Mobility and Address Freedom in AllNet

Edoardo Biagioni
University of Hawai'i at Mānoa
esb@hawaii.edu

ABSTRACT

Mobile devices can be addressed through a variety of means. We propose that each device select its own addresses, we motivate this choice, and we describe mechanisms for delivering data using these addresses.

Hierarchical point-of-attachment addresses are not effective with mobile devices. The network has to maintain a global mapping between addresses and locations whether or not the address is topological. Since this mapping is needed anyway, there is not much point in having the structure of the address encode device location. Instead, we have designed a network protocol, AllNet, to support self-selected addressing. When data is transmitted over the Internet, a Distributed Hash Table (DHT) provides a connection between senders and and receivers.

The advantages of self-selected addresses include the ability of devices to join and form a network without any need for prior agreement, and the ability to choose a personal, memorable address. When multiple devices choose the same address another mechanism, such as signed and encrypted messages, provides the necessary disambiguation.

Keywords

Network Architecture, Infrastructureless Communication, Interpersonal Communication, Routing, Ad-Hoc Network, Delay-Tolerant Network, Networking Protocol

1. INTRODUCTION

Ubiquitous communication cannot always be supported by expanding the fixed infrastructure. For the foreseeable future, there will always be areas that are not covered and people who cannot afford to pay for service.

Instead, mobile devices can themselves forward messages when there is no infrastructure to do so. Message forwarding may be very limited, but still provide basic communication infrastructure for emergencies and text messages. When desired, message forwarding can be extensive, with each mobile device essentially acting as a wireless access point as part of a mesh network.

As long as the mobile device keeps track of its owner's social network, the degree of service provided can be dependent on the owner's social distance from the parties in the communication. Specifically, each packet received by a mobile device and intended to be forwarded is assigned a local priority based on a number of factors, one of which is whether the sender or receiver can be identified within the social network stored in the device. This local priority determines how often the packet is forwarded and how long it is stored locally. The lowest level of service is designed to balance the selfish interests of the owner of the device by limiting the resources used to about 1% of what the device provides, with the community interest of providing a network that can be used by anyone within range for at least basic text messaging service.

Because messages are carried by mobile devices belonging to unknown persons, it is important that personal messages be encrypted. Similarly, because addresses are chosen independently by each device, the address offers is no guarantee that a message purporting to be from a friend is indeed from that friend. Instead, messages are authenticated by digital signatures.

Based on these principles, we have designed and built a prototype networking system called AllNet.

A crucial part of AllNet is the management of public keys. The most straightforward means of securely exchanging public keys is for two people to be operating their mobile devices within range of each other. One mobile devices will give a brief string to its user, who communicates it to the other person, who enters it into their own mobile devices. Once both devices share the same secret string, they can exchange their public keys together with an HMAC that proves that each knows the secret string [1].

With knowledge of all contacts' public keys, a device receiving a message carrying its own address can verify that it comes from a known contact, and only then decrypt it. Signature verification is relatively quick [2], so a mobile device can efficiently discard messages carrying its address, but not sent by a known contact.

Given that addresses are not required for correct delivery, it is worth pondering whether addresses are necessary at all. Consider that:

- mobile devices do not benefit from addresses that are determined by their position in the network topology, as that position is likely to change over time. As a result, there needs to be a way for messages to reach the destination device no matter where in the network that device may be connected.

- including addresses in messages makes it easier for an attacker to do traffic analysis.

- If a device is not connected to the Internet, and all data communication is via ad-hoc[1] or delay-tolerant[2] networks, then addresses seem even more superfluous. Data in such networks can be broadcast to all devices, as long as the network is sufficiently small and the traffic sufficiently low.

After considering current schemes for mobile device addressing in Section 2 and describing the addressing scheme of AllNet in Section 3, in Section 4 we look at each of these three disadvantages of addressing in mobile networks, and we point out the strengths of self-selected, position-independent addresses. Section 5 then considers the many uses to which self-selected addresses can be put.

