Fine-grained Encountering Information Collection under Neighbor Anonymity in Mobile Opportunistic Social Networks

Kang Chen
Department of Electrical and Computer Engineering
Southern Illinois University, Carbondale, IL 62901
Email: kchen@siu.edu

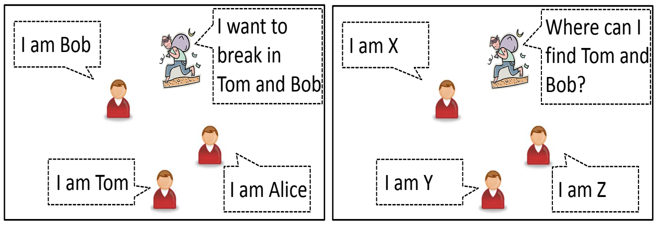
Haiying Shen
Department of Electrical and Computer Engineering
Clemson University, Clemson, SC 29631
Email: shenh@clemson.edu

***Abstract*—In mobile opportunistic social networks (MOSNs), mobile devices carried by people communicate with each other directly when they meet for proximity-based MOSN services (e.g., file sharing) without the support of infrastructures. In current methods, when nodes meet, they simply communicate with their real IDs, which leads to privacy and security concerns. Anonymizing real IDs among encountering neighbor nodes solves such concerns. However, this prevents nodes from collecting real ID based encountering information, which is needed to support MOSN services. Therefore, in this paper, we propose FaceChange that can support both anonymizing real IDs among neighbor nodes and collecting real ID based encountering information. To realize neighbor node anonymity, two encountering nodes communicate anonymously. Then, when the two nodes disconnect, each node forwards an encrypted encountering evidence to the encountered node to enable encountering information collection. A set of novel schemes are designed to protect the confidentiality and uniqueness of encountering evidences. FaceChange also supports fine-grained control over what encountering information should be forwarded based on attribute similarity (i.e., trust) without disclosing attributes. Extensive analysis and experiments show the effectiveness of FaceChange on protecting node privacy and meanwhile supporting the encountering information collection in MOSNs. Real implementation on smartphones also demonstrates its energy efficiency.**

I. Introduction

As a special form of delay tolerant networks (DTNs) [1], mobile opportunistic social networks (MOSNs) [2], [3] have attracted much attention due to the increasing popularity of mobile devices, e.g., smartphones and tablets. In MOSNs, mobile devices carried by people communicate with each other directly when they meet (i.e., within the communication range of each other) opportunistically. Therefore, since the encountering between devices reflects the meeting of people holding them, MOSNs can support proximity-based social network services without the support of infrastructures.

For example, based on the MOSN model, we can realize various applications without the support of infrastructures such as packet routing between mobile nodes [4], encountering based social community/relationship detection [5], [6], and distributed file sharing and Question & Answer (Q&A) [7]–[9] in a community. In each system, a node is uniquely labeled by an unchanging ID (defined real ID), which is obtained from the trust authority, for the corresponding service. Furthermore, since these services are built upon node encountering, nodes need to collect real ID based encountering information. For example, a node needs to know whom they have met to identify proximity based social community/relationship. In packet routing, a packet is always forwarded to the node that can more frequently meet its destination. Thus, nodes need to collect the encountering information to deduce their meeting frequencies with others for relay node selection.



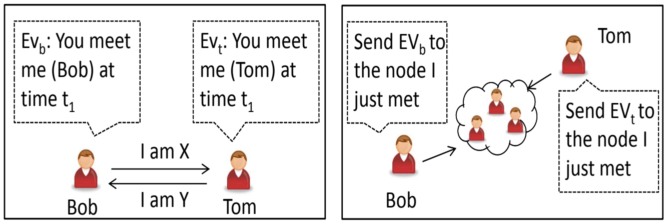
(a) Possible privacy issue. (b) Solution: neighbor Anonymity.

Fig. 1: Demonstration of a privacy issue and a possible solution in MOSNs.

In current MOSN applications, nodes can collect real ID based encountering information easily since neighbor nodes communicate with real IDs directly. We define two nodes as neighbor nodes when they are within the communication range of each other. However, when using real IDs directly, the disclosure of node ID to neighbor nodes would create privacy and security concerns. For example, a malicious node can first know the IDs of some central nodes or nodes with specific interests. Then, as shown in Figure 1(a), when neighbor nodes communicate with real IDs, a malicious node can easily identify attack targets from neighbors and launch attacks to degrade the system performance or steal important documents. Also, without protection, malicious nodes can easily sense the encountering between target nodes for attacks.

Therefore, it is critical to provide neighbor node anonymity to prevent the disclosure of real IDs to neighbors. Clearly, a permanent pseudonym cannot achieve such a goal since it can be linked to a node, which can still enable malicious nodes to recognize targets from neighbor nodes. Then, an intuitive method to realize the neighbor node anonymity is to let each node continuously change its pseudonym used in the communication with neighbors, as shown in Figure 1(b). However, when neighbor node anonymity is enforced, nodes cannot collect the real ID based encountering information, which disables aforementioned MOSN services.

Consequently, there is a challenge on anonymizing neighbor nodes for privacy protection and meanwhile still supporting MOSN services. Though there are rich investigations on protecting node privacy in MONs through anonymization [10]–



Ev_b: You meet me (Bob) at time t_1
Ev_t: You meet me (Tom) at time t_1
I am X
I am Y
Bob
Tom
Send EV_b to the node I just met
Tom
Send EV_t to the node I just met
Bob

(a) Create the encountering evidence under neighbor node anonymity.

(b) Route the encountering evidence to the other after separation.

Fig. 2: General solution for encountering record collection.

[18], such a problem has not been well solved. Therefore, we propose FaceChange to solve the challenge by keeping nodes anonymous during the encountering and postponing the real ID based encountering information collection to a moment after two neighbor nodes disconnect with each other. This is because disconnected nodes cannot communicate with each other directly in MOSNs, which makes attacking disconnected nodes hard. Thus, knowing real IDs after the encountering does not compromise the privacy protection.

Figure 2 illustrates the design of FaceChange. When two nodes meet, they communicate anonymously. However, each of them creates an encountering evidence that contains their real IDs. The evidences are sent to the other node only when they separate, thus enabling the encountering information collection while keeping the anonymity during the encountering. For an evidence, we call the node that creates it as the creator and the other node as the recipient. In this process, FaceChange needs to handle below challenges.

- The safety of the encountering evidence needs to be ensured. An encountering evidence can only be accessed by its creator and recipient and cannot be forged by others.
- An encountering evidence needs to be successful delivered to its recipient even when the real ID of the recipient node is unknown due to neighbor node anonymity.
- When creating an encountering evidence, a node can have fine-grained control over what contents (e.g., node ID, encountering time, or location) are included in the encountering evidence based on its trust on the encountering node. The calculation of the trust should also be privacy-preserving.

FaceChange incorporates the following schemes to handle the three challenges.

Encountering Evidence Encryption and Validation Scheme. For each encountering evidence, FaceChange uses the bilinear pairing technique [19] to generate an encryption key and a pair of uniquely matched token and commitment for it with efforts from both encountering nodes. The property of the bilinear pairing ensures that nodes other than the creator and recipient, even eavesdroppers, cannot know the key. Further, the token is attached to the evidence and the commitment is stored on the recipient node for validation, thereby ensuring the uniqueness of each encountering evidence.

Encountering Evidence Relaying Scheme. In this scheme, during the encountering, the recipient node specifies a relay node and encrypts its real ID with the public key of the relay node. It then forwards such information to the creator. Later, after the two nodes separate, the creator routes the encountering evidence to the relay node, which decrypts the ID of the recipient node and further routes it to the recipient node, thereby delivering the encountering evidence.

