http://dx.doi.org/10.14236/ewic/ICS2018.9

Optimizing Blockchain for data integrity in Cyber Physical Systems

Konstantinos Koumidis
KIOS Center of Excellence
Department of Electrical and Computer Engineering
University of Cyprus
kkoumi02@ucy.ac.cy

Panayiotis Kolios
KIOS Center of Excellence
Department of Electrical and Computer Engineering
University of Cyprus
pkolios@ucy.ac.cy

Christos Panayiotou
KIOS Center of Excellence
Department of Electrical and Computer Engineering
University of Cyprus
christosp@ucy.ac.cy

Securely maintaining log records for audit and accountability purposes is key for the proper operation of Cyber-Physical Systems. Thus, integrity of data used by Supervisory Control and Data acquisition (SCADA) components for monitoring and control functionalities must be ensured.
In this work, we consider a blockchain-based scheme for enhancing the integrity of measurements recorded in ledger blocks while taking into account particular application constraints within our problem formulation. Particularly, our formulation considers the real-time requirements of the monitoring and control functions and optimizes the blockchain computations for efficient resource utilization in order to deliver hard to tamper blocks of measurements. Performance analysis of the resulting mathematical programming solution is conducted through extensive simulation results.

Blockchain, Cyber-Physical Systems, Security, Data Integrity, Logging

1. INTRODUCTION

Cyber-Physical Systems (CPSs), are equipped with sensing, networking, and processing capabilities to reliably monitor and control physical processes. These systems underlay and impact a broad field of critical infrastructures including road transportation, water distribution, and the medical sector. Under normal operation, measurement information is communicated from sensors to local or remotely located Supervisory Control and Data Acquisition (SCADA) components which provide high level monitoring and control capabilities. A few examples of services offered by SCADA include online computations for estimating the state of the underlying physical system and offline computations for forensic audits.

Due to their ubiquitous nature, possible cyber-attacks pose a significant threat to the public, the infrastructure, and the environment. In recognition to this issue, guidelines for secure design and maintenance of CPSs have been developed recently in Ross (2016) while the report in Stouffer (2015) provides comprehensive guidelines on how to secure a specific class of CPS, the Industrial Control Systems (ICS) which is used to monitor and control industrial manufacturing processes. Some key recommendations are network segmentation and segregation to restrict access to sensitive information with the use of firewalls and routers, patch management strategies to address flaws in existing software, use of tools such as intrusion detection systems that analyze information to identify and isolate anomalies, and the need to maintain accurate logs of system behavior to protect against modifications. As indicated in Bellare (2003), Bellare (1997) intelligent attackers tamper logs to cover all traces of their malicious activities and to suppress any raised flags.

In this work we investigate how to further improve the integrity of logged measurement data for real-time monitoring and control. To date, authentication codes and digital signatures in symmetric and public key cryptographic are the main methods for verifying authenticity of both communicated and stored information. Under the assumption that cryptographic keys remain secret, an adversary is unable to perform any stealthy data modification. To circumvent this assumption, side channel Yongbin (2005) and insider attacks Stouffer (2015) have been used by adversaries to learn cryptographic keys.

In this work, a novel data authentication mechanism is considered that is complementary to traditional key authentication schemes. The proposed solution introduces a blockchain setup that takes as input measurement data directly from the sensors and computes blocks that contain batches of measurements entangled in a linked list. In doing so, it becomes computationally very hard for any adversary to modify logged data. These blocks are then accessible by SCADA components through the public ledger for monitoring and control purposes. In this way, the blockchain creates a tamper-proof log of measurements which is computationally hard to alter. Specifically, an adversary would need to compute hard cryptographic puzzles not just for the block containing the desired data to be altered but also solve the puzzles for all the subsequent blocks until the head of the list.

Importantly, as opposed to traditional blockchain solutions, measurement data in CPS comes with hard deadlines associated with the real-time constraints of monitoring and control functionalities. Hence computing, communicating and appending blocks of measurement records to the blockchain before a predefined time thresholds is critical for normal operation. To achieve this, several competing parameters of the whole process of computing, communicating and appending the blockchain records have properly tuned to address the real-time requirements utilizing the least number of resource. Our contribution is to model and optimize those parameters in order reduce the operational cost of the required resources while maintaining tamper-proof logs of measurements.

To do so, a mathematical program is formulated to solve the problem that emerges. The proposed formulation takes into account the desired security level that blockchain records should have, as defined by the system operator and derives the configuration parameters to minimize the number of utilized resources necessary to compute blocks of sensor measurements. As indicated above, the complete workflow is considered, including the time needed to compute the blocks, the time for those blocks to be communicated for validation and the time to be appended on the public ledger.

The rest of the paper is structured as follows. We review related work and explain in depth governed blockchains Lundbaek (2016) (which is a specific type of blockchain that we use hereafter) that is used in our proposed architecture in Section 2. The proposed blockchain architecture and the relevant problem definition are presented in 3. In 4 derives a mathematical programming formulation for optimizing the operational parameters of the blockchain. The performance of the optimized blockchain architecure is evaluated in Section 5 and section 6 provides concluding remarks.

