Noname manuscript No.
(will be inserted by the editor)

Secure Data Sharing Framework via Hierarchical Greedy Embedding in Darknets

Yanbin Sun · Mohan Li · Shen Su · Zhihong Tian(🖂) · Wei Shi · Meng Han

Received: date / Accepted: date

Abstract Geometric routing, which combines greedy embedding and greedy forwarding, is a promising approach for efficient data sharing in darknets. However, the security of data sharing using geometric routing in darknets is still an issue that has not been fully studied. In this paper, we propose a Secure Data Sharing framework (SeDS) for future darknets via hierarchical greedy embedding. SeDS adopts a hierarchical topology and uses a set of secure nodes to protect the whole topology. To support geometric routing in the hierarchical topology, a two-level bit-string prefix embedding approach (Prefix-T) is first proposed, and then a greedy forwarding strategy and a data mapping approach are combined with Prefix-T for data sharing. SeDS guarantees that the publication or request of a data item can always pass through the corresponding secure node, such that security strategies can be performed. The experimental results show that SeDS provides scalable and efficient end-to-end communication and data sharing.

Keywords Darknets · Secure framework · Geometric routing · Greedy embedding · Data sharing

1 Introduction

Darknets [1] exploit the infrastructure of the Internet but are not limited by the Internet rules. They do not use standard protocols and ports, and support anonymous communication and data sharing via the encryption technology and the Peer-to-Peer technology. Due to these characteristics, the censorship of darknets is difficult, and darknets may cause some potential criminal threats. However, as Steve Mansfield-Devine said, ”Darknets are just a technology, it's what you do with them that counts” [2]. Using darknets under current laws can also bring lots of positive effects and benefits. For instance, darknets can provide much stronger anonymous communication and privacy protection. Currently, darknets have become one of the hotspots in the cyber security research.

Darknets always adopt overlay communication systems for data sharing. Some typical applications of darknet, such as Freenet, oneSwarm and GNUnet, are all based on overlays. Different from existing anonymous communication systems like Tor [3], the topology of darknet is not derived from the infrastructure topology of Internet, but depends on the trust relationship or other social relationships between two users. For example, in friend-to-friend networks [4], the data is transported from a node to its trusted friend node regardless the links between them on the internet topology. In this paper, we focus on the darknet topology rather than the Internet topology.

The darknet is treated as ”non-public Internet”, it is as important as the Internet and needs prospective studies. Though the darknet has developed for many years, it still faces two challenges. (1) Efficent routing and content distribution scheme for data sharing. Various routing and content distribution schemes are

🖂E-mail: tianzhihong@gzhu.edu.cn
Yanbin Sun · Mohan Li · Shen Su · Zhihong Tian
Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou 510006, China.
Wei Shi
School of Information Technology Faculty of Engineering and Design, Carleton University Ottawa, Ontario, 520-2600, Canada.
Meng Han
Kennesaw State University, 1100 South Marietta Pkwy Marietta, GA 30060

proposed, such as flooding approach [5], probabilistic forwarding [6] and virtual DHT design [7]. However, these schemes cause either high overhead or inefficient routing. (2) Secure framework. Like traditional P2P networks, data items in darknets are also stored or resolved by users in darknets. Due to the lack of verification mechanism, a large number of illegal or harmful content may be stored or spread by the normal user inadvertently. Darknets also suffer from cyber attacks, such as DDos attack, Sybil attack, etc. Obviously, security becomes more and more important in the future research on darknets. To solve above problems, a secure framework providing efficient data sharing is indeed needed for current and future darknets.

Geometric routing [8] is a promising approach to provide efficient and scalable data sharing, and it is originally used for wireless networks [9, 10]. Different from the routing of the Internet [11], geometric routing greedily embeds the topology into a metric space, i.e., assigning each node a coordinate in the metric space. Each node only stores the coordinates of its neighbors, and greedily forwards the message hop-by-hop according to the coordinate distance. Geometric routing has following advantages for data sharing in darknets: First, geometric routing maintains a low path stretch (the ratio of the routing path length to the shortest path length) by using a small amount of routing information. Each node only stores the coordinate of its neighbors. Second, the embedded topology is potential to construct a darknet topology based DHT. The data item is mapped to a node which is responsible to store the data item, and the request can be greedily forwarded to a node hop-by-hop on the darknet topology. Though some schemes [12, 4] are proposed for data sharing using geometric routing in darknets, there are few security mechanisms to protect normal users.

In this paper, we proposed a Secure Data Sharing framework (SeDS) via hierarchical greedy embedding. SeDS divides the nodes in darknets into two categories: secure nodes and normal nodes. Each secure node is responsible for the security of a set of normal nodes. To support geometric routing on the hierarchical topology, a two-level bit-string prefix embedding approach (Prefix-T) is proposed. By combining Prefix-T with a greedy forwarding strategy and a data mapping approach, data sharing is implemented. SeDS supports efficient communication for arbitrary two nodes, and guarantees that the data item which is published to a node can always pass through the corresponding secure node, such that security strategies can be performed. SeDS is not only used for darknets, but also be used for other data sharing systems.