Encountering Evidence Generation Scheme. More similar attributes (e.g., affiliation and reputation) between two nodes often denote higher trust between them [12]. Thus, we realize the fine-grained control on the contents in an encountering evidence based on the attribute similarity. We use the commutative encryption [20] and the solution for "the millionaire's problem" [21] to calculate the attribute similarity blindly in this process, which protects node privacy.

In summary, the major contribution of this paper is to propose a novel design that supports both neighbor node anonymity and real ID based encountering information collection in MOSNs. FaceChange just prevents two encountering nodes from disclosing the real IDs during the encountering, so malicious nodes cannot identify targets from neighbors for attack. When nodes move away from each other, they know the real IDs of nodes they have met to support MOSN services. This is acceptable since a malicious node cannot communicate with a disconnected node and attack it in MOSNs.

In the following, Section II introduces related work. Section III presents the preliminary background. Section IV introduces the design of FaceChange. Section V evaluates FaceChange through trace-driven and smartphone-based experiments. Section VI concludes this paper with future work.

II. Related Work

A. Social Network based Applications in MOSNs

There are already many social network based MOSN routing algorithms [6], [22]–[25]. These works utilize various social factors such as frequently met friends, co-location records, centrality, transient contacts, and contact-based community to deduce a node's future meeting abilities with other nodes. Then, packets are always forwarded to the node with higher ability to meet their destinations.

There are also some applications in MOSNs. The work in [5] proposes three distributed community detection methods in DTNs. In SMART [6], each node constructs a social map including frequently met nodes to guide packet routing. The works in [7] and [8] realize peer-to-peer (P2P) file sharing and publish/subscribe overlay in DTNs, respectively. In PeopleNet [9], questions are first forwarded to matched geographical community and then propagated within the community via P2P connectivity to seek for answers.

Neighbor nodes in these algorithms communicate directly to collect encountering information for these services. Then, mobile users may be reluctant to participate in the MOSN services due to privacy concerns. Therefore, it is essential to provide neighbor node anonymity for privacy protection.

B. Privacy Protection in MOSNs

Some works [10]–[13] anonymize node interests or attributes for privacy protection in MOSNs. The work in [10] uses the solution for "the millionaire's problem" [21] to blindly check whether two nodes have similar interests. PreFiler [11] and the work in [12] adopt attribute-based encryption and/or bilinear pairing technique to blindly check whether a packet matches the destination's interests and whether a node owns the attributes to hold a packet, respectively. In STAP [13], packets for a node are cached in places where it visits frequently. As a result, nodes can fetch packets for them without disclosing their location information.

The works in [14]–[16] provides anonymous profile matching between nodes in MOSNs. FindU [14] leverages the secure multi-party communication techniques to enable a user to find the best match user with limited information exchange. The work in [15] designs a fine grained profile matching algorithm based on Paillier Cyptosystem. Liang *et al* [16] further propose a serials of profile matching algorithms with full anonymity. The work in [17] lets each node continually change its pseudonyms to protect its privacy in MOSNs. In [18], an anonymous architecture (e.g., ID anonymity) is proposed to provide anonymous communication in DTNs.

Although previous method are effective on protecting various privacies in MOSNs, they fail to investigate how to safely collecting real ID based encountering information under neighbor node anonymity, which is crucial for MOSNs. There are also researches working on secure and privacy-preserving communication between neighboring mobile devices [26]–[28]. However, these systems rely on infrastructures to set up trust or ensure privacy protection, which does not apply to pure MOSN scenario that does not have infrastructures.

III. Preliminaries

A. Network Model

We focus on a mobile opportunistic social network with m human-carried mobile devices, denoted by N_i ($i \in [1, m]$). We assume that the network size is large. Otherwise, a node can easily guess the identity of its neighbors. These devices/nodes move in the network following the mobility of people carrying them. We use node and user interchangeably in this paper. Each node (i.e., device) has a limited communication range, and two nodes can communicate only when they are within the communication range of each other.

We assume a Trust Authority (TA) in the system responsible for system parameters and certificates distribution and attribute validation (e.g., reputation, affiliation, and ID), both of which can be conducted off-line. Each node has a unique real ID in the network for MOSN services, denoted by NID_i. The real ID of each node is assigned by the TA with a signature generated by the TA's private key. Then, nodes can verify the authenticity of received real IDs efficiently.

B. Adversary Model

In this paper, we assume malicious nodes can attack target nodes only when they find targets from neighbor nodes. This is reasonable since 1) an attacker in MOSNs cannot communicate with the target directly if they are not neighbors, and 2) it is costly to attack every encountered node. This means that malicious nodes can steal privacies or launch attacks only after identifying target nodes from neighbor nodes. Thus, we focus on preventing real ID leakage during the communication between neighbor nodes in this paper.

C. Cryptographic Techniques

1) Bilinear Pairing: Let $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_\mathbb{T}$ be three cyclic groups with the same prime order q, and $P \in \mathbb{G}_1$ and $Q \in \mathbb{G}_2$ be generators of $\mathbb{G}_1$ and $\mathbb{G}_2$, respectively. A bilinear pairing is a map e: $\mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_\mathbb{T}$ satisfying the following properties [19]:

- Bilinearity: $\forall\ a, b \in \mathbb{Z}_q^*$: $e(aP, bQ) = e(P, Q)^{ab}$
- Non-degeneracy: $e(P, Q) \neq 1$
- Computability: e can be computed efficiently

We utilize symmetric pairing in this paper, in which $\mathbb{G}_1 = \mathbb{G}_2 = \mathbb{G}$ and they have the same generator P. As mentioned in Section IV-A, upon the start of the system, the TA first generates parameters for adopted bilinear pairing, i.e., $BiParas$. In this step, TA randomly selects a security parameter ς and runs the bilinear pairing generation function $\mathcal{F}(\varsigma)$ to generate these parameters ($BiParas$): $(e, q, P, \mathbb{G}, \mathbb{G}_T)$.

2) Commutative Encryption: A commutative encryption algorithm $\mathbb{E}(\cdot)$ [20], [29] satisfies the commutative property. That is, for any encryption keys k_i and k_j, message M, rational number t and $\gamma < 1/2^t$, it holds

- $\mathbb{E}_{k_i}(\mathbb{E}_{k_j}(M)) = \mathbb{E}_{k_j}(\mathbb{E}_{k_i}(M))$,
- $\forall\ M_1 \neq M_2$, $Pr(\mathbb{E}_{k_i}(\mathbb{E}_{k_j}(M_1)) = \mathbb{E}_{k_j}(\mathbb{E}_{k_i}(M_2))) < \gamma$.

where $\mathbb{E}_{k_i}(M)$ is the result of encrypting M with key k_i.

IV. System Design of FaceChange

A. System Setup

Upon the bootstrap of the system, the TA first generates parameters for the adopted bilinear pairing, i.e., $BiParas$, the detail of which is introduced in Section III-C1. TA also selects a secure commutative encryption algorithm $\mathbb{E}()$ [20] and a collision-resistant hashing function $\mathcal{H}()$ [30], which are used for encountering evidence encryption. Additionally, TA generates a pair of public key and private key (PK_T, SK_T) through the public-key cryptography, e.g., RSA [29]. Finally, TA generates the system parameter $SysPara = (BiParas, \mathbb{E}(), \mathcal{H}(), PK_T)$, where $BiParas$ represents the bilinear pairing parameters

When a node N_i joins in the system, it registers to the TA through the following steps:

- N_i creates a pair of public/private key (PK_i, SK_i) by the same method used by TA and reports PK_i to TA.
- N_i fetches the system parameter $SysPara$ and its unique real ID NID_i from TA.

