Privacy Protection Routing and a Self-organized Key Management Scheme in Opportunistic Networks

Yang Qin(✉), Tiantian Zhang, and Mengya Li

School of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), Shenzhen, China
csyqin@hit.edu.cn, 3344316263@qq.com, 1532732482@qq.com

Abstract. The opportunistic network adopts the disconnected store-and-forward architecture to provide communication support for the nodes without an infrastructure. As there is no stable communication link between the nodes, so that forwarding messages is via any encountered nodes. Social networks based on such opportunistic networks will have privacy challenges. In this paper, we propose a privacy protection scheme routing based on the utility value. We exploit the Bloom filter to obfuscate the friends lists and the corresponding utility values of nodes in order to make the routing decisions. This is easy to implement with high performance. Considering no infrastructure and stable link in opportunistic networks, this paper presents a self-organized key management system consisting of an identity authentication scheme based on the zero-knowledge proof of the elliptic curve and a key agreement scheme based on the threshold cryptography. The nodes prove their identities by themselves, and each node carries a certificate library to improve the authentication efficiency and success rate. In order to ensure the forward security and improve the session key agreement rate and the success rate, we exploit threshold cryptography to divide the session key, which could reduce the communication consumption of the traditional Diffie-Hellman (DH) algorithm. The experimental simulation results show that the proposed schemes are much better than the existing schemes for opportunistic networks.

Keywords: Opportunistic network · Routing · Privacy protection · Key management system

1 Introduction

In recent years, the rapid popularization and development of mobile devices have promoted many new technologies with taking advantages of their growing processing power and storage space. One of the most rapidly developing technologies is the opportunistic network, which organizes these mobile devices in a disconnected ad hoc manner. Such opportunistic networks can be used to create new applications, such social networks, et.al. However, because of no infrastructure and stable communication

X. Chu et al. (Eds.): QShine 2019, LNICST 300, pp. 252–268, 2020.
https://doi.org/10.1007/978-3-030-38819-5_16

link, traditional security schemes, such as the public key infrastructure (PKI), are not suitable for opportunistic networks. Shikfa [1] pointed out the security problems in opportunistic networks, such as node selfishness [2, 3], routing security, and privacy protection. To ensure the practicability of the network, security issues, such as identity authentication and message transmission, must be considered. Since the opportunistic network needs to rely on intermediary devices to forward messages, there may be different privacy issues for different routing protocols. For example, the routing protocol that forwards messages based on the similarity of user attributes [4–6] needs to protect the attribute information. The routing protocol that forwards messages based on the utility values [7–10] needs to protect the friends list and the corresponding utility values. The routing protocol that forwards messages based on location information [11] needs to protect the location of the node. In addition, the social network routing forwards messages via nodes in the sender or recipient's social networks, which may disclose the friends lists of nodes. So the privacy disclosure is an important problem with social networks. How to choose the appropriate next-hop node while protecting the user's privacy is a hot topic in current research.

In order to ensure opportunistic networks security and availability, the key management system must be considered. The traditional PKI key management system based on certificate authority (CA) is not applied to opportunistic networks because there is no infrastructure and all nodes are equally self-organized in opportunistic networks. There is no CA that can always stay online to obtain the node's public key certificate.

In addition, during message transmission, in order to implement the PGP, forward security usually needs to be maintained. The existing solution generally adopts the DH key agreement protocol [9] or its variants, such as ECC-based key agreement protocol [10]. However, the communication overhead is relatively large, and thus is not applicable in the opportunistic network due to the high delay. Therefore, new algorithms are needed to solve these problems.

Therefore, we propose a privacy protection routing scheme and a self-organized key management scheme, which makes the following contributions:

- We introduce the Bloom filter to protect nodes privacy in the opportunistic network. We use the Bloom filter to store the friends list and the corresponding utility values of nodes, which obfuscates node privacy and introduces uncertainty to ensure network security.
- We present a self-organizing key management system based on the zero-knowledge proof of the elliptic curve and the threshold cryptography, which consists of identity authentication and key agreement. The system generates certificates, including identity information, public and private keys, and TTL relying on nodes themselves. In addition, the nodes also prove their identity on their own.
- We exploit threshold cryptography to divide the shared session key, and separate key agreement from message transmission to ensure forward security and speed up the session key agreement procedure.

The rest of the paper is organized as follows. Section 2 shows the related works. Section 3 is the detailed design of the privacy protection scheme. Section 4 is the design of the self-organized key management scheme, including identity authentication based on the zero-knowledge proof and the key agreement algorithm based on the threshold cryptography. We evaluate the performance of all schemes proposed in the paper in Sect. 5. And Sect. 6 concludes the paper.

