Access Path Based Source Address Validation in Mobile IPv6

Min Zhu, Ke Xu, and Qi Li

Tsinghua National Laboratory for Information Science and Technology,
Department of Computer Science and Technology, Tsinghua University,
Beijing, China
{zhumin,xuke,liqi}@csnet1.cs.tsinghua.edu.cn

Abstract. Mobile IPv6 runs high risk of being attacked by IP spoofing due to the introduction of mobility and route optimization. In this paper, an authentic IP address validation scheme is proposed to protect mobile nodes in Mobile IPv6 against IP spoofing attack. The mobile nodes' historical traffic information is leveraged to validate the authenticity of its claimed home address in the scheme. Compared with other authentication schemes, this scheme is much simpler to implement and easier to deploy based on the usage of real data, and does not require additional computational overhead. It also solves the address ownership problem and the unauthenticated binding update issue in Mobile IPv6. Real traces are used to demonstrate the applicability of the scheme in this paper. The experimental results show that only three consecutive historical packet records are required to construct a unique authentication key, which can identify forged home address efficiently.

1 Introduction

As an enhancement to IPv6, Mobile IPv6 provides more flexible and open communication with support of the binding update protocol and the route optimization protocol, which allows mobile nodes to roam freely in the network with two addresses[7]. When the mobile node is in the home subnet, it is identified by its home address. When it moves to a foreign subnet, it obtains a new address as care of address. The key point of mobile node is to notify the home agent and the correspondent node its current care of address in a reliable and timely way through binding update request.

However, exploiting the security vulnerability of unauthenticated binding update, an attacker can masquerade as a mobile node (MN) and send a binding update request to the home agent (HA) and the correspondent node (CN) easily[10]. As IP address plays an important role in Internet, it is a disaster if such request is trusted and accepted: the normal traffic direction in the network will be changed, and the network resources will be consumed. Moreover, the victim's confidential information may be stolen[5]. Obviously, mechanisms to authenticate the identity of the mobile nodes are crucial to network communication and management.

E. Altman and W. Shi (Eds.): NPC 2011, LNCS 6985, pp. 334–349, 2011.

The IETF has proposed IPsec[2] and return routability (RR)[7] mechanism to verify the identity of the mobile nodes. Some cryptography-based mechanisms with the mobility support such as AIP[1] and CGA[3] have also been proposed to generate accountability address in order to verify the identity of the source address. But these mechanisms have not been widely applied yet, because of their computation complexity, heavy storage overhead and long communication latency. As a result, further studies on source address validation in Mobile IPv6 are strongly needed.

The main reason behind the success of masquerade attack in Mobile IPv6 is that the home agent and the correspondent node cannot determine the ownership of the address claimed by the mobile node with unauthenticated binding update. If the mobile node can prove its identity after its move, the masquerade attack can be distinguished. SAVA[16,17] is the first method that proposed authentic source IP address. SAVA guarantees address authenticity in a hierarchical way. First, it completes the source address validation in local subnet by exploiting the dynamic binding relationship between the switch port and the valid source IP address. Second, it completes authentication at Intra-AS level through the ingress/egress filtering. Finally, it completes Inter-AS source address validation by alliance. SAVA has currently been deployed in CNGI-CERNET2[1], which is a large-scale pure IPv6 backbone. However, SAVA does not consider the authentic source IP address issue in Mobile IPv6. Our work is an extension of SAVA, and we will solve the authentic source address validation problem in Mobile IPv6 in this paper.

Identity authentication is not unique to network security, but exists in our daily life. For example, in credit cards customer service, some authentication questions will be asked to verify our identity, such as credit card number, home address, phone number, and the amount of recent consumption. These information together forms a key for a certain card holder. Following the same logic, our study attempts to address the following questions:

- Is there any information only known to the mobile node and the home agent or correspondent node that can be used to verify the node's identity?
- How does a mobile node use such information to signal its ownership of an IP address?
- How to determine the effectiveness of the authentication information?

In this paper, we try to extract authentication information from historical traffic generated by users (or end hosts/nodes). This is possible because users' traffic is private and very diverse in certain dimensions. For example, our analysis of the data collected from DragonLab[2] (Dataset 1 described in Section 5) suggests that different users access the Internet at different time and sequence (Section 2). The result shows that although 67.11% of the access behaviors are concentrated on only 2 of the 21 monitored destinations, the online activities of different users actually happen at very different times and in very different sequences.

[1] `http://www.edu.cn/cernet2_7948/`
[2] `http://dragonlab.org/`

As a result, users in Dataset 1 need no more than two consecutive packets to differentiate themselves from each other. If the mobile node can tell when and what it has done correctly, its identity can be verified.

