electronics

MDPI

Article

HadithTrust: Trust Management Approach Inspired by Hadith Science for Peer-to-Peer Platforms

Amal Alqahtani [1], Heba Kurdi [1,2,*] and Majed Abdulghani [3]

[1] Computer Science Department, King Saud University, Riyadh 12371, Saudi Arabia; 437203170@student.ksu.edu.sa
[2] Mechancial Engineering Department, Massachusetts Institute of Technology (MIT), Cambridge, MA 02142, USA
[3] Computer Science Department, Faculty of Science, Engineering and Computing, Kingston University, Kingston upon Thames KT1 2EE, UK; Majed.Abdulghani@mis.com.sa
* Correspondence: hkurdi@ksu.edu.sa

Abstract: Peer-to-peer (P2P) platforms are gaining increasing popularity due to their scalability, robustness and self-organization. In P2P systems, peers interact directly with each other to share resources or exchange services without a central authority to manage the interaction. However, these features expose P2P platforms to malicious attacks that reduce the level of trust between peers and in extreme situations, may cause the entire system to shut down. Therefore, it is essential to employ a trust management system that establishes trust relationships among peers. Current P2P trust management systems use binary categorization to classify peers as trustworthy or not trustworthy. However, in the real world, trustworthiness is a vague concept; peers have different levels of trustworthiness that affect their overall trust value. Therefore, in this paper, we developed a novel trust management algorithm for P2P platforms based on Hadith science where Hadiths are systematically classified into multiple levels of trustworthiness, based on the quality of narrator and content. To benchmark our proposed system, HadithTrust, we used two state-of-art trust management systems, EigenTrust and InterTrust, with no-trust algorithm as a baseline scenario. Various experimental results demonstrated the superiority of HadithTrust considering eight performance measures.

Keywords: peer-to-peer; distributed systems; file sharing; reputation; trust; security; Hadith

Citation: Alqahtani, A.; Kurdi, H.; Abdulghani, M. HadithTrust: Trust Management Approach Inspired by Hadith Science for Peer-to-Peer Platforms. *Electronics* **2021**, *10*, 1442. https://doi.org/10.3390/electronics10121442

Academic Editors: Emanuele Bellini, Fiammetta Marulli and Stefano Marrone

Received: 1 May 2021
Accepted: 10 June 2021
Published: 16 June 2021

1. Introduction

Peer-to-peer (P2P) platforms have gained immense popularity due to their success in many large-scale distributed applications, such as file sharing networks [1], social networks [2], and content delivery systems [3]. In a P2P system, peers interact directly with one another to exchange files or perform a distributed task at a reasonably low operation and maintenance cost [4,5]. P2P systems offer many advantages over traditional client–server systems, such as scalability, robustness, and a wide range of offered resources [6].

However, as P2P systems lack centralized control, authentication and authorization are difficult to implement [7]. Moreover, peers are unknown entities who can join and leave the network at any time [8]. Consequently, P2P systems are vulnerable to many types of security threats and malicious attacks, such as buggy or inauthentic files. This results in a lack of trust between peers, which causes peers to refrain from sharing their resources and, in extreme cases, exit the network, resulting in the entire system's failure. Therefore, it is essential for a P2P system to employ a trust management system to ensure trustworthy file sharing between peers [9,10].

A trust management system checks whether the file to be exchanged is authentic or inauthentic and/or whether the provider peer is honest or dishonest. This way, risky transactions can be denied before taking place. Many trust management systems have been proposed to differentiate between authentic and inauthentic files and/or between

honest and dishonest peers. However, such binary classification of files and peers into two extremes is inefficient in many scenarios and unrealistic in many real-world contexts, as files and peers can be, on some level, in between the two extremes. Consequently, some files might be rejected even if they are not truly bad, and some bad files might be accepted even if they are not actually good.

Interestingly, Muslims face a similar trust problem when assessing the trustworthiness of *Hadiths* (sayings from the Prophet Mohammed) and their narrators. To tackle this problem, they employ a systematic, multilevel classification scheme that accurately helps a user decide whether to accept or reject these sayings. To the best of our knowledge, exploiting this scheme in trust management algorithms for P2P systems is an unexplored area.

Therefore, this paper proposes a new trust management algorithm, HadithTrust, inspired by the Hadith classification science, which follows a systematic approach to classify Hadiths (files) and narrators (peers) in gradual quality levels. It assigns a specific trust level for a certain Hadith (file) based on the provider peer's (narrator) honesty and *Matn* (file content) authenticity. HadithTrust classifies peers' honesty into four reputation quartiles (Q4, Q3, Q2, and Q1), in which Q4 is the highest reputation level and Q1 the lowest. These fourlevels have been determined on the basis of empirical observations, unlike the narrators' trust levels in Hadith science, which classifies narrators into twelve trust levels [11]. Accordingly, file authenticity is classified into one of four levels of authenticity (authentic, good, weak, or bad). The experimental results demonstrate HadithTrust's superiority in terms of success rate and run time.

This remainder of this paper is structured as follows. Section 2 discusses the related work. Section 3 introduces the system design. In Sections 4 and 5, we present the evaluation methodology and experimental results, respectively, and Section 6 concludes the paper and outlines directions for future work.

2. Literature Review

Trust management systems can be classified on the basis of the main entity considered in the reputation calculation into three categories: peer-based, file-based, and hybrid trust management systems [8].

In peer-based trust management systems, reputation is calculated on the basis of the peer's past behavior regardless of the files past ratings. In other words, the choice of a server is based on the peer's reputation, not the file's trustworthiness [8]. One of the most well-known peer-based trust management systems is EigenTrust [6]. In EigenTrust [6], each peer calculates his or her own normalized local trust values based on satisfactory and unsatisfactory numbers in past transactions with other peers. Then, each peer's global trust value is obtained by aggregating his or her local trust values and weighting them with corresponding global trust values. However, pretrusted peers that are used in trust calculations have several potential risks [12–14]. Therefore, many reputation systems were proposed to enhance EigenTrust [8,12–17]. The TNA-SL trust management system [18] introduced a new notation for expressing trust based on subjective logic. The trust values are represented as opinions, in which each opinion is expressed by four tuples: belief, disbelief, uncertainty, and base rate. These tuples are incorporated into mathematical equations to define trust values. Many trust management systems were proposed to simplify the heavy calculations of TNA-SL, such as TrustyFeer [19] and InterTrust [20]. Recently, some peer-based trust management solutions were proposed based on blockchain technology. In [21], a P2P trust management system was developed based on the aggregated feedbacks from customers to evaluate the reputation of the manufacturers who provided manufacturing services and deployed in a blockchain industrial IoT environment. In [22], a reputation-based mechanism is utilized to improve the security in blockchain. The reputation is calculated based on the past behavior of the peer, participation in voting for others and transactions. Then, the peer with the highest reputation is selected to create the new block and the block is evaluated by reputable peers in the blockchain. In Reference [23] a trust management model was presented for a semantic P2P grid environment. The model

integrates fuzzy theory and reputation technique, to gather locally-created feedback and produce a global node trust degree. A trust management mechanism, named fuzzy-based TM mechanism is proposed in [24] to prevent sybil attacks in the Internet of Medical Things. The proposed mechanism helps network nodes to calculate a node trust value using fuzzy logic and trust attributes, such as integrity, receptivity, and compatibility of a node.

In file-based trust management systems, reputation is calculated on the basis of the file regardless of the peer's past behavior. After each transaction, the file is rated positively or negatively, and the system keeps track of each file rating to measure the file's trustworthiness [19]. The amount of research on file-based reputation systems is scant, and to the best of our knowledge, only two studies have examined the file-based approach [8]. Walsh and Emin [25] developed a new approach for calculating file-based reputations for a structured P2P network. A file's reputation is calculated by aggregating positive and negative feedback on that file. However, the system has two repositories: a file repository that stores file information and a peer repository that stores the file reputation given by each peer. The system was benchmarked against a peer reputation system and had a lower percentage of inauthentic downloads. Credence [26] focused only on file authenticity through a simple voting algorithm that stores a file's positive and negative ratings. When a peer needs a file, the algorithm computes the available files' authenticity value based on aggregated feedback weighted by the experience correlation between the file requester and the file provider. The system classifies 80% of files correctly after a period of time. Moreover, the system's scalability is proved during the time it takes to estimate a file's authenticity while increasing the network's size.

In hybrid trust management systems, the reputation involves both peer and file trustworthiness. the number of existing hybrid trust management systems can exploit the advantages of both peer-based and file-based reputation systems. However, the number of existing hybrid systems is low. AuthenticPeer [8] was the first hybrid trust management system. The algorithm combines both a file-based approach and a peer-based approach to calculate trust values in the system. The number of transactions is considered to select the algorithm to run. If the number of transactions is lower than a certain threshold, the EigenTrust algorithm [6] is used; otherwise, the file-based algorithm [25] is used. AuthenticPeer++ [27] enhances the performance of AuthenticPeer by using the peer's global trust value to weigh the opinion on the shared files.

On the basis of the reviewed extant literature, peer-based trust management systems ignore files'/services' credibility, which is more reliable and stable than peers' reputations. Additionally, file-based trust management systems require massive storage and heavy processing due to the huge number of files in the network. Therefore, hybrid trust management systems are emerging to ensure accurate trust calculations, while reducing execution overhead. However, the number of studies that have examined hybrid trust management systems is scant, and none of the previous trust management systems have used the Hadith science approach to overcome the trust issue with P2P systems.

3. System Design

3.1. System Architecture

The Hadith-inspired approach is a hybrid file/peer trust management system. As shown in Figure 1, HadithTrust consists of two subsystems: reputation manager and files-providers manager.

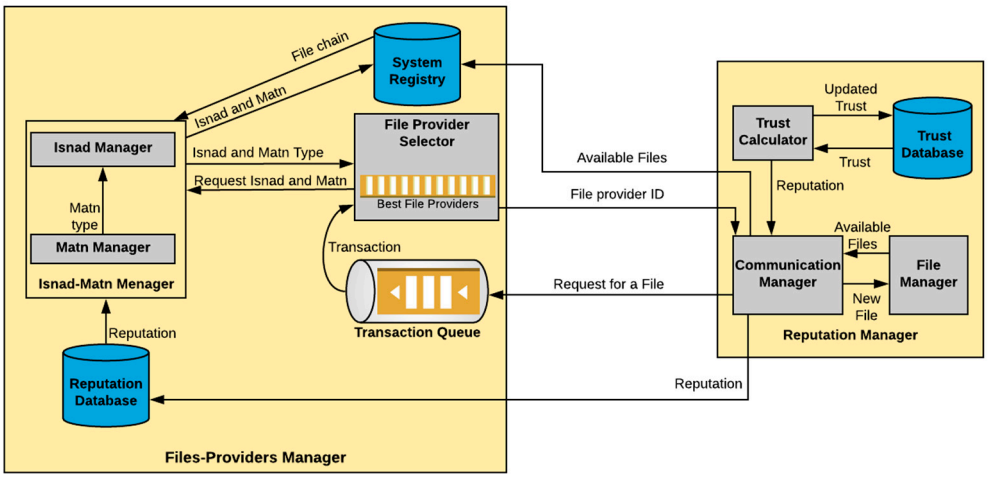


Figure 1. System architecture.