2. BACKGROUND

2.1 Mobile IP

IP addresses contain a network part, zero or more subnetwork parts, and a host part. This hierarchical arrangement is ideal for minimizing the size of routing tables in the core of the Internet. Minimizing routing table size has the direct benefit of lessening the data structure to search through when forwarding a packet, and also reduces the amount of information that must be exchanged to maintain the routing tables. Packets are forwarded to networks, then within networks to subnets, and in the final network, forwarded based on the entire IP address.

The main limitation of hierarchical point-of-attachment addresses is that when the point of attachment to the network changes, the address must also change. This has been recognized since the work on Mobile IP [8] [9] [10]. In Mobile IP, a "mobile node is also associated with a care-of address, which provides information about its current point of attachment to the Internet." [9]. IP datagrams are routed to the destination network, where the home agent forwards the datagrams to the destination through an IP tunnel.

The collection of the home agents provides a global distributed database mapping each mobile node's permanent address to its current point of attachment address.

2.2 Mobile Telephony

Mobile devices using mobile telephony (and mobile data) protocols have to announce their presence at their current location. Similar to Mobile IP, this announcement updates the Home Location Register at the cellphone's home location. This Home Location Register is then used to route incoming calls [7].

As an optimization, the mobile device's location is usually cached in the Mobile Switching Center's Visitor Location Register of its current location.

The collection of strategies used for mobile telephony optimizes call routing for mobile devices that are in their home location, or at least haven't moved recently. When these optimization mechanisms do not deliver, however, the call is ultimately routed based on information stored in the Home Location Register.

The collection of Home Location Registers across the world essentially provides a global distributed database mapping each mobile node's permanent address to its current point of attachment address.

2.3 Global Database of Locations

Any mobile system is likely to have some similarities to both mobile IP and mobile telephony. As we have seen, except in small networks where broadcasting can be useful, mobility requires a global database that tracks the location of mobile devices. The mobile device address is then an index into the database. Calls or datagrams are forwarded based on the database contents.

It seems reasonable to state that a fixed hierarchical address cannot possibly consistently indicate the changing location of a mobile device, and that some sort of database is required to support mobility.

In such a database, an address is an indication of where to locate the correct database entry rather than an indication of the location of the mobile device.

With the way Mobile IP and Mobile Telephony databases are distributed, the IP address or mobile telephone number does loosely indicate which server contains the desired translation. However, this is a matter of management rather than technology: both Mobile IP and Mobile Telephony assume a hierarchical address assignment in which addresses are received from a central authority, perhaps through intermediate providers. Alternative technologies, such as Distributed Hash Tables [11, 13, 12, 15], can provide distributed mappings for arbitrary addresses that are not assigned hierarchi-

[1]In an ad-hoc network, messages are forwarded by intermediate devices who happen to be between a source and destination, no matter what the main functionality of the device is.

[2]In a delay-tolerant network, messages are carried by the physical motion of devices on which the messages are cached, and delivered when the destination device is in range.

cally.

Similar to Mobile IP and Mobile Telephony, AllNet uses the destination address to identify the destination of a message but not the location or point of attachment of the destination. Unlike these systems, AllNet allows devices to select their own address. In AllNet, addresses need not be unique, since AllNet uses encryption to identify messages sent to or from a specific device.

2.4 Secure Networks

Revelations of message interceptions by governments and large corporations have prompted the recent development of secure communication systems. Older secure communication systems have sprung from the awareness of the vulnerability to interception of normal means of communications, including particularly electronic mail.

Systems such as PGP [16] and TOR [6] have been set up so individuals can attempt to communicate securely. However, the onus is generally on the individual to do what is necessary to preserve security. Technically unsophisticated users, and sometimes even technically sophisticated users, may employ these systems yet still inadvertently leak information that is intended to be kept confidential.

The remainder of this section describes in some detail two systems, one recent and one older, that are designed to provide secure communications even to relatively unsophisticated users.

2.4.1 BitMessage

The core philosophy of BitMessage is to securely encrypt every message, then broadcast it to all, or a subset of, nodes in the BitMessage network [14].