The organization of the rest of this paper is as follows. Section 2 discusses the related work, Section 3 presents the detailed design of SeDS. Section 4 evaluates the performance of SeDS. Section 5 concludes the paper.

2 Related work

Data sharing provides efficient content distribution and retrieval based on overlay networks. It can be used in various scenarios, such as personal data sharing, the knowledge transport of enterprises [13], IPVT [14]. To realize anonymous, censorship-resistant data sharing, darknets [12], which restrict overlay links to trusted parties, become promising approaches.

Various approaches are used for data sharing in darknets, most of them face scalability or efficiency issues. Turtle [5] builds a data sharing overlay on top of preexisting trust relationships. The query of a data item is broadcast on the overlay and the data item is returned via the reverse path. OneSwarm [6] supports a mix of trusted and untrusted peers rather than the only trusted peers, and it combines broadcast and probabilistic forwarding for content retrieval. Both of them are suitable for the small size of network. R^5N[15] combines a random walk with recursive Kademlia-style routing for data retrieval, but the high success rate demands a large number of replicas. X-Vine [7] constructs a virtual DHT overlay similar to Chord and adopts VRR-like protocol [16] for routing. However, it does not avoid the shortcomings of VRR, i.e., long routing paths and lots of routing information.

To solve above problems, geometric routing may be a promising approach for efficient and scalable routing. In geometric routing, the topology is embedded into a metric space, i.e., each node is assigned a coordinate in the metric space. Each node only stores the coordinates of its neighbors, and the packet is greedily forwarded according to the coordinate distance. The data can also be assigned a coordinate, and is registered or retrieved via greedy forwarding according to the coordinate.

In the early days of the geometric routing scheme, physical coordinates or virtual coordinates are used for greedy routing [17]. The physical coordinate and the virtual coordinate can also be used for data collection, which is similar to data sharing. Some approaches based on DHTs are proposed, such as [18]. Compared with traditional DHT designs, the data is mapped to a coordinate, and is greedily routed to corresponding node according to the coordinate. These approaches can reduce the stretch and routing information by embedding the network topology into a geographic or virtual coordinate space. However, the routing may fail because the embedding of topology does not satisfy the greediness.

To guarantee the greediness of embedding, some geometric routing schemes based on greedy embedding have been developed. Geometric spaces [19] are studied for embedding. Kleinberg [8] proposed a universal greedy embedding approach for arbitrary graphs via the greedy embedding of a spanning tree. MobiCCN [20] used the greedy embedding scheme of [8] for content routing. However, the coordinate needs $O(n)$ bits, which may cause high overhead for networks.

To obtain succinct coordinates, some succinct greedy embedding approaches are proposed. Herzen and Westphal [21] isometrically embedded a spanning tree into $l_\infty^{O(\log n)}$. It provides $O(\log^3(n))$-bit coordinates in power law graphs with $2<\lambda<3$. Based on this work, Hfer et al. [12] proposed a prefix embedding approach and a virtual tree embedding approach for content addressing in darknets, but the greediness and the load balance cannot be guaranteed simultaneously. Then, they proposed a privacy-preserving routing scheme in darknets with multiple embeddings [4]. This work focuses on anonymous problem and the multiple embeddings may bring long coordinates.

From the above, the scheme using geometric routing for data sharing is a promising approach in darknets, but it still has some problems to be solved. For one thing, geometric routing should satisfy greediness, scalability and efficiency. For the other thing, the security of data sharing should be guaranteed, i.e, the data cannot be illegal or harmful. Due to the censorship-resistant property, most schemes do not consider the security requirement.

Our framework also focuses on darknets, but not limited to darknets. Different from previous data sharing schemes using geometric routing [12] [4], SeDS guarantees both the requirements of geometric routing and the security of data sharing using a geometric routing scheme with a two-level prefix embedding.

3 Desgin of SeDS

The idea of SeDS is simple. In geometric routing, we only need to find a set of nodes responsible for security and ensure that the publication or request of data item passes through these nodes. Meanwhile, the efficiency of geometric routing should not be affected.

SeDS consists of two parts: a geometric routing scheme and a mapping approach. The former combines greedy embedding and greedy forwarding, and it provides a secure framework and routing services. The later maps a data item to a coordinate. According to the coordinate, the node, which is responsible for the data item, can be found via geometric routing.

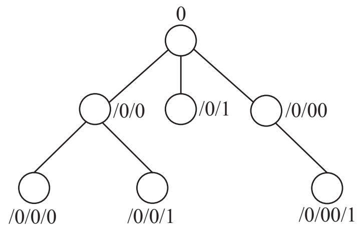

Fig. 1 Embedded tree via bit-string prefix embedding

In this section, we first review a bit-string prefix embedding scheme (Prefix-B) [22] which we have proposed. Based on this scheme, a two-level bit-string prefix embedding scheme (Prefix-T) is proposed. Then, we discuss the greedy forwarding strategy and the mapping approach.

