OMiN - An Opportunistic Microblogging Network

Neil Wells & Tristan Henderson

ABSTRACT

OMiN is a pocket switched network running on smartphones. It allows users to send and receive microblog posts without using any global infrastructure such as the internet. Smartphones in close proximity to each other pass on messages according to a variation of the PROPHET routing protocol. Steps have been taken to secure the network and protect it from known attack vectors.

DECLARATION

I declare that the material submitted for assessment is my own work except where credit is explicitly given to others by citation or acknowledgement. This work was performed during the current academic year except where otherwise stated.

The main text of this project report is **NN,NNN* TODO** words long, including project specification and plan.

In submitting this project report to the University of St Andrews, I give permission for it to be made available for use in accordance with the regulations of the University Library. I also give permission for the title and abstract to be published and for copies of the report to be made and supplied at cost to any bona fide library or research worker, and to be made available on the World Wide Web. I retain the copyright in this work.

CONTENTS

INTRODUCTION

TODO

OBJECTIVES

PRIMARY OBJECTIVES

- Design and implement a protocol for discovering nodes in close proximity and passing messages and necessary metadata between them.
- Create a core library to manage message storage and routing.
- Implement a simple epidemic routing algorithm to send messages to all available nodes.
- Design a routing algorithm using user metadata to route messages while disguising message content and metadata.

SECONDARY OBJECTIVES

- Create a smartphone UI.
- Implement a more advanced routing algorithm.
- Design and implement a mechanism to decide whether a node is trustworthy or not.
- Evaluate the performance of the implemented routing algorithms.

TERTIARY OBJECTIVES

- Compare the real world vs simulated performance of the routing algorithms.

CONTEXT SURVEY

The following provides brief summary of opportunistic networks and the current state-of-the-art in opportunistic network technology. Only the most relevant subjects will be addressed in order to give the reader sufficient background information to fully understand the project.

OPPORTUNISTIC NETWORKS

An opportunistic network is a network where connections between nodes are sparse and a direct path from source to destination cannot be guaranteed. For example, a common form of opportunistic network (and the form we will focus on) is the Pocket

Switched Network (PSN) - a network of smartphones carried around by people. Connections are made between smartphones in close proximity using a short distance protocol such as Bluetooth. Because of the predictable nature of human behaviour, much research has been done to improve PSN algorithms.

Opportunistic networks must be able to store messages and forward them when connections become available. This makes it much harder to solve problems that have been solved in conventional connected networks (security, routing etc.).

SIMILAR PROJECTS

Haggle

Haggle (http://www.haggleproject.org)[1] - a pocket switched network designed to run on smartphones - is one of the largest opportunistic networks. There are implementations for a number of clients including Android (play.google.com/store/apps/details?id=org.haggle.kernel) and Windows Mobile.

By monitoring use of the platform, the authors discovered trends in inter-contact times and contact durations, showing that existing opportunistic routing algorithms are poorly suited to real world pocket switched networks[2].

FireChat

FireChat (opengarden.com/firechat) is a smartphone application used for off-the-grid messaging between nearby users. It has been used to circumvent government restrictions in Iraq (http://www.theguardian.com/technology/2014/jun/24/firechat-updates-as-40000-iraqis-download-mesh-chat-app-to-get-online-in-censored-baghdad) and during the Hong Kong protests (http://www.theguardian.com/world/2014/sep/29/firechat-messaging-app-powering-hong-kong-protests).

However, the app mostly relies on an internet connection, and its simple protocol is insecure (http://breizh-entropy.org/~nameless/random/posts/firechat_and_nearby_communication) and unable to implement the store-and-forward functionality of a proper opportunistic network.

OMiN will be a secure alternative to firechat which does not rely on an internet connection.

SWIM

The Shared Wireless Infostation Model (SWIM) is a proposed opportunistic network to monitor whales[3]. Small nodes are attached to the whales, which record data such as location and interaction with other whales. Connected nodes transfer this data between each other. Whenever data is transferred to a base station (the paper jokingly proposes using seabirds), it can be collected and stored.

