Decentralized Content Vetting in Social Network with Blockchain

Chapter · October 2021

DOI: 10.1002/9781119790839.ch12

2 authors:

Subhasis Thakur
National University of Ireland, Galway

John G Breslin
National University of Ireland, Galway

12

Decentralized Content Vetting in Social Network with Blockchain

Subhasis Thakur and John G. Breslin

National University of Ireland, Galway, Galway H91 TK33, Ireland

12.1 Introduction

Fake news and misinformation in online social network (OSN) such as Facebook and Twitter are a major financial, social, and political risk [1]. Social bots [2, 3] can facilitate the creation and propagation of fake news in social network. All major social network platforms have recognized the presence of numerous social bots. Malicious entities can create problems in financial, social, and political sectors with fake news.

Financial Markets: Fake news can be used to impact trading in stock exchanges. For example, misinformation regarding US politics had caused a major shock in the US stock market [4]. This misinformation caused a loss of US$17 billion per year for the US retirement savings sector.

Healthcare Systems: Fake news can spread misinformation during a pandemic such as Covid-19. Fake news can encourage people not to participate in vaccination. False information on vaccination costs US$9 billion per year [1].

Political Systems: Fake news is now a political tool. There are numerous examples where fake news was used to gain political advantages by shaping public opinion with false information. Political parties are nowadays hiring companies with expertise in spreading false information on the social network. It is estimated that US$400 million are spent every year on generating fake news for political influence maximization.

Detecting rumor and the source of the rumor is well researched, and several algorithms are developed [5, 6]. Rumor detection techniques use machine learning-based solutions. Identifying the source of rumor [7, 8] can also be an effective tool to deter malicious entities from creating misinformation in social network. However, most of these solutions are centralized solutions. OSNs such as Facebook, Twitter, etc., are centralized platforms. Such centralized platform operators can employ algorithms to detect and prevent misinformation. However, being a centralized entity, an OSN platform operator may be biased and selectively remove misinformation. There are several biased content vetting incidents [9, 10]. A biased content vetting may reduce the credibility and revenue of a social network. In this chapter, we develop a blockchain-based decentralized content vetting for centralized social network.

Wireless Blockchain: Principles, Technologies and Applications, First Edition.
Bin Cao, Lei Zhang, Mugen Peng, and Muhammad Ali Imran.

We use proof of work-based public blockchain (Bitcoin) as the underlying blockchain to execute the decentralized vetting procedure. However, public blockchains have scalability problems. Hence, we use the offline channel network to execute the vetting algorithm. In this vetting procedure, all users get a chance to vote for and against content. If content continuously receives more positive votes, then it continues to propagate. We use token transfer methods for the offline channel network to execute such a voting procedure. Our main contributions in this chapter are as follows:

Unidirectional Offline Channel Model: We have developed an unidirectional offline channel for Bitcoin where a peer can send only a finite number of transactions to another user. Such an offline channel allows us to develop a secure voting mechanism where a user cannot manipulate the voting system.

High-Scale Vetting with Offline Channels: We have developed a high-scale vetting procedure using blockchain offline channels that significantly reduces the number of transactions of the blockchain network.

Channel Network Topology: We developed a method to reduce the number of offline channels needed to execute the vetting procedure.

Evaluation: We prove the efficiency of the content vetting solution using experimental evaluation.

The chapter is organized as follows: In Section 12.2, we discuss the related literature, in Section 12.3, we describe the content propagation model, in Section 12.4, we discuss the decentralized content vetting solution, in Section 12.5, we discuss the method to reduce the number of channels needed to execute the vetting procedure, in Section 12.6, we discuss the social network content propagation simulation algorithms, in Section 12.7, we discuss an experimental evaluation of the content vetting solution, and conclude the chapter in Section 12.8.

12.2 Related Literature

Detecting rumor and the source of the rumor is well researched, and several algorithms are developed [5, 6] to detect rumors. Rumor detection techniques have used machine learning-based algorithms. Identifying the source of rumor [7, 8] can also be an effective tool to deter malicious entities from creating misinformation in social network. In this chapter, we develop a rumor prevention mechanism using content vetting by the users.

Blockchain is recently applied to design several social media platforms. SteemIt [11] is a blockchain-based online social media platform that rewards its users for creating and rating new content. SteemIt uses Steem [12], which uses delegated proof of work [13] and is more scalable than a proof of work-based blockchains. Lit [14] is a blockchain-based social network platform that is developed using Ethereum. Users are rewarded for creating content in this social media platform, and the amount of reward depends on the popularity of the content. Sapien [15] is a blockchain-based (Ethereum) social network platform designed to deter false news. A user can join the Sapien platform by locking funds into a smart contract and it may lose these funds if it generates a false content. SocialX [16] is a decentralized social media platform designed to deter fake users from a social media

platform. SocialX uses Ethereum as the blockchain. Users are rewarded for checking the validity of media content. Foresting [17] is a blockchain-based social media platform where users are rewarded for creating valuable content and the usefulness of content is judged by users of the Foresting network. Minds [18] is an Ethereum-based social media platform that guarantees that there is no censorship of the content created in this social media platform. Decentralization of the social media platform immunes content from censorship. Minds platform uses both on-chain and off-chain transactions. Guidi [19] presented a detailed characterization of these social media platforms. Jiang and Zhang [20] developed a blockchain-based decentralised social network (DSN). In this social network, user data are kept in the blockchain and a user can modify and delete its data. Additionally, this DSN uses attribute-based encryption to preserve the privacy of the users, and as such, encryption allows access to only a subset of the user data. In [21], a blockchain and IPFS-based DSN model was proposed. It uses Ethereum smart contracts to develop DSN functionalities. Ur Rahman et al. [22] developed a blockchain-based DSN with Ethereum as the blockchain. It uses Ethereum smart contracts for access control over the user data in this DSN. Bahri et al. [23] analyses the security and privacy challenges in developing a DSN platform. Fu and Fang [24] used blockchains for privacy-preserving data management in social network. Yang et al. [25] provides a survey on blockchain-based social network and social media. Guidi et al. [26] analyses reward models for users in DSN. Freni et al. [27] discusses how blockchain can solve privacy and security problems with OSN. Yang et al. [28] proposed a blockchain-based secure friend matching algorithm for OSN.

In this chapter, we use proof of work-based blockchains. It was proposed in [29]. There are several variations of blockchains in terms of consensus protocols. Applications of these various types of blockchains are in various application areas such as energy trade [30], IoT service composition [31], etc. Bitcoin lightning network was proposed in [32], which allows peers to create and transfer funds among them without frequently updating the blockchain. Similar networks were proposed for Ethereum [1] and credit networks [33]. A privacy-preserving payment method in the credit network was proposed in [34]. A routing algorithm for the Bitcoin lightning network was proposed in [35].

Our contributions advance the state of the art in securing social network in the following directions:

- We have developed a content vetting method that allows the users of a social network to evaluate the validity of social media content. The proposed method can securely record such evaluation in a blockchain. It ensures that the evaluation of a user cannot be overwritten.
- We advance state of art in designing social network operations with blockchains by executing social network operation in blockchain offline channels. It improves the scalability of the solution.

12.3 Content Propagation Models in Social Network

There are several models of content propagation in social network [36]. In this chapter, we will use the influence maximization model [37] of content propagation. In this model, a

user v_i will share content with its neighbors if it has received the content from at least Δ fraction of its neighbors. $\Delta \in [0, 1]$ is chosen by the user.[1] Δ shows the difficulty to influence a user. We will use the following model of content propagation:

1. Let user B is the creator of content in the social network.
2. User A's decision to share or not share content is as follows:
   (a) If the distance (length of the shortest path between A and B) between A and B is less than λ then A, any neighbor of A can send the content to A and A can share the content if A considers the content as correct information.
   (b) Else, with a fixed probability, A is influenced by its neighbors.
   (c) If A is not influenced, then it makes its own decision regarding the validity of the content. Otherwise, if the number of neighbors who had shared this content is more than a threshold, then A will share the content with its neighbors. Otherwise, it will wait until such several neighbors share the same content (Figure 12.1).

In this content propagation model, an adversarial user will create and share content that will be considered rumor or misinformation by other users (Figure 12.2). We will use the following model of an adversarial user:

1. An adversarial user will create a rumor or misinformation.
2. An adversarial user will share a rumor or misinformation.
3. An adversarial user will share a rumor or misinformation irrespective of how many of its neighbors have shared it.

Social bots can be used by the adversarial user to propagate misinformation and prevent the propagation of correct information. Social bots can be part of the social neighborhood of genuine users. Additionally, social bots may use malware to control information to and from a genuine user. We assume that the adversarial user can control only a finite number of users

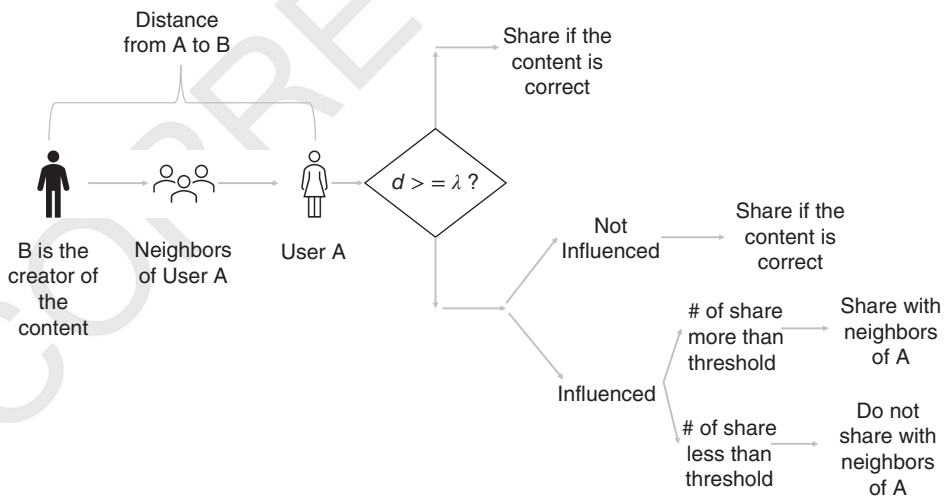

Figure 12.1 Content propagation model.