B. Neighbor Node Anonymity Provided by FaceChange

Neighbor node anonymity means that each node does not know the real IDs of its neighbor nodes. To realize this goal, FaceChange lets each node communicate anonymously with neighbor nodes. Specifically, whenever a node disconnects with a neighbor node, it randomly changes its pseudonyms in all communication layers (e.g., MAC address, IP address and application pseudonym) and communication parameters (e.g., signal strength), which will be used for the communication with the next encountered node. Note that both MAC and IP addresses can be easily modified through software [31].

Therefore, the pseudonyms and parameters used by a node are non-linkable, which means that malicious nodes cannot directly identify a node from the IDs and parameters it uses. We further carefully design the encountering evidence generation and collection in FaceChange to ensure that neighbor node anonymity is maintained in these processes. Section IV-H1 gives out the final analysis to prove the neighbor node anonymity. For easy description, we use PID_i to uniformly represent node N_i's various pseudonyms. We also use NID_i to represent N_i's unique real ID.

C. Challenges on Encountering Information Collection

In FaceChange, neighbor nodes communicate anonymously to protect their privacy. However, MOSN services require the real ID based encountering information. To solve such a problem, each node creates an encountering evidence for the other to know the encountering information (e.g., whom it has met), as shown in Figure 2(a). To ensure neighbor anonymity, the encountering evidence is routed to the other node only after they separate from each other, as shown in Figure 2(b).

However, there are several challenges in this solution. First, the safety of encountering evidences needs to be ensured against privacy leakage and fabrication during the routing. Second, the encountering evidence needs to be successfully and uniquely collected. Third, when creating an encountering evidence, a node may want to control the content in the evidence based on its trust on the encountering node. Sections IV-D, IV-E, and IV-F present the detail of proposed schemes that can solve the three challenges, respectively.

In the following, we use the case in which N_i creates an encountering evidence for N_j to illustrate the three schemes. The major notations are illustrated in Table I.

TABLE I: Notations.

Notation	Meaning
N_i	The i-th node in the MOSN
NID_i	The real ID of node N_i
$PID_i(t)$	The pseudonym of node N_i at time t
$\mathcal{EV}_{ij}(t)$	The encountering evidence generated by N_i for N_j at time t
$\mathcal{EV}'_{ij}(t)$	The encountering evidence $\mathcal{EV}_{ij}(t)$ after encryption
$\mathcal{Y}_i$	The encountering evidence generation policy of N_i
$\mathcal{S}_i$	The attribute set of N_i
$\mathcal{S}_{ti}$ & $\mathcal{S}_{vi}$	The type-based & value-based attribute subset of N_i
t_{ik}	The k-th type-based attribute of N_i
v_{ik}	The k-th value-based attribute of N_i
a_k & val_k	The name and value of v_{ik}
$\mathbb{E}(\cdot)$	The adopted commutative encryption algorithm
$Enc(\cdot)$	The adopted public-key encryption algorithm
$\mathcal{H}(\cdot)$	The adopted collision-resistant hashing function

D. Encountering Evidence Encryption and Validation

When N_i meets N_j, it creates an encountering evidence for N_j, denoted by $\mathcal{EV}_{ij}(t)$, to record their encountering. We introduce the encountering evidence creation process later in Section IV-F. N_i then routes $\mathcal{EV}_{ij}(t)$ to N_j after it disconnects with N_j. Since the evidence is routed by nodes in the network, its safety and confidentiality needs to be ensured.

We propose a novel encountering evidence encryption and validation scheme to solve the problem. In this scheme, an encryption key is generated with data from both encountering nodes. Then, other nodes cannot deduce the key and know the contents of the encountering evidence. Further, a pair of uniquely matched token and commitment is created for each encountering. The token is attached to the encountering evidence, while the commitment is stored on the recipient node. As a result, forged encountering evidences can be identified by the recipient node. The scheme is also resistant to the collusion attack, as proven in Section IV-D2.

Below, we first introduce the details of the proposed scheme and then present the security and cost analysis.

1) Ensuring the Safety of Encountering Evidences: To protect the safety of encountering evidences, FaceChange uses the bilinear pairing to generate the encryption key, token, and commitment. Generally, each of the two encountering nodes, i.e., N_i and N_j, first generates a random number, i.e., r and s. They then generate the encryption key as $e(rP, sP)$. They further reutilize s and r to generate the token and commitment as $r(s + \mathcal{H}(PID_j(t)))P$ and $\frac{P}{s+\mathcal{H}(PID_j(t))}$, respectively, where $\mathcal{H}(PID_j(t))$ is the hashing value of the pseudonym of the recipient (N_j). The detailed security analysis of this scheme is provide in Section IV-D2.

Specifically, N_i and N_j first select a random number $r \in \mathbb{Z}_q^*$ and $s \in \mathbb{Z}_q^*$, respectively. N_j selects a s that is not used by any commitments in its commitment list. N_j then sends sP and the hash of its pseudonym at the encountering time t, i.e., $\mathcal{H}(PID_j(t))$, to N_i for encryption key and token generation

$$N_j \rightarrow N_i : sP \quad and \quad \mathcal{H}(PID_j(t))$$

N_i also randomly generates a key k_r. Then, N_i computes the encrypted encountering evidence as $\mathcal{EV}'_{ij}(t) = (Z_1, Z_2, Z_3, Z_4)$, where

$$\begin{cases} Z_1 = r(sP + \mathcal{H}(PID_j(t))P) \\ Z_2 = e(rP, P) = e(P, P)^r \\ Z_3 = \mathbb{E}_{k_s}(\mathcal{EV}_{ij}(t)), \ k_s = e(rP, sP) \\ Z_4 = \mathbb{E}_{k_s}(k_r) \end{cases} \tag{1}$$

In $\mathcal{EV}'_{ij}(t)$, Z_1 is the token, Z_2 is the verification number for the commitment, Z_3 is the encountering evidence encrypted by key k_s, and Z_4 is key k_r encrypted by k_s. Note that Z_2 is also used for the recipient to deduce the encryption key.

N_i further sends its real ID encrypted by k_r, i.e., $\mathbb{E}_{k_r}(NID_i)$, to N_j

$$N_i \rightarrow N_j : \mathbb{E}_{k_r}(NID_i)$$

Then, N_j computes the commitment as below and inserts it into its commitment list.

$$CT_{js}: \ < \frac{P}{s + \mathcal{H}(PID_j(t))}, \mathbb{E}_{k_r}(NID_i), s > \tag{2}$$

We can see that in this commitment, NID_i represents the ID of the node that N_j actually meets during the encountering corresponding to this commitment. It is stored in the commitment to prevent encountering evidence fabrication under eavesdropping, as introduced in the next subsection.

When N_j receives an encrypted encountering evidence $\mathcal{EV}'_{xj}(t_k) = (Z_1, Z_2, Z_3, Z_4)$, it checks whether Z_1 matches with any commitment in its commitment list. Suppose there is a commitment $CT_{ju} :< \frac{P}{s+\mathcal{H}(PID_j(t_k))}, \mathbb{E}_{k_r}(NID_x), u >$ satisfying $e(Z_1, \frac{P}{s+\mathcal{H}(PID_j(t_k))}) = e(P,P)^r = Z_2$, this means that the received $\mathcal{EV}'_{xj}(t_k)$ matches the commitment based on the properties of bilinear pairing and the fact that the s in each commitment in the commitment list is unique. Then, Z_3 is decrypted with key $k_s = (Z_2^u) = e(P,P)^{ru}$ to obtain the content of the encountering evidence and k_r. After this, the commitment is removed from the node's commitment list.

2) Security Analysis for Evidence Encryption and Validation: The above scheme can ensure the confidentiality and uniqueness of each encountering evidence.

