Private and Flexible Urban Message Delivery

Shan Chang, Member, IEEE, Hongzi Zhu, Member, IEEE, Mianxiong Dong, Member, IEEE, Kaoru Ota, Member, IEEE, Xiaojing Liu, and Xuemin (Sherman) Shen, Fellow, IEEE.

Abstract—With the popularity of intelligent mobile devices, enormous urban information has been generated and required by the public. In response, ShanghaiGrid (SG) aims to providing abundant information services to the public. With fixed schedule and urban-wide coverage, an appealing service in SG is to provide free message delivery service to the public using buses, which allows mobile device users to send messages to locations of interest via buses. The main challenge in realizing this service is to provide efficient routing scheme with privacy preservation under highly dynamic urban traffic condition. In this paper, we present an innovative scheme BusCast to tackle this problem. In BusCast, buses can pick up and forward personal messages to their destination locations in a store-carry-forward fashion. For each message, BusCast conservatively associates a routing graph rather than a fixed routing path with the message in order to adapt the dynamic of urban traffic. Meanwhile, the privacy information about the user and the message destination is concealed from both intermediate relay buses and outside adversaries. Both rigorous privacy analysis and extensive trace-driven simulations demonstrate the efficacy of BusCast scheme.

Index Terms—Anonymous communication, backward unlinkability, message delivery, traffic analysis attacks, and vehicular networks.

I. INTRODUCTION

WTH the prosperity of powerful intelligent mobile devices, e.g., tablets and smart phones, urban sensing information, such as photos of events, audio and video records, has been largely enriched. The ever-increasing demands for sharing such information posed from the public have become a serious challenge. In response to the challenge, the Shanghai government has established the ShanghaiGrid (SG) project since 2005, with the ambitious goal of building a metropolitan-scale information service system. Among others, one promising application in SG is to provide message delivery service in which mobile users can send messages to some locations of interest, such as homes, workplaces and public agents. The goal of the application is three-fold. First, it should guarantee anonymous data communication for users, which hides the privacy information stating who is communicating with whom and for what purpose from others. For example, Alice may take a picture of an event and would like to send it to the police station as evidence. Certainly, she will not send her ID information with the picture or want her identity information being exposed by any means. Second, the end-to-end delivery may not have to be real-time but should be short. Last, the system should provide large service coverage to the public in terms of both geographical and temporal distributions.

To achieve the message delivery service, one possible solution is to use conventional cellular networks (e.g., GPRS and 3G) or satellite techniques, which can provide very short delivery delay. However, the privacy of mobile users in terms of identities and their interested location information is not well protected from the network operators. In addition, it also causes tremendous communication cost for data transmission. Recently, vehicular networks [1], [2], [3] have emerged as the new landscape of mobile ad hoc network, in which data communication is carried out in a store-carry-and-forward fashion. In SG, we consider to use buses (forming a bus network) for message delivery because of three major reasons. First, in urban settings, with the dense and wide distribution, commuting buses can reach a very high coverage. For example in Shanghai, with a communication range of 600 meters, the area covered by buses can be reached up to 90% of the downtown area. Second, the achievable end-to-end delay is convincing for most delay tolerant applications with fixed bus routes and schedules. Last, as buses are public vehicles, it is practical to provide such a service to the public without the concern of failures caused by selfish behavior.

To achieve efficient and anonymous message delivery with buses, however, is very challenging for three reasons. First, due to the dynamic surface traffic, buses may experience unexpected delays, which means the contacts between a pair of buses cannot be accurately predicted even with the fixed bus schedules. In this case, simply using a pre-determined shortest routing path calculated based on the static bus schedules is not feasible. Another naive solution can flood a message over the network, which can achieve shortest delivery delay but arouse prohibitive network traffic. Second, because of the requirement of anonymous communication, all identification information such as identities, locations, and routing paths must be removed from messages before being sent over open channels. Without the knowledge about the receiver, it is hard for an intermediate node to make an efficient routing decision. Last, even though identifying information can be well

Copyright (c) 2015 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

This research is supported by the National Natural Science Foundation of China (Grant Nos. 61300199, 61202375, 61472255, 61420106010), the Fundamental Research Funds for the Central Universities (2232014D3-21), the Innovation Program of Shanghai Municipal Education Commission (12ZZ060), the JSPS KAKENHI (Grant Nos. 26730056, 15K15976), and the JSPS A3 Foresight Program.

Shan Chang and Xiaojing Liu are with the School of Computer Science & Technology, Donghua University, Shanghai, 201620, P.R. China (e-mail: changshan, liuxq@dhu.edu.cn).

Hongzi Zhu is with the Department of Computer Science and Technology, Shanghai Jiao Tong University, Shanghai, 200000, P.R. China (e-mail: hongzi@cs.stju.edu.cn).

Mianxiong Dong and Kaoru Ota are with the Department of Information and Electronic Engineering, Muroran Institute of Technology, Japan (e-mail: mx.dong, k.ota@ieee.org).

Sherman Shen is with the Department of Electrical and Computer Engineering, University of Waterloo, Canada N2L 3G1 (e-mail: xshen@bcbu.uwaterloo.ca).

0018-9545 (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
protected, it is much harder to defend against traffic analysis
attacks, where adversaries observe the encrypted traffic flow
to infer the relationship between messages. In consequence,
messages can be traced forward to the destinations or back-
ward to the sources [5]. One possible solution is to use Onion
routing [6], in which a message encrypted by the sender can
be stripped off layer by layer by intermediate relay buses.
Each relay bus can only get the information of its next-hop
and forward the message accordingly. This method needs the
sender to determine the chain of all relay buses in advance.
In the dynamic scenario of bus network, however, it is hard
to accurately pre-determine an optimal routing path in terms
of minimizing the delivery delay. Another alternative solution
is to use universal re-encryption [7], [8] in which each relay
bus re-encrypts a message without knowing the identity of the
receiver. The solution needs to explicitly provide the identity
of the receiver to relay buses, which would disclose the privacy
of the receiver. As a result, there is no successful solution, to
the best of our knowledge, to provisioning efficient private
message delivery in bus networks.