1 This value represents the likelihood that a user can be influenced by its neighbors.

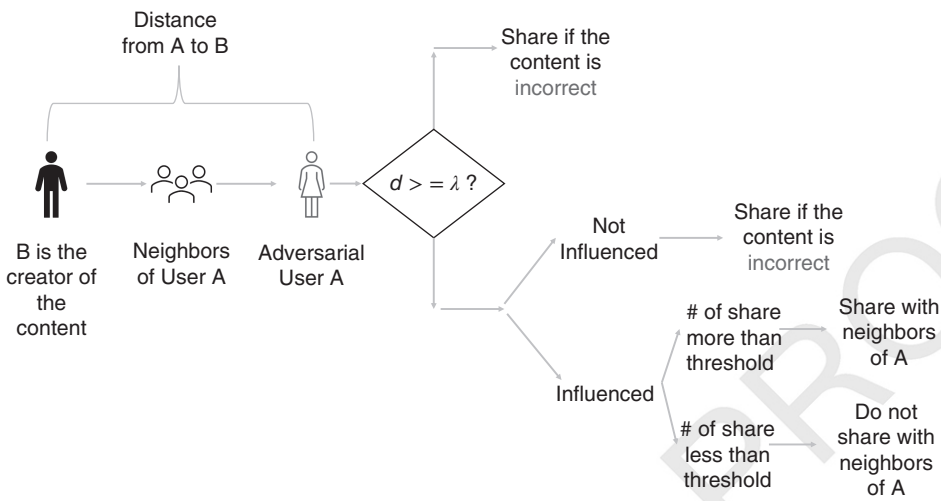

Figure 12.2 Behavior of an adversarial user.

in a social network and there is a cost associated with the number of users the adversarial user can control. $C(k) \in \mathcal{R}^+$ be the function indicating the cost of controlling a fraction k ($k \in [0, 1]$) of social network users. The utility of the adversarial user if misinformation spreads to k fraction of social network users is $U^-(k) \in \mathcal{R}^+$, and utility of the adversarial user if correct information spreads to k fraction of social network users is $U^+(k) \in \mathcal{R}^+$.

12.4 Content Vetting with Blockchains

12.4.1 Overview of the Solution

We use blockchains to develop a decentralized content vetting procedure. It is as follows:

1. Each user of the blockchain network is assumed to be part of a public blockchain network.
2. In the proposed social media sharing procedure, A can share content with B if (a) A can produce a self-attestation that A has examined the content and considered it correct, (b) A can produce a neighborhood content vetting, i.e. similar self-attestation from the neighbors of A.
3. The self-attestation is a random string S for which Hash of S is recorded in the blockchain and anyone can verify the existence of $H(S)$. Self-attestation of a social media content by any user A can be the string (Figure 12.3 S.
4. Blockchains ensure that a user cannot reuse the self-attestation for multiple content i.e. one content and one vote.

12.4.2 Unidirectional Offline Channel

Blockchain offline channels [32] uses multi-signature addresses to open an offline channel among peers of the blockchain. This offline channel [32] is bidirectional and potentially

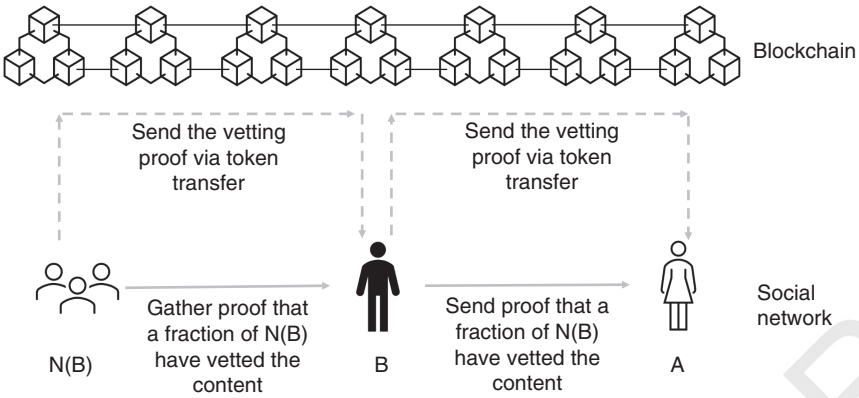

Figure 12.3 Overview of the decentralized vetting procedure.

infinite, i.e. it can execute the infinite number of transfers between two peers provided they do not close the channel and each of them has sufficient funds. We construct an offline channel for proof of work-based public blockchain with the following properties:

- We construct a unidirectional channel between two peers, i.e. only one peer can send funds to another peer of this channel.
- We construct a unidirectional channel that can be used for a finite number of transfers from a designated peer to another peer.

The procedure for creating the unidirectional channel from A to B (A transfers token to B) is as follows: Let A and B are two peers of the channel network H. $M_{A,B}$ is a multi-signature address between A and B. This is a unidirectional channel from A to B.

1. A creates a set of k (k is a positive even integer) random strings $S_A^1, \dots, S_A^k$. Using these random strings, A creates a set of Hashes $H_H^1 = H(S_B^1), H_B^2 = H(S_B^1), \dots, H_B^k = H(S_B^k)$, where H is the Hash function (using SHA256). A creates a Merkle tree order λ using these Hashes. Thus, there are k leaf nodes and $k-1$ non-leaf nodes of this Merkle tree. We denote the non-leaf nodes as $H_A'^1, \dots, H'(k-1)_A$.
2. B creates a set of $k1$ random strings $S^1, \dots, S^k$ and corresponding Hashes $H_B^1, \dots, H_B^k$.
3. A sends the Merkle tree to B and B sends the set of Hashes $H_B^1, \dots, H_B^k$ to A.
4. A sends a Hashed time-locked contract $HTLC_A^1$ to B as follows:
   (a) From the multi-signature address $M_{A,B}$, 1 token will be given to A after time T if B does not claim these tokens before time T by producing the key to $H_A'^1$ and 0 token will be given to A if it can produce the key to H_B^1.
   (b) A sends $HTLC_A^1$ to B.
5. Now, A sends 1 token to $M_{A,B}$. A includes the Merkle tree and $H_B^1, \dots, H_B^k$ in this transaction. This records the Merkle tree and $H_B^1, \dots, H_B^k$ in the blockchain and any other peer can verify the existence of these Hashes by checking transactions of the public blockchain. Also, at this stage, A's funds are safe as it can get the tokens from $M_{A,B}$ after time T as B does not know $H_A'^1$.

6. Next to send another $(1/k)$ tokens to B, A sends S_A^1 to B and B sends H_B^1 to A. Then, A forms the following HTLC:
   (a) From the multi-signature address $M_{A,B}$, $1 - 1/k$ token will be given to A after time T if B does not claim these tokens before time T by producing the key to $H_A^{'2}$ and $1/k$ token will be given to A if it can produce the key to H_B^2.
   (b) A sends $HTLC_A^2$ to B.
7. This process continues until all keys of the Hashes of non-leaf nodes are revealed by A.

In this model of the unidirectional channel, A is sequentially releasing the keys of the Merkel tree of the HTLCs. Its fund in this channel is decreasing with time. It cannot prevent B from obtaining the tokens as only B can publish the HTLCs. B will publish the HTLC where it gets the maximum value.

12.4.3 Content Vetting with Blockchains

We will use the unidirectional channels described in the previous section to execute the content vetting procedure. It is as follows: We assume that all users of the social network are peers of a blockchain network. We use proof of work-based public blockchains, i.e. Bitcoin. Each user will establish an offline channel with all of its neighbors in the social network. All such channels are unidirectional channels. All channels are marked (such information can be included in the transaction funding the multi-signature address to start the channel) as a positive or negative ballot. Such marking can be verified by any user by checking the transaction record of the blockchain and such marking cannot be changed due to the immutability of blockchain transactions. Briefly, the content vetting procedure is as follows:

1. A user needs to share proof of vetting and proof of neighborhood vetting with its immediate neighbors to share the content.
2. A user's proof of content vetting can be bought by its neighbor by paying the user via a unidirectional channel between them. Such proof is an unknown key of a Hash recorded in the unidirectional channel. For example, (Figure 12.4) A can pay B for content vetting using the channel between them. It will cost A $1/k$ tokens and the most recent key revealed by B for the Hashes $H_B^1, H_B^2, \ldots$, will be regarded as the proof of content vetting.
3. A user's proof of neighborhood vetting can be bought by its neighbor by paying the user via a unidirectional channel between them. Proof of neighborhood vetting is the set of content vetting a user has already bought from its neighbors. A proof of neighborhood vetting can be considered valid by a user if the number of neighbors whose content vetting are included in the neighborhood vetting is more than US50%.
4. Further, proof of content and neighborhood vetting can be categorized as positive and negative voting. Proof of content vetting is regarded as positive voting if the corresponding key belongs to a channel marked as a positive ballot.
5. A rational user will only pay for content vetting if it can sell such vetting information.

We will explain the content vetting procedure with an example. We will use the influence maximization model [37] of content propagation. According to this model, neighbors

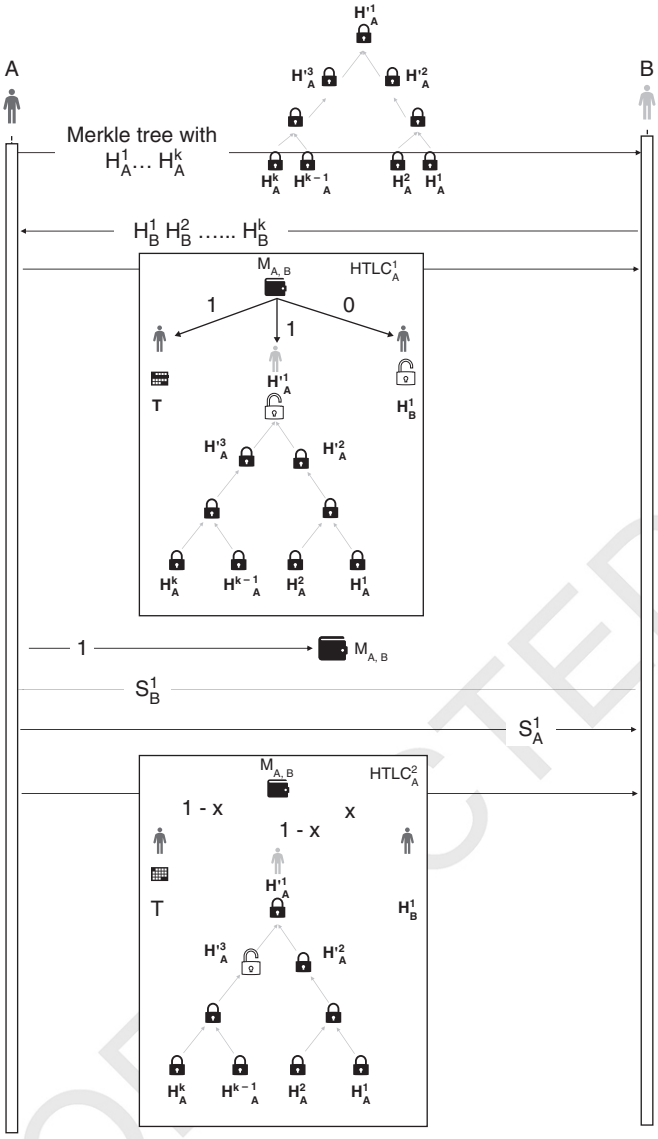

Figure 12.4 Procedure of creating unidirectional offline channels.

