Artificial Impostor Profiling for Keystroke Analysis on a Mobile Handset

J. Lecomte[1], N. Clarke[1] and S. Furnell[1,2]

[1]Network Research Group, University of Plymouth, Plymouth, United Kingdom
[2]School of Computer and Information Science, Edith Cowan University, Perth, Australia
e-mail: info@network-research-group.org

Abstract

Keystroke Analysis is a biometric approach that utilises the typing characteristics of a user to perform identity authentication, and has two key advantages in a mobile context – the necessary authentication hardware (i.e. the keypad is already present) and the technique can operate transparently. Although studies have proved the feasibility of such an approach on a mobile handset, a failing exists in the practical deployment of the system. Classification is performed by neural networks that are trained using both the authorised users samples and impostors as a means of comparison. However, in the real world, the availability and suitability of impostor samples will be limited. This paper proposes a means of artificially creating impostor data directly based upon samples from the authorised user in order to provide optimally configured classification engines. These artificial impostor approaches have not only solved the availability issue but have improved the system performance (in comparison to the traditional approach) by up to 25%.

Keywords

Biometrics, Keystroke Analysis, Keystroke Dynamics, Impostor Profiling, User Authentication

1. Introduction

The mobile telecommunications industry has experienced formidable growth in recent years with in excess of 1.3 billion subscribers worldwide (Cellular Online, 2003). In order to capitalise on possible revenue, network operators have moved from a voice centric telephony device to a multimedia communications device capable of providing a wide variety of data based services. These services permit the subscriber to access a number of potentially sensitive locations, including, corporate networks, personal bank accounts and share dealing services (Giussani, 2001). In parallel with this increase in ownership there has been a corresponding increase in mobile handset theft, with over 700,000 stolen in the UK during 2001 (BBC News, 2002). With the increase in data sensitivity and handset misuse, the need to ensure subscriber identity becomes paramount.

The use of biometrics, or specifically unique human characteristics, has existed for hundreds of years in one form or another, whether it is a physical description of a person or perhaps more recently a photograph. The application of biometrics to telephony devices is an intriguing proposal given that many of the new third generation handsets will incorporate the hardware required to capture the biometric sample. For instance, the video conferencing camera could be utilised for facial recognition and the microphone for voice verification.

Unfortunately however, the application of biometrics in a practical sense introduces a number of additional problems to be solved. For instance, there are computational overheads when using biometrics, which current handsets may have problems processing.

One biometric of particular interest within a handset context, due to its non-intrusiveness is keystroke analysis. This technique utilises the typing characteristics of subscribers to differentiate between them and can therefore (in principle) authenticate users during their normal handset interactions, such as when they are dialling telephone numbers and entering PINs. Although feasibility studies have demonstrated promise in utilising such a technique (Clarke et al, 2003; Clarke et al, 2004) an issue arises in the practical implementation and evaluation of the method. The current classification process utilises neural networks, where data from the authorised subscriber is used alongside impostor data to teach the network the difference in input characteristics. So the network is taught which input data belongs to the authorised user and which belongs to impostors. In practicality however, the suitability and availability of the impostor data limits the performance and implementation of the technique, for the following reasons:

- Availability – a bank of impostor data will always be required for each user to teach the neural network
- Suitability – the bank of impostor data may or may not be similar to the authorised users' dataset. Impostor data that is able to surround an authorised users dataset would be the ideal.
- Performance – the network is evaluated using the same impostor users with which it was explicitly trained to reject (although the particular samples have not been used in the training), giving rise to possibly skewed performance rates.

This paper presents a number of algorithms design to artificially create impostor data, based specifically on the authorised user. Creating impostors that closely imitate (but does not duplicate) the authorised user's input distribution, should result in removing the availability issue and improve the suitability of impostor data and increase the performance of the overall classifier.

2. Keystroke Analysis Investigation

The experimental procedure to evaluate the impostor algorithms sought to duplicate the investigations described in Clarke et al (2002). This permits a comparison between the original results and those generated using the impostor algorithms. To this end, the impostor algorithms were tested against three types of input data:

1. Entry of a fixed four-digit number, analogous to the PINs used on many current systems.
2. Entry of a series of telephone numbers. The classification of dynamic inputs is expected to increase intra-user variance, and thereby make it harder for the network to classify.
3. Entry of a fixed telephone number in order to facilitate a comparison against the results from the second experiment.

