An Algorithm that Prevents SPAM Attacks using Blockchain

Koichi Nakayama, Yutaka Moriyama
Faculty of Science and Engineering,
Saga University
Saga, Japan

Chika Oshima
Faculty of Medicine,
Saga University
Saga, Japan

***Abstract*—There are many systems and methods for preventing spam attacks. However, at present there is no specific tried-and-true method for preventing such attacks. In this paper, we propose an algorithm, "SAGA$_{BC}$" to prevent spam attacks using a blockchain technique and demonstrate its effectiveness by a simulation experiment. A person who sends an email using the "SAGA$_{BC}$" must pay the processing cost with cryptocurrency. If an e-mail sent using this algorithm is received normally at a destination e-mail account, this fee is refunded. However, a lot of spam e-mails are not received normally, because addresses of the spam e-mails are indiscriminate. If a spammer sends spam using the "SAGA$_{BC}$," he/she will lose the cryptocurrency fee for each such message. Thus, if using the "SAGA$_{BC}$" to send e-mail becomes a standard practice for the general public, receiving e-mail servers and/or mailers will be able to easily judge incoming messages without using the "SAGA$_{BC}$," because spammers cannot use the "SAGA$_{BC}$" without losing their cryptocurrency.**

Keywords*—Cryptocurrency; wallet account; Mail Send Coin (MSC)

I. Introduction

Unwanted electronic mail (e-mail), known as "spam" appear in many people's inboxes every day. People who send spam e-mail (called "spammer" in this paper) aim to spread advertisements and computer viruses and play tricks on their targets. Many spams impose a high load on a network and may affect the processing of legitimate e-mails.

Some technical methods to protect e-mail users from spam do exist. When a receiving server receives an e-mail, it authenticates the validity of the message using information from the sending DNS server. The technical methods of authenticating sender domains include "Sender Policy Framework (SPF) [1]", "Sender ID", "DomainKeys Identified Mail (DKIM [3])", "Domain Name System Blacklist (DNSBL [2])", among others. Whereas the SPF and the SenderID use IP addresses to authenticate a sender domain, the DKIM uses an electronic signature.

The receiving e-mail server refuses any e-mail from an IP address that does not have an SPF record. The SPF record is a text record verifying that a domain's administrator made and is registered to DNS. The SPF record includes the IP address of the server permitted to send e-mail using the domain name as an e-mail source. The receiving server checks the SPF record against the IP address of the sending server which forwarded the e-mail; if the IP address of the sending server does not match that of the SPF record, the receiving e-mail server considers the e-mail to be spam. Although the SPF refers to the sender's e-mail address as designated in the "From": field, the Sender ID refers to that of the header. The DKIM is a way of sending an electronic signature to the receiving e-mail server, which then acquires a public key from the sending DNS server and verifies the DKIM signature. The DNSBL is a software mechanism to stop spam. Lists consisting of the IP addresses of servers sending spam are then supplied to receiving e-mail servers.

These problems may be negligible on a daily basis, but over timeweeks or even dayshundreds and thousands of spam messages make it through flawed filters. We outline these problems below.

1) **Cannot verify the e-mail account unit.**
When a sending server is determined to be a spam server, all the e-mail accounts that use that server are prevented from sending and receiving e-mail. On the other hand, if a spammer switches to a different sending server, the attack can no longer be prevented.
2) **Cannot prevent a first spam attack.**
Most spam attacks go through one or more sending servers which are not the true server, and the number of these "zombie computers" can increase exponentially. This method also ensures that traditional protections cannot prevent a first spam attack.
3) **Sometimes a normal e-mail is erroneously identified as spam.**
Most filtering functions provided by Internet service providers and e-mail software identify spam according to contents of the message, and thus may erroneously mark normal messages as spam. Moreover, if the contents of an e-mail message contain graphics, the filtering function may not work because it scans the graphic but cannot determine what the graphic portrays.
4) **Infringing e-mail users' privacy.**
Because Internet service providers and e-mail software judges whether an e-mail is spam according to its contents, individual information may be disclosed.
5) **Receiving servers and e-mail software carry a heavy workload.**
Current spam filters screen all incoming e-mail to identify spam, and receiving servers and e-mail software need to renew their filtering function constantly to catch new types of spam. The workload for identifying spam is heavier than that of launching a spam attack. Receiving servers and e-mail software incur

considerable financial and processing costs in dealing with spam.