2 Related Works

In order to prevent malicious nodes from stealing the privacy of the node, Cadger and Curran [11] separated the privacy of the node from the real ID of the node to protect the node's privacy by geographic routing. Zhi [12] proposed an anonymous geographic routing algorithm. An anonymous table was used to store the node's fake ID and the corresponding geographic locations in order to avoid the leakage of identity and location information by geographic routing in communication. Zhou [13] proposed a novel threshold credit incentive strategy (TCBI) for a vehicle delay tolerant network and a TCBI-based privacy-preserving packet forwarding protocol, which can resist harmful attacks on vehicles and protect vehicle privacy well. Pidgin [14] is a privacy-preserving interest and content sharing scheme for opportunistic networks that does not disclose privacy to the untrusted party. Its main idea is to use CP-ABE to regulate content access, and Pidgin uses public key encryption and keyword search (PRKS) scheme to encrypt plaintext CP-ABE policy to protect privacy. TRSS [15] is a trust routing scheme based on social similarity, which establishes the social trust of nodes according to the trustworthiness of nodes and their encounter history. On the basis of social trust, the untrusted nodes are detected and deleted from the trusted list. When forwarding messages, only the data packets of trusted nodes are forwarded to ensure system security. Boldrini [16] proposes a context-aware framework for routing and forwarding in opportunistic networks, which uses user behavior and social relationships to drive the forwarding process. The framework divides network users into groups by using key management, and protects the privacy of nodes through strict contact control. However, introducing key management is too complex for opportunistic networks. Parris et al. [17] proposed the Obfuscated Social Network Routing embedding each node in the nodes' social networks into a Bloom filter, and making a routing decision according to the Bloom filter. However, the Obfuscated Social Network Routing - only obfuscating the node's identifier – cannot compare the utility values to make the correct decision.

In order to be able to obtain the user's public key without a CA, Shamir [18] proposed an identity-based cryptosystem (IBC), which allowed users to verify other users' signatures without exchanging public keys. However, the scheme assumed the existence of a trusted key generation center. Boneh [19] first proposed a practical algorithm for identity encryption. The system was based on the Weil pairing and has chosen ciphertext security. Seth [20] proposed a hierarchical identity-based

cryptograph (HIBC), which was the first proposed identity cryptography for DTN networks. This scheme proposed a solution that used opportunistic connections to initiate secure channels by disconnected users, authenticate each other through opportunistic links, and protect disconnected users from identity leakage attacks. Kumar [21] proposed a secure and effective threshold key distribution protocol. The protocol does not require any secure channel to issue the private key, and is secure until the threshold number of KPAs is compromised. Another solution is the certificateless encryption proposed by Al-Riyami and Paterson [22], which needs a trusted third-party key generation center (KGC) that contains the system's master key. Liu [23] proposed IKM, an identity based key management scheme, which is a new combination of identity based and threshold cryptography. IKM is a certificateless solution, because the public key of mobile node can be directly derived from its known IDs plus some public information, which eliminates the need of certificate-based authentication public key distribution in traditional public key management scheme. Capkun [24] proposed an ad-hoc key management system in which each node acts as a CA to authenticate other nodes, eventually forming a chain of certificates that authenticate the node by looking up the chain of certificates. However, the certificate chain needs to form a complete trusted link, which is less efficient and has a lower success rate of authentication.

3 Privacy Protection Routing Scheme

In this paper, a scheme based on the Bloom filter is proposed to obfuscate the friends list and the corresponding utility values. It can protect the privacy of nodes.

3.1 Bloom Filter

The Bloom filter is a probabilistic data structure that maps elements to vectors by multiple hashes, which supports the probabilistic querying. Here, it is assumed that the *ID* information ID_1 of a node's friend is embedded within the Bloom filter. First use *10* different random number generators (F_1, F_2, ..., F_{10}) to generate *10* fingerprints information (f_1, f_2, ..., f_{10}). Then using a random number generator *G* maps (f_1, f_2, ..., f_{10}) to *10* integers g_1, g_2, ..., g_{10} in the range of *1* to *100,000*, and set the value of the *10* positions to *1* (see Fig. 1).

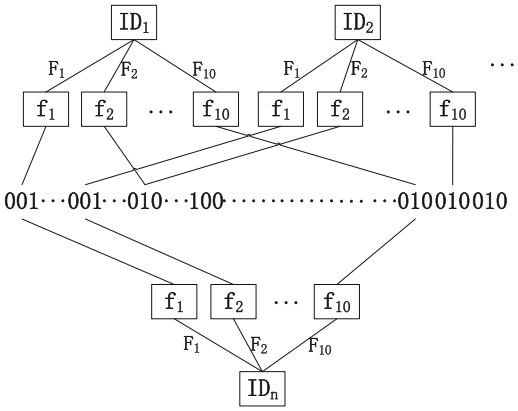


Fig. 1. The principle of the Bloom filter.

Suppose that we want to determine whether the node ID_n is in the Bloom filter. We use the same mapping function to execute the same processing, and observe whether all the *10* positions are *1*. If the nodes are in the Bloom filter, they must be all *1*. However, at this time, a "false positive" may occur, that is, the actual node is not in the, but all its positions are found to be *1*. This is because these *10* positions may be cross-mapped by other nodes, so that misjudgment may occur. We evaluate the misjudgment probability further in Sect. 3.3. Because of the misjudgment, the Bloom filter is used to protect the privacy.

3.2 Privacy Protection Routing Scheme

Parris et al. [17] adopted the Bloom filter to obfuscate the friends list to protect privacy, which can only verify whether node *a* satisfies the Bloom filter. Assuming that node *a* forwards a message to node *d* through intermediate node *b* or intermediate node *c*, the next-hop node needs to be correctly chosen without knowing the accurate utility values of node *b* and *c* reaching to node *d*. However, the scheme Parris proposed cannot compare the utility values and make correct decisions. Therefore, this paper makes improvements to this scheme by adding more information about the utility values in the Bloom filter. As shown in Table 1, we add a binary vector (Vector *2*) in which each element corresponds to each element of the previous binary vector (Vector *1*).