Such evidence allows us to propose an authentic source address validation scheme in Mobile IPv6 based on the historical traffic of the mobile nodes. To implement this scheme, we first collect the mobile nodes' traffic at both the home agent and the mobile node, and then construct the consecutive packets of each mobile node in time as access path. We also employ some storage techniques to strengthen the security of the access path to defend against the eavesdropped traffic used by the attackers. Such access path can serve as authentication key to validate the ownership of the home address claimed by the mobile node after its move (Section 3). Theoretical analysis on performance shows that our scheme has light overhead and is able to complete the source address validation in a secure way(Section 4). Our experimental results with real traces demonstrate that the access path with only three consecutive packets can differentiate a mobile node in a subnet completely under reasonable settings (Section 5). Our scheme has the following advantages:

- Accountability: The mobile node can prove the ownership of its claimed home address efficiently using access path generated from its historical traffic.
- Light overhead: Without using cryptography, our scheme has no computational overhead. It has light storage overhead also because it needs a few packets for validation.
- Easy to implement: Leveraging the real traffic generated in the network, the implement and the deployment of the scheme is easy. Minimum data collection is required in this scheme.

The rest of the paper begins with the motivation for our research in Section 2. Section 3 describes the trustable communication architecture in Mobile Ipv6, the construction of the access path, and the source address authentication details. The performance and security analysis of our scheme are presented in Section 4. The applicability of the scheme is demonstrated in Section 5 with real traces. Section 6 introduces related work. Section 7 concludes the paper.

2 Problem Statement

Since it is a complicated and daunting task to generate a key to complete authentication with cryptography and PKI's support, we look for alternative ways in this paper.

As is commonly known, numerous historical traffic will be generated in the network when the nodes access the Internet. However, only information that satisfies the three criteria of privacy, uniqueness and anti-replay attack ability can be used for the authentication purpose. Our mission in this paper is to extract information from the historical traffic to generate the authentication key satisfying above three criteria.

Since it is impossible for attackers to monitor the network all the time, we can assume that the traffic is only known to the nodes that send it and the devices

that collect it. Even though some traffic may be eavesdropped by attackers, we can use some techniques to make them useless to attackers. In essence, we assume that the traffic information is not known to attackers in authentication.

It is a great challenge to ensure the uniqueness of the authentication key generated from the historical traffic because the traffic of different nodes can be same at the same time. However, our analysis based on real traces reveals that online activity differences always occur among individual users even though they may behave exactly the same at some time (Figure3 and Figure 8). For example, only 1.81% of 2856 users in Dataset 1 have exactly the same access behavior at some particular point of time, and all the nodes can be differentiated with access behaviors of two visits. Therefore, it is feasible to construct a unique authentication key according to time (access sequence) and space (packet composition information) properties of the traffic.

The ability of anti-reply attack is important to validation because attackers may eavesdrop the traffic in the network. In general, there are two classes of methods to protect the authentication key. One is using encryption, and the other is generating a new key whenever it is needed. The volume of the traffic information in the network allows us to take the second approach because the nodes usually generate a large number of packets every day.

In summary, it is possible to construct authentication key from mobile nodes' historical traffic that satisfying the three criteria of privacy, uniqueness, and anti-reply attack ability.

3 Access Path Based Trust Communication Architecture in Mobile IPv6

The purpose of this paper is to validate the home address claimed by the mobile node using its historical traffic after its move. There are three important issues to address. The first is what traffic we should collect, where to collect it and how to store it for easy access. The second is how to construct the authentication key. And the third is how to validate the home address of the mobile node according to the authentication key. We discuss these issues in this section.

3.1 Overview

Figure 1 presents our trustable communication architecture based on mobile nodes' historical traffic information. It consists of two modules. The first one is collecting module, which resides in the home agent and the mobile node. This module decides what information to collect and how to store it. The second one is communication module. This module accomplishes the mobile nodes' authentication after its move. It involves binding update authentication both to the home agent and to the correspondent node. We elaborate on the function of each module below.

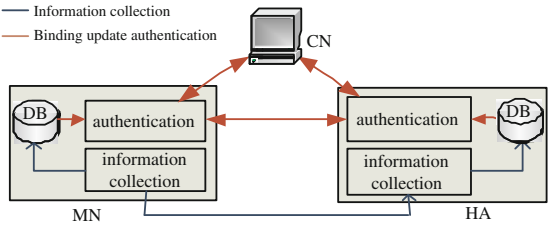

Fig. 1. Historical traffic based trustable communication architecture in Mobile IPv6

3.2 Information Collecting and Recording

As the basis of our scheme, traffic collection is a crucial part in our study. We think that the best location to collect and record the traffic is the home agent and the mobile node because of their positions. Packet is chosen as the collection unit because it is easy to collect by both the home agent and the mobile node. Because of the privacy and overhead issues, we will not collect packet payload in this paper.

