Distributed Computing Power Market

Xiaoqi CHEN, Zhen XIAO

***Abstract*—This paper was originally submitted to Xinova as a response to a Request for Invention (RFI) on cyber-attack forecasting, detection, and stabilization methods. In this paper, a unique method of distinguishing real users from bots is proposed.**

In more detail, this describes an authentication system and method to authenticate human users by the user electing to assist in solving an NP-hard calculation problem that is too costly for bots to perform.

I. Introduction

1. User identification

For The identification of DDoS attackers requires significant computing power. The more sophisticated the equipment (e.g., IoT devices) employed by attackers becomes, the more real the attack traffic looks like, and hence the more computing power is required for defense (distinguishing attackers from normal visitors). For example, an attacker may have a complete IoT device system stack, and thus become capable of launching attacks after establishing TCP connections rather than sending individual packets. This would make SYN cookie-based SYN-flood defense no longer effective.

Although existing solutions by means of the idea of proof-of-work (asking a visitor to calculate a hash function to prove his/her identity while helping the owner of a website to "mine") are available, such approach is unable to handle the situation where an attacker is in control of a large botnet having more computing power than normal visitors.

Moreover, with the rise of machine learning, it may be difficult for visitor verification solely based on the form of verification code (i.e., a machine makes up a question, and a person answers it) to defend against an attacker having a large amount of computing power for training machine learning models. Existing ReCaptcha solutions create access authenticity indicators of a user by tracking access history of the user in the past, so that the user only needs to "tick off" to complete the verification. However, this approach poses a risk for privacy. On the other hand, having human visitors verify each other (i.e., a human makes up a question, and another human answers it) might be a better approach that is capable of effectively verifying real human visitors, assuming that machines are still unable to pass a complete Turing test.

2. Payment for website content

In the present business environment, users pay for the content of a website mainly in two ways. The first way is through direct payments such as membership fees. However, many people do not want to pay for content directly. The other way is to watch advertisements. Advertisements may result in poor user experience and can be easily blocked by browser plug-ins such as adblock.

3. Grid computing

A number of complex problems such as the prediction of spatial conformations of proteins are of great importance for drug design, but consume enormous computing power. In order to solve those problems, existing technologies use grid computing to assign computation to a large number of PC users. However, grid computing still relies on the donation of computing power in which individual users donate their computing time to public welfare projects for free, and commercial grid computing has not yet been well developed.

4. Digital cryptocurrencies

The exchange rate of digital cryptocurrencies to real currencies waves violently, making it difficult to price real commodities using digital cryptocurrencies. People are currently purchasing digital cryptocurrencies just for speculation, while the trading of real commodities using cryptocurrencies remains impractical.

II. Summary of the Invention

A demander for commercial grid computing raises a complex problem (typically, a NP-hard problem), put it into a smart contract, and pays for an answer, which can be efficiently verified. A user to be verified extracts the problem from the smart contract, and performs computation by consuming the computing power of his/her own PC to solve the problem. If an answer to the problem is successfully obtained and submitted into the smart contract, the user can be guaranteed to be a real user and given a real user ticket for a certain period of valid time. When the user accesses a website during this period of valid time, it would be possible to prove that he/she is a real user simply by submitting the ticket, and the website may charge a fee from the smart contract based on the user's ticket. The demander for commercial grid computing obtains the answer to the complex problem from the smart contract, the

website earns money from the smart contract, and the user's computer pays out its computing power while keeping browsing experience unaffected.

III. Detailed Explanation

Existing technologies:

- As attackers are using sophisticated equipment generating attack traffic that looks like real traffic, the computing power required for defending attacks (distinguishing attackers from normal visitors) are increased, and therefore makes SYN cookie-based SYN-flood defense no longer effective.

- Visitors to a website may be disgruntled with the fact that they have to prove their identities by calculating hash functions and helping the owner of the website in mining, and thus waste their computing power.

- With the rise of machine learning, it may be difficult for visitor verification solely based on the form of verification code (i.e., a machine makes up a question, and a person answers it) to defend against an attacker having a large amount of computing power for training machine learning models.