Table 1. The information is stored by nodes.

Vector	Description	Example
1	Storage of the friend information of the node	1010110010101…10010…010000
2	Storage of the utility value of the node	0010010010100…10010…010000

It is used to store utility values of the node reaching to other nodes. For example, the utility value of a node reaching to another node is *0.7*. Then *7* positions among *10* positions chosen randomly, where the friend node *ID* is hashed in Vector *1*, in Vector *2* to *1* (assuming that *10* hash functions are chosen).

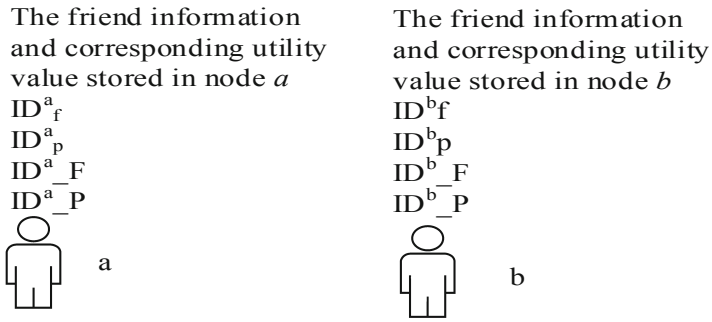


Fig. 2. The information is stored in nodes.

Algorithm 1：Initialize Network

```
Procedure initNetwork()
m presents the size of vectors in the Bloom filter, and the initial value of m is set to 500.
k refers to the number of hash functions in the Bloom filter, and its initial value is 10.
n refers to the size of the network.
for ID^i in network:
  init(ID^i_f) //Initialize the friends list of ID^i.
  init(ID^i_p) //Initialize the utility values list of ID^i.
The initial value of P_INIT is set to 0.75.
The initial value of BETA is set to 0.25.
The initial value of GAMMA is set to 0.98.
end procedure
```

After initialization by Algorithm *1*, supposing that node *a* encounter node *b*, Fig. 2 shows information stored in the nodes. If node *a* needs to send a message to node *d*, node *b* transmits ID_f^b and ID_p^b to node *a*. Node *a* decides whether to send the message to node *b* by comparing P_{f_bd} which is the probability that node *b* reaches the destination node *d* with P_{t_ad} which is the probability that node *a* itself reaches the destination node *d*. Then ID^a_F, ID^a_P, ID_f^a and ID_p^a are updated, which includes an aging update and

direct encounter nodes and transfer node update. At this time, updating and forwarding operations are carried out on node a via executing Algorithm *2*, and the same operations are carried out on node *b*.

Algorithm 2: Aging Updating and The Utility Value Update of Directly Encountered Nodes and Transferring Nodes

1. Procedure updatePrediction (ID^a, ID^b)
2. Node *b* transmits ID^b_f and ID^b_p to node *a*.
3. for *c* in ID^a_F:
4. $P_{t\_ac} = P_{t\_ac\_old} * (GAMMA\hat{}\ timeDiff)$
5. // Predict aging probability of P_{t_ac}.
6. $P_{t\_ab} = P_{t\_ab\_old} + (1 - P_{t\_ab\_old}) * P\_INIT$)
7. //Predict initial probability of P_{t_ab}.
8. if $ID^b_f[H^1_{ID}{}^d], ID^b_f[H^2_{ID}{}^d], \ldots, ID^b_f[H^k_{ID}{}^d] = 1$ then
9. $P_{f\_bd} = (ID^b_p[H^1_{ID}{}^d] + ID^b_p[H^2_{ID}{}^d] + \ldots + ID^b_p[H^k_{ID}{}^d])/k$
10. //Calculate the utility value of node *b* related to node *d*
11. if $P_{f\_bd} > P_{t\_ad}$ then
12. Transfer to node *b*
13. Reset ID^a_f, ID^a_p to *0* and update them according to ID^a_F and ID^a_P
14. for *c* in ID^a_F:
15. if *c* in ID^b_f then
16. $P_{t\_ac} = P_{t\_ac\_old} + (1 - P_{t\_ac\_old}) * P_{t\_ab} * P_{f\_bc} * BETA$
17. Calculate $H^1_{ID}{}^c, H^2_{ID}{}^c \ldots H^k_{ID}{}^c$
18. $ID^a_f[H^1_{ID}{}^d], ID^a_f[H^2_{ID}{}^d], \ldots, ID^a_f[H^k_{ID}{}^d] = 1$
19. Select Pt_ac*k random positions in the
20. corresponding ID^a_p positions, and set the
21. value of these positions to *1*. Assuming that there is
22. already *s* positions, then set the value of the other
23. Pt_ac*k-s random positions to *1* in addition
24. End procedure

Important notations for Algorithm 1 and 2 are summarized in Table 2.