First, the privacy in the encountering evidence can be protected. Recall that in the encryption key generation process, N_j only sends sP to N_i, and N_i only attaches $e(rP, P)$ to the encountering evidence. This means that a malicious node can at most know sP and $e(rP, P)$, which are not sufficient to deduce the encryption key ($e(rP, sP)$). Therefore, an encountering evidence's contents are protected against nodes other than its creator and recipient.

Second, the encountering evidence forgery can be prevented. Based on the discussion in Section III-C1, the token $r(s + \mathcal{H}(PID_j(t)))P$ is uniquely matched with the commitment $\frac{P}{s+\mathcal{H}(PID_j(t))}$. Therefore, malicious nodes cannot create a valid token for fabricated encountering evidences that can pass the check on the recipient node.

However, a malicious node, say N_m, can eavesdrop the communication between N_i and N_j and know sP and $\mathcal{H}(PID_j(t))$. Then, it can generate a random number r^* and a key k_r^* to forge an encrypted encountering evidence $\mathcal{EV}'_{mj}(t)$

$$\left\{ \begin{array}{l} Z_1^* = r^*(sP + \mathcal{H}(PID_j(t))P) \\ Z_2^* = e(P,P)^{r^*} \\ Z_3^* = \mathbb{E}_{k_s}(\mathcal{EV}_{mj}(t)),\ k_s = e(r^*P, sP) \\ Z_4^* = \mathbb{E}_{k_s}(k_r^*) \end{array} \right. \tag{3}$$

We can see that $e(Z_1^*, \frac{P}{s+\mathcal{H}(PID_j(t_k))}) = e(P,P)^{r^*} = Z_2^*$, which means that the faked encountering evidence matches the commitment created for the encountering between N_i and N_j. Then, N_m can make N_j believe a non-existing encountering. However, the design of $\mathbb{E}_{k_r}(NID_i)$ in the commitment can prevent this attack. This is because the decryption of $\mathbb{E}_{k_r}(NID_i)$ with k_r^* would lead to a node ID that is different with the one claimed in the faked evidence $\mathcal{EV}_{mj}(t)$. This decrypted ID can be regarded as some sort of random since it is encrypted by one key and decrypted by another key. Then, N_j can know it is not a valid ID based on the list of legal user IDs in the system and drop the faked encountering evidence shown in Formula (3). As a result, FaceChange ensures the uniqueness of each encountering evidence.

Furthermore, as shown in [32], the k-CAA (collusion attack algorithm with k traitors) can hardly work in above commitment scheme. That is, given $(P, Q = sP, h_1, h_2, \cdots, h_k \in \mathbb{Z}_q^* \textit{ and } \frac{1}{h_1+s}P, \frac{1}{h_1+s}P, \frac{1}{h_2+s}P, \cdots, \frac{1}{h_k+s}P)$, there is no polynomial-time algorithm that can compute $\frac{1}{h^*+s}P$ for some $h^* \notin \{h_1, h_2, \cdots, h_k\}$ with non-negligible probability. This means that a commitment $(\frac{P}{s+\mathcal{H}(PID_j(t))})$ can hardly be forged by nodes other than its creator (N_j). Therefore, even when a malicious node can intrude another node, it cannot purposely create commitments on the node that can match the tokens on fabricated encountering evidences.

3) Cost Analysis: In the commitment generation process, bilinear pairing accounts for major computing. As introduced in [11], we can use Tate pairing, in which each element in $\mathbb{G}$ is 512-bit and q is a 160-bit prime. The computation cost for a pairing then is around 8.5 ms in a Pentium III 1GHz machine [11]. Therefore, considering modern devices (e.g., smartphones) usually have higher capacity than the Pentium III machine, the cost of the bilinear pairing is acceptable.

E. Encountering Evidence Relaying Scheme

After disconnects with N_j, N_i routes the created encountering evidence to N_j. However, due to node anonymity, N_i cannot know the real ID of N_j, which is the recipient of the evidence. We propose an encountering evidence relay scheme to solve this problem. In this scheme, during the encountering, the recipient node N_j specifies a relay node and encrypted its real ID with the public key of the relay node. Such data is forwarded to the evidence creator N_i. Then, after the two nodes disconnect, the creator routes the encountering evidence to the relay node, which first decrypts the recipient node's ID and then routes the evidence to the recipient node.

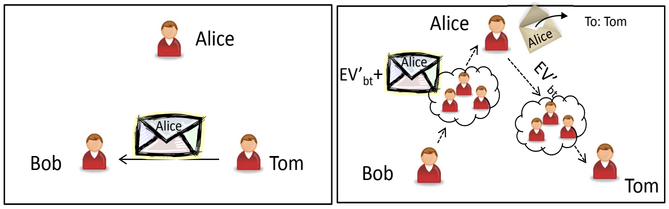
Figure 3 demonstrates this scheme. When Bob and Tom meets, Tom informs Bob that the encountering evidence should be relayed by Alice and inserts its real ID inside the envelope. His real ID can only be seen by Alice and cannot be seen by Bob. Then, when Alice receives it, as shown in Figure 3(b), it finds that the recipient is Tom and routes the encountering evidence to Tom. The two clouds in Figure 3(b) mean that the message is routed by a certain routing algorithm.

In the following, we first introduce the details of the relay scheme and then present the security and cost analysis.

1) Relay Node Selection: In this process, to prevent privacy leakage, we do not allow the two nodes (i.e., N_j and N_i) to communicate to select the relay node. Instead, the recipient node of an encountering evidence, say N_j, randomly selects a relay node from the set of nodes it trusts.

We assume the chance of an individual malicious node inferring a neighbor's real identity from the relay node it selects is negligible. Such a claim holds because 1) the system usually includes a large number of nodes, 2) each node has a large number of trusted relay nodes, and 3) relay nodes are shared across a large number of nodes in the system. This means that a relay node may be used by many nodes, and a node may use many nodes as relay nodes. We further assume that there are few repeated and consecutive encounters between any pair of nodes, and nodes do not collude to collect the relay nodes of a node. As a result, a node can hardly deduce a neighbor's identity based on its relay node.

2) Relaying the Encountering Evidence: We use RN to denote the selected relay node. The recipient node, i.e.,



(a) Select the relay node.

(b) Relay to the recipient.

Fig. 3: Relaying encountering evidence to the recipient.

N_j, generates a random key k_y to encrypt its real ID, i.e., $\mathbb{E}_{k_y}(NID_j)$, and then encrypts k_y with the public key of the relay node: $Enc_{PK_r}(k_y)$. $\mathbb{E}$ and Enc refer to the commutative encryption algorithm and the public-key encryption algorithm, respectively. Then, N_j sends both encrypted items to N_i when they are still neighbors of each other. The design of k_y is to prevent disclosing N_j's real ID. That is, if N_j forwards $Enc_{PK_r}(NID_j)$ directly to N_i, N_i can deduce N_j since it knows PK_r and all real node IDs. Finally, N_i generates the encountering message as below.

$$(RN, Enc_{PK_r}(k_y), \mathbb{E}_{k_y}(NID_j), \mathcal{EV}'_{ij}(t)), Sign_{SK_i}(\mathcal{EV}'_{ij}(t))) \quad (4)$$

where RN denotes the relay node, $\mathcal{EV}'_{ij}(t)$ is the encrypted encountering evidence (Formula (1)), and $Sign_{SK_i}(\mathcal{EV}'_{ij}(t))$ is a signature generated by N_i that can ensure the integrity and authenticity of the encrypted evidence.

After the two nodes separate, N_i routes the message to RN. Upon receiving the message, RN decrypts $Enc_{PK_r}(k_y)$ with its private key SK_r and knows that the recipient of the message is NID_j. Then, it routes below to N_j:

$$(NID_j, \mathcal{EV}'_{ij}(t), Sign_{SK_i}(\mathcal{EV}'_{ij}(t))) \quad (5)$$

After receiving the above message, N_j can obtain the encountering evidence from $\mathcal{EV}'_{ij}(t)$ by following the decryption procedure mentioned in Section IV-D1.