3.1.1. Reputation Manager

Each peer has his or her own reputation manager to calculate trust and reputation values locally. The reputation manager comprises four components:

- Communication manager: Sends file requests to the transaction queue, receives the reputation value from the trust calculator to update the reputation database after each transaction, and receives the selected file provider's ID from the file provider selector. The communication manager also updates the system registry with newly available file copies for each peer.
- Trust calculator: Is responsible for calculating its own local trust values and sending them to the trust database. In addition, it is responsible for calculating peers' reputation values after each transaction and sending them to the communication manager.
- Trust database: Stores all calculated provider peers' trust values after each transaction.
- File manager: Keeps track of all files owned by peers. It is updated after each transaction.

3.1.2. Files-Providers Manager

This is a centralized component responsible for selecting the best file provider for the requester peer on the basis of the quality of both the content (Matn) and providers (Isnad). The centralization of files-providers manager helps to ensure the system consistency and efficiency without degrading the performance, since all reputation calculations are distributed and calculated at each peer. The files-providers manager comprises five components:

- System registry: Stores all file copies in the system with corresponding file IDs, copy IDs, owner IDs, file chains, Isnads, and Matns. In addition, it is updated after each transaction.
- Transaction queue: Holds requests for files for further commitment. The transaction requests that come from malicious peers are denied directly before commitment, which is better than spending time to provide bad files for malicious peers as in [28]. Therefore, the time complexity of the system is maintained at low level.
- Reputation database: Stores all peers' reputations and sends them to the Isnad–Matn manager.
- File provider selector: Receives all available files' Matns and Isnads for the current transaction request. It stores the best-file-providers' IDs based on each file's Isnad and Matn in the best-file-providers list, and then it randomly selects a file provider from the list and sends the file provider's ID to the file requester
- Isnad–Matn manager: Receives the chains of the requested file in the system registry and analyzes them to dynamically determine the types of Isnad and Matn. The Isnad–Matn manager comprises two main components:

- Matn manager: Keeps track of all file ratings while the file is sent from one peer to another, and then analyzes each chain's ratings to determine the Matn type.

Isnad manager: Evaluates file content validity on the basis of the highest and lowest reputation values of all providers for the requested file. The range between these two reputation values is divided into four equal quartiles (Q1, Q2, Q3, and Q4), in which Q1 is the lowest reputation quartile and Q4 is highest. Experimental observations suggest that the best number for reputation quartiles is four. These quartiles are then used to classify all Isnads of that file. Section 3.3. describes the four quartiles.

3.2. Hadith Inspiration

In Hadith science, Hadiths (sayings of Prophet Mohammed) can be classified based on Matn (the content of Hadith) and Isnad (the chain of narrators). To determine a Hadith's authenticity, the Matn is analyzed, and its validity evaluated. In addition, the chain of narrators is analyzed, and their trustworthiness evaluated. Then, the Hadith is classified into one of the following categories: Sahih, Hasan, Daif, or Munkar. Some Hadiths have multiple copies that come from different narrators with different trustworthiness levels, while a Hadith's copies may have different Matn validity.

The similarity between Hadith science and trust management is interesting. Therefore, we implemented the HadithTrust trust management system based on some aspects of the Hadith classification model. In HadithTrust, a file can be classified on the basis of the validity of file ratings (akin to Matn validity) and trustworthiness of the chain of file providers who sent the file (akin to trustworthiness of narrators in Isnad). Then, the file is classified into one of four categories: authentic, good, weak, or bad. In this paper, a file corresponds to a Hadith, the file's content corresponds to Matn, the file content's quality ratings correspond to the validity of Matn, and the chain of providers (narrators) that sent the file corresponds to Isnad.

3.3. Matn and Isnad in Trust Management

When a file is sent from one peer to another, a new file copy is added to the system registry containing an owner ID, a file ID, a copy ID, and a file chain. Figure 2 illustrates a file chain for file ID 1 (F_1) from the initial owner to the last peer who receives the file. Initially, the file copy (C_{10}) was owned by peer P_a, who sent the file to peer P_b, who gave the file a positive rating. Then, P_b sent the file copy C_{11} to peer P_c, who gave the file a negative rating. Then, the file was sent to peer P_d and then to peer P_e. The final file chain is shown in Figure 2.

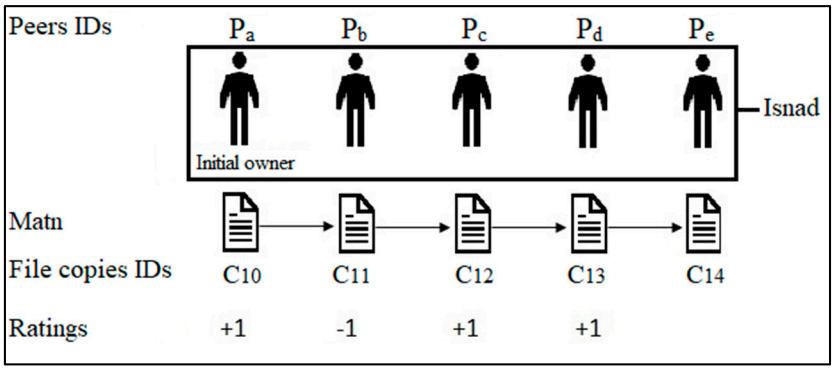


Figure 2. Example of a file chain.

The Matn can be valid, invalid, or unknown. The Matn is valid when all honest narrators (pretrusted and good) give positive ratings to the file. The Matn is invalid if one or more honest narrators give negative ratings to the file. Otherwise, if no honest narrator rates the file, the Matn is unknown. Figure 3 illustrates a valid and an invalid Matn.

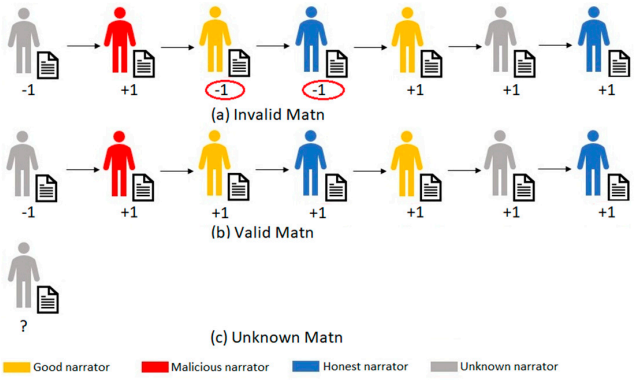


Figure 3. Matn types.

The Isnad can be one of four types. An authentic Isnad is when the Matn is valid and the Isnad's reputation, the average reputation for all peers involved in the file chain, is higher than Q1, as illustrated in the example in Figure 4, based on the assumed current peers' reputation values in the table in Figure 4. A good Isnad has two possible cases: when the Matn is valid and the Isnad's reputation is less than Q2 and when the Matn is unknown and the Isnad's reputation is higher than Q3; Figure 5 illustrates the two cases. A weak Isnad is when the Matn is unknown and the Isnad's reputation is higher than Q1 and less than Q4, as shown in Figure 6. A bad Isnad also has two possible cases: when the Matn is invalid without considering the value of the Isnad's reputation since the file is a bad file and when the Matn is unknown and the Isnad's reputation is less than Q2; Figure 7 illustrates the two cases.

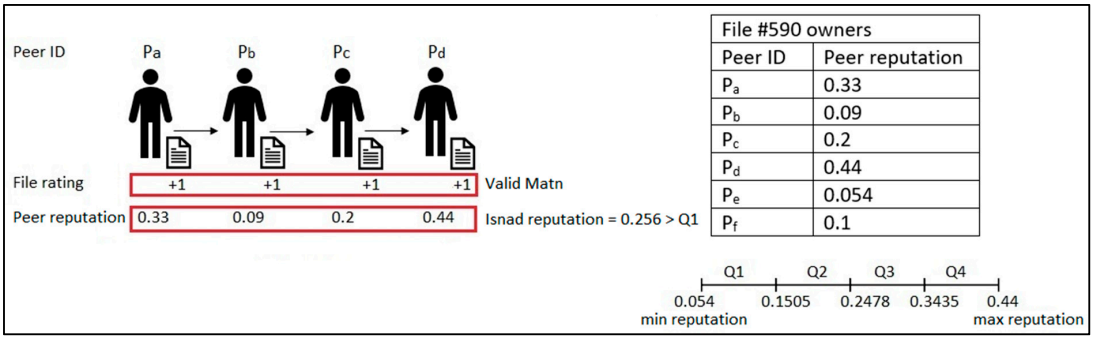


Figure 4. Authentic Isnad.

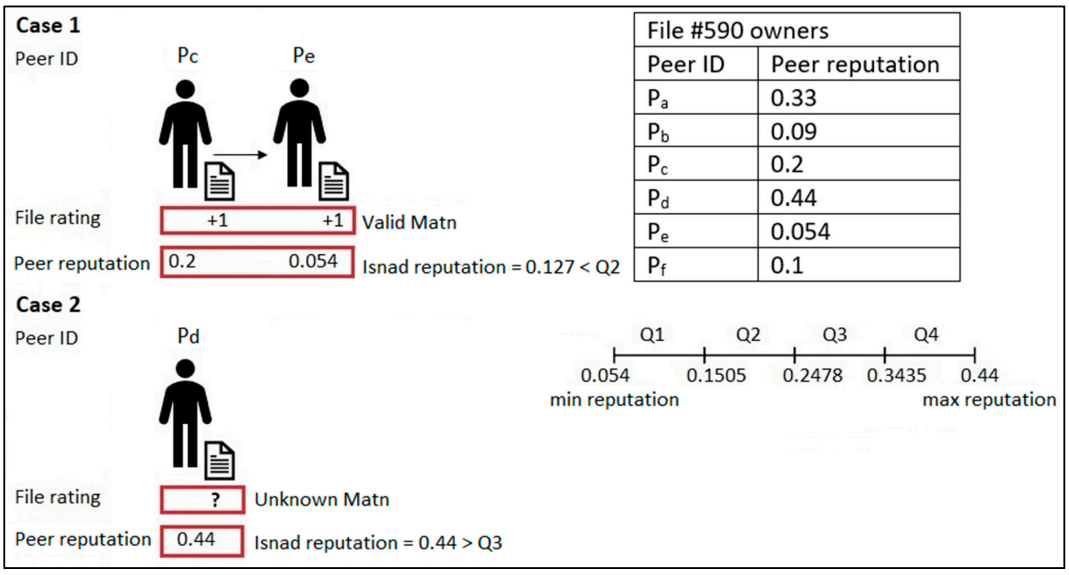


Figure 5. Good Isnad.

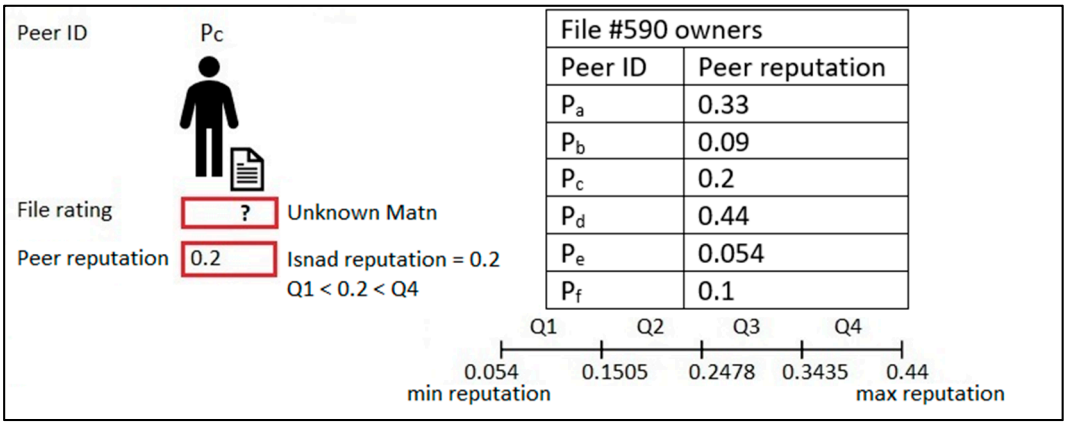


Figure 6. Weak Isnad.