of user A can share content with A. If the number of neighbors who had shared the content is more than a fixed threshold, then A may be influenced by its neighbors and may share the content with its other neighbors. The proposed content vetting procedure works similarly. However, a user should produce proof of content vetting and proof of neighborhood vetting while sharing a content with its neighbors. Consider the following scenario (shown in Figure 12.5): Let the creator of content is V_0, v_1 is the neighbor of v_0, v_2 is the neighbor

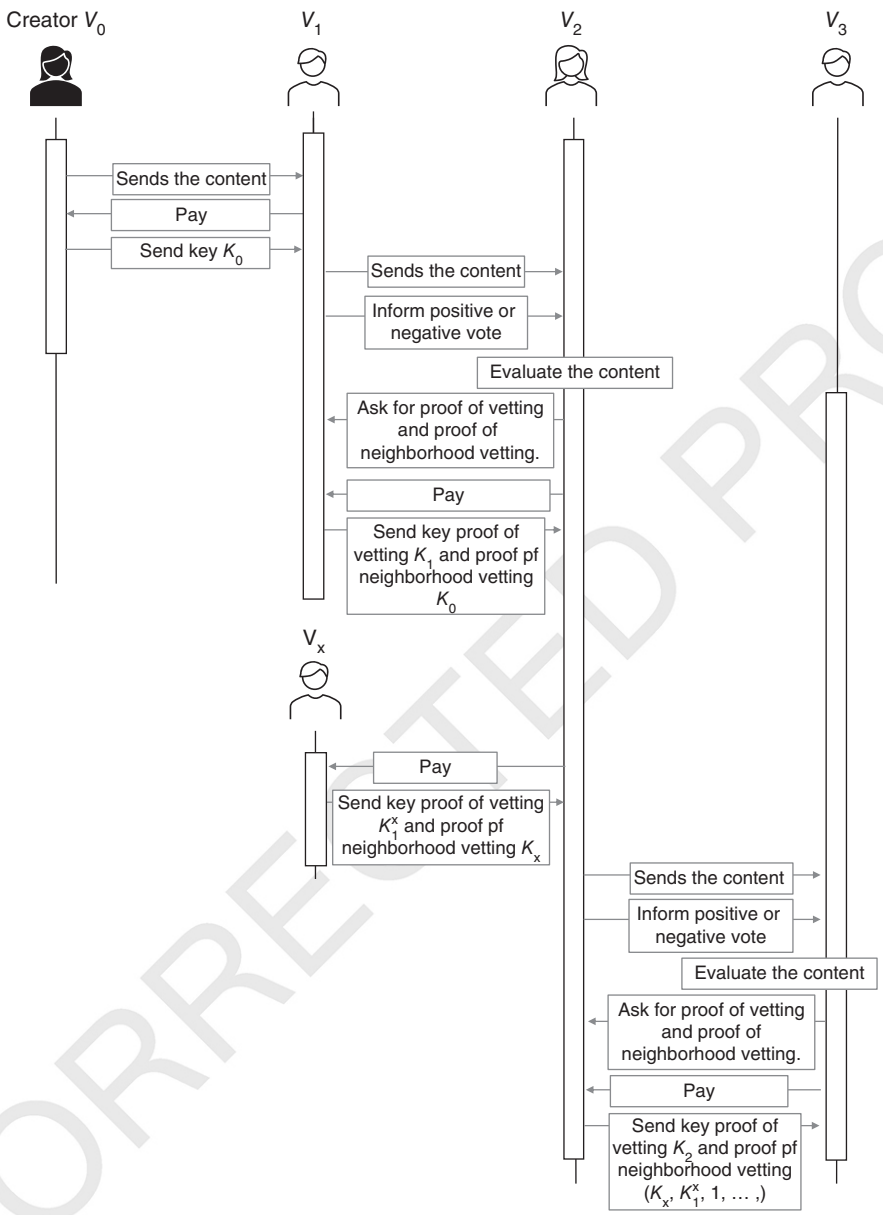

Figure 12.5 Content vetting procedure.

of v_1, and v_3 is the neighbor of v_2. Propagation of the content from v_0 to v_2 with content vetting is as follows:

1. From v_0 to v_1:
   (a) v_0 sends the content to v_1.
   (b) v_1 pays v_0 $1/k$ tokens to get the key k_0 using the unidirectional channel from v_1 to v_0.

2. From v_1 to v_2:
   (a) v_1 sends the content v_2.
   (b) v_1 informs if it will vote positive or negative to v_2.
   (c) v_2 evaluate the content.
   (d) If v_2 evaluate that the content is misinformation (correct) and v_1 informed that it will send a negative (positive) vote then, it asks v_1 to send proof of vetting and proof of neighborhood vetting.
   (e) v_1 sends the proof of vetting by sending the key k_1 in the channel from v_2 to v_1.
   (f) v_2 sends $1/k$ tokens to v_1 using the channel from v_2 to v_1.
   (g) v_1 sends k_0 to v_2 as proof of neighborhood vetting.
3. From v_2 to v_3:
   (a) v_2 sends the content v_3.
   (b) v_2 informs if it will vote positive or negative to v_3.
   (c) v_3 evaluate the content.
   (d) If v_3 evaluate that the content is misinformation (correct) and v_2 informed that it will send a negative (positive) vote then, it asks v_2 to send proof of vetting and proof of neighborhood vetting.
   (e) v_2 sends the proof of vetting by sending the key k_2 in the channel from v_3 to v_2.
   (f) v_3 sends $1/k$ tokens to v_1 using the channel from v_3 to v_2.
   (g) v_3 sends $k_1, k_1^x, \ldots$ (total 0.5λ of such keys from its neighbors) to v_3 as proof of neighborhood vetting.

Note that,

1. Uniqueness of content vetting is guaranteed as an unidirectional channel between two users is updated every time one user asks and pays for content vetting.
2. A user can check the existence of Hashes of the keys presented as neighborhood vetting in the public blockchain. Also, as an unidirectional channel can be used for a finite number of transfers, it ensures that old keys cannot be used as proof of neighborhood vetting.
3. A user pays for the content and neighborhood vetting from its neighbor. A rational user will only do so if it can sell such information to recover such a cost. This means if a user evaluates that content is misinformation and its neighbor is willing to provide a positive vote for it, then it will not buy the content and neighborhood vetting. Similarly, a user will not buy content vetting and neighborhood vetting where the neighbor informed that it is willing to provide a negative vote if it does not consider the content as misinformation.
4. We will show that users who buy negative(positive) vote while the content is correct (incorrect) will have too low funds to buy and share content vetting.

12.5 Optimized Channel Networks

In the above-mentioned content vetting mechanism, we assumed the offline channel network among all users of a social network. It may be difficult to build such a channel network as the number of social neighbors for a user may be too high, and establishing a

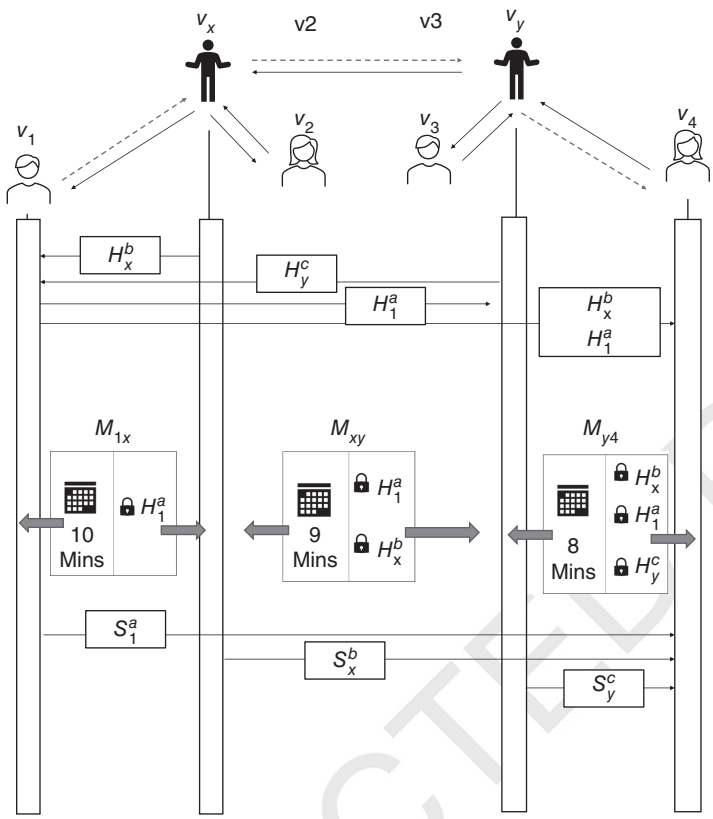

Figure 12.6 Optimized content vetting procedure.

channel will require certain funds in terms of tokens of a blockchain network. We mitigate this problem as follows: We assume that the operator of the social network is represented by multiple nodes of a blockchain network, i.e. the social network operator registers multiple accounts in a blockchain network. A user can establish unidirectional offline channels to and from such peers of the blockchain network representing social network operators. Consider the scenario as shown in the figure in 12.6, v_1 shares unidirectional channels with v_x, v_4 shares unidirectional channels with v_y. v_x and v_y are peers representing the social network operators. v_1 can send proof of content vetting to v_4 as follows (Figure 12.6):

1. $v1$ collects next Hash H_x^b to be used in updating the channel $v_x \rightarrow v_y$.
2. $v1$ collects next Hash H_y^c to be used in updating the channel $v_y \rightarrow v_4$.
3. Let H_1^a be next Hash H_y^c to be used in updating the channel $v_1 \rightarrow v_x$.
4. v_1 informs v_x, v_y, v_4 about all these Hashes.
5. Next, 3 HTLCs are created as follows:
   (a) $HTLC^{1,x}$ states that $1/k$ tokens will be given to v_1 from $M_{1,x}$ (multi-signature address between v_1 and v_x) after 10 minutes unless v_x claims these tokens before 10 minutes by producing the key S_1^a.

