Mobility-Based Anomaly Detection in Cellular Mobile Networks

Bo Sun
Dept. of Computer Science
Lamar University
Beaumont, TX 77710
bsun@cs.lamar.edu

Fei Yu
Dept. of Electrical and Computer Engineering
University of British Columbia
BC, Canada V6T 1Z4
feiy@ece.ubc.ca

Kui Wu
Dept. of Computer Science
University of Victoria
BC, Canada V8W 3P6
wkui@cs.uvic.ca

Victor C.M. Leung
Dept. of Electrical and Computer Engineering
University of British Columbia
BC, Canada V6T 1Z4
vleung@ece.ubc.ca

ABSTRACT

This paper presents an efficient on-line anomaly detection algorithm that can effectively identify a group of especially harmful *internal* attackers - *masqueraders* in cellular mobile networks. Our scheme is derived from a well-developed data compression technique. We use cell IDs traversed by a user as the feature value. Based on this, the mobility pattern of a user is characterized by a *high order Markov model*. Ziv-Lempel data compression algorithms are utilized to parse the data and store relevant statistical information in a *mobility trie*. Moreover, the technique of Exponentially Weighted Moving Average (EWMA) is used to efficiently update the mobility trie in order to modify the user's normal profile constantly. In this way, an up-to-date normal profile is maintained. The proposed normal profile can characterize the normal behavior of each user accurately and is sensitive to abnormal changes. A *threshold* scheme is then used to determine whether the mobile device is potentially compromised or not. Simulation results demonstrate that our proposed detection algorithm can achieve good performance in terms of false alarm rate and detection rate for users having regular itineraries.

Categories and Subject Descriptors

C.2 [**Computer Systems Organization**]: Computer Communication Networks; C.3 [**Computer Systems Organization**]: Special-Purpose and Application-Based Systems; I.6 [**Computing Methodologies**]: Simulation and Modeling

WiSe'04, October 1, 2004, Philadelphia, Pennsylvania, USA.

General Terms

Design, Security, Performance

Keywords

Cellular Mobile Networks, Anomaly Detection, Data Compression

1. INTRODUCTION

With the introduction of data services into cellular based mobile wireless networks, people begin to rely on cellular phone in their daily lives for important and sensitive tasks such as E-shopping and E-banking. The booming new services, while bringing great convenience, have naturally caused serious security concerns. Although there are many authentication protocols in cellular mobile networks, security is still a very challenging task due to the open radio transmission environment and the physical vulnerability of mobile devices.

In general, two complementary approaches exist to protect a system: *prevention* and *detection*. Prevention based techniques, like authentication and encryption, can effectively reduce attacks by keeping illegitimate users from entering the system. They are usually based on some symmetric and asymmetric mechanisms to assure that users conform to predefined security policies. Nevertheless, in cellular wireless networks, the mobile devices are not physically secure: they could be lost or stolen. Since tamper-resistant hardware and software are still too costly for most users, such insecurity makes all secrets of the device open to malicious attackers. An attacker, once possesses the device as well as all secrets associated with the device, he becomes an internal user and is able to cause great damage to the whole network. All prevention-based methods will be rendered helpless in this situation. At this time, Intrusion Detection (ID) approaches, utilizing different techniques to model the users' normal behavior and system vulnerabilities, come into place to help identify malicious activities.

Generally, there are two intrusion detection techniques, *misuse* based detection and *anomaly* based detection [5]. A misuse based detection technique encodes the known attack

signatures and system vulnerabilities. If it finds a match against current activities, an alarm is generated. Misuse detection techniques are not effective to detect novel attacks. An anomaly based detection technique creates normal profiles of system states and user behaviors and compares them against current activities. If a significant deviation is observed, an alarm is triggered. Anomaly detection can detect unknown attacks. However, the normal profiles are usually very difficult to build. This is especially true for cellular mobile networks due to the mobility of end users. Therefore, how to establish normal profiles for mobile users is crucial in designing an efficient intrusion detection algorithm.

Our work is based on such a belief: the mobile device and all its secrets can be possessed by an attacker who in turn can do whatever he wants without being caught (unless the authentic owner claims the loss of device in advance). But the attacker cannot change his personal mobility patterns to the same as that of the authentic user, since a user's mobility pattern is a reflection of the routines of his life and different mobile users have different favorite routes and habitual movement patterns. In this paper, on the basis of optimal data compression [2] [24], we propose a novel approach to construct the normal profile of a mobile user, from which an efficient detection algorithm is designed. The list of cells traversed by a mobile user is used as the feature value. When an intrusion occurs, the attacker *masquerading* the legitimate user tends to have a different movement pattern. Therefore, we can detect anomaly by comparing the movement patterns.

For most users, movement patterns can be captured and modeled, and such patterns have been used broadly in improving the performance of QoS provision and resource allocation [23]. Nevertheless, there is a certain number of users such as taxi drivers who do not exhibit regular movement. It will be very hard to model those users' movement patterns. In addition, it is normal for people to occasionally change their normal routines. For example, people on vacation may exhibit significant deviation from their normal movement patterns. To summarize, we should not expect that the detection based on mobility patterns is accurate for all users in all situations.

