JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

EntrapNet: a Blockchain-Based Verification Protocol for Trustless Computing

Mr. RAMBABU ATMAKURI - Head, Department of CSE, Anurag College of Engineering (Aushapur, Ghatkesar, Telangana 501301)

Mr. VALLANDAS SANDEEP - Student, Department of CSE, Anurag College of Engineering (Aushapur, Ghatkesar, Telangana 501301)

Mr. PINNITY PRAMOD REDDY - Student, Department of CSE, Anurag College of Engineering (Aushapur, Ghatkesar, Telangana 501301)

Mr. PAGILLA RAKESH - Student, Department of CSE, Anurag College of Engineering (Aushapur, Ghatkesar, Telangana 501301)

Mr. M. SHIVA KUMAR - Student, Department of CSE, Anurag College of Engineering (Aushapur, Ghatkesar, Telangana 501301)

Abstract

In this paper, we propose a blockchain-based computing verification protocol, called EntrapNet, for distributed shared computing networks, an emerging underlying network for many internets of things (IoT) applications. EntrapNet borrows the idea from the practice of entrapment in criminal law to reduce the possibility of receiving incorrect computing results from trustless service providers who have offered the computing resources. Furthermore, we mathematically optimize EntrapNet to deal with the fundamental tradeoff of a network: security and efficiency. We present an asymptotic optimal solution to this optimization. It will be seen that EntrapNet can be performed as an independent and low-cost layer atop of any trustless network that requires outsourced computing, thus making secure computing affordable and practical.

1. INTRODUCTION

Like electricity, computing power is now an essential utility in human daily life, spanning from education and science to marketing and media. The analysis of big data is reshaping many real-world problems, driving an increased demand for computing power. However, access to powerful computing devices, such as cloud servers, has historically been unfair and undemocratic due to their high costs. Conversely, the waste of computing power from billions of idle devices in homes and data centers worldwide is substantial. Therefore, the challenge lies in effectively utilizing these idle computing devices to meet the growing demand for computing power.

Blockchain technology holds promise in addressing this challenge by connecting idle, geo-distributed computing devices through a secure and server-less network. In this network, clients/users can rent computing power from providers located anywhere in the world to complete various tasks, such as machine learning training, 3D rendering, and scientific computations. Clients pay for these services using cryptocurrencies like Bitcoin, while providers are rewarded for offering their computing resources. Essentially, this creates a distributed shared computing platform or marketplace aimed at maximizing the utilization of computing resources globally.

However, a longstanding issue with outsourcing computational tasks to other parties is how clients/users can efficiently verify the results without re-executing the tasks. Providers may lack strong incentives to ensure correctness, and for complex and large-scale providers like cloud servers, it is challenging to guarantee correct execution due to factors such as misconfigurations and hardware randomness. This problem, known as verifiable computing, has been a subject of study for computer scientists for nearly a decade.

2. LITERATURE SURVEY

A straightforward solution to this problem is to replicate computations on multiple computing devices. However, this solution implicitly assumes that failures from these computing devices are uncorrelated. This assumption can be invalid in many cases, for example, cloud servers always have homogeneous hardware and software platforms. Another solution is based on the method of taking a small group of samples and then auditing the responses in these samples. However, if the incorrect outputs do not occur frequently, this solution may not perform well. One can also find other solutions such as attestation and trusted hardware, but these solutions always require a chain of trust and guaranteed correct hardware computation. One groundbreaking approach that uses the traditional technique is to force the provider to provide a short proof which can prove the correctness of the computation. This short proof should be “short and easy-to-check” compared to re-executing the computing task. Many proof-based systems have been developed recently, including Pinocchio, TinyRAM, Pantry, and Buffet. However, due to the tremendous overhead of the cryptographic setup between the client and provider, and the computational cost for the provider to construct such a short proof, these systems are only near practical. Another novel approach leverages the blockchain technology to secure outsourced computation. One example is TrueBit, a smart contract that verifies the computational results through a trustless economic protocol. However, the 5-to-50 times more cost of using Truebit and its restriction to Ethereum network could be the bottleneck for its massive adoption. In addition, a new approach that has a similar idea with our work is to add some precomputed subtasks in the original data set and verify them upon task completion. For example, [17] has presented a mechanism to verify the outsourced computation for biometric data, which inserts some precomputed fake items into the biometric data set to detect misbehavior of lazy servers, which the malicious node cannot distinguish. However, this method only focused on specific computation algorithms and data set structures. Furthermore, inserting fake items into every computation task for every server is lack of efficiency.