In this paper, we propose an innovative message delivery
scheme in bus networks, called BusCast, which provides a
set of routing mechanisms flexible to the uncertainty of bus
contacts caused by dynamic surface traffic while provisioning
anonymous communication for users. BusCast elegantly in-
tegrates three key techniques. First, BusCast users can plan a
routing graph for a message, which is made up of a set of relay
rules indicating how the packet would transfer between bus
routes. Second, we design an Anonymous Routing Structure
(ARS) to indicate routing information for intermediate relay
buses by embedding the routing graph in the message. With
the ARS, relay buses can only recover their own routing
instructions about which bus routes are the next hop presented
in the routing paths. Last, each relay bus re-generates the
ARS, and the confidential messages of users are also re-
encrypted on each relay bus using a universal re-encryption
scheme, which requires no public key of message receivers.
Combining the two together eliminates the linkability between
incoming and outgoing packets, which can defend against
traffic analysis attacks. The strong point of BusCast design is
that even if the secrets of one or more bus routes are exposed,
anonymous communication can still be achieved. Thorough
privacy analysis shows that the BusCast design can protect
user privacy well. We also verify the routing performance of
BusCast though extensive trace-driven simulations that involve
199 bus routes in Shanghai city.

We highlight our main contributions in this paper as follows:

- • We have considered the dynamic of sufficient traffic in real-
izing message delivery service and design an ARS, which
conceals identification information from other buses and
outside adversaries and provides plenty of flexibility in
making routing decision.

- • We allow intermediate buses embedded in an ARS to re-
genenerate the ARS without the need to know the routing
graph, which makes buses act as both message routers
and mix proxies of a mix net [4] and therefore can defend
against traffic analysis attacks and guarantee anonymous
communications.

- • We have conducted both privacy, performance analysis
and extensive trace-driven simulations to demonstrate the
efficacy of BusCast design.

The remainder of this paper is organized as follows. Section
II introduces related work. In Section III, we characterize the
unique features of bus networks. Section IV describes the
system and attack models in bus networks and presents the
design goals. Section V present the overview of BusCast. We
elaborate the technique of flexible routing with routing graph
in Section VI. In Section VII, the details of privacy preserving
packet forwarding using Anonymous Routing Structure are
described. Section VIII presents the privacy and performance
analysis of BusCast. Several design issues that may be encoun-
tered in practice are discussed in Section IX. In Section X, we
conduct trace-driven simulations to evaluate the performance
of BusCast and present the results. Finally, we conclude and
outline the directions for future work in Section XI.

II. RELATED WORK

Since the concept of mix net was first introduced by Chaum
for anonymous communications in Internet [4], many studies
have followed Chaum’s approach such as Web-MIXes [9],
Tarzan [10], Mixminion [11], AOS [12]. A mix node is a
proxy batched modifies input messages and outputs them in
a random order, called mixing. In this way, it is hardly to
correlate a comes in message with a goes out message, which
can be leveraged to defend against the traffic analysis attacks.

Several mixing schemes are designed for providing anony-

mous routing in mobile networks, such as ANODR [13],
SDAR [14], AnonDSR [15]. These schemes share two com-
mon features. First, all these schemes demand a route discov-
ery phase before forwarding a packet, which enables the sender
to discover and establishes a secure routing path via a number
of intermediate wireless nodes to the receiver. Second, all these
schemes use the layer-by-layer en/decryptions. The significant
feature of these schemes is that the layers should be peeled
in sequence, and each time can only be peeled one layer.
However, the strictly defined routing path lack of flexibility
to adapt rapid changes of mobility of vehicular networks. Y.
Fan et al. [16] proposed a network coding scheme combining
with homomorphic encryption functions to protect the source
anonymity from traffic analysis and flow tracing attacks in
multi-hop wireless networks. They considered a multicast
network. Intermediate nodes buffer the received packets until
all the packets belonging to same session are available and
perform random linear coding on these packets. However,
the scheme cannot work well in vehicular networks, since
the opportunistic routing makes it hard to collect all packets
belonging to the same session at the same intermediate.

There are two works closely related to this paper, R. Jansen
et al. introduced a TPS scheme [17] to address the anonymity
issues in DTN. In the scheme, nodes are divided into several
groups. Senders generate a one-time secret key to encrypt each
message and the identities of receivers. Senders use secret
sharing scheme to divide the secret into pieces, and each piece
is encrypted using a group key. Each relay node decrypts a
piece of secret using the group key it holds. Node decrypting
the last piece of secret can recover the original secret key. Then the message and identity of receiver can be revealed and be routed to receiver. However, this scheme leaks identities of receivers to a part of untrusted intermediate nodes. R. Lu et al. put forward a social-based privacy-preserving packet forwarding protocol in VANETs [8]. The authors deployed RSUs at high social degree intersections to assist in packet forwarding between vehicles by temporarily storing packet, and to carry out re-encryption on the packet in order to construct mix network. However, locations of receivers are provided to all relay RSUs and vehicles. Furthermore, in order to protect the privacy, each packet should be at least temporarily stored on and re-encrypted by an RSU.

III. CHARACTERISTICS OF BUS NETWORK

Understanding the properties of the network composed of urban buses is essential to the performance of message delivery. In this section, we study the key features of bus network based on a real Global Positioning System (GPS) trace collected from 2,358 buses on 199 bus lines between Feb. 19th and Mar. 5th, 2007. It covers the downtown area of Shanghai city of about 120 square kilometers.

We first illustrate the geographical distribution of all bus lines in Fig. 1, where the red lines represent the aggregated itineraries of all bus lines. It can be seen that bus lines show a dense and relatively uniform distribution throughout the whole region. We refer to the coverage of bus network as the area that messages can be delivered by a bus of the network, using short-range wireless communication (e.g., Dedicated Short Range Communications: DSRC, i.e., 802.11p).