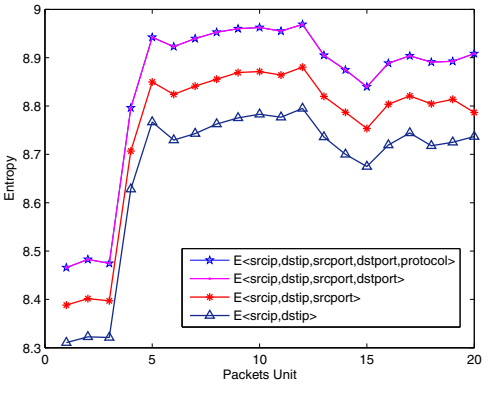

Fig. 2. The entropy of traffic which is composed of different tuples

Generally, we need use a 5-tuple <srcIP,dstIP,srcport,dstport,protocol> to denote a flow in the traffic. But in this paper, we only need use the 4-tuple <srcIP,dstIP,srcPort,dstPort> to construct the authentication key because we find that the information *protocol* has marginal effect in determining the distribution of the traffic. This can be seen from the entropy statistics of Dataset 2 shown in Figure 2. Entropy is a concept to measure the probability of particular information in information theory. It has been shown to contribute to network monitoring[6] recently. The smaller the entropy is, the more regular the information distribution is. Therefore, it is more difficult to distinguish between mobile nodes when the entropy is low. From Figure 2, we can see that the traffic's entropy of 5-tuples is almost identical to that of 4-tuples and the entropy of 4-tuples is bigger than the others. This suggests that the choice of 4-tuple is

the most suitable one to construct the authentication key because the packets of 4-tuples is much easier to differentiate. We will verify this conjecture with real data in Section 5.

To guarantee the consistency between the traffic collected in the home agent and that collected in the mobile node, we collect the traffic through the asynchronous transmission. We also focus on the packet order instead of the precise time in the procedure of authentication to further reduce the impact of time difference between the traffic in the home agent and that in the mobile node.

After determining the information to be collected, the home agent and the mobile node find a way to store the collected traffic in easy-to-access types. The traffic collected is usually stored in a database in the home agent. However, it is inconvenient to access such information if all users' traffic is stored in one table. So, each home agent in our work builds a table for each mobile node to record their traffic in time order. To reduce the access time, the home agent also creates a hash table to maintain the relationship between the mobile nodes and their traffic table. Compared with traffic storage in the home agent, the traffic storage in the mobile node is much simpler. Each mobile node only needs maintaining its own traffic table in time order.

As there is so much traffic in the network, it is impossible to collect all the packets generated by the mobile nodes. In our work, we use stationary space to store the traffic for each mobile node. Meanwhile, we apply several strategies to reduce the frequency and quantity of data collection.

First, we reduce the collection frequency by just collecting special data. Some measurements based on mobile circumstance find that the traffic generated by the mobile nodes largely focuses on individual ports and services[8,14], such as port 80 or web service. This allows us to collect the packets of special application, such as the packet whose destination port is 80. This collection strategy not only reduces the storage overhead but also reduces the risk of attackers' eavesdropping, as attackers need more time and energy to get the special application information they want.

Second, we update the storage depending on the availability of the space and the frequency of collection. Once the storage space is full, the collection will be stopped temporarily. So, even if the nodes do not access the Internet for a long time, or they have a prolonged absence from the local subnet, the traffic records will be sustained unless the space is full and new collection begins. This collection method increases the difficulty of attackers' eavesdropping because they do not know what information to collect and what sequence to store it.

3.3 Access Path Construction

As shown in Figure 3, there may be many nodes in a subnet and they may exhibit the same online activities at a particular point of time. Therefore, it is insufficient to use just one packet to differentiate different users. However, we can combine the time and space properties of the traffic to construct the authentication key. In order to do so, we define the list of consecutive packets from one mobile node N as its access path indicated by

$Apath_{N,timestart} = \{(number_i, P_i)\}$, where,
$P_i = \{srcIP_i, dstIP_i, srcPort_i, dstPort_i\}$.

