2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)

SmartCrowd: Decentralized and Automated Incentives for Distributed IoT System Detection

Bo Wu[*§], Ke Xu[*§], Qi Li[*§], Zhuotao Liu[†], Yih-Chun Hu[†], Zhichao Zhang[*§],
Xinle Du[*§], Bingyang Liu[‡], Shoushou Ren[‡]
[*]Tsinghua University [†]University of Illinois Urbana-Champaign [‡]Huawei Technologies
[§]Beijing National Research Center for Information Science and Technology (BNRist)
Emails: {wub14@mails, xuke@, qi.li@sz, zc-zhang17@mails, dxl18@mails}.tsinghua.edu.cn,
{zliu48, yihchun}@illinois.edu, {liubingyang, renshoushou}@huawei.com

***Abstract*—Internet of Things (IoT) devices achieve the rapid development and have been widely deployed recently. Meanwhile, inherent vulnerabilities of IoT systems (including firmware and software) have been continually uncovered and thus the systems are always exposed to various attacks. The root cause of the issue is that IoT systems always have design flaws and implementation bugs. In particular, the released systems (e.g., by third-party marketplaces and IoT vendors) may be maliciously repackaged with malware. Unfortunately, IoT consumers are not able to effectively capture such vulnerabilities because of the limited detection capabilities. In this paper, we propose SmartCrowd, a blockchain-based platform that aims to outsource security detection of IoT systems to distributed detectors with strong detection incentives. SmartCrowd enables built-in accountability for IoT providers and authoritative references of detection results for IoT consumers. By building smart contracts, we can incentivize the efficient and high-coverage security detection of IoT systems, while providing decentralized and automated incentives for both IoT providers releasing secure IoT systems and detectors uncovering vulnerabilities. We present the security and theoretical analysis that demonstrates the security of SmartCrowd and the incentives for participators. We prototype SmartCrowd by using Ethereum and the experimental results show that SmartCrowd has both technical feasibility and financial benefits, which can be applied to build a secure IoT ecosystem.**

***Index Terms*—Incentives, Blockchain, Decentralization, Automation, IoT System Detection**

I. Introduction

Internet of Things (IoT) is widely deployed in recent years. According to the IDC report [1], the number of IoT devices will reach 28.1 billion by 2020. However, most IoT systems [1] are not securely designed and implemented, or repackaged with malware created by adversaries, making them exposed to various attacks. For instance, a large number of unsecured IoT devices are exploited, e.g., by the Mirai botnet, to launch distributed denial-of-service (DDoS) attacks against various Internet services, e.g., Dyn [2]. In order to fix the covered vulnerabilities, IoT system providers including IoT vendors and IoT software developers always upgrade their systems. However, the newly released systems might still introduce new vulnerabilities. For example, the systems may be repackaged with a malware by a compromised IoT provider.

TABLE I
The detection results of two IoT Apps performed by different third-party services are partially overlapped.

Third-Party Services	Samsung Connect			Samsung Smart Home		
	High	Medium	Low	High	Medium	Low
VirusTotal [3]	0	0	0	0	0	0
Quixxi [4]	4	6	3	3	8	4
Andrototal [9]	0	0	0	0	0	0
jaq.alibaba [10]	1	14	32	21	46	55
Ostorlab [11]	0	2	0	0	2	2
htbridge [12]	1	6	5	1	4	6

Here, ‘Hign’, ‘Medium’ and ‘Low’ denote the amount of high-, medium- and low-risk vulnerabilities, respectively.

Unfortunately, IoT consumers cannot easily identify vulnerabilities in these IoT systems due to the constrained resource or limited detection capability. Although several centralized third-party detection services (e.g., VirusTotal [3], Quixxi [4]) are available for IoT device security detection, their detection capabilities vary so greatly that their detection results are often different and non-overlapping, yielding inconsistent and incomplete reference for IoT consumers to protect their systems. For instance, as shown in Table I, the detection results for two IoT Apps (`Samsung Connect` [5] and `Samsung Smart Home` [6]) in Google Play generated by several popular centralized detection services share very limited commonality. Besides those centralized services, several decentralized solutions have been proposed. For instance, CloudAV [7] introduces N-version protection by enabling multiple endhosts to perform virus detection in parallel; Vigilante [8] provides end-to-end collaborative detection for Internet worm containment. These methods are essentially outsourcing-based solutions that rely on distributed detectors (i.e., endhosts) to achieve complementary detection. However, these solutions fail to provide incentives for attracting detectors' participation, which is impractical since security detection typically incurs non-trivial overhead. Additionally, none of the prior solutions consider *accountability*: IoT providers can still publish arbitrary software without taking responsibilities of releasing vulnerable IoT systems. Such lack of enforceable accountability fundamentally prevents these solutions from deterring malicious IoT providers.

This paper aims to answer the following question: *is it possible to build a platform that can attract different de-*

[1] In this paper, ‘IoT systems’ refers to the firmware and software of IoT devices instead of networks, which can be interchangeable with ‘IoT devices’.

2575-8411/19/$31.00 ©2019 IEEE
DOI 10.1109/ICDCS.2019.00112

tectors to participate in IoT system vulnerability detection and ultimately build a secure IoT ecosystem? The answer is positive. Our high-level idea is inspired by crowdsourcing platforms, which enable packagers to outsource their tasks to different workers, and the workers earn rewards by answering the tasks [13]. However, in the context of IoT ecosystem that is composed of many independent IoT providers, building a crowdsourcing platform is non-trivial due to the lack of a centralized packager that is trustable to properly allocate incentives. Ideally, the platform should provide following three desirable features. First, detectors that discover IoT system vulnerabilities automatically earn rewards, motivating all entities with detection abilities to participate. Second, the platform is capable of holding IoT providers accountable for releasing any vulnerable IoT systems. Such built-in accountability not only deters untrustworthy IoT providers from releasing buggy software, but also ensuring well-behaved IoT providers can receive proper rewards for releasing secure software. We believe that accountability is an ultimate driving force to create the more secure IoT ecosystem. Finally, the platform as a whole provides authoritative references to IoT consumers, minimizing their chances of deploying insecure IoT systems.

In this paper, we propose SmartCrowd, a blockchain-powered vulnerability detection platform that achieves the above three critical features. First, leveraging the "nonstoppable" nature of smart contracts, SmartCrowd ensures that detectors are rewarded automatically once their detection results are accepted, without relying on a centralized authority to allocate incentives. Additionally, since all vulnerability detection results are recorded in SmartCrowd's blockchain, SmartCrowd holds IoT providers accountable for any of their released IoT systems. Such built-in accountability effectively deters untrustworthy IoT providers from releasing vulnerable IoT systems meanwhile providing incentives for IoT providers to release more secure IoT systems, which eventually benefits the entire ecosystem. Finally, SmartCrowd's blockchain provides an authoritative, complete and consistent reference for IoT system vulnerabilities, allowing IoT consumers to better understand any possible security issues of the IoT systems that they are about to deploy.

We prototype SmartCrowd based on Ethereum [14], and use the experiment results to evaluate the performance of SmartCrowd. By evaluating the financial incentives, we find that SmartCrowd is also economically sound for IoT providers and detectors. Moreover, the security and theoretical analysis shows SmartCrowd achieves several desirable security goals by decentralized and automated incentives.

In summary, the contributions of the paper are as follows:

- We propose SmartCrowd, a blockchain-powered, decentralized vulnerability detection platform for IoT systems, which offers three critical features: strong detection incentives, built-in accountability and authoritative reference of detection results.
- We present the incentive scheme in SmartCrowd that is fully automated to incentivize IoT providers for releasing more secure IoT systems, and detectors for vulnerability detection, without relying on any centralized authority.
- We conduct security and theoretical analysis to show that SmartCrowd achieves ensured security goals while introducing expected incentives in an untrusted IoT ecosystem.
- We perform experimental evaluations to demonstrate SmartCrowd's feasibility and financial benefits.