Because data is shared between nodes, it is no longer necessary to find a whale with a sensor in order to acquire data from that sensor. This is a perfect example of the power of opportunistic networks in an environment with very limited connectivity.

ROUTING ALGORITHMS

Routing in opportunistic networks can be reduced to the problem of finding a route in a constantly changing graph. Most routing algorithms therefore build on existing graph search techniques. However, because the graph is constantly changing and is not necessarily random, such techniques are not necessarily the most effective (as shown by the Haggle project).

Context Based Routing

Context based routing is a form of greedy best-first search, where a single message is continually passed to the node most likely to reach the destination. There are a variety of methods to compute the utility of a node, including CAR[12] and MobySpace[13]. While it is not guaranteed to find the optimum path (or any path) to the destination, it uses very few resources as there is only one message being stored.

Epidemic Routing

The opposite of context based routing is epidemic routing - a form of uniform cost search[4]. Copies of the message are passed at every opportunity until the network is saturated. This is often likened to the spread of a virus. While this approach will always find the optimal path (because it takes all possible paths), it is very resource intensive - all nodes are expected to store every possible message. For this reason,

routing protocols that use similar techniques (known as dissemination based routing) concentrate on avoiding unnecessary use of resources.

PROPHET

The Probabilistic Routing in Intermittently Connected Networks (PROPHET) algorithm[5] uses a form of the A* algorithm **TODO: REALLY A*?**. A utility function (derived from recent encounters with nodes) is used to predict whether the message should be passed on to a node. This heuristic based approach uses fewer resources than traditional epidemic routing.

Bubble RAP

The Haggle project discovered that algorithms that treat routing as a generic graph search problem are often unsuited to PSNs. Bubble RAP[6] works on the idea that a social connections graph has tree like structure, where close nodes form a community. In order to send messages to a different community, the message is sent towards highly connected nodes near the root, and then towards the destination community and, eventually, the destination node.

This has been shown using the data collected from Haggle to be much more effective than standard routing algorithms **[TODO: CITATION NEEDED]**.

SECURITY

Security can be compromised in an opportunistic network by controlling a node or by intercepting messages during transmission. Common attack types include:

- Sybil attacks: impersonating another node in order to send messages that appear to be from that node or to receive messages intended for the node.
- Majority attack: by controlling a large number of nodes, an attacker can control a network which assumes that the majority of nodes can be trusted.
- Eavesdropping: gathering information such as message metadata to discover private information such as message contents and user location.
- Denial of Service: saturating the network with unwanted messages.
- Blackhole attack: failing to pass on messages to either reduce resource usage or as part of another attack.

Trust Based Security

Trust based security mechanisms depend on generating a list of trusted or untrusted nodes. This is commonly based on trusting connections in a social network[8] or distrusting nodes exhibiting strange behaviour[9]. While trust based security mechanisms can lower the chances of an attack, they are not infallible and can cripple the performance of the network if most nodes are not trusted.

Certificate Based Security

In order to increase security, a security mechanism must reduce the number of trusted parties to an absolute minimum. This can be done using an infrastructure of public key certificates to encrypt messages or verify their origin. However the problem of distributing this infrastructure without trusting an arbitrary number of nodes or contacting a central server has not been solved yet. On the internet, trusted certificate authorities manage and release certificates. However having a central trusted authority is infeasible in a scalable opportunistic network, as all nodes would have to connect to it directly at some point. Some mechanisms, like the one proposed by Shikfa et al[10] do use a central server, but only require it to be available for nodes joining the network. Other mechanisms split the responsibility over a number of nodes. Mechanisms for distributed certificate distribution require some level of trust in network nodes. For example Capkun et al's approach[11] does this by building a graph of certificates determining who trusts who. Any abnormalities in the trust graph may indicate foul play.