The parameter *timestart* represents the collection time of the first packet in the access path. The parameter $number_i$ indicates the sequence number of the packet in the storage. It is an auxiliary parameter to strength the security of the authentication key in the construction of the access path. The parameter P_i is used to denote the packet in the list and it is represented by a 4-tuple. P_1 is named starting point of the access path. We call the length of the access path access hop in our paper, which is equal to the number of packets in the list. The most important question for our architecture is how many hops we should use to construct a unique access path. We answer this question in Section 5.

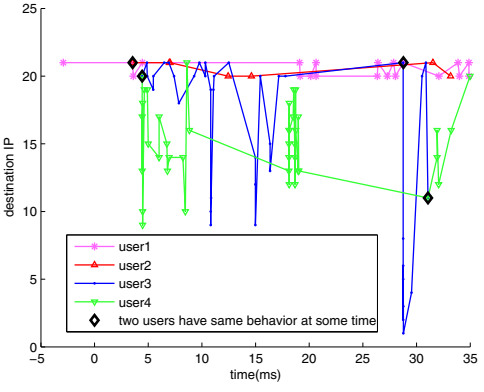

Fig. 3. The map sites access path of five users in some time interval in Dataset 1

3.4 Access Path Based Authentic Source Address Validation

The access path based source address validation includes two parts, as shown in Figure 4.

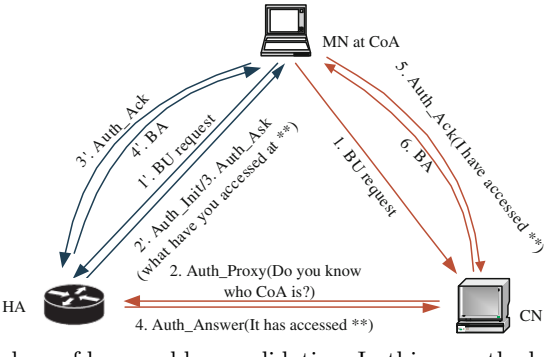

Fig. 4. The procedure of home address validation. In this way, the home agent and the correspondent node can authenticate the binding update and distinguish the ownership of the home address claimed by the mobile node.

In the first part, the home agent validates the home address of the mobile node once it receives a binding update request from the mobile node. It searches through traffic records table according to the home address claimed by the mobile node, constructs a challenge question and sends it to the mobile node with message Auth Init as shown in Figure 4. After the mobile node gets the question, it looks for the answer in its own storage table, builds the corresponding authentication key, and returns the key back to the home agent through message Auth Ack to prove its identity. The home agent can compare the answer from the mobile node with the records in its own table to validate the identify of the mobile node.

In the second part, the correspondent node needs to validate the identity of the mobile node as well after it receives a binding update request from the mobile node. The authentication to the correspondent node is more complicated because it has no records of mobile nodes' historical traffic. Usually, the correspondent node is a server that has large access volumes. The cost of monitoring and collection on the correspondent node is very expensive. However, the correspondent node can depend on the home agent to complete validation. The procedure is shown in Figure 4. After the correspondent node receives a binding update request, it sends an authentication request to the home agent through message Auth Proxy. The home agent then sends a new challenge question to the mobile node through message Auth Ask. At the same time, the home agent also sends the correspondent node the hash value of the right answer to the challenge question. After the correspondent node receives the answer from the mobile node through message Auth Ack, it computes the hash value of the answer and compares the result with that from the home agent. Such comparison allows the correspondent node to distinguish the identity of the mobile node.

How to construct the challenge question is an important task in authentication. Since our validation is based on the access path, it is easy to construct a challenge question. For example, we can ask "what information do you have between number 5 and 8?" or "what did you do after you accessed the 120.23.*.* at port 80 at 11:00?". Since neither storage order nor the accurate content can be known by the attacker, a right answer must come from the right mobile node. Of course, we can ask more than two questions to strengthen the security.

4 Performance and Security Analysis

4.1 Storage overhead

We illustrate the storage overhead of our scheme using the example of our campus network (see Section 5). Suppose all 48,000 users in the campus network are mobile nodes, and each packet record in our case is 40B. The relationship between the storage overhead and the number of packet records for each mobile node is shown in Figure 5.