The remainder of this paper is organized as follows. In Section II, the blockchain technology is briefly reviewed. We present the problem statement in Section III. The overview and design details of SmartCrowd are respectively introduced in Section IV and V. Security and theoretical analysis is performed in Section VI. Section VII evaluates the performance of SmartCrowd. We discuss the related work and conclude our paper in Section IX and X, respectively.

II. Background of Blockchain

In this section, we briefly review the blockchain technology that is commonly used in cryptocurrencies, e.g., Bitcoin [15]. We present the basic design principles, its consensus schemes and smart contracts, which are all used in SmartCrowd.

The blockchain is essentially a public decentralized ledger (or database) that is established and maintained by multiple distributed peers. Different blocks are linked one by one to construct the blockchain, and each block records several transactions that have been conducted in a distributed system. The blockchain is determined by the majority of participants while the minority, e.g., a few unreliable entities, would not affect this ledger. Once one block is confirmed, the transactions in it would never be tampered and could be publicly inquired by anyone at anytime. The blockchain is totally decentralized and no longer needs the trusted authority to validate each transaction. The blockchain is based on such cryptography that the real address is not used as an identifier. Instead, the address in the blockchain is generated using cryptographic algorithms (e.g., SHA-256 [16] and RIPEMD-160 [17]) to ensure its privacy and anonymity.

The consensus scheme is commonly used in the blockchain for enabling overall system consistent and reliable while facing a number of misbehaved participants. Many consensus algorithms (e.g., PoW [15], PoS [14], PBFT [18]) are employed for generating a block and maintain the consistency of the entire blockchian. Note that proof of work (PoW) is most commonly used in current blockchain system (e.g., Bitcoin network), where participants try to handle a cryptographic proof-of-work issue. Concretely, participants attempt to find a random number that will be used to make the hash of an entire block meet some requirements, which is related to the computing capability of participants. The consensus made by PoW can be easily verified by others that only require to perform a hash calculation.

Ethereum [14] is a blockchain-based technology that actually is a decentralized virtual machine [19]. In Ethereum, smart contracts are the terms used to describe computer program code that can facilitate and enforce the negotiation of an agreement (i.e., a contract) using the blockchain technology.

They are written in a Turing-complete bytecode language (called EVM bytecode), and can be carried out automatically once some events happen and trigger the contracts. Smart contracts also support distributed system without relying on a centralized authority to handle these contracts.

III. Problem Statement

A. Adversary Model

In this paper, we consider vulnerable IoT systems that may be buggy systems provided by benign IoT providers (including IoT device vendors, or platform and application developers) or malicious systems repackaged by a malicious third-party marketplace. We consider the IoT provider is misbehaved if it releases vulnerable IoT systems or systems repackaged with malware. By leveraging the vulnerabilities in IoT systems, the adversary can control the IoT devices, and the compromised IoT devices can leak private information or be exploited to launch attacks (e.g., DDoS attack [20]).

Facing an untrusted IoT ecosystem, detectors can also be compromised, especially when there are incentives for IoT detections. The malicious detectors can attempt to outplay IoT security detection, for example by trying to earn incentives without doing actual work. The detector can i) simply declare a forged detection report without even having detected the IoT system, or ii) plagiarize detection results of benign detectors. Meanwhile, the compromised detector can also attempt to accuse other detectors to have performed an incorrect detection by tampering their detection reports. This can directly prevent benign detectors from obtaining the allocated incentives.

B. Desired Properties

By enabling automated incentive allocation, SmartCrowd platform can attract more detectors to detect existing IoT systems and advise IoT consumers to securely employ them, which can effectively mitigate the harm caused by vulnerable IoT systems. In particular, to achieve the above goal, SmartCrowd should have the following design properties:

Incentives for IoT detection. Detectors who discover and report any vulnerability of IoT systems should be able to gain a reward. Meanwhile, IoT providers who are reported to release a vulnerable IoT system should be punished.

Decentralized process. The incentives of IoT security analysis should be decentralized for avoiding the dependence on any centralized authorities, which can prevent attacks constructed by a compromised or misconfigured authority.

Automated allocation. The platform should automatically allocate incentives for benign detectors and punish misbehaved IoT providers once any vulnerability is reported in the released IoT systems. It can allow IoT consumers to understand the risk of deploying a released IoT system instantly.

IV. SmartCrowd Overview

In this section, we present an overview of SmartCrowd platform that is an IoT system publishing infrastructure with strong detection capabilities. In particular, it enables decentralized and automated incentives for system security detection.

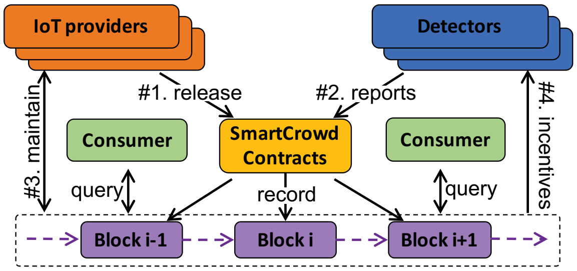


Fig. 1. The overview of SmartCrowd platform that introduces the decentralized and automated incentives by using blockchain and smart contracts.

A. Architecture

We leverage the blockchain for crowdsourcing new released IoT systems to distributed detectors while ensuring IoT providers are held accountable for releasing vulnerable IoT systems, detectors gain incentives automatically once catching any vulnerability, and consumers obtain an authoritative reference regarding with the security of IoT systems. Fig. 1 shows the architecture of SmartCrowd, where three stakeholders are involved in a nutshell:

IoT providers are held accountable for their released IoT systems that are allowed for security detection by detectors. With relatively unrestricted resource, IoT providers take responsibility to construct and maintain the blockchain in SmartCrowd. Besides, existing trustworthy IoT providers can serve as the initiators to bootstrap SmartCrowd.

Detectors play a key role to take distributed detections for securing IoT ecosystems. They can be IoT providers, consumers or third-party services who have IoT detection abilities. Upon discovering a vulnerability, detectors will submit their detection reports, making them automatically gain incentives. Their detection reports will be recorded in the blockchain.

Consumers can access the public blockchain for learning the authoritative references regarding with the security of IoT systems. They can deploy IoT systems only if no (or less) vulnerability is discovered. Consumers can be users or IoT devices that focus on their own security in an IoT ecosystem.

B. Workflow and Challenges

At a high level, SmartCrowd achieves decentralized and automated incentives for distributed IoT system detection through the following phases.

Phase #1: Decentralized verification for system release. IoT system release announcements (SRAs) are disseminated among all stakeholders, including detectors and consumers. In SmartCrowd, an IoT system is eligible for final release only if it has been verified by the majority of detectors via SmartCrowd's decentralized verification process. Verification results are recorded in SmartCrowd's blockchain.

Phase #2: Lightweight and distributed IoT detection. Upon receiving an SRA, detectors that pursue incentives start to detect this IoT system. In particular, detectors will report the uncovered vulnerabilities that will be verified by other entities. SmartCrowd introduces lightweight detectors to mitigate constrained resource, where detectors no longer construct, synchronize and store a heavyweight blockchain locally.

Phase #3: Fault-tolerant verification and storage. The detection reports will be propagated until they are verified and stored in the blockchain. Consumers can quickly learn the system security analysis by querying the related detection results in the blockchain. Leveraging blockchain consensus, SmartCrowd is fault-tolerant for verifying and storing detection results that is determined by the majority of IoT providers.

Phase #4: Decentralized and automated incentives. IoT providers are held accountable for releasing vulnerable systems and earn incentives for maintaining the blockchain, where the detector can automatically gain incentives from them once uncovering any vulnerability. Using smart contracts, the incentive allocation is decentralized and automated once SmartCrowd contracts are triggered.

However, the proposed SmartCrowd still face many challenges as follows.

IoT SRA spoofing. As doing SRA is free, a misbehaved IoT entity can launch spoofing attack and frame benign IoT providers by faking them to release a vulnerable IoT systems.

Plagiarizing detection results. A compromised detector can interfere with the incentive allocation by reporting plagiaristic detection results without doing actual work.