We will resolve these problems using a block-chain technology. In the next section, we offer precise definitions of the relevant terms in this paper before explaining "SAGA$_{BC}$". Then, in Section IV, we present the results of a simulation experiment using SAGA$_{BC}$. We also discuss the method's ability to prevent spam, based on the experiment's results. The paper's conclusion follows:

II. SAGA$_{BC}$

A. Concept

A genuine e-mail from a harmless person can be received normally. However, some e-mail addresses used in spam are fictitious or are rejected by receiving servers. Therefore, only a fraction of the spam sent out by a spammer reach their targets. In our proposed method, anyone who sends an e-mail message must pay a processing fee in cryptocurrency, but if the e-mail is received correctly, that fee will be refunded to the sender. Those sending genuine, harmless e-mail will pay a little, but spammers must spend much more to launch a spam attack. We expected that this will reduce spam attacks. We call this method the "SPAM Attack Guard Algorithm Using Block Chain (SAGA$_{BC}$)".

B. Definition of Terms

Because cryptocurrency is a relatively new concept, the definitions of relevant terms offered by various publications have been vague and sometimes contradictory. Therefore, we will offer precise definitions of the relevant terms in this paper.

Blockchain
: Blockchain is a kind of a distributed database (Distributed Ledger Technology, or DLT). Data is accumulated per a unit "block". Each block records the Hash values of the unit immediately preceding it. Therefore, it is necessary to calculate Hash values for all the data leading to a falsified block to falsify the data on the way. In other words, it is very difficult to falsify the data for a blockchain. "Bitcoin [6]" and "Ethereum [7]" are well-known kinds of blockchain cryptocurrencies.

Cryptocurrency
: "Electronic money is commonly defined as value stored electronically, issued on receipt of funds of an amount not less in value than the monetary value issued, and accepted as a means of payment by parties other than the issuer [8]". "Digital currency is a type of currency available only in digital form, not in physical. Examples include virtual currencies and crypto currencies or even central bank issued [9]". "Virtual currency is a digital payment mechanism for (and denominated in) fiat currency [10]". "Digital and virtual currencies can either be centralized or decentralized [4]". "Cryptocurrency refers to any electronic money created using a cryptographic technology [4]". "Cryptocurrency is a purely decentralized peer-to-peer electronic cash system for validating value transfers [5]". "BTC" and "ETH" are types of cryptocurrency comprised of blockchains, such as "Bitcoin [6]", and "Ethereum [7]".

Wallet
: The "wallet" is a means of storing cryptocurrency. Anyone can freely create a wallet, and "users can send and receive bitcoins electronically using wallet software on a personal computer, mobile device, or web application [11]". We consider the wallet to be a mechanism for managing cryptocurrency.

Wallet account
: A "wallet account" is an ID used to identify an individual wallet. Wallet users manage their cryptocurrencies using unique wallet accounts.

Transaction
: A "transaction" is a record of sending cryptocurrency from one's own wallet account to another wallet account. The digital signature of the owner of the cryptocurrency as well as his/her private key are needed to issue the transaction.

Mining
: "When the sending user transfers cryptocurrency to a recipient, the transaction is verified by a process called 'mining [13]'." There are public keys used to verify information and permit the execution of transactions requested by others (called "miners"). Once a transaction is verified and approved by a miner, it is executed and stored in a digital block [12]. The entire transaction, from issuance to verification, only takes a few minutes.

C. System Set-up

The SAGA$_{BC}$ cooperates with an e-mail account associated with a wallet account to prevent spam attacks. Generally, an e-mail client has one or more e-mail accounts. One or more wallet accounts are assigned to each e-mail account by the SAGA$_{BC}$. The e-mail client cooperates with one e-mail account associated with a wallet account.