2. BACKGROUND AND RELATED WORK

An encouraging technology to consider for enhancing data integrity is blockchain. It has drawn a lot of attention recently for enabling a set of entities to secure their in-between transactions in a decentralized manner. Blockchain was originally developed for participating nodes in Cryptocurrency networks such as Bitcoin Nakamoto (2008) to preserve a common financial state. It is one form of a distributed ledger technology which is consists of a database replicated across several entities, and maintained independently by each entity using an agreement protocol. Each created transaction is broadcast in the blockchain network but first is digitally signed to prevent any unauthorized changes and also for source identification. Transactions are bundled into blocks which are recorded and linked, forming a chain of blocks; the so-called blockchain. Blocks are authenticated according to a common agreed consensus mechanism to distributedly maintain a consistent state among entities in adding blocks to the existing chain.

The success around the blockchain technology triggered interest in finding innovative uses to a broad range of applications. The UK Government Office for Science published a report in Walport (2017) pointing out potential contribution of distributed ledgers to ensure integrity in system operation, secured telemetry transmission and resilient firmware distribution for over-the-air updates of IoT devices. A detailed review of blockchain technology and its applicability in the IoT sector along with existing applications and issues for researchers and developers is presented in Christidis (2016). Work in Nikitin (2017) presents a Decentralized software-update framework with multiple entities verify conformance

of software updates and collectively sign them to create a tamper-proof release log and eliminate single points of failure.

An auditable and distributed access control Blockchain-based design for securely storing and sharing IoT data streams is presented in Shafagh (2017). In Hashemi (2017) blockchain is used for persistent data storage with a publish-subscribe client access model where incoming traffic is filtered out by publishers based on a set of subscribed queries provided from clients. Only matching queries are communicated to the subscribers to minimize the processing load placed on the client. Such design gives to users the power to access and control their collections of data.

In the majority of blockchain solutions, the Proof of Work (PoW) mechanism is employed. PoW is a consensus mechanism initially prosposed in Dwork (1992) as an anti-spamming countermeasure for attaching computational cost in resource allocation requests. Specifically, in PoW the input data is associated with computational effort to generate a tuple of the output data coupled with a block header. PoW is an asymmetric scheme where computational work is moderately hard to make but easy for other devices to verify its correctness, so a piece of PoW data is computationally hard to forge. Computational work is obtained by iteratively evaluating the output of a cryptographic hash function (e.g. SHA-256) and at each iteration the input field (called a nonce) in the block header changes, while the rest of the input remains the same. The process of finding a valid nonce for a set of PoW configuration parameters is also known as mining and miners are those computing devices that solve a computationally hard PoW configuration Lundbaek (2016), Luu (2017). The mining process terminates when a value of the nonce is found that produces an output hash with a minimum number of leading bits to zero. Notably, having multiple miners evaluating the cryptographic hash function, each using different nonce values, speeds up the mining process.

In governed blockchains, the entire blockchain infrastructure is owned and controlled by individual organizations Lundbaek (2016). In this way the system (including the PoW mechanism) and its configuration parameters can be accurately modelled to optimize computations for conflicting objectives such as security, and operational cost. Miners operate in rounds where in each round the output hash for a block header evaluates the input data using a cryptographic hash function. The computation of the cryptographic hash function is modelled as a Random Oracle Model where the probability of success and failure for a particular input is random and independent in each round. Under these setting the PoW configuration parameters are the difficulty d (that is the minimum number of most significant bits output that must be zero for the nonce to be considered valid), the dimension r of search space of possible nonce values, and the number of miners s. Modelling the probability of success and failure for multiple miners is described by eq. (1) and (2), respectively.

$$P^s = (1 - 2^{-d})^{g*s}[1 - (1 - 2^{-d})^s] \tag{1}$$

$$\bar{P}^s = (1 - 2^{-d})^{s*(\psi+1)} \tag{2}$$

where eq. (2) expresses the probability of s independent events where each miner explores all search space of possible nonce values and fails to find one that validates input data. On the other hand, miners having g consecutive failures and success in round $g + 1$ where $0 \leq g \leq 2^r - 1$, is defined in eq. (1). The term $[1-(1-2^{-d})^s]$ is the complement of the event where no miner will succeed in a single round. The search space size for each miner is $\psi = \lfloor \frac{2^r}{s} \rfloor$.

Finding a valid nonce is a probabilistic process where the chances of successfully computing a valid nonce are higher when more rounds are evaluated (eq.1). In contrast, given the found nonce value to verify a PoW block only a single hash execution is needed. For probabilistic assurances in the behavior of the mining process, bounds for the possible events that occur during the mining process can be set with constraints in eq. (3) - (5). These constraints return feasible triples of PoW configuration parameters d, r, s which satisfy the probabilistic bounds. For the expressions below, let $y = (1 - 2^{-d})^s$. Then, constraint eq. (3) bounds the expected time to obtain a valid PoW block, where T is the time of a single round with s miners and the fractional term $E^s(noR) = \frac{1-y^{\psi+1}-(\psi+1)\cdot(1-y)\cdot y^{\psi+1}}{1-y}$ provides the expected number of rounds. Eq. (4) bounds by δ_2 the probability that more than one miner finds the PoW within the first θ seconds in the same synchronous race. This probability is derived by subtracting the probabilities that no miner and a single miner find the PoW block within θ seconds, i.e. $P_d^s(\theta) = 1-(1-2^{-d}\cdot[1-z])^{s-1}\cdot[1+(s-1)\cdot 2^{-d}\cdot[1-z]]$. Constraint eq. (5) bounded above with δ, limits the probability for the actual time to mine a block exceeding the threshold θ.

$$\tau_u \geq T \cdot E^s(noR) \geq \tau_l \tag{3}$$

$$\delta_2 \geq P_d^s(\theta) \tag{4}$$

$$\delta \geq P_c^s(PoWTime > \theta) = y^{\lceil(\theta/T)-1\rceil} - y^{\psi+1} \tag{5}$$

In addition to the PoW parameters, we also consider hereafter that all measurement data have time-to-leave (TTL) constraints within which data needs to be delivered to SCADA for monitoring and control actions and beyond that all data becomes obsolete. Thus, PoW configuration parameters (including the number of miners), must be decided for each block such that secured transactions containing the data are delivered on time. A detailed modelling setup is presented below while the problem formulation and its performance analysis are included in subsequent sections of this work.