Identity Based Certification

Identity based encryption is an increasingly common form of encryption where the user's details (such as a username) acts as a public key, and a private key will be generated by a central private key generator (PKG). However, having a central PKG is problematic in a distributed network. Some security frameworks assume that there is a central PKG that can and will be accessed occasionally[14]. Others split up the PKG into multiple nodes, all of whom must collaborate to generate a private key[15] or split users into communities, each with their own PKGs[16]. The advantage of the identity based approach is that it is no longer necessary to distribute public keys - private key distribution is still necessary but can happen less frequently (e.g. when a central PKG on the internet is available).

REQUIREMENTS SPECIFICATION

USER REQUIREMENTS

Non-Functional requirements

- High: The user shall be able to create a unique identity.
- High: The user shall be able to send plain text messages to all others who follow the user or a hashtag in the message.
- High: The user shall be able to ‘follow’ any user and receive messages sent by that user.
- Medium: The user shall be able to cancel any message that they have sent.
- Low: The user shall be able to ‘follow’ any hashtag and receive messages containing that hashtag.
- Low: The user shall be able to send encrypted direct messages to a single user.
- Low: The user shall be able to send multimedia messages in addition to plain text.

Functional Requirements

None

SYSTEM REQUIREMENTS

Non-Functional Requirements

- High: The system shall work on smartphones or tablets capable of connecting to a wifi network.
- High: The system shall allow creation of user identities with a unique cryptographic identity.
- High: The system shall automatically connect to nearby nodes and pass on relevant information.
- Medium: The system shall provide a mechanism for securely distributing the cryptographic identity of a user.
- Medium: The system shall protect user metadata such as location and friends list from all other nodes.
- Medium: The system shall ensure that messages cannot be modified in transit or that such modifications can be detected.
- Medium: The system shall ensure that nodes cannot send a message that appears to be from another user.

- Medium: The system shall be robust and able to continue functioning when it encounters an unexpected state such as a malfunctioning or untrustworthy node.
- Medium: The system shall ensure that encrypted direct messages cannot be read by third parties.
- Low: The system shall be able to support multiple user identities on a single node.

Functional Requirements

- High: The system shall collect anonymous logging data for debugging and profiling purposes.
- Medium: The system shall ensure that a reasonable number of messages reach their intended recipients.
- Medium: The system shall restrict the size of the message buffer by evicting messages.
- Medium: The system shall reduce power usage where possible.

SOFTWARE ENGINEERING PROCESS

TODO

ETHICS

In order to test the real world performance of the network, we may ask people to use the application. In this case, some metadata will be collected on users, with their consent. This may include:

- An anonymous user ID.
- Anonymised 'Friends list' (or equivalent) of users.
- Metadata of messages passed during encounters, including message ID and origin ID, but NOT message contents.
- Times and locations of encounters between anonymous users.

DESIGN

USE CASES

We propose the following use cases for our network:

Disaster Area

A tsunami has wiped out all communications infrastructure in the area and injured a lot of people. Our opportunistic network is the quickest way contact medical teams to inform them of injured people who need help. We can assume that most people have smartphones and will be moving about regularly. Alice is injured and must contact the nearest free medical team so that they can help her. She uses her smartphone to publish a message with the #medical hashtag, her location, and her status to all nearby nodes. The message is distributed in this manner until it reaches doctor Bob, who subscribes to #medical. Bob sends a reply message to indicate that help is coming and goes to help Alice. Once she is happy that help is coming she sends out another message to indicate that she no longer needs help. Doctor Carol, who was also on her way after receiving the first message now knows that she is no longer required.

Privacy

A group of activists are concerned that their internet access is being monitored by their government and their online identity could be hacked or blocked in order to oppress them. They use our network to communicate in a way that cannot be blocked or subverted while revealing as little metadata (such as location) as possible. In order to organise a protest, Alex, a well known activist whose identity is unknown, sends out a message with details of the protest. Brian cannot come at that time because he has an exam, so he informs Alex of this via a secure direct message. Alex sends out a new message with a new date for the protest.