From Figure 5, we can see that only 0.076MB of overhead is needed to store 2000 packet records for each mobile node, and only 3.58G of overhead is needed to store all the packet records for all mobile nodes in the home agent. Obviously, this

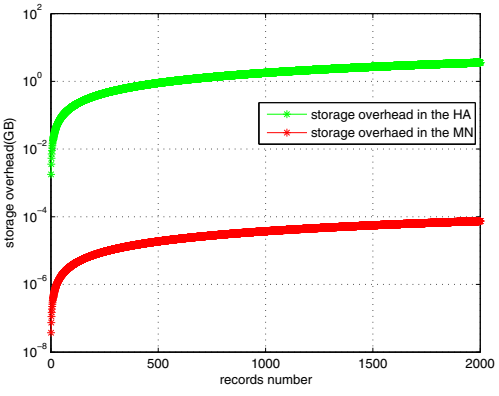

Fig. 5. Storage overhead in the home agent and the mobile node

light storage overhead will not lead to overhead burdens on the home agent or the mobile nodes. Since our experimental results show that only three consecutive packets are sufficient to complete validation (see Section 5.2), the actual overhead in real practice should be much smaller than that in Figure 5.

4.2 Security Analysis

Attackers are most likely to eavesdrop at three locations in the network: on the path between the home agent and the correspondent node (HA-CN path), on the path between the home agent and the mobile node (HA-MN path), and on the path between the mobile node and the correspondent node (MN-CN path). We analyze the security of our scheme under scenarios of attacks to these three paths.

- On HA-CN path: The only useful message to the attacker on the HA-CN path is the hash value of the authentication key generated by the home agent. Even if the attacker is able to get the hash value, it is almost impossible for the attacker to decode the original authentication key from it.
- On HA-MN path: In our scheme, the messages transmitted on the HA-MN path include the mobile nodes' traffic, the binding update request, the challenge question and the authentication key. The authentication key is useless for attackers, because it changes all the time. The mobile nodes'traffic is also useless for attackers, because the attackers cannot know the sequence number of the packets required to construct the access path unless they collect the traffic continuously from the very beginning. But this is impossible.
- On MN-CN path: The eavesdropping on the MN-CN path is not a concern to our scheme because that the authentication key will change in every communication under our scheme.

4.3 Feasibility Analysis

The authentication key used in our scheme comes from real traffic data. No additional device or software is needed except for a collection module. As a

result, it is simple to deploy our scheme in practice. Moreover, most core routers have collected the historical traffic passing by for the purpose of measurement and management. Our scheme can use the data collected by routers directly, or add some additional features to the collecting function of routers without introducing too much overhead.

5 Experimental Study

In this section, we demonstrate the optimal choice of access hops in the construction of authentication key. We use two datasets collected from Campus Network of Tsinghua University (TUNET), which serves for about 40,000 hosts and 48,000 connected end users. TUNET connects to the CERNET (China Education and Research Network) that connects to the Internet through a 2 Gbps full-duplex link.

One dataset was collected from the platform of DragonLab (Dataset 1). It monitored the Web/HTTP traffic of TUNET that interacts with online map sites including Google Map, Microsoft Live Maps, Yahoo Maps, Baidu Maps, Koubei Maps, Sogou Maps, Edushi and Dushiquan. This dataset consists of about 2GBs of map traffic data for the 18 days between April 7^{th} and 24^{th} of 2009.

The other dataset was collected by a core router in TUNET (Dataset 2). It includes all the packets without payload for ten minutes between 16:40 and 16:50 on December 7^{th} of 2006. The size of the data is around 2GBs.

Notice that these datasets are used for methodology analysis, other datasets with packet records can also be used.

5.1 Effectiveness Evaluation Algorithm

Before determining how many access hops are needed to construct a unique authentication key, we first propose a standard used to evaluate the effectiveness of an access path. In order to do so, we design an access path based forest through three steps (To strengthen the security, we do not consider the packet sequence number of the access path in the construction of the forest):

– Create an access path graph for the nodes that have the same behaviors at the same point of time.
  We first create a graph to cluster all the nodes that have the same behaviors at the same point of time. In the graph, each point (node) represents a packet in the access path, which is denoted by Pi; and each directed link reflects the sequence of packets. We illustrate the graph construction with an example. Suppose we need to validate a mobile node N1. First, an access path $Apath_{N1,t0} = \{P1, P2, P3, P4\}$ is chosen from the home agent's records table randomly and a graph is constructed as shown in Figure 6. Next, all the nodes that have the same packet $P1$ around the same point of time are extracted from the database. Suppose four nodes are found and their access paths are $Apath_{N2,t0} = \{P1, P2, P3, P5\}$, $Apath_{N3,t0} = \{P1, P2, P6, P7\}$,

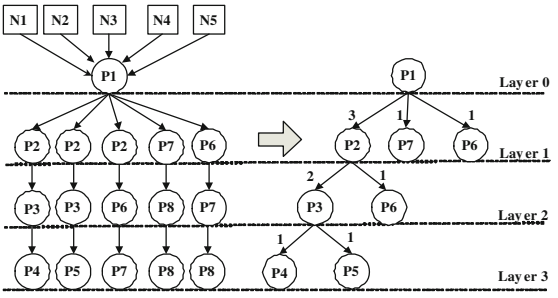

Fig. 6. The construction of the access path graph and forest. Each node represents a packet and the directed link denotes access order.