3.1 Review of Bit-string Prefix Embedding

Prefix-B adopts a bit-string prefix tree metric space. In the metric space, each node is assigned a label (bit-string) which is different from the labels of its siblings. The coordinate of a node is obtained by concatenating labels on the path of the prefix tree from the root node to the node. Note that, the bit-string prefix tree is predetermined. It is only determined by the number of labels. If the number of label is not limited, the bit-string prefix tree is an infinite tree.

For a connected graph $G(V,E)$, the greedy embedding of G is divided into two steps: (1) Extract a spanning tree T from G. (2) Embed T into a bit-string prefix tree metric space X, i.e., assign each node of T a coordinate. The first step adopts the SPT protocol [23] to construct and maintain a spanning tree. Since the metric space is an infinite tree, the greedy embedding of a spanning tree into the bit-string prefix tree metric space turns to construct a sub-tree of the bit-string prefix tree based on the spanning tree. Thus, the greedy embedding is obtained by a top-down traversal of the spanning tree T from the root node to leaves.

The embedding process is as follows. The root node is first assigned an initial coordinate 0. For an embedded non-leaf node u, the coordinate of u is expressed as $C_u{=}/w_u^0b_u^0/w_u^1b_u^1/.../w_u^kb_u^k$, where b_u^i is a bit-string and w_u^i is an edge weight. After u obtained its coordinate, it assigns different labels to its children. If a child v of u is assigned a label b and the edge weight between u and v is w. The coordinate of a child is obtained by appending the label to the coordinate of its parent, i.e., $C_v{=}/w_u^0b_u^0/w_u^1b_u^1/.../w_u^kb_u^k/wb$. Fig. 1 shows the example of an embedded spanning tree.

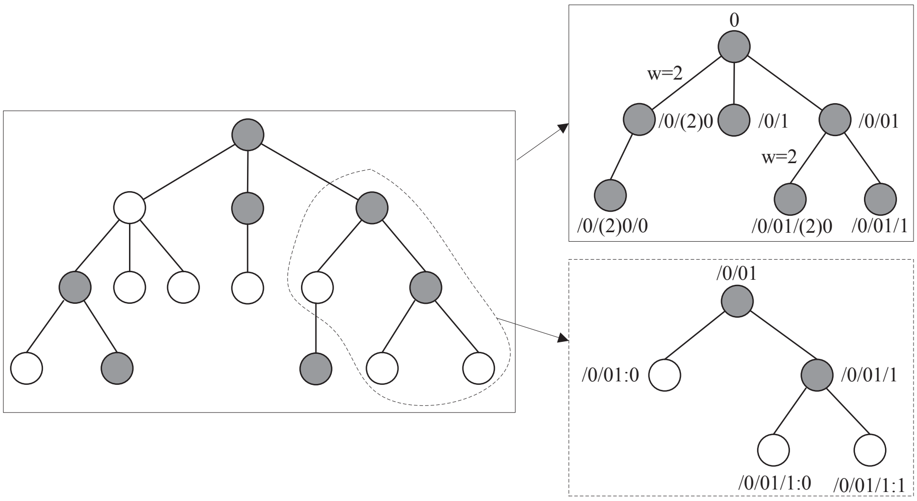

Fig. 2 Two-level bit-string prefix embedding

3.2 Two-level Bit-string Prefix Embedding

In SeDS, the topology is divided into two levels according to two types of nodes: secure nodes and normal nodes. Secure nodes construct the outer layer topology, and normal nodes construct the inner layer topology. The two types of nodes both participate in global communication and store data items. Differently, the secure node is responsible for more tasks, such as data forwarding, security authority, and it provides services for normal nodes. The secure node should be trusted in darknets and can adopt different security strategies, e.g. the attack detection [24, 25], trust models [26], the key management [27, 28, 29], the digital forensics [30]. The secure node can be selected randomly or according to the node capacity (computation, bandwidth, storage), authority, or other properties.

3.2.1 Greedy Embedding

Prefix-T adopts two-level embedding based on Prefix-B. Greedy embedding of a topology via Prefix-T is also based on a spanning tree. First, we assume that the spanning tree T of a graph $G(V, E)$ has been extracted, and guarantee that the root node is always a secure node. Then, the greedy embedding is obtained by a top-down traversal of the spanning tree. Different type of nodes is assigned a coordinate via different strategies.

For better understanding, the two-level bit-string embedding process is presented from two aspects: the greedy embedding of secure nodes and the greedy embedding of normal nodes. As shown in Fig.2, the tree on the left side is the spanning tree of a graph with two types of nodes. The gray node is the secure node and the white node is the normal node.

All secure nodes construct a virtual tree (see upper right corner of Fig.2). Particularly, the virtual tree is a weighted graph. The edge weight between two secure nodes in the virtual tree is a sum of edge weights between the two nodes on the spanning tree. The embedding of the secure node turns to the greedy embedding Prefix-B of the virtual tree.

Normal nodes and a secure node construct a subtree of the spanning tree. Each subtree contains only one secure node, and the tree is rooted at the node. The normal node is the descendant of the secure node. At the lower right corner of Fig.2, the two secure nodes construct two subtrees. Thus, the embedding of the normal node is obtained by the greedy embedding Prefix-B of the subtree.