Our research is not motivated to build a system to accurately detect intrusions. Instead, we are aimed at providing an optional service to end users as well as a useful administration tool to the service provider. The attacker can cause a huge loss for the authentic owner if the attacker makes many long distance calls before the real owner discovers the loss. Because of this reason, the real owner might need some warning information via other channels (e.g., email, phone call to home) if the system observes some abnormal behavior. Such warning could be something like "we observe that you are having a significant change of movement patterns. Is your handheld still safe?" We believe such an optional service will be popular. For the service provider, the system can build a "gray list" to include the users who exhibit dramatic changes of movement patterns. The traffic patterns and the behavior of the users in the "gray list" need to be monitored with caution. As long as they try to issue some dangerous commands to the network, immediate response is required to avoid potential financial loss. The "gray list" should be updated dynamically. For instance, a person who leaves on holidays may be added into the "gray list" but will be removed when he resumes his normal routines.

Our algorithm is derived from data compression techniques [2] [24] that are both theoretically optimal and good in practice. It has been demonstrated that data compression is synonymous with prediction. The cells traversed by a user during his or her call forms a string, which is modeled as a *high order Markov source*. The string is parsed into phrases. A mobility trie, or a multiway tree, is constructed on-line to record these phrases efficiently. Based on this, we can construct an on-line probabilistic model of user activities. In the meantime, the technique of Exponentially Weighted Moving Average (EWMA) is used to exponentially fade the probabilistic model in order to make the long-term profile up-to-date. In the on-line working phase, the user mobility pattern is observed in terms of cell number. Finite-context models are used to compute the probability of a particular symbol based on the sequence of characters immediately preceding it. A *blending* strategy is applied to handle the impacts of models of different orders and compute the normal probability of current activity. A *threshold* scheme is then utilized to decide whether current activities are abnormal or not. We also present simulation results to demonstrate the effectiveness of the proposed approach.

The rest of the paper is organized as follows. In Section 2, we present our assumptions in developing detection techniques for cellular mobile networks. Section 3 describes the threat model, network model, and mobility model. Section 4 presents the details of constructing the detection algorithm, including the methods of modeling the users' mobility pattern, constructing a mobility trie, and calculating the probability of the user's activity. The analysis of our algorithm is also provided. Section 5 presents the simulation study of our proposed detection approach. Section 6 describes the relevant work. In Section 7, we conclude the paper and point out future work.

2. ASSUMPTIONS

Our detection algorithm relies on the following assumptions:

First, we assume that each mobile user has a mobility database that describes his normal activities. In a cellular mobile network, this mobility database is stored together with the mobile user's personal information, such as billing information, in the Home Location Register (HLR). We assume HLR is secure and the movement information is accurate. Usually, because of its importance, HLR is protected with highly secure measures, and thus it is extremely hard to attack HLR. Also, the update of location is usually based on the device's current serving cell and the hardware registration such as the series number of SIM card. It will be hard for the attacker to hide or fabricate his location if he uses the captured mobile device. Even if an attacker finds some magical way to fabricate his location, he still has no idea what is the normal movement profile of the real device owner.

Second, we assume mobile devices can be compromised and all secrets associated with the compromised devices are open to attackers. This assumption is reasonable since currently tamper-resistant hardware and software are still costly to handheld devices. This assumption justifies our research in anomaly detection, since all prevention techniques will be rendered helpless once the mobile device is captured and compromised.

Finally, we assume most users have favorite or regular

itineraries. Therefore, it is viable to create the normal movement profile for each user. Actually, all research on intrusion detection is based on two assumptions: 1) the subject activity is observable, and 2) the normal and malicious activities demonstrate distinct behavior. If a user has totally random behavior, for example, the movement of a taxi driver, it will be very difficult, if not impossible, to create his normal movement profile. For such type of users, our current feature selection based on mobility patterns is inaccurate. Nevertheless, our method is automatically user-selective since the optional warning service mentioned in the introduction section will tend to give many false warning messages to this type of users and force them to unsubscribe such service.

3. MODEL DESCRIPTION

3.1 Network Model

Most of the previous work on wireless cellular networks uses structured graph network topology models, such as hexagonal or square cell configurations. However, these models may not accurately represent a cellular network in practice, where the cell shape and size may vary depending on the antenna radiation pattern and propagation environment. In wireless cellular networks, each cell usually has a base station to serve it. Therefore, in our system, the wireless cellular network is modeled as a generalized graph $G = (V, E)$. The vertex set V represents all the base stations. If two cells are adjacent to each other, there is an edge between their two vertices. An example of the model is illustrated in Fig. 1 and Fig. 2.

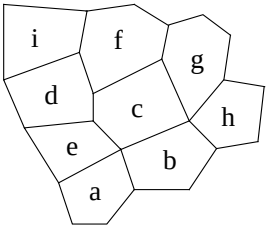

Figure 1: An Example of Cellular Network.