The SAGA$_{BC}$ system comprises the following components:

1). **Cryptocurrency: Mail Send Coin**
The Mail Send Coin (MSC) is one of the cryptocurrencies implemented by the SAGA$_{BC}$. The MSC is not a monetary token but a kind of utility token. Anyone using the SAGA$_{BC}$ can also use existing cryptocurrencies, e.g., Ethereum. However, in this paper, we will explain the SAGA$_{BC}$ with specifically in terms of using MSC.

2). **Mailers**
In the SAGA$_{BC}$, an expanded function (add-on) of the general mailer is implemented.

(2-1) The account management function
: As shown in Fig. 1, the account management function extracts those wallet accounts that correspond not only to the owner's e-mail account but also to a destination e-mail account. This function then inquires of the blockchain whether the

Fig. 1. Extracting wallet accounts.

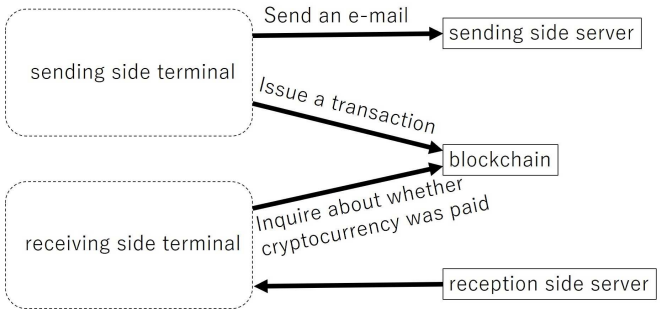

Fig. 2. Validating wallet accounts.

sending wallet account has paid the MSC into the receiving wallet account.

(2-2) Inquiring whether MSC was paid
This function inquires a blockchain about whether the MSC was paid from the sending side wallet. Any data gathered from such reference results are then stored in this function.

(2-3) The sorting function
This function assesses whether e-mails are spam according to the amount of MSC paid to send them and sorts them into a spam e-mail folder.

(2-4) The remittance function
The remittance function is an MSC payment function operating from the wallet account corresponding to the owner's e-mail account that contacts the wallet account corresponding to the destination (receiving) e-mail account.

(2-5) The validation function
The first time an e-mail is sent to a new recipient, this function validates the wallet account corresponding to the destination e-mail account. The sending mailer then checks the associated wallet account and determines whether it has already remitted the MSC fee to the receiving mailer. The receiving mailer connects its associated wallet account with the sending mailer depending on the identity of the sending wallet account and whether the appropriate amount of MSC has been paid. The sending mailer then sends the MSC fee to the receiving wallet account.

(2-6) The mining function
If the MSC paid is insufficient, a user can supplement it by mining MSC transactions that other users have issued. A spammer can also supplement his or her MSC in the same way, but it costs much more for an illegitimate user to do so.

D. Procedure to be Followed when Both the Sending and Receiving Mailers use the SAGA$_{BC}$

1) **Sending mailers**
As shown in Fig. 2, when a mailer sends an e-mail, it issues a certain number of transactions sending MSCs to the wallet account corresponding to the destination e-mail account.

2) **Receiving mailers**
The receiving mailer determines whether a received e-mail message is spam based on the amount of MSC attached and sorts the spam into the spam folder. The receiving mailer then automatically decides whether to refund the MSC fee paid according to how the e-mail message is processed. If the message is deleted or sorted into the spam folder, the MSCs paid for it is not refunded. However, if the message is not processed within a certain period of time, the amount of MSC paid can be refunded to the sending wallet account.

3) **Mining**
Issued transactions are recorded at the head of the blockchain by a miner. All dealings related to the transaction are then concluded.

E. Procedure to be Followed when Either the Sending or the Receiving Mailer does not use the SAGA$_{BC}$

1) **When only the sending mailer uses the SAGA$_{BC}$**
Sending mailers can determine whether the receiving mailer uses the SAGA$_{BC}$ with the validation function (see (2-5)). In this case, the sending mailer can send a regular e-mail without paying a transaction fee in MSC.