For any node u, the universal coordinate consists of two levels: C_u=C_{u1}:C_{u2}. Each layer is a hierarchical bit-string, e.g. C_{u1}=$/w^0_{u1}b^0_{u1}/w^1_{u1}b^1_{u1}/.../w^k_{u1}b^k_{u1}$, C_{u2} =$/w^0_{u2}b^0_{u2}/w^1_{u2}b^1_{u2}/.../w^k_{u2}b^l_{u2}$. Specially, the second layer of the secure node coordinate is null. The coordinate is stored in the form of a bit-string, and the hierarchical structure is divided according to two masks similar to [31]. For example, the mask of the coordinate /0/0/110 is 11001. Each '1' in the mask represents an end of a bit-string. Thus, 11001 is divided into three parts: 1, 1 and 001, each part corresponds to a bit-string of the coordinate. For a coordinate with weight, the weight is also represented by the mask. For example, the coordinate /0/(3)0/110 is first changed to /(1)0/(11)0/(1)110 (binary form), then the mask 110111001 can be obtained. For the two-level coordinate C_{u1}:C_{u2}, each layer is represented via a mask.

Prefix-T of a spanning tree is implemented as follows. The root node coordinate is first initialized to /0. Each embedded node checks the node type of its child.

For each type children, the parent adopts the reusable coding strategy [22] to produce a bit-string for each child. The bit-string combined with the parent coordinate and the edge weight are sent to the child. When an unembedded node u receives the message from its parent t, the coordinate is obtained according to different conditions (Algorithm.1).

(1) If u is a secure node, it obtains its coordinate by appending the bit-string b and the edge weight w to the first layer of parent coordinate C_t=C_{t1}:C_{t2}. The coordinate C_u is expressed as follows:

$$C_u = C_{t1}/w_{u1}b : null \tag{1}$$

$$w_{u1} = \sum_{i=0}^{|C_{t2}|} w_{t2}^{i} + w \tag{2}$$

where $|C_{t2}|$ denotes the layer number of C_{t2}.

(2) If u is a normal node, it obtains its coordinate by appending the bit-string b and the edge weight w to the second layer of parent coordinate C_t=C_{t1}:C_{t2}. The coordinate C_u is expressed as follows:

$$C_u = C_{t1} : C_{t2}/wb \tag{3}$$

When the children get their coordinates, they repeat above process. After a top-down traversal of the spanning tree T from the root node to leaves, each node is assigned a node coordinate and the spanning tree is greedily embedded. Fig.3 shows an example of an embedded spanning tree.

Algorithm 1 *Greedy Embedding (Prefix-T)* (at node u)

```
$u.child\_num$ is the children number of $u$;
$C_u$ is the coordiante of $u$;
$u.type$ is the node type of $u$;
Case 1: $u$ is an unembedded node, and it receive a message (bit-string $b$, edge weight $w$ and coordinate $C_t$) from its parent $t$.
if $u.type$ = secure then
    $u$ obtains its coordinate according to Eq.(1) and Eq.(2);
else
    $u$ obtains its coordinate according to Eq.(3);
end if
Case 2: after $u$ is embedded, assume that $v$ is a child of $u$.
$u$ compute a bit-string $b'$ for $v$;
$u$ sends the bit-string $b'$, a edge weight $w'$ and $C_u$ to $v$;
```

Algorithm 1 shows the distributed implementation of Prefix-T, it combines the process of spanning tree extraction and coordinate assignment. After the spanning tree is obtained, the embedding can also be finished. Thus, the communication complex of Prefix-T is the same with that of the spanning tree protocol.

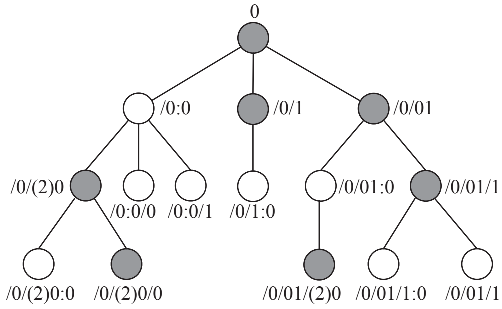

Fig. 3 Embedded tree via two-level bit-string prefix embedding

3.2.2 Coordinate Distance

Since the coordinate of Prefix-T contains two layers, for better understanding, we first discuss the distance of one layer coordiante, and then put the two layers together to compute the universal coordinate distance.