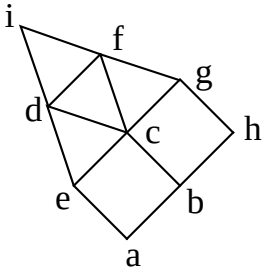

Figure 2: The Graph Model of Cellular Network.

In this example, the vertex set is $V = \{a, b, c, d, e, f, g, h, i\}$, and the edge set is $E = \{(a, b), (b, c), ...(f, i)\}$.

3.2 Mobility Model

The *random walk model* has been widely used in the literature, in which a mobile user will move to any one of the neighboring cells with equal probability after leaving a cell. This may not be realistic in practice, since mobile users normally travel with a destination in mind. Therefore, we use the *m-th order Markov model* in this paper. In such a model, the mobility of a user can be represented by a sequence of symbols, $C_1, C_2, C_3, \ldots, C_i, \ldots$, where C_i denotes the identity of the cell visited by the mobile. Since the future locations of the mobile are likely to be correlated with its movement history, the sequence of symbols $C_1, C_2, C_3, \ldots, C_i, \ldots$ is assumed to be generated by an m-th order Markov source, where the states correspond to the contexts of the previous m symbols. The probability that the user moves to a particular cell depends on the location of the current cell and a list of cells recently visited.

4. MOBILITY PREDICTION BASED ANOMALY DETECTION

In this section, we introduce the construction of the probabilistic prediction algorithm with user's mobility patterns. The anomaly detection problem is to characterize the behaviors of each individual in terms of temporal cell sequences. The mobility database of each specific user with regular itineraries, the mobility trie, is constructed from the accumulative history of the user's movement pattern. The recent normal profile of the user is built by applying EWMA techniques to the mobility trie. This modified mobility trie will serve as the normal profile of the user in the recent past. It reflects the *stationary* part of the user's regular mobility pattern. Based on this, we use a *blending scheme* to calculate the probability of each user's activity. The whole scheme is illustrated in Fig. 3.

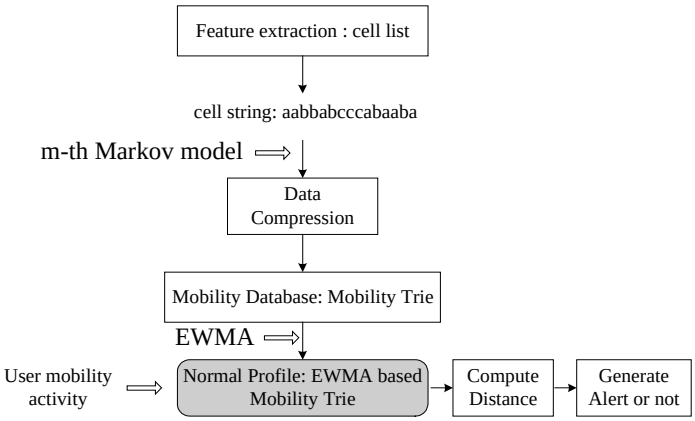

Figure 3: General Strategy of Mobility Prediction Based Anomaly Detection.

Similar approaches have been used in [1] to solve the location management problem and also in [23] to solve the call admission control and bandwidth reservation problem. However, the constructed mobility database cannot be used in the field of intrusion detection effectively because it does not take into consideration mechanisms to reflect the recent activities of the subject. We also derive a new *Distance*

measure that could provide good criteria to evaluate the normalcy of the user's itineraries.

4.1 Feature Extraction

The first step in intrusion detection is to extract effective features. Features are security related measures that could be used to construct suitable detection algorithms. Effective features must be selected to reflect the subject activities. For example, the short sequence of system calls of privileged program is stable and used to construct detection models with good performance [25]. In our environment, we build the normal profiles of mobile users with regular movement patterns in cellular mobile networks. Under the assumption that each user will have his own favorite itineraries, cell numbers traversed by each user is an ideal candidate feature for our usage. It is relatively stable and the resulting alphabet is small. To be specific, we denote each cell as a symbol. Therefore, a string could represent the path taken by a user. This string will feed into our model to construct the mobility trie.

4.2 Optimal Data Compression

Ziv-Lempel algorithms [2] have been widely used in data compression. Since its invention, many variations have been developed. $LZ78$ is one of the most popular one. It is both theoretically optimal and good in practice. Being a character based Ziv-Lempel algorithm, $LZ78$ parses the input string in a greedy manner and breaks the input string (i.e. a cell list) S into variable length phrase $x_1, x_2, ..x_n$ with the following property: for $j > 1$, there exists a number $i < j$, which makes x_j equal to x_i plus some character c, where c is one character in the alphabet. This is called the *prefix property* [2]. A *mobility trie* is suitable to store the parsed phrases.

A trie is a multiway tree with a path from the root to a unique node for each string represented in the tree. In a trie, only the unique prefix of each string is stored because the suffix can be determined by searching the string. A longest match is found by following down the tree until no match is found, or the path ends at a leaf.