2) **When only the receiving mailer uses the SAGA$_{BC}$**
Receiving mailers can determine whether the sending mailer uses MSCs by the function for inquiring whether MSC was paid (see (2-2)). If the sending mailer does not use the SAGA$_{BC}$, the receiving mailer will know this because of the account management function (see (2-1)). In this case, the receiving mailer deals with incoming messages as normal e-mail that cannot be confirmed as not being spam.

3) **When neither the sending nor the receiving mailer uses the SAGA$_{BC}$**
In this case, e-mail is sent and received using a traditional method.

F. Anticipated Effects of a Spam Attack

When using the SAGA$_{BC}$, the e-mail sender must simultaneously send an MSC fee to the receiving wallet when sending

an e-mail message. Because spammers send vast amounts of e-mail, they will lose MSCs doing this, which will eventually discourage them from sending e-mail. When the normal e-mails are received correctly, those sending such messages do not lose their MSCs: Even if the MSCs they sent disappears, they can restock by mining. We therefore expect that SAGA$_{BC}$ users will cease to receive spam.

III. Simulation Experiment

In this section, we verify whether the SAGA$_{BC}$ can prevent spam attacks using a simulation. The experimental simulation will not include e-mail senders who do not use the SAGA$_{BC}$.

A. Experiment Model

Fig. 3 shows the main routine of the simulation experiment.

1) **Initial setting**
The number of SAGA$_{BC}$ users is indicated by "N". The initial value of the MSC that all users possess is indicated by "M". Of N, the number of spammers and the number of genuine users are indicated by "S" and "(N-S)", respectively.
2) **Sending e-mails and MSCs**
A genuine SAGA$_{BC}$ user sends an e-mail and 1 MSC to an address selected from those of the other users (except for the user's own address and the spammers addresses). If a genuine user does not have any MSC, the e-mail cannot be sent to an address using the SAGA$_{BC}$.
3) **Refunds**
None of the e-mails sent by the (N-S) genuine users are spam. The 1 MSC fee sent to the receiving wallet is refunded to the wallet associated with the user's e-mail account.
4) **Loop for genuine users**
The simulation repeats routines 1, 2, and 3 above for the (N-S) users. In general users do not perform mining.
5) **Sending e-mails and MSCs**
A spammer uses the SAGA$_{BC}$ to send a spam message and 1 MSC to an address selected from those of the other users (except for the spammer's own address. If the wallet associated with the spammer's account has no MSC, the spammer cannot send the spam message.
6) **Refunds**
Any e-mail sent by a spammer is considered spam, and thus the MSC that spammers send to receiving wallets are not refunded.
7) **Profit**
Spammers make a profit b via the probability p per spam message sent. Spammers acquire the same amount of MSC through the profit b they make from mining.
8) **Loop for sending spam**
The simulation repeats the above routines 5, 6, and 7 T times. T is selected as a uniform random number from natural numbers that satisfy 0T<N. Namely, each spammer will send T spam messages to an e-mail account except for his/her own e-mail account, without overlapping the unit time for each.

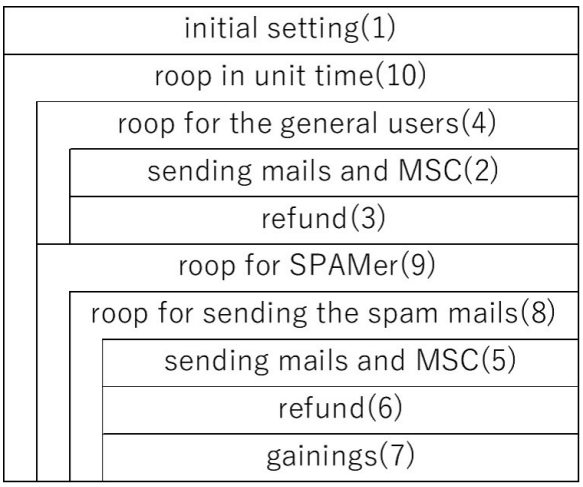

Fig. 3. Main routine of the simulation experiment.