One layer coordinate distance is similar to Prefix-B which depends on the longest prefix matching. The longest prefix between two coordinates is obtained by comparing the bit-strings of the two coordinates regardless weight. For any two coordinates C_{u1}= $/w_{u1}^{0}b_{u1}^{0}/w_{u1}^{1}b_{u1}^{1}/\cdots/w_{u1}^{h}b_{u1}^{h}$ and c_{v1}= $/w_{v1}^{0}b_{v1}^{0}/w_{v1}^{1}b_{v1}^{1}/\cdots/w_{v1}^{k}b_{v1}^{k}$, assuming that the length of the longest common prefix is l, then the distance between them is the sum of the weights of two coordinates except the weights of the longest common prefix. The distance function is expressed as follows:

$$d'(C_{u1}, C_{v1}) = \sum_{i=l}^{h} w_{u1}^{i} + \sum_{j=l}^{k} w_{v1}^{j} \tag{4}$$

The coordinate distance of Prefix-T is obtained based on the virtual tree and the local subtree. As shown in Eq.(5), these four formulas correspond to two cases according to whether u and v are in the same local subtree. (1) If the two nodes belong to different subtrees, the coordinate distance is calculated based on the longest prefix of the first layer with the following cases (the first three conditions in the equation): (a) u and v has the same nearest common ancestor x, where $x \neq u$ and $x \neq v$. (b) u is the ancestor of v. (c) v is the ancestor of u. (2) If the two nodes are in the same subtree, the distance is calculated according to the second layer of the coordinate (the fourth condition).

According to prefix embedding, the embedding of virtual tree is isometric according to the first layer coordinates of any two nodes, but isometric embedding of spanning tree cannot be guaranteed Since the first layer coordinate only contains the topology information of virtual tree and the second layer reveals local topology information, the node coordinate cannot reflect the

$$d(C_u,C_v)=\begin{cases} d'(C_{u1},C_{v1})+d'(C_{u2},null)+d'(null,C_{v2}) & if\ l\neq h\bigwedge l\neq k,\\ d'(C_{u1},C_{v1})-d'(C_{u2},null)+d'(null,C_{v2}) & if\ l=h\bigwedge l\neq k,\\ d'(C_{u1},C_{v1})+d'(C_{u2},null)-d'(null,C_{v2}) & if\ l\neq h\bigwedge l=k.\\ d'(C_{u2},C_{v2}) & if\ l=h\bigwedge l=k.\end{cases} \quad (5)$$

complete spanning tree. For any two nodes, the coordinate distance may not equal to the node distance on the spanning tree. Thus, Prefix-T is not isometric. Even so, we can still prove that Prefix-T is greedy.

Theorem 1 *For any connect graph G, the embedding of G produced by Prefix-T is greedy.*

Proof According to the definition of greedy embedding [32], to prove the greediness of Prefix-T, we should guarantee that for any two nonadjacent node u and v, there must exist a neighbor w of u such that $d'(u,v) > d'(w,v)$. We analyze the greediness from two cases according to Eq.5

(1) If u and v are in different subtrees, there are three cases.

(a) If u and v has the same nearest common ancestor x ($x \neq u$ and $x \neq t$), then

$$d'(u,v)=d'(C_{u1},C_{v1})+d'(C_{u2},null)+d'(null,C_{v2})$$
$$=d'(r_u,r_v)+d'(u,r_u)+d'(r_v,v)$$

where r_u is the root node of the subtree that u belongs to. Accordingly, the coordinate distance of w and v is expresed as follows,

$$d'(w,v)=d'(r_u,r_v)+d'(w,r_u)+d'(r_v,v)$$

If w is the parent of u in the spanning tree, $d'(w,r_u) < d'(u,r_u)$. Thus, $d'(u,v) > d'(w,v)$, the neighbor can be otained.

(b) If u is the ancestor of v, then

$$d'(u,v)=d'(r_u,r_v)-d'(u,r_u)+d'(r_v,v)$$

Accordingly, the coordinate distance of w and v is expresed as follows,

$$d'(w,v)=d'(r_u,r_v)-d'(w,r_u)+d'(r_v,v)$$

If w is the child of u in the spanning tree, $d'(w,r_u) > d'(u,r_u)$. Thus, $d'(u,v) > d'(w,v)$, the neighbor can be otained.

(c) If v is the ancestor of u, then the analysis is similar to (a).

(2) If u and v are in the same subtree, Prefix-T is a prefix embedding. Obviously, the greediness can be guaranteed.

Above all, the greediness of Prefix-T can be guaranteed.

3.3 Greedy Forwarding, Data Mapping and Sharing

3.3.1 Greedy Forwarding

Greedy forwarding uses local routing information. Each node only stores the coordinates of its neighbors. The greedy forwarding process is straightforward: when a message reaches a node u, u selects a neighbor which is the nearest to the destination according to the coordinate distance, and then forwards the message to the neighbor. After the message reaches a new node, the process is repeated until the packet reaches the destination. The greedy embedding of the topology guarantees that the next hop can always be found if the destination is a node on the topology. Thus, the ene-to-end communication in darknets can be supported.

3.3.2 Data Mapping and Sharing

For data sharing, the data item is published or copied to a node on the topology. To determine which node a data item corresponds to, the coordinate of data item is needed. The mapping approach is used to map the name of data item to a coordinate. The mapping approach may affect the distribution of data items. Existing mapping approaches, such as [12, 33], map the content name to a coordinate in the metric space rely on the topology information. Since our focus is on the secure framework, we adopt an existing approach [33] for mapping.