We adopt MOSN routing algorithms, e.g., RAPID [1] and PROPHET [33], to route an encountering evidence to the relay node or N_j. The delay of such routing usually is large, and some packets may fail to reach the destination, as shown in Section V-B, since they use the hop-by-hop relay to forward packets and assume no network infrastructure. However, we can import network infrastructures to reduce the routing delay and ensure evidence delivery (i.e., allow packets with a large delay to be forwarded through infrastructures).

3) Security Analysis for Evidence Relaying: The designed scheme can provide safe encountering evidence relay.

First, the confidentiality of the encountering evidence is maintained. The content of $\mathcal{EV}'_{ij}(t)$ cannot be seen by any intermediate nodes. This is because the encryption key k_s is only known by N_i and N_j, as proven in Section IV-D2. Further, the signature of the encrypted encountering evidence $\mathcal{EV}'_{ij}(t)$ in the relayed message, as shown in Formulas (4) and (5), ensures its integrity and authenticity.

Second, by requiring N_j to select relay node only from nodes it trusts, the possibility that N_i and the selected relay node RN collude can be greatly limited in FaceChange. Otherwise, by colluding with RN, N_i can know the private key of RN (PK_r) and know N_j's real ID during the encountering.

Third, since the relay node is selected by N_j, it may collude with the relay node or even use itself as the relay node. However, even in such an attack, N_j still cannot know the real ID of N_i during the encountering. This is because N_i forwards the encountering message to other nodes only after it separates with N_j. Then, when N_j receives the message from another node, say N_x, it cannot determine that N_x is N_i since N_x may be a node that just relays the message.

4) Cost Analysis: The extra costs in this step are mainly from the encountering evidence relaying. In MOSNs, nodes usually are sparsely distributed and meet opportunistically, which means that the number of encountering evidences in a unit time is limited. Further, an encountering evidence only contains simple information with a limited size. It can be attached to the packet routing with no additional processing. Therefore, the cost on relaying encountering evidences is constraint and will not drain the network resources.

F. Encountering Evidence Generation Scheme

We introduce how to create encountering evidence when two nodes meet in a privacy-preserving manner in this section. The basic idea is to create the encountering evidence based on the trust. In FaceChange, each node, say N_i, maintains a policy, $\mathcal{Y}_i$, to decide what information can be included in the encountering evidence for each trust level. Below, we first define attributes and evidence creation policy and then present the encountering evidence generation process. We also present the security and cost analysis of this scheme in the end.

1) Attribute Definition: Both type-based and value-based attributes are supported in FaceChange. The type-based attributes, e.g., organization and interests, refer to those that represent certain properties with no numerical meaning. The value-based attributes, e.g., reputation and age, refer to those that can be represented by numeric values. How a node's attributes are obtained is not the focus of this paper.

Then, the attribute set of a node, say N_i, can be expressed as $\mathcal{S}_i : \{y_{i1}, y_{i2}, y_{i3}, \cdots, v_{i1}, v_{i2}, v_{i3}, \cdots\}$, where y_{im} and v_{in} represent a type-based attribute and a value-based attribute, respectively. v_{in} is represented as a $[name : value]$ pair. For example, the attribute set of a student can be expressed as $\mathcal{S}_i : \{ABCUniv., Student, [reputation : 0.8], [age : 20]\}$.

2) Evidence Creation Policy: We follow the concept in [12], [34] to decide a node's trust on an encountering node based on the similarity between attributes. For two encountering nodes, the more common attributes they have, the more trust they have on each other and the more information about the encountering can be disclosed. Then, the evidence creation policy on a node, say N_i, is built upon the match value between its attributes with those of the encountering node, say N_j, denoted by $MatchV = |\mathcal{S}_i \cap \mathcal{S}_j|$. The match value is compared with a set of monotonically increasing thresholds, i.e., $\{T_1, T_2, \cdots, T_n\}$, to determine the amount of information in the encountering evidence. Specifically,

- If $MatchV \leq T_1$, this means that N_j is not trustable. Then, N_i does not create the encountering evidence.

- If $T_1 < MatchV \leq T_2$, N_i creates an evidence with basic information needed by the MOSN application, such as the real ID of N_i (NID_i) and the encountering time.
- $\cdots\cdots$
- If $T_n < MatchV$, N_i creates an evidence with full information, including NID_i, encountering time, contact length and location, and any other useful information.

The above policy is only an example to demonstrate fine-grained control over the content in the encountering evidence.

3) Blind Attribute Checking: FaceChange utilizes the commutative encryption and the solution for "the millionaire's problem" [21] to calculate the match value blindly.

The attribute set of N_i, denoted $\mathcal{S}_i$, can be split into two subsets consisting of the two types of attributes: $\mathcal{S}_i = \{\mathcal{S}_{yi} \cup \mathcal{S}_{vi}\}$, where $\mathcal{S}_{yi}$ and $\mathcal{S}_{vi}$ represent the type-based attribute subset and the value-based attribute subset, respectively. We introduce how to calculate them separately.

Calculating the Match Value between Type-based Attribute Subsets ($|\mathcal{S}_{yi} \cap \mathcal{S}_{yj}|$): $|\mathcal{S}_{yi} \cap \mathcal{S}_{yj}|$ is calculated as the number of shared attributes in the two subsets. This process is conducted without disclosing each node's attributes by using a commutative encryption algorithm.

Specifically, N_i and N_j first select a random encryption key, say k_i and k_j, respectively. Then, each node encrypts the attributes in its type-based attribute subset with its encryption key. As a result, N_i has $\mathcal{S}'_{yi} = \{\mathbb{E}_{k_i}(y_{i1}), \mathbb{E}_{k_i}(_{i2}), \mathbb{E}_{k_i}(y_{i3}), \cdots\}$ and N_j has $\mathcal{S}'_{yj} = \{\mathbb{E}_{k_j}(y_{j1}), \mathbb{E}_{k_j}(y_{j2}), \mathbb{E}_{k_j}(y_{j3}), \cdots\}$. Then, each node sends the encrypted attributes to the other node.

$$N_i \to N_j : \mathcal{S}'_{yi} \quad and \quad N_j \to N_i : \mathcal{S}'_{yj}$$

Upon receiving $\mathcal{S}'_{yi}$ and $\mathcal{S}'_{yj}$, each node again encrypts each encrypted attribute with its key. Then, N_i has $\mathcal{S}''_{yj} = \{\mathbb{E}_{k_i}(\mathbb{E}_{k_j}(y_{j1})), \mathbb{E}_{k_i}(\mathbb{E}_{k_j}(y_{j2})), \mathbb{E}_{k_i}(\mathbb{E}_{k_j}(y_{j3})), \cdots\}$, and N_j has $\mathcal{S}''_{yi} = \{\mathbb{E}_{k_j}(\mathbb{E}_{k_i}(y_{i1})), \mathbb{E}_{k_j}(\mathbb{E}_{k_i}(y_{i2})), \mathbb{E}_{k_j}(\mathbb{E}_{k_i}(y_{i3})), \cdots\}$. After the second round of encryption, each node further sends the encrypted attributes to the other node.

$$N_i \to N_j : \mathcal{S}''_{yj} \quad and \quad N_j \to N_i : \mathcal{S}''_{yi}$$

Then, both nodes have both $\mathcal{S}''_{yi}$ and $\mathcal{S}''_{yj}$. They can check the number of the same attributes in $\mathcal{S}''_{yi}$ and $\mathcal{S}''_{yj}$ based on the aforementioned property of the commutative encryption: if $\mathbb{E}_{k_j}(\mathbb{E}_{k_i}(y_{ia})) = \mathbb{E}_{k_i}(\mathbb{E}_{k_j}(y_{jb}))$, then $y_{ia} = y_{jb}$.