THREATS

From the use cases, we can construct a model of potential threats and their motives.

Disaster Area

In a disaster area, people tend to act altruistically. Therefore they are unlikely to attempt to subvert the network. However, people may act selfishly by attempt to conserve battery or memory space on their phone.

- Goal: Selfishly reduce personal resource usage
  - Avoid accepting messages to be passed on (blackhole attack)
  - Avoid making connections unless it directly benefits the user (e.g. to allow the user to send a message)

Privacy

In this scenario, it is very important that the network cannot be compromised by attackers. These attackers may have significant resources available to them.

- Goal: Prevent a message from being disseminated
  - Force all nodes to discard message
    - Control most network nodes (majority attack)
      - Create lots of blackhole nodes
      - Gain control of nodes
    - Overload most nodes (denial of service attack)
- Goal: Modify message while in transit
  - Control a node in the message path
    - Create node and get it into the message path
    - Gain control of a node in the path
- Goal: Prevent a user from sending messages
  - Force all nodes to discard messages from user
    - Control most network nodes (majority attack)
      - Create lots of blackhole nodes
      - Gain control of nodes
    - Discredit user
      - Send fake messages from user (sybil attack)
      - Exploit trust mechanism
  - Hack user's device
  - Legal action
    - Discover user identity
- Goal: Identify physical identity of a user
  - Derive network topology and route taken by a message to identify user location
  - Identify unique information about a user (such as phone number)
- Goal: Identify contents and recipient of an encrypted message
  - Derive route taken by a message to identify recipient
  - Discover private encryption key of recipient

- Control master PKG
- Brute force attack

AUTHENTICATION MODELS

Authentication is a major issue in opportunistic networks: how can we be sure that a message was sent by a user, and has not been modified in transit?

The simplest option is to forfeit authentication entirely and trust every node to behave correctly. This is possible for small networks, but as the network grows, the chances of having a deceitful user grows (even if each user is 99.9% trustworthy, a network with 700 users has a 50% probability of having a deceitful user). Therefore, this strategy is unfeasible if the network is to be scalable.

Networks which may contain deceitful users must use measures to prevent this. A common approach is for each user to identify nodes which may be trustworthy or untrustworthy. Social authentication is a common strategy: only 'friends' of users are considered trustworthy. The IRONMAN algorithm takes a different approach by detecting nodes displaying unusual behaviour. These nodes are then marked as untrustworthy. This approach of using heuristics to separate trustworthy and untrustworthy nodes is useful, but is still susceptible to attack by a resourceful attacker.

The best approach to avoid trusting an attacker is to trust as few nodes as possible. The only way to verify that a message actually came from a user is to have some certificate irrefutably proving that a message was sent by a user. If a message comes with a cryptographic certificate then anyone with the sender’s public key can verify its origin. So we now encounter a new problem: how can we distribute a public key while trusting as few nodes as possible?

KEY DISTRIBUTION

In order for a key based approach to work, we must be able to verify that an arbitrary user identifier (i.e. a username chosen by the user) is uniquely associated with an arbitrary (randomly chosen) public key. Modern Internet based techniques involve a trusted certificate authority (CA) who will distribute certificates guaranteeing that a key is associated with a user. This is impractical in a pure opportunistic network

setting (totally free of infrastructure), as it involves every node having contact with the CA, which considerably restricts the size of the network.

Instead of formally verifying identities, we could build up a trust based infrastructure for distribution of public keys. While this is a common method used in opportunistic networks, trust based approaches cannot be 100% secure (in particular, they are vulnerable to attackers controlling the majority of nodes).

In cases where a sporadic internet connection is available, a central CA can be used to allow identity based encryption by generating a private key for a user, so their identity (username) can be used as their public key. This has the added advantage of preventing sybil attacks because the CA can ensure that every user has a unique username.