To make sure that the publication or request of data item passes through the secure node such that security strategies can be performed, the name of a data item is mapped to a one-layer coordinate without weight according to the virtual tree. When a node gets a data item coordinate, the publication or request packet using the coordinate as the destination is greedily forwarded to a secure node. The next hop of greedy forwarding is chosen according to the length of the longest common prefix between the data item coordinate and the first layer of node coordinate regardless the weight. According to the greedy forwarding, for any non-destination node, there must be a parent node or a child node which is nearer to the destination than current node on the virtual tree. Thus, it is guaranteed that there must be a greedy path to the secure node regardless where the data item is published or requested.

After a publication packet reaches a secure node, the secure node checks the sender or the packet according to some secure rules. The trusted publication is sent to one of the normal nodes on the subtree rooted at the secure node. The normal node is chosen according to a random strategy or others. After the normal node is determined, the normal node and the sender can communicate with each other through their node coordinates. Finally, the data item is registered or copied to a normal node. The process of data request is similar. For any node, it can find corresponding secure node via greedy forwarding according to the data item coordinate, then, the normal node is obtained.

4 Discussion

In this section, we discuss the properties of SeDS by quantitative or qualitative analysis.

(1) Scalability. SeDS inherits the scalability of geometric routing. For SeDS, each node only need to store routing entries to its neighbors, i.e., the coordinates of neighbors. Meanwhile, the coordinate is also succinct. Similar to the analysis of prefix embedding [12, 22], in power law topologies with $2 < \lambda < 3$, the coordinate length is $O(\log^2 n)$ bits, where n is the size of topology.

(2) Greediness and Efficiency. SeDS provides 100% routing success rates and efficient routing. According to Theorem 1, the greediness of Prefix-T can be proved. Thus, for any two nodes, a greedy path between them can always be found. Though the upper bound of path stretch cannot be guaranteed, our experiments show that routing of SeDS is also efficient.

(3) Security. SeDS provides an adjustable secure framework in darknets. The number of secure nodes can be adjusted according to different security requirements and node capabilities. In extreme cases, each node can be a secure node and provide service for itself. Correspondingly, each node can also be a normal node and does not consider the security requirement. Since the publication or request of a data item is based on the virtual tree, corresponding packets can finally reach a secure node. Thus, the publication or request of a data item can always be checked by the secure node.

Compared with previous data sharing schemes using geometric routing, Prefix-T of SeDS is greedy but not isometric. The embedding of previous schemes is greedy and isometric. The routing of previous schemes may be more efficient than that of SeDS, but SeDS can provide more flexible and secure framework than previous schemes.

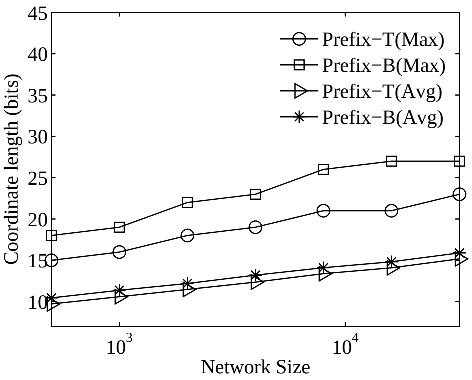

Fig. 4 Max coordinate and average coordinate

5 Evaluation

SeDS is evaluated from three aspects: the coordinate length, the path stretch of geometric routing and the path length of data sharing. The purpose of the evaluation is to verify that our secure framework works well for data sharing and end-to-end communication. There are two types of topologies: the real-world topologies from CAIDA [34] and the synthetic topologies generated by GLP model [35]. Each topology is transformed to an unweighted and undirected graph.

5.1 Coordinate Length

The coordinate length is an important factor of scalability. It affects both the size of routing table and the payload of packet header. We measured the maximum and average coordinate lengths on different size of topologies (from 500 to 30000). Fig.4 shows the two types of coordinate length when the network grows. Both the maximum coordinate length and the average coordinate length of Prefix-T are better than those of Prefix-B. The main reason is that the coordinate of Prefix-T stores less topology information than that of Prefix-B, which also makes Prefix-T more secure. For Prefix-T, the coordinate length has a linear relationship with $O(\log(n))$, which is better than the theoretical result. In different size of networks, coordinate maintains at a low value. Even in the network with 30000 nodes, the maximum coordinate length is no more than 25. Obviously, our embedding scheme Prefix-T provides succinct coordinates.

5.2 Path Stretch

Path stretch is the ratio of the routing path length to the shortest path length. Here, we adopted the path

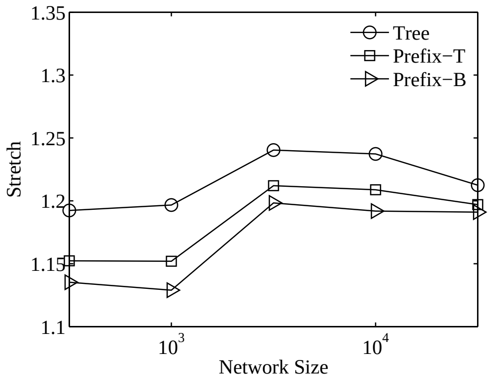

Fig. 5 Average path stretch

stretch to measure the efficiency of end-to-end communication. Our scheme is compared with Tree-based routing and Prefix-B based routing. For geometric routing schemes using a spanning tree for greedy embedding, the choice of spanning tree affects the path stretches. Thus, we kept the spanning tree for different schemes.