3. SYSTEM MODEL

In this section, we describe the details of a PoW blockchain employed in a CPS system where data measurements have timing deadlines. Messages with measurement data are send from sensors to a blockchain computing infrastructure and mined using PoW to output blocks containing secured transactions that are then subsequently forwarded to SCADA components for logging and other real-time services. To aid understanding, a schematic of the proposed architecture is shown in fig. 1.

Each generated message includes a header and measurement data. In the header of each message, there is a time to live (TTL) field indicating the maximum delay to consider the measurement valid at the destination node. Aggregated messages are batched into blocks and for each block miners put computational effort to find a valid nonce, as explained in previous section, using the PoW. Once such a nonce is found, the particular block gets broadcast to the miners' network for validation, appended to the blockchain with immediate delivery to the destination nodes. Miners receiving a block verify that it has sufficient computational effort as defined in the difficulty field of the block header and the message hashes give the merkle root of the block header. The security level of the PoW (i.e., difficulty and associated probabilistic bounds) for mining blocks are defined explicitly by the system operators according to the desired levels of resilience.

We assume computing resources in the blockchain (i.e. miners) can be dynamic allocated based on demand. We also assume that the communication channels between source-destination pairs have limited data rate so the number of messages per block can not be arbitrarily large since communication times would increase proportionally.

Specifically, we assume there are n of sensor nodes and l SCADA components. Let $v \in V$ be the set of sensor nodes, $u \in U$ the set of destination SCADA components, where $n = |V|$, $l = |U|$ and $j \in J$ the set of messages that encapsulate measurement data. Then a message send from sensor node v through the blockchain to destination node u, has the following set of attributes:

- A time to live (TTL), t_j
- Message size, B_j
- The data rate between v and u, $R_{v,u}$

Using this setup, the problem that arises is how to decide on the computing resources to be committed for computing blocks containing batches of measurements that are to be appended to a secured blockchain and presented to SCADA components on time. The key decision variables of this problem include the number of miners to be committed, which messages should be selected for each block based on their deadlines and how many blocks to generate based on the available messages in the pool. Each combination of parameters provides different probabilistic guarantees (that have to exceed some threshold) for the event of successfully mining blocks within the set time constraints in order to deliver messages to destination nodes according to their TTL attributes. Moreover, communication channels have finite capacity so network delays for communicating blocks to SCADA components must be taken into account as well. Indicatively, including more messages in a block increases throughput of messages communicated but also increases transmission times that make it harder for blocks to reach destination nodes within their indicated TTL constraints. Thus, assigning messages into blocks must be done is such a way as to prevent any TTL violations.

3.1. Security Assessment

As detailed above, the system model does not consider any detection mechanism to indicate if received measurement data has been tampered before reaching the blockchain infrastructure. Instead this work considers an adversary who attempts to generate new blocks and insert those blocks into blockchain to alter logged data. In order to produce blocks faster than honest miners, an adversary must retain sufficient computational power forcing the adoption of an non-legal path in the blockchain. The most well known attack of PoW mechanism is when an adversary is in poses more than 51% of the total computational power Nakamoto (2008). The work in Ittay (2014) proposes a scheme which limits the amount of computational work that can be outsourced to external miners in public blockchains where anyone can participate. In this scheme, miners of a pool must cryptographically sign each computed hash with a

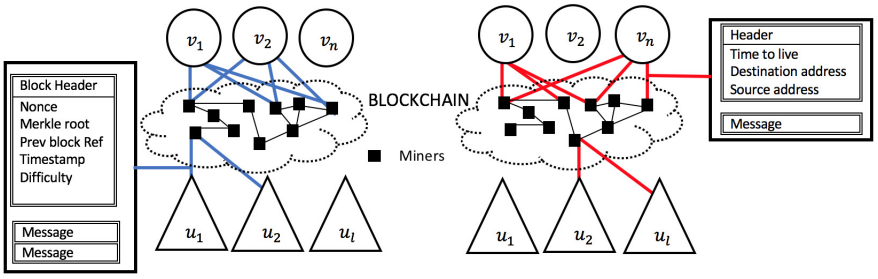

Figure 1: *Blockchain architecture integrated in CPS process.*

private key shared only among them. When governed blockchains are employed each miner must explore a search space for nonce values over a specific range. Identifying the source of a valid PoW block in the same way as in in Ittay (2014) but with each miner using a unique secret key instead of a shared, prevents adversary from creating valid PoW blocks without possession of private key. Moreover, in governed blockchains for an adversary who has in his possession l out of total s miners the chances of mining successfully c consecutive PoW blocks is $(l/s)^c$. In addition, the event where more than one miners find a PoW block within a certain time interval has direct impact on the blockchain's security since the longest PoW chain can not be determined. The probability of such an event occuring is eq. (4) and can be controlled by raising the difficulty level as the number of miners increases, preserving at the same time the highest level of security offered by the blockchain technology.