Like AllNet, BitMessage relies on encryption to determine whether a message is intended for a particular mobile device. Unlike AllNet, BitMessage supports no data destination addresses at all. This, together with the broadcast mechanism, causes foreseeable problems as the BitMessage network grows. To accomodate such growth, BitMessage uses a number of mechanisms:

- Proof of work: like Bitcoin, BitMessage requires that messages sent through the network carry a proof of work. This lessens the number of messages that can be sent on the network.
- Each node in the BitMessage network only keeps and forwards messages that are less than 3 days old.
- As traffic grows, it can be split into separate streams, and messages are only broadcast to nodes within a stream.

Like AllNet, BitMessage relies on key hashes to identify users that one wishes to communicate with. A mechanism for exchanging keys lets the hash be used to verify that indeed this is the individual one wishes to communicate with. BitMessage calls these hashes "addresses", since BitMessage has no message destination addresses as such. The partitioning of destinations is reserved for streams, which are only used to reduce the amount of messages that a server must cache.

Addresses in AllNet are used for message routing, in the same way that streams are used for BitMessage, but with the assumption that broadcast is used as a backup communication mechanism rather than the principal way of distributing data.

AllNet also supports AllNet Human Readable Addresses (AHRAs), which are essentially hashes of public keys. Once the AHRA is known, a key request can provide the matching key (non-matching keys are ignored) so secure communication can be provided in one direction.

AHRAs are described in Section 5.2.

2.4.2 Freenet

Like BitMessage, the purpose of Freenet is to anonymously and securely distribute information [4]. Unlike BitMessage, Freenet is intended to securely distribute content rather than interpersonal messages. Freenet is a well-known, mature and yet still evolving project.

Similarities between Freenet and AllNet include the technical anonymity of participants[3] and the pervasive use of encryption to maintain this anonymity. The use of hashes to identify entities of interest is also similar, as is the strategy of moving content closer to its destination.

Freenet has also been moving closer to the AllNet strategy of establishing connections primarily among people already known to each other. This network is only accessible to people who have a personal connection to someone already in the network, and is thus known as the Darknet, as opposed to the regular Freenet which is available to anyone.

Even with these similarities, there are many differences between AllNet and Freenet. Primarily, Freenet is for content distribution, whereas AllNet is intended to be general purpose, while designed specifically to support secure interpersonal communication.

3. ALLNET ADDRESS DESIGN

3.1 AllNet Overview

AllNet is a protocol designed primarily to support secure interpersonal communication among people who already know each other, but also useful for other forms of secure communication.

AllNet relies on two main underlying means of communication: the Internet, and direct ad-hoc and delay-

[3]The technology provides anonymity only as long as the participants don't distribute self-identifying information.

tolerant networking between mobile devices. Each of these is used as available.

Secure key exchange is provided either through direct communication among mobile devices in close proximity, or by the authenticated exchange of Allnet Human Readable Addresses (AHRAs). The first mechanism requires users to enter a short string, to prevent spoofing by other nearby devices. The second mechanism leverages any authenticated (not necessarily encrypted – for example, a telephone conversation or a business card) side channel to exchange AHRAs, which then provide assurance that the key exchange matches the party providing the AHRA.

Unlike the proof-of-work mechanism of BitMessage, AllNet devices accept all messages they receive. Because a device may receive more messages than it is willing or able to forward and cache, each device has an automatic prioritization mechanism based on information available in each message. The prioritization may be changed by each device as appropriate, but by default gives priority to messages recognizably to or from friends, and to a lesser extent, to messages that will consume the least device and network resources.

Within ad-hoc and delay-tolerant networks of mobile devices forwarding and sometimes physically carrying messages, messages are essentially broadcast. At any give time, the highest priority messages are forwarded. One of the goals of AllNet is to forward messages for unknown people using at most a small fraction of available resources, typically 1%. Given the priority and resource limitations, broadcast should deliver messages whenever allowed by the resource constraints.