Fig.5 shows the average path stretch with the growth of network.The average stretch remains stable as the network grows. For these three schemes, the result of tree based routing is the worst. The stretch of Prefix-T is slightly higher than that of Prefix-B. The reason is that: though Prefix-T guarantees greedy embedding, the coordinate distance is distorted according to the spanning tree distance, i.e. the isometric property cannot be guaranteed. The isometric property results in the gap of path stretch between Prefix-T and Prefix-B. Though the greedy path is extended, it is an acceptable cost for secure framework.

5.3 Hop Count for Data Sharing

The process of data publication or request is different from that of end-to-end communication. The data item is first routed to a secured node, and then is sent to a normal node. Here, we used hops to measure the path length of data sharing.

We first produced 500 data items. For each data item, we randomly selected 100 nodes as source to publish the data item. The topology is obtained from a real-world network with over 10000 nodes. Fig.6 shows the CDF of hops for different percentage of secure nodes. It is obvious that the larger the percentage is the better the routing performance is. The main reason is still the distortion of embedded path. Satisfyingly, the worst routing path length is no more than 9 hops.

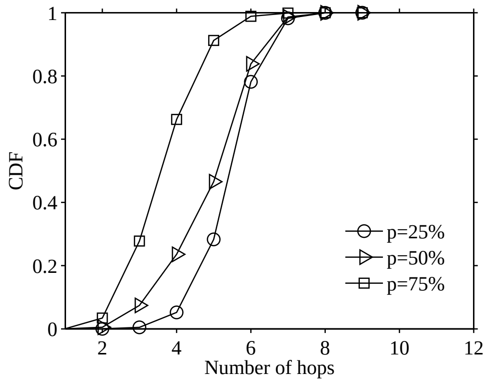

Fig. 6 CDF of path length with different percentage of secure nodes

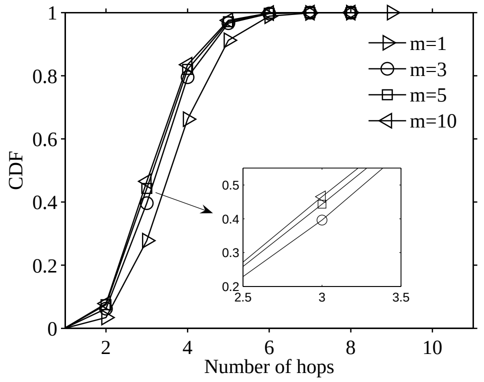

Fig. 7 CDF of path length with different number of replicas

To request a data item, the user can first map the data name to a coordinate, and then directly forwarded to the secure node of the data. The routing path is the same with that of data publication. To decrease the length of routing path, the data replica can be published. Fig.7 shows the path length for data with different number of replicas. m is the number of replicas. Obviously, the path length is worse than others when m=1. According to the larger version of Fig.7 , the length of routing path decreases with the increase of replica number, but the variation is getting smaller.

6 Conclusion and Future work

This paper focus on the secure data sharing using geometric routing in darknets and propose a secure framework SeDS in a hierarchical topology based on a two-level bit-string prefix embedding scheme. The publication or request of data item can always pass through the corresponding secure node, such that security strategies can be performed. SeDS provides efficient end-to-end communication and data sharing. SeDS is not limited to darknets, it can also be used for wireless networks or other data sharing systems. Since SeDS is only a secure framework without specific security scenarios and security strategies, how to use SeDS to solve the specific security problem can be our future work.

Acknowledgements This work is funded by the National Natural Science Foundation of China (No. 61702223, 61702220, 61871140, 61572153, U1636215) and the National Key Research and Development Plan (Grant No. 2018YFB0803504).