4. PROBLEM FORMULATION

To increase the throughput of authenticated measurements multiple such messages can be grouped in every block. However, the size of each block is limited by the channel's capacity and the TTL constraints of messages included.

Let t_j^k be the TTL of message $j \in J$ indexed with time $k \in K = 0, .., T$. At time $k = 0$ the pool contains all messages received up to the current time while for $k > 0$ predictions can be made on messages that are expected to arrive in the future. Messages having the same time index k are consider to arrive in batches. Also let the arrival rate of each message batch be set by $\lambda(t)$.

Then the number of miners assigned to block $i \in I$ containing messages received with time index k is determined by s_i^k. Also decision variable $x_{ij}^k \in \{0, 1\}$ indicates that block i contains message j at time epoch k. Since all messages with $k > 0$ are forecasted measurements, they are not included in any of the blocks generated for $k = 0$. Instead forecasted measurements (and their associated blocks) are used to make informed decisions on the resources to be committed at $k = 0$ to compute blocks on time to meet their deadlines. The time which block i has to be successfully mined is θ_i^k. For message $j \in J$ with source and destination nodes v and u, the propagation time becomes $r_{ij} = \frac{B_j}{R_{v,u}}$ where $R_{v,u}$ is the data rate and B_j^k the block size containing the particular measurement data.

Considering all the non-linear constraints that arise (as detailed in eq. (3)-(5)) in a single problem formulation is very hard to solve in practice, especially in the presence of terms with exponents of large magnitude that arise for higher difficulty targets. Hence, an alternative formulation is considered hereafter where non-linear equations are transformed into a piecewise linear objective function. Specifically, since PoW configuration parameters such as the level of difficulty d, dimension of search space r and probabilistic bounds of constraints in eq. (3)- (5) are all defined by operators then the minimum required number of miners to find a valid nonce before time θ_i^k and satisfy the non linear constraints can be found by enumeration. Thus, the non linear relationship between the minimum required number of miners s_i^k to successfully mine a PoW block $i \in I$ before θ_i^k is computed when θ_i is defined over a finitely long horizon to obtain piecewise linear function $s_i^k = f(\theta_i^k)$.

The plot in Fig. 2 displays the relationship between the minimum required number of miners to mine a PoW block before a certain time threshold which is used at the linear piecewise objective function $s_i^k = f(\theta_i)$ in formulation (P1). The relationship is evaluated with $d = 40$, $r = 70$ and for probabilistic bounds of different orders of magnitude. Specifically, we have probabilistic bounds $\delta = \delta_2$ that take values of 10^{-5},10^{-8} and 10^{-11}, respectively. Evidently, there is an exponential relationship between the two terms and for stricter probabilistic bounds the steeper is the curve with higher amount of needed miners. Moreover, mining blocks with the same PoW

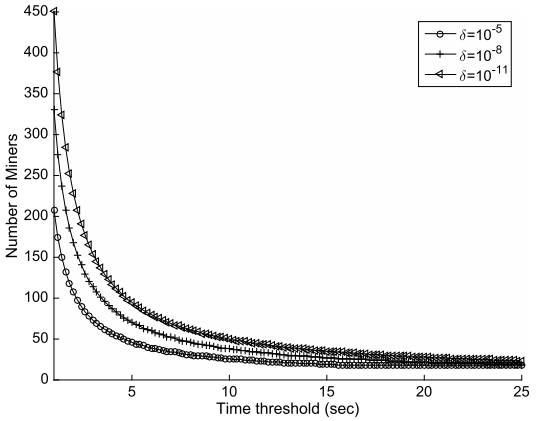

Figure 2: *Piecewise linear objective function $f(\theta_i^k)$ evaluated for different probabilistic bounds*

configuration parameters fewer miners are requried when blocks are mined for longer time periods since less miners have more time to achieve the same amount of computational effort. Finally, in all cases the demand in the number of miners when the threshold is larger than 20 seconds converges and becomes constant.

From the aforementioned parameters the following mathematical programming formulation is derived to minimize the number of miners over a time horizon k.

$$(P1)\min \sum_{k\in K}\sum_{i\in I} f(\theta_i^k) \tag{6}$$

$$\text{s.t.} \sum_{j\in J} B_j^k \cdot r_{ij}^k + \sum_{w=1}^{k}\sum_{q=1}^{i}\theta_q^w \leq (M - t_j^k)x_{ij}^k + M,\ \forall j\in J, i\in I, k\in K \tag{7}$$

$$\sum_{i\in I} x_{ij}^k = 1,\ \forall j\in J,\ k\in K \tag{8}$$

$$\sum_{j\in J} R_{ij}^k \cdot x_{ij}^k \geq \gamma_i\ ,\ \forall i\in I,\ k\in K \tag{9}$$

$$\gamma_i \geq R_{ij}^k \cdot x_{ij}^k\ ,\ \forall i\in I,\ j\in J,\ k\in K \tag{10}$$

$$\overline{R_{ij}^k} \cdot x_{ij}^k \geq r_{ij}^k\ ,\ \forall i\in I,\ j\in J,\ k\in K \tag{11}$$

$$r_{ij}^k \geq \gamma_i - \overline{R_{ij}^k}(1 - x_{ij}^k)\ ,\ \forall i\in I,\ j\in J,\ k\in K \tag{12}$$

$$\theta_{i+1}^k \geq \theta_i^k,\ \forall i\in I,\ k\in K \tag{13}$$

$$x_{ij}^k \in \{0,1\}\ \ \forall j\in J,\ \ \forall i\in I,\ k\in K \tag{14}$$

$$\exists\, \theta_i^k \geq 1 \iff \sum_{j\in J} x_{ij}^k \geq 1,\ \forall i\in I,\ k\in K \tag{15}$$