Collusion of stakeholders. A compromised detector can collude with an IoT provider, hoping its forged detection report can be accepted in the blockchain by this IoT provider.

Repudiating incentives and punishments. The IoT providers in SmartCrowd can refuse to accept punishment by transferring no incentive to detectors, disabling the incentives allocation.

The goal of the paper is to allow IoT consumers easily detect and understand the security of IoT systems by using SmartCrowd platform before deploying IoT systems. Therefore, we can significantly reduce the possibility of deploying vulnerable IoT systems and then build a secure IoT ecosystem. SmartCrowd extends the standard blockchain architecture. Besides transactions, the blocks of SmartCrowd also record SRAs and detection reports. By leveraging smart contracts, SmartCrowd develops a new contract to automatically perform an incentive allocation. In this paper, SmartCrowd platform aims to build decentralized and automated incentives for IoT system detection so that different participators can earn incentives by detecting IoT systems published in the platform. Note that, detectors can use existing detection services (e.g., VirusTotal [3] and Quixxi [4]) or build their own systems (e.g., analysis engines of CloudAV [7] and self-certifying alert (SCA) verification of Vigilante [8]) to detect the security of IoT systems and earn incentives as rewards. In this paper, we assume the majority of calculation capability (i.e., hashing power) in an IoT system cannot be controlled by an adversary since it is hard to construct such attacks in practice (see Section VIII). Also, this paper does not address the inherent security problems of blockchain, e.g., eclipse attacks [21] and routing attacks [22], which have been addressed in [23] [24].

V. SmartCrowd Design Details

In this section, we present the design details of SmartCrowd that can address the challenges mentioned above. SmartCrowd introduces insuranced SRA for decentralized verification, and makes lightweight detectors take two-phase report submission. Using the PoW consensus, SmartCrowd prevents collusions of stakeholders for verifying and storing detection results and ensures the decentralized and automated incentives.

A. Decentralized Verification for Insuranced SRA

SmartCrowd makes IoT providers be held accountable for SRAs and prevents them from releasing spoofed systems by enabling decentralized verification for insuranced SRA such that it can accurately identify and punish a misbehaved IoT provider. The IoT SRA verification is decentralized and performed by distributed detectors without relying on centralized services. When doing an SRA, an IoT provider P_i will firstly broadcast an announcement Δ using SmartCrowd contracts, which contains an insurance that will not be refunded once any vulnerability is detected. With the following structure, Δ is used to inform all IoT entities that P_i releases an IoT system.

$$\Delta = \{\Delta_{id}, P_i, U_n, U_v, U_h, U_l, I_i, P_{Sign}\}, \tag{1}$$

where P_i is the unique identifier of an IoT provider. U_n, U_v, U_h and U_l are the name, version, hash value and download link of the released IoT system, respectively. I_i is the submitted insurance of P_i, which helps to prevent a spoofed SRA. Δ_{id} is the identifier of Δ, which is a hash value: $\Delta_{id} = H(P_i||U_n||U_v||U_h||U_l||I_i)$. P_{Sign} is the signature of P_i and is computed as Eq. 2 shows, where sk_{P_i} is the private key of P_i. In SmartCrowd, every IoT entity (e.g., IoT provider, detector, and consumer) has long-time lived public key pk and private key sk. P_{Sign} can help to ensure the authenticity of P_i's SRA.

$$P_{Sign} = Sign_{sk_{P_i}}(\Delta_{id}). \tag{2}$$

SmartCrowd introduces a decentralized verification for an insuranced SRA, which is performed by multiple IoT providers . On receiving Δ, the distributed P_i will firstly check its integrity and authenticity by verifying U_h, Δ_{id} and P_{Sign}, respectively, which prevents spoofed SRAs. Only no error occurs can P_i propagate Δ to its neighbors. Thus, the counterfeit of SRA can be effectively eradicated. Note that the verification for insuranced SRA does not rely on an authority that may be compromised. Instead, using PoW consensus, only a new IoT system that has been verified by the majority of IoT providers ($>$50% hashing power) can be successfully released. Namely, an SRA is available until it has been verified and recorded in the blockchain. This ensures the correctness of IoT SRAs and constrain the misbehavior of IoT providers.

B. Lightweight Detector for Distributed IoT System Detection

SmartCrowd enables lightweight detectors to detect IoT systems without storing or synchronizing their detection results in the local blockchian. Specially, detectors generate the detection reports that can be available to IoT providers. In order to prevent compromised detectors from reporting a forged or plagiarized detection result, SmartCrowd introduces a two-phase report submission that divides detection report into an initial report and a detailed report. Concretely, the detectors download and obtain the released IoT system from U_l. Then, each detector D_i detects and analyzes the security of the IoT system. This can be achieved by its own system (e.g., the detection engine in Vigilante [8] and CloudAV [7]) or existing third-party service (e.g., Quixxi [4] and Ostorlab [11]). Once discovering any vulnerability, D_i will generate and submit its detection report to all IoT providers.

Phase I. Submitting initial report. To quickly declare the vulnerability discovery and prevent theft of detection results, SmartCrowd enables detectors not to submit the detailed detection reports at first. Instead, the detector D_i has to announce its initial report (denoted by $R_i^{\dagger}$), which only includes some simple information (shown in Eq. 3).

$$R_i^{\dagger} = \{ID_i^{\dagger}, \Delta, D_i, H_{R_i^*}, W_{D_i}, D_{Sign}^{\dagger}\}, \tag{3}$$

where $H_{R_i^*}$ is the hash value of R^* (described shortly), and W_{D_i} is the payee address of D_i's wallet. $ID_i^{\dagger}$ is the identifier of $R_i^{\dagger}$, which is a hash value: $ID_i^{\dagger} = H(\Delta||D_i||H_{R_i^*}||W_{D_i})$. $D_{Sign}^{\dagger}$ is D_i's signature that is calculated with sk_{D_i}.

$$D_{Sign}^{\dagger} = Sign_{sk_{D_i}}(ID_i^{\dagger}). \tag{4}$$

In SmartCrowd, $R_i^{\dagger}$ will be recorded in the blockchain by IoT providers according to the consensus scheme (Section V-C). D_i then submits its detailed detection report R_i^*.

Phase II. Submitting detailed report. Detectors report detailed detection results for obtaining more incentives and these reports will be delivered to all IoT providers. When the block containing $R_i^{\dagger}$ is confirmed, D_i will publish the detailed detection report R_i^* to the network. With the following structure (see Eq. 5), R_i^* lists the details of discovered vulnerabilities.

$$R_i^* = \{ID_i^*, \Delta, D_i, W_{D_i}, Des, D_{Sign}^*\}, \tag{5}$$

where Des is the description of discovered vulnerabilities, and ID_i^* is the identifier of R_i^*, which is the hash of Δ, D_i, W_{D_i} and Des. D_{Sign}^* is D_i's signature calculated with sk_{D_i}. On receiving R_i^*, IoT providers will perform correctness verification for Des. Only passing the verification can make R_i^* written in a block (Section V-C). Once this block is confirmed, R_i^* can be recorded in blockchain forever, while the related D_i gain a reward automatically (Section V-D).

C. Fault-Tolerant Verification and Storage for Reports

SmartCrowd enables IoT providers to verify and store the received detection results, and construct and maintain the blockchain using PoW consensus. This can defend against the collusion between IoT providers and detectors, and improve the fault tolerance capability of SmartCrowd, where a small amount of compromised IoT providers will not outplay the whole SmartCrowd platform. Besides detection results ($R_i^{\dagger}$ and R_i^*), SmartCrowd can also verify and store SRAs, which is similar to the verification and storage of detection results.