(b) $HTLC^{x,y}$ states that $1/k$ tokens will be given to v_x from $M_{x,y}$ (multi-signature address between v_x and v_y) after 9 minutes unless v_y claims these tokens before 9 minutes by producing the keys S_1^a and S_x^b.
(c) $HTLC^{y,4}$ states that $1/k$ tokens will be given to v_y from $M_{y,4}$ (multi-signature address between v_y and v_4) after 8 minutes unless v_4 claims these tokens before 8 minutes by producing the keys S_1^a, S_y^c and S_x^b.

6. After constructing these HTLCs, v_1 sends the key S_1^a to v_4, v_x sends the key S_x^b to v_4, and v_y sends the key S_y^c to v_4.
7. v_4 starts sequential execution of the HTLCs and v_1 gets paid by v_4 via v_x and v_y.
8. In this case, the proof of vetting will be the all keys used in the path from v_1 to v_4, i.e. S_1^a, S_y^c and S_x^b.

The proposed content vetting solution can prevent social bots from spreading rumor or obstructing propagation of correct news:

- A user A (not a social bot or controlled by an adversarial user who wants to spread rumor and prevent propagation of correct information) can verify self-attestation of content vetting and neighborhood vetting by checking if Hash of such proofs exists in the blockchain.
- The same user A can also verify the uniqueness of proof of content vetting and proof of neighborhood vetting, i.e. the same keys are not shared by its neighbors.
- A social bot can bypass the content vetting procedure if A does not follow the content vetting procedure, i.e. does not check for proof of content vetting and proof of neighborhood vetting. However, in this case, A may not be able to spread the rumor as it has not collected proof of neighborhood vetting.

12.6 Simulations of Content Propagation

We will use agent-based modeling of a social network. We will use two sets of simulations, one for modeling the propagation of misinformation and another for the propagation of correct information. The first simulation on the propagation of misinformation (shown in Algorithm 12.1) is as follows:

1. Let $\mathbb{M}$ be the message list of the users, $\mathbb{T}$ be the trust, Δ be the threshold, A be the set nodes controlled by the adversarial node, *Frd* the number of users who have forwarded the news, and *vlen* the number of users.
2. At every iteration of the simulation, each user's behavior is as follows:
3. If user i has not sent the social media content, then let d be its distance from the creator of the content. i can check if the creator of the content $\mathbb{S}$ is in its immediate social neighborhood.
4. If $\mathbb{S}$ is not in its immediate social neighborhood, then i will follow the following steps:
   (a) If i is not controlled by the adversarial user then, if the weighted number of neighbors who have sent this content to i (calculated using the trust of i on its neighbors) is more than the threshold $\Delta(i)$ then, with a fixed probability, i will be influenced by its neighbors. In this case, neighbors can influence i to share the content.
   (b) If i is controlled by the adversarial user then, it will share the content with its neighbors.

Algorithm 12.1: Propagation of Misinformation.

```
Data: g, 𝕋, Δ, 𝕊, 𝔸
Result: Spread = Number of users who had shared the news.
begin
    𝕄 = [0, n × n], 𝕄[N´(𝕊), 𝕊] ← 1, Frd ← [0, 1 × n], Frd[𝕊] ← 1,
     sent ← [0, 1 × n], sent[𝕊] ← 1, Spread ← [1]
    while Simulation is not stopped do
        for i ∈ [1 : n] do
            if sent[i] == 0 then
                if shortest.paths(𝕊, i) > 1&sent[i] == 0 then
                    n2 ← which(𝕄[i, ] > 0), y = ∑(𝕋[i, n2])
                    if |n2| > 0 then
                        if i ∉ 𝔸 then
                            if y > Δ[i] then
                                if Random(1) > .7 then
                                    Frd[i] ← 1, sent[i] ← 1, n3 ← N′(i),
                                     𝕄[n3, i] ← 1
                                else
                                    sent[i] ← 1
                        else
                            sent[i] ← 1, n3 ← N´(i), 𝕄[n3, i] ← 1,
                             Frd[i] ← 1
                if shortest.paths(𝕊, i) == 1&sent[i] == 0 then
                    if |which(𝕄[i, ] > 0)| > 0 then
                        if i ∉ 𝔸 then
                            n1 ← N(i), n2 ← which(𝕋[i, ] > 0),
                            mean1 ← mean(𝕋[i, n2])
                            if 𝕋[i, start1] > mean1 then
                                if runif(1) > .7 then
                                    sent[i] ← 1, n3 ← N´(i), 𝕄[n3, i] ← 1,
                                    Frd[i] ← 1
                                else
                                    sent[i] ← 1
                        else
                            sent[i] ← 1, n3 ← N´(i), 𝕄[n3, i] ← 1,
                             Frd[i] ← 1
        Add ∑Frd to Spread
    Return(Spread)
```

5. If $\mathbb{S}$ is in its immediate social neighborhood, then i will follow these steps:
   (a) If i is not controlled by the adversarial user then, if i's trust in the creator of the content is more than i's average on trust on its neighbors, then with a fixed probability, i will be influenced by the creator of the content and it will share the content.
   (b) If i is controlled by the adversarial user then, it will share the content with its neighbors.
6. Simulation records the number of users who share the content with its neighbors for every iteration.

The second simulation on the propagation of misinformation with content vetting (shown in Algorithm 12.2) is as follows:

1. Let $\mathbb{P}$ be the message list of the users with a positive vote for the social media content, $\mathbb{N}$ be the message list of the users with a positive vote for the social media content, $\mathbb{T}$ be the trust, Δ be the threshold, A be the set nodes controlled by the adversarial node, *Frd* the number of users who have forwarded the news, and n the number of users.
2. At every iteration of the simulation, each user's behavior is as follows:
3. If user i has not sent the social media content, then let d be its distance from the creator of the content. i can check if the creator of the content $\mathbb{S}$ is in its immediate social neighborhood.
4. If $\mathbb{S}$ is not in its immediate social neighborhood, then i will follow these steps:
   (a) If i is not controlled by the adversarial user then, if the weighted ratio between negative and positive votes (weight as trust value) is less than the threshold $\Delta(i)$ then, with a fixed probability i will be influenced by its neighbors. In this case, neighbors can influence i to share the content (and it will send positive votes to such neighbors). Otherwise, it will send negative votes about the content to its neighbors.
   (b) If i is controlled by the adversarial user then, if it will share the content with its neighbors only if the weighted ratio between negative and positive votes is less than the threshold. This is because i needs to prove to its neighbors that it has such a ratio of weighted negative and positive votes.
5. If $\mathbb{S}$ is in its immediate social neighborhood, then i will follow these steps:
   (a) If i is not controlled by the adversarial user then, if i's trust in the creator of the content is more than i's average on trust on its neighbors then with a fixed probability i will be influenced by the creator of the content and it will share the content.
   (b) If i is controlled by the adversarial user then, it will share the content with its neighbors.
6. Simulation records the number of users who share the content with its neighbors for every iteration.

The third simulation on the propagation of correct information (shown in Algorithm 12.3) is as follows:

1. It is similar to the first simulation except:
2. If i is controlled by the adversarial user, then i will not share the content.

The fourth simulation on the propagation of correct information with content vetting (shown in Algorithm 12.4) is as follows:

1. It is similar to the second simulation except:
2. If i is not controlled by the adversarial user and if the weighted ratio between negative and positive votes is less than the threshold, then i will share the content.

Algorithm 12.2: Propagation of Misinformation with Content Vetting.

```
Data: g, T, Δ, 𝕊, 𝔸, ℙ, ℕ
Result: Spread = Number of users who had shared the news.
begin
    ℙ = [0, n × n], ℕ = [0, n × n] ℙ[N′(𝕊), 𝕊] ← 1Frd ← [0, 1 × n)  Frd[𝕊] ← 1
    sent ← [0, 1 × n] sent[𝕊] ← 1 Spread ← [1]
    while Simulation is not stopped do
        for i ∈ [1 : n] do
            if sent[i] == 0 then
                if shortest.paths(𝕊, i) > 1&sent[i] == 0 then
                    n2 ← which(ℙ[i, ] > 0), y ← sum(𝕋[i, n2]),
                     n21 ← which(ℕ[i, ] > 0), y1 ← ∑(𝕋[i, n21])
                     if |n2| > 0 then
                         if i ∈ 𝔸 then
                             if (y1/y) < Δ[i] then
                                 if Random(1) > .7 then
                                     Frd[i] ← 1, sent[i] ← 1, n3 ← N´(i),
                                      ℙ[n3, i] ← 1
                                 else
                                     sent[i] ← 1, n3 ← N′(i), ℕ[n3, i] ← 1
                         else
                             if (y1/y) < .3 then
                                 sent[i] ← 1, n3 ← N′(i), ℙ[n3, i] ← 1,
                                  Frd[i] ← 1
                if d == 1&sent[i] == 0 then
                    if |(which(ℙ[i, ] > 0))| > 0 then
                        if i ∉ 𝔸 then
                            n2 ← which(𝕋[i, ] > 0)
                             mean1 ← mean(𝕋[i, n2])if 𝕋[i, 𝕊] > mean1 then
                                if random(1) > .7 then
                                    sent[i] ← 1 n3 ← N′(i) ℙ[n3, i] ← 1
                                     Frd[i] ← 1
                                else
                                    sent[i] ← 1, n3 ← N′(i), ℕ[n3, i] ← 1
                        else
                            sent[i] ← 1, n3 ← N′(i), ℙ[n3, i] ← 1,
                             Frd[i] ← 1
        Add ∑(Frd) to Spread.
    Return(Spread)
```