Here is an example of how to construct a mobility trie. Suppose the alphabet is (a, b, c), and the string is *aabbabcccabaaba*. Based on the greedy parsing manner, this string will be parsed into $(a)(ab)(b)(abc)(c)(ca)(ba)(aba)$. After the whole string is processed, the constructed mobility trie is shown in Fig. 4. The number associated with each node indicates the frequency this node has been parsed in the construction of the mobility trie.

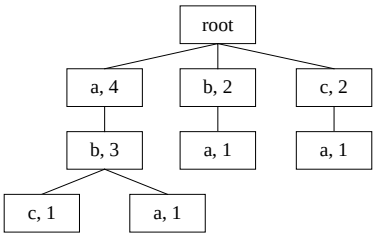

Figure 4: An Example Mobility Trie.

This trie characterizes the transition probability of the string *aabbabcccabaaba*. Given a trie and an order, we can calculate the probability of a given string from the mobility trie. For example, in Fig. 4, there are 4 a from the root. Therefore, the probability of a at the root is $\frac{4}{8} = \frac{1}{2}$. Similarly, the probability of b and c at the root is both $\frac{1}{4}$. The probability of b from a is $\frac{3}{4}$. We will demonstrate later how to use this probability to measure the normalcy of a string.

4.3 Probability Calculation

The probability calculation is based on the Prediction by Partial Matching (PPM) [4] scheme. Here, we use a context model to predict the next symbol based on the previous consecutive symbols. Specifically, we use a *m-th Markov model* to model the sequence. That is, we use the consecutive previous m symbols to predict the next symbol and calculate its probability.

We have a tradeoff here. If the order m is too small, the prediction will be poor in the long run because little of the audit data will be available to make decision. However, if the order is too large, most contexts will seldom happen, and initially the probability estimation will have to solely rely on the resolve of zero-frequency problems [2]. Based on these considerations, we take a *blending* approach, where the predications of several contexts of different lengths are combined into a single overall probability. It uses a number of models with different orders to compute the probabilities respectively and assign a weight to each model and calculate the weighted sum of the probabilities.

Let's denote the maximum order as m. The next character, denoted by α, is predicted on the basis of previous i characters. For each character α, let $p_i(\alpha)$ be the probability assigned to α by the finite-context model of order i. Note that when i is zero, the probability of each character is estimated independently of other characters. If the weight given to the model of order i is w_i, the blended probability $p(\alpha)$ is computed as:

$$p(\alpha) = \sum_{i=0}^{m} w_i * p_i(\alpha)$$

where the sum of weights should be normalized to 1. Generally, the larger the order, the larger the weight assigned to it, because context models with larger orders tend to be more accurate and should weight more in the current normal profile. The maximum order m and the weight w_i are design parameters. We will discuss them later in Section 5.

4.4 Anomaly Detection Algorithm

We take a data compression based method to deal with the anomaly detection problem, which trains a classifier with known "normal" data to distinguish normal from anomalous behaviors.

4.4.1 Integration of EWMA into Mobility Trie

In anomaly detection, each subject has a normal profile. Even for an individual subject, its activity may change over time. Therefore, it is necessary for the normal profile to be updated in order to reflect the recent activities in time. In our situation, the normal profile of the user activity should be dynamic. Generally, activities in the recent past should weight more than activities long time ago. A suitable mechanism should be applied to adaptively modify the normal profile correspondingly.

Based on the above considerations, we integrate EWMA

[26] to the mobility trie constructed above. The mobility trie is modified when a new phrase is formed during the string parse. When a new phrase is inserted or used to modify the frequency of the mobility trie, we say an *event* happens and the time increases by 1. Note that this event corresponds to a sequence of symbols. We do not need to do an extra trie search to modify the frequency. Instead, it could be done at the same time with the update of the mobility trie. In this way, the modifications can be done efficiently.

The mobility trie is modified in the following way. At time t, the frequency of each node in the mobility trie is updated as:

- $$F_i(t) = \lambda * 1 + (1 - \lambda) * F_i(t-1)$$

  where node i is one item of the corresponding events;

- $$F_i(t) = \lambda * 0 + (1 - \lambda) * F_i(t-1)$$

  where node i is not one item of the corresponding events.

Here $F_i(t)$ is the frequency value stored in node i at time t. λ is a smoothing constant that determines the decay rate. The frequency value of a node that has not been observed from time $(t-k)$ to t will be decayed to $(1-\lambda)^k$. In this way, the frequency of each node measures the intensity of this node over the recent past.

In our later experiment, we set λ to 0.3, a commonly used value for the smoothing constant [3], that is, when a node is newly inserted into the mobility trie, its frequency value is set to 0.3.

Continuing the example illustrated in Fig. 4, we illustrate how to integrate EWMA into the construction of the mobility trie. When the first symbol a is parsed, the corresponding mobility trie is illustrated in Fig. 5.

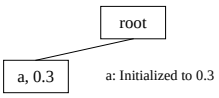

Figure 5: When a is parsed.