9) **Loop for Spammers**
Routines $5-8$ are repeated for all spammers.
10) **Loop in unit time**
Routines $2-9$ are considered one unit time (t) and repeated.

B. Parameter

In this simulation, the parameters are set as follows: N=10,000 $(M \subset 980, 1000, 1020)$, $(S \subset 300, 500, 700)$. The probability distribution P of the profit G is calculated as follows:

$$P = 1000 \times (C)^{(-X)}, \qquad (1)$$

where the constant C is (27), and x is the uniform distribution of random numbers satisfying $0 < x < 330$.

C. Result

This simulation was performed 100 times for each of the three kinds of initial values of MSC, satisfying $S = 500$. Fig. 4 shows the shift of the average throughout each of the 100 runs for the three conditions ($M \subset 980, 1000, 1020$). The horizontal axis of the figure indicates the unit time t. The vertical axis of the figure indicates the ratio of spam to all e-mails sent.

This figure shows that the ratio of spam to all e-mails sent clearly decreases, although the speed of this decrease differs for each of the three kinds of initial values of MSC.

The next simulation was performed 100 times for each of the three numbers of spammers, satisfying $(M = 1,000$. Fig. 5) shows the shift of the average throughout each of the 100 simulations for the three conditions $(S \subset 300, 500, 700)$.

This figure shows that the ratio of spam to all e-mail sent clearly decreases, although the speed of this decrease differs for each of the three kinds of profit that spammers make.

IV. Discussion

The results of the simulation show that the SAGA$_{BC}$ can prevent spam. The SAGA$_{BC}$ is more effective than traditional spam prevention methods. Because the spam prevention

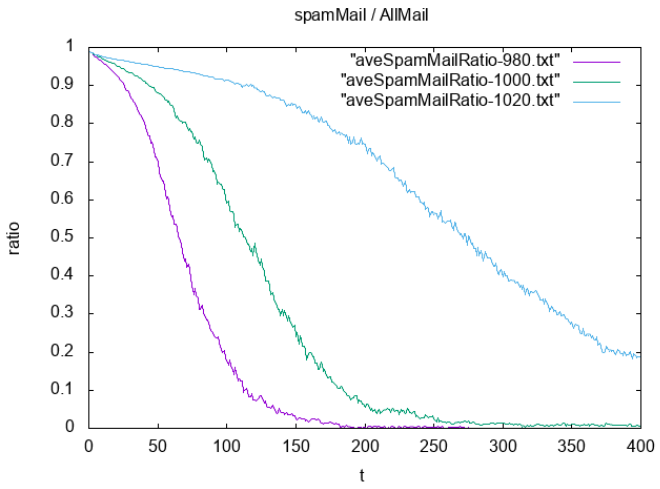

Fig. 4. Result of the simulation ($M \subset 980, 1000, 1020$).

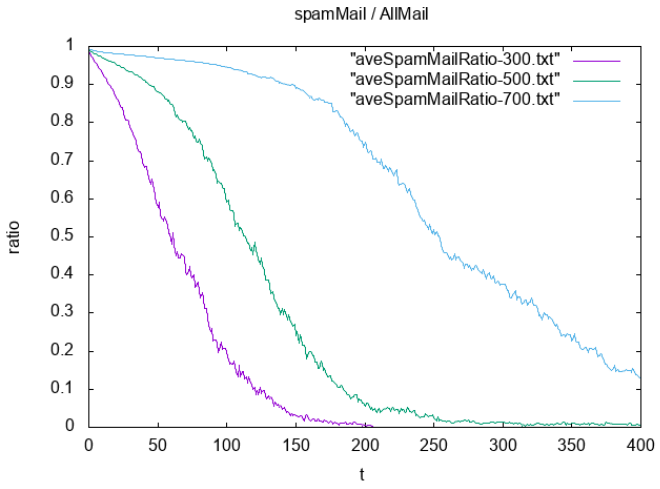

Fig. 5. Result of the simulation ($S \subset 300, 500, 700$).

takes place in both the sending server and the filter of the receiving side server, there are distinct advantages in using the SAGA$_{BC}$.