The objective function minimizes the total operational cost in terms of the total number of utilized miners. The piecewise linear function describes the non-linear relationship between variables θ, s with the desired probabilistic guarantees, where selecting θ_i^k as the time needed to authenticate a block i with time index k provides the minimum number of miners s_i^k needed for mining the particular block according to (3)-(5). This is directly translated to the operational cost of successfully authenticating blocks.

Constrain eq. (7) ensures that the computation and communication time is limited by the shortest TTL of all messages in block i. It must hold for all possible message combinations in all blocks and thus the large constant M is used. The term $\sum_{w=1}^{k}\sum_{q=1}^{i}\theta_q^w$ is the time needed to mine block $i \in I$ and all blocks before i. Finally, the term $B_j^k \cdot r_{ij}^k$ is the time it takes for a block of size $\sum_{j\in J} B_j^k \cdot x_{ij}^k$ to reach the destination node that has the lowest bandwidth between the recipients of the particular block. The rest of the constraints eq. (8 - 15) hold for all messages both actual and predicted with time index $k \in K = 0,..,T$. Each received message $j \in J$ must be included only in one block and no two messages with different time index k are to be included in the same block as indicated by constrain eq. (8). Equation (9 and (10) return γ_i which is the multiplicative inverse of the minimum channel bit-rate of the path for all messages in each block. For each message, the term r_{ij}^k is equal to γ_i if message $j \in J$ belongs to block $i \in I$ otherwise is zero as dictated by eq. (11) and (12). Due to the problem nature messages with small TTL values are included into blocks before messages with larger TTL values, so earliest mined blocks will be mined for shorter times compared to blocks mined later. Thus, eq. (13) ensures that blocks to be mined later to at least have as much computation time as their predecessors. In addition, it reduces the search space of possible message/block combinations by eliminating symmetries from redundant solutions. The binary and continuous variables are defined in eq. (14) and (15), respectively. The condition in (15) prevents allocation of any miners in blocks with no messages while the minimum mining time of 1 second is considered for blocks containing at least one message.

5. EXPERIMENTAL EVALUATION

In this section, we evaluate the performance of proposed formulation to run a PoW blockchain. Utilization performance is measured with the amount of miners required to obtain PoW blocks given the time constraints set by messages. First, the way messages are chosen to be mined with PoW blocks is explored for a single time step. We then investigate the performance of the proposed formulation over time where a forecast is made for future messages to

arrive to PoW blockchain. A comparison is made with the case when only the actual received messages are considered by the formulation for arrangement of messages into blocks.

For the simulations we define blocks that contain at least one message to have minimum mining duration 1 sec. The size attribute for messages is generated uniformly random between 20 bytes and 252 bytes where the value of lower bound was taken from a laboratory-scale water storage tank system data set Morris (2014), Morris (2011) whereas the maximum message size is the maximum payload size of the Modbus communication protocol. The hash of mined blocks must have at least 40 leftmost bits to zero ($d = 40$) and size is search space 2^r where $r = 70$. The bounds of constraint (eq.3) are $\tau_l = 0$, $\tau_l = 300$ and for constraints (eq.4,5) to δ = $\delta_2 = 10^5$. To obtain the results for every scenario, 50 Monte Carlo iterations are performed.

Having the aforementioned parameters we first evaluate proposed formulation (P1) for a single time step in order to verify its correctness and get a better understanding how messages are grouped into PoW blocks. An input dataset of 50 messages with the maximum tolerable delay of messages reaching destination to $\overline{t_j^k} = 30, \ \forall \ j \in J, \ k \in K$ is considered. Thus, each message is assigned a randomly generated numerical value for TTL attribute within the interval of 1 - 30 seconds. The channel bit rate is constant for all network links to 5Mbps.

Compact box plot in fig. 3b shows the number of messages bundled in consecutive blocks and the computing time θ_i^k these blocks have to be successfully mined can be seen in fig.3a. As blocks with lower numerical index are mined for shorter time fig.3a, they include messages with shorter TTL numerical attribute. On the other hand, blocks mined for longer time without violating any time constrains contain messages with high Time to live (TTL) numerical value and pack more messages fig.3b. Moreover, blocks to be computed for longer time require fewer miners fig.2. As seen in the following set of simulations where formulation is evaluated over time, computin blocks for extended amount of time may impact system performance in delivering messages on time before as specified with TTL.

For subsequent set of simulations we consider 5 batches of messages arrive with rate $\lambda(t)$ to blockchain to be included into PoW blocks. The number of messages in each batch is randomly decided with maximum 30 messages per batch and minimum 10. Moreover, the maximum delay for TTL in a message is $\overline{t_j^k} = 25, \ \forall \ j \in J, \ k \in K$. Once miners start computing a particular block the process can not be interrupted until a valid nonce is found or entire search space is explored. Thus, received messages can be computed into blocks when the ongoing mining process terminates. The bit rate of network links is between 5Mbps and 10Mbps. We compare the performance when predicted datasets of messages taken into account by the formulation against the case where no forecast is made and only the actual received messages are considered. Specifically, a forecast of two consecutive batches $k = 2$ is made with fixed number of messages. In this case, the PoW blocks with predicted messages are never computed. However, they affect the computing time of blocks that contain the actual received messages.