References

1. Biddle P, England P, Peinado M, Willman B (2002) The darknet and the future of content protection. In: ACM Workshop on Digital Rights Management, Springer, pp 155–176
2. Mansfield-Devine S (2009) Darknets. Computer Fraud & Security 2009(12):4–6
3. Tan Q, Gao Y, Shi J, Wang X, Fang B, Tian ZH (2018) Towards a comprehensive insight into the eclipse attacks of tor hidden services. IEEE Internet of Things Journal
4. Roos S, Beck M, Strufe T (2016) Anonymous addresses for efficient and resilient routing in f2f overlays. In: Proceedings of the 35th Annual IEEE International Conference on Computer Communications (INFOCOM), IEEE, pp 1–9
5. Popescu B (2004) Safe and private data sharing with turtle: friends team-up and beat the system (transcript of discussion). In: International Workshop on Security Protocols, Springer, pp 221–230
6. Isdal T, Piatek M, Krishnamurthy A, Anderson T (2010) Privacy-preserving p2p data sharing with oneswarm. In: ACM SIGCOMM Computer Communication Review, ACM, vol 40, pp 111–122
7. Mittal P, Caesar M, Borisov N (2011) X-vine: Secure and pseudonymous routing using social networks. arXiv preprint arXiv:11090971
8. Kleinberg R (2007) Geographic routing using hyperbolic space. In: Proceeding of the 26th IEEE International Conference on Computer Communications (INFOCOM), IEEE, pp 1902–1909
9. Du X, Chen HH (2008) Security in wireless sensor networks. Wireless Communications IEEE 15(4):60–66
10. Tian Z, Gao X, Su S, Qiu J, Du X, Guizani M (2019) Evaluating reputation management schemes of internet of vehicles based on evolutionary game theory. IEEE Transactions on Vehicular Technology
11. Tian Z, Su S, Shi W, Du X, Guizani M, Yu X (2019) A data-driven method for future internet route decision modeling. Future Generation Computer Systems 95:212–220
12. Höfer A, Roos S, Strufe T (2013) Greedy embedding, routing and content addressing for darknets. In: Conference on Networked Systems (NetSys), IEEE, pp 43–50
13. Wu C, Zapevalova E, Chen Y, Li F (2018) Time optimization of multiple knowledge transfers in the big data environment. Computers, Materials & Continua 54(3):269–285
14. Yang X, Du X, Zhang J, Fei H, Guizani S (2007) Internet protocol television (iptv): The killer application for the next-generation internet. IEEE Communications Magazine 45(11):126–134
15. Evans NS, Grothoff C (2011) R5n: Randomized recursive routing for restricted-route networks. In: 2011 5th International Conference on Network and System Security (NSS), IEEE, pp 316–321
16. Caesar M, Castro M, Nightingale EB, O'Shea G, Rowstron A (2006) Virtual ring routing: network routing inspired by dhts. In: ACM SIGCOMM Computer Communication Review, ACM, vol 36, pp 351–362
17. Karp B, Kung HT (2000) GPSR: Greedy perimeter stateless routing for wireless networks. In: Proceedings of the 6th annual international conference on Mobile computing and networking, ACM, pp 243–254
18. Ratnasamy S, Karp B, Yin L, Yu F, Estrin D, Govindan R, Shenker S (2002) GHT: a geographic hash table for data-centric storage. In: Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications, ACM, pp 78–87
19. Abdel-Aziz H, Saad MK (2018) On special curves according to darboux frame in the three dimensional lorentz space. Computers, Materials & Continua 54(3):229–249
20. Wang L, Waltari O, Kangasharju J (2013) Mobiccn: Mobility support with greedy routing in content-centric networks. In: IEEE Global Communications Conference (GLOBECOM), IEEE, pp 2069–2075
21. Herzen J, Westphal C, Thiran P (2011) Scalable routing easy as PIE: A practical isometric embedding protocol. In: Proceeding of the 19th IEEE International Conference on Network Protocols (ICNP), IEEE, pp 49–58
22. Sun Y, Zhang Y, Fang B, Zhang H (2017) Succinct and practical greedy embedding for geometric routing. Computer Communications
23. Perlman R (1985) An algorithm for distributed computation of a spanning tree in an extended LAN. In: ACM SIGCOMM Computer Communication Review, ACM, vol 15, pp 44–53
24. Jiang F, Fu Y, Gupta BB, Lou F, Rho S, Meng F, Tian Z (2018) Deep learning based multi-channel

intelligent attack detection for data security. IEEE Transactions on Sustainable Computing

25. Tian Z, Shi W, Wang Y, Zhu C, Du X, Su S, Sun Y, Guizani N (2019) Real time lateral movement detection based on evidence reasoning network for edge computing environment. IEEE Transactions on Industrial Informatics
26. Ahmed A, Bakar KA, Channa MI, Khan AW (2016) A secure routing protocol with trust and energy awareness for wireless sensor network. Mobile Networks and Applications 21(2):272–285
27. Du X, Guizani M, Xiao Y, Chen HH (2011) Transactions papers a routing-driven elliptic curve cryptography based key management scheme for heterogeneous sensor networks. IEEE Transactions on Wireless Communications 02(05):1223–1229
28. Rayi VK, Xiao Y, Sun B, Du XJ, Hu F (2007) A survey of key management schemes in wireless sensor networks. Computer Communications 30(11):2314–2341
29. Du X, Xiao Y, Guizani M, Chen HH (2007) An effective key management scheme for heterogeneous sensor networks. Ad Hoc Networks 5(1):24–34
30. Tian Z, Li M, Qiu M, Sun Y, Su S (2019) Block-def: A secure digital evidence framework using blockchain. Information Sciences pp 151–165
31. Thorup M, Zwick U (2001) Compact routing schemes. In: Proceedings of the 13th annual ACM symposium on Parallel algorithms and architectures, ACM, pp 1–10
32. Papadimitriou CH, Ratajczak D (2005) On a conjecture related to geometric routing. Theoretical Computer Science 344(1):3–14
33. Yanbin S, Yu Z, Shen S, Hongli Z, Binxing F (2015) Geometric name routing for icn in dynamic world. China Communications 12(7):47–59
34. (2012) The IPv4 Routed/24 AS Links Dataset-Jue, 2012. `http://www.caida.org/data/active/ipv4_routed_topology_aslinks_dataset.xml`
35. Bu T, Towsley D (2002) On distinguishing between internet power law topology generators. In: Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), IEEE, vol 2, pp 638–647