Calculating the Match Value between Value-based Attribute Subsets ($|\mathcal{S}_{vi} \cap \mathcal{S}_{vj}|$): In this paper, we define $|\mathcal{S}_{vi} \cap \mathcal{S}_{vj}|$ as the number of N_j's value-based attributes that satisfy N_i's requirement on their values. In FaceChange, a node's requirement on a value-based attribute is represented by a threshold and an indication on the comparison direction, i.e., larger or smaller than the threshold. Specifically, suppose N_i's requirement on attribute a_n is $(VT_{ia_n}, \geq)$. Then, N_j's attribute $v_{jn} = [a_n, val_n]$ satisfies N_i's requirement if $val_n \geq VT_{ia_n}$.

In detail, $|\mathcal{S}_{vi} \cap \mathcal{S}_{vj}|$ is calculated by the following steps:

- N_i and N_j first decide the list of names of value-based attributes to compare, e.g., $\{a_1, a_2, a_3, \cdots\}$, and handle these names one by one.
- For each attribute name, say a_x, N_i picks its requirement for it: $(VT_{ia_x}, \geq)$, and N_j picks its value: v_{jx}.
- N_i and N_j compare VT_{ia_x} and v_{jx} by the solution for "the millionaire's problem" [21] without disclosing the values of VT_{ia_x} and v_{jx} to the other node.
- N_i checks whether the result satisfies the comparison direction (i.e., whether $v_{jx} \geq VT_{ia_x}$). If yes, $|\mathcal{S}_{vi} \cap \mathcal{S}_{vj}|$ increases by one. Otherwise, it remains unchanged.

The solution for "the millionaire's problem" enables two people ($Alice$ and Bob), each of whom has one number, to compare their numbers without disclosing their values. Please refer to [21] for the detail of this algorithm. We assume equal weight for each attribute in this paper. We can easily expand current design to the case with different attribute weights.

4) Fine-grained Evidence Generation: In summary, when N_i meets N_j at t, N_i first calculates the match value of its attribute set with that of N_j blindly (i.e., $MatchV = |\mathcal{S}_i \cap \mathcal{S}_j|$), as in Section IV-F3. Then, $MatchV$ is applied to its encountering evidence creation policy $\mathcal{Y}_i$ to decide what information can be included in the encountering evidence, as in Section IV-F2. Finally, N_i creates the evidence $\mathcal{EV}_{ij}(t)$ accordingly.

5) Security Analysis on Evidence Generation: First, with the commutative encryption, N_i cannot know the type-based attributes of N_j from $\mathcal{S}'_{yj}$ since it is encrypted by k_j, which is not known by N_i. Similarly, N_j cannot know the type-based attributes of N_i either. This means that $|\mathcal{S}_{yi} \cap \mathcal{S}_{yj}|$ is calculated blindly. Second, with the solution to "the millionaire's problem", N_i obtains $|\mathcal{S}_{vi} \cap \mathcal{S}_{vj}|$ blindly, i.e., without disclosing its thresholds or knowing the values of N_j's value-based attributes. In summary, attributes are compared blindly in FaceChange, thereby effectively protecting node privacy.

6) Cost Analysis: The extra costs in blind policy checking are incurred by the commutative encryption and the solution for "the millionaire's problem". For the commutative encryption algorithm, we can choose a suitable one to control the complexity. Note that a good property of our scheme is that the key used by a node can change after each policy checking. Then, simple commutative encryption algorithm, e.g., XOR, can provide reliable encryption at a low cost.

The complexity of the solution for "the millionaire's problem" is $O(d^2)$ [21], where d is the length of the binary representation of the compared value. While d can be controlled to be 8, i.e. char, the extra cost for this step is acceptable.

G. Realizing General Packet Routing in FaceChange

In this section, we take packet routing as a case to show how a MOSN service is realized under FaceChage.

1) Routing Utility Update: The received encountering evidences on each node are utilized to update the routing utility used for packet routing. One common routing utility is the future meeting probability with a node.

However, encountering evidences may not arrive in the same order in which they are created due to the opportunistic packat routing in MOSNs. Therefore, FaceChange adopts a cache period, denoted T_c, to maximally solve this problem. When a node receives an encountering evidence, it stores it in its memory. Then, at the end of the N-th cache period ($N > 2$), i.e, at $N * T_c$, the received encountering evidences that are created before $(N-1) * T_c$ are handled in the order of their creation times to update related routing utilities.

2) Packet Routing Process: In traditional MOSN packet routing, two encountering nodes first delivers packets destined for the other node. They then compare routing utilities and forward the other node packets that the other node has higher routing utilities for their destinations.

In FaceChange, neighbor node anonymity blocks the first step by preventing nodes from recognizing the destinations of their packets even when meeting them. To solve this problem, we let each node claim to have higher routing utility for itself to fetch packets for it. In detail, N_i only tells N_j that it is more suitable to carry packets for N_i (i.e., has higher routing utility for N_i). Then N_j would send N_i packets destined for it. Following such a scheme, nodes can correctly compare utilities to forward packets and deliver packets to their destinations.

H. Security Analysis of FaceChange

We further analyze how FaceChange ensures node anonymity and the safety of the encountering evidence collection from the perspective of the system.

1) Ensuring Neighbor Node Anonymity: First, neighbor nodes are anonymized in FaceChange by constantly changing their pseudonyms (Section IV-B). The encountering evidence relaying scheme (Section IV-E) allows two nodes to collect the encountering information without disclosing their real IDs during the encountering, as proven in Section IV-E3.

Second, nodes cannot be linked in FaceChange. As previously explained, two neighbor nodes do not transmit any linkable information in encountering evidence encryption/commitment (Section IV-D2), encountering evidence collection (Section IV-E3), and encountering evidence generation (Section IV-F5). To receive packets destined for it, a node just claims to be a better forwarder for these packets. As a result, a node can not be linked by tracking packets for it. In summary, none linkable information of a node is disclosed.

The above two features ensure that node anonymity is maintained under the bogus attack. Since neighbor nodes are anonymized and none linkable information of a node is disclosed to any neighbor nodes, creating many sybils cannot help deduce the real ID of a neighbor node.

2) Ensuring Encountering Information Collection: The encountering information can be confidentially and correctly collected by nodes in FaceChange. As introduced in Section IV-D2 and Section IV-E3, the encryption key k_y and the signature ensure the confidentiality, integrity, and authenticity of each encountering evidence.

3) Preventing Fabricating Encountering: With the designed commitment scheme in Section IV-D, nodes cannot claim non-existing encountering with others. As introduced in Section IV-D2, the generated token and commitment are uniquely matched, which prevents attackers from arbitrarily creating fake encountering evidences. A commitment is deleted after a successful match, which prevents attackers from poisoning the system by re-sending overheard evidences.

Malicious nodes may eavesdrop commitment parameters to forge an encountering evidence that can pass the commitment verification. However, this can be prevented since the creator of the forged evidence cannot be the same with the one in the commitment (Section IV-D2). Furthermore, there is no polynomial-time algorithm that can generate a fake commitment on a node with non-negligible probability. Then, even the intruder of a node cannot create commitments for its forged encountering evidences on the node.

V. Performance Evaluation

In this section, we first examine the effectiveness of neighbor node anonymity and encountering evidence collection and then show the energy consumption of FaceChange.

We adopted two real traces in the tests: the MIT Reality trace [35] and the Haggle project trace [36]. The former trace records the meetings between students and teachers on MIT campus for about 30 days, while the latter trace includes the encountering between scholars attending Infocom 2006 for about 4 days. We adopt the two traces since they represent typical MOSN scenarios in which mobile devices meet opportunistically. We wrote an event-driven simulator for the experiment. The connectivity between nodes is inferred from contact times in the trace.