We evaluate the formulation for three different arrival rates. Specifically, arrival time between consecutive batches is generated uniformly random within time interval of 15, 25 and 30 seconds. Fig.4a shows the amount of iterations where no violation of the time constraints occurred in delivering messages to destination nodes. Feasible Iterations are more than double when forecast of future messages is taken into account. In contrast, when no prediction is considered it more likely to fail delivering on time at destination the messages with small numerical value for TTL attribute .

The box plot in Fig. 5a displays the distribution for the mean number of blocks to be mined per time epoch in each iteration. The average time a single block has for computation and the number of miners are shown in Fig.5b and Fig.5c, respectively. The distributions on the left hand side are for the case when formulation considers only the actual received messages with $k = 0$ and on the right hand side when prediction is considered. Only results from feasible iterations where every message is successfully delivered within the specified TTL are shown in plots fig.5-7. When predicted messages are taken into account by the formulation fewer PoW blocks need to be mined overall to include all received messages Fig. 5a. In contrast, when no prediction is considered the number of blocks needed to deliver messages to destination nodes within the demanded Time to live increases. Blocks have on average have one second less for computation time in the second set-up fig.5b and the number of miners employed when taking into account future incoming messages in the formulation is higher. This is due to the higher number of input datasets that have a feasible assignment of messages into blocks and blocks having shorter available time for computation. For fig.6 where $\lambda(t) = 25$ the number feasible iterations increases and the mean number of blocks created

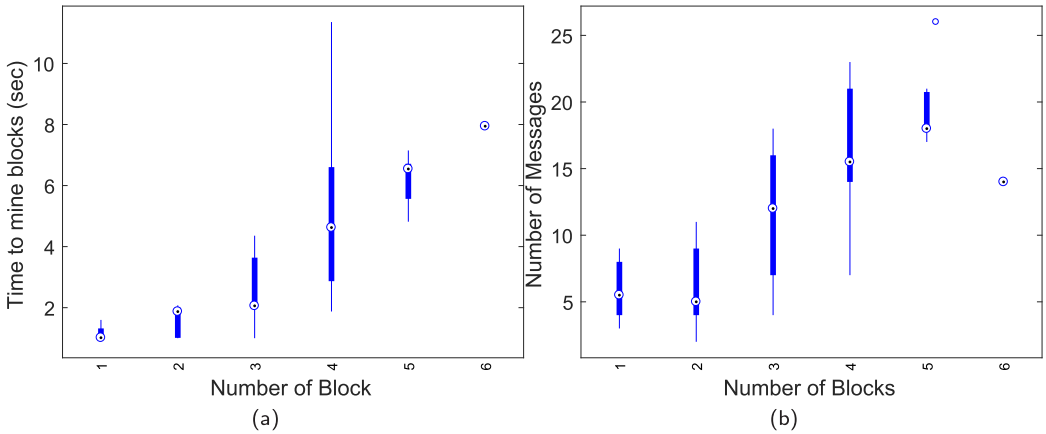

Figure 3: *Number of messages in blocks and time to compute blocks for* $\delta = \delta_2 = 10^{-5}$, $\overline{t_j^k} = 30$ *and* $R = 5Mbps$

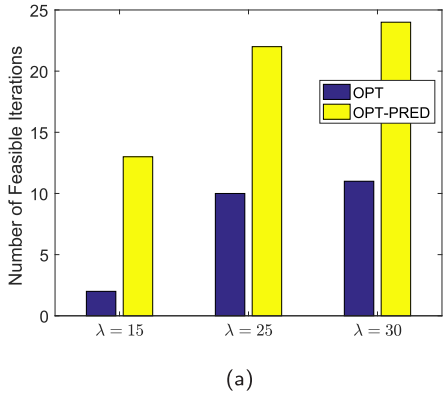

Figure 4: *Feasible Iterations where no violation occurs in the time constraints of messages*