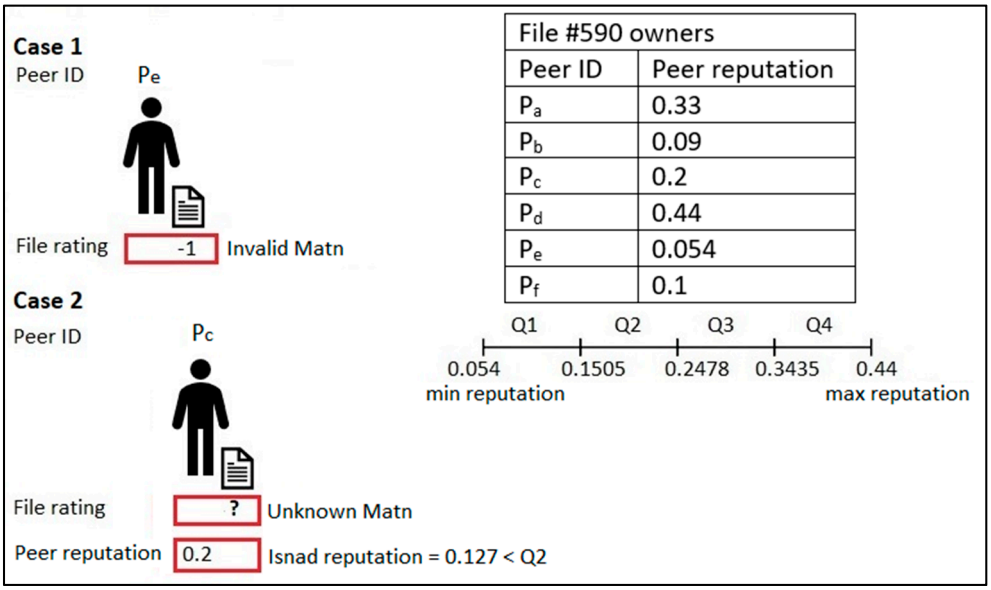


Figure 7. Bad Isnad.

3.4. Reputation Calculations

HadithTrust considers two aspects to calculate the reputation of files/peers: Matn, the trustworthiness of files in the network on the basis of file ratings given by good narrators, and Isnad, the trustworthiness of all narrators in the file chain with respect to the Matn type.

However, as illustrated in Figure 8, after each transaction, the percentage of the participation of each peer in authentic Isnads APcti is calculated as follows:

$$\text{APct}_i = \frac{\#\text{Authentic Isnads involving peer i}}{\#\text{ All Isnads involving peer i}} \quad (1)$$

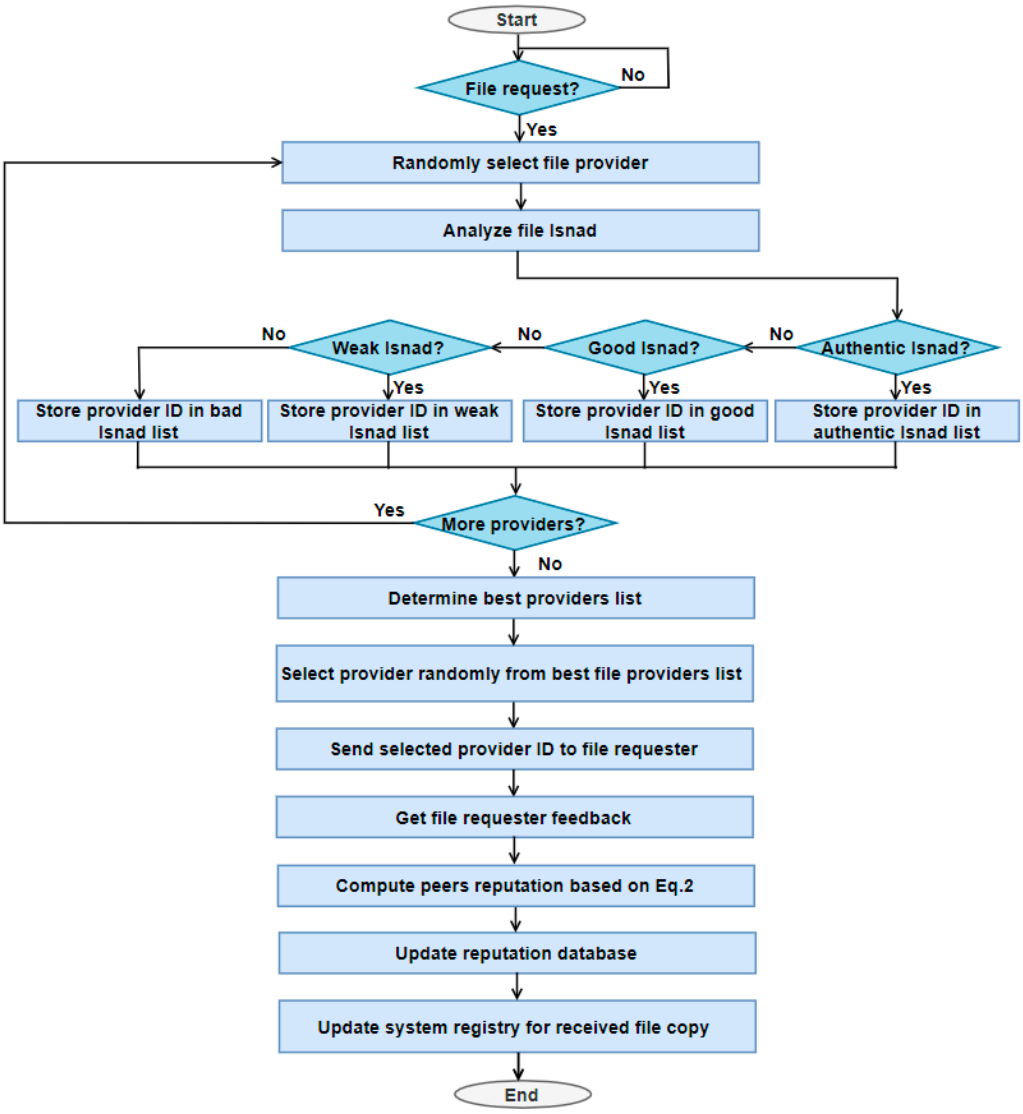


Figure 8. HadithTrust algorithm flowchart.

Then, the trust manager calculates reputation values at each peer, as Equation (2) demonstrates, following the EigenTrust's formula [6] with a little improvement. We replaced the extra weight given for pretrusted peers with APcti, as it is more accurate to describe the quality of files owned by that peer:

$$t_i^{(k+1)} = (1-a)\left(c_{1i}t_1^{(k)} + \cdots + c_{ni}t_n^{(k)}\right) + a\mathrm{APct}_i \quad (2)$$

where $t_i^{(k+1)}$ is the reputation of peer i, a is a constant factor less than 1, APct_i is the percentage of peer i's participations in authentic Isnads, k is a peer who is trusted by peer i indirectly (when peer i trusts peer j and peer j trusts peer k, then peer i trusts peer k indirectly), and c_{ni} is the normalized trust value of peer n in peer i, calculated as follows [6]:

$$C_{ni} = \frac{max(S_{ni}, 0)}{\sum_n max(S_{ni}, 0)} \quad (3)$$

where c_{ni} is the summation of positive and negative feedbacks that peer n gave for peer i in the past transactions.

Then, as illustrated in Figure 9, the number of positive ratings given by honest peers is calculated and compared with the number of honest narrators. The Matn has three possible types:

- If the number of positive ratings = 0, the Matn is classified as unknown.
- If the number of positive ratings < the number of honest narrators, the Matn is classified as invalid.
- Else, the Matn is classified as valid.

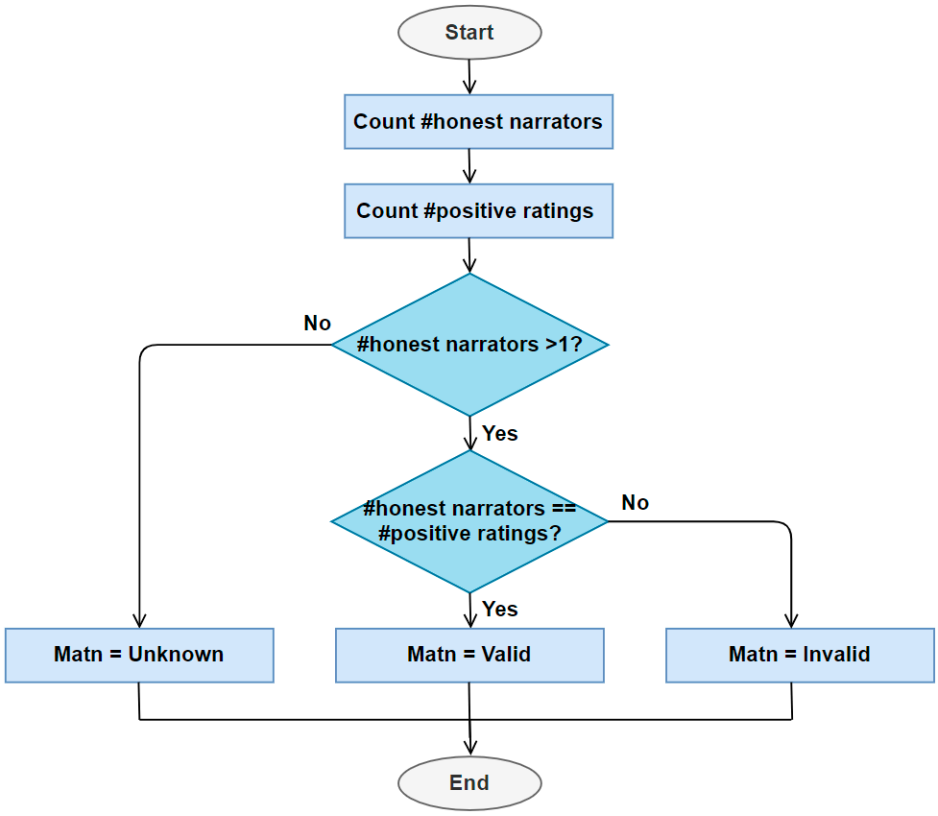


Figure 9. Matn analysis algorithm flowchart.

