may not want to disclose their identity to other parties, so to ensure their anonymity, variants of zero-knowledge proofs can be used. Moreover, to achieve the privacy of ICs, encryption schemes such as broadcast or multicast encryption could be applied.

V. Summary and Conclusion

In this manuscript, we focus on the disruption that the Blockchain technology can bring in the IC supply chain management industry. Blockchain technology has potential to become the new engine of growth in digital economy where we are increasingly using Internet to conduct digital commerce and sharing data and value. In particular, this paper proposes a blockchain protocol, titled 'BLIC' that can be used to manage the IC manufacturing and supply chain industry to tackle the various counterfeit issues arising in today's IC supply chains. The {PUF + IBE} model used in BLIC gives an insight about IC authentication without storing challenge-response pair (CRP). We propose a hybrid consensus protocol which provides a better understanding on how two party authentication (sender and recipient) takes place in a supply chain and how the transaction is validated and added as secure, immutable and irrefutable blocks in the blockchain. Security analysis describes how security is achieved in IC supply chain and how the possible counterfeits are detected and avoided in our model. In this paper, we have also analyzed the scalability of our overlay network and storage analysis introduces the proposed methods to limit or reduce the space in the public cloud used in secure IC transaction.

Acknowledgement

We would like to acknowledge the support from *<list of funding agencies and corresponding account numbers>*. Additionally we would like to thank Dr. Sumit Kumar Pandey for valuable discussion on secure IC transaction protocol.

References

[1] Year book of Taiwan Semiconductor Industries. ITIS Project, IEK Center of Industrial Technology Research Institute, Hsinchu (Taiwan), 2007.

[2] J. C. Chen, H. Rau, C.-J. Sun, H.-W. Stzeng, and C.-H. Chen, "Workflow design and management for IC supply chain," in *2009 Intl. Conference on Networking, Sensing and Control*, March 2009, pp. 697–701.

[3] Atrenta, "The IC Supply Chain: The Day After Tomorrow," http://semiengineering.com/ic-supply-chain-day-tomorrow/, 2013, [Online; accessed 03-Dec-2017].

[4] "State of China IC Design Industry 2012," https://www.gsaglobal.org/2012/12/20/state-of-china-ic-design-industry-2012/, 2012.

[5] U. Guin, K. Huang, D. DiMase, J. M. Carulli, M. Tehranipoor, and Y. Makris, "Counterfeit integrated circuits: A rising threat in the global semiconductor supply chain," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1207–1228, Aug 2014.

[6] U. Guin, D. DiMase, and M. Tehranipoor, "Counterfeit Integrated Circuits: Detection, Avoidance, and the Challenges Ahead," *J. Electronic Testing*, vol. 30, no. 1, p. 9, 2014. [Online]. Available: https://doi.org/10.1007/s10836-013-5430-8

[7] J. A. Roy, F. Koushanfar, and I. L. Markov, "EPIC: Ending Piracy of Integrated Circuits," in *2008 Design, Automation and Test in Europe*, 2008, pp. 1069–1074.

[8] M. Tehranipoor and F. Koushanfar, "A survey of hardware trojan taxonomy and detection," *IEEE Design Test of Computers*, vol. 27, no. 1, pp. 10–25, Jan 2010.

[9] X. Zhang and M. Tehranipoor, "Design of On-Chip Lightweight Sensors for Effective Detection of Recycled ICs," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 5, pp. 1016–1029, May 2014.

[10] M. M. Tehranipoor, U. Guin, and D. Forte, *Counterfeit Integrated Circuits*. Cham: Springer International Publishing, 2015, pp. 15–36. [Online]. Available: https://doi.org/10.1007/978-3-319-11824-6_2

[11] U. Rührmair, X. Xu, J. Sölter, A. Mahmoud, M. Majzoobi, F. Koushanfar, and W. Burleson, *Efficient Power and Timing Side Channels for Physical Unclonable Functions*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 476–492. [Online]. Available: https://doi.org/10.1007/978-3-662-44709-3_26

[12] E. Charbon, "Hierarchical watermarking in ic design," in *Proceedings of the IEEE 1998 Custom Integrated Circuits Conference*, May 1998, pp. 295–298.

[13] A. B. Kahng, J. Lach, W. H. Mangione-Smith, S. Mantik, I. L. Markov, M. Potkonjak, P. Tucker, H. Wang, and G. Wolfe, "Constraint-based watermarking techniques for design ip protection," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 20, no. 10, pp. 1236–1252, Oct 2001.