Algorithm 1 Verification of Detection Report ($R_i^{\dagger}$ and R_i^*).

```
function VERIFICATION FOR $R_i^{\dagger}$ ( )
Require: $R_i^{\dagger}$ and $pk_{D_i}$
    Compute: $ID_i = H(\Delta||D_i||H_{R_i^*}||W_{D_I})$
    if $(ID_i == ID_i^{\dagger})$ && $(CheckSign_{pk_{D_i}}(D_{Sign}^{\dagger}))$ then
        Temporarily record $R_i^{\dagger}$ in a local blockchain;
    else
        Drop the initial report $R_i^{\dagger}$ and break;
    end if
end function
function VERIFICATION FOR $R_i^*$ ( )
Require: $R_i^*$, $R_i^{\dagger}$ and $pk_{D_i}$
    Compute: $ID_i == H(\Delta||D_i||W_{D_i}||Des)$
    if $(ID_i == ID_i^*)$ && $(CheckSign_{pk_{D_i}}(D_{Sign}^*))$ then
        if $H_{R_i^*}$ == H($R_i^*$) then
            if $AutoVerif(P_i, R_i^*)$ then
                Temporarily record $R_i^*$ in a local blockchain;
            end if
        else
            Drop the detailed report $R_i^*$ and break;
        end if
    else
        Drop the detailed report $R_i^*$ and break;
    end if
end function
```

Automatical verification for detection results. SmartCrowd ensures each detection result that will be recorded in a block should be reliable and correct. SmartCrowd enables IoT providers to perform some verification before generating a block. In SmartCrowd, $R_i^{\dagger}$ and R_i^* will be received by all IoT providers, each of which will verify the integrity and authenticity of $R_i^{\dagger}$ and R_i^* by checking the report identifiers ($ID_i^{\dagger}$ and ID_i^*) and the signatures ($D_{Sign}^{\dagger}$ and D_{Sign}^*), respectively (see Algorithm 1). Besides, each P_i will also calculate the hash value of R_i^*, and compare it with $H_{R_i^*}$ in $R_i^{\dagger}$, which can help to defend against such spoofing attack of a misbehaved detector. Meanwhile, the correctness verification of R_i^* is carried out by verifying the result description Des. In SmartCrowd, we define a function $AutoVerif()$ that automatically verifies R_i^* and outputs `TRUE/FALSE`, as Eq. 6 shows.

$$AutoVerif\ (P_i, R_i^*) \rightarrow \texttt{TRUE/FALSE}. \tag{6}$$

Note that $AutoVerif()$ can be deployed as a machine-automatical verification engine using existing services or their own powerful systems. For example, IoT providers can employ analysis engines of CloudAV [7] or an SCA verification of Vigilante [8] to automatically verify viruses and worms detected by distributed detectors. In this case, simply submitting a forged detection report will make $AutoVerif()$ output `FALSE`, where SmartCrowd can isolate a compromised detector by enabling P_i to filter this detector's next reports.

Blockchain-based storage for detection results. SmartCrowd supports fault-tolerant storage for detection results by constructing and maintaining the blockchain among IoT providers. Using PoW consensus, IoT providers can aggregate and record the received detection results in the blockchain. Fig. 2 shows the blockchain architecture of SmartCrowd for storing detection results. PreBlockID and CurBlockID are the identifiers of

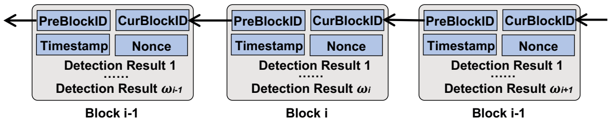


Fig. 2. The architecture of the blockchain in SmartCrowd.

the previous and current block, respectively, which help to link multiple blocks into an order chain. Timestamp is the block generation time, and Nonce is the random number that IoT providers try to seek so that the hash value of the whole block reaches the requirement for generating a new block. Note that block i contains ω_i detection results, which is organized based on the Merkle tree structure like the transaction organization in Bitcoin. Once a new block is generated, it will be broadcast and synchronized among IoT providers. Like Bitcoin system, this block recording detection results will be finally confirmed when 6 newly generated blocks are linked to this blockchain.

D. Decentralized and Automated Incentives Allocation

In order to attract IoT providers and detectors to participate in IoT system detection, SmartCrowd enables decentralized and automated incentives. By leveraging PoW consensus, SmartCrowd allows IoT providers to verify detection results and record them in the blockchain. This helps motivate IoT providers to perform well for generating a new block and automatically obtain rewards. With smart contracts, SmartCrowd forces each IoT provider to submit a security deposit in the blockchain, for being held accountability for releasing vulnerable systems. Once any vulnerability is detected, which can trigger off some smart contract staked in SRAs, the security deposit can be allocated to detectors as incentives, automatically. This can help to address the challenge of repudiating incentives and punishments without relying on a centralized authority. Therefore, SmartCrowd provides completely decentralized incentives to automatically incentivize well-performed IoT providers and detectors and punish misbehaved IoT providers.

Automated incentive allocation. In SmartCrowd, the incentives are automatically allocated to two types of stakeholders: detectors and IoT providers. Distributed detectors can detect IoT systems and build a secure IoT ecosystem; IoT providers can verify and audit detectors' reports and ensure a consistent blockchain. Each IoT provider presets an incentive μ for each detected vulnerability in SmartCrowd contracts while releasing an IoT system. When $R^\dagger$ and R^* are all confirmed and recorded in the blockchain, SmartCrowd contracts will be triggered, resulting in an automatic incentive allocation to this detector. We define the average number n_i of D_i's detected vulnerabilities for an IoT system, which will be finally written in the blockchain with the proportion ρ_i. Thus, D_i will obtain the incentives $in_i^\dagger$ for an IoT system detection as Eq. 7 shows.

$$in_i^\dagger = \ \mu \cdot n_i \cdot \rho_i. \tag{7}$$

The IoT provider whose newly generated block is successfully confirmed will also automatically obtain incentives. Based on blockchain technology, SmartCrowd enables an IoT provider to obtain χ incentives, each of which worth ν. Meanwhile, to motivate IoT providers to verify detection results as much as possible, SmartCrowd allows each IoT provider to obtain the transaction fee in $R^\dagger$ or R^*. We assume the newly generated block contains ω detection reports, each of which includes a transaction fee worth ψ. Thus, the incentives in_i^* allocated to an IoT provider P_i are as Eq. 8 shows.

$$in_i^* = \ \chi \cdot \nu + \psi \cdot \omega. \tag{8}$$

Punishment and cost incurred in SmartCrowd. As the misbehaved IoT entity may exist in IoT ecosystem, SmartCrowd provides an IoT provider with punishments for releasing vulnerable IoT system and costs for deploying smart contracts. Eq. 9 shows the punishments and costs for an IoT provider, where m is the quantity of all detectors and cp_i is the cost of deploying the smart contracts for releasing a new IoT system.

$$pu_i = \ \mu \cdot \sum_{i=1}^{m} n_i \cdot \rho_i + cp_i. \tag{9}$$

SmartCrowd introduces a cost for each detector to submit its detection report. This can prevent detectors from attempting to earn incentives by simply submitting a forged or plagiarized detection report without even having detected IoT systems.

$$co_i = \ n_i \cdot (c + \rho_i \cdot \psi). \tag{10}$$

Eq. 10 shows the cost (denoted by co_i) of detector D_i's reporting detection results, where c is the cost of submitting a detection report and $\rho_i \cdot \psi$ is the average transaction fee for each detection result. We can learn more submitted reports will bring more cost for each detector because only a detection result that has been written in the blockchain can be charged. Note that this cost can also prevent detectors from casually submitting a report for attempting to earning more incentives.

VI. Analysis

In this section, we make the security analysis and theoretical analysis for the ensured security goals and the expected incentives of SmartCrowd.

A. Security Analysis

Security against vulnerable IoT systems. SmartCrowd supports IoT system detection when an IoT provider issues a new IoT version, protecting consumers from attacks when installing an IoT system. Concretely, when an IoT provider releases an IoT system, detectors will obtain this system version from U_l, and then perform security detection. The detection reports will be submitted to all IoT providers, which will be verified and stored in the blockchain permanently. Before installing an IoT system, consumers firstly look up the blockchain and learn the related detection results. In SmartCrowd, consumers can deploy IoT systems with less or no vulnerabilities. Therefore, this can significantly reduce the possibility of deploying vulnerable IoT systems released by misbehaved IoT providers, which can be detected and effectively avoided in SmartCrowd.