3. OVERVIEW OF THE SYSTEM

3.1 Existing System

A straightforward solution to the problem of verifying computational tasks is to replicate computations on multiple computing devices. However, this approach assumes that failures from these devices are uncorrelated, which may not always hold true. For instance, cloud servers typically have homogeneous hardware and software platforms, meaning failures could be correlated. Another approach involves auditing a small group of samples to verify responses, but this method has limitations.

3.1.1 Disadvantages of Existing System

Assumption of Uncorrelated Failures: Replicating computations on multiple devices assumes uncorrelated failures, which may not be the case in scenarios like cloud computing.

Limited Effectiveness of Sampling: Auditing responses from a small group of samples may not effectively detect incorrect outputs, especially if such errors are rare.

3.2 Proposed System

In criminal law, entrapment is a tactic used by law enforcement to induce a person to commit a criminal offense they would otherwise be unlikely to commit. The EntrapNet protocol adopts this concept, with a client/user in the network playing the role of the "officer" aiming to catch a malicious provider. This is done by assigning the provider a task with a known outcome (a "fishing task"), allowing any misconduct to be easily detected.

3.2.1 Advantages of Proposed System

Detection of Malicious Providers: By assigning predictable tasks to providers, EntrapNet enables easy detection of misconduct in computing tasks.

Enhanced Security: EntrapNet enhances the security of distributed shared computing networks by reducing the risk of incorrect computing results from malicious providers.

Adaptation of Legal Concept: By borrowing from the legal concept of entrapment, EntrapNet introduces a novel approach to ensuring the integrity of computing tasks in distributed networks.

Two-Phase Mechanism: The EntrapNet mechanism is discussed in two phases, providing a comprehensive overview of its operation and effectiveness.

Prevention of Unfair Practices: EntrapNet prevents providers from engaging in unfair practices by ensuring that their outputs can be easily verified by the client/user.

3.3 Proposed System Design

In this project work, there are three modules and each module has specific functions, they are:

1. Officer Module
2. Provider Module
3. Block chain Module

3.3.1 Officer Module

client/user in the network, being a role of "officer", aims to catch a malicious provider in the network by assigning the provider with a fishing task. Since the outcome of the fishing task is predictable/known in advance by the officer, any mis conduct on computing can be easily detected by the officer. Using this module officer will register with application and generate few questions and send those questions to provider and answers to block chain server. Officer will verify results from block chain and know who is fake and genuine provider based on answers.

3.3.2 Provider Module

Providers do not necessarily have strong incentives to ensure correctness. On the other hand, for complex and large-scale providers (e.g., cloud servers). it is unlikely to guarantee that the execution is always correct due to mis-configurations, randomness in hardware and more.

Using this module provider will register with application and login with valid username and password and check questions from officer and answer to those questions some providers will give correct answers some will give wrong answers and send to block chain server and block chain server will send block chain key to provider if he is genuine.

3.3.3 Blockchain Module

The information of all the tasks in the task pool are written into the task pool smart contract on blockchain, the task scheduler will invoke the smart contract that randomly assigns a task to providers

Using the fishing-task information submitted by the officer, the contract will first confirm if this fishing task is a network-verified task. In particular, the contract will generate the abstract from the fishing-task information and compare it with the one stored on the blockchain which was submitted by the officer in the phase of building his fishing-task repository.