Fig. 2 shows the coverage as the function of wireless communication range. It can be seen that, when communication range is above 600m, the coverage ratio can reach over 90% of the total area. Besides, it can also be seen that bus lines are well inter-connected with different bus lines sharing part of their routes in common. Both properties indicate that bus network is ideal for message delivery.

Ideally, buses travel between stops on their routes on time no matter how the surface traffic changes during the day. In this case, a simple yet effective way to deliver a message is to forward the message along the shortest path calculated based on the fixed bus routes and regular schedules. In reality, however, the mobility of buses varies dramatically at different time, especially in metropolises like Shanghai. For example, Fig. 3 shows the cumulative distribution function of the traverse time for a bus to travel from one terminal to the other on its route over all buses during the peak and off-peak hours in a day, respectively. It can be seen that 80% buses can finish one-way traversing within 45 minutes during the off-peak hours but the ratio drops dramatically to about 20% during the peak hours. The huge variation makes pre-determining the shortest routing path very hard, if not impossible.

IV. MODELS AND DESIGN GOALS

A. System Model

In BusCast, there are two components: buses and Trust Authority (TA).

- **Buses**: are equipped with On-Board Units (OBUs), which typically consist of a CPU, a large storage, a GPS module and wireless communication modules. Moving buses can talk with other buses via short-range wireless communication (e.g., DSRC) and with the TA via long-range wireless communication (e.g., GPRS, 3G).

- **Trust Authority**: is a trustworthy authority, which can communicate with buses all the time. TA generates common secret information for buses on the same route.

We assume that buses carry out their functionality properly but may cause secret information generated by TA leaked under intrusion of adversaries. We assume each bus has an Intrusion Detection System (IDS). If an intrusion is detected, it reports to TA to take corresponding security responses. We also assume that routes information is available to the public (e.g., from the website).

B. Attack Model

We characterize adversaries from four perspectives.

First, adversaries can mount both passive and active attacks which implies not only eavesdropping but also packets injection and modification on the wireless channel.

Second, adversaries can have the global view of the whole network traffic by eavesdropping on the open channel.

Third, adversaries behave rationally which means they launch attacks in order to gain benefits. The goal of adversaries is to jeopardize use’s privacy. Particularly, we consider privacy jeopardizing attacks in two aspects:
Confidentiality violation: attackers eavesdrop on the shared medium to catch others communications and recover the content of packets in order to obtain the confidential information.

Anonymity violation: even the content of packets is protected by encryptions, attackers can also obtain sensitive information related to identities, locations of victims, by launching traffic analysis or packet marker attacks. In traffic analysis attacks, adversaries intercept and examine encrypted messages in order to deduce information from patterns in communication. In packet marker attacks, adversaries insert some distinguishable markers into packets in order to track them.

Last, although intrusions of attackers can be detected by IDSs, we do not assume that the damage results (secret exposure) can be avoided perfectly. Adversaries can make use of the exposed secrets, trying to recover other sensitive information. Additionally, we assume that the invaded routes should be a small fraction of all the bus routes.

C. Design Goals

The BusCast design should meet both privacy-related and routing performance-related requirements.

1) Privacy Requirements

We aim to enable anonymous message delivering. More specifically, the following properties should be guaranteed.

• Unlinkability between users and destinations should be guaranteed, which means that it is not possible to trace who communicates with whom.

• Backward unlinkability between users and destinations should be guaranteed, which means that even after some buses are invaded and consequently get secret exposed, past communications remain untraceable.

• Confidentiality of the messages should be protected. Sensitive information shared among senders and receivers will never be disclosed to others.

2) Routing Performance Requirements

Since the number of potential users may be very large, it is of great importance to consider the scalability of the system. The BusCast design should minimize the message delivery delay and the corresponding network cost aroused.

V. OVERVIEW OF BUSCAST DESIGN

To tackle the challenges in realizing message delivery service, BusCast elegantly integrates three techniques: flexible routing with routing graph, constructing ARS, and regenerating ARS and re-encrypting user data (to form mix net).

Specifically, BusCast uses a routing graph instead of one single shortest path to forward a message, in which each edge indicates how the packet would transfer between bus routes. By elaborately constructing the routing graph for the message, plenty of flexibility can be achieved to adapt the uncertainty of bus mobility caused by dynamic suffice traffic. Given a routing graph of the message, we construct and associate an ARS with the message to indicate routing information for intermediate relay buses by associating the routing graph with the message. With the ARS, relay buses can only recover their own routing instructions about which bus routes are the next hop presented in the routing paths. To break the linkability between incoming and outgoing packets, each relay bus re-generates the ARS without the need to know what the routing graph is. Furthermore, the confidential messages of users are also re-encrypted on each relay bus using a universal re-encryption scheme, which requires no public key of message receivers. Without linkability during the routing process, BusCast can defend against traffic analysis attacks.

VI. FLEXIBLE ROUTING WITH ROUTING GRAPH

A. Contact Graph of Buses

Since buses on the same route share the same itinerary, we consider message delivery problem at the level of bus routes. Two bus routes are referred to as neighbors if they share partial itinerary or have intersections.

Suppose that a mobile device user Bob wants to send a message m to his friend Alice using bus networks, and Bob knows her location L. Bob forwards m to a nearby passing bus, then m will be relayed between buses and finally reach to a bus route passing through L. Fig. 4 illustrates an example of a message delivery using bus networks. The different colors of solid lines denote different route paths, and the number in each circle represents the route ID.

We use a contact graph G = (V, E, W) to represent contacts between bus routes. A contact graph consists of a set of vertices V, a set of edges E, and a set of weights W each of which is assigned to an edge of the graph. Since route paths are public information, it’s convenient to determine V and E using neighbor relationship between routes:

• Each vertex v_i ∈ V denotes a bus route (line).

• If two bus lines v_i, v_j ∈ V are neighbors, there are two edges e(v_i, v_j), e(v_j, v_i) ∈ E between v_i and v_j. (Edges between two vertices come in pairs, since neighborhood is bidirectional.)