$Apath_{N4,t0} = \{P1, P7, P8, P8\}$ and $Apath_{N5,t0} = \{P1, P6, P7, P8\}$ respectively. These access paths are then inserted into the directed graph one by one as shown in Figure 6. In order to improve the security, we do not require other nodes to access $P1$ exactly at $t0$ as node N1 does, but allow other nodes to access $P1$ between $t0 - t$ and $t0 + t$, where t is a threshold (We use 0.5 second as the threshold in this paper).

– Construct the access path forest. The forest can be constructed according to the following steps:
  First, we make layers for the access path graph shown in Figure 6. The layer of the starting point $P1$ is defined as layer 0. The layer number is one less than the corresponding access hops.
  Second, we set the starting point of the access path as the origin node of the forest and define the number of mobile nodes in the access path graph as the in-degree of the origin node. As shown in Figure 6, the in-degree of the point $P1$ is 5.
  Third, we consolidate the points in the graph layer by layer to construct the forest. The same points at the same layer in the graph are consolidated into one point in the forest at the same layer. The in-degree of the point in the forest is the number of the mobile nodes that have accessed this point at the same layer of the graph. In effect, the in-degree of the point means the number of users that have the same starting point and the same access path up until this point. We call points whose in-degree is one as "leaf nodes", and we do not add points under them in the next layer of the forest. Point $P6$ in Figure 6 is an example of leaf point.
– Compute the effectiveness of the access path.
  The ultimate goal of constructing the access path forest is to determine how many hops are needed to construct an effective authentication key. Based on the in-degree of the points in the forest, the effectiveness of the access path with hops i can be defined as follows:

$$Disc_i = \begin{cases} 0 & (\text{if i=1 \& N>1 }); \\ 1 & (\text{if i=1 \& N =1}) ; \\ \frac{\sum_{k=1}^{(} i-1)(leaf_k)}{N} & (\text{if i>1}); \end{cases} \quad (1)$$

In formula (1), the parameter N represents the in-degree of the origin node, the parameter $leaf_k$ denotes the number of leaf points at layer k, and $Disc_i$ defines the effectiveness of the access path with i hops. The result of $Disc_i$ allows us to find the most suitable access hops needed for authentication purpose. For example, the effectiveness of the access path in Figure 5 is illustrated in formula (2). The result suggests that the access hops must be at least 4 in order to construct an authentication key to identify node N1 or differentiate all the mobile nodes. However, the number of access path hops can be less under certain circumstances.

$$Disc_i = \begin{cases} 0 \quad (\text{i=1 }); \\ 40\% \quad (\text{i=2 }) \ ; \\ 60\% \quad (\text{i=3}); \\ 100\% \quad (\text{i=4}); \end{cases} \tag{2}$$

5.2 Choice of Access Hops

The authentication key constructed with different information may have different effects on validation efficiency. In this section, we detect the optimal number of access hops under different tuples to construct the authentication key. To evaluate the impact of alternative choices of tuples on authentication key, we change the definition of packet from 2-tuple to 4-tuple to calculate the distribution of mobile nodes' traffic for both Dataset 1 (as shown in Figure 7(a)) and Dataset 2 (as shown in Figure 7(b)). In both Figures, each column in the m-tuple group represents the frequency of a m-tuple packet re-appear (does not consider $srcIP$ information since it is the producer of the packet). In fact, it reflects the number of mobile nodes that have the same behaviors at the same point of time. For example, the first column of the first group in Figure 7(a) stands for the number of unique 2-tuple <srcIP,dstIP> packets in Dataset 1; and the second column stands for the number of 2-tuple <srcIP,dstIP> packets that is common to 2