Table 2. Notation

Symbol	Description
friends	Vector 1, mapping the friend *ID* to *k* different positions by *k* hash functions
predictions	Vector 2, storing the utility value
m	The length of friends and predictions
friends[i]	The value of the *ith* position in friends
predictions[i]	The value of the *ith* position in predictions
n	The size of the network
k	The number of hash functions in the Bloom filter
H^i	The ith hash function

(*continued*)

Table 2. (*continued*)

Symbol	Description
$H_{ID}^{i\_a}$	The positions that ID^a is hashed to by hash functions
ID_f^a	The friend array of ID^a
ID_p^a	The predictions array of ID^a
ID^a_F	The friends set of ID^a {ID^a_{F1}, ID^a_{F2}, …, ID^a_{Fn}}. Each node stores their friend list and utility value to generate the friends and predictions, which is the node privacy
ID^a_P	The friends' utility value set of ID^a {P_{t_ab}, P_{t_ac}, …,}. This is the node privacy
P_{t_ab}	The utility value of ID^a about ID^b
P_{f_ab}	The utility value of ID^a related to ID^b is calculated according to ID_f^a and ID_p^a
network	All nodes of the network
P_INIT	Predict probability initialization constant, $P_{t\_ab} = P_{t\_ab\_old} + (1 - P_{t\_ab\_old}) * P\_INIT$)
BETA	Predict probability transfer value scaling constant, and calculate the transferring probability from a to c through intermediate node b, $P_{t\_ac} = P_{t\_ac\_old} + (1 - P_{t\_ac\_old}) * P_{t\_ab} * P_{f\_bc} * BETA$
timeDiff	The time from the last update used to calculate the aging probability
GAMMA	Predict probabilistic aging constants, $P_{t\_ac} = P_{t\_ac\_old} * (GAMMA\^timeDiff)$

3.3 Misjudgment Probability Analysis

Assuming that there are m bits in each vector, the encountered node has n friends, and there are k random hash functions that are independent of each other. The friend node's *ID* is mapped to k positions which are set to *1* after hashing once, then the probability that a position is not set to 1 is $1 - 1/m$. The probability that this position is not set to *1* after hashing for k times is $(1 - 1/m)^k$. Then the probability that the position has not been set to *1* yet is $(1 - 1/m)^{kn}$ after n friends are hashed. So the probability of a position being set to *1* in an array is $1 - (1 - 1/m)^{kn}$. In order to determine whether a node is its friend, k hash functions are required to hash the node to k positions, so the probability of all these positions being 1 is $(1 - (1 - 1/m)^{kn})^k \approx (1 - e^{-kn/m})^k$. If the bloom filter is used for storage, the probability of misjudgment is generally very low for $k * n < m$, such as about 0.01 for $k * n = m$. However, when it is used for privacy protection as here, it can make $k * n > m$. Table 3 lists the comparison of misjudgment probability when $k = 10$ and m/(k * n) under different circumstances, where the greater the misjudgment probability, the greater the privacy protection degree.

Table 3. Comparison of misjudgment probability under different m/(k * n).

m/(k * n)	0.4	0.45	0.5	0.55	0.6	0.65	0.7	0.75	1
$(1 - e^{-kn/m})^k$	0.424	0.317	0.233	0.170	0.123	0.089	0.064	0.046	0.01

4 Self-organized Key Management System

Considering that there is no infrastructure in the opportunistic network such as the certificate authority (CA) and there is no stable link causing the communication overhead being relatively large, the traditional identity authentication scheme and key agreement scheme are not suitable. Therefore, according to the characteristics of opportunistic networks mentioned above, we design a self-organized key management system, including an identity authentication scheme based on the zero-knowledge proof and a key agreement scheme based on the threshold cryptography.

4.1 Identity Authentication Scheme

The identity authentication scheme is applied for opportunistic networks that automatically allocate IP addresses, such as the allocation scheme proposed by Weniger [25]. The detailed process of the scheme based on elliptic curve based the zero-knowledge proof protocol is shown below.

System Initialization. Firstly, broadcasting the generator P and hash function H over the whole network. Then each node i generates public and private key pairs (P_i, S_i), and private identity $X_i = [x_1, x_2, \ldots\ldots, x_k]$ and public identity $Q_i = [Q_1, Q_2, \ldots\ldots, Q_k]$ on its own. Then the node's certificate is $(Q_i, TTL, P_i, H(Q_i), S_i(TTL))$, where $H(Q_i)$ is the IP address of the node that is unique in the network. TTL is the life time of the public-private key pair (P_i, S_i). Finally, sending its certificate to the encountered node to store and verify.

Certificate Verification Phase. At the phase, if node A requests to verify the identity of node B, then node B transmits the certificate $(Q_i, TTL, P_i, H(Q_i), S_i(TTL))$ and $V = r * P$ (r is randomly generated by B) to A.

Firstly, node A checks whether the certificate is timeout. If not, $H(Q_i)$ is calculated and verifies whether the certificate is correct. If not, it indicates that the identity is wrong, otherwise it randomly generates k random numbers $(m_1, m_2, \ldots\ldots, m_k)$, and transmits them to node B. Then node B calculates $r + \sum_{i=1}^{k} m_i * x_i$ and returns to node A. Because $Q_i = x_i * P$, $V = r * P$, node A can verify whether $(r + \sum_{i=1}^{k} m_i * x_i) * P$ is equal to $V + \sum_{i=1}^{k} m_i * Q_i$. If so, B's public identity can be trusted.

Certificate Exchange Phase. Since there is no stable communication link between the source node and the destination node in an opportunistic network, if we use the above zero-knowledge proof authentication scheme to verify node's identity, the verification between the nodes needs to be verified through multiple communications. Therefore, if the destination node is not in the communication range of source node, the delay may be large and the efficiency is low.