Security against misbehaved IoT providers. In addition to releasing vulnerable IoT systems, the misbehaved IoT provider can also attempt to outplay the incentives allocation of SmartCrowd by refusing to pay detectors for their security detection. SmartCrowd can defend against this misbehavior by introducing smart contracts that can automatically allocate incentives to detectors and punish misbehaved IoT providers without relying on a centralized authority. Meanwhile, the misbehaved IoT provider can generate an illegitimate block that may contain incorrect detection results. SmartCrowd can defend against this misbehavior by enabling each newly generated block to be correctly verified by IoT providers as the majority of IoT providers are trustworthy in SmartCrowd.

Security against compromised detectors. SmartCrowd has the ability to resist attacks from compromised detectors, who can try to interfere with incentives allocation or pursue more incentives without actual work. i) SmartCrowd has insights into forged detection results declared by compromised detectors. Each detection result can be correctly verified by the majority of IoT providers, in which the forged reports can be indeed ignored and not be written in the blockchain. ii) SmartCrowd can prevent compromised detectors from plagiarizing detection results by introducing the two-phase submission for a detection report. When $R_i^\dagger$ is written in the blockchain, R_i^* can then be declared to IoT providers. With this method, the compromised detector will not obtain incentives even it reports the plagiarized detection result because it has not declared the initial detection report. iii) SmartCrowd can also prevent compromised detectors from tampering others' detection reports, avoiding maliciously accusing benign detectors to perform an incorrect detection. Using the verification of authenticity and integrity (see Algorithm 1), SmartCrowd can easily identify the modification and counterfeit for detection reports.

Security against collusion attacks. SmartCrowd provides resistance against the collusion of stakeholders, which can be launched by two IoT providers, one IoT provider and another detector, or two detectors. The collusion attacks may be launched for the following purposes: i) decreasing punishments for IoT provider's releasing vulnerable systems, or detector's submitting faked detection reports; ii) obtaining more incentives for IoT provider's creating more blocks, or more detector's reports recorded in the blockchain. SmartCrowd employs PoW-based consensus scheme to ensure the consistency of detection results among the majority of IoT providers. In other words, if there are two collusive stakeholders trying to launching attacks, their operations will not be accepted by other participators. This is because SmartCrowd provides the fault-tolerant verification and storage for detection reports (detailed in Section V-C). More importantly, SmartCrowd provides decentralized and automated incentives/punishments that can help to regulate the behaviors of stakeholders and make them tend to behave normally.

B. Theoretical Analysis

Now we make a theoretical analysis for the total detection capability (denoted by DC_T) of all IoT detectors in our proposed SmartCrowd platform. Then, the balances of detectors and IoT providers are analyzed, respectively, which includes the earned incentives and the incurred punishments and costs. This help us understand the changing trend in gaining incentives.

Total detection capability. In SmartCrowd, D_i's detection capability (denoted by DC_i) can be expressed as the probability for identifying a vulnerability. From Section V-D, we can learn only the detection result that has not been submitted before can be recorded in the blockchain with the probability of ρ_i ($0 \leq \rho_i \leq 1$). Meanwhile, there is up to one detection result can be confirmed for one vulnerability, i.e., $0 \leq \sum_{i=1}^{m} \rho_i \leq 1$, where $\sum_{i=1}^{m} \rho_i$ approaches to 1 when the value of m becomes larger. Therefore, DC_T can be calculated based on Eq. 11.

$$DC_T = \sum_{i=1}^{m} DC_i \cdot \rho_i, \tag{11}$$

where $DC_i \cdot \rho_i$ denote the probability that D_i can discover a vulnerability that would be finally recorded in the blockchain. So $0 \leq DC_i \cdot \rho_i \leq 1$. We can learn the value of DC_T has a positive correlation with m, in which an increased m will introduce a larger DC_t approaching to 1. In other words, more detectors' participation attracted by the incentives in SmartCrowd will introduce a more comprehensive detection results, which helps to provide build-in accountability for IoT providers and authoritative references to IoT consumers.

The balance of detectors. Each detector's balance contains the allocated detection incentives and the cost for reporting detection results. We assume the average period of SRAs is θ so there are t/θ systems being released during a period of time t. According to the allocated incentives $in_i^\dagger$ and cost co_i for an SRA (Section V-D), D_i has the following balance bd_i:

$$bd_i = \ (in_i^\dagger - co_i) \cdot t/\theta. \tag{12}$$

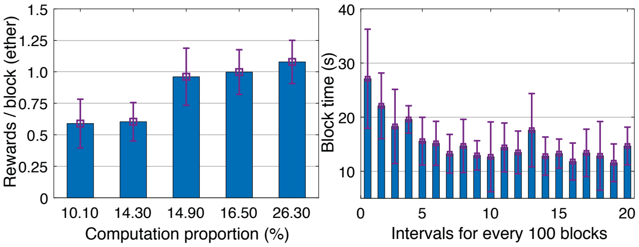
We assume there are averagely N vulnerabilities that will be detected for an SRA during t. The detection capability proportion (denoted by ξ_i) shows proportion of DC_i among DC_T. Therefore, the number (n_i) of D_i's detected vulnerabilities for an IoT system is $N \cdot \xi_i$. Based on Eq. 7, 10 and 12, the balance of D_i can be expressed as Eq. 13 shows.

$$bd_i = \ N \cdot \xi_i \cdot t \cdot [\rho_i \cdot (\mu - \psi) - c] / \theta. \tag{13}$$

The balance of IoT providers contains the allocated incentives for constructing the blockchain and incurred punishments for SRAs. We assume the block time (i.e., the average time of generating a block) is ϑ. Thus, the total number of generated blocks is t/ϑ. In this paper, we use ζ_i to denote the proportion of P_i's computing capability among all IoT providers so that P_i can newly generate $t \cdot \zeta_i/\vartheta$ blocks in the period of t. Therefore, P_i's balance bp_i of is as Eq. 14 shows.

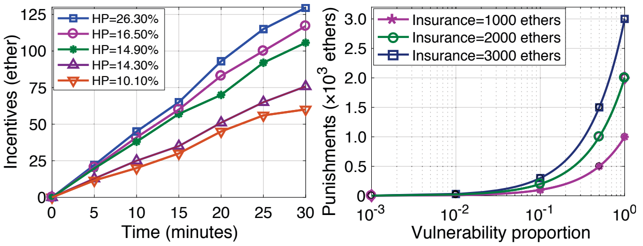
$$bp_i = \ (\zeta_i \cdot in_i^* - pu_i) \cdot t/\vartheta. \tag{14}$$

We can learn more reliable IoT providers, especially with higher computing capability can gain more incentives in SmartCrowd, which can attract the participation of IoT providers for maintaining blockchain-based ledger.



(a) Rewards per mined block (b) Measured block time

Fig. 3. Experimental setup for SmartCrowd.



(a) Incentives for IoT providers (b) Punishments for IoT provider

Fig. 4. Incentives and punishments of IoT providers.

VII. Experimental Evaluation

In this section, we implement SmartCrowd platform based on Ethereum blockchain and evaluate the incentives and the balance of IoT providers and detectors.

We implement SmartCrowd prototype and use Ethereum geth [25] to build our private test blockchain. The platform runs on Ubuntu 14.04 (Dell PowerEdge R710, Inter(R) Xeon(R), CPU X5560 @ 2.80GHz and 35G memory). We implement SmartCrowd contracts with 350 lines of solidity language [26] for simulating the process of both IoT system releases and automated incentive allocations. The IoT detection function of detectors is simulated in Python script. We use the Ethereum JSON API [27] and a python module library of Web3 [28] to implement data interaction between detectors and smart contracts. We face some challenges during our implementation. For example, there are inconsistent function parameters between the hash function SHA-3 in Solidity and the corresponding function SHA-3 in JSON API, which results in an error occurred in the process of verifying signatures. Accordingly, we use the python module library named Web3 to solve the related problems. SmartCrowd supports ECDSA signature [29] and hashing function SHA-3. When receiving detection reports $\mathtt{R}^{\dagger}$ or $\mathtt{R}^{*}$, SmartCrowd enables IoT providers firstly to perform correctness verification (i.e., Algrithm 1) using secp256k1 curve and SHA-3.