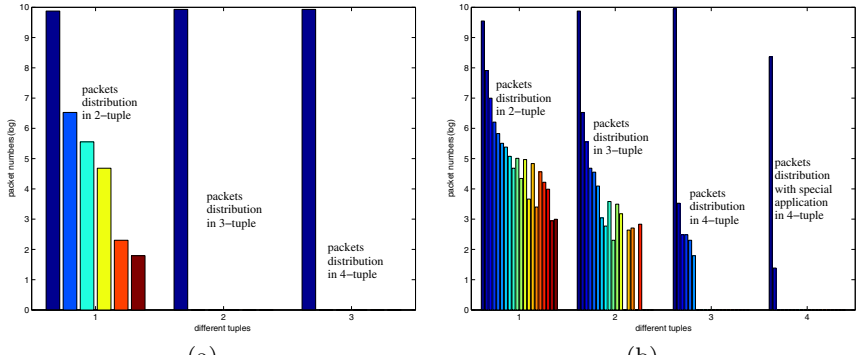

Fig. 7. Packets distribution under different tuples in Dataset 1(a) and Dataset 2(b)

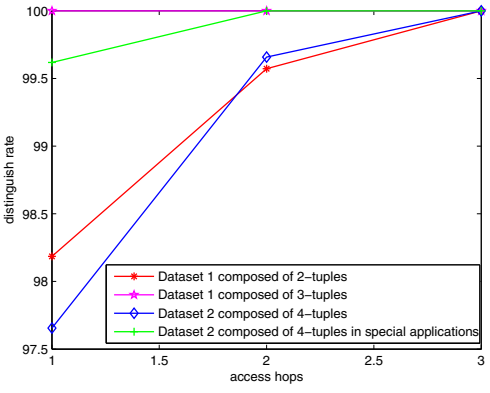

Fig. 8. Nodes' discrimination rate in different hops

different mobile nodes in their access history at some point of time. Apparently, the 4-tuples packets have a lower probability to re-appear than others.

Using the effectiveness evaluation algorithm discussed earlier, we compute the smallest number of access hops needed to construct the authentication key as the choice of the tuples changes for both Dataset 1 and Dataset 2. The result is shown in Figure 8.

We can see that most nodes (98.19%) in Dataset 1 can be distinguished by one 3-tuple <srcIP,dstIP,dstPort> packet, and all nodes can be distinguished by one 4-tuple <srcIP,dstIP,srcPort,dstPort> packet. For Dataset 2, at most three access hops are needed to construct a unique authentication key. We can also notice from Figure 8 that the access hops needed for Dataset 2 is much longer than that for Dataset 1. This is because the traffic in Dataset 1 is more special. In general, the access hops based on special application traffic is always much shorter. However, it may need more time to collect.

6 Related Work

The IETF uses the IPsec[2] to assure the authenticity of the binding update to the home agents. It is secure in theory. However, it has not been widely deployed because of its complexity and computational overhead.

The IEFT has also proposed return routability (RR)[7] mechanism to overcome the security vulnerability introduced by the route optimization. The RR mechanism uses reachable check to assure the right mobile node is sending the binding update message. When the mobile node moves to a new location, it sends two Init messages to the correspondent node through two paths. One is sent directly to the correspondent node. The other is sent via the home agent. The correspondent node generates two tokens for these two Init messages according to a secret only known by himself and sends them back to the mobile node through the same two paths. After that, the mobile node sends back a

binding management key (named kbm in IETF) that is constructed from these two tokens to the correspondent node to prove its identity. Although it is difficult for attackers to intercept on two paths at the same time, some have managed to do so[13]. [9,12,15] propose some improvements to RR mechanism, but they have not been able to make it applicable to a larger extent.

Our scheme shares some similarities with RR mechanism in design. Both RR mechanism and our scheme rely on the home agent to help the correspondent node to complete the binding update authentication. However, there are two essential differences in our scheme. First, the source of the authentication key is different in these two mechanisms. It is generated by the mobile node in our scheme but generated by the correspondent node in RR mechanism. This makes our scheme more secure than RR mechanism. For example, if the attackers manage to eavesdrop on the HA-CN path, they are able to obtain one token of the RR mechanism. If they obtain the other token directly from the correspondent node, they can generate kbm to win the correspondent node's trust. While our scheme has no such kind of risk. Second, the implementation of RR mechanism requires the support of the security tunnel between the home agent and the mobile node. However, the security tunnel such as IPsec has not been deployed widely because of its complexity. As a result, RR mechanism is not widely employed although the concept of security tunnel is theoretically sound. In the contrast, our scheme does not require the support of a pre-created security tunnel.