Algorithm 12.3: Propagation of Correct Information.

```
Data: g, T, Δ, 𝕊, 𝔸
Result: Spread = Number of users who had shared the news.
begin
  𝕄 = [0, n × n], 𝕄[N′(𝕊), 𝕊] ← 1, Frd ← [0, 1 × n], Frd[𝕊] ← 1,
    sent ← [0, 1 × n], sent[𝕊] ← 1
  while Simulation is not stopped do
    for i ∈ [1 : n] do
      if sent[i] == 0 then
        d ← shortest.paths(g, 𝕊, i)
        if d > 1&sent[i] == 0 then
          n1 ← N(i) n2 ← which(𝕄[i, ] > 0) y = ∑(𝕋[i, n2])
          if |n2| > 0 then
            if i ∈ 𝔸 then
              if y > Δ[i] then
                if runif(1) > .7 then
                  Frd[i] ← 1 sent[i] ← 1 n3 ← N′(i)
                  𝕄[n3, i] ← 1
                else
                  Frd[i] ← 1 sent[i] ← 1 n3 ← N′(i)
                  𝕄[n3, i] ← 1
            else
              sent[i] ← 1
        else
          if |which(𝕄[i, ] > 0)| > 0 then
            if i ∉ 𝔸 then
              n1 ← N(i) n2 ← which(𝕋[i, ] > 0)
               mean1 ← mean(𝕋[i, n2])
              if 𝕋[i, 𝕊] > mean1 then
                if runif(1) > .7 then
                  Frd[i] ← 1 sent[i] ← 1 n3 ← N′(i)
                  𝕄[n3, i] ← 1
                else
                  Frd[i] ← 1 sent[i] ← 1 n3 ← N′(i)
                  𝕄[n3, i] ← 1
            else
              sent[i] ← 1
      Add ∑(Frd) to Spread
    Return(Spread)
```

Algorithm 12.4: Propagation of Correct Information with Content Vetting.

```
Data: g, T, Δ, 𝕊, 𝔸
Result: Spread = Number of users who had shared the news.
begin
    ℙ = [0, n × n], ℕ = [0, n × n], ℙ[N(𝕊,"out"), 𝕊] = 1, F = [0, 1 × n], F[𝕊] = 1,
     s = [0, 1 × n], s[𝕊] = 1
    while Simulation is not stopped do
        for i ∈ [1 : n] do
            if s[i] == 0 then
                if shortest.paths(𝕊, i) > 1&s[i] == 0 then
                    n2 = which(ℙ[i, ] > 0), y = ∑(𝕋[i, n2]),
                     n21 = which(ℕ[i, ] > 0), y1 = ∑(𝕋[i, n21])
                    if |n2| > 0 then
                        if i ∉ 𝔸 then
                            if y1/y < Δ[i] then
                                if runif(1) > .7 then
                                    F[i] = 1, s[i] = 1 n3 = N′(i) ℙ[n3, i] = 1
                                else
                                    F[i] = 1, s[i] = 1, n3 = N′(i), ℙ[n3, i] = 1
                            else
                                if y + y1 > .3 × |N(i)| then
                                    s[i] = 1, n3 = N′(i), ℕ[n3, i] = 1
                        else
                            if ∑(ℙ[i, ]), ∑(ℕ[i, ]) > 0 & ∑(ℙ[i,])/∑(ℕ[i,]) < .14 then
                                s[i] = 1, n3 = N′(i), ℕ[n3, i] = 1
                else
                    if |which(ℙ[i, ] > 0)| > 0 then
                        if i ∉ 𝔸 then
                            n2 ← which(𝕋[i, ] > 0)
                            if 𝕋[i, 𝕊] > mean(𝕋[i, n2]) then
                                if Random(1) > .7 then
                                    F[i] = 1, s[i] = 1, n3 = N′(i), ℙ[n3, i] = 1
                                else
                                    F[i] = 1, s[i] = 1, n3 = N′(i), ℙ[n3, i] = 1
                        else
                            s[i] = 1, n3 = N′(i), ℕ[n3, i] = 1
            Add ∑(F) to Spread
```

3. If i is controlled by the adversarial user, then i will not share the content (i.e. it will send a negative vote regarding the content) if its fraction of negative and positive votes is more than the threshold and it has received votes from at least a fixed fraction of its neighbors.

Let $\mathbb{M}$ is the message list of the users, $\mathbb{T}$ is trust, Δ is the threshold, A is set nodes controlled by the adversarial node, *vlen* is the number of users, *Frd* is the number of users who have forwarded the news, $\mathbb{P}$ and $\mathbb{N}$ are messages with positive and negative vote, respectively, and *Spread* is the number of users who had shared the content.

12.7 Evaluation with Simulations of Social Network

We use Facebook network data from [38]. This social network is a directed graph with 4039 nodes and 88234 edges. We simulate content propagation in social network using the simulation algorithms shown in Section 12.6. We simulate a blockchain network using agent-based modeling of the blockchain network. We use an asynchronous event simulator (using the SIMPY library of Python). The workflow of each agent (who simulates a peer of the blockchain network) is as follows:

1. Each peer executes four processes in parallel.
2. $Process_3$ receives messages from its neighbors, and if the message is not received before, then it checks if the message contains a new transaction or a new block. If it receives a transaction, then it informs $Process_2$ about the new transaction. If it receives a new block, then it informs $Process_4$ about the new block.
3. $Process_2$ gathers new transactions from $Process_3$, and the new transaction is placed in a queue of undocumented transactions. We assume that the queue model is First In First Out. After adding the new transaction to its queue, a peer forwards the message containing the new transaction to its neighbors.
4. $Process_1$ empties the first k transactions from its queue of undocumented transactions and creates a new block. Then, it solves the puzzle of proof of work protocol and publishes the new block.
5. $Process_4$ examines the new block from $Process_3$, if all transactions of the new block are valid then: if the parent block of the new block is the last blockchain head known to the peer, then it augments its blockchain by placing the new block as child block of its last known blockchain head and recognize the new block as the last know blockchain head. Otherwise, it finds the parent block of the new block in its blockchain and augments the blockchain by adding the new block as its child block.
6. We simulate offline channel network as follows:
   (a) Two multi-signature addresses are needed to create an offline channel between two peers. We simulate such multi-signature address as lists. The initial and final balance of such lists are accessible by all peers, i.e. these lists are shared variables among all instances of the peer class.
   (b) Two peer exchanges messages to exchange HTLCs.

First, we simulate the propagation of incorrect social media content. We execute two sets of experiments. In the first set of experiments, we increase the number of users controlled

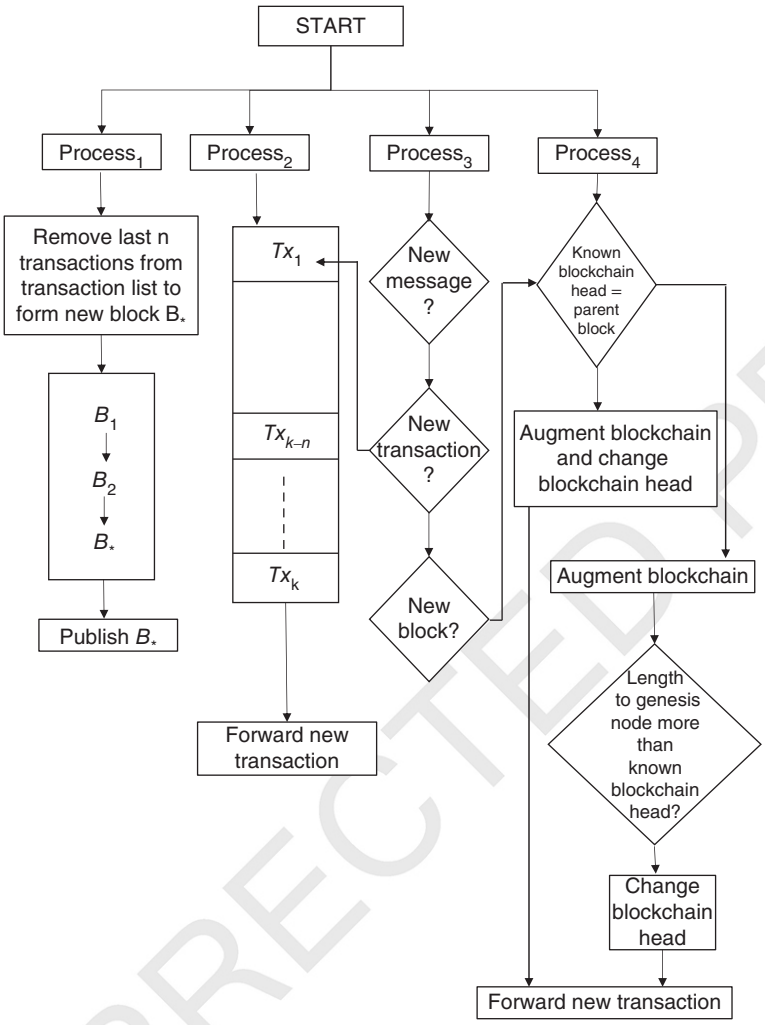

Figure 12.7 Workflow of each miner of the blockchain network.

by the adversarial user from 15% to 22% (1% in each increment) (Figure 12.7). The result of this set of simulations is shown in Figures 12.8 and 12.9. It shows that with content vetting, the incorrect content is shared by a less number of users and more quickly compared with content propagation without vetting. In the second set of experiments, we keep the number of users who are controlled by the adversarial user to 20% but we gradually increase the threshold from 0.3 to 0.65 (increment of 0.5 for each experiment). The result of this experiment is shown in Figures 12.10 and 12.11. It shows that with content vetting, the incorrect content is shared by a fewer number users compared with content propagation without vetting.