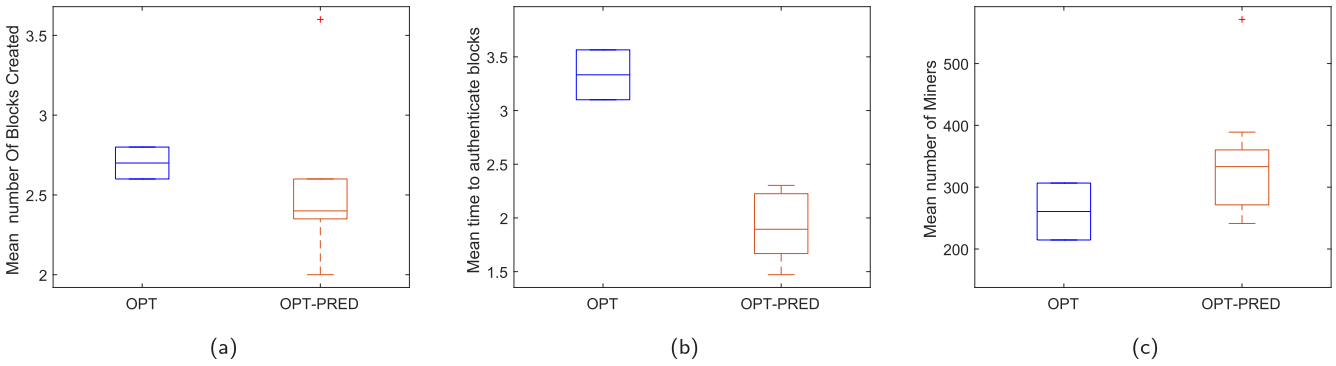

Figure 5: *For* $\lambda(t) = 15$

in the two set-ups is the same. Once messages with small TTL numerical value arrive to the system while a block of measurements is currently mined it will result to delay in computing the newly received messages. This event is more likely to occur when arrival time between consecutive batches to the blockchain is shorter as formulation prolongs the duration of mining blocks fig.3a, so blocks with newly received messages must have low computation time. Consequently, in fig.6b blocks have on average more time for computation in comparison to fig.5b. Moreover, fewer resources are required when $\lambda(t) = 25$ while the utilization of miners is higher as blocks have approximately 1 second less for computation with the second set-up. In cases where the available computation time due to considering prediction is decreased as seen in fig.5,6 it is less resource intensive to mine blocks with higher number of messages. For the third case when $\lambda(t) = 30$ the computation time of blocks in the second set-up is