When forwarding data on the Internet, AllNet sends messages to any IP address it has that identifies the destination. This includes the destination itself, any designated Rendezvous Point (RP), and nodes in the AllNet Distributed Hash Table (DHT). Again, only the highest priority messages are forwarded, but this may include all packets if traffic is low.

AllNet messages may be acknowledged. When an acknowledgement is requested, the sender includes in each message a 16-byte random string called the message ACK. If the message is encrypted, the message ACK is encrypted along with the message itself, and the sender also includes in the message the first 16 bytes of the unencrypted hash of the message ACK. This hash is called the message ID. Only the intended recipient will be able to decrypt the message and recover the message ACK. When the recipient returns this ACK to the sender, every other node can hash the ACK and, if it corresponds to the message ID of a previously seen message, delete the corresponding message from its cache.

AllNet is used primarily for interpersonal communication, so AllNet maintains a list of contacts and the public key for each contact. That is, each user's social network is kept only on the user's device, with enough information to allow secure and authenticated communication with each person in the social network.

Additional details about AllNet are available from prior papers [3, 1, 17].

3.2 AllNet Addresses

Section 2 explained some of the ways in which AllNet addresses resemble and differ from addresses used in other networks. Like some other networks, AllNet has two kinds of addresses: addresses used for data routing and delivery, and AllNet Human Readable Addresses, or AHRAs. The latter combine some of the properties of Domain Names with the hashes used in BitMessages to provide both human-readable and memorable addressing and a degree of security. AHRAs are described in Section 5.2. This section describes the AllNet addresses used in data delivery.

This explanation of the design of AllNet addresses begins by describing a simplified version of AllNet that is a pure broadcast network. In this simplified network, every participating node receives every message, and no addresses are used. Since most AllNet messages are signed and encrypted[4], every receiver of a message checks to see if the message is signed by a sender known from the social network. In the absence of addresses, this can be done only by trying to verify the signature with the public key of every contact in the social network. If one of these verifications succeeds, then the receiver attempts to decrypt the message. If the decryption succeeds, then with overwhelming likelihood, the message was encrypted with this receiver's public key. Therefore, in theory AllNet can be used even without addresses.

Since message addresses can be used by an attacker for Traffic Analysis, sending and receiving without addresses is supported by AllNet. Each packet carries both a 64-bit sending and receiving address, and also the number of bits of each address that are meaningful. If the number of bits (of either source or destination address, or both) is set to zero, then that address cannot be used for Traffic Analysis.

Messages with no addresses can be useful on relatively small networks or where security is sufficiently important that traffic analysis should be thwarted and the overhead of verifying all the signatures in the social network is acceptable. However, this is not practical on most mobile devices, which are resource constrained and where the threat of traffic analysis is not substantial. So most AllNet messages carry a number of address bits > 0, currently typically 8 bits.

As well as storing the public key, AllNet stores a local and remote address for each contact. The number

[4] AllNet also supports broadcast messages, which are signed but not encrypted.

of bits in the address is decided independently by each peer. When peers exchange messages, they may specify any number of bits. A source or destination address matches the remote or local address for a peer when the number of matching bits is at least the number of bits specified in the packet, or at least the number specified in the original exchange. For example, if Alice gives to Bob an 8-bit address, and Bob gives to Alice a 16-bit address, then Bob may attempt to verify and decrypt any packet with s source bits and d destination bits as long as the first $\min(s, 8)$ bits of the source address match Alice's address, and the first $\min(d, 16)$ bits of the destination address match the address that Bob gave to Alice.

This flexibility means each communication can be configured to trade off power consumption with security. When more address bits are specified, each device needs to verify and decrypt fewer packets, which can be important for low-power energy-constrained devices. With fewer bits, more verifications and decryptions must be carried out to find out which the packets are meaningful to a particular user. When resources are limited and traffic is high, a device may only verify and decrypt packets with at least a given number of address bits.

In practice, the current version of AllNet (3.0) always exchanges 16-bit addresses, and sends 8-bit addresses. Even with a relatively large network, 8-bit source and destination addresses mean that typically $2^8 < 1\%$ of the public keys in a user's social network will be tried for any received packet. The 16-bit address exchange allows for longer addresses and better filtering to be used in the future.