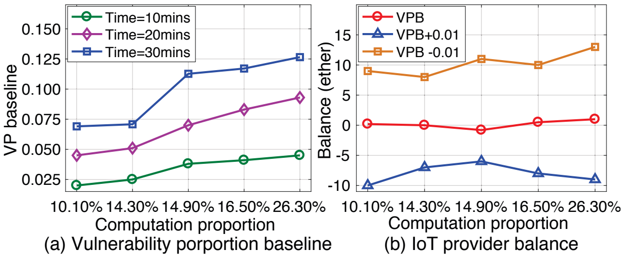
Based on the current hashing power in Ethereum system [30], we set 5 nodes as IoT providers and adjust the thread numbers in function `miner.start()` to simulate top 5 computation proportions. In SmartCrowd, we use 'ether', the cryptocurrency in Ethereum, to evaluate the allocated incentives to IoT providers and detectors, where an IoT provider can gain 5 ethers once creating a block. We set `0xf00000` as the block difficulty in SmartCrowd platform. Fig. 3 shows our experimental setup for SmartCrowd, where Fig. 3(a) shows the average rewards for different computation proportions when one block is created (or mined); Fig. 3(b) illustrates the block time of SmartCrowd, where we have measured 2000 blocks and found the average block time is 15.35 seconds. Using this experiment setup, we evaluate SmartCrowd by using the following important metrics: i) the balance of IoT providers that contains incentives for maintaining the blockchain and punishments for releasing vulnerable IoT systems; ii) the balance of detectors that indicates the earnings for participating in IoT system detection.

A. Balance of IoT providers

In SmartCrowd, we evaluate the balance of IoT provider by measuring and analyzing the incentives and the punishments of IoT providers. In particular, an IoT system release is implemented by deploying a smart contract that records an announcement Δ (see Eq. 1). We find each IoT provider will consume around 0.095 ether as the cost (or gas) for releasing an IoT system. As described in Section V-D, the incentives for IoT providers consist of two parts: i) mining rewards for blockchain construction and ii) transaction fees for recording detection results in the blockchain.

We measure the incentives for IoT providers with different hashing power (HP) proportion, which is showed in Fig. 4(a). We can learn the incentives increase with time, where the longer participation time can make IoT providers gain more rewards. This is because an IoT provider acting as a blockchain miner can create more blocks and gain more transaction fees for a longer time. We can also learn IoT providers with higher HP can gain higher rewards. This can motivate IoT providers to join in SmartCrowd platform for obtaining more incentives. Note that the amount of incentives gained by IoT providers is not strictly obeying their computation proportions, such as the IoT provider with 26.30% HP does not gain 2.6 times incentives compared to the one with 10.10% HP. This is because discovering a `Nonce` of a block or identifying a vulnerability is probabilistic, demonstrating a powerful IoT provider may not create a new block earlier than others. SmartCrowd enables IoT providers to release IoT systems with the insurance that is recorded in smart contracts. We define vulnerability proportion (VP) as the probability that the IoT system released by IoT provider is vulnerable. Fig. 4(b) shows the relationship between punishments and VP with different insurances, where we can learn a high VP can introduce more punishments for a misbehaved IoT provider. This can help to regulate the behaviors of IoT providers and make them tend to release more secure and reliable IoT systems.

In SmartCrowd platform, we define the VP baseline (VPB) that enables an IoT provider achieve a balance of payments (i.e., the incentives are equal to the punishments). Based on Fig. 4, we can get the VPB value of IoT providers with 1000 ethers as insurances for the time period of 10, 20 and 30 minutes, as Fig. 5(a) shows. We can learn an IoT provider with a higher hashing power has a larger VPB, because more incentives (caused by higher hashing power) can offset the



(a) Vulnerability porportion baseline (b) IoT provider balance

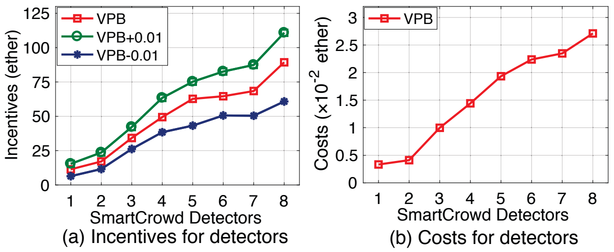
Fig. 5. Balance of IoT providers.

punishments caused by a larger VP. Using the VPB for the time period of 10 minutes, we evaluate the balance of IoT providers with the insurance of 1000 ethers when the VP is VPB, VPB+0.01 and VPB-0.01, as Fig. 5(b) shows. We can learn IoT providers can achieve the balance of incentives and punishments at VPB while a larger and a smaller VP can make IoT providers financially lossy and profitable, respectively. Meanwhile, IoT providers can obtain an additional 10 ethers when the VP is reduced by 0.01. This can incentivize IoT providers to release more non-vulnerable IoT systems, contributing to building a more secure IoT ecosystem.

B. Balance of Detectors

In order to evaluate the balance of SmartCrowd detectors, we simulate 8 detectors to perform distributed detection for a new IoT system release. We preset the detection capabilities of detectors by adjusting thread numbers ($1 \sim 8$) allocated to them. As described in Section V-D and VI-B, the balance of SmartCrowd detectors consists of two parts: i) incentives for IoT system detection, ii) cost of detection report submission.

We consider the SRA from an IoT provider with 14.90% computation proportion as an example to evaluate the balance of detectors. From Fig. 5(a), we can learn that the VPB value is 0.038 for the time period of 10 minutes and the insurance of 1000 ethers. Fig. 6(a) shows the incentives (measured for 100 times) that are allocated to detectors for VPB, VPB+0.01 and VPB-0.01, respectively. We can learn the larger detection capability makes detectors easier gain more incentives such that the incentives allocated to the detector with 8 threads are around 7.8 times as much as the one with 1 thread. This can help to attract detectors with larger detection capabilities to participate in SmartCrowd platform for IoT system detection. Meanwhile, a larger VPB can introduce more incentives. For example, whenever VPB increases 0.01, the detectors can gain $3 \sim 23.5$ ethers (as incentives) more. This is conducive to holding IoT providers accountable for releasing any vulnerable IoT systems. We measure the cost (gas) of detectors' reporting detection result under VPB, as shown in Fig. 6(b), where each detection report can consume around 0.011 ether. We can learn that the cost is negligible compared to the allocated incentives. This demonstrates the balance of detectors is almost equal to the incentives in Fig. 6(a), which encourages benign detectors to try to participate in SmartCrowd for detecting more vulnerabilities, enhancing the security of IoT ecosystem.



(a) Incentives for detectors (b) Costs for detectors

Fig. 6. Balance of SmartCrowd detectors.

VIII. Discussion

Deployment benefits. IoT consumers, providers, and detectors can benefit from SmartCrowd that enhances the security of IoT ecosystem by incentivizing system detection. i) With detection results in the blockchain, customers can install securer systems that have no or less vulnerabilities. This prevents hackers from enrolling them into a botnet. ii) Detectors can participate in IoT detection more actively for obtaining more rewards financially. This enables to introduce a more comprehensive security analysis for the released IoT systems. iii) SmartCrowd can help to regularize IoT providers that can gain rewards for maintaining the blockchain and also be punished for releasing vulnerable IoT systems.

51% attack. The blockchain technology like SmartCrowd that uses PoW consensus scheme is vulnerable to 51% attack [31]. Anyone who controls the majority of hashing power can destroy the PoW consensus and introduce double-spending problem [32]. Besides, the IoT provider that launches 51% attack can maliciously modify the unfavorable detection results. However, according to the statistics [30], no miner or pool has occupied more than 30% hashing power in current Ethereum system, in practice. Thereby, 51% attack will also hardly happen in SmartCrowd platform.