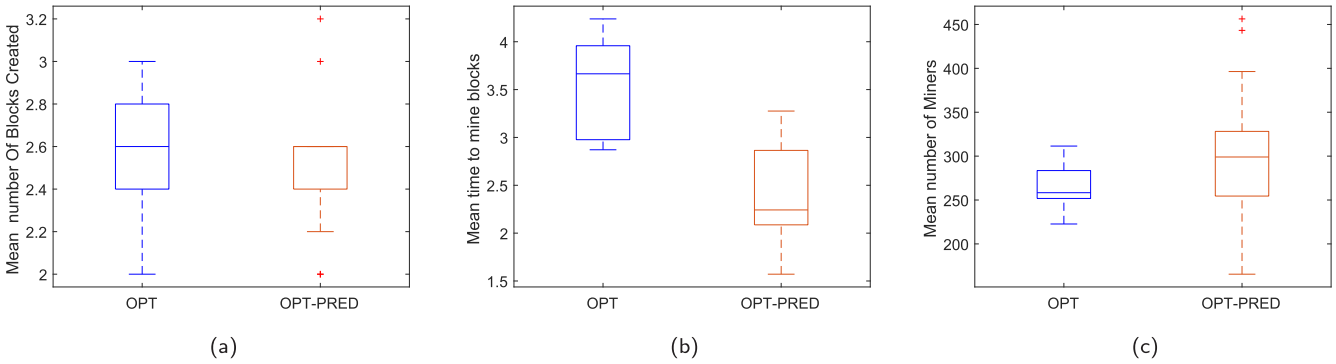

Figure 6: *For* $\lambda(t) = 25$

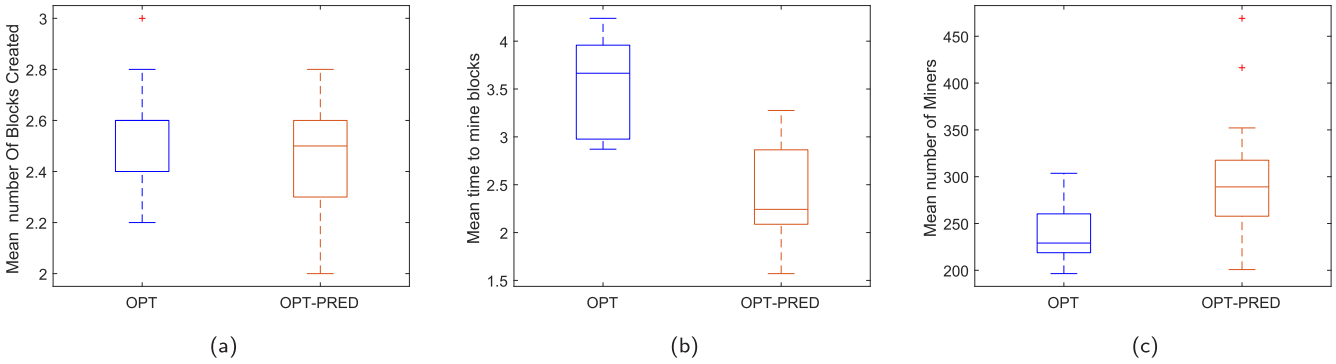

Figure 7: *For* $\lambda(t) = 30$

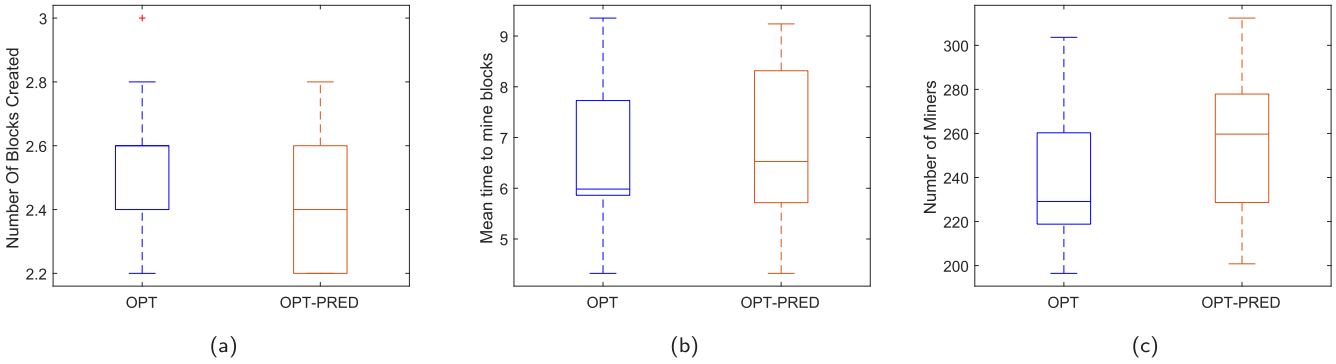

Figure 8: *Only feasible iterations for both set-ups when* $\lambda(t) = 30$

about the same or slightly less compared to first set-up. Fig.7a shows minor deferences on the number of blocks needed to be mined to include all messages. Thus, a similar pattern is observed among fig.6c and fig.7c but few dozen miners less are required when $\lambda(t) = 30$ as blocks have slightly more time in order to be computed fig.7b.

Finally, in 8 shows only the results from iterations that were feasible by both set-ups when $\lambda(t) = 30$. While similar patterns as in fig.7 are obvious, the variance between the two corresponding distributions is about the same. Its important to point out resource over provisioning when considering prediction of future messages has low overhead in resource utilization due to miss-prediction. However, it greatly improves the chances of delivering messages in blocks within the specified TTL constraints.

6. CONCLUSIONS

In this paper we use blockchain for enhancing log integrity where we consider including messages of sensor measurements in authenticated blocks delivered to SCADA. Consequently, it becomes computationally hard for adversary to create valid blocks. We optimize the allocation of resources for computing

blocks in PoW consensus mechanism while respecting time constraints set by messages. Proposed formulation considers forecast of messages for provisioning future arrivals over time. From simulations we show the resource utilization overhead when taking into account prediction of future messages to arrive in blockchain. For future work we will investigate incoming messages with uncertain arrival rates and find optimal PoW configuration parameters when uncertainty is defined over a deterministic set.

ACKNOWLEDGEMENTS

Funded by the European Unions Horizon 2020 research and innovation programme under grant agreement No. 739551 (KIOS CoE).

REFERENCES

Zhou Yongbin and Feng, DengGuo (2005) *Side-Channel Attacks: Ten Years After Its Publication and the Impacts on Cryptographic Module Security Testing* IACR Cryptology ePrint Archive.

K. Stouffer, V. Pillitteri, S. Lightman, M. Abrams, and A. Hahn (2015) *NIST Special Publication 800-82: Guide to Industrial Control Systems (ICS) Security*.

Hashemi, Sayed Hadi, Faraz Faghri, and Roy H. Campbell (2017) *Decentralized User-Centric Access Control using PubSub over Blockchain* abs/1710.00110.

Hossein Shafagh, Lukas Burkhalter, Anwar Hithnawi, and Simon Duquennoy (2017) *Towards Blockchain-based Auditable Storage and Sharing of IoT Data.* Proceedings of the 2017 on Cloud Computing Security Workshop.

K. Nikitin, E. Kokoris-Kogias, P. Jovanovic, L. Gasser, N. Gailly, N., I. Khoffi,, J. Cappos, B. (2017) *Proactive Software-Update Transparency via Collectively Signed Skipchains and Verified Builds.* 26th USENIX Security Symposium.

Ittay Eyal and Emin Gn Sirery (2014) *How to Disincentivize Large Bitcoin Mining Pools.* http://hackingdistributed.com/2014/06/18/how-to-disincentivize-large-bitcoin-mining-pools/

S. Nakamoto (2008) *Bitcoin: A Peer-to-Peer Electronic Cash System", May 2008.*

C. Dwork and M. Naor. (1992) *ricing via processing or combatting junk mail.* Lecture Notes in Computer Science No. 740.

M. Walport (2017) *Distributed ledger technology: Beyond block chain.* UK Government Office for Science.

K. Christidis and M. Devetsikiotis, (2016) *Blockchains and Smart Contracts for the Internet of Things.* IEEE Access.

Ross (2016) *NIST SP 800-160 systems security engineering: Considerations for a multidisciplinary approach in the engineering of trustworthy secure systems.* Nat. Inst. Standards Technol., US Dept. Commerce, Gaithersburg, MD, US

L. Luu, Y. Velner, J. Teutsch, and P. Saxena (2017) *SMART POOL : Practical Decentralized Pooled Mining.* IIACR Cryptology ePrint Archive.

L.-N. Lundbaek, A. C. D'Iddio, and M. Huth (2016) *Optimizing Governed Blockchains for Financial Process Authentications.* http://arxiv.org/abs/1612.00407.

Morris, T., Gao, W.(2014)] *Industrial Control System Network Traffic Data sets to Facilitate Intrusion Detection System Research.* Critical Infrastructure Protection VIII

Morris, T. Srivastava, A., Reaves, B., Gao, W., Pavurapu, K., Reddi, R.(2011) *A Control System Testbed to Validate Critical Infrastructure Protection Concepts.* International Journal of Critical Infrastructure Protection.

M. Bellare and B. Yee .(2003) *Forward-security in private-key cryptography.* Proc. Topics Cryptol

M. Bellare and B. Yee .(1997) *Forward-security in private-key cryptography.*