When ab is parsed, the corresponding mobility trie is illustrated in Fig. 6.

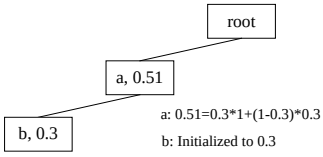

Figure 6: When $(a)(ab)$ is parsed.

When b is parsed, the corresponding mobility trie is illustrated in Fig. 7.

As we can see, the frequency value associated with each node is exponentially faded.

The EWMA based mobility trie construction is summarized in Fig. 8.

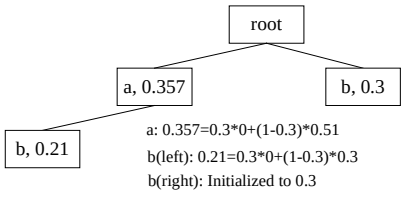

Figure 7: When $(a)(ab)(b)$ is parsed.

initialize mobility database := null

LOOP

wait for a sequence s

IF (the mobility trie of the mobile exists)
  IF (a path p corresponding to s is found)
    add s to the mobility trie
    using EWMA to modify the frequencies of nodes
  ELSE
    create new nodes, and initialize their frequencies to λ
ELSE
    1) create a mobility trie := single sequence s
    2) initialize the frequencies for every node in sequence s to λ

FOREVER

Figure 8: Integrating EWMA into Mobility Trie Construction.

4.4.2 The Distance Measure

EWMA based mobility trie maintains the stationary part of each user's recent activities. Based on this, we could accurately predict whether the future activities are normal or not.

Let $S = (X_1, X_2, \ldots, X_n)$ denote the observed activities of the user, where X_i denotes a cell number. We want to identify whether it is normal or not based on our constructed mobility trie. First, we use a high order Markov model to compute its blending transition probabilities.

For order $i \geq 1$, suppose its associated weight is w_i, we define its o-th order transition probabilities as:

$$P_o = \sum_{i=1}^{n-o} P(X_{i+o} | X_i, X_{i+1}, \ldots, X_{i+o-1})$$

When it is order-0 model, the probability is calculated as:

$$P_0 = \sum_{i=1}^{n} P(X_i)$$

To calculate the probability of the transition $(X_i, X_{i+1}, \ldots, X_{i+o-1}) \longrightarrow X_{i+o}$, we search from the root the path $(X_i, X_{i+1}, \ldots, X_{i+o-1})$. If the path could be found, the probability $P(X_{i+o} | X_i, X_{i+1}, \ldots, X_{i+o-1})$ is defined as:

$$P(X_{i+o}|X_i, X_{i+1}, \ldots, X_{i+o-1}) = \frac{F(X_{i+o})}{F(X_{x+o-1})}$$

If the path $(X_i, X_{i+1}, \ldots, X_{i+o-1})$ is not found, its probability is assigned 0.

Suppose the blending weight vector is $[w_0, w_1, \ldots, w_m]$, the probabilities of string S is defined as:

$$P = \sum_{i=0}^{m} w_i * P_i$$

Intuitively, P increases with the increase of S's length because more transitions will be considered when S is longer. Therefore, P is not a good metric. We propose to use the following metric as our *distance* measure:

$$Distances(S) = \frac{P}{Length(S)}$$

where $Length(S)$ is the length of string S.

Based on our definition, the distance measure could be normalized by the length of the string and provides good criteria to evaluate its normalcy. Intuitively, *Distance* indicates how good a mobile user follows its routines.

For the input string S, we calculate its $Distance(S)$. When a user follows one of its favorite itineraries, because this path is reflected in the mobility trie, many of its transitions at different orders will be found. Based on our definition, $Distance(S)$ will be a relatively large value. However, when the mobile is stolen, and the intruder takes an infrequent path, the distance of this string tends to be a very small value, because many transitions cannot be found in the mobility trie.

We introduce a threshold, P_{thr}, which is a design parameter. When $Distance(S) \geq P_{thr}$, string S is evaluated as normal, otherwise string S is identified as anomalous. Suitable mechanisms should be developed to establish the corresponding connection between the mobility level and the threshold. This is one of our important future work.

Because our mobility trie records the most frequently used path of a user, it is very sensitive to anomalous paths, even if they are very short strings. This enables our detection algorithm to detect the abnormal very *quickly* - an important quality for reducing potential damage by a malicious user. At the same time, our detection algorithm enjoys a very high detection rate. Also, when a frequently used path is taken, our detection algorithm can tolerate its slightly variation and lead to small false positive rate.

4.5 Theoretical Analysis of the Intrusion Detection Scheme

Since our intrusion detection scheme is derived from Ziv-Level data compression algorithm, we first analyze the optimality of word-based Ziv-Level algorithm and show that the character-based Ziv-Lempel algorithm is at least as good as the word-based scheme. Then, we will show that our intrusion detection scheme inherits the optimality of these data compression algorithms.