Detection capability. SmartCrowd enables distributed detectors to detect released IoT systems or IoT providers to verify detection results of detectors. The detection capability can be achieved by making IoT detectors or providers i) construct their own vulnerability/virus libraries, for example, integrating the published CVE [33], NVD [34], and SecurityFocus [35]; or ii) using the existing third-party services, such as VirusTotal [3], Ostorlab [11], and Andrototal [9]. SmartCrowd enables incentives not only for static detection, but also for dynamic or fuzzy testing as long as IoT detectors or providers have these detection capabilities.

N-version vulnerability descriptions. The problem of differently-worded versions for the same vulnerability is not detailed in this paper, which can be addressed using existing methods. For example, CloudAV [7] enables the analysis engines in network service for detection result aggregation that can filter other descriptions of a same virus. Vigilante [8] introduces a common description language for self-certifying alert to depict the detected vulnerabilities so that it can avoid the problem of differently-worded detection results.

Market competition between SmartCrowd detectors. SmartCrowd never involves the market competitions among different detectors (e.g., security companies), which will not make these participating detectors lose their own market shares. This is because only the detection results instead of their core technologies are shown and recorded in the blockchain without any technical leaks. In other words, the involved detectors only announce their detection reports for some released IoT system instead of showing how to identifying vulnerabilities. The proposed SmartCrowd can protect the core technologies of detectors while enhancing their participation through incentives.

IX. Related Work

In this section, we describe the related works of SmartCrowd from the following three areas: IoT system detection, blockchain-based verification, and incentives schemes.

IoT system detection. Securing IoT systems has been widely studied. Byung-Chul et al. propose a secure firmware validation and update scheme for consumer devices in a home networking [36]. Nilsson et al. introduce a lightweight and secure firmware upgrade scheme, which provides data integrity, data verification, data confidentiality, and freshness for in intelligent vehicles [37]. Muhammad et al. propose a firmware update protocol for a new security architecture within the vehicle, which facilitates the update processes by combining updated of hardware and software modules [38]. Wu et al. propose RFL [39] and PPV [40] that can be used to ensure the secure packet transmission between IoT devices. Tian et al. propose a security technology called SmartAuth for IoT Apps in terms of user-centered authorization, which ensures the consistency verification between explained functionalities and actual operations for an IoT App [41]. However, these approaches all rely on a centralized authority, where the security and ability of the third party significantly impact the effectiveness of vulnerability detection.

Security enhancements built upon blockchain. The blockchain can be used to achieve security verification and privacy protection [42] [43]. IKP is a blockchain-based platform to report unauthorized certificates, contributing to the security of PKIs while it fails to consider the accountability for the misbehaviors of certificate authorities [44]. Chen et al. present a blockchain-powered decentralized secure audit scheme for TLS connections, which relies on a distributed dependability-rank based consensus protocol for avoiding centralization [45]. Wu et al. introduce a decentralized incentives (called SmartRetro) for IoT system retrospective detection, which automatically sends security notifications to IoT consumers once discovering any vulnerabilities [46]. Ali et al. propose Blockstack that is based on existing Namecoin blockchain system for guaranteeing the security of global naming and storage [47]. Hawk is a blockchain-based framework to ensure the privacy protection in smart contracts, which does not store financial transactions in plain text on the blockchain [48]. Garman et al. construct a decentralized anonymous credential system that uses a public append-only ledger, avoiding a trusted credential issuer [49]. Boudguiga et al. propose an IoT update scheme build upon a blockchain to construct an infrastructure for a better availability and accountability [50]. The last three schemes do not consider incentives that can encourage users to be involved in building secure systems.

Incentives schemes. Most previous studies of incentives [51] [52] [53] are mainly used to build reputation systems. Mira et al. [51] analyzed and proposed incentives for outsourced computation in a reputation or credit system. Gilad et al. [52] developed a fair and efficient secure multiparty computation in reputation systems. However, these reputation-based incentives are not automatic and require manual interference. With cryptocurrencies, blockchain-based incentives can be achieved automatically, where participants are rewarded with reliable work and fined for misbehavior. Ranjit et al. performed a detailed analysis for using bitcoin to incentivize correct computation [53]. Matsumoto et al. utilized blockchain-based consensus and smart contracts to achieve the fully decentralized and automated incentives for responding to the misbehaviors of certificate authorities [44]. However, their approaches over-depend on the authority for data verification.

X. Conclusion

In this paper, we propose SmartCrowd, a platform built upon the blockchain that implements decentralized and automated incentives for distributed IoT system detection. SmartCrowd crowdsources IoT system detection to distributed detectors so that a complete and authoritative reference regarding with the security detection of IoT systems can be provided to IoT consumers. Meanwhile, SmartCrowd enables IoT providers to be held accountable for their released IoT systems, where releasing more secure systems can help to gain more incentives and vulnerable systems can incur punishments. By decentralized and automated incentives, detectors can gain rewards automatically when catching any vulnerability in released IoT systems. Therefore, IoT consumers can quickly understand the vulnerabilities of IoT systems by looking up the detection results in SmartCrowd blockchain. We implement SmartCrowd prototype based on Ethereum and use real experiments to evaluate its performance. The results show SmartCrowd has both technical feasibility and financial benefits for stakeholders involved in SmartCrowd. We hope that the decentralized and automated incentives of SmartCrowd can become an essential primitive to construct a secure IoT ecosystem.

Acknowledgments

This work was supported in part by National Key R&D Program of China under Grant 2018YFB0803405, the National Natural Foundation of China under Grants 61572278 and U1736209, China National Funds for Distinguished Young Scientists under Grant 61825204, EU Marie Curie Actions CROWN under Grant FP7-PEOPLE-2013-IRSES-610524, Beijing Outstanding Young Scientist Project, NSF under Grant CNS 17-17313 and the Huawei Technologies Entrustment Project (HF2019015003).

References

[1] International Data Corporation (IDC). Worldwide and regional internet of things (iot) 2014 - 2020 forecast: A virtuous circle of proven value and demand. https://www.business.att.com/content/article/IoT-worldwide_regional_2014-2020-forecast.pdf,2014.

[2] Manos Antonakakis, Tim April, Michael Bailey, Matt Bernhard, Elie Bursztein, Jaime Cochran, Zakir Durumeric, J Alex Halderman, Luca Invernizzi, et al. Understanding the mirai botnet. In *USENIX Security Symposium (USENIX Security)*, pages 1093–1110. USENIX, 2017.

[3] VirusTotal. https://www.virustotal.com,2018.

[4] QUIXXI. How secure is your mobile app? https://quixxi.com,2018.

[5] Samsung Electronics Co., Ltd. SmartThings (Samsung Connect). https://play.google.com/store/apps/details?id=com.samsung.android.oneconnect,2018.

[6] Samsung Electronics Co., Ltd. Samsung Smart Home. https://play.google.com/store/apps/details?id=com.samsung.smarthome,2018.

[7] Jon Oberheide, Evan Cooke, and Farnam Jahanian. Cloudav: N-version antivirus in the network cloud. In *USENIX Security Symposium (USENIX Security)*, pages 91–106. USENIX Association, 2008.

[8] Manuel Costa, Jon Crowcroft, Miguel Castro, Antony Rowstron, Lidong Zhou, Lintao Zhang, and Paul Barham. Vigilante: End-to-end containment of internet worms. In *ACM SIGOPS Operating Systems Review*, volume 39, pages 133–147. ACM, 2005.

[9] Andrototal. AndrolTotal is a free service to scan suspicious APKs against multiple mobile antivirus apps. http://andrototal.org,2018.

[10] Alibaba. Alijuanquan. http://jaq.alibaba.com,2018.

[11] Ostorlab. Secure your mobile app. https://ostorlab.co,2018.