[14] F. Koushanfar and G. Qu, "Hardware metering," in *Proceedings of the 38th Annual Design Automation Conference*. New York, NY, USA: ACM, 2001, pp. 490–493. [Online]. Available: http://doi.acm.org/10.1145/378239.378568

[15] Y. M. Alkabani and F. Koushanfar, "Active hardware metering for intellectual property protection and security," in *Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium*, ser. SS'07. Berkeley, CA, USA: USENIX Association, 2007, pp. 20:1–20:16. [Online]. Available: http://dl.acm.org/citation.cfm?id=1362903.1362923

[16] R. Pappu, B. Recht, J. Taylor, and N. Gershenfeld, "Physical one-way functions," *Science*, vol. 297, no. 5589, pp. 2026–2030, 2002. [Online]. Available: http://science.sciencemag.org/content/297/5589/2026

[17] G. E. Suh and S. Devadas, "Physical unclonable functions for device authentication and secret key generation," in *2007 44th ACM/IEEE Design Automation Conference*, June 2007, pp. 9–14.

[18] G. T. Becker, M. Fyrbiak, and C. Kison, *Hardware Obfuscation: Techniques and Open Challenges*. Cham: Springer International Publishing, 2017, pp. 105–123. [Online]. Available: https://doi.org/10.1007/978-3-319-50380-6_6

[19] R. S. Chakraborty and S. Bhunia, "Harpoon: An obfuscation-based soc design methodology for hardware protection," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 28, no. 10, pp. 1493–1502, Oct 2009.

[20] International Chamber of Commerce, "Estimating the global economic and social impacts of counterfeiting and piracy", (published by Frontier Economics) 2011.

[21] International Chamber of Commerce, "The Economic Impacts of Counterfeiting and Piracy", (published by Frontier Economics) 2016.

[22] J. Castonguay, "International Shipping: Globalization In Crisis, *Witness magazine*," http://www.visionproject.org/images/img_magazine/pdfs/international_shipping.pdf, 2007, [Online; accessed 14-Jan-2018].

[23] K. Sadouskaya, "Adoption of Blockchain Technology in Supply Chain and Logistics," https://www.theseus.fi/bitstream/handle/10024/126096/Adoption%20of%20Blockchain%20Technology%20in%20Supply%20Chain%20and%20Logistics.pdf, 2017, [Online; accessed 14-Jan-2018].

[24] W. E. F. I. collaboration with Bain & Company and the World Bank), "Enabling Trade Valuing Growth Opportunities," http://www3.weforum.org/docs/WEF_SCT_EnablingTrade_Report_2013.pdf, 2013, [Online; accessed 14-Jan-2018].

[25] K. Korpela, J. Hallikas, and T. Dahlberg, "Digital Supply Chain Transformation toward Blockchain Integration," in *Proceedings of the 50th Hawaii International Conference on System Sciences*, 2017.

[26] S. Virthachalam, "Future of Automotive Supply Chain," http://www.umtri.umich.edu/sites/default/files/Senthil.Virthachalam.IBM_.IT_.2017.pdf, 2017, [Online; accessed 14-Dec-2018].

[27] P. Rizzo, "World's Largest Mining Company to Use Blockchain for Supply Chain," https://www.coindesk.com/bhp-billiton-blockchain-mining-company-supply-chain/, 2016, [Online; accessed 14-Jan-2018].

[28] G. Intelligence, "Blockchain: Beyond Bitcoin to Agriculture," https://gro-intelligence.com/insights/blockchain-in-agriculture, 2017, [Online; accessed 14-Jan-2018].

[29] Cognizant, "Retail: Opening the Doors to Blockchain," https://www.cognizant.com/whitepapers/retail-opening-the-doors-to-blockchain-codex2879.pdf, 2017, [Online; accessed 14-Jan-2018].

[30] A. N. Mencias, "IBM Blockchain: High Secure Business Network," http://www.vm.ibm.com/education/lvc/LVC0517.pdf, 2017, [Online; accessed 14-Jan-2018].

[31] M. Crosby, Nachiappan, P. Pattanayak, S. Verma, and V. Kalyanaraman, "BlockChain Technology Beyond Bitcoin," http://scet.berkeley.edu/wp-content/uploads/BlockchainPaper.pdf, 2015, [Online; accessed 03-Dec-2017].

[32] H. Fabric, "Sawtooth's Ledger — Consensus — Proof of Elapsed Time (PoET)," https://sawtooth.hyperledger.org/docs/core/releases/latest/introduction.html#proof-of-elapsed-time-poet, 2017, [Online; accessed 08-Dec-2017].