In scenarios where the central CA is unavailable, we can resort to a hierarchical encryption model. Every node with a private key is capable of becoming a PKG and issuing private keys to other users. If the master PKG isn't available via the internet, another node can act as a delegate PKG. In this way we can create a chain of key generators where the master PKG (accessible via the internet) delegates PKG responsibilities down the chain. A node's private key will be compromised if one of its parents is compromised, so it is wise to keep this chain as short as possible. The risk of being compromised can be reduced by allowing users to get private keys from multiple PKGs. If all private keys are needed to verify a user, this reduces the risk of a node n being compromised by its PKG parents.

We can quantify the risk of a node being compromised with the formula **TODO: verify** $C(n) = 1 - (1 - V)(1 - \prod_{p \in P(n)} C(p))$ where the chance of a node being untrustworthy is V, $C(master) = V$, and $P(n)$ is the set of parent PKGs of n. As long as $0 < V < 1$, we believe (although it has not been proven) that the exact value of V will not affect the relative trustworthiness of two nodes (i.e. if $\exists\, 0 < V < 1$ such that $C(a) > C(b)$ then $C(a) > C(b)\ \forall\ 0 < V < 1$). Nodes should always seek to decrease this risk by finding new PKGs.

This scheme has the following advantages:

- It is scalable to any number of users.
- Users with the same identity can coexist iff they avoid using the same PKGs (i.e. if they are far away from each other and will never interact).
- If all users can connect to the master PKG over the internet, the scheme simplifies to normal (totally secure) ID based certification.
- The probability of a user being compromised can be directly quantified.

BLACKHOLE ATTACKS

TODO

ROUTING ALGORITHMS

Now we come to the problem of routing. Simply put, a routing algorithm takes all available information about a message and uses that information to decide where to send it. In trust based networks, this information is freely available to all trusted nodes (friends lists, recipients, previous paths etc). However, we have opted for a model where all nodes are considered untrustworthy. This means that we must obscure or remove all of this information, while still allowing a routing algorithm to use it.

Our network will be sending messages to multiple users, so a dissemination based routing protocol is advised. Given the disadvantages of epidemic routing, we will use a variation of the PROPHET routing algorithm[5] to distribute messages. We can use a variation of the SSNR-OSNR algorithm[17] to obfuscate sensitive metadata.

MESSAGE BUFFER EVICTION

When the message buffer is too large, it must evict a message. This message should be the least likely to be used again - the least popular. The simplest approach is the LRU (Least Recently Used) algorithm. Here, the oldest messages are removed on the assumption that they have probably reached their destination.

However, we have better ways of measuring message popularity - a message that has recently been passed to another node is likely to be more popular than a message that hasn't been passed on for a while. This heuristic allows us to identify the less popular messages that can be safely evicted.

IMPLEMENTATION

ENCRYPTION SCHEME

Requirements

The encryption algorithm should implement the following:

- Public keys are small enough to be distributed easily.
- Users can start without having to contact a central server to obtain a private key.
- ID hierarchy should be unbounded (unrestricted in depth) - i.e. PKG authenticates U_1 who authenticates U_2 and so on up to U_x for some arbitrary x.
- Verifiable message source.
- Verifiable message integrity.
- Message contents can be obscured from all but the intended recipient.
- Encryption scheme will not be broken in the foreseeable future.
- Encryption scheme has an existing implementation which will work on the target platform (Android).

Disaster Area Scenario

In the disaster scenario, the following requirements are necessary:

- Small public keys.
- Users can start without having to contact a central server.
- Verifiable message integrity.

Privacy Scenario

In the privacy scenario, the following requirements are necessary:

- Verifiable source.
- Verifiable integrity.
- Unbroken encryption scheme.
- Obscured message contents.

All other requirements are optional but would improve the flexibility of the network.