[12] High-Tech Bridge. Mobile App scanner. https://www.htbridge.com/mobile,2018.

[13] Jeff Howe. The rise of crowdsourcing. *Wired magazine*, 14(6):1–4, 2006.

[14] Gavin Wood et al. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper*, 151:1–32, 2014.

[15] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2008.

[16] National Institute of Standards and Technology (NIST). Fips 180-2. https://csrc.nist.gov/encryption/tkhash.html,2002.

[17] Hans Dobbertin, Antoon Bosselaers, and Bart Preneel. RIPEMD-160: A strengthened version of RIPEMD. In *Fast Software Encryption*, pages 71–82. Springer, 1996.

[18] Miguel Castro and Barbara Liskov. Practical byzantine fault tolerance. In *Proceedings of USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, pages 173–186. USENIX, 1999.

[19] Nicola Atzei, Massimo Bartoletti, and Tiziana Cimoli. A survey of attacks on ethereum smart contracts (sok). In *International Conference on Principles of Security and Trust (POST)*, pages 164–186. Springer, 2017.

[20] Wikipedia. 2016 Dyn cyberattack. https://en.wikipedia.org/wiki/2016_Dyn_cyberattack.

[21] Atul Singh, Tsuen-Wan Ngan, Peter Druschel, and Dan S. Wallach. Eclipse attacks on overlay networks: Threats and defenses. In *IEEE International Conference on Computer Communications (INFOCOM)*. IEEE, 2006.

[22] BGPStream. https://bgpstream.com,2018.

[23] Ethan Heilman, Alison Kendler, Aviv Zohar, and Sharon Goldberg. Eclipse attacks on bitcoin's peer-to-peer network. In *USENIX Security Symposium (USENIX Security)*, 2015.

[24] Maria Apostolaki, Aviv Zohar, and Laurent Vanbever. Hijacking bitcoin: Routing attacks on cryptocurrencies. In *IEEE Symposium on Security and Privacy (S&P)*, pages 375–392. IEEE, 2017.

[25] Official Go implementation of the Ethereum protocol. https://github.com/ethereum/go-ethereum, 2018.

[26] The solidity contract-oriented programming language. https://github.com/ethereum/solidity, 2018.

[27] JSON RPC. https://github.com/ethereum/wiki/wiki/JSON-RPC, 2018.

[28] A Python interface for interacting with the Ethereum blockchain and ecosystem. https://github.com/ethereum/web3.py, 2017.

[29] Wikipedia. Elliptic curve digital signature algorithm. https://en.wikipedia.org/wiki/Elliptic_Curve_Digital_Signature_Algorithm, 2018.

[30] Etherscan. Ethereum top 25% miners by blocks. https://etherscan.io/stat/miner, 2018.

[31] Majority attack. https://en.bitcoin.it/wiki/Majority_attack, 2018.

[32] Meni Rosenfeld. Analysis of hashrate-based double spending. *arXiv preprint arXiv:1402.2009*, 2014.

[33] CVE. Common vulnerabilities and exposures. http://cve.mitre.org,2018.

[34] National Institute of Standards and Technology (NIST). National vulnerability database (NVD). https://nvd.nist.gov,2018.

[35] SecurityFocus. The vulnerabilites developed by SecurityFocus. http://www.securityfocus.com,2018.

[36] Byung-Chul Choi, Seoung-Hyeon Lee, Jung-Chan Na, and Jong-Hyouk Lee. Secure firmware validation and update for consumer devices in home networking. *IEEE Transactions on Consumer Electronics*, 62(1):39–44, 2016.

[37] Dennis K Nilsson and Ulf E Larson. Secure firmware updates over the air in intelligent vehicles. In *IEEE International Conference on Communications (ICC) Workshops*, pages 380–384. IEEE, 2008.

[38] Muhammad Sabir Idrees, Hendrik Schweppe, Yves Roudier, Marko Wolf, Dirk Scheuermann, and Olaf Henniger. Secure automotive on-board protocols: a case of over-the-air firmware updates. In *International Workshop on Communication Technologies for Vehicles*, pages 224–238. Springer, 2011.

[39] Bo Wu, Ke Xu, Qi Li, and Fan Yang. Robust and lightweight fault localization. In *IEEE 36th International Performance Computing and Communications Conference (IPCCC)*, pages 1–8. IEEE, 2017.

[40] Bo Wu, Ke Xu, Qi Li, Zhuotao Liu, Yih-Chun Hu, Martin J Reed, Meng Shen, and Fan Yang. Enabling efficient source and path verification via probabilistic packet marking. In *IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*, pages 1–10. IEEE, 2018.

[41] Yuan Tian, Nan Zhang, Yue-Hsun Lin, XiaoFeng Wang, Blase Ur, Xianzheng Guo, and Patrick Tague. Smartauth: User-centered authorization for the internet of things. In *USENIX Security Symposium (USENIX Security)*, pages 361–378. USENIX, 2017.

[42] Meng Shen, Xiangyun Tang, Liehuang Zhu, Xiaojiang Du, and Mohsen Guizani. Privacy-preserving support vector machine training over blockchain-based encrypted iot data in smart cities. *IEEE Internet of Things Journal*, 2019.

[43] Feng Gao, Liehuang Zhu, Meng Shen, Kashif Sharif, Zhiguo Wan, and Kui Ren. A blockchain-based privacy-preserving payment mechanism for vehicle-to-grid networks. *IEEE Network*, (99):1–9, 2018.

[44] Stephanos Matsumoto and Raphael M Reischuk. Ikp: Turning a pki around with decentralized automated incentives. In *IEEE Symposium on Security and Privacy (S&P)*, pages 410–426. IEEE, 2017.

[45] Jing Chen, Shixiong Yao, Quan Yuan, Kun He, Shouling Ji, and Ruiying Du. Certchain: Public and efficient certificate audit based on blockchain for tls connections. In *IEEE Conference on Computer Communications (INFOCOM)*, pages 2060–2068. IEEE, 2018.

[46] Bo Wu, Qi Li, Ke Xu, Ruoyu Li, and Zhuotao Liu. Smartretro: Blockchain-based incentives for distributed iot retrospective detection. In *International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, pages 308–316. IEEE, 2018.

[47] Muneeb Ali, Jude Nelson, Ryan Shea, and Michael J Freedman. Blockstack: A global naming and storage system secured by blockchains. In *USENIX Annual Technical Conference (USENIX ATC)*, pages 181–194. USENIX, 2016.

[48] Ahmed Kosba, Andrew Miller, Elaine Shi, Zikai Wen, and Charalampos Papamanthou. Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. In *IEEE Symposium on Security and Privacy (S&P)*, pages 839–858. IEEE, 2016.

[49] Christina Garman, Matthew Green, and Ian Miers. Decentralized anonymous credentials. In *Proceedings of Network and Distributed System Security Symposium (NDSS)*, 2014.

[50] Aymen Boudguiga, Nabil Bouzerna, Louis Granboulan, Alexis Olivereau, Flavien Quesnel, Anthony Roger, and Renaud Sirdey. Towards better availability and accountability for iot updates by means of a blockchain. In *European Symposium on Security and Privacy Workshops (EuroS&PW)*, pages 50–58. IEEE, 2017.

[51] Mira Belenkiy, Melissa Chase, C Chris Erway, John Jannotti, Alptekin Küpçü, and Anna Lysyanskaya. Incentivizing outsourced computation. In *Proceedings of the International Workshop on Economics of Networked systems (NetEcon)*, pages 85–90. ACM, 2008.

[52] Gilad Asharov, Yehuda Lindell, and Hila Zarosim. Fair and efficient secure multiparty computation with reputation systems. In *International Conference on the Theory and Application of Cryptology and Information Security (AsiaCrypt)*, pages 201–220. Springer, 2013.

[53] Ranjit Kumaresan and Iddo Bentov. How to use bitcoin to incentivize correct computations. In *Proceedings of the Conference on Computer and Communications Security (SIGSAC)*, pages 30–41. ACM, 2014.