[33] D. Ongaro and J. Ousterhout, "In Search of an Understandable Consensus Algorithm," https://raft.github.io/raft.pdf, 2014, [Online; accessed 08-Dec-2017].

[34] U. Chatterjee, V. Govindan, R. Sadhukhan, D. Mukhopadhyay, R. S. Chakraborty, D. Mahata, and M. Prabhu, "PUF+IBE: Blending Physically Unclonable Functions with Identity Based Encryption for Authentication and Key Exchange in IoTs," *IACR Cryptology ePrint Archive*, vol. 2017, p. 422, 2017. [Online]. Available: http://eprint.iacr.org/2017/422

[35] A. Shamir, *Identity-Based Cryptosystems and Signature Schemes*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1985, pp. 47–53. [Online]. Available: https://doi.org/10.1007/3-540-39568-7_5

[36] D. Boneh and M. Franklin, *Identity-Based Encryption from the Weil Pairing*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 213–229. [Online]. Available: https://doi.org/10.1007/3-540-44647-8_13

[37] K. E. Lauter and K. E. Stange, "The elliptic curve discrete logarithm problem and equivalent hard problems for elliptic divisibility sequences," 2008.

[38] N. Koblitz and A. Menezes, *Pairing-Based Cryptography at High Security Levels*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 13–36. [Online]. Available: https://doi.org/10.1007/11586821_2

[39] R. M. Ham, G. Linden, and M. M. Appleyard, "The Evolving Role of Semiconductor Consortia in the United States and Japan," *California Mgmt. Rev.*, vol. 41, no. 1, p. 137, 1998. [Online]. Available: https://doi.org/10.2307/41165979

[40] D. Schwartz, N. Youngs, and A. Britto, "The Ripple Protocol Consensus Algorithm," https://ripple.com/files/ripple_consensus_whitepaper.pdf, 2014, [Online; accessed 09-Dec-2017].

[41] D. Mazières, "The Stellar Consensus Protocol: A Federated Model for Internet-level Consensus," https://www.stellar.org/papers/stellar-consensus-protocol.pdf, 2016, [Online; accessed 09-Dec-2017].

[42] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," https://bitcoin.org/bitcoin.pdf, 2009, [Online; accessed 09-Dec-2017].

[43] B. Colombier and L. Bossuet, "A survey of hardware protection of design data for integrated circuits and intellectual properties," vol. 8, pp. 274–287, 11 2014.

[44] D. Agrawal, S. Baktir, D. Karakoyunlu, P. Rohatgi, and B. Sunar, "Trojan Detection using IC Fingerprinting," in *2007 IEEE Symposium on Security and Privacy (SP '07)*, May 2007, pp. 296–310.

[45] Y. Zhou and D. Feng, "Side-channel attacks: Ten years after its publication and the impacts on cryptographic module security testing," 2005.

[46] L. Zhang, L. Vega, and M. Taylor, "Power Side Channels in Security ICs: Hardware Countermeasures," 05 2016.

[47] A. Miller, I. Bentov, R. Kumaresan, C. Cordi, and P. McCorry, "Sprites and State Channels: Payment Networks that Go Faster than Lightning," 2017.

[48] J. Poon and V. Buterin, "Plasma: Scalable Autonomous Smart Contracts," http://plasma.io/plasma.pdf, 2017, [Online; accessed 14-Jan-2018].

[49] J. Teutsch and C. R. ner, "A scalable verification solution for blockchains," https://people.cs.uchicago.edu/~teutsch/papers/truebit.pdf, 2017, [Online; accessed 14-Jan-2018].

[50] L. Luu, V. Narayanan, C. Zheng, K. Baweja, S. Gilbert, and P. Saxena, "A Secure Sharding Protocol For Open Blockchains," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. New York, NY, USA: ACM, 2016, pp. 17–30. [Online]. Available: http://doi.acm.org/10.1145/2976749.2978389

[51] U. W. Chohan, "The Cryptocurrency Tumblers: Risks, Legality and Oversight," *Social Science Research Network (SSRN)*, 2013.

[52] T. Ruffing, P. Moreno-Sanchez, and A. Kate, *CoinShuffle: Practical Decentralized Coin Mixing for Bitcoin*. Cham: Springer International Publishing, 2014, pp. 345–364. [Online]. Available: https://doi.org/10.1007/978-3-319-11212-1_20