Given a sequence x^n of length n, word-based Ziv-Lempel data compressor parses it into different phrases, $x_1, x_2, \ldots, x_t$. Let $t(x^n)$ denote the maximal possible number of distinct phrases. Define $q(x^n) = t(x^n)\log(t(x^n))/n\log(\alpha)$. For an *information lossless* (IL) data compressor C accepting a sequence x^n of length n over alphabet A of α letters, let $|C(x^n)|$ denote the length of the output from C. The compression ratio $\rho_C(x^n)$ can be calculated as [24]: $\rho_C(x^n) = |C(x^n)|/n\log(\alpha)$. Let $\rho_\sigma(x^n)$ denote the best compression ratio attainable for x^n by any IL compressor. It is shown that [24]

$$\rho_\sigma(x^n) \geq q(x^n) - \delta(\sigma, n) \text{ with } \lim_{n\to\infty} \delta(\sigma, n) = 0.$$

The above result shows that word-based Ziv-Lemple algorithm achieves a compression ratio that is (asymptotically) equal to $q(x^n)$, which means that the algorithm is universal and asymptotically optimal.

The coding length obtained in the character-based Ziv-Lempel algorithm is shown in [2] to be as least as good as that obtained using the word-based approach. Therefore, the character-based Ziv-Lempel algorithm is also universal and asymptotically optimal.

Define the *false alarm rate* to be the total number of event false alarms that our scheme incurs divided by the total number of alarms. Moreover, we define the *expected false alarm rate* to be the best possible false alarm rate achievable by any intrusion detection algorithm that makes its prediction based only on the past history.

THEOREM 1. *If the source is a stationary m-th order Markov source, the expected value of the false alarm rate of the intrusion detection scheme derived from the Ziv-Lempel algorithm is within an additive factor of $O(1/\sqrt{n})$ from the expected false alarm rate of the source, where n is the length of the source sequence.*

For a proof of Theorem 1, please refer to [27]. The same is true for *detection rate*. This theorem shows that our intrusion detection algorithm inherits the asymptotic optimality of the Ziv-Lempel algorithm after it converges.

4.6 Implementation issues

In practice, an important issue is how to store the mobility information in a trie. A trie is actually a multiway tree with a path from the root to a unique node for each string represented in the tree. The fastest approach for processing is to create an array of pointers for each node in the trie with a pointer for each character of the input alphabet. Although this approach is easy for processing, it wastes memory space. Another approach is to use a linked list at each node, with one item for each possible branch. This method uses memory economically, but the requirements for processing are intensive. A trie can also be implemented as a single harsh table with an entry for each node. For further details, the reader can consult books on algorithms and data structures.

5. SIMULATION STUDY

5.1 Data Sets

A generalized graph model is used in our simulations to represent a cellular network of 40 cells, each having six neighbors on average. The average distance between two base stations is 1 mile. To avoid the edge effect of the finite network size, wrap-around is applied to the edge cells. Since most mobile users have favorite routes in reality, we assume that each mobile user has five possible paths in the network.

A mobile user will take these five paths with probabilities of $0.6, 0.2, 0.1, 0.05, 0.05$, respectively. The paths are generated as follows. (1) Select two cells in the graph randomly as original and destination cells. (2) Whenever the mobile user leaves the current cell, it moves to a neighboring cell that is closest to the destination. Call durations are the same for all calls and exponentially distributed with mean value of 3 minutes. With a fixed call duration, the higher the speed, the longer the cell list. Since mobile users travel with different speeds, we consider five cases of user mobility. The speeds of mobiles are 20, 30, 40, 50, 60 miles/hour in the five cases, respectively.

In our simulation, we manually set m to 2. That is, we apply a blended Markov model with order-0, order-1, and order-2, respectively, to the data sets. For order-0, we set w_0 to 0.7. For order-1 and order2, we set w_1 and w_2 to 0.2 and 0.1, respectively.

Note that the mobility data set we generated are generic enough for most users. However, it may not be suitable for users with totally random movement behavior such as taxi drivers.

5.2 Performance Metric

We use the following two metrics to evaluate the performance of our proposed detection algorithm:

- False Alarm Rate: It is measured over normal itineraries. Suppose m normal itineraries are measured, and n of them are identified as abnormal, *false alarm rate* is defined as n/m.
- Detection Rate: It is measured over abnormal itineraries. Suppose m abnormal itineraries are measured, and n of them are detected, *detection rate* is defined as n/m.

5.3 Simulation Results

In this section, we present and analyze the simulation results at different mobility levels.

5.3.1 False Alarm Rate

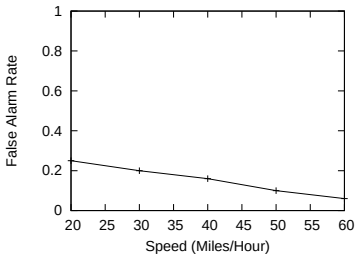

Figure 9: False Alarm Rate at Different Mobility Levels.