Using this module block server will login to application and view answers submitted by officer and provider and verify correctness of answers and if they are valid then block chain is generated to genuine user and details are sent to officer.

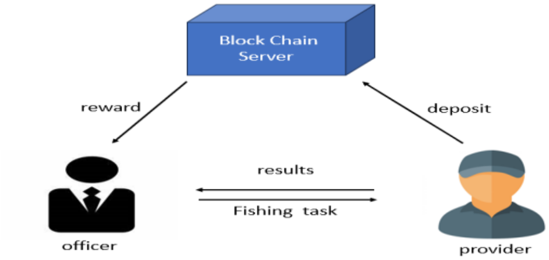
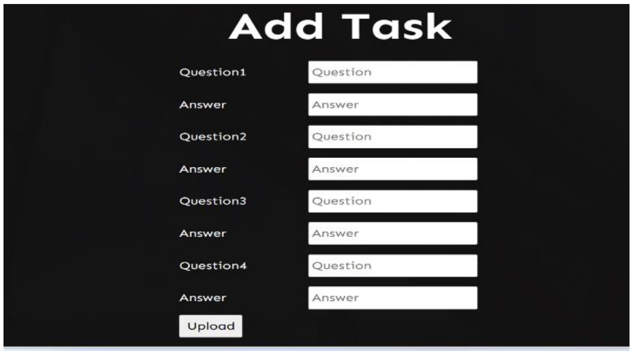


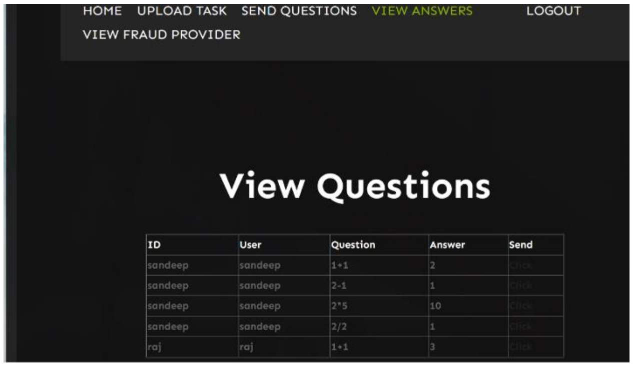
Fig 1: System Architecture

4. RESULT SCREEN SHOTS



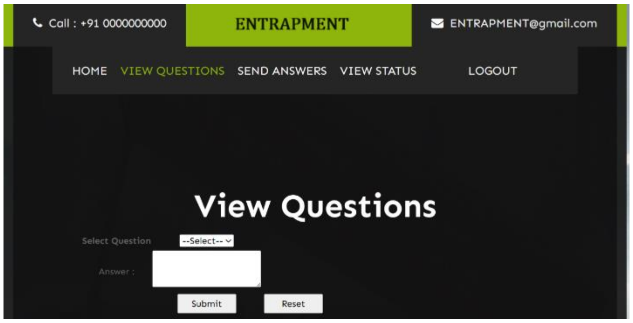
Add Task

Question1 Question
Answer Answer
Question2 Question
Answer Answer
Question3 Question
Answer Answer
Question4 Question
Answer Answer



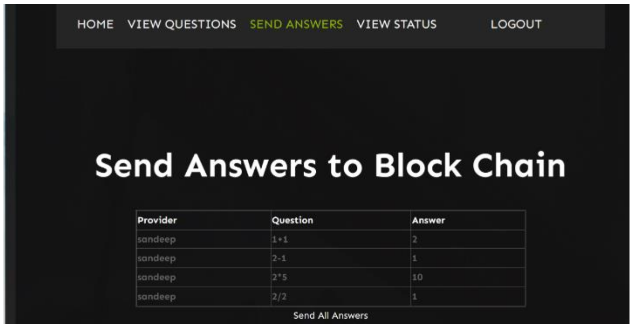
View Questions

ID	User	Question	Answer	Send
sandeep	sandeep	1+1	2	Click
sandeep	sandeep	2-1	1	Click
sandeep	sandeep	2*5	10	Click
sandeep	sandeep	2/2	1	Click
raj	raj	1+1	3	Click