We adopted PROPHET [33] as the underlying routing algorithm in the experiments. In PROPHET, each node maintains its future meeting probabilities with other nodes based on previous records to guide packet routing.

A. Effectiveness of Privacy Protection

We first evaluate the effect of privacy protection. In this test, we measured the privacy leakage as duplicate pseudonyms (i.e., the average number of identical pseudonyms seen by a node) and disposed IDs (i.e., the number of identical pseudonyms used by a node). The pseudonyms include those advertised by each node for the communication with neighbor nodes and those encrypted IDs in the encountering evidences.

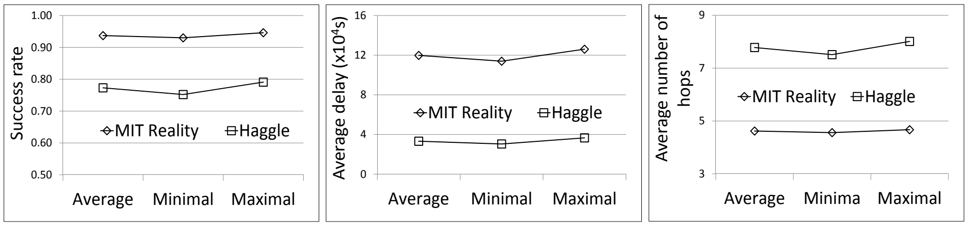
TABLE II: Effectiveness of Privacy Protection.

	Duplicate Pseudonyms	Disposed IDs
MIT Reality	8	0
Haggle	4	0

The test results are shown in Table II. We found that only few identical pseudonyms can be seen by each node and all identical pseudonyms are from different nodes in the system in the experiments with both traces. This means that nodes cannot use the transmitted pseudonyms to identify neighbor nodes. Such a result in conjunction with the analysis in Sections IV-B, IV-D2, IV-E3, and IV-F5 justify that FaceChange can effectively protect node privacy.

B. Efficiency of the Encountering Evidence Collection

In this test, we measured the success rate, average delay, and average number of hops of collected encountering evidences. The success rate refers to the percentage of successfully collected encountering evidences. The average delay and average hops denote the time and the forwarding hops each



(a) Success rate. (b) Average delay. (c) Average number of hops.

Fig. 4: Evidence Collection Efficiency with Both Traces.

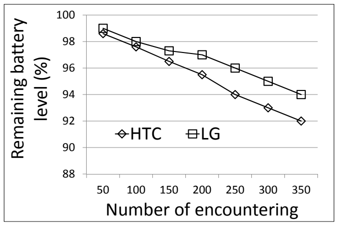
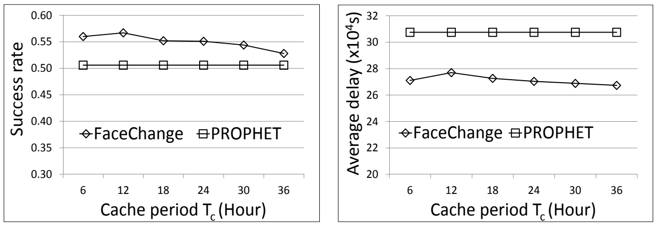
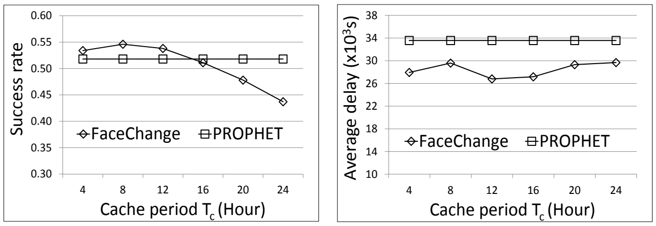


Fig. 5: Energy consumption in real test.



(a) Success rate. (b) Average delay.

Fig. 6: Packet routing efficiency with the MIT Reality trace.



(a) Success rate. (b) Average delay.

Fig. 7: Packet routing efficiency with the Haggle trace.

collected encountering evidence experiences on average. The test results are shown in Figure 4.

We see from the figure that the success rates reach about 93% and 77% in the tests with the MIT Reality trace and the Haggle trace, respectively. This shows that most encountering evidences can be successfully collected in FaceChange. The success rate is low in the Haggle trace because some nodes only exist for a short period of time in the trace.

We find that the average delays are about 120,000 seconds and 33,000 seconds in the tests with the two traces, respectively. Since the encountering frequencies between nodes in MOSNs usually follow a certain pattern, such delays do not degrade the packet routing efficiency significantly, as shown in next section. We also find that the average number of hops is small in the tests. This shows that the extra costs on encountering evidence relay are acceptable in FaceChange.

Combining the above results, we conclude that FaceChange can enable nodes to collect encountering evidences efficiently with acceptable costs. Therefore, it can well support MOSN applications, which will be demonstrated in next section by taking the packet routing as an example.

C. Influence on Packet Routing

We also evaluated the efficiency of PROPHET under FaceChange. In the test, 15,000 packets were generated with randomly selected sources and destinations. Since encountering evidence may not arrive at a node sequentially following their creation times, we cache each arrived evidence for a period of time (T_c) before processing it for packet routing. We varied T_c in this test to see its influence. We measured success rate and average delay in the test. The former refers to the percentage of successfully delivered packets and the latter refers to the average delay of these packets.

1) Success Rate: The Success rates of the two methods in the tests with the two traces are shown in Figure 6(a) and 7(a), respectively. We see that FaceChange has higher success rate than PROPHET for most of T_c values in tests with both traces. This is because in PROPHET, the meeting probability is updated immediately after an encountering happens, which may cause it deviate from the average value due to a burst on meeting nodes, leading to inaccurate packet forwarding. FaceChange has a delay in handing the encountering evidences, so it can calculate the meeting probability more fairly. Such a result demonstrates that FaceChange does not degrade the success rate of packet routing in MOSNs.

We also find that when T_c further grows, the success rate of FaceChange decreases in the test with the Haggle trace. This is because when T_c is very large, the meeting probabilities are not updated quickly enough to reflect the changes on meeting frequencies among nodes, leading to inaccurate guidance on packet routing and degraded success rate.

2) Average Delay: The average delays of the two methods in the tests with the two traces are shown in Figure 6(b) and 7(b), respectively. We find that FaceChange has smaller delay than PROPHET, which is caused by the same reasons as explained in the previous subsection.

Combining the above results, we conclude that FaceChange can support packet routing in MOSNs efficiently.

D. Energy Consumption

To evaluate the energy consumption of FaceChange, we conducted experiments on two Windows Phones: HTC Surround and LG Quantum. We let the two phones communicate with a server through WiFi connection to realize the key components in FaceChange, i.e., blind policy checking and packet/encountering evidence relaying. We did not include the energy cost of bilinear pairing since it has been proven to be acceptable in a previous literature [11].

All phones were restored to factory setting and were fully charged before each test. We measured the energy consumption as the percentage of remaining battery level after certain rounds of encountering. In blind policy checking, we assume each phone has 5 type-based attributes and 5 value-based attributes. In packet and encountering evidence relaying, we assume a phone exchanges N_p packets and N_e evidences in each encountering. N_p and N_e were randomly obtained from [100, 300]. Such a setting matches the situation in the real trace. We measured the percentage of remaining battery level after every 50 encounters. Each test was run for 10 times. The test results are shown in Figure 5.

We see from the figure that 50 encounters consume roughly about 1% of total battery for current smartphones. Such a result shows that the extra energy consumption incurred by FaceChange is acceptable for modern devices. We again examined the real traces and found that each person (node) has 117 and 340 encounters every day on average in the MIT Reality trace and the Haggle trace, respectively. Combining with the results in the table, we can see that FaceChange only consumes less than 6% of total battery daily even in the crowd conference scenario. This further demonstrates the applicability of FaceChange in real applications.