Additionally, the Isnad is classified on the basis of the narrator's reputation and Matn type. When a request for a transaction is issued, the Isnad–Matn manager requests the current reputation values of all requested file providers. The highest reputation (HR) and the lowest reputation (LR) are stored for further Isnad analysis. The interval between these two values is divided into four equal quartiles ($Q1$, $Q2$, $Q3$ and $Q4$) as follows:

$$Q1 = \left\{ x | LR \leq x \leq LR + \frac{(HR - LR)}{4} \right\} \tag{4}$$

$$Q2 = \left\{ x | LR + \frac{(HR - LR)}{4} < x \leq LR + 2\frac{(HR - LR)}{4} \right\} \tag{5}$$

$$Q3 = \left\{ x | LR + 2\frac{(HR - LR)}{4} < x \leq LR + 3\frac{(HR - LR)}{4} \right\} \tag{6}$$

$$Q4 = \left\{ x | LR + 3\frac{(HR - LR)}{4} < x \leq HR \right\} \tag{7}$$

Then, as illustrated in Figure 10, the Isnad's reputation is calculated (the average reputation of all narrators involved in a file chain) as follows:

$$Isnad\ reputation = \frac{\sum Reputation\ of\ narrators\ in\ the\ file\ chain}{Length\ of\ the\ file\ chain} \tag{8}$$

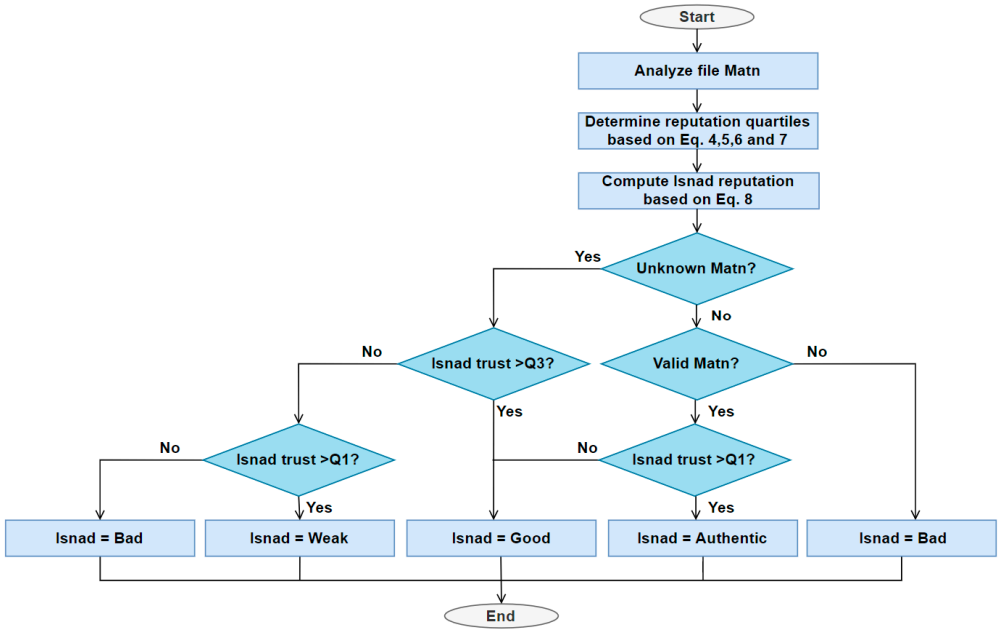


Figure 10. Isnad analysis algorithm flowchart.

On the basis of the Isnad's reputation and Matn type, the Isnad is classified as follows:

- If the Matn is unknown:
  1. If the Isnad's reputation > $Q3$, the Isnad is classified as good.
  2. If the Isnad's reputation > $Q1$, the Isnad is classified as weak.
  3. Else, the Isnad is classified as bad.
- If the Matn is valid:
  1. If the Isnad's reputation > $Q1$, the Isnad is classified as authentic.
  2. Else, the Isnad is classified as good.
- Else, the Isnad is classified as bad.

After classifying the Isnads for all file copies of the requested file, the file provider selector stores the IDs of the providers who own the file copies with the best Isnads in a best-file-providers list, a file provider is randomly selected from the list, and the file provider's ID is sent to the file requester, as illustrated in Figure 11.

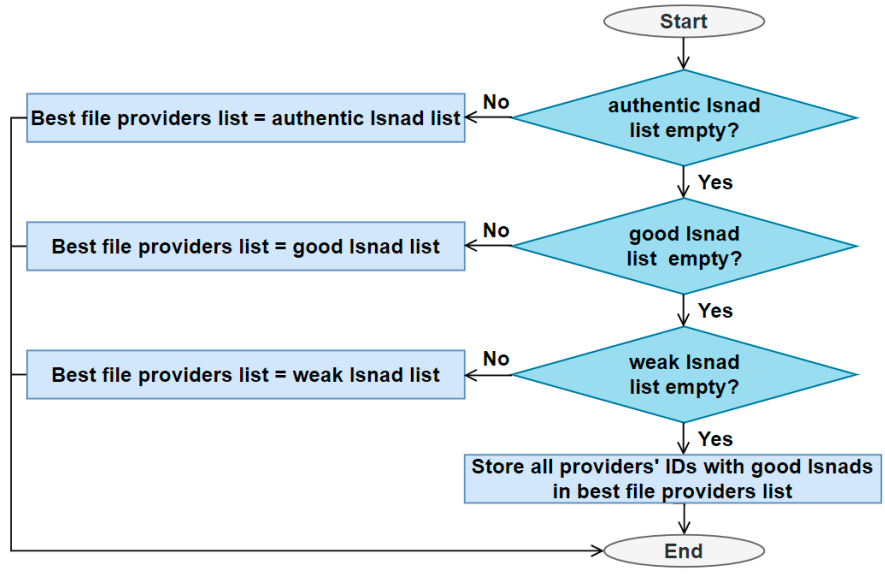


Figure 11. Best provider list selection flowchart.

4. Evaluation Methodology

We used the Quantitative Trust Management (QTM) simulator [28] to test HadithTrust. QTM is a simulator that has been developed specifically to evaluate trust management systems in P2P networks. It supports various malicious models and provides ready implementation for two well-established algorithms, EigenTrust and TNA-SL. In addition to the baseline scenario where no-trust algorithm (None) is used. The selected application model to test the proposed system is a P2P file sharing system since it is the most representative application for P2P systems. This model is implemented using an intelligent query model, where all peers have equal library size and chance of being a file requester based on the Zipf distribution [28].

The proposed system was evaluated using a well-designed evaluation where several variables were considered at different values, including:

- Percentage of malicious peers, which varied from 15% to 75% in steps of 15%
- Malicious strategies: Two different strategies are considered:
  - Collective strategy: when malicious peers form groups, then raise their reputations by giving each other positive feedback and giving other peers negative feedback
  - Naive strategy: when malicious peers behave independently of other malicious peers
- Honest peer models: Two models are considered:
  - Good peers, who always provide honest feedback about other peers and usually provide authentic files to others, and pretrusted peers, who have higher initial reputation values and are important in helping guide new peers who join the network
  - Unknown peer model considered in all experiments that represents newcomers with unknown behaviors
- Malicious peer models: Four models are considered:
  - Malicious feedback peers, who provide authentic files and dishonest feedback
  - Malicious peers, who provide inauthentic files and honest feedback
  - Purely malicious peers, who provide inauthentic files and dishonest feedback
  - Camouflaged malicious peers, who provide authentic files and honest feedback 50% of the time but behave as purely malicious peers the rest of the time

In all experiments, the percentages of pretrusted peers and unknown peers were fixed at 10%. Additionally, the number of files remained constant, at 1000, the number of peers is constant at 128, and the number of transactions remained constant, at 2500.

To evaluate the proposed system's efficacy, the following performance measures were used:

1. Success rate, which is calculated as the number of authentic file downloads by good peers divided by the number of all downloads by good peers:

$$\text{Success rate} = \frac{\#\text{Authentic file downloads by good peers}}{\#\text{All file downloads by good peers}} \times 100 \quad (9)$$

2. Percentage of authentic downloads, which is calculated as the number of authentic Isnads downloaded by good peers divided by the number of all good peer downloads:

$$\text{Authentic downloads} = \frac{\#\text{Authentic Isnads downloaded by good peers}}{\#\text{ All good peer downloads}} \times 100 \quad (10)$$

3. Percentage of good downloads, which is calculated as the number of good Isnads downloaded by good peers divided by the number of all good peer downloads:

$$\text{Good downloads} = \frac{\#\text{Good Isnads downloaded by good peers}}{\#\text{ All good peer downloads}} \times 100 \quad (11)$$

4. Percentage of weak downloads, which is calculated as the number of weak Isnads downloaded by good peers divided by the number of all good peer downloads:

$$\text{Weak downloads} = \frac{\#\text{Weak Isnads downloaded by good peers}}{\#\text{ All good peer downloads}} \times 100 \quad (12)$$

5. Percentage of bad downloads, which is calculated as the number of bad Isnads downloaded by good peers divided by the number of all good peers' downloads:

$$\text{Bad downloads} = \frac{\#\text{Bad Isnads downloaded by good peers}}{\#\text{All good peer downloads}} \times 100 \quad (13)$$

6. Percentage of downloads from pretrusted peers, which is calculated as the number of good peer downloads from pretrusted peers divided by the number of all good peer downloads:

$$\text{Downloads from pretrusted peers} = \frac{\#\text{Good peers' downloads from pretrusted peers}}{\#\text{ All good peer downloads}} \times 100 \quad (14)$$

7. Running time, which is calculated as the time from the start of the first simulated transaction to the last one, in seconds
8. Transaction service time in milliseconds

Simulation setup is summurized in Table 1. The total number of experiments that is run in this paper is 4 (algorithms) × 5 (malicious%) × 4 (malicious models) × 2 (malicious strategies) × 10 (10 runs for each experiment) = 1600 experiment.

Table 1. Summary of simulation setup.

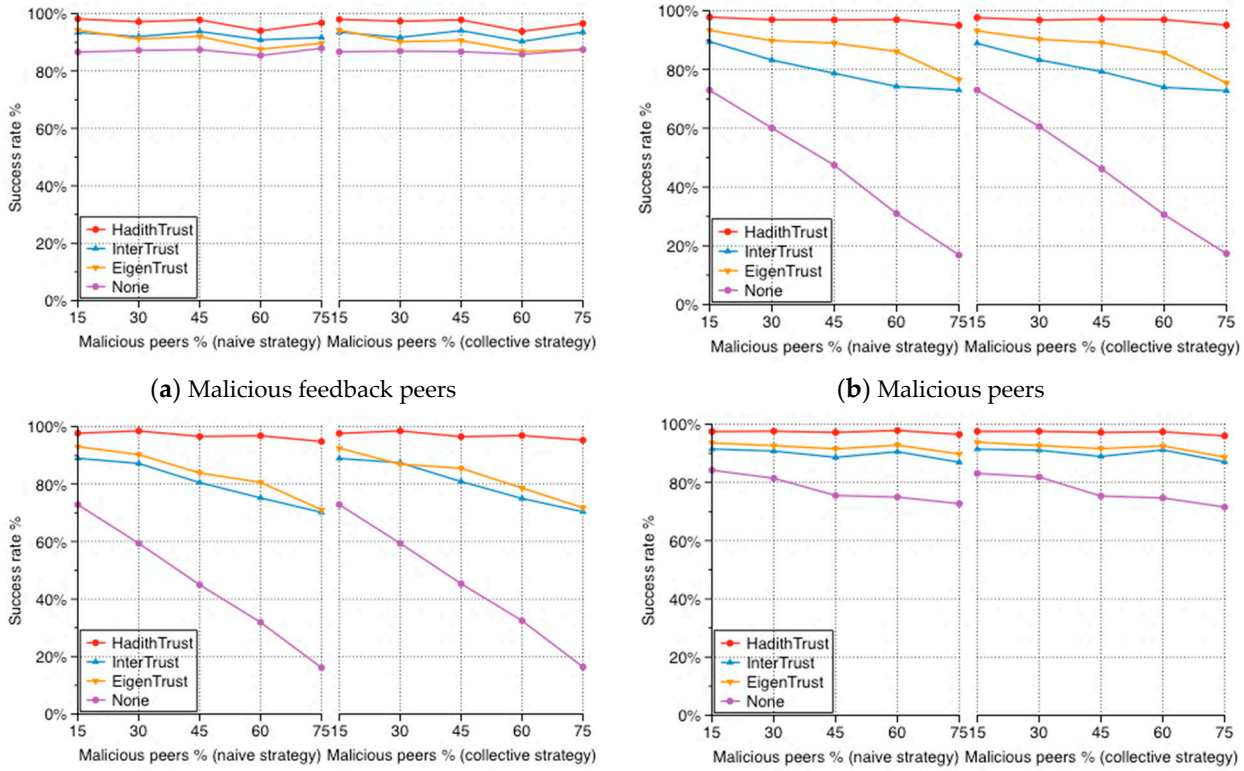
Exp	No. of Peers	No. of Transactions	Malicious Peers %	Unknown Peers %	Pre-Trusted Peers %	Malicious Models	Strategy
1	128	2500	15%	10%	10%	Feedback	Naïve/collective
2			30%				
3			45%				
4			60%				
5			75%				
6	128	2500	15%	10%	10%	Malicious	Naïve/collective
7			30%				
8			45%				
9			60%				
10			75%				
11	128	2500	15%	10%	10%	Pure	Naïve/collective
12			30%				
13			45%				
14			60%				
15			75%				
16	128	2500	15%	10%	10%	Feedback	Naïve/collective
17			30%				
18			45%				
19			60%				
20			75%				