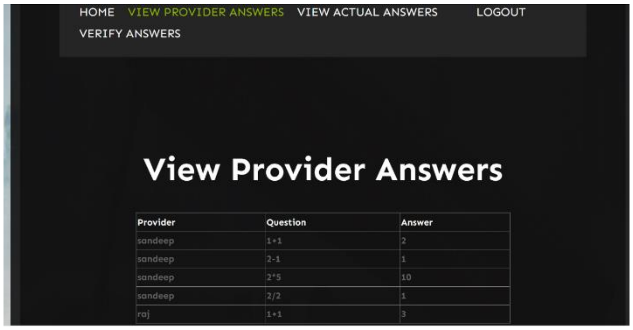
View Questions

Select Question --Select--
Answer :



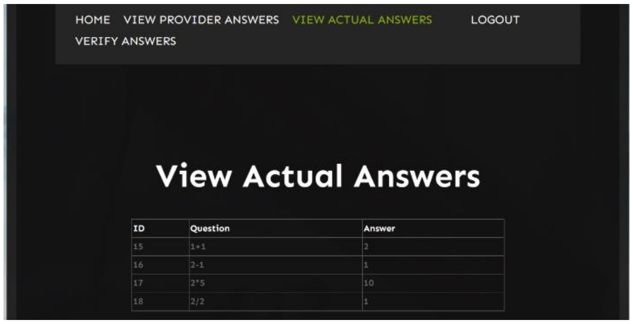
Send Answers to Block Chain

Provider	Question	Answer
sandeep	1+1	2
sandeep	2-1	1
sandeep	2*5	10
sandeep	2/2	1



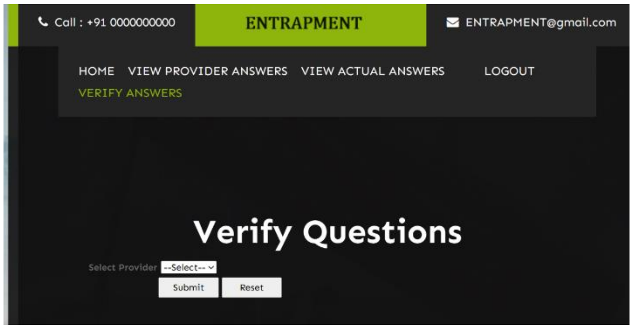
View Provider Answers

Provider	Question	Answer
sandeep	1+1	2
sandeep	2-1	1
sandeep	2*5	10
sandeep	2/2	1
raj	1+1	3