Fig. 5a shows the contact graph G of the bus network in Fig. 4. According to the static and dynamic features of buses, w_i,j ∈ W can be set in two ways. Assume that the length of bus line v_i is l_i, there is an edge e(v_i, v_j) ∈ E, and v_i and v_j share x intersections and several road segments. The total length of share road segments is y. The communication range between buses is r. According to the above static geographical feature of v_i and v_j, w_i,j = 1/(x + 1) + y/r, where 1/(x + 1) is an approximation of overlap ratio of v_i and v_j, and it is proportional to the contact opportunities between v_i and v_j.
An example of routing graph

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2015.2466651, IEEE Transactions on Vehicular Technology

Hence \( w_{i,j} \) is proportional to their inter-contact time. A more realistic (dynamic) way to determine \( w_{i,j} \) is using inter-contact time of routes directly, which denotes the average time elapsed between two successive contacts of a certain bus on \( v_i \) and any one buses on route \( v_j \).

B. Extracting Routing Graph

In order to deliver \( m \) to Alice effectively, Bob should indicate (implicit or explicit) location (or identity) information of Alice to relay buses. Basically, there are two kinds of routing strategies. First, Bob indicates \( L \) explicitly in the packet. However, it is prohibited for privacy concerns. Second, Bob plans a routing path using \( G \) of the bus network in advance (called deterministic routing), and uses particular privacy preserving techniques, such as Onion routing, making each relay bus only knows a small part of the fix routing path. However, deterministic routing, such as computing a shortest path, usually does not work well due to large variation of contact time and locations between buses, which can tremendously deviate the values predicted using bus schedules.

Consequently, we introduce flexibility in deterministic routing. First, senders designate a number of relay instructions between neighbor bus lines for each packet rather than a fixed routing path, according to \( G \) and performance requirements. Relay instructions form a routing graph \( G' = (V', E', W') \), which is an induced subgraph of \( G \), i.e., \( V' \subseteq V \), \( E' \subseteq E \). Directed edges \( \hat{e}(v_i, v_j) \in E' \) denotes a relay rule. \( G' \) contains at least one source vertex \( v_s : \forall v_i \in V' \), \( \hat{e}(v_s, v_i) \in E' \) and one destination vertex \( v_d : \forall v_i \in V' \), \( \hat{e}(v_d, v_i) \in E' \). For the example in Fig. 4, a user located near bus line 4 and 7. The user wants to send a packet to a place located near bus line 23. Fig. 5b gives an example of \( G' \). A simple method to extract \( G' \) is to pick k-shortest paths from \( v_s \) to \( v_d \) on \( G \) (removing duplicate edges). Sophisticated methods can be used if more information is available, e.g., schedule of buses.

During message delivering, senders deliver a packet to a passing-by bus of route \( v_s \), then the bus carries the packet until encountering \( v_i \) that \( \hat{e}(v_s, v_i) \in E' \), and forwards the packet to \( v_i \). In the same way, the packet is relayed by buses in a route-by-route manner according to \( G' \) until reaching to \( v_d \). Then the packet is carried by \( v_d \) and broadcasted in the vicinity of destination \( L \).

VII. PRIVACY PRESERVING PACKET FORWARDING USING ANONYMOUS ROUTING STRUCTURE

A. System Initialization

Given a contact graph \( G \), each edge \( \hat{e}(v_i, v_j) \in E \) is mapped to a routing instruction, referring to a Relay Indicator (RI), denoted as \( I_{r(i,j)} \), to indicate one-hop relay from \( v_i \) to \( v_j \). We also define Broadcast Indicators (BI), denoted as \( I_{b(s,i)} \), which are used to indicate broadcasting areas for the buses.

Each indicator (BI or RI) has two parts: a public Indicator \( (P) \) and a secret Indicator \( (S) \). Public indicators are known to all entities, and are used for constructing ARSs. Secret indicators are kept by certain routes secretly for verifying ARSs. It is similar with public/private key pairs, however indicators serve to build ARS rather than encrypt messages.

TA is responsible for constructing all the indicators in the system. In addition, TA generates the public parameters and master-key of the system.

1) Generating Public Indicators

For each route \( R_i \), TA assigns a set of public RIs for \( R_i \) corresponding to edges \( \hat{e}(v_i, v_j) \in E \) in \( G \). In other words, each RI relates to a neighbor of \( R_i \). For its neighbor \( R_j \), the corresponding public RI is \( I_{p_{fly}(R_i, R_j)} = R_i|R_j \), which indicates one-hop relay from \( R_i \) to \( R_j \). The symbol \( | \) represents concatenation between strings.

For example in Fig. 4, \( R_7 \) has three neighbors which are \( R_4, R_6 \) and \( R_{12} \). TA assigns three public RIs \( I_{p_{fly}(R_i, R_j)} \) and \( I_{p_{fly}(R_i, R_j)} \) for \( R_i \). It should be noted that neighbors do not share RIs. \( R_4 \) and \( R_7 \) hold \( I_{p_{fly}(R_i, R_j)} = R_7|R_4 \) and \( I_{p_{fly}(R_i, R_j)} = R_4|R_7 \), respectively.

TA also assigns a set of BIs for \( R_i \) representing certain areas where it should broadcast packets. We divide the urban area into small cells according to the communication range of buses. Each cell has an unique identity. Cells within the coverage area of \( R_i \) are organized in sequence of locations from its departure to terminal station. Assume that \( n \) cells are in the coverage area of \( R_i \), which are \( \{c_1, c_2, ..., c_n\} \). TA executes the following operations to construct a binary tree of public BIs of \( R_i \), so that a BI can be targeted efficiently. The height of the BI tree is \( \lceil \log n \rceil + 1 \):

- Root node of the tree is \( I_{p_{bst}(R_i)} = R_i \), which is used to indicate whether \( R_i \) is one of the last relay routes.
- Split the original set of cells into \( \{c_1, c_2, ..., c_{n/2}\} \) and \( \{c_{n/2}+1, c_{n/2}+2, ..., c_n\} \). The first layer public BIs are \( R_i||c_1||c_{n/2}||c_{n/2}+1||...\) and \( R_i||c_{n/2}||c_{n/2}+1||...\), denoted as \( I_{p_{b}(R_i, 0)} \) and \( I_{p_{b}(R_i, 1)} \).
- Execute the split recursively on two resulting sets, and get the corresponding public BIs until only one cell left in each set.