5. Results and Discussion

All experiments were carried out on a personal computer (PC) with an Intel Core i7 CPU, with 1.8 GHz of speed, 6 GB of RAM, and 1000 GB of hard disk space. To benchmark our proposed system, we used EigenTrust [6] and InterTrust [20], with no-trust algorithm as a baseline scenario. This section presents the average results of 10 runs, considering the 7 metrics identified earlier.

5.1. Success Rate

The success rate of good peers, which is calculated as the number of good peers' authentic downloads divided by the number of all good peers' downloads, is illustrated in Figure 12. Four malicious models were considered, each of which works with collective and naive strategies. Generally, the success rate of each algorithm with different malicious strategies is almost similar to many previous experiments [8,19,20], which are implemented on the same simulator. Unlike the success rate, running time is obviously different between the two strategies.



(**a**) Malicious feedback peers

(**b**) Malicious peers

(**c**) Purely malicious peers

(**d**) Camouflaged malicious peers

Figure 12. Success rate.

Figure 12a shows the success rate of good peers with increasing percentage of malicious feedback peers. In all scenarios, it is clearly seen that HadithTrust's success rate is relatively higher than that of the benchmark algorithms. The success rate of EigenTrust, InterTrust, and no-trust (None) systems oscillates as the percentage of malicious feedback peers increases, whereas the success rate of HadithTrust is more stable, and as expected, the no-trust (None) system has the lowest success rate. In HadithTrust, file authenticity is considered and related to Isnad and Matn, which is not the case with EigenTrust and InterTrust; thus, if a file request is issued, the algorithm first searches for authentic Isnads for downloads, which is a 100% authentic file since good peers have experienced this file before. If no authentic Isnads are available, the algorithm searches for the next best Isnads. Consequently, the success rate with HadithTrust is higher and more stable than with other systems.

Figure 12b shows the success rate of good peers with increasing percentage of malicious peers. The success rate rapidly decreases as the percentage of malicious peers increases for EigenTrust, InterTrust, and no-trust (None) algorithms. In contrast, HadithTrust's success rate decreases slightly as the percentage of malicious peers increases. Nevertheless, HadithTrust achieves the highest success rate with all malicious provider percentages. In HadithTrust, the algorithm avoids invalid Matns for downloads, which prevents the propagation of inauthentic files. In this type of malicious provider, malicious

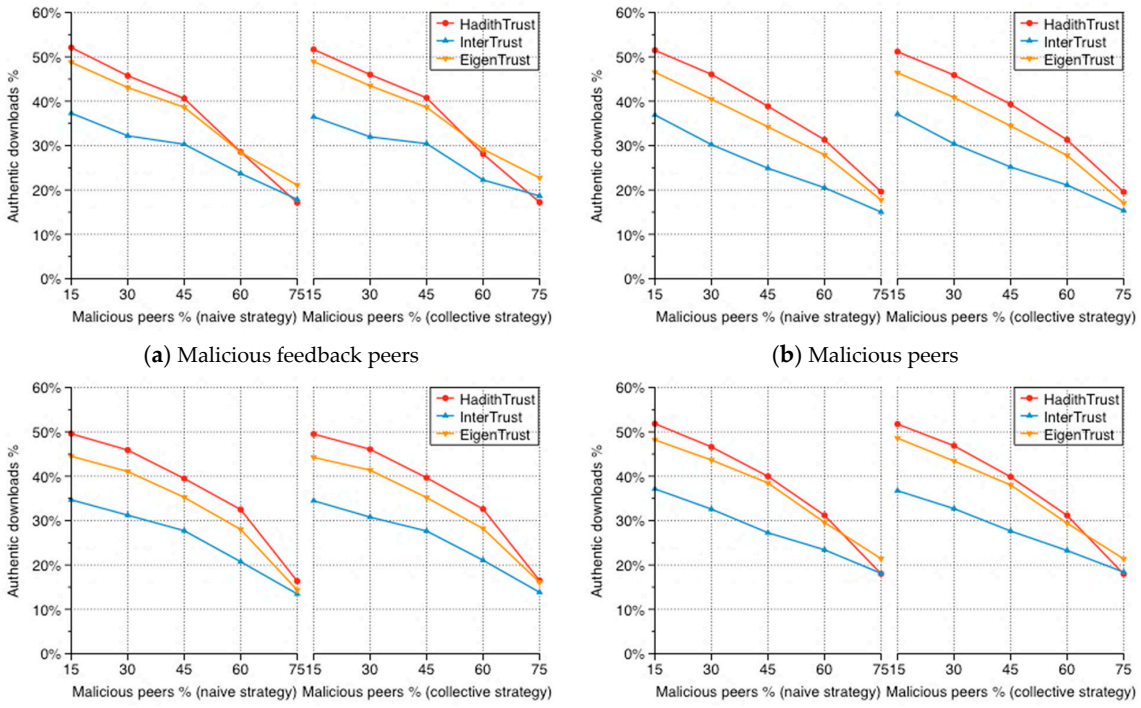
peers always provide inauthentic files, which affect the success rate in general, more than dishonest feedback. Therefore, the success rate of EigenTrust, InterTrust, and no-trust (None) systems is lower than that of HadithTrust.

Figure 12c shows the success rates of good peers with increasing percentage of purely malicious peers. Overall, Figure 12c highlights HadithTrust's superiority in the success rate compared with benchmark systems. The success rate of EigenTrust, InterTrust, and no-trust (None) dramatically decreases as the percentage of purely malicious peers increases. This contrasts with HadithTrust's success rate, which is not affected by the percentage of purely malicious peers. Moreover, the magnitude of success rate reduction for EigenTrust, InterTrust, and no-trust (None) is higher than what is shown in Figure 12a,b, since dishonest feedback has less effect on the success rate than inauthentic files provided by purely malicious peers.

Figure 12d presents the success rate of good peers with increasing percentage of camouflaged malicious peers. Generally, this malicious model is hard to detect, but its effect on the success rate is not as bad as purely malicious and malicious peers, because the peers behave as good peers half of the time. Therefore, it generally elicits less of an effect on the success rate. In Figure 12d, the success rates of EigenTrust and InterTrust gradually decrease as the percentage of camouflaged malicious peers' increases. The success rate of HadithTrust remains nearly constant. In HadithTrust, when the Matn is invalid, the inauthentic file is always rejected, even if it comes from a well-behaved peer, such as the starting behavior of camouflaged malicious peers. This is not the case with EigenTrust and InterTrust, in which only peer trustworthiness is considered on the basis of historical behavior, without considering the file's path and ratings.

5.2. Percentage of Authentic Downloads

Authentic downloads, calculated as the number of authentic Isnad downloads for good peers divided by the number of all Isnads downloaded by good peers, is presented in Figure 13 for HadithTrust and benchmark systems EigenTrust and InterTrust under different scenarios. However, the case of no-trust (None) is excluded because it is considered an important factor to calculate different Isnad types (authentic, good, weak, and bad).



(**a**) Malicious feedback peers

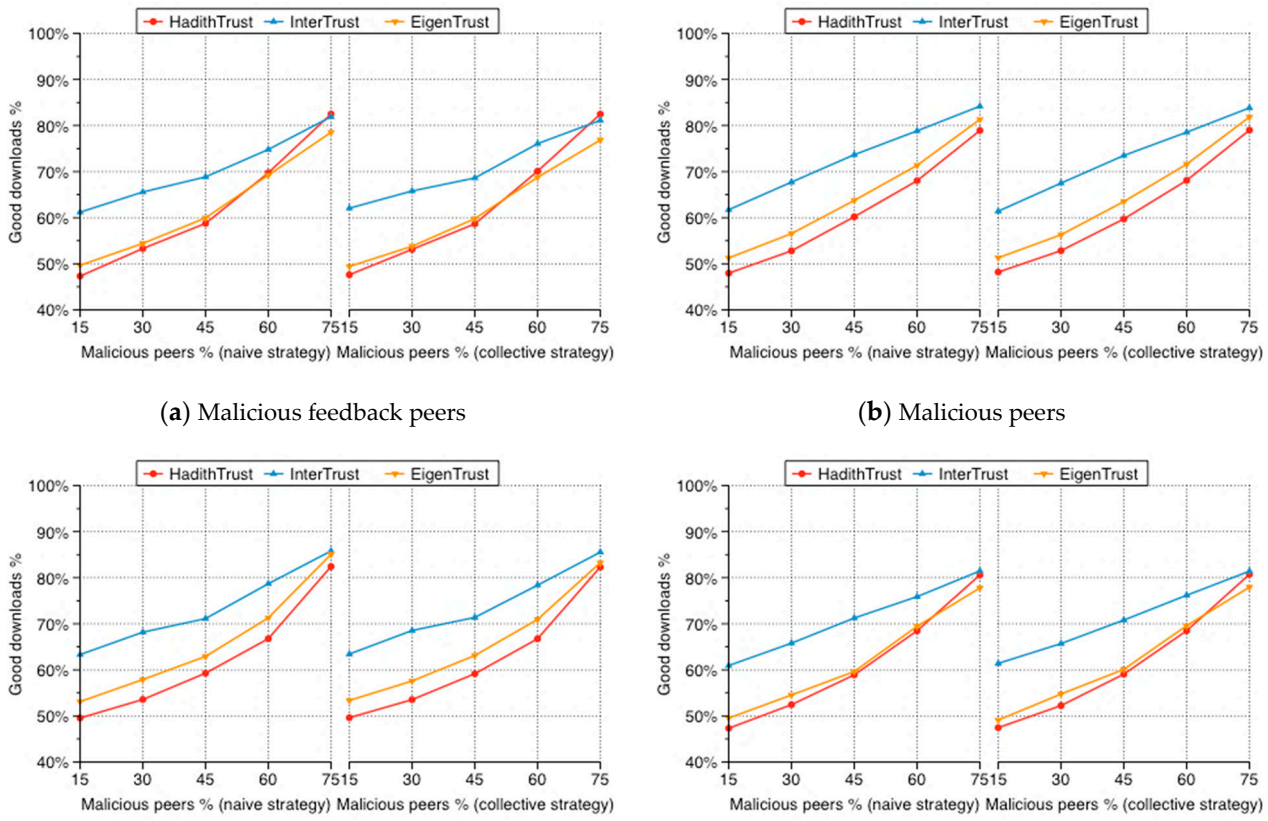
(**b**) Malicious peers

(**c**) Purely malicious peers

(**d**) Camouflaged malicious peers

Figure 13. Percentage of authentic downloads.