Candidate Algorithm - LW11 Unbounded HIBE Encryption

Lewko and Waters describe an unbounded hierarchical identity-based encryption algorithm [http://eprint.iacr.org/2010/197.pdf] with small public keys, the ability to encrypt message contents and an existing implementation [http://gas.dia.unisa.it/projects/jpbc/schemes/uhibe_lw11.html#.VNEBgM2sXeQ]. It has not been broken yet, although it has not received much attention.

Candidate Algorithm - DIP10 Bounded HIBE Encryption

De Caro, Lovino and Persiano describe a bounded hierarchical identity-based encryption algorithm [http://eprint.iacr.org/2010/197.pdf] with small public keys, the ability to encrypt message contents and an existing implementation [http://gas.dia.unisa.it/projects/jpbc/schemes/ahibe_dip10.html#.VNEG282sXeR]. It also has not been broken yet, although it has not received much attention.

Candidate Algorithm - PS06 IBE Signing

Paterson and Schuldt's ID-based signing algorithm [http://eprint.iacr.org/2009/380.pdf] has been implemented [http://gas.dia.unisa.it/projects/jpbc/schemes/ibs_ps06.html#.VNEHI82sXeQ] and provides a mechanism for verifying the source and integrity of a message, although it does require contact with a central server to distribute the private key (private keys cannot be generated by any user).

Algorithm Choice

Considering the needs of the the use cases and the capabilities of the algorithms, I believe the best choice to be a combination of PS06 for signing messages and LW11 for encrypting them. The only caveat of approach is that it does not allow an ID hierarchy - each user must communicate with a central PKG server to receive their private key.

DATABASE LIBRARY

The app needs to store messages and other data in a database. In order to simplify implementation, I decided to use an Object Relational Model (ORM) library to allow database records to be treated as objects. Some research showed that the Sugar ORM (http://satyan.github.io/sugar) library provided the necessary functionality and was easy to integrate with the application.

MESSAGE PASSING MEDIUM

There are a number of methods for smartphones in close proximity to interact. I considered the following methods:

- LAN communication - easy to implement but requires a LAN, which may not be possible for many use cases.
- Wifi-Direct - good range but requires that smartphones are not connected to a LAN.
- Bluetooth Low Energy - only supported by Android API 18+ (about 25% of devices) and research has shown that it is only as efficient as normal bluetooth [citation].
- Bluetooth - well supported although limited connectivity.

Considering the pros and cons of all of them, I decided to use Bluetooth to pass messages as it is almost universally supported and does not rely on smartphones being connected or disconnected from a LAN.

TODO

EVALUATION AND CRITICAL APPRAISAL

TODO

CONCLUSIONS

TODO

REFERENCES

[1] Scott J, Crowcroft J, Hui P, Diot C, Others. Haggle: A networking architecture designed around mobile users. In: WONS 2006: Third Annual Conference on Wireless On-demand Network Systems and Services; 2006. p. 78-86.

[2] Chaintreau A, Hui P, Crowcroft J, Diot C, Gass R, Scott J. Impact of human mobility on opportunistic forwarding algorithms. Mobile Computing, IEEE Transactions on. 2007;6(6):606-620.

[3] Small T, Haas ZJ. The Shared Wireless Infostation Model: A New Ad Hoc Networking Paradigm (or Where There is a Whale, There is a Way). In: Proceedings of the 4th ACM International Symposium on Mobile Ad

Hoc Networking &Amp; Computing. MobiHoc '03. New York, NY, USA: ACM; 2003. p. 233-244. Available from: http://doi.acm.org/10.1145/778415.778443.

[4] Vahdat A, Becker D, Others. Epidemic routing for partially connected ad hoc networks. Technical Report CS-200006, Duke University; 2000.

[5] Lindgren A, Doria A, Schelén O. Probabilistic Routing in Intermittently Connected Networks. SIGMOBILE Mob Comput Commun Rev. 2003 Jul;7(3):19-20. Available from: http://doi.acm.org/10.1145/961268.961272.