Next, we simulate the propagation of correct social media content. We execute two sets of experiments. In the first set of experiments, we keep the number of users who are

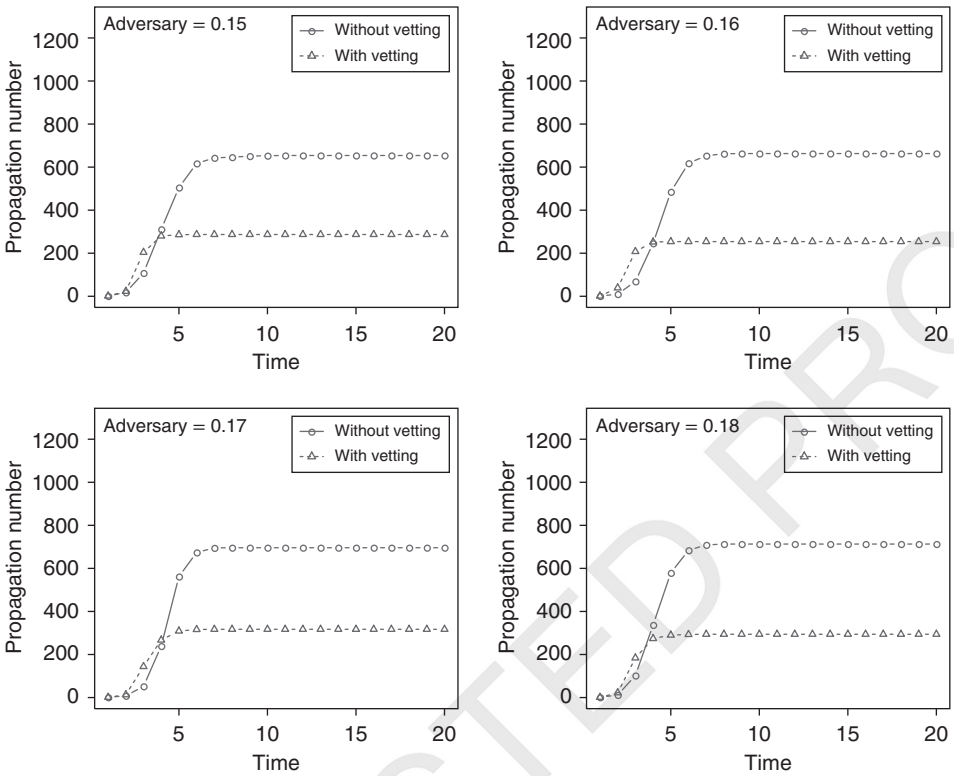

Figure 12.8 Propagation of incorrect information with increasing number of users controlled by the adversarial user. We increase the number of such users from 15% to 18%. It shows that with content vetting, the number of users who shared the incorrect information remains low. Also, it shows that the number of users who shared the incorrect information without content vetting increases as the number of users controlled by the adversarial user is increased.

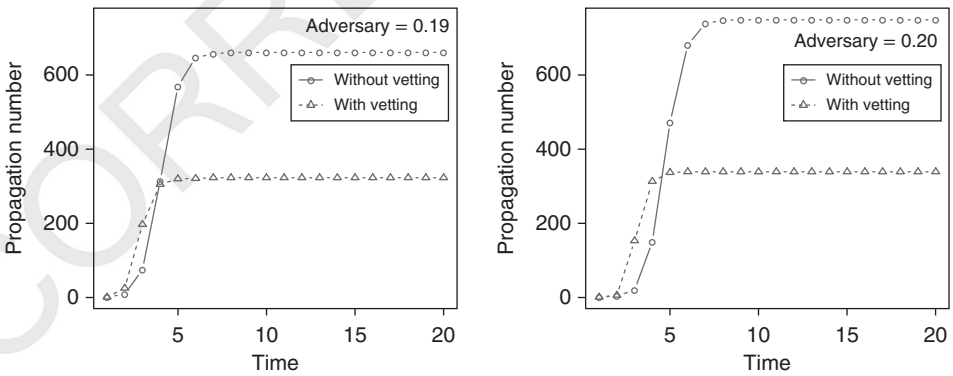

Figure 12.9 Propagation of incorrect information with increasing number of users controlled by the adversarial user. We increase the number of such users from 19% to 22%. It shows that with content vetting, the number of users who shared the incorrect information remains low. Also, it shows that the number of users who shared the incorrect information without content vetting increases as the number of users controlled by the adversarial user is increased.

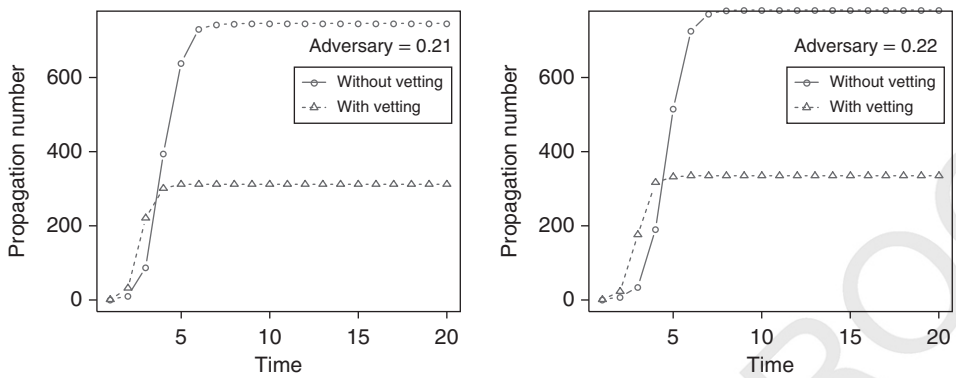

Figure 12.9 (*Continued*)

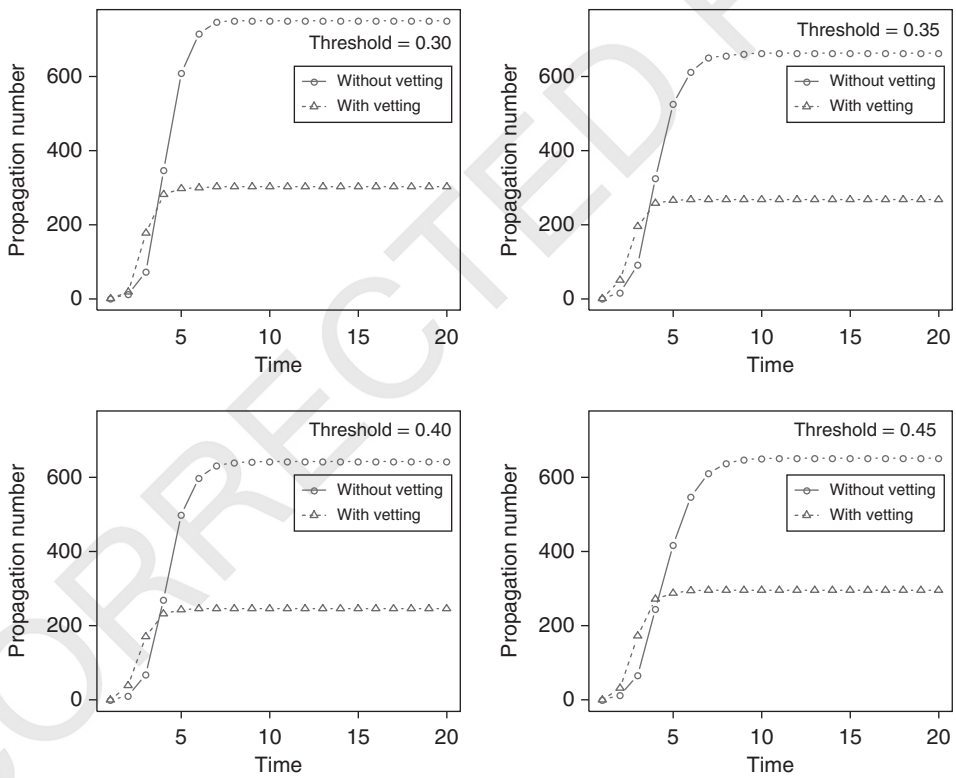

Figure 12.10 Propagation of incorrect information with increasing threshold of the users. We increase the threshold from 0.3 to 0.45. The threshold is the minimum weighted (calculated with using trust among the users) number of neighbors who had shared the content with the user. This experiment shows that content vetting reduces the number of users who shared incorrect information.

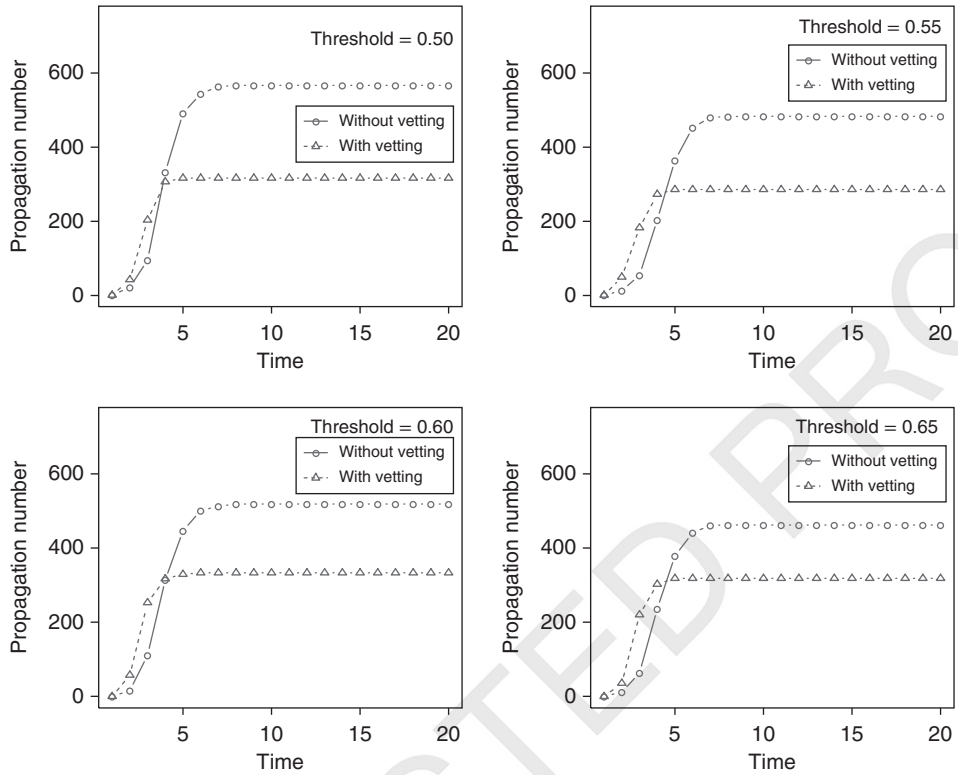

Figure 12.11 Propagation of incorrect information with increasing threshold of the users. We increase the threshold from 0.5 to 0.65. The threshold is the minimum weighted (calculated with using trust among the users) number of neighbors who had shared the content with the user. This experiment shows that content vetting reduces the number of users who shared incorrect information. It also shows that as the threshold is increased, the number of users who share the incorrect content is reduced. This is because as we increase the threshold, it becomes more difficult to influence a user.

controlled by the adversarial user to 20%, but we gradually increase the threshold from 0.3 to 0.55 (increment of 0.5 for each experiment). The result of this experiment is shown in Figures 12.12 and 12.13. It shows that with content vetting, the correct content is shared by more users and more quickly compared with content propagation without vetting.

In the second set of experiments, we keep the threshold level of the users constant (0.3), but we increase the number of users controlled by the adversarial user from 35% to 56% (3% in each increment). The results of this set of simulations are shown in Figures 12.14 and 12.15. It shows that with content vetting, the correct content is shared by more users and more quickly compared with content propagation without vetting. It also shows that an increment of the number of users controlled by the adversarial user significantly reduces the number of users who have shared the correct content.