In Figure 13a, the percentage of authentic downloads for HadithTrust, EigenTrust, and InterTrust with increasing percentage of malicious feedback peers is shown. The percentage of authentic downloads for all algorithms dramatically decreases as the percentage of malicious feedback peers increases. Moreover, HadithTrust maintains a higher percentage of authentic downloads than EigenTrust and InterTrust when the percentage of malicious feedback peers is less than 75%. After this percentage, HadithTrust provides less authentic downloads than EigenTrust and more authentic downloads than InterTrust. To calculate authentic Isnads, two conditions are considered: valid Matn, which could be satisfied by malicious feedback peers when they are file providers, and a high Isnad reputation. In HadithTrust, the files from malicious feedback peers can be accepted if two conditions are met: the previous narrators in the file chain have high reputation values and the Matn is valid. Therefore, when the percentage of malicious feedback peers reaches 75%, they could raise their reputations at the expense of other peers. Thus, in HadithTrust, authentic Isnads decrease. However, the reduction only happens in one scenario. It is not a bad situation, because the percentage of good downloads–not bad downloads–increases in the same scenario, as shown in Figure 14a. Additionally, the success rate in the same scenario does not decrease. This is not the case for EigenTrust and InterTrust, in which the choice of file provider is based only on the provider's reputation.



(**a**) Malicious feedback peers

(**b**) Malicious peers

(**c**) Purely malicious peers

(**d**) Camouflaged malicious peers

Figure 14. Percentage of good downloads.

Figure 13b shows the percentage of authentic downloads for HadithTrust, EigenTrust, and InterTrust with the percentage of malicious peers varying from 15% to 75%. The percentage of authentic downloads for all systems falls rapidly as the percentage of malicious peers' increases. Moreover, as expected, HadithTrust has a higher percentage of authentic downloads in all scenarios. HadithTrust rejects files with an invalid Matn (inauthentic files) from malicious peers after they are downloaded once. Therefore, HadithTrust's

authentic downloads are still higher than those of EigenTrust and InterTrust, even when the percentage of malicious peers reaches 75%.

Figure 13c shows the percentage of authentic downloads for HadithTrust, EigenTrust, and InterTrust with varying percentage of purely malicious peers. The percentage of authentic downloads for all systems rapidly decreases, and HadithTrust has a higher percentage of authentic downloads in all cases (Figure 13b). However, when the percentage of purely malicious peers is 75%, authentic downloads by HadithTrust and EigenTrust are mostly similar because purely malicious peers combine the work of malicious feedback peers and malicious peers. Consequently, the authentic download percentage of HadithTrust with purely malicious peers falls between the authentic download percentage in the case of malicious feedback peers and malicious peers.

Figure 13d illustrates the authentic download percentage for HadithTrust compared with EigenTrust and InterTrust with increasing percentage of camouflaged malicious peers from 15% to 75%, respectively. Authentic downloads decrease as the percentage of camouflaged malicious increases, as with previous malicious models.

5.3. Percentage of Good Downloads

In Figure 14, the percentage of good downloads is plotted, calculated as the number of good Isnad downloads for good peers divided by the number of all Isnads downloaded by good peers. Three systems are compared: HadithTrust, EigenTrust, and InterTrust. Three variables are used: percentage of malicious peers, malicious strategy, and model of malicious peers. As in Section 5.2, the case of no-trust (None) is excluded from this measure.

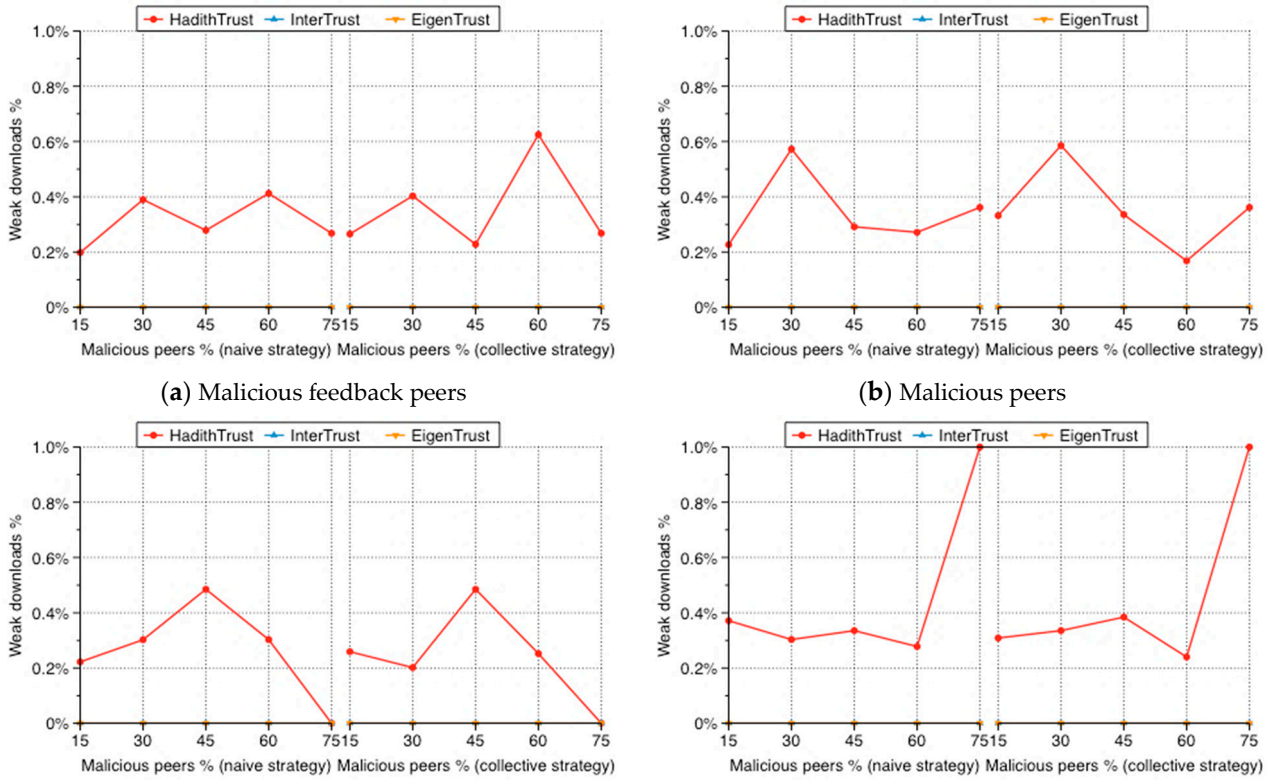
Figure 14a shows the percentage of good downloads for good peers with increasing percentage of malicious feedback peers for HadithTrust compared with EigenTrust and InterTrust. The graphs clearly show that the percentage of good downloads for all systems increases steadily when the percentage of malicious feedback peers increases. The performance of all systems is almost opposite to that shown in Figure 13a because all systems search for the most reputable provider, and in most cases falls under conditions of authentic or good Isnads. Thus, most of the other downloads, without authentic downloads, would be good downloads. Moreover, HadithTrust selects as many authentic Isnads as possible. Therefore, in most cases, HadithTrust provides a higher number of authentic downloads and a lower number of good Isnads.

In Figure 14b, the percentage of good downloads for all systems increases as the percentage of malicious peers increases for the same reason described above, which is opposite of that shown in Figure 13b. Figure 13c illustrates the percentage of good downloads with increasing percentage of purely malicious peers for HadithTrust compared with EigenTrust and InterTrust. The graphs clearly show that the percentage of good downloads for all systems increases in line with the percentage of malicious peers. This also is the opposite of the authentic download percentages shown in Figure 13c. Figure 14d shows the percentage of good downloads for the three systems as the percentage of camouflaged malicious peers varies from 15% to 75%. As expected, all algorithms provide good files in percentages with inverse relevance to the percentage of authentic download in the same scenarios shown in Figure 13d.

5.4. Percentage of Weak Downloads

Figure 15 illustrates the percentage of weak downloads for good peers in HadithTrust, EigenTrust, and InterTrust with the percentage of malicious peers varying from 15% to 75%. Generally, in all scenarios, the percentage of weak downloads for HadithTrust is 1% or less, whereas the percentage of weak downloads for EigenTrust and InterTrust is always zero. In addition, the variation in the percentage of weak downloads for HadithTrust is irrelevant, both to the model and to the percentage of malicious peers. In fact, weak downloads are those with a low Isnad reputation and an unknown Matn. Thus, according to the initial file distribution, all files have multiple copies, and intuitively for each file, some copies are owned by relatively reputable peers. Therefore, under these circumstances, the file with a

high Isnad reputation and an unknown Matn is considered a good file. Consequently, weak downloads are rarely chosen in HadithTrust. The only case when a weak Isnad is chosen is when no authentic or good Isnad is available for the requested file. However, in EigenTrust and InterTrust, the file provider is chosen on the basis of its reputation; therefore, the file owned by the most reputable provider always is chosen for downloads, even if the Matn is invalid, which is considered a bad Isnad.



(**a**) Malicious feedback peers

(**b**) Malicious peers

(**c**) Purely malicious peers

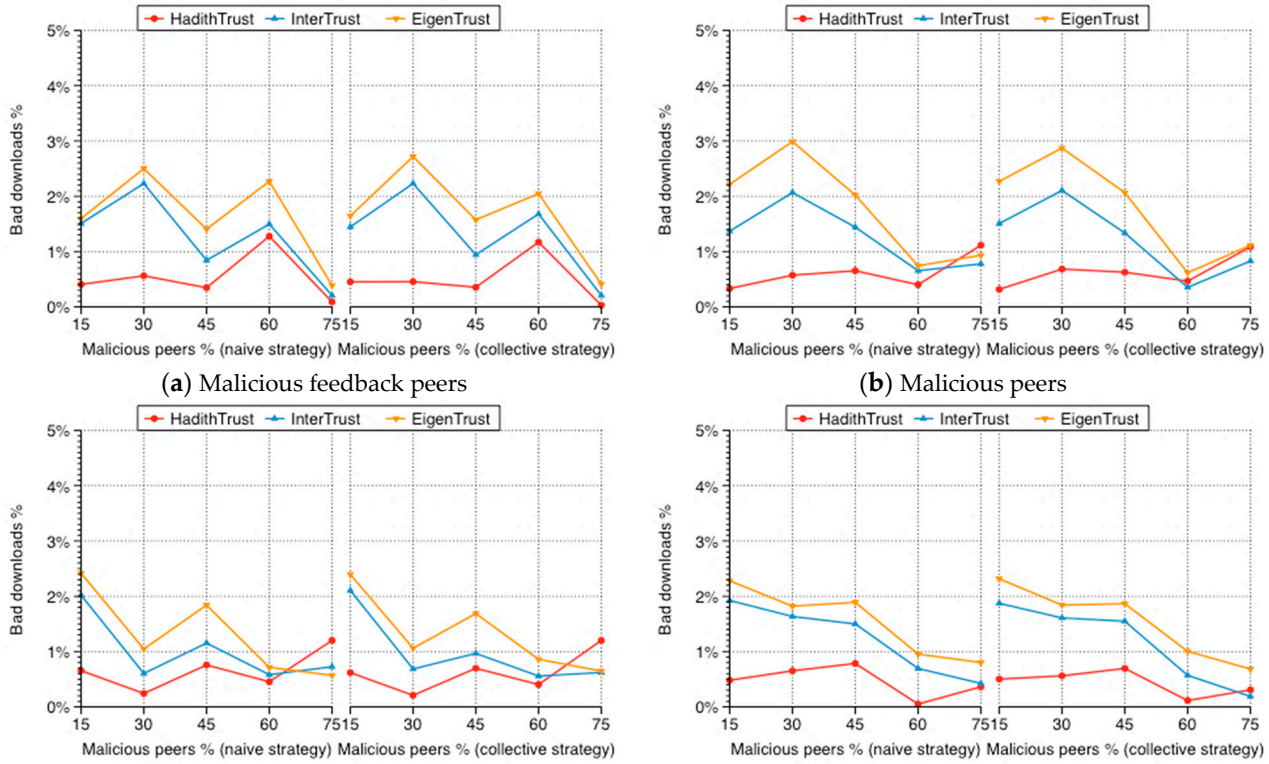
(**d**) Camouflaged malicious peers

Figure 15. Percentage of weak downloads.