[6] Hui P, Crowcroft J, Yoneki E. Bubble Rap: Social-based Forwarding in Delay Tolerant Networks. In: Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing. MobiHoc '08. New York, NY, USA: ACM; 2008. p. 241-250. Available from:http://doi.acm.org/10.1145/1374618.1374652.

[7] Pelusi L, Passarella A, Conti M. Opportunistic networking: data forwarding in disconnected mobile ad hoc networks. Communications Magazine, IEEE. 2006 Nov;44(11):134-141. Available from: http://dx.doi.org/10.1109/MCOM.2006.248176.

[8] Trifunovic S, Legendre F, Anastasiades C. Social Trust in Opportunistic Networks. In: INFOCOM IEEE Conference on Computer Communications Workshops, 2010; 2010. p. 1-6. Available from: http://dx.doi.org/10.1109/INFCOMW.2010.5466696.

[9] Bigwood G, Henderson T. IRONMAN: Using Social Networks to Add Incentives and Reputation to Opportunistic Networks. In: Privacy, Security, Risk and Trust (PASSAT) and 2011 IEEE Third Inernational Conference on Social Computing (SocialCom), 2011 IEEE Third International Conference on; 2011. p. 65-72. Available from: http://dx.doi.org/10.1109/PASSAT/SocialCom.2011.60.

[10] Shikfa A, Onen M, Molva R. Bootstrapping security associations in opportunistic networks. In: Pervasive Computing and Communications Workshops (PERCOM Workshops), 2010 8th IEEE International Conference on; 2010. p. 147-152. Available from:http://dx.doi.org/10.1109/PERCOMW.2010.5470676.

[11] Capkun S, Buttyan L, Hubaux JP. Self-organized public-key management for mobile ad hoc networks. Mobile Computing, IEEE Transactions on. 2003 Jan;2(1):52-64. Available from: http://dx.doi.org/10.1109/TMC.2003.1195151.

[12] Musolesi M, Hailes S, Mascolo C. Adaptive routing for intermittently connected mobile ad hoc networks. In: World of Wireless Mobile and Multimedia Networks, 2005. WoWMoM 2005. Sixth IEEE International Symposium on a; 2005. p. 183-189. Available from:http://dx.doi.org/10.1109/WOWMOM.2005.17.

[13] Leguay J, Friedman T, Conan V. Evaluating Mobility Pattern Space Routing for DTNs. In: INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings; 2006. p. 1-10. Available from: http://dx.doi.org/10.1109/INFOCOM.2006.299.

[14] Kamat P, Baliga A, Trappe W. An Identity-based Security Framework For VANETs. In: Proceedings of the 3rd International Workshop on Vehicular Ad Hoc Networks. VANET '06. New York, NY, USA: ACM; 2006. p. 94-95. Available from: http://doi.acm.org/10.1145/1161064.1161083.

[15] Kong J, Petros Z, Luo H, Lu S, Zhang L. Providing robust and ubiquitous security support for mobile ad-hoc networks. In: Network Protocols, 2001. Ninth International Conference on; 2001. p. 251-260. Available from:http://dx.doi.org/10.1109/ICNP.2001.992905.

[16] Tseng FK, Zao JK, Liu YH, Kuo FP. Halo: A Hierarchical Identity-Based Public Key Infrastructure for Peer-to-Peer Opportunistic Collaboration. In: Mobile Data Management: Systems, Services and Middleware, 2009. MDM '09. Tenth International Conference on; 2009. p. 672-679. Available from: http://dx.doi.org/10.1109/MDM.2009.115.

[17] Parris I, Henderson T. Privacy-enhanced social-network routing. Computer Communications. 2012;35(1):62-74. Available from:http://www.sciencedirect.com/science/article/pii/S0140366410004767.

APPENDICES

APPENDIX A - TESTING SUMMARY

TODO

APPENDIX B - STATUS REPORT

TODO

APPENDIX C - USER MANUAL

TODO

APPENDIX D - MAINTENANCE DOCUMENT

TODO