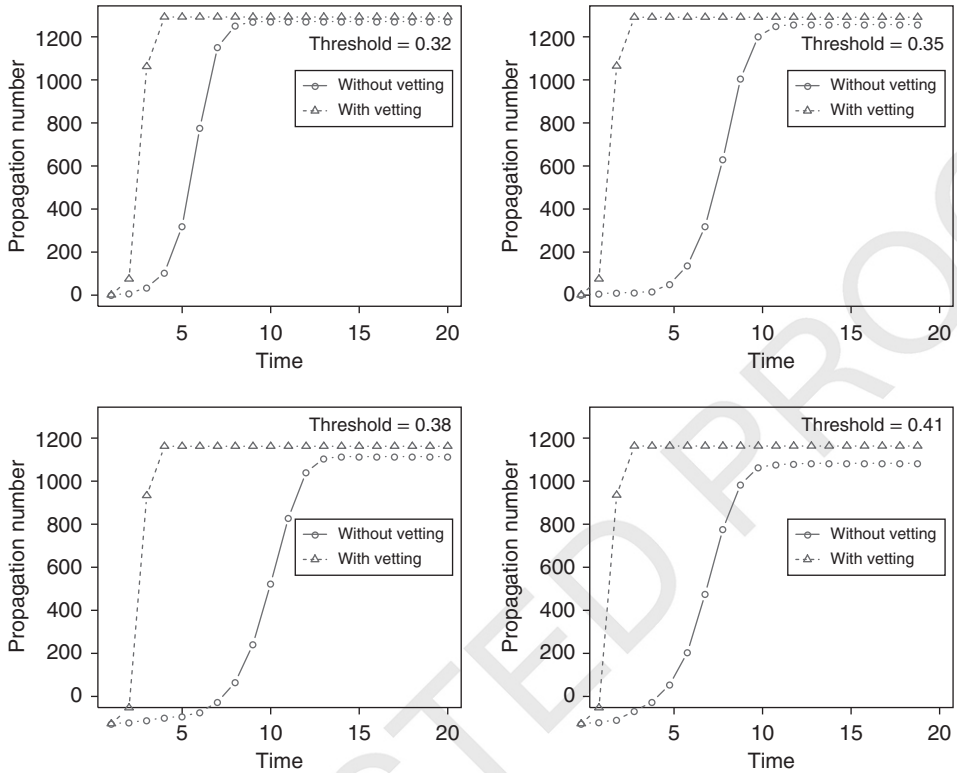

Figure 12.12 Propagation of correct information with increasing threshold of the users. We increase the threshold from 0.32 to 0.41. The threshold is the minimum weighted (calculated with using trust among the users) number of neighbors who had shared the content with the user required to influence a user. This experiment shows that content vetting increases the number of users who shared the correct information.

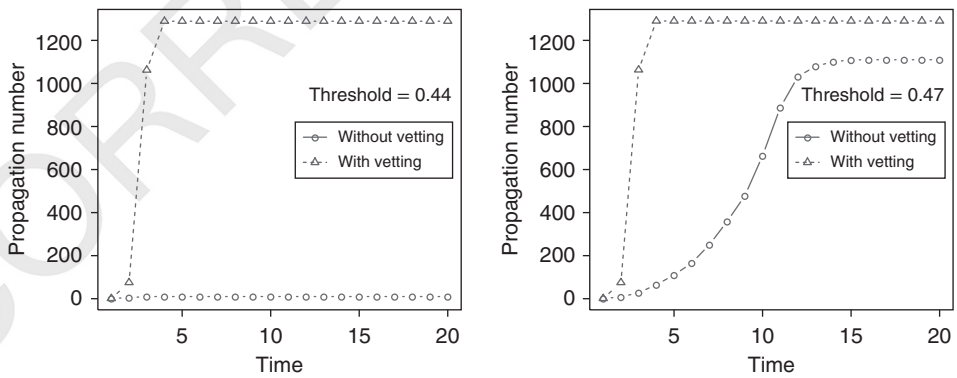

Figure 12.13 Propagation of correct information with increasing threshold of the users. We increase the threshold from 0.44 to 0.53. The threshold is the minimum weighted (calculated using trust among the users) number of neighbors who had shared the content with the user to influence a user. This experiment shows that content vetting increases the number of users who shared incorrect information. It also shows that as the threshold is increased, the number of users who share the correct content without vetting is reduced. This is because as we increase the threshold, it becomes more difficult to influence a user.

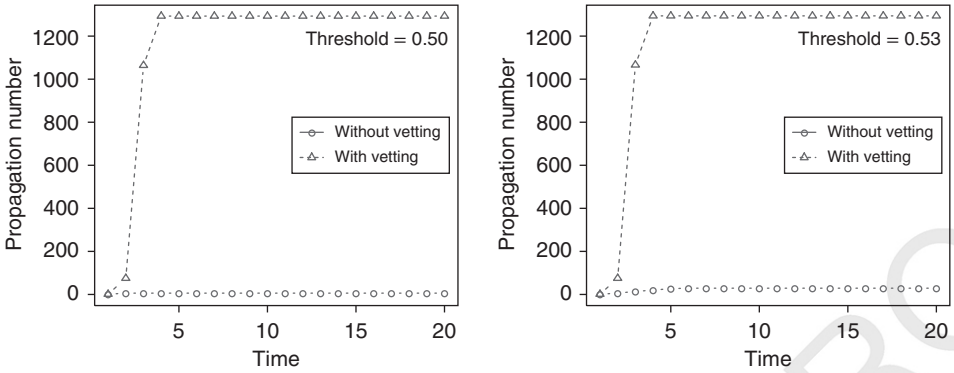

Figure 12.13 (*Continued*)

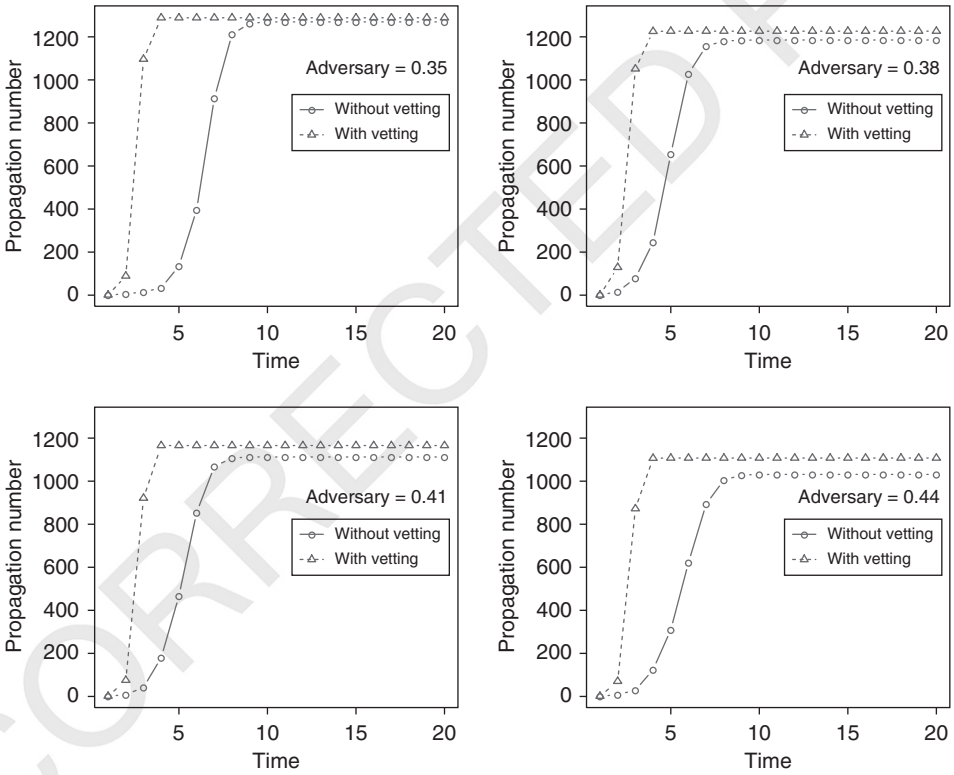

Figure 12.14 Propagation of correct information with increasing number of users controlled by the adversarial user. We increase the number of such users from 35% to 44%. It shows that with content vetting, the number of users who shared the correct information remains higher. Also, it shows that increasing the number of users controlled by the adversarial user decreases the number of users who shared the correct content.

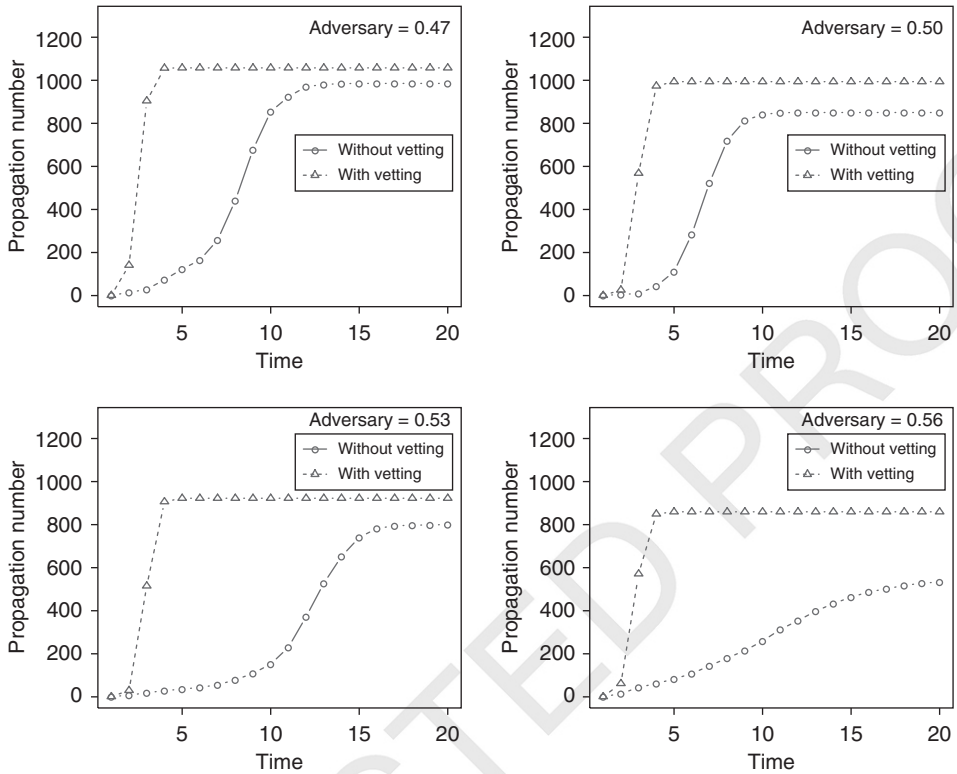

Figure 12.15 Propagation of correct information with increasing number of users controlled by the adversarial user. We increase the number of such users from 47% to 56%. It shows that with content vetting the number of users who shared the correct information remains higher. Also, it shows that increasing the number of users controlled by the adversarial user decreases the number of users who shared the correct content.