To sum up, in all scenarios, HadithTrust offers 1% or fewer weak downloads, and benchmark systems never provide weak downloads. However, a weak file provided by a provider with a low reputation value is always better than a bad file provided by a highly trusted provider, which occurs with EigenTrust and InterTrust, as shown in Figure 15.

5.5. Percentage of Bad Downloads

Figure 16 shows the percentage of bad downloads for HadithTrust compared with EigenTrust and InterTrust with increasing percentage of malicious peers. The percentage of bad downloads is not relevant to the percentage of malicious peers or the type of malicious peers. Indeed, the percentage of bad downloads is proportional to two cases. In the first case's initial file distribution, peers own a low percentage of inauthentic files (from 0% to 10%). In the second case, when a pretrusted peer downloads an inauthentic file, the file is spread to other peers since it comes from a trusted peer. In both cases, in EigenTrust and InterTrust, the bad file is downloaded on the basis of the file provider's reputation regardless of the file's authenticity. In contrast, HadithTrust searches for better Isnads, even if the file provider is not the most reputable one. Therefore, in most scenarios in Figure 16, HadithTrust provides the lowest percentage of bad files.



(**a**) Malicious feedback peers

(**b**) Malicious peers

(**c**) Purely malicious peers

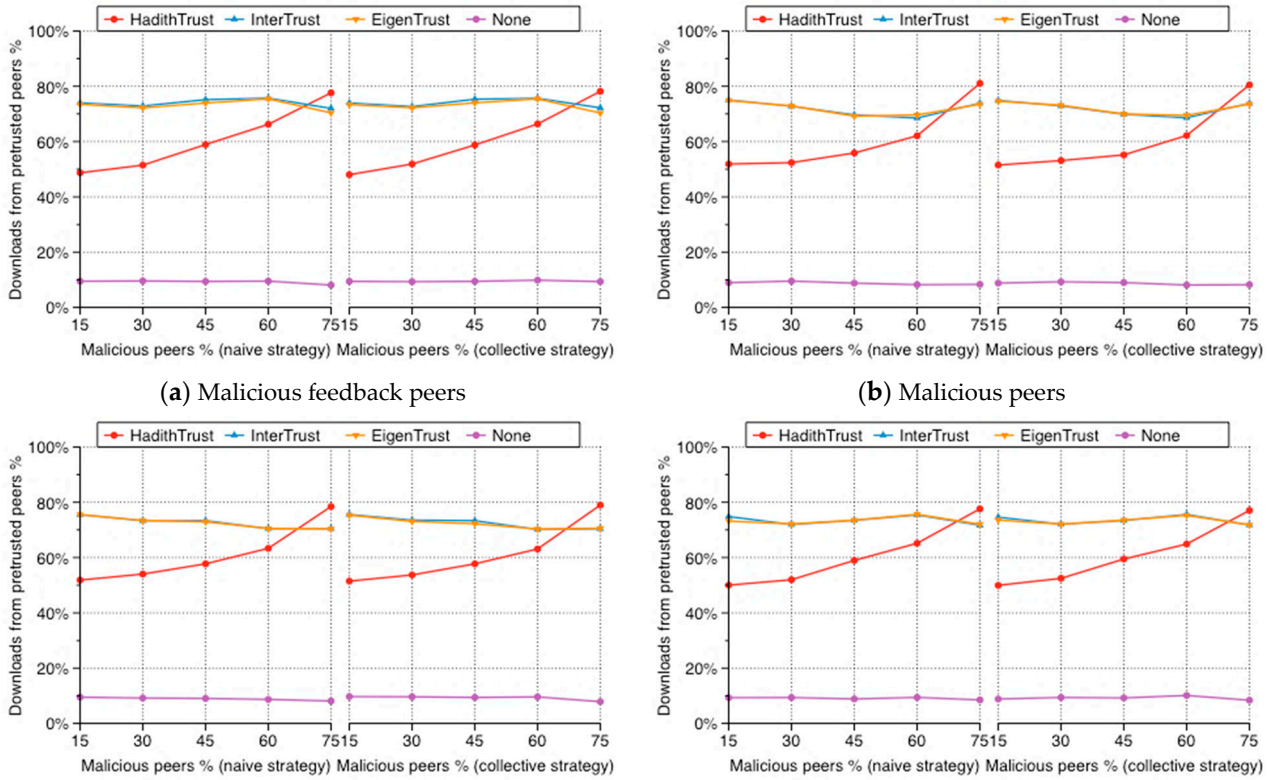
(**d**) Camouflaged malicious peers

Figure 16. Percentage of bad downloads.

5.6. Percentage of Downloads from Pretrusted Peers

Figure 17 illustrates the percentage of good peers' downloads from pretrusted peers, calculated as the number of good peers' downloads from pretrusted peers divided by the number of all good peers' downloads, for HadithTrust, EigenTrust, InterTrust, and no-trust (None) systems with the percentage of malicious peers varying from 15% to 75%. It is better to give good peers more chances to upload files to raise their reputations and decrease congestion around pretrusted peers. However, it is clearly shown in Figure 17 that HadithTrust, in most scenarios, maintains a lower percentage of downloads from pretrusted peers. In addition, the percentage of good peers' downloads from pretrusted peers when the no-trust (None) system is used is always ~10%, which is the actual percentage of pretrusted peers in the network.

In Figure 17a–d, the percentage of good peers' downloads from pretrusted peers for HadithTrust increases gradually as the percentage of malicious peers increases, which is reasonable because as malicious peers increase in the network, the percentage of pretrusted peers among the rest of the peers increases. In contrast, EigenTrust and InterTrust get almost 75% of peers' downloads from pretrusted peers when the percentage of malicious peers is 15%. Then, the percentage of good peers' downloads from pretrusted peers' decreases as the percentage of malicious peers' increases. In fact, in EigenTrust, pretrusted peers are given extra weight in reputation calculations, and in InterTrust, pretrusted peers are given the highest-possible values in initial opinion (uncertainty = 1 and base rate = 1). Therefore, most downloads come from pretrusted peers. In contrast, in HadithTrust, reputation is divided into four quartiles, and the Matn is considered when selecting a file provider. Therefore, when the Matn is valid, it is enough for peers to have a reputation within Q1 to construct a good Isnad and select for downloads (case 1 in good Isnads). However, when pretrusted peers have an unknown Matn, the Isnad is also classified as good (case 2 in good Isnads). Therefore, good peers are given a higher probability of providing files for others without scarifying the success rate.



(**a**) Malicious feedback peers

(**b**) Malicious peers

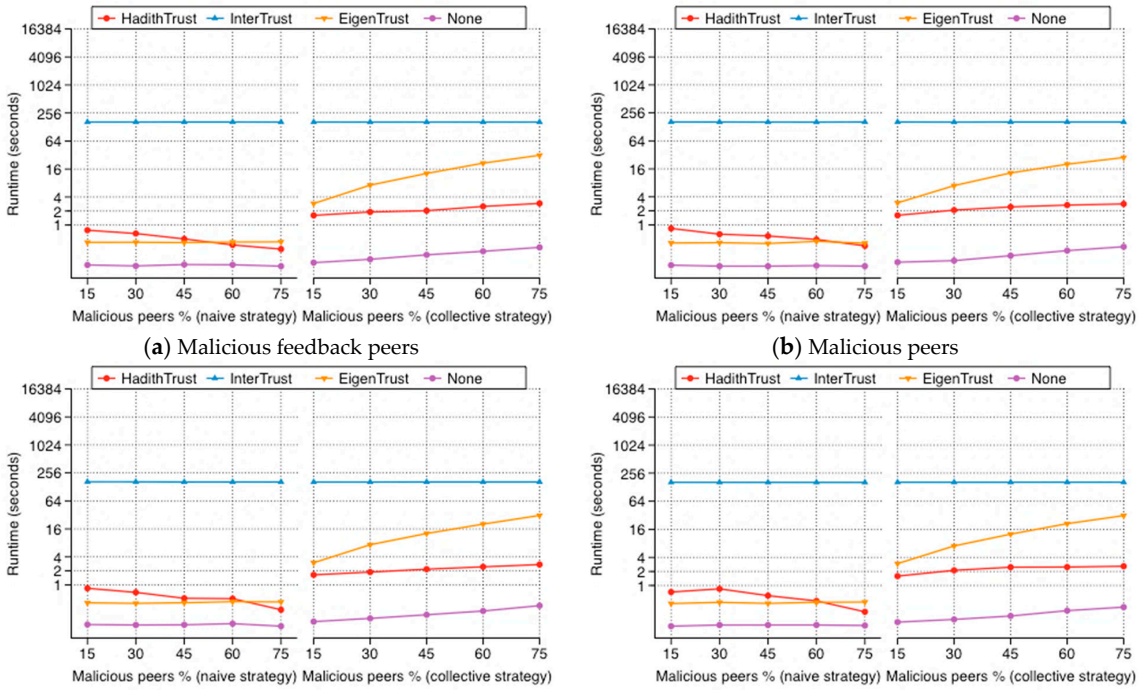
(**c**) Purely malicious peers

(**d**) Camouflaged malicious peers

Figure 17. Percentage of good peers' downloads from pretrusted peers.

5.7. Running Time

Figure 18 illustrates the running time, starting from 0 to 4.5 h (in seconds), with steps of powers of 2, for HadithTrust, EigenTrust, InterTrust, and no-trust (None) systems with increasing percentage of malicious peers. Four malicious models (malicious feedback, malicious, purely malicious, and camouflaged malicious peers) and two malicious strategies (naive and collective) were examined. In contrast to the success rate, the running time is obviously different between the two malicious strategies, because with the collective malicious strategy, malicious peers form a group and then local trust values are calculated and propagated for the group members, which takes extra time. Moreover, the no-trust (None) system is the fastest. In addition, for each algorithm, the difference between the running time with different malicious models is marginal because the same calculations are performed regardless of the model. With the naive strategy, HadithTrust is slightly slower than EigenTrust because of the calculations for Isnad and Matn during each transaction. However, as the percentage of malicious peers' increases, HadithTrust becomes faster and overtakes EigenTrust because HadithTrust denies file requests from malicious peers. Thus, as the malicious peer percentage increases, the algorithm works more quickly. Generally, HadithTrust runs more quickly than InterTrust and in some scenarios more quickly than EigenTrust. The no-trust (None) system is always faster because it has no-trust calculations.