Therefore, each node need store the certificates of other nodes locally. The number of certificates stored can be adjusted according to the node's own storage resources.

Messaging Phrase. Suppose node A wants to transmit a message to node B, then node A first checks whether there is the certificate of node B in the local trusted certificate library, and if so, it transmits, otherwise, node A performs the above authentication procedure.

Certificate Update Phase. Each node regularly updates the local certificate library. If P_i time out, the certificate will be removed from the certificate library, which is a passive update. In addition, if the node thinks that their public key is insecure, it can re-generate a new public-private key pair to form a new certificate and sends it to their neighbors.

4.2 Key Agreement Scheme

The key agreement protocol using the asymmetric key and DH key exchange algorithm can effectively resist man-in-the-middle attack and solve forward security. However, the DH key agreement protocol requires one-trip communication, whereas the zero-knowledge proof scheme requires two trips for identity authentication. Therefore, the DH key agreement protocol can be used to generate the session key during authentication. However, if the session key needs to be updated with the DH algorithm, the delay in the opportunistic network will be large.

Considering the characteristics of opportunistic networks, this paper presents a solution using threshold cryptography to encrypt the session key. The transmission process is divided into two parts. One part of the transmission for messages that is longer, and the other part of the transmission for session key, which is shorter.

We use the classical Lagrange interpolation polynomial threshold cryptography algorithm, proposed by Shamir [18], to divide the session key using (t, n) threshold scheme $(t \leq n)$, which divides it into n sub-session keys transmitted via the intermediate nodes to the destination node. If the number of sub-session keys transmitted successfully is greater than or equal to t sub-session keys, the destination node can restore the session key. Otherwise, the session key cannot be restored.

First, we need to select a finite field F_q, which satisfies the condition of $q \geq n$. Let t be the threshold, and encrypt the session key $SKey$. The source node sends $SKey's$ fragments to other intermediate nodes, and the intermediate nodes are represented by $p = \{p_1, p_2, \ldots\ldots, p_n\}$.

At the phase of key distribution, the source node randomly generates a $t-1$ degree polynomial G in a finite field $F_{q,}$ denoted as

$$g(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_{k-1} x^{t-1} \tag{1}$$

Then n sub-session keys $s_i = g(x_i)$, $i = 1, 2, \ldots\ldots, n$, are generated [18], which are transmitted to n encountered nodes.

Suppose that there are t nodes involved in the reconstruction, in which the master sub-session key is set as (i_r, s_{i_r}), $r = 1, 2 \ldots, t$. Then according to the Lagrange interpolation polynomial formula (2), we can calculate the $t - 1$ degree polynomial G [18].

$$\sum_{r=1}^{t} s_{i_r} \prod_{\substack{j=1 \\ j \neq r}}^{t} \frac{x - i_j}{s_{i_r} - s_{i_j}} \tag{2}$$

Then substitute $x = 0$ for the final calculation of the session key $SKey = g(0)$ [18]. It can be seen that only when at least t nodes collaborate, the forward security will be broken. The security of this scheme is high enough in some cases with low requirements.

4.3 Safety Analysis

Due to the puzzle of the elliptic curve encryption in zero-knowledge proof, an adversary cannot obtain r randomly generated by nodes. Then $S_A(\sum_{I=1}^{K} m_i * x_i)$ cannot be obtained, so the authentication protocol proposed in this paper is theoretically safe.

For key agreement scheme proposed in this paper, even if one party leaks fragments of the key, nodes in the network must have at least four fragments (for example, using the (4, 10) threshold) to restore a shared key, which greatly increases the difficulty for the malicious nodes to recover the session key. Moreover, since the shared key is generated by source node, the shared key can be transmitted at the same time with the message. Though the safety factor is smaller than using the DH algorithm, it is more suitable for the opportunistic network.

4.4 Performance Analysis

Figure 3 shows the interaction of the zero-knowledge proof authentication protocol.

Small Communication Overhead. Compared with other schemes using the certificate chain [26–28], authentication between nodes does not need to form a certificate chain. Even if the nodes are not in contact with each other, validation can be done via a certificate store or by actively sending authentication requests. Figure 3 shows that certificate verification requires two rounds of communication, with relatively low traffic.

This paper is based on ECC encryption authentication scheme, so the performance is relatively high.

High Reliability. Compared with authentication schemes based on identity cryptography, certificateless authentication and threshold cryptography, the proposed scheme in this paper are more suitable for the opportunistic network without an infrastructure, because all the schemes existing assume that there are trusted nodes in the network.

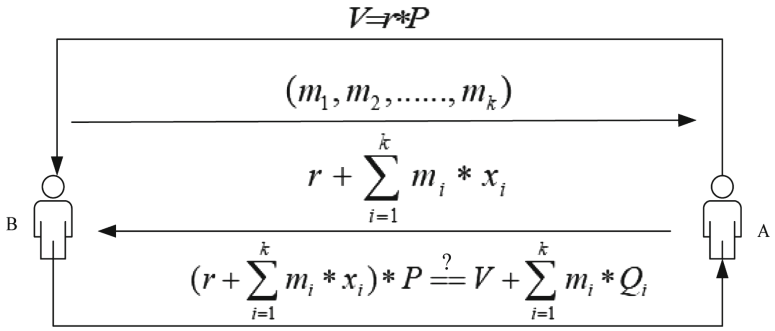


Fig. 3. Zero-knowledge Identity Authentication

Schemes based on threshold cryptography only solve the problem of private key hosting, but authentication requires the node to connect to a certain number of server nodes. The scheme proposed in this paper can be verified without trusted server nodes, so its reliability is higher than other schemes.