For example in Fig. 4, assume \( R_7 \) covers 7 cells which are \( \{c_1, c_2, ..., c_7\} \). The four layer public BI tree of \( R_7 \) is illustrated in Fig. 6. \( R_i||c_1||c_{n/2}||c_{n/2}+1||...c_n \) indicates that the destination cell belongs to the set of \( \{c_j, ..., c_k\} \).

2) Generating Secret Indicators

For a given public indicator \( I_P \in \{0, 1\}^* \) (either \( I_{p_{fly}} \) or \( I_{p_{bst}} \)), TA computes \( Q_I = H_1(I_P) \in \mathbb{G}_1^\ast \), and sets the corre-
spending secret indicator as $s = s(Q_1 + P_1)$. For example, given $I_{\text{bl}(R, R_a)} = R_2 || R_6$, TA generates $I_{\text{bl}(R, R_b)} = s(H_1(R_2 || R_6) + P_1)$; for $I_{\text{bl}(R_0, R_0)} = R_7 || c_2$, corresponding secret BI $s(H_1(R_7 || c_2) + P_1)$. TA distributes secret indicators to the bus routes belonging to, via a secure channel (e.g., secret indicators are signed and encrypted using a signcryption scheme [18], and transmitted using 3G network).

3) Generating System Parameters

Given a security parameter $n \in Z^+$, TA runs the following algorithms to generate the public parameters and master-key in the system.

a. Generate a large prime $q$, an additive cyclic group $G_1$ and a multiplicative cyclic group $G_2$ of order $q$ and a bilinear mapping $e : G_1 \times G_1 \rightarrow G_2$ [19]. Choose a random generator $P \in G_1$.

b. Pick $s \leftarrow Z_q^*$ as master-key, set $P_{\text{pub}} = sP$.

c. Pick $P_1 \leftarrow G_1$.

d. Choose cryptographic hash functions:

\[
H : \{0, 1\}^* \rightarrow Z_q^*, \quad H_1 : \{0, 1\}^* \rightarrow G_1^*, \\
H_2 : G_2 \rightarrow \{0, 1\}^*, \quad H_3 : \{0, 1\}^* \times \{0, 1\}^* \rightarrow Z_q^*
\]

e. Set $\text{Par}_A = \{q, G_1, G_2, e, P, P_1, P_{\text{pub}}, H, H_1, H_2, H_3\}$.

f. Pick another large prime $p$ that discrete logarithm problems are difficult on $Z_p^*$, and $g \in Z_p^*$ which is a primitive root of $Z_p^*$. Set $\text{Par}_E = \{p, g\}$.

Then TA publishes all public indicators, $\text{Par}_A$ and $\text{Par}_E$ while keeping the master-key $s$ secret.

B. Operations Conducted by Message Senders

Recall the example of Bob and Alice in Section VI. After extracting the routing graph $G'$ of $m$ according to $G$, current location and $L$, Bob encrypts $m$ and constructs ARS for $m$.

1) Encrypting User Data $m$

Bob uses the universal re-encryption scheme proposed by P. Golle et al. [7] to encrypt $m$. The scheme has the feature that intermediate nodes can re-encrypt $m$ without knowing the identity or public key of the receiver. Specifically, Bob uses Alice’s public key $pk_A = g^{sk_A}$ ($sk_A \in Z_p$) to encrypt $m$, so that only Alice who holds the secret key $sk_A$ can recover $m$. The ciphertext $M$ has two parts. The second part is used for the future re-encryptions. The encryption process proceeds as follows:

a. Pick a pair of random encryption factors $(\tau_1, \tau_2) \in Z_p^*$.

b. Compute $M = \{(m \cdot pk_A^{\tau_1}, g^{\tau_1}); (pk_A^{\tau_2}, g^{\tau_2})\}$.

2) Constructing ARS

Assume the routing graph $G'$ includes $\mu$ routes $\{R_1, R_2, ..., R_\mu\}$, $R_\mu$ is the last relay route which covers $n$ cells $\{c_1, c_2, ..., c_n\}$, and $L$ located in cell $c_\rho$ ($1 \leq \rho \leq n$). Then Bob imbeds relay and broadcast indicators into ARS, according to edges of $G'$ and $c_\rho$. Specifically, Bob carries out the following steps to pick public RIs and BIs.

a. Choose public relay indicators: if $e(v_{R_i}, v_{R_j}) \in G'$, $1 \leq i, j < \mu$, then pick $I_{\text{bl}(R_i, R_j)} = R_\mu || c_\rho$.

b. Choose public broadcast indicators:

- Pick all public BIs on the path from $R_\mu || c_\rho$ to the root of BI tree, i.e.,

\[
\{I_{\text{bl}(R_\mu || c_\rho, i, \rho)} : 0 < i < \mu \} = \{I_{\text{bl}(R_\mu || c_\rho, 1, \rho)} : 1 \leq \rho \leq n \}
\]

For the example in Fig. 6, if $L$ is in the cell $c_3$, then $I_{\text{bl}(R_0, R_0)} = I_{\text{bl}(R_6, R_6)}$, and $I_{\text{bl}(R_7, R_0)}$ (gray circles) will be selected as public BIs.