(**a**) Malicious feedback peers (**b**) Malicious peers

(**c**) Purely malicious peers (**d**) Camouflaged malicious peers

Figure 18. Running time.

5.8. Transaction Service Time

Table 2 shows the average transaction service time in milliseconds for HadithTrust, EigenTrust, InterTrust, and no-trust (None) systems with the percentage of malicious peers varying from 15% to 75%. For each algorithm we compute the average transaction service time for the four considered malicious models (malicious feedback, malicious, purely malicious, and camouflaged malicious) in both strategies (naïve and collective). It is clearly shown that HadithTrust has the best performance among the four systems, except for no-trust since no trust calculation is performed. However, the difference between EigenTrust and HadithTrust is marginal since both algorithms utilized similar formulas to compute the peers' reputation. Moreover, InterTrust is the slowest algorithm because of the heavy opinions' calculations involved in each transaction.

Table 2. Average transaction service time.

Algorithm	Malicious%	Time Per Trans (ms)
InterTrust	15%	47.5
EigenTrust		15.2
HadithTrust		10.2
None		1.9
InterTrust	30%	47.1
EigenTrust		15.2
HadithTrust		10.8
None		1.9
InterTrust	45%	47.3
EigenTrust		15.9
HadithTrust		11.9
None		1.8
InterTrust	60%	47.1
EigenTrust		16.6
HadithTrust		13.2
None		1.9
InterTrust	75%	46.8
EigenTrust		17.1
HadithTrust		15.9
None		1.7

6. Conclusions

Trust management systems are employed to measure trustworthiness among peers. One of the main drawbacks of existing trust management systems is the binary classification for files as either authentic or inauthentic, which does not reflect real-world files and affects overall trust evaluations within systems.

In this paper, we proposed HadithTrust as a new trust management system inspired by the classification model from Hadith science. In this system, we considered the validity of Matns, the trustworthiness of a file on the basis of file ratings, and Isnads, the trustworthiness of the narrators in the file chain with respect to Matns to select file providers. The hybrid Isnad–Matn approach offers multiple benefits. First, by checking the Matn type, the possibility of malicious peers becoming file providers is minimized, thereby reducing the propagation of malicious files on the whole network. Second, by checking the APct value and involving it in reputation calculations, good peers are given a greater opportunity to provide good files to others and raise their reputations, rather than relying on pretrusted peers who come with multiple risks. Third, by selecting the best Isnads in each download, the downloaded files' overall quality in the network is enhanced.

HadithTrust is superior in terms of the success rate of good peers, while maintaining a low percentage of downloads from pretrusted peers. In addition, as the percentage of malicious peers' increases, HadithTrust's success rate becomes more stable than benchmark systems. Moreover, by fuzzifying the file trust to four levels, HadithTrust's downloads offer better quality compared with benchmark systems. We measured download quality using four metrics inspired by Hadith science: authentic, good, weak, and bad downloads. When considering percentages of all downloads (authentic, good, weak, and bad) overall, HadithTrust provides the best file quality because it selects the best files for each transaction, selecting them in the order authentic > good > weak > bad. In addition, HadithTrust maintains a minimum running time as it denies malicious requests, rather than providing bad files, which takes extra time.

As a future work, HadithTrust approach can be extended to other types of networks such as social networks and IoT. Moreover, digital forensics evidence is sent from a user to others forming a file chain that can be analyzed using HadithTrust approach [29]. Thus, the digital investigators can assess the credibility of the digital forensics evidence.

Author Contributions: Conceptualization, A.A. and M.A.; Formal analysis, A.A. and H.K.; Investigation, H.K.; Methodology, A.A. and M.A.; Software, A.A.; Supervision, H.K.; Validation, A.A.; Visualization, A.A.; Writing—original draft, A.A.; Writing—review & editing, H.K. and A.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by a grant from Researchers Supporting Unit, Project number (RSP- 2021/204), King Saud University, Riyadh, Saudi Arabia.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hou, L.-Y.; Tang, T.-Y.; Liang, T.-Y. OTA-BT: A P2P file-sharing system based on IOTA. *Electronics* **2020**, *9*, 1610. [CrossRef]
2. Almuzaini, F.; Alromaih, S.; Althnian, A.; Kurdi, H. WhatsTrust: A trust management system for WhatsApp. *Electronics* **2020**, *9*, 2190. [CrossRef]
3. Zhou, R.; Hwang, K. PowerTrust: A robust and scalable reputation system for trusted peer-to-peer computing. *IEEE Trans. Parallel Distrib. Syst.* **2007**, *18*, 460–473. [CrossRef]
4. Li, X.; Zhou, F.; Yang, X. Scalable Feedback Aggregating (SFA) overlay for large-scale P2P trust management. *IEEE Trans. Parallel Distrib. Syst.* **2012**, *23*, 1944–1957. [CrossRef]
5. Wang, Y.; Vassileva, J. Trust and reputation model in peer-to-peer networks. In Proceedings of the Third International Conference on Peer-to-Peer Computing (P2P2003), Linköping, Sweden, 1–3 September 2003; pp. 150–157.
6. Kamvar, S.D.; Schlosser, M.T.; Garcia-Molina, H. The Eigentrust algorithm for reputation management in P2P networks. In Proceedings of the 12th International Conference on World Wide Web—WWW '03, New York, NY, USA, 20–24 May 2003; pp. 640–651.
7. Josang, A.; Ismail, R.; Boyd, C. A survey of trust and reputation systems for online service provision. *Decis. Support Syst.* **2007**, *43*, 618–644. [CrossRef]

8. Kurdi, H.; Alnasser, S.; Alhelal, M. AuthenticPeer: A reputation management system for peer-to-peer wireless sensor networks. *Int. J. Distrib. Sens. Netw.* **2015**, *11*, 637831. [CrossRef]
9. Alharbi, A.R.; Aljaedi, A. Peer-to-peer network security issues and analysis: Review. *IJCSNS* **2020**, *20*, 74–88.
10. Chuang, Y.-T.; Li, F.-W. TCR: A trustworthy and churn-resilient academic distribution and retrieval system in P2P networks. *J. Supercomput.* **2020**, *76*, 7107–7139. [CrossRef]
11. Hajar, I. *Taqrib Al-Tahzib*, 1st ed.; Halab: Dar Alrasheed, Saudi Arabia, 1986.
12. Kurdi, H.A. HonestPeer: An enhanced EigenTrust algorithm for reputation management in P2P systems. *J. King Saud Univ. Comp. Inf. Sci.* **2015**, *27*, 315–322. [CrossRef]
13. Chiluka, N.; Andrade, N.; Gkorou, D.; Pouwelse, G. Personalizing EigenTrust in the face of communities and centrality attack. In Proceedings of the 2012 IEEE 26th International Conference on Advanced Information Networking and Applications, Fukuoka, Japan, 26–29 March 2012; pp. 503–510.
14. Carchiolo, V.; Longheu, A.; Malgeri, M.; Mangioni, G. The effects of pre-trusted peers misbehaviour on EigenTrust. In Proceedings of the 6th International Symposium on Intelligent Distributed Computing (IDC), Calabria, Italy, 24–26 September 2012; pp. 187–197.
15. Abrams, Z.; McGrew, R.; Plotkin, S. A non-manipulable trust system based on EigenTrust. *ACM SIGecom Exch.* **2005**, *5*, 21–30. [CrossRef]
16. Lu, K.; Wang, J.; Li, M. An Eigentrust dynamic evolutionary model in P2P file-sharing systems. *Peer-to-Peer Netw. Appl.* **2016**, *9*, 599–612. [CrossRef]
17. Nishikawa, T.; Fujita, S. A reputation management scheme for peer-to-peer networks based on the EigenTrust trust management algorithm. *J. Inf. Process.* **2012**, *20*, 578–584. [CrossRef]
18. Jøsang, A.; Hayward, R.; Pope, S. Trust network analysis with subjective logic. In Proceedings of the 29th Australasian Computer Science Conference, Darlinghurst, Australia, 16–19 January 2006; pp. 85–94.
19. Kurdi, H.; Alshayban, B.; Altoaimy, L.; Alsalamah, S. TrustyFeer: A subjective logic trust model for smart city peer-to-peer federated clouds. *Wirel. Commun. Mob. Comput.* **2018**, *2018*, 1073216. [CrossRef]
20. Kurdi, H.; Alfaries, A.; AI-Anazi, A.; Alkharji, S.; Addegaither, M.; Altoaimy, L.; Ahmed, S.H. A lightweight trust management algorithm based on subjective logic for interconnected cloud computing environments. *J. Supercomput.* **2019**, *75*, 3534–3554. [CrossRef]
21. Lee, Y.J.; Lee, K.M.; Lee, S.H. Blockchain-based reputation management for custom manufacturing service in the peer-to-peer networking environment. *Peer Peer Netw. Appl.* **2020**, *13*, 671–683. [CrossRef]
22. Zhuang, Q.; Liu, Y.; Chen, L.; Ai, P. Proof of reputation: A reputation-based consensus protocol for blockchain based systems. In Proceedings of the 2019 International Electronics Communication Conference, Okinawa, Japan, 7–9 July 2019; pp. 131–138.
23. Javanmardi, S.; Shojafar, M.; Shariatmadari, S.; Ahrabi, S.S. FR TRUST: A fuzzy reputation-based model for trust management in semantic P2P grids. *Int. J. Grid Util. Comput.* **2015**, *6*, 57–66. [CrossRef]
24. Almogren, A.; Mohiuddin, I.; Din, I.U.; Almajed, H.; Guizani, N. FTM-IoMT: Fuzzy-Based Trust Management for Preventing Sybil Attacks in Internet of Medical Things. *IEEE Internet Things J.* **2021**, *8*, 4485–4497. [CrossRef]
25. Lee, S.Y.; Kwon, O.-H.; Kim, J.; Hong, S.J. A reputation management system in structured peer-to-peer networks. In Proceedings of the 14th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprise (WETICE'05), Linköping, Sweden, 13–15 June 2005.
26. Walsh, K.; Sirer, E.G. Fighting peer-to-peer SPAM and decoys with object reputation. In Proceedings of the 2005 ACM SIGCOMM Workshop on Economics of Peer-to-Peer Systems, Philadelphia, PA, USA, 22–26 August 2005; pp. 138–143.
27. Alkharji, S.; Kurdi, H.; Altamimi, R.; Aloboud, E. AuthenticPeer++: A trust management system for P2P networks. In Proceedings of the 2017 European Modelling Symposium (EMS), Manchester, UK, 20–21 November 2017; pp. 191–196.
28. West, A.G.; Kannan, S.; Lee, I.; Sokolsky, O. An evaluation framework for reputation management systems. In *Trust Modeling and Management in Digital Environments: From Social Concept to System Concept*; Zheng, Y., Ed.; IGI Global: Hershey, PA, USA, 2009; pp. 282–308.
29. Yusoff, Y.; Ismail, R.; Hassan, Z. Adopting hadith verification techniques into digital evidence authentication. *J. Comput. Sci.* **2010**, *6*, 613–618. [CrossRef]