High Robustness. In this paper, the session key generated by one party is used to improve the success rate of communication, which is a completely self-organized authentication scheme.

For the session key encrypted with threshold cryptography, we consider that n nodes in the network carry session key fragments and the probability of reaching the destination node within the network lifetime (*TTL*) is p, then the probability of at least t nodes reaching the destination node within the *TTL* can be calculated according to binomial distribution

$$\sum_{i=t}^{n} p^i (1-p)^{n-i} \tag{3}$$

For DH key agreement algorithm, it is enough to have a single successful transmission, so the probability can be calculated as:

$$(1-(1-p)^n)^3 \tag{4}$$

In the ideal case where each node has the same probability of reaching the destination node, considering that $t = 4$, $n = 10$, and the p is relatively large, the probabilities are both basically closed. But for the expected time, the scheme proposed in this paper is a one-way transmission, while the DH algorithm is a round-trip transmission. In the specific environment, the success rate of the scheme proposed here is higher with smaller transmission delay.

5 Evaluation

5.1 Privacy Protection Experiment Design and Result Analysis

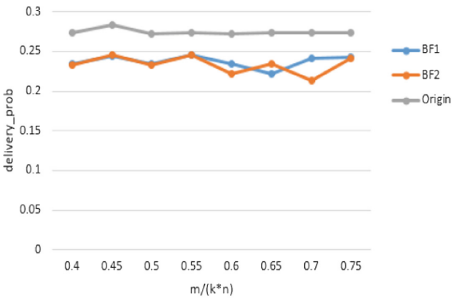
Simulation. According to Sect. 3.3, the ratio of $m/(k*n)$ can directly influence the misjudgment probability. The greater $m/(k*n)$, the smaller the misjudgment probability. And the greater the misjudgment probability, the greater the privacy protection level will be. So we used $m/(k*n)$ to measure the privacy protection level. We conducted simulation experiments to observe the effects of different privacy protection levels on message delivery probability, overhead and average latency in the opportunistic network.

The simulation experiment was carried out with *ONE* (Opportunitistic Network Environment simulator). Assuming that the size of the network n is 120, and the number of hash functions k is 10. The values of the vector's size m in the Bloom filter are [504; 567; 630; 693; 756; 819; 882; 945]. So $m/(k*n) = [0.4, 0.45, 0.5, 0.55, 0.6, 0.65, 0.7, 0.75]$. Considering that according to Algorithm 2, when mapping the utility values of the node to the Bloom filter, we need randomly choose $P_{f\_bd} * k$ positions from the k positions that the corresponding friend node *ID* is mapped to, and the $P_{f\_bd} * k$ positions are set to 1. So the values of P_{f_bd} may be different for each experiment under the same experiment environment. Then the repetitive experiment results may be different under the same experiment environment. Therefore, in order to ensure the accuracy of the experiment results, we performed two groups of experiments under the same experiment environment, denoted as *BF1* and *BF2*, respectively. The results show the comparisons of our scheme and ProphetRouter routing protocol without privacy protection, denoted as *Origin*, in terms of message delivery probability, overhead, and average latency.

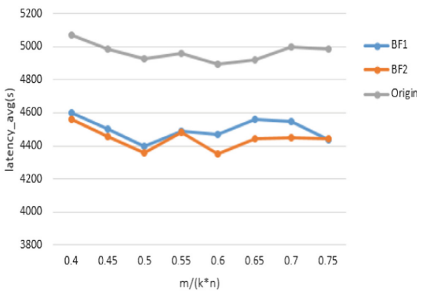
Result Analysis. Figure 4 shows that with the increase of the $m/(k*n)$, the message delivery probability decreases within an acceptable range, and the average transmission delay is significantly reduced, which may be associated with the decrease of message delivery probability. In addition, because the intermediate node needs to transmit ID_f^b and ID_p^b to the source node, the increase of overhead is relatively obvious. Though introducing the privacy protection scheme has decreased the message delivery probability, the privacy protection level of the network has been increased.

5.2 Key Agreement Experiment Design and Result Analysis

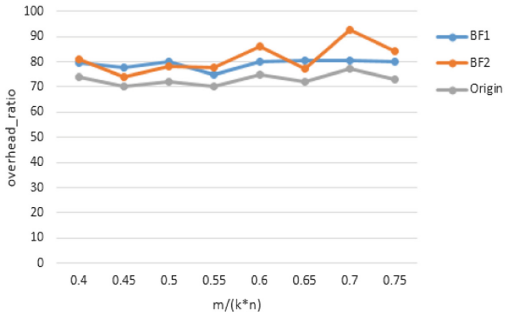
Simulation. The comparative experiment between DH algorithm and threshold cryptographic key agreement (TE) algorithm was also carried out with *ONE* using SprayAndWaitRouter routing protocol. We verify the performance of the TE algorithm according to three indicators: message delivery probability, overhead, and average latency in opportunistic networks. The number of hosts in the network is varied among [120, 150, 180, 210, 240, 270, 300]. We performed four experiments with DH (5), DH (10), TE (4, 5), and TE (4, 10) in order to observe the effects of DH algorithm and TE algorithm on message delivery probability, overhead and average latency in the



(a) Comparison of delivery probability.