Because the bit pattern in the address is not meaningful to AllNet, each device can select any suitable address, without coordinating with any central authority or even a group consensus. Instead, each device can get online without prior authorization, and may even choose different addresses for communication with different peers.

The ability to self-select an address fits well with the AllNet emphasis on distributed communication where each device is an equal peer, and with enabling communication whenever each device can possibly communicate, without waiting for registration or approval.

Whenever addresses are self-selected, there is the possibility that two systems will select the same address. However, as illustrated by the simplified example of AllNet using no addresses, duplicate addresses only require additional verification or decryption attempts, and do not lead to incorrect behavior.

AllNet addresses can be selected at random, and random addresses are perfectly reasonable for security. On the other hand, many individuals would want an address that is memorable and can easily be communicated to others. In this case, the actual AllNet address can be the hash of a self-selected Personal Name or PN. Such a personal name can range from a simple name such as "Alice", to more precise names such as "Acme Software Co., Peoria". Either of these will hash to a seemingly random bitstring, the first few bits of which can be used as the local address.

To minimize the effect of mis-spellings and confusing fonts, each letter is mapped to 4 bits, and multiple, sometimes indistinguishable letters are mapped to the same pattern. For example, the letters "1", "l", "L", "i", and "I" are all mapped to the same bit pattern `0001`.

3.3 Forwarding and Routing

In the hypothetical simplified AllNet described above, all messages were broadcast to all devices. This is effective in smaller networks, for example when sending data in an ad-hoc or delay-tolerant fashion directly between mobile devices. However, broadcast does not scale to larger networks. When AllNet devices are connected to the Internet, they also forward data using two well-established mechanisms, Rendezvous Points and Distributed Hash Tables.

An AllNet Rendezvous Point (RP) is a device reachable at a known IP address, which is willing to accept and relay messages to and from other AllNet devices. The RP may be open, accepting arbitrary AllNet data, or may be closed, only accepting data signed with known keys. This distinction is similar to the Opennet and Darknet versions of Freenet. With either kind of RP, the sender sends data to the RP, which caches it and forwards it to any other AllNet nodes that have registered as listeners. At a later time, another device may send a data request to the the RP, specifying what addresses it is interested in receiving. In response, the RP returns any cached messages that match the receiver's specifications.

The cache is maintained in priority order, using any priority scheme acceptable to the manager of the RP, or the default AllNet priority if no other priority has been specified. In addition, an RP will typically delete from its cache any message that has been acknowledged by the receiver, as described in Section 3.1.

Although RPs can be very useful, they must either be set up individually, or an AllNet user must select a public RP, which may well be oversubscribed to the point of deleting useful data before it is forwarded. So as well as RPs, AllNet supports Distributed Hash Tables, or DHTs, which can scale to large sizes and still distribute load reasonably well.

The design of the AllNet DHT is conventional. AllNet nodes individually decide to become DHT nodes if they have sufficient resources, including an IP address that can be used to reach them (i.e. not behind a NAT). The address space is 64 bits, which matches the AllNet

address space. Each node selects its own 64-bit identifier – in case of collision, all nodes with the same address store the same messages. Ideally, each AllNet message is cached in at least the 4 DHT nodes preceding the destination address of the message.

What is less conventional about the AllNet DHT is that it supports partially specified addresses, that is, AllNet addresses with fewer than 64 bits. When the DHT is small, the first few bits of the address are sufficient to identify which DHT nodes might hold messages for a given address. As the DHT grows, increasingly more bits are needed. If a message address has too few bits to distinguish a target DHT node from its neighbors, the DHT node will cache it with lowered probability.

As a specific example, consider a message for destination `1011...`, which only specifies four bits. Ideally, the message would be stored in all the DHT nodes responsible for addresses `1011 0000 0000...` through `1011 1111 1111....` Since the goal is to store the message in at least four nodes, each DHT node keeps track of how many DHT nodes match each prefix of its identifier. If an address matches $n > 4$ DHT nodes, each of the matching DHT nodes saves the message with probability $4/n$. The receiver will find this message after querying $n/4$ DHT nodes on average.