The idea of accountability is widely applied in IPv6 to validate the identity of the nodes, such as in AIP[1], CGA[3] and HIP[11]. These mechanisms often bind the source address onto public key to generate a self-certification address, and most of them support mobility as well. However, the computational complexity caused by encryption limits their deployment. For example, the CGA(Cryptographically Generated Addresses) is composed of subnet prefix (64 bits) and interface identifier (64 bits) that is generated by two hash changes according to the address owner's public key and some CGA parameters. The corresponding private key and the interface identifier that is unchanged no matter how host roaming assert the senders identity in mobility. Unfortunately, the time required for generating a valid high secure CGA address is overwhelming from a practical perspective[4]. In this paper, the scheme we proposed also builds on the notion of accountability. However, our scheme only requires existing data in the network, which solves the overhead and deployment issues of traditional cryptography-based mechanisms.

7 Conclusion

In this paper, we propose an authentic source IP address validation scheme in Mobile IPv6. Using historical traffic generated by the mobile nodes, our methodology generates a unique authentication key to validate the home address claimed by the mobile node after its move. This key is a sequence of access paths composed by 4-tuple <srcIP, dstIP, srcPort, dstPort> packets. It is difficult for attackers to forge because the access path and the sequence number of the packets

in the storage is only known by mobile nodes and the home agent. In addition, any effort from attackers to extract authentication key is meaningless because authentication key is changing all the time.

Our experiment with real traces demonstrates that a unique authentication key can be constructed using only three 4-tuple <srcIP, dstIP, srcPort, dstPort> packets. Compared with other cryptography-based methods, our scheme consumes no computational overhead and less storage overhead. Most important of all, it is easy to deploy using available data existing in the network.

Acknowledgment. We are greatly indebted to Wenlong Chen for fruitful discussion. We also owe our deepest gratitude to Bingqing Xu and Kaijun Zhang for data collection support.

References

1. Andersen, D., Balakrishnan, H., Feamster, N., Koponen, T., Moon, D., Shenker, S.: Accountable internet protocol(aip). In: Proceedings of ACM SIGCOMM (2008)
2. Arkko, J., Devarapalli, V., Dupont, F.: Using ipsec to protect mobile ipv6 signaling between mobile nodes and home agents. RFC 3776 (June 2004)
3. Aura, T.: Cryptographically generated addresses(cga). RFC 3972 (March 2005)
4. Bos, J.W., Özen, O., Hubaux, J.: Analysis and optimization of cryptographically generated addresses(cga). In: Proceedings of ISC (2009)
5. Elgoarany, K., Eltoweissy, M.: Security in mobile ipv6: a survey. Information Security Technical Report 12(1), 32–43 (2007)
6. Hu, Y., Chiu, D.-M., Lui, J.C.S.: Entropy based adaptive flow aggregation. IEEE/ACM Transactions on Networking(TON) 17(3), 115–139 (2009)
7. Johnson, D.B., Perkins, C., Arkko, J.: Mobility support in ipv6. RFC 3775 (June 2004)
8. Kivi, A.: Mobile data adoption in finland 2005-2006. In: Proceedings of the 6th Conference on Telecommunication Techno-Economics(CTTE), Helsinki, Finland (June 2007)
9. Li, J., Zhang, P., Sampalli, S.: Improved security mechanism for mobile ipv6. International Journal of Network Security 6(3), 291–300 (2008)
10. Mankin, A., Patil, B., Harkins, D., Nordmark, E., Nikander, P., Roberts, P., Narten, T.: Threat models introduced by mobile ipv6 and requirements for security in mobile ipv6. IETF draft-ietf-mipv6-scrty-reqts-02.txt (2001)
11. Moskowitz, R., Nikander, P.: Host identity protocol (hip) architecture. RFC 4423 (May 2006)
12. Nikander, P., Aura, T., Arkko, J., Montenegro, G.: Mobile ip version 6 (mipv6) route optimization security design. In: Proceedings of the IEEE Vehicular Technology Conference Fall 2003 (2003)
13. Ren, K., Lou, W., Zeng, K., Bao, F., Zhou, J., Deng, R.H.: Routing optimization security in mobile ipv6. Computer Networks: The International Journal of Computer and Telecommunications Networking 50(13), 2401–2419 (2006)
14. Riikonen, A.: Mobile internet usage - network traffic measurements. Master's Thesis. Department of Communications and Networking, Helsinki University of Technology, Espoo (September 2009)

15. Song, S., Choi, H.-K., Kim, J.-Y.: A secure and light weight approach for routing optimization in mobile ipv6. EURASIP Journal on Wireless Communications and Networking (2009)
16. Wu, J., Bi, J., Li, X., Ren, G., Xu, K., Williams, M.: A source address validation architecture (sava) testbed and deployment experience. RFC 5210 (June 2008)
17. Wu, J., Ren, G., Li, X.: Source address validation: Architecture and protocol design. In: Proceedings of ICNP (2007)