- Existing ReCaptcha solutions create access authenticity indicators of a user by tracking access history of the user in the past. Such solutions pose a risk for privacy.

In this proposed solution:

- There is no privacy risk, and an owner of a website can profit from selling computing power of users rather than their personal information.

- Visitors' computers perform meaningful computation rather than wasting their computing power in hash mining. A company could, if appropriate, disclose to the users the motivation and value to the society of the specific computation they participated (e.g., protein folding – in the process of calculating your verification code, you are effectively helping the development of new drugs for fighting cancer).

- The payment process is diverted so that the visitors can pay for website content by proving their computing power as needed, and thereby get rid of unwanted advertising, and the content provider no longer has to directly charge each of the visitors.

- A distributed computing power market may be formed by anyone who employs the solution

Overview of usage scenario

- A drug company P needs significant computing power for the prediction of spatial conformations of proteins in order to develop new drugs, and therefore buys computing power from Company A's computing power market.

- A video website W recently faces a jump in the number of visitors, and it is not yet clear whether the cause is related to a DDoS attack or real increase of users. Therefore, the video website W seeks help from Company A's computing power market.

- Company A's computing power market decomposes the computation required by drug company P into parallel computing problems and put them into a smart contract. A computer of each user visiting website W needs to compute a part of the computing problems proposed in the smart contract. The user is authorized to visit website W if a correct answer is obtained, or otherwise it is considered that the user is an attacking bot.

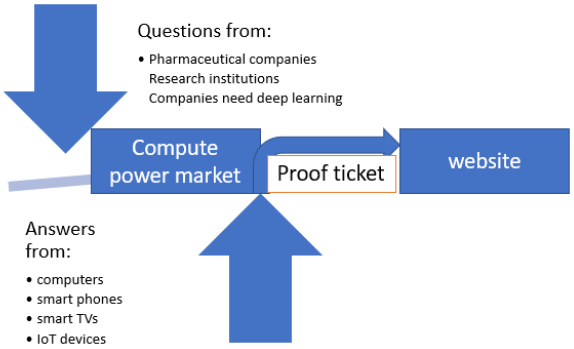
- In return, the user visiting website W no longer has to watch advertisements that appear before desired videos, neither does the user have to purchase a membership since the membership fee is covered by the computing power bought by the drug company P.

- The user's computer may perform computation in background while the user is watching a video on website W, or complete the computing problem at midnight when it is not in use, and thereby earn credits.

Overview of smart contract

A smart contract is a computer protocol intended to digitally facilitate, verify, or enforce the negotiation or performance of a contract. Smart contracts allow the performance of credible transactions without third parties. In the present solution, a smart contract can be considered as a distributed computing task assignment program running on a digital cryptocurrency protocol. The program is able to assign computing tasks in a distributed manner, retrieve results, and automatically transfer tokens for calculating credits.

Figure 1: Conceptual overview



Design of smart contract

- As described in the overview of usage scenario, the smart contract of the present solution carries out the following tasks: The demander for computing power, drug company P, buys tokens in cash from Company A's computing power market, and uses the tokens to obtain the right to add distributed computing problems into a problem pool.

- The smart contract extracts a set of problems from the problem pool and proposes those as challenge questions to a users' computer.

- The user's computer performs computation and returns an answer to the smart contract for verification, and the smart contract sends a certain number of credit tokens to the user's computer if the answer is correct.

- The user's computer needs to pay credit tokens to website W when the user visits website W.

- Website W may exchange the collected tokens into cash from Company A's computing power market. The language used for writing smart contracts in Ethereum (ETH) is taken as an example to partially illustrate the mechanism of the present solution.

Problem pool

A problem pool is an array of problems each being a Struct:

```Solidity
Struct Question{
        String problem description or problem pointer; // describing each problem
        Float[] answer array; // collected answers to problems
        Bool correctness marker; // True: problem has been answered; False: problem has not yet been answered
        Uint acknowledging time; // the time the problem was answered
        Address question owner; // questioner who proposed the problem - entitled to extract the answer to the problem
}
Question[] questionList;
```

- The proposed problem shall be one that has a definitive answer, such as matrix multiplication. If a problem is too complex, a URL may be recorded in the problem description from where a user client can obtain the problem in order to avoid putting too much storage load onto the smart contract.