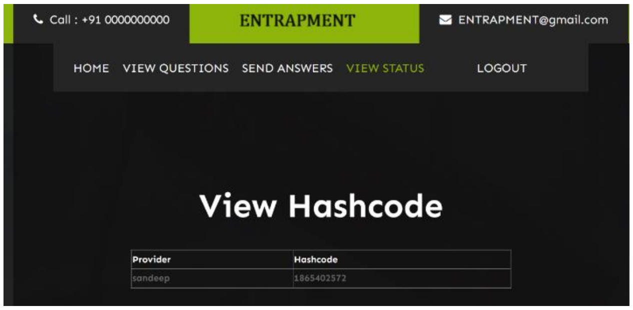
View Actual Answers

ID	Question	Answer
15	1+1	2
16	2-1	1
17	2*5	10
18	2/2	1



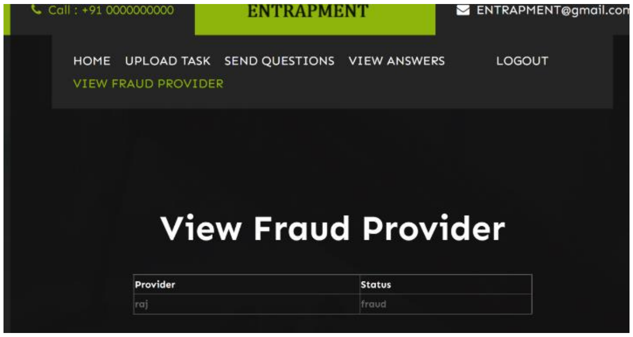
Verify Questions

Select Provider --Select--



View Hashcode

Provider	Hashcode
sandeep	1865402572



View Fraud Provider

Provider	Status
raj	fraud

5. CONCLUSION

In this paper, we have proposed a blockchain-based computing verification protocol, called EntrapNet, which borrows the idea from the practice of criminal entrapment. The EntrapNet can complete the verification on outsourced computational tasks with very low cost. We have further proposed a mathematical framework to analyze EntrapNet with the objective to optimize the tradeoff between the network security and efficiency. EntrapNet is agnostic to the underlying blockchain and can be directly applied to any distributed computing network that requires outsourced computation to trustless parties. In the end, we have presented a simulation example of EntrapNet that is built on the Ethereum blockchain, and provided insights in how to optimize the network.

We have further proposed a mathematical framework to analyze EntrapNet with the objective to optimize the tradeoff between the network security and efficiency. In the end, we have presented a simulation example of EntrapNet that is built on the Ethereum blockchain, and provided insights in how to optimize the network. In the future, more sophisticated blockchain-based schemes can be designed based on EntrapNet. For example, how the current approach can be applied to tasks that really need to be split amongst several providers and how would security and efficiency be calculated in that case.

6. REFERENCES

[1] Canonchain project whitepaper, http://www.canonchain.com/.
[2] Golem project whitepaper, https://golem.network/home/.
[3] Sonm project whitepaper, https://sonm.com/.
[4] Zcash project whitepaper. http://z.cash/.
[5] R. Canetti, B. Riva. and G. Rothblum. “Practical delegation of computation using multiple servers”. In Proceedings of the 18th ACM conference on computer and communications security, 2011, pp. 445-454.
[6] M. Castro, and B. Liskov. “Practical Byzantine fault tolerance and proactive recovery”. ACM Trans. on Comp. Sys, vol. 20, no. 4, 398-461, 2002.
[7] D. Malkhi and M. Reiter, “Byzantine quorum systems”, Distributed Computing vol. 11, no. 4, pp. 203-213, 1998.
[8] A.R.Sadeghi, T. Schneider and M. Winandy, “Token-based cloud computing: Secure outsourcing of data and arbitrary computations with lower latency”. In Proceedings of TRUST, 2010.
[9] B. Parno, J.M. McCune, and A. Perrig,“Bootstrapping Trust in Modern Computers”. Springer Science & Business Media, 2011.
[10] B. Parno and C. Gentry, “Pinocchio: Nearly practical verifiable computation”, IEEE Symposium on Security and Privacy, Oakland, 2013, pp. 238-252.
[11] E. Ben-Sasson, A. Chiesa, D. Genkin, E. Tromer, and M. Virza, “SNARKs for C: verifying program executions succinctly and in zero knowledge”. In Advances in Cryptology-CRYPTO, pp. 90-108, 2013.
[12] B. Braun, A. J. Feldman, Z. Ren, S. Setty, A. J.

Blumberg, and M. Walfish. “Verifying computations with state”. In Proceedings of SOSP, Nov. 2013.

[13] R. S. Wahby, S. Setty, Z. Ren, A. J. Blumberg and M. Walfish. “Efficient RAM and control flow in verifiable outsourced computation.” In Proceedings of NDSS, Feb. 2015.

[14] M. Walfish and A. J. Blumberg, “Verifying computations without reexecuting them.” Communications of the ACM 58.2, pp. 74-84, 2015.

[15] Wikipedia, “https://en.wikipedia.org/wiki/Entrapment”.

[16] J. Teutsch and C. Reitwiessner, “A scalable verification solution for blockchains”, online, Mar 2017.