1) Even if the sending server of the user is same as that of the spammer, the SAGA$_{BC}$ can prevent spam attacks because the SAGA$_{BC}$ determines whether an e-mail is spam or legitimate in each e-mail account.
2) Even if the spammer switches to a different sending server, the SAGA$_{BC}$ will prevent him/her from sending spam unless he/she acquires MSC.
3) Because the receiving e-mail is paid for with MSC by the sender's wallet, the receiving server and mailer do not need to assess the contents of an e-mail and thus have a small workload.
4) Users sending e-mails have the assurance that their messages will not be classified as spam as long they pay the MSC fee.
5) If a user receives an e-mail for which the MSC has been paid that turns out to be spam, he or she accrues the MSC paid because the fee is not refunded to the spammer.
6) The results of the simulations indicate that spam attacks will decrease when e-mails are sent using the SAGA$_{BC}$.

Because the SAGA$_{BC}$ creates disadvantages for spammers, they will not use it. However, genuine users can be assured that e-mail they receive which have been paid for MSC are unlikely to be spam.

V. Conclusion

In this paper, we propose the SAGA$_{BC}$ algorithm to prevent spam attacks using a blockchain. SAGA$_{BC}$ cooperates with an e-mail account associated with a wallet account to achive this. Anyone who sends an e-mail message must pay a processing fee in cryptocurrency MSC. However, if the e-mail is received correctly, the fee will be refunded to the sender. We conducted a simulation experiment to demonstrate that spam attacks decreased when using SAGA$_{BC}$.

In a future experiment, we will add more effective functions to the algorithm, to ensure that it has a variety of uses.

Acknowledgment

This work was supported by JST-Mirai Program Grant Number JPMJMI17D3, Japan.

References

[1] W. Meng, and W. Schlitt, Sender policy framework (SPF) for authorizing use of domains in e-mail, version 1. No. RFC 4408. 2006.

[2] A. Ramachandran, N. Feamster, and D. Dagon, Revealing Botnet Membership Using DNSBL Counter-Intelligence. SRUTI 6. pp. 49-54. 2006.

[3] L. Barry and J. Fenton, DomainKeys Identified Mail (DKIM): Using Digital Signatures for Domain Verification. CEAS. 2007.

[4] G. C. Pieters, The Potential Impact of Decentralized Virtual Currency on Monetary Policy, Globalization and Monetary Policy Institute 2016 Annual Report. 2017.

[5] J. Herbert and A. Litchfield, A Novel Method for Decentralised Peer-to-peer Software License Validation Using Cryptocurrency Blockchain Technology, Proceedings of the 38th Australasian Computer Science Conference (ACSC 2015), Vol. 27, 2015.

[6] S. Nakamoto, Bitcoin: A Peer-to-peer Electronic Cash System. 2008.

[7] G. J. Wood, Ethereum: A Secure Decentralised Generalised Transaction Ledger, Ethereum project yellow paper, 151, pp. 1-32, 2014.

[8] P. L. Serge, Cryptocurrency and E-money Should not be Conflated. Medium. 2017.

[9] S. Yunus, What is the Difference between a Cryptocurrency, a Digital Currency, and a Virtual currency?. QUORA. 2018.

[10] D. He, K. Habermeier, R. Leckow, V. Haksar, Y. Almeida, M. Kashima, N. K. Saad, H. Oura, T. S. Sedik, N. Stetsenko, and C. V. Yepes, Virtual Currencies and Beyond: Initial Considerations, IMF STAFF DISCUSSION NOTE, SDN, 16 (3), 2016.

[11] A. Hayes, What factors give cryptocurrencies their value: An empirical analysis. 2015.

[12] A. K. M. Meera, Cryptocurrencies From Islamic Perspectives: The Case Of Bitcoin, Buletin Ekonomi Moneter Dan Perbankan, Vol.20, No.4, pp.443-460, 2018.

[13] M. Omri, A conceptual framework for the regulation of cryptocurrencies, U. Chi. L. Rev. Dialogue, Vol. 82, pp. 53-68, 2015.