Assume Bob has chosen $t$ public indicators $I_t, (1 \leq i \leq t)$ (both RIs and BIs), he generates the ARS as follows:

a. For each $I_t$, compute $x_i = H(I_t)$ and $Q_i = H_1(I_t)$.

b. Compute

\[
\ell_i(x_j) = \prod_{1 \leq j \neq i \leq t} \frac{x - x_j}{x_i - x_j} = a_{i,1} + a_{i,2}x + ... + a_{i,t}x^{t-1}
\]

where $a_{i,1}, a_{i,2}, ..., a_{i,t} \in Z_q$. Then

\[
\ell_i(x_j) = \begin{cases} 
1, & \text{if } i = j \\
0, & \text{if } i \neq j
\end{cases}
\]

c. Set $\alpha_i = [a_{i,1}, a_{i,2}, ..., a_{i,t}]^T$. Compute the routing information

\[
\begin{bmatrix}
A_1 \\
A_2 \\
\vdots \\
A_t
\end{bmatrix} = [a_{1,1}, a_{2,2}, ..., a_{t,t}] \times [Q_1, Q_2, ..., Q_t]^T
\]

\[
\begin{bmatrix}
a_{1,1}Q_1 + a_{2,1}Q_2 + ... + a_{t,1}Q_t \\
a_{1,2}Q_1 + a_{2,2}Q_2 + ... + a_{t,2}Q_t \\
\vdots \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
a_{1,1}Q_1 + a_{2,1}Q_2 + ... + a_{t,1}Q_t \\
a_{1,2}Q_1 + a_{2,2}Q_2 + ... + a_{t,2}Q_t \\
\vdots \\
\end{bmatrix}
\]

d. Pick a string $\sigma \leftarrow \{0, 1\}^*$, and set $\alpha = H_3(\sigma, M)$.

e. Pick an integer $\beta \leftarrow Z_q^*$, and set $y = (\beta - \alpha) \pmod q$.

f. Compute $y \cdot [A_1, A_2, ..., A_t] = [B_1, B_2, ..., B_t] \in \mathbb{B}$.

g. Set ARS = $\{\mathbb{B}, \alpha, P_{\text{pub}}, \sigma \oplus H_2(e(P_{\text{pub}}, P_t)^y)\}$.

Then Bob sends message $C = \langle ARS, M \rangle$ and forwards $C$ to one source route $v_r$ in $G'$ directly.

C. Operations Conducted by Buses

When a bus on $R_i$ receives message $C$, it verifies whether $R_i$ was selected as a relay route. If so, the bus re-constructs $C$, and relays the updated $C$ according to the indicators.
1) Verifying ARS

The bus on $R_i$ uses its public/secret indicator pairs $(P, I^*)$ to verify which indicators were imbedded in the ARS. It checks the ARS as follows:

a. Verify if the root of $R_i$’s public BI tree, $I_{bst(R_i)}^P$, was imbedded in ARS, which means $R_i$ is the last relay route of $C$. If so, go to step 2. Otherwise, go to step 3.

b. Verify if its second layer BIs $\{I_{bst(R_i,0)}^P, I_{bst(R_i,1)}^P\}$ were imbedded in ARS. One of them is included in ARS means that $C$ should be broadcasted within the corresponding region. If one of them was in ARS, $R_i$ further verifies if any of its two children BIs was imbedded in ARS, and so on, until reaching to the highest layer BIs. The highest layer BI imbedded in ARS indicates final broadcasting cell.

c. Verify relay indicators of $R_i$ to see if some of them were imbedded in ARS, which means $R_i$ is one of intermediate relay routes, and $C$ should be sent to one of the routes included in the corresponding RIs.

Given an indicator pair $(P, I^*)$, the verification can be done using Algorithm 1 (Table 1).

If all verifications return false, it means that $R_i$ is not in $G'$. Hence, $R_i$ simply ignores it. Otherwise, $R_i$ should be either an intermediate or one of the last relay route.

2) Re-generation of ARS and $M$

If $R_i$ is an intermediate relay route, the bus on $R_i$ re-generates the ARS and $M$ in order to build the mix net, defending against traffic analysis attacks. Then $R_i$ prepares $C = <ARS, M>$ for next relay. If $R_i$ is the last relay route, (i.e., $R_p$) the bus simply discards the ARS and re-encrypts $M$. Then $R_i$ broadcasts $M$ in cell $c_p$ calculated before. The re-encryption of $M$ proceeds as follows:

a. Choose a random re-encryption factor $(\tau_1', \tau_2') \in Z_p^2$.

b. Compute

$$M' = \left\{ (m \cdot pk_A^{\tau_1'}, (pk_A^{\tau_2'}) \cdot g, g^{\gamma_1}, (g^{\gamma_2})^{\tau_2'}); ((pk_A^{\tau_1})^{\tau_2}, (g^{\gamma_2})^{\tau_2'}) \right\} = \left\{ (\lambda_0, \rho_0); (\lambda_1, \rho_1) \right\}$$

The re-generation of the ARS proceeds as follows:

a. Pick a string $\sigma' \leftarrow \{0, 1\}^\omega$, and set $\alpha' = H_3(\sigma', M')$.

b. Pick an integer $\beta' \leftarrow Z_q$.

Note $R_{ID_i}$ can obtain the value of $\alpha$ from Algorithm 1. Set $y' = ((\beta')^{-1} \cdot \alpha^{-1} \cdot \alpha') \mod q$.

Fig. 7 illustrates the whole procedure data forwarding using $ARS$, including the encryption of user data, construction of $ARS$, verification of $ARS$, re-generation of $ARS$ and ciphertext $M$, and decryption of user data.

VIII. ANALYSIS

A. Privacy Analysis

We first analyze abilities of different entities on understanding packets, which facilitates privacy analysis of the proposed scheme. First, for buses in the $G'$ of a message $C(C')$, after receiving the message, they can verify that some of their indicators are imbedded in ARS. For the intermediate and the last relay routes in $G'$, they can get one-hop routing instructions and broadcasting region of the packet, respectively. Each route in $G'$ can only reveal indicators of their own, no one has the knowledge of a whole routing graph. Second, for buses which are not in $G'$, after receiving $C(C')$, they verify the ARS in $C(C')$ and all the verification fails. Hence they only know that they are not in $G'$. Third, because of the hop-by-hop re-construction of $C$, neither inside nor outside

---

TABLE I: ALGORITHM 1

<table>
<thead>
<tr>
<th>Algorithm 1: Verify($P, I^*, ARS, M$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) Verify if $I^* \in ARS$.</td>
</tr>
<tr>
<td>2) Generate $\alpha = H_3(P, I^*)$.</td>
</tr>
<tr>
<td>3) Check if $\alpha \in c$.</td>
</tr>
</tbody>
</table>

---

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TVT.2015.2466651, IEEE Transactions on Vehicular Technology.