VI. Conclusion

In this paper, we propose FaceChange, a system that supports both neighbor anonymity and real ID based encountering information collection in MOSNs. In FaceChange, each node continually changes its pseudonyms and parameters when communicating with neighbors nodes to hide its real ID. Encountering evidences are then created to enable nodes to collect the real ID based encountering information. After two encountering nodes disconnect, the encountering evidence is relayed to the encountering node through a selected relay node. Advanced techniques are adopted in these steps to ensure the safety and efficiency of the encountering evidence collection. Fine-grained control over what information can be included in the encountering evidence is also supported in FaceChange. Extensive analysis and experiments are conducted to prove the effectiveness and energy efficiency of FaceChange in protecting node privacy and supporting the encountering information collection in MOSNs. In the future, we plan to integrate social factors, e.g., trustable communities, to facilitate privacy protection in FaceChange.

Acknowledgements

This research was supported in part by U.S. NSF grants NSF-1404981, IIS-1354123, CNS-1254006, CNS-1249603, Microsoft Research Faculty Fellowship 8300751. We would like to thank reviewers and the shepherd for our paper. Their insights and suggestions have helped us improve the quality of the paper significantly.

References

[1] S. Jain, K. R. Fall, and R. K. Patra, “Routing in a delay tolerant network,” in *Proc. of SIGCOMM*, 2004.

[2] J. Wu, M. Xiao, and L. Huang, “Homing spread: Community home-based multi-copy routing in mobile social network,” in *Proc. of INFOCOM*, 2013.

[3] T. Ning, Z. Yang, H. Wu, and Z. Ha, “Self-interest-driven incentives for ad dissemination in autonomous mobile social networks,” in *Proc. of INFOCOM*, 2013.

[4] A. Balasubramanian, B. N. Levine, and A. Venkataramani, “DTN routing as a resource allocation problem.” in *Proc. of SIGCOMM*, 2007.

[5] P. Hui, E. Yoneki, S.-Y. Chan, and J. Crowcroft, “Distributed community detection in delay tolerant networks,” in *Proc. of MobiArch*, 2007.

[6] K. Chen and H. Shen, “SMART: Lightweight distributed social map based routing in delay tolerant networks.” in *Proc. of ICNP*, 2012.

[7] K. Chen, H. Shen, and H. Zhang, “Leveraging social networks for p2p content-based file sharing in disconnected manets.” *IEEE TMC*, 2014.

[8] F. Li and J. Wu, “MOPS: Providing content-based service in disruption-tolerant networks,” in *Proc. of ICDCS*, 2009.

[9] M. Motani, V. Srinivasan, and P. Nuggehalli, “PeopleNet: engineering a wireless virtual social network.” in *Proc. of MOBICOM*, 2005.

[10] G. Costantino, F. Martinelli, and P. Santi, “Privacy-preserving interest-casting in opportunistic networks.” in *Proc. of WCNC*, 2012.

[11] R. Lu, X. Lin, T. H. Luan, X. Liang, X. Li, L. Chen, and X. Shen, “Prefilter: An efficient privacy-preserving relay filtering scheme for delay tolerant networks.” in *Prof. of INFOCOM*, 2012.

[12] L. Guo, C. Zhang, H. Yue, and Y. Fang, “A privacy-preserving social-assisted mobile content dissemination scheme in DTNs.” in *Proc. of INFOCOM*, 2013.

[13] X. Lin, R. Lu, X. Liang, and X. Shen, “STAP: A social-tier-assisted packet forwarding protocol for achieving receiver-location privacy preservation in vanets.” in *Proc. of INFOCOM*, 2011.

[14] M. Li, N. Cao, S. Yu, and W. Lou, “FindU: Privacy-preserving personal profile matching in mobile social networks.” in *Proc. of INFOCOM*, 2011.

[15] R. Zhang, Y. Zhang, J. Sun, and G. Yan, “Fine-grained private matching for proximity-based mobile social networking.” in *Proc. of INFOCOM*, 2012.

[16] X. Liang, X. Li, K. Zhang, R. Lu, X. Lin, and X. Shen, “Fully anonymous profile matching in mobile social networks.” *IEEE JSAC*, vol. 31, pp. 641–655, 2013.

[17] R. Lu, X. Lin, Z. Shi, B. Cao, and X. S. Shen, “IPAD: An incentive and privacy-aware data dissemination scheme in opportunistic networks.” in *Proc. of INFOCOM*, 2013.

[18] A. Kate, G. M. Zaverucha, and U. Hengartner, “Anonymity and security in delay tolerant networks.” in *Proc. of SecureComm*, 2007.

[19] M. K. F. Dan Boneh, “Identity-based encryption from the weil pairing advances in cryptolog.” in *Proc. of CRYPTO*, 2001.

[20] M. Kantarcioglu and C. Clifton, “Privacy-preserving distributed mining of association rules on horizontally partitioned data.” *IEEE TKDE*, vol. 16, no. 9, pp. 1026–1037, 2004.

[21] A. C. Yao, “Protocols for secure computations,” in *Proc. of FOCS*, Washington, DC, USA, 1982.

[22] E. M. Daly and M. Haahr, “Social network analysis for routing in disconnected delay-tolerant manets,” in *Proc. of MobiHoc*, 2007.

[23] P. Hui, J. Crowcroft, and E. Yoneki, “Bubble rap: social-based forwarding in delay tolerant networks,” in *Proc. of MobiHoc*, 2008.

[24] W. Gao and G. Cao, “On exploiting transient contact patterns for data forwarding in delay tolerant networks.” in *Proc. of ICNP*, 2010.

[25] X. Zhang and G. Cao, “Transient community detection and its application to data forwarding in delay tolerant networks.” in *Proc. of ICNP*, 2013.

[26] P. Aditya, V. Erdélyi, M. Lentz, E. Shi, B. Bhattacharjee, and P. Druschel, “Encore: Private, context-based communication for mobile social apps,” in *Proc. of MobiSys*, 2014.

[27] J. Manweiler, R. Scudellari, and L. P. Cox, “Smile: encounter-based trust for mobile social services,” in *Proc. of CCS*, 2009.

[28] L. P. Cox, A. Dalton, and V. Marupadi, “Smokescreen: flexible privacy controls for presence-sharing,” in *Proc. of MobiSys*, 2007.

[29] R. L. Rivest, A. Shamir, and L. M. Adleman, “A method for obtaining digital signatures and public-key cryptosystems.” *Commun. ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[30] C. Paar and J. Pelzl, *Understanding Cryptography - A Textbook for Students and Practitioners.* Springer, 2010.

[31] “Ubuntu network configuration,” https://help.ubuntu.com/community/NetworkConfigurationCommandLine/Automatic.

[32] F. Zhang, R. Safavi-Naini, and W. Susilo, “An efficient signature scheme from bilinear pairings and its applications.” in *Proc. of PKC*, 2004.

[33] A. Lindgren, A. Doria, and O. Schelén, “Probabilistic routing in intermittently connected networks.” *Mobi. Comp. and Comm. Rev.*, 2003.

[34] J. Golbeck, “Trust and nuanced profile similarity in online social networks,” *ACM Transactions on the Web (TWEB)*, 2009.

[35] N. Eagle, A. Pentland, and D. Lazer, “Inferring social network structure using mobile phone data,” *PNAS*, vol. 106, no. 36, 2009.

[36] A. Chaintreau, P. Hui, J. Scott, R. Gass, J. Crowcroft, and C. Diot, “Impact of human mobility on opportunistic forwarding algorithms,” in *Proc. of INFOCOM*, 2006.