Simulation results of the false alarm rate are illustrated in Fig. 9. As we can see, generally, the false alarm rate is very low. Also, the false alarm rate decreases with the increase of the mobility, This is what we have expected. With the increase of mobility, the user tends to traverse more cells in a call. Therefore, for a normal user with regular movement pattern, his itinerary will demonstrate more resemblance to his regular activities that is recorded in the mobility trie. Therefore, the probability normalized by the itinerary length is relatively stable. In the way, the false positives are reduced.

However, when the mobility is very low, each user will traverse only one or at most two cells in a call. This makes it very difficult to identify whether it is normal or not. Some users may occasionally demonstrate abnormal behaviors. This could also lead to false positives. When the itinerary is relatively long, it is still possible to generate false positives. This is because the valid paths taken in the training data may happen with very low probability. Therefore, even if the similar path is taken again, it is still possible to be identified as an abnormal path.

5.3.2 Detection Rate

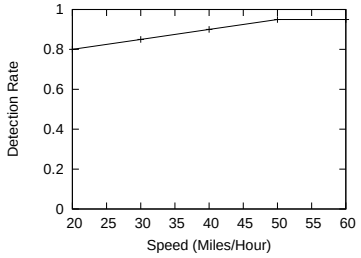

Figure 10: Detection Rate at Different Mobility Levels.

Simulation results of the detection rate are illustrated in Fig. 10. As we can see, generally, the detection rate is very high. Also, the detection rate increases with the increase of the mobility. The reason is similar. With the increase of the mobility level, each user tends to have more cells traveled. Therefore, for a masquerader, his itinerary tends to deviate significantly from the normal profile. In this way, the detection rate is improved with the increase of mobility.

When mobility is very low, because of similar reasons, it is still difficult to detect these abnormal itineraries. From the simulation, we cannot achieve 100% detection rate in any case, since when the itinerary is relatively long, it is possible that part of the intruder's path overlap with some normal path. Therefore, it is still possible to miss the detection of these kinds of itineraries.

6. RELATED WORK

There are two important intrusion detection techniques: *misuse detection* and *anomaly detection*. [5] presents a good taxonomy of existing technologies. The research of intrusion detection began with a report by Anderson [6] followed by Denning's seminal paper [7]. Since then, many research efforts have been devoted to different detection techniques. For example, Expert system [8] [9], pattern recognition [10], colored petri nets [11], and state transition analysis [12] [13] have been used to construct misuse based detection techniques. Different statistical approach [8] and Neural Networks [14] have been used to construct anomaly based detection techniques. All existing approaches take into consideration domain specific knowledge to build suitable detection systems.

Relatively few research efforts have been devoted to intrusion detection research of wireless networks. In [15], Samfat *et al.* proposed IDAMN (Intrusion Detection Architecture

for Mobile Networks) that includes two algorithms to model the behavior of users in terms of both telephony activity and migration patterns. Marti *et al.* [16] proposed to install extra facilities, *watchdog* and *pathrater*, to identify a particular routing misbehavior in Mobile Ad Hoc Networks (MANETs). Zhang *et al.* [17] proposed a general intrusion detection and response mechanism for MANETs, in which each IDS agent participates the intrusion detection and response tasks independently. Continuing their research, Yian *et al.* [18] used cross feature analysis to analyze the routing activities and investigate how to improve the anomaly detection approach, and provided more details on attack types and sources in [19]. Sun *et al.* [20] presented a Markov Chain based anomaly approach for MANETs and proposed a ZBIDS framework [21] to enable alert aggregation and correlation.

7. CONCLUSIONS AND FUTURE WORK

Based on optimal data compression, this paper presents a novel approach to construct the mobility profile of users in wireless cellular networks. Each user's itinerary is modeled as a m-th Markov source and EWMA is applied to make the normal profile up-to-date. An intrusion detection algorithm is then developed to detect potential *internal* attackers - masqueraders. Simulation results demonstrate that our approach can achieve desirable performance in terms of false alarm rate and detection rate for users having normal movement patterns. Our detection method can be used to build an appealing service to end users as well as a useful tool to service provider.

Our system right now only considers the mobility patterns, which may not be accurate for some particular type of users such as taxi drivers. More features such as call history and activities will be accommodated into the system to make it suitable to all users.

8. REFERENCES

[1] A. Bhattacharya, and S.K. Das, "LeZi-update: an information-theoretic approach to track mobile users in PCS networks", *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking (Mobicom'99)*, Seattle, WA, Aug., 1999.

[2] T.C. Bell, J.G. Cleary, and I.H. Witten, *Text Compression*, Prentice-Hall Advanced Reference Series, Prentice-Hall, Englewood Cleffs, NJ 1990.

[3] SPSS Inc., *AnswerTree 2.0: Users Guide*. Chicago, IL: SPSS, Inc.

[4] J.G. Cleary and I.H. Witten, "Data compression using adaptive coding and partial string matching," IEEE Transactions on Communications, Vol. 32, No. 4, pp. 396-402, Apr. 1983.

[5] H. Debar, M. Dacier, and A. Wespi, "A Revised Taxonomy for Intrusion-Detection Systems," Annales des Tlcommunications, vol. 55, 2000, pp. 361 - 378.