0018-9545 (c) 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.
observers can link $C$ to $C'$. No one has global view on the routing path of $C$. Although adversaries have global view of traffic in bus networks, they can hardly induce the travelling path of single packet. Hence the mix net is constructed from both the viewpoint of outside and inside observers. Last, $m$ is encrypted by Alice’s public key, none of them can decrypt $M(M')$ except Alice. Since the confidentiality can be achieved obviously, we focus on analyzing the unlinkability and backward unlinkability of the proposed method.

1) Unlinkability

Adversaries cannot launch traffic analysis attacks successfully by monitoring communication channel for two reasons. First, buses serving as mix proxies re-construct all communications in each hop. Content relevance between incoming and outgoing packets on buses is wiped off. Hence adversaries cannot link different packets by their contents. Second, buses pick and forward packets in a store-carry-forward fashion. Buses receive packets continuously, thus a number of packets are stored waiting for transmitting. Once a relay opportunity arises or a bus approaches to broadcasting region, packets under the same indicator stored on the bus will be relayed all together. In this way, the spatial and temporal correlations between incoming and outgoing packets on buses are eliminated.

Adversaries cannot launch packet marker attacks to track packets. In such attacks, an adversary eavesdrops packet $C$, and replaces $\mathcal{M}$ with an encrypted marker $\mathcal{N}$ that can be decrypted by it. However, Relay buses can verify that the ARS is not be generated for $\mathcal{M}$ (Since $\alpha^i = H_3(\alpha, N)$, however $\alpha^i P \neq \alpha P$). Then $\mathcal{N}$ will be dropped.

2) Backward Unlinkability

If an adversary invades a bus in $\mathcal{G}'$ and gets packet $C$ recorded by it. By verifying ARS, it can only recover one-hop routing information. Moreover, any relay bus re-encrypting $\mathcal{M}$ does not know the identity of Alice, which implies that the adversary cannot get any information of Alice by revisiting $\mathcal{M}$. Hence backward unlinkability can be achieved.

B. Performance Analysis of Privacy Preserving Mechanisms

The proposed scheme contains operations related to ARS and $m$. The performance of the re-encryption scheme on $m$ has been analyzed in [7]. Hence we focus on the performance of ARS operations. Table II summarizes the notations used.

Since $\mathcal{G}$ and cells are static for a given bus network. All RIs and BIs over the network are determined. Hence, senders can conduct one-time computations on $x_i = H(I_{i}^p)$ and $Q_1 = H_1(I_{i}^p)$ for all indicators $I_{i}^p$ in advance. $(x_i, Q_i)$ can be used on all ARS constructions in future. For this reason, the generation of $(x_i, Q_i)$ was not counted in the computation cost. Assume $t$ public indicators are imbedded in an ARS. Computational and storage complexities of the ARS are summarized in Table III.

<table>
<thead>
<tr>
<th>Notation</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$t$</td>
<td>The number of public indicators imbedded on ARS</td>
</tr>
<tr>
<td>$e$</td>
<td>Bilinear map</td>
</tr>
<tr>
<td>$\phi$</td>
<td>Bit length of elements in $G_1$</td>
</tr>
<tr>
<td>$\omega$</td>
<td>Bit length of strings outputted by $H_2(\cdot)$</td>
</tr>
<tr>
<td>$GA$</td>
<td>Addition over $\mathbb{G}_1$</td>
</tr>
<tr>
<td>$GM$</td>
<td>Multiplication over $\mathbb{G}_1$</td>
</tr>
<tr>
<td>$GE$</td>
<td>Exponentiation over $\mathbb{G}_2$</td>
</tr>
<tr>
<td>$EX$</td>
<td>Exponentiation over $\mathbb{Z}_q^\ast$</td>
</tr>
<tr>
<td>$MUL$</td>
<td>Multiplication over $\mathbb{Z}_q^\ast$</td>
</tr>
</tbody>
</table>

IX. DISCUSSION

A. Increasing the Number of Edges in Routing Graph

In BusCast, increasing the number of edges $t$ in $\mathcal{G}'$ implies more flexibility. As $t$ grows, the size of ARS is also linearly increasing which consumes more communication bandwidth. In vehicular scenario, where the wireless link quality is very dynamic, long messages may suffer failures. In BusCast, for $\phi = 160$ bits (with security comparable to 1024-bit RSA encryption), the length of ARS is $20\cdot(t+2)+\omega$ bytes. Simulation results show that a relatively small $t$ can significantly boost packet forwarding. Typical length of $t$ is several tens. Buses re-generate and verify ARSs without interactions with others. The operations related to ARS are not restricted by the connecting time between buses. Hence, we do not analyze the computation time of ARS in detail.