- The proposed problem may be one that requires a human action such as marking or identification of images, or it may also be one that can be completed merely by performing background computation by the computer. The latter is preferred for improving browsing experience of the user.

- The same problem can be proposed to multiple users, and answers can be collected and recorded in the answer array. If a majority of the answers are consistent, for example, 4/5 of the users have got the same answer, the "correctness marker" = True, and the problem is deemed resolved.

- The problems in the problem pool are ordered by the times they are answered. In the case a certain number of answered problems are held in the problem pool, the questioner may take away those ones and delete them from the pool.

- When initializing the problem pool, a portion of the problems put into the pool must have their corresponding correct answers.

Questioner

The questioner needs to pay Company A to add new problems into the problem pool. In doing so, payment recorded in the smart contract can be made directly with digital cryptocurrencies, or it can be coins internally used in the smart contract as a voucher for collecting payment from the questioner to avoid the issue of exchange rate fluctuation in the direct use of digital cryptocurrencies.

The questioner may extract answers to the problems it proposed.

``` Solidity
Function AddQuestion(Address Questioner, Question specific problem) {
   questionList += specific problem
 }

Function Extract_Answer(Address Questioner) returns (Question) {
    If total number of correctness markers in questionList>threshold{
       For problems in questionList {
```

```
        if problem. problem owner==Questioner {return problem}
      }
   }
}
```

User verification

The smart contract extracts a set of problems from the problem pool as challenge questions to a user to be verified. The set of problems contains at least an answered problem, and the correctness of the answer has been verified.

The user's computer or cell phone needs to compute answers to the challenge questions and submit the answers to the smart contract.

A program required by the computation can be called by a plug-in of a browser,

```
``` solidity
Function answer(Question[] challenge questions, Address user to be verified)
answers (Question[] challenge questions){
   For question in challenge questions{
         answer array += call external computation to get answer to question
         }
   }
   ```
```

If the user is unable to answer to a question that has been marked as correct, it is considered that all answers from the user are not trustworthy, and the user would not be verified. If the user gives a correct answer to a marked question, answers from the user to other questions are accepted.

When the user passes verification, a certain number of tokens are transferred to the user's coin account.

Information including user IP and port can be recorded in user information to submit to the website to be visited

```
   ```solidity
   Struct User{
      Address user e-wallet address; // equivalent to a marker of user identity
      uint token credits; // number of verifications
      Struct user attached information{
      user IP address;
      user port number;
            }
       }
      ```
```

Upon visiting the website, tokens of the user will be consumed, and the user will become unable to visit the website when the number of tokens drops to 0.

Website

The website may charge certain tokens from the user for the provided content or time duration. The website may exchange tokens it held into a certain amount of cash from Company A.

Cash flow

- The questioner buys tokens from Company A to obtain rights to propose questions and extract answers.
- Company A keeps a part of the tokens as profit.
- The user answers questions to earn tokens by consuming computing power.
- The website provides content to the user and charge tokens in return.
- The website exchange its tokens into cash from Company A.

The above process does not require the user, the website or the questioner to buy or exchange digital cryptocurrencies such as BitCoin, ether, etc., and thereby avoids losses from exchange rate fluctuation.

IV. Conclusion

Overall, the solution describes an authentication system and method to authenticate human users by having the user elect to assist in solving an NP-hard calculation problem. By donating a portion of the user's resources to solve a real problem before granted access to restricted content and resources, something that bots will find computationally expensive, a more secure environment can be created with minimal to no burden on the human.

Acknowledgment

This research was originally submitted to Xinova, LLC by the author in response to a Request for Invention. It is among several submissions that Xinova has chosen to make available to the wider community. The author wishes to thank Xinova, LLC for their funding support of this research.

More information about Xinova, LLC is available at www.xinova.com.