[6] J. P. Anderson, "Computer Security Threat Monitoring and Surveillance," Technical Report, James P. Anderson Co., Fort Washington, PA, April, 1980.

[7] D. E. Denning, "An Intrusion-Detection Model," *IEEE Transactions on Software Engineering,* vol. 13, no. 7, pp. 222-232, Feb. 1987.

[8] P. Porras and A. Valdes, "Live Traffic Analysis of TCP/IP Gateways," *Proceedings of the 1998 ISOC Symposium on Network and Distributed System Security (NDSS'98),* San Diego, CA, March 1998.

[9] U. Lindqvist, and P.A. Porras, "Detecting Computer and Network Misuse through the Production-Based Expert System Toolset (P-BEST)," *Proceedings of the 1999 IEEE Symposium on Security and Privacy,* Oakland, CA, pp. 146-161, May 9-12, 1999.

[10] Internet Security Systems, "RealSecure Network Protection," Nov. 2003, Available at *http://www.iss.net/products_services/enterprise_protection/rsnetwork*.

[11] S. Kumar and E. Spafford, "A Pattern Matching Model for Misuse Intrusion Detection," *Proceedings of the 17th National Computer Security Conference,* pp. 11-21, Oct. 1994.

[12] P.A. Porras and R. Kemmerer, "Penetration State Transition Analysis C a Rule-Based Intrusion Detection Approach," *Proceedings of the 8th Annual Computer Security Application Conference,* pp. 220-229, Nov. 1992.

[13] K. Ilgun, "Ustat: A Real-time Intrusion Detection System for Unix," *Proceedings of IEEE Symposium on Research in Security and Privacy,* Oakland, CA, pp. 16-28, May, 1993.

[14] H. Debar, M. Becker and D. Siboni, "A Neural Network Component for an Intrusion Detection System," *Proceedings of 1992 IEEE Symposium on Research in Security and Privacy,* Oakland, CA, pp. 240-250, May, 1992.

[15] D. Samfat, and R. Molva, "IDAMN: An Intrusion Detection Architecture for Mobile Networks," *IEEE Journal on Selected Areas in Communications,* vol. 15, no. 7 , pp. 1373-1380, Sept. 1997.

[16] S. Marti, T. Giuli, K. Lai, and M. Baker, "Mitigating Routing Misbehavior in Mobile Ad Hoc Networks," *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (ACM MobiCom'00)*, Boston, MA, pp. 255-265, Aug. 2000.

[17] Y. Zhang and W. Lee, "Intrusion Detection in Wireless Ad Hoc Networks," *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (ACM MobiCom'00),* Boston, MA, pp. 275-283, Aug. 2000.

[18] Y. Huang, W. Fan, W. Lee, and P. S. Yu, "Cross-Feature Analysis for Detecting Ad-hoc Routing Anomalies," *Proceedings of the 23rd International Conference on Distributed Computing Systems*, Providence, RI, pp. 478-487, May 2003.

[19] Y. Huang, and W. Lee, "A Cooperative Intrusion Detection System for Ad Hoc Networks," *Proceedings of the ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN'03),* Fairfax VA, October 2003.

[20] B. Sun, K. Wu, and U. Pooch, "Routing Anomaly Detection in Mobile Ad-Hoc Networks," IEEE International Conference on Computer Communications and Networks (ICCCN'03), Dallas, TX, 2003, pp. 25-31.

[21] B. Sun, K. Wu, and U. Pooch, "Alert Aggregation in Mobile Ad Hoc Networks," *Proceedings of the 2003 ACM Workshop on Wireless Security (WiSe'03) in*

Conjunction with Mobicom'03, San Diego, CA, pp. 69-78, Sept. 2003.

[22] D. Sheinwald, "On the Ziv-Lempel proof and related topics," *Proceedings of IEEE*, vol. 82, pp. 866-871, June 1994.

[23] F. Yu and V.C.M. Leung, "Mobility-Based Predictive Call Admission Control and Bandwidth Reservation in Wireless Cellular Networks," *Elsevier Computer Networks,* vol. 38, no. 5, pp. 577-589, Apr. 2002.

[24] J. Ziv, and A. Lempel, "Compression of individual sequences via variable-rate coding," *IEEE Transaction of Information Theory*, Sept., pp. 530-536, 1978.

[25] C. Warrender, S. Forrest, and B. Pearlmutter, "Detecting Intrusions Using System Calls: Alternative Data Models," *Proceedings of 1999 IEEE Symposium on Research in Security and Privacy,* Oakland, CA, pp. 133-145, May 1999.

[26] R.A. Johnson and D.W. Wichern, *Applied Multivariate Statistical Analysis,* Upper Saddle River, NJ: Prentice Hall, 1998.

[27] J.S. Vitter and P. Krishnan, "Optimal Prefetching Via Data Compression," *J. of ACM*, vol. 43, no. 5, pp. 771-793, Sept. 1996.