12.8 Conclusion

In this chapter, we developed a decentralized content vetting in social network procedure using public proof of work blockchains. We use offline channels to execute the vetting procedure. We showed that the vetting procedure can significantly reduce the propagation of social media content that the users have voted as rumor or misinformation. In the future, we will extend this vetting procedure with functionalities that can protect the privacy of the users who have vetted content.

Acknowledgment

This publication has emanated from research supported in part by a research grant from the Department of Enterprise, Trade and Employment's Disruptive Technologies Innovation Fund (DTIF), managed by Enterprise Ireland on behalf of the Government of Ireland under

Grant Number DT20180040C, and by a research grant from Science Foundation Ireland (SFI) and the Department of Agriculture, Food and the Marine on behalf of the Government of Ireland under Grant Number SFI 16/RC/3835 (VistaMilk) and also by a research grant from SFI under Grant Number SFI 12/RC/2289_P2 (Insight), with the latter two grants co-funded by the European Regional Development Fund.

References

1 University of Baltimore The economic cost of bad actors on the internet: fake news in 2019. https://www.cheq.ai/fakenews. Accessed: 2021-02-17.

2 Y. Boshmaf, I. Muslukhov, K. Beznosov, and M. Ripeanu. Design and analysis of a social botnet. *Computer Networks*, 57 (2): 556–578, Feb. 2013. ISSN 1389-1286. https://doi.org/10.1016/j.comnet.2012.06.006.

3 E. Ferrara, O. Varol, C. Davis,F. Menczer, and A. Flammini. The rise of social bots. *Communications of the ACM*, 59 (7): 96–104, June 2016. ISSN 0001-0782. https://doi.org/10.1145/2818717.

4 Ironman at Political Calculations Follow The cost of fake news for the S&P 500. https://seekingalpha.com/article/4129355-cost-of-fake-news-for-s-and-p-500. Accessed: 2021-02-17.

5 A. R. Pathak, A. Mahajan, K. Singh, A. Patil, and A. Nair. Analysis of techniques for rumor detection in social media. *Procedia Computer Science*, 167: 2286–2296, 2020. ISSN 1877-0509. https://doi.org/https://doi.org/10.1016/j.procs.2020.03.281. International Conference on Computational Intelligence and Data Science.

6 A. Zubiaga, A. Aker, K. Bontcheva, M. Liakata, and R. Procter. Detection and resolution of Rumours in social media: a survey. *ACM Computing Surveys*, 51 (2), Feb. 2018. ISSN 0360-0300. https://doi.org/10.1145/3161603.

7 J. Jiang, S. Wen, S. Yu, Y. Xiang, and W. Zhou. Rumor source identification in social networks with time-varying topology. *IEEE Transactions on Dependable and Secure Computing*, 15 (1): 166–179, 2018. https://doi.org/10.1109/TDSC.2016.2522436.

8 M. Farajtabar, M. G. Rodriguez, M. Zamani, N. Du, H. Zha, and L. Song. Back to the Past: Source Identification in Diffusion Networks from Partially Observed Cascades. In G. Lebanon and S. V. N. Vishwanathan, editors, *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics*, volume 38 of *Proceedings of Machine Learning Research*, pages 232–240, San Diego, California, USA, 09–12 May 2015. PMLR. http://proceedings.mlr.press/v38/farajtabar15.html.

9 E. A. Vogels, A. Perrin, and M. Anderson. Most Americans think social media sites censor political viewpoints. https://www.pewresearch.org/internet/2020/08/19/most-americans-think-social-media-sites-censor-political-viewpoints/. Accessed: 2021-02-17.

10 Social media: is it really biased against us republicans? https://www.bbc.com/news/technology-54698186. Accessed: 2021-02-17.

11 A. Kiayias, B. Livshits, A. M. Mosteiro, and O. S. T. Litos. A puff of steem: security analysis of decentralized content curation. *CoRR*, abs/1810.01719, 2018. http://arxiv.org/abs/1810.01719.

12 C. Li and B. Palanisamy. Incentivized blockchain-based social media platforms: a case study of steemit. In *Proceedings of the 10th ACM Conference on Web Science*, WebSci '19, pages 145–154, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450362023. https://doi.org/10.1145/3292522.3326041.

13 DPOS consensus algorithm - the missing white paper. https://steemit.com/dpos/@dantheman/dpos-consensus-algorithm-this-missing-white-paper. Accessed: 2021-02-17.

14 Mith. https://mith.io/en-US/. Accessed: 2021-02-17.

15 N. Aurélien B. Ankit, and G. Robert. Sapien. Decentralized social news platform. 2018. https://www.sapien.network/static/pdf/SPNv1_4.pdf.

16 S.P. Ltd. The socialx ecosystem takes the social media experience to the next level. 2018. https://socialx.network/wp-content/uploads/2018/09/Whitepaper-SocialX-v1.1.pdf.

17 Foresting. Rewarding lifestyle social media. 2019. https://cdn.foresting.io/pdf/whitepaper/FORESTING_Whitepaper_Eng_Ver.1.0.pdf?ver0.2.

18 Minds. The crypto social network. 2019. https://cdn-assets.minds.com/front/dist/en/assets/documents/Whitepaper-v0.5.pdf.

19 B. Guidi. When blockchain meets online social networks. *Pervasive and Mobile Computing*, 62: 101131, 2020. ISSN 1574-1192. https://doi.org/https://doi.org/10.1016/j.pmcj.2020.101131.

20 L. Jiang and X. Zhang. BCOSN: a blockchain-based decentralized online social network. *IEEE Transactions on Computational Social Systems*, 6 (6): 1454–1466, 2019.

21 Q. Xu, Z. Song, R. S. Mong Goh, and Y. Li. Building an ethereum and IPFS-based decentralized social network system. In *2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS)*, pages 1–6, 2018.

22 M. Ur Rahman, B. Guidi, and F. Baiardi. Blockchain-based access control management for decentralized online social networks. *Journal of Parallel and Distributed Computing*, 144: 41–54, 2020. ISSN 0743-7315. https://doi.org/https://doi.org/10.1016/j.jpdc.2020.05.011.

23 L. Bahri, B. Carminati, and E. Ferrari. Decentralized privacy preserving services for online social networks. *Online Social Networks and Media*, 6: 18–25, 2018. ISSN 2468-6964. https://doi.org/https://doi.org/10.1016/j.osnem.2018.02.001.

24 D. Fu and L. Fang. Blockchain-based trusted computing in social network. In *2016 2nd IEEE International Conference on Computer and Communications (ICCC)*, pages 19–22, 2016.

25 F. Yang, Y. Pu, C. Hu, and Y. Zhou. A blockchain-based privacy-preserving mechanism for attribute matching in social networks. In D. Yu, F. Dressler, and J. Yu, editors, *Wireless Algorithms, Systems, and Applications*, pages 627–639, Cham, 2020. Springer International Publishing. ISBN 978-3-030-59016-1.

26 B. Guidi, V. Clemente, T. García, and L. Ricci. A rewarding model for the next generation social media. In *Proceedings of the 6th EAI International Conference on Smart Objects and Technologies for Social Good*, GoodTechs '20, pages 169–174, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450375597. https://doi.org/10.1145/3411170.3411247.

27 P. Freni, E. Ferro, and G. Ceci. Fixing social media with the blockchain. In *Proceedings of the 6th EAI International Conference on Smart Objects and Technologies for*

Social Good, GoodTechs '20, page 175–180, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450375597. https://doi.org/10.1145/3411170.3411246.

28 F. Yang, Y. Wang, C. Fu, C. Hu, and A. Alrawais. An efficient blockchain-based bidirectional friends matching scheme in social networks. *IEEE Access*, 8: 150902–150913, 2020.

29 S. Nakamoto. Bitcoin: a peer-to-peer electronic cash system. www.bitcoin.org, 2008.

30 B. P. Hayes, S. Thakur, and J. G. Breslin. Co-simulation of electricity distribution networks and peer to peer energy trading platforms. *International Journal of Electrical Power & Energy Systems*, 115: 105419, 2020. ISSN 0142-0615. https://doi.org/https://doi.org/10.1016/j.ijepes.2019.105419.

31 I. A. Ridhawi, M. Aloqaily, A. Boukerche, and Y. Jaraweh. A blockchain-based decentralized composition solution for IoT services. In *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, pages 1–6, 2020.

32 J. Poon and T. Dryja. The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments. https://lightning.network/lightning-network-paper.pdf.

33 G. Malavolta, P. Moreno-Sanchez, A. Kate, and M. Maffei. SilentWhispers: enforcing security and privacy in decentralized credit networks. *IACR Cryptology ePrint Archive*, 1054, 2016.

34 P. Moreno-Sanchez, A. Kate, M. Maffei, and K. Pecina. Privacy preserving payments in credit networks: enabling trust with privacy in online marketplaces. In *NDSS*, 2015.

35 P. Prihodko, S. Zhigulin, M. Sahno, A. Ostrovskiy, and O. Osuntokun. Flare: an approach to routing in lightning network white paper. 2016.

36 A. Guille, H. Hacid, C. Favre, and D. A. Zighed. Information diffusion in online social networks: a survey. *SIGMOD Record*, 42 (2): 17–28, July 2013. ISSN 0163-5808. https://doi.org/10.1145/2503792.2503797.

37 D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '03, pages 137–146, New York, NY, USA, 2003. Association for Computing Machinery. ISBN 1581137370. https://doi.org/10.1145/956750.956769.

38 J. Leskovec and J. Mcauley. Learning to discover social circles in ego networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25, pages 539–547. Curran Associates, Inc., 2012. https://proceedings.neurips.cc/paper/2012/file/7a614fd06c325499f1680b9896beedeb-Paper.pdf.