(b) Comparison of average latency.



(c) Comparison of overhead.

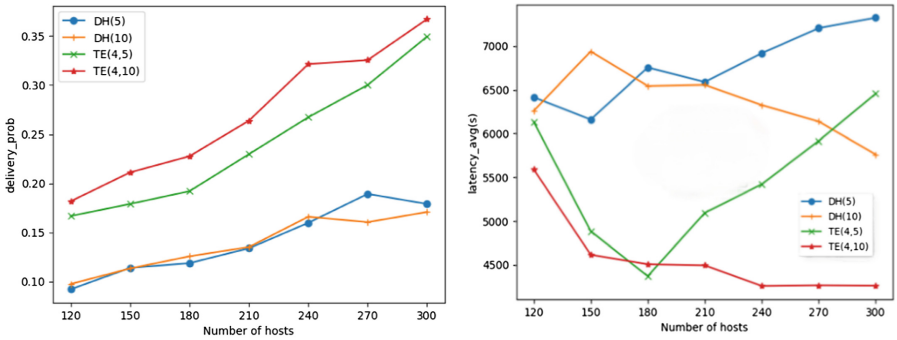
Fig. 4. Performance comparison before and after adding the privacy protection routing scheme.

opportunistic network. For the simulation of the key agreement algorithm based on threshold cryptography (TE algorithm), TE (4, 5) indicates that at least 4 of 5 packets transmitted successfully can recover the key, and so does TE (4, 10). DH (5) represents that 1 of 5 packets transmitted successfully can restore the key, and so does DH (10). Considering that message *M1* need to be transmitted, firstly it is determined whether *M1* is a shared key or not. If so, *M1* is divided into 10 fragments according to TE algorithm, which is presented as *M1* = {*M1S0*, *M1S1*, *M1S2*, *M1S3*, *M1S4*, *M1S5*, *M1S6*, *M1S7*, *M1S8*, *M1S9*}. During transmission, if the encounter node is the destination node, *M1* will be directly transmitted to the destination node. Otherwise, it determines whether the message fragment has been sent to the encounter. If not, the fragment will be sent to the encounter.

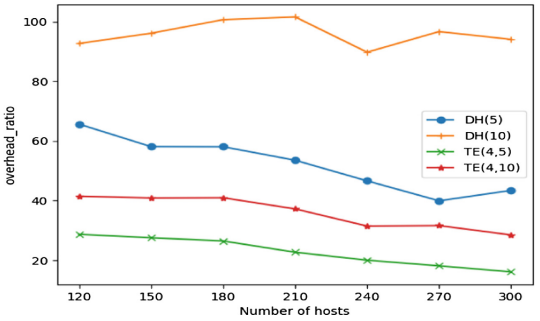
Result Analysis. Figure 5(a) shows that as the number of nodes increases, the success rate of transmission shows the upward trend. And it can be seen that in the cases of TE (4, 5) and TE (4, 10), the TE algorithm is significantly better than the DH algorithm.

Figure 5(b) shows that with the increase of the number of nodes, for DH (5) and TE (4, 5), the transmission delay shows an overall increasing trend. Because if the number of nodes in the network increases with a small number of sprays, the probability of forwarding to the effective node is reduced, resulting in the increase of transmission delay. And if increasing the number of sprays, such as DH (10) and TE (4, 10), the

transmission delay shows an overall decreasing trend. In addition, we can get that the average latency of TE algorithm is lower than the traditional DH scheme due to the smaller communication overhead.



(a) Comparison of delivery probability. (b) Comparison of average latency.



(c) Comparison of overhead.

Fig. 5. Performance comparison of the TE algorithm and the DH algorithm.

Figure 5(c) shows that as the number of nodes increases, the overhead of the system changes. It can be seen that the greater the number of the sprays, the bigger the system overhead. And because the DH scheme needs to response, the overhead of DH scheme is bigger than the DH scheme. In the case that the TE algorithm transmits 5 session key messages, the DH algorithm needs to transmits 10 session key messages for a round-trip. Therefore, the TE algorithm is obviously better than the DH algorithm.

According to the experiments, using privacy protection routing scheme improves the security of node privacy without significantly reducing the success rate of transmission. The average delay is not significantly changed, but the overhead of the system has increased. From the simulation experiment performed for the key agreement scheme, it can be seen that this scheme has obvious advantages in performance except for the loss of forward security.

6 Conclusion

The routing in opportunistic networks needs to compare the utility values of the source node and the intermediate nodes reaching to the destination node, which will reveal the privacy of the nodes. Therefore, this paper designs a lightweight privacy protection routing scheme based on the Bloom filter to obfuscate the node friends list and utility values introducing uncertainty to protect node privacy.

In addition, the lack of infrastructure and stable link in the opportunistic network leads to that there is no trusted third party to verify node's identity and increases the communication cost when performing key agreement using the DH algorithm. Therefore, we propose the identity authentication scheme based on zero-knowledge proof to verify certificates without a third party. What is more, we present a key agreement algorithm based on threshold cryptography, which only needs one-way communication to negotiate the session key. In general, the schemes proposed in this paper are more suitable for opportunistic network than other traditional schemes, and they can significantly improve the performance of the networks.