B. Buses Intrusion Countermeasure

Once TA receives an intrusion warning report from $R_i$, TA takes following responses to reset indicators of $R_i$. First, TA re-allocates a new ID for the invaded routes, and generates corresponding secret indicators using master-key $s$. Second, resets all indicators of the route $R_i$ using a secure channel. Third, notifies other buses and users for ID updating.
B. Effect of Routing Graph Size

We first examine the effects of the size of routing graph $G'$ to the delivery performance. For each packet, $G'$ is generated by picking $k$-shortest paths from the source bus to the destination on contact graph $G$. Duplicate edges between paths are removed. In this set of simulations, $G$ is constructed in a static way described in Section VI B. Unicast is used to forward packets. We change $k$ from 1 to 20 at an interval of one. In Fig. 8, the red line indicates that increasing $k$ will result in higher delivery ratio. The delivery ratio increases very fast when $k$ is smaller than 6, after that the growth becomes slow. It implies that a high cost performance ratio can be achieved when $k$ is 6. We also verify the compression ratio of edges in $G'$. Denote the number of the edges on path $i$ is $p_i$, and the number of edges in $G$ is $q$. The compression ratio of edges is defined as $1 - (q/\sum p_i)$. The blue line shows that the compression ratio larger that 40% even for very small $k$, and the compression ratio increases up to about 60% when $k$ larger than 8. Fig. 8 also plots the end-to-end delay as a function of the number of $k$ in routing graph. It is clear to see that as the number of paths increases the delivery delay dramatically drops. The average traffic per packet is 6 hops.

C. Performance Comparisons under Dynamic Traffic

In this experiment, we compare BusCast with other alternative schemes under dynamic traffic conditions, using both static and dynamic contact graph where the weight $w_{i,j}$ of edge $\hat{e}(v_i,v_j)$ are calculated in different ways.

- In static graph, according to the geographical feature of $v_i$ and $v_j$, $w_{i,j}$ is proportional to the contact opportunities between $v_i$ and $v_j$ (see detail in Subsection VI B).
- In dynamic contact graph, we use average inter-contact time of routes to determine $w_{i,j}$. We refer to inter-contact time as the time elapsed between two successive contacts of two bus routes. In order to obtain the inter-contact time of route $v_i$ and $v_j$, we first extract all contacts between buses from $v_i$ and $v_j$, respectively, and sort the contacts in terms of time, then the inter-contact time is computed at the end of each contact, as the time period between the end of this contact and the start of the next contact between the same two routes.

We change $k$ from 2 to 20 at an interval of two and conduct the experiments. Fig. 9 and Fig. 10 plot the end-to-end delay during off-peak and peak hours, respectively. It can be seen that BusCast can achieve very short delay comparing with Onion routing. Epidemic routing has the shortest delay due to flooding guarantees that the optimal path can always be found, however it also generates prohibitive network traffic. As $k$ increasing, the delay of BusCast is reduced. When $k$ reaches to 20, BusCast can achieve a very close delay comparing with epidemic routing. Meanwhile, BusCast only generate moderate traffic. For example, when $k = 6$, BusCast generates 9.3 hops traffic per packet when using dynamic contact graph during peak hours. It can also be seen that using dynamic contact graph is always better than using static graph especially when routing during peak hours.

XI. CONCLUSION AND FUTURE WORK

In this paper, we have developed a message delivery scheme BusCast which ensures both efficiency and users privacy in time insensitive scenarios. A flexible routing strategy is proposed to adapt highly dynamic changes of bus network topologies. A three-part privacy preserving mechanism is
introduced to ensure anonymous communications. We have demonstrated the efficacy of BusCast through rigorous analysis and extensive trace-driven simulations. For our future work, we intend to investigate different schemes of routing graph generation, as well as develop shortened ARS to reduce the cost of bandwidth. Furthermore, we will validate our design and study its performance under real complex environments. Improvements will be made based on the realistic studies before it comes to be deployed in SG.

REFERENCES


Shan Chang (M’08) received the B.S. degree in computer science and technology from the Xian Jiaotong University in 2004 and the Ph.D. degree in computer software and theory from the Xian Jiaotong University in 2013. She is now an assistant professor at the Department of Computer Science and Technology, Donghua University, Shanghai. Her research interests include security and privacy in mobile networks and sensor networks. She is a member of the IEEE Computer Society and Communication Society.

Hongyi Zhu (M’06) received his Ph.D. degree in computer science from Shanghai Jiao Tong University in 2009. He is now an associate professor at the Department of Computer Science and Engineering in Shanghai Jiao Tong University. His research interests include vehicular networks, mobile computing and smart computing. He is a member of the IEEE Computer Society and Communication Society.

Miaoxiong Dong (M’13) received B.S., M.S. and Ph.D. in Computer Science and Engineering from The University of Aizu, Japan. He is currently an Assistant Professor with Department of Information and Electronic Engineering, Muroran Institute of Technology, Japan. Dr. Dong is currently a research scientist with A3 Foresight Program (2011-2016) funded by Japan Society for the Promotion of Sciences (JSPS), NSFC of China, and NRF of Korea. His research interests include sensor networks, vehicular ad-hoc networks and wireless security.

Kaoru Ota (M’12) received M.S. degree in Computer Science from Oklahoma State University, USA in 2008 and Ph.D. degree in Computer Science and Engineering from The University of Aizu, Japan in 2012. She is currently an Assistant Professor with Department of Information and Electronic Engineering, Muroran Institute of Technology, Japan. Her research interests include wireless sensor networks, vehicular networks, and ubiquitous computing.

Xiaoping Liu received the B.S. and M.S. degree in computer science and technology from Harbin Institute of Technology, Harbin, China, in 1990 and 1995 and the Ph.D. degree in control theory and control engineering from Donghua University, Shanghai, China, in 2003. Since 2009, she has been an professor in the School of Computer Science and Technology, Donghua University, Shanghai. Her research interests include Adaptive Information System and Cloud Computing.

Xuemin (Sherman) Shen (M’97-SM’02-F’09) received the B.Sc. degree from Dalian Maritime University, Dalian, China, in 1982, and the M.Sc. and Ph.D. degrees from Rutgers University, New Brunswick, NJ, USA, in 1987 and 1990, respectively. He is currently a Professor and University Research Chair with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. His research focuses on resource management in interconnected wireless/wired networks, wireless network security, wireless body area networks, smart grid, and vehicular ad hoc and sensor networks. Dr. Shen is a registered Professional Engineer of Ontario, Canada, a Fellow of the Canadian Academy of Engineering, a Fellow of the Engineering Institute of Canada, and a Distinguished Lecturer of the IEEE Vehicular Technology Society and IEEE Communications Society.