An AllNet device can retrieve its messages from the DHT in a manner analogous to retrieving messages saved in an RP, once the DHT node(s) likely to hold its messages have been identified.

4. ADVANTAGES OF ALLNET ADDRESSING

Since AllNet addresses are self-selected,

1. they have no relation to the device's point of attachment, so they need not change when the mobile device actually moves

2. being self-selected, they never need to change unless the device's owner wishes to change them, and

3. the AllNet addresses can be chosen to correspond to meaningful names that are easy to communicate and remember.

These benefits follow directly from giving the user the power to select his or her own address, and from AllNet working well even when addresses are duplicate.

For comparison, BitMessage addresses have the first two of these benefits but not the last one. Existing mobile IP and mobile Telephony addresses only provide the first of these benefits, and even then, only as long as the address assignment lasts. For mobile Telephony (and sometimes for Mobile IP) this requires a financial commitment.

Other advantages of AllNet addresses were mentioned in Section 3.2. These include the option of masking most or all of the bits of the address to foil traffic analysis, and the competing benefit of using addresses help filter out packets for which verification and decryption need not be attempted.

When AllNet is used for wireless ad-hoc and delay-tolerant message transmission, addresses are also useful to select a priority for outgoing messages. Since wireless spectrum is sometimes a valuable commodity, and in any case using onboard radios consumes energy, it is beneficial to prioritize the transmission of packets that are of use to the recipient. In the handshake at the beginning of an AllNet ad-hoc exchange, receivers may indicate addresses they are particularly interested in, and senders may prioritize such messages.

5. USING ALLNET ADDRESSES

5.1 Multiple Addresses per Device

One further benefit of having self-selected addresses is the ease with which one may select multiple addresses to be used in different circumstances. It it is easy to use a different destination address for every contact in the social network, for example.

IPv6 [5] has already introduced the world to the notion that it is reasonable to assign different IP addresses to a single interface. With a device having multiple addresses, receiving is relatively easy – the device simply receives any message for which the destination address matches any one of its addresses. Sending requires that one of the available addresses be selected as the source address for outgoing messages, and so rules are needed to select one address over the other addresses.

For AllNet, the rule is relatively simple. The public IP address is only used as a source address if there is no shared key with the destination (if there are multiple public IP addresses, then the user can select one). Whenever keys are exchanged as part of the initial handshake in any private communication, randomly-selected addresses valid only for this social connection are exchanged as well. Then, each local address is associated with a specific remote address, local and remote public keys, and local secret key, and all communication can use these specific addresses.

5.2 AllNet Human-Readable Addresses

Section 2 introduced the notion of AllNet Human-Readable Addresses, or AHRAs. Unlike regular AllNet addresses, AHRAs are designed specifically to exchange among humans, in a way analogous to current email addresses, but hopefully more easily remembered. This section describes AHRAs in detail.

The basic format of an AHRA resembles an email address:

`personal_name@word_pair.word_pair`

The personal name (PN) is the one of the public addresses for this individual. An address including only the PN (`name@`) is valid, but may not be unique.

The word pairs (WPs) are a way of encoding a hash of the public key stored in the device in a memorable way. Each word pair encodes 14 bits of the hash, with successive pairs encoding successive bits. The 14 bits are encoded as two seven-bit parts, each taken from a dictionary of 128 common words. For example, in English, the first few words in the dictionary used for the first part of a word pair includes the words "the", "be", "of", "to", "a", and so on. The dictionary for the second word in the pairs includes the words "time", "people", "year", "well", "work", and so on. Word pairs might then be "of-time", "to-work", and so on.