Acknowledgements. The work is supported by the Science and Technology Fundament Research Fund of Shenzhen under grant JCYJ20170307151807788, JCYJ20160318095218091.

References

1. Shikfa, A.: Security issues in opportunistic networks. In: International Workshop on Mobile Opportunistic Networking, pp. 215–216 (2010)
2. Ciobanu, R.I., et al.: Sprint-self: social-based routing and selfish node detection in opportunistic networks. Mob. Inf. Syst. **15**(6), 1–12 (2015)
3. Li, L., Qin, Y., Zhong, X., et al.: An incentive aware routing for selfish opportunistic networks: a game theoretic approach. In: International Conference on Wireless Communications & Signal Processing, pp. 1–5 (2016)
4. Nguyen, H.A., Giordano, S., Puiatti, A.: Probabilistic routing protocol for intermittently connected mobile ad hoc network (propicman). In: World of Wireless, Mobile and Multimedia Networks, pp. 1–6 (2007)
5. Daly, E.M., et al.: Social network analysis for information flow in disconnected delay-tolerant manets. IEEE Trans. Mob. Comput. **8**(5), 606–621 (2009)
6. Hui, P., Crowcroft, J., Yoneki, E.: Bubble rap: social-based forwarding in delay-tolerant networks. IEEE Trans. Mob. Comput. **10**, 1576–1589 (2008)
7. Juang, P., Oki, H., Yong, W., et al.: Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with zebranet. In: International Conference on Architectural Support for Programming Languages and Operating Systems, pp. 96–107 (2002)
8. Lindgren, A., et al.: Probabilistic routing in intermittently connected networks. ACM Sigmobile Mob. Comput. Commun. Rev. **7**(3), 19–20 (2004)
9. Boldrini, C., Conti, M., Jacopini, J., et al.: Hibop: a history based routing protocol for opportunistic networks. In: World of Wireless, Mobile and Multimedia Networks, pp. 1–12 (2007)
10. Pan, H., et al.: Bubble rap: social-based forwarding in delay-tolerant networks. In: IEEE Educational Activities Department, pp. 1576–1589 (2011)

11. Cadger, F., et al.: A survey of geographical routing in wireless ad-hoc networks. IEEE Commun. Surv. Tutorials **15**(2), 621–653 (2013)
12. Zhi, Z., Choong, Y.K.: Anonymizing geographic ad hoc routing for preserving location privacy. In: IEEE International Conference on Distributed Computing systems Workshops, pp. 646–651 (2005)
13. Zhou, J., et al.: Secure and privacy preserving protocol for cloud-based vehicular DTNs. IEEE Trans. Inf. Forensics Secur. **10**(6), 1299–1314 (2017)
14. Asghar, M.R., Gehani, A., Crispo, B., et al.: Pidgin: privacy-preserving interest and content sharing in opportunistic networks. In: ACM Symposium on Information, Computer and Communications Security, pp. 135–146 (2014)
15. Yao, L., et al.: Secure routing based on social similarity in opportunistic networks. IEEE Trans. Wirel. Commun. **15**(1), 594–605 (2016)
16. Boldrini, C., et al.: Exploiting users' social relations to forward data in opportunistic networks: the hibop solution. Pervasive Mob. Comput. **4**(5), 633–657 (2008)
17. Parris, I., Henderson, T.: Privacy-enhanced social-network routing. Comput. Commun. **35** (1), 62–74 (2012)
18. Shamir, A.: Identity-based cryptosystems and signature schemes. Lect. Notes Comput. Sci. **21**(2), 47–53 (1984)
19. Dan, B., Franklin, M.: Identity-based encryption from the weil pairing. SIAM J. Comput. **32** (3), 213–229 (2001)
20. Seth, A., Keshav, S.: Practical security for disconnected nodes. In: Secure Network Protocols (2005)
21. Kumar, K.P., Shailaja, G., et al.: Secure and efficient threshold key issuing protocol for ID-based cryptosystems. IACR Cryptology ePrint Archive 2006/245 (2006)
22. Al-Riyami, S.S., Paterson, K.G.: Certificateless public key cryptography. In: Laih, C.-S. (ed.) ASIACRYPT 2003. LNCS, vol. 2894, pp. 452–473. Springer, Heidelberg (2003). https://doi.org/10.1007/978-3-540-40061-5_29
23. Liu, W., et al.: Securing mobile ad hoc networks with certificateless public keys. IEEE Trans. Dependable Secur. Comput. **3**(4), 386–399 (2006)
24. Capkun, S., et al.: Self-organized public-key management for mobile ad hoc networks. IEEE Trans. Mob. Comput. **2**(1), 52–64 (2003)
25. Weniger, K., Zitterbart, M.: IPv6 autoconfiguration in large scale mobile ad-hoc networks. In: Proceedings of European Wireless (2002)
26. Yi, S., Kravets, R.: Composite key management for ad hoc networks. In: International Conference on Mobile and Ubiquitous Systems: Networking and Services, pp. 52–61 (2004)
27. Ngai, E.C.H., Lyu, M.R.: Trust and clustering-based authentication services in mobile ad hoc networks. In: International Conference on Distributed Computing Systems Workshops, pp. 582–587 (2004)
28. Chang, C.P., Lin, J.C., Lai, F.: Trust-group-based authentication services for mobile ad hoc networks. In: International Symposium on Wireless Pervasive Computing, pp. 16–18 (2006)