When a user wishes to create an AllNet address, the user puts his or her device to work creating keys. Keys must have certain properties (described in the next paragraph), so in general a large number of public/secret key pairs must be generated. All the keys that satisfy the properties needed of an AHRA are recorded and saved for the user's perusal. The user then gets to choose which of these keys the user prefers. For example, AllNet has a time server that sends a time message once an hour. The AHRA for this time server is `allnet-hourly-time-server@if-wish.think-past.get-future`. This was one of many AHRAs generated for the PN "allnet-hourly-time-server", and was selected by the author who believes these word pairs are memorable in the context of this PN.

The property that makes a public key valid for use in an AHRA is that the cyphertext from encrypting the PN with the public key must contain a minimum of n 16-bit strings taken sequentially from the hash of the PN.

For example, consider a PN that hashes to a value ending[5] with (hex) `5518 22B5 5D7C` . Then, if $n = 3$, the only acceptable public keys for this PN are those where the PN, when encrypted with the key, contains all of the 16-bit strings `5518`, `22B5`, and `5D7C`. If $n = 2$, only the last two are required.

Assuming that the PN, when encrypted using the public key, is no longer than $2^{14} = 16,384$ bits, then only 14 bits are needed to encode the position of each of the bit strings found. These positions are what is encoded in the word pairs.

With this scheme,

- verifying that a public key matches a given AHRA is very fast, requiring only an encryption and a few lookups
- generating a key given a personal name (PN) is somewhat slower, but still
- is much faster than generating a key to match a complete AHRA

To see that the last is true, consider that in generating a key, approximately $(2^b/l)^n$ keys must be examined to find one with n matches of the last b-bit (in AllNet, $b = 16$) strings of the hash of the PN, given that the encrypted PN has l bits. Conversely, given an AHRA with n word pairs, approximately 2^{bn} keys must be examined to find a match. The ratio of the work needed to steal a key to the work needed to generate a key is then l^n.

For an $l = 4,096$-bit encrypted PN, the attacker must then do over 4,000 times more work than the generator for each word pair in the AHRA. With just 3 word pairs, the attacker must search $68,719,476,736$ more keys than the generator to find a match. This is assuming that:

- the public key encryption is independent from hashing, so that the likelihood of finding the bit strings of the hash in the ciphertext is effectively random
- exhaustive search is the most effective way for an attacker to build a public key matching a given AHRA

An AHRA with n word pairs can always be given to somebody with only the first $m < n$ word pairs. This lessens security, but makes the AHRA easier to communicate and remember. In the example of the hourly time server, it would be sufficient to rememeber `allnet-hourly-time-server@if-wish`. There is thus no penalty to choosing an AHRA with a large number n of word pairs, providing security only when desired.

6. CONCLUSIONS

6.1 Summary

We have shown that in mobile systems the address space is essentially flat, and reliance on a global translation database is a necessary part of any mobile system. This database may be distributed, and the address is used as an index into the database.

The design of AllNet takes advantage of this design requirement to dispense with the need for addresses to be assigned hierarchically. Since AllNet already provides encryption and authentication, these are leveraged to support non-unique addresses. Addresses can then be self-selected, and used both as an optimization to avoid having to verify and decrypt every message, and as a key for locating messages in a distributed hash table.

Self-selected, not globally unique addresses may be chosen to represent a meaningful name without the contortions needed in systems requiring globally unique

[5]The beginning of the hash is used as the local address, so the bit strings are taken from the end of the hash.

names. The AllNet Human-Readable Addresses combine personal names with sets of word pairs identifying public keys.

6.2 Implementation and Future Work

Everything described here has been implemented in AllNet version 3 [17], with the exception of the the Distributed Hash Tables, for which implementation is ongoing (the current implementation relies on a single server as a Rendezvous Point, which is effective as long as traffic is light and the server is reliable). The wireless ad-hoc portion of AllNet, which has not been emphasized in this paper, is also still being improved. Efforts are underway to port the original Linux implementation of AllNet to other platforms, initially Android, but also foreseeably IOS and Windows.

Many improvements are possible, but further experience with the current system will guide priorities for the near future.

One theoretically interesting issue is how to use addresses while defeating traffic analysis. For example, two peers might agree to change their address after every message. The function to change the address would be known to the peers, but not to outsiders – for example, the HMAC of the current address with a secret string only known only to the peers. Then, a peer would check the hash table locations corresponding to the next address in the sequence. This is more efficient than sending messages to addresses with few or no bits. An attacker eavesdropping on a given device would be able to tell that messages have been sent or received, but have no indication of who the message was from. An attacker eavesdropping on AllNet messages but with no indication of which device had sent or received the message, would have no information whatsoever.

6.3 Conclusion

Hierarchically assigned point-of-attachment globally unique addresses have been and continue to be amazingly useful, and indeed the current Internet would be unimaginable without them. However, these addresses simply don't work well for mobile nodes. Either the mobile node must change address every time its point of attachment to the network changes, or the address cannot identify the point of attachment.

This limitation, while it requires redirection and updating of the mobile database whenever the mobile node moves, is also liberating, providing opportunities to choose meaningful and memorable addresses that reflect a person's preference rather than the person's or device's position in a hierarchy.

These opportunities extend to the technical side of addressing. Since an intermediary is required to locate mobile nodes, AllNet uses this intermediary as a Rendezvous Point, usually a node in the Distributed Hash Table, to cache and forward messages. This is analogous to an email server, which can be contacted at a known location by both the sender and the receiver of the email, and stores email until deleted by the receiver.

The indirection required to support mobility can be an opportunity for new designs. We have presented the design choices for AllNet, but many other choices are possible, and we hope this paper inspires others to take advantage of this new freedom.

7. REFERENCES

[1] E. Biagioni, "Ubiquitous Interpersonal Communication over Ad-Hoc Networks and the Internet", 47th Annual Hawaii Convention, January 2014, Waikoloa, Hawaii, USA.

[2] E. Biagioni, Y. Dong, W. Peterson, K. Sugihara, "A Protocol for Secure Electronic Remote Voting". IFIP International Conference on Network and Service Security (N2S), Paris, France, June 2009.

[3] Edoardo Biagioni, "A Ubiquitous, Infrastructure-Free Network for Interpersonal Communication", 4th Int'l Conf. on Ubiquitous and Future Networks (ICUFN), July 2012.

[4] Ian Clarke, "A Distributed Decentralized Information Storage and Retrieval System", Div. of Informatics, Univ. of Edinburgh 1999, `https://freenetproject.org/papers/ddisrs.pdf`

[5] S. Deering and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, Dec 1998.

[6] R. Dingledine, N. Mathewson, P. Syverson, "Tor: The Second-Generation Onion Router", Usenix Security 2004.

[7] "Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); LTE; Location management procedures" 3GPP TS 23.012 version 11.2.0 Release 11, Jan. 2013.

[8] C. Perkins, ed., "IP Mobility Support", RFC 2002, Oct. 1996.

[9] C. Perkins, "IP Mobility Support for IPv4, Revised", RFC 5944, Nov. 2010.

[10] C. Perkins, ed., D. Johnson, J. Arkko, "Mobility Support in IPv6", RFC 6275, Jul. 2011.

[11] Ratnasamy, Francis, Handley, Karp and Shenker, "A scalable content-addressable network", ACM SigCOMM 2001.

[12] Rowstron and Druschel, "Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems", 18th Int'l Conf. on Distributed Systems Platforms, 2001.

[13] Stoica, Morris, Karger, Kaashoek, and Balakrishnan, "Chord: A scalable peer-to-peer lookup service for Internet applications", ACM SigCOMM 2001.

[14] Jonathan Warren, "Bitmessage: A Peer-to-Peer Message Authentication and Delivery System" Nov. 2012, https://bitmessage.org/bitmessage.pdf
[15] Zhao, Huang, Stribling, Rhea, Joseph, Kubiatowicz, "Tapestry: A Resilient Global-Scale Overlay for Service Deployment", IEEE JSAC, vol. 22, 2004.
[16] Philip Zimmermann, "Why I Wrote PGP", in the Original 1991 PGP User's Guide (updated in 1999).
[17] http://alnt.org/