A total of thirty two test subjects provided the input data required for all three investigations. Table 1 illustrates the dataset sample sizes after outliers have been removed. For the traditional tests, each user is taken in turn as the authorised user with all the remaining users acting as impostors and trained using the training dataset. The evaluation of the neural networks is then performed by a validation dataset – containing data not used in the training procedure.

	Total # of Samples	**# of Training Samples**	**# of Validation Samples**
4-Digit PIN	25	16	9
Varying Telephone	38	26	12
Fixed Telephone	21	14	7

Table 1 Number of Samples in Investigations

The artificial impostor tests will only utilise the authorised users training samples during the training stage and to create the impostor data. The networks will again be validated using the identical validation dataset as before in order for a fair comparison of results to be made.

A specially written application was used to collect the sample data. However, it was considered that the standard numerical keypad on a PC keyboard would not be an appropriate means of data entry, as it differs from a mobile handset in terms of both feel and layout, and users would be likely to exhibit a markedly different style when entering the data. As such, the data capture was performed using a modified mobile handset, interfaced to a PC through the keyboard connection.

Due to the limitations of data collection, the input data required for training and testing of the authentication system had to be collected in a single session. Ideally, the data would be collected over a period of time, in order to capture a truer representation of the users typing pattern. For example, by asking the user to type in 50 telephone numbers all at once, could result in an exaggerated learning curve.

3. Artificial Impostor Algorithms

Impostor algorithms were created using traditional statistical tools used normally to study natural phenomena and pattern recognition in general (Jain et al., 1999). Each approach attempts replicate the authorised user's profile, but by adding and subtracting a noise content in order to bound all authorised samples with unauthorised samples. Three approaches are described and evaluated in this paper:

- bootstrap sampling of vector components with noise;
- weighted intervals with noise;
- manipulation of normal distribution parameters.

3.1 Bootstrap Sampling of Vector Components

The concept behind bootstrapping involves choosing at random samples with replacement from a dataset and analysing each sample in an identical manner. In this particular approach,

rather than taking complete samples each component of the sample or vector is taken independently. Given the 4-digit PIN input where there are 16 samples, each of the 4 latencies will be taken in turn, bootstrapping from the 16 available samples from that placement, creating a new four latency sample using the authorised user's data. By subsequently adding noise to each of the four latencies this will shift the sample away from the authorised users' distribution. The reason sample components are taken from their same respective positions rather than from the complete dataset, are in order to conserve inherent typing characteristics within the sample, as illustrated in Figure 1. For instance, the fifth sample component in the telephone input investigations tends to be typically larger than the others as it represents the point between the end of the area code and start of the individual number (in a UK format telephone number) – a normal stage to pause.

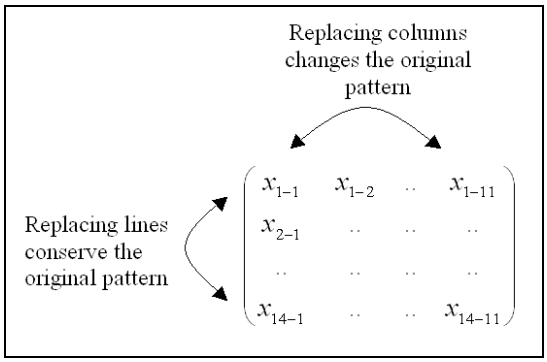

Figure 1 Typing Pattern Conservation

An important consideration in utilising this approach is the size of the noise added to the vector components. Care must be taken not to generate the same distribution or even a very close distribution to the authorised user, so that the intra-user space (space between an authorised users own input samples) is not affected. The noise is used to create some sufficient distance from the authorised user. The noise is randomly chosen over a bounded interval, for each sample component. The algorithm can be optimised by monitoring the evaluation stage and increasing or decreasing the noise, thereby moving the distance of the impostor samples from the authorised user's distribution. The equation for the algorithm is illustrated in Equation 1.

$$\text{Impostor vector} = (\text{Random}(x_{1-1} \quad \ldots \quad x_{14-1}) + n_1, \text{Random}(x_{1-2} \quad \ldots \quad x_{14-2}) + n_2 \\ ,\ldots, \text{Random}(x_{1-11} \quad \ldots \quad x_{14-11}) + n_3)$$

Equation 1 Bootstrapping Sampling of Vector Components

3.2 Weighted Intervals with Noise

This approach takes a more pragmatic approach than the first by assigning probabilities to vector components in defined intervals. As before, this technique splits the sample into its constituent latencies in order to conserve any typing characteristic, as illustrated in figure 2, however, instead of subsequently performing a bootstrapping method, this approach takes all samples of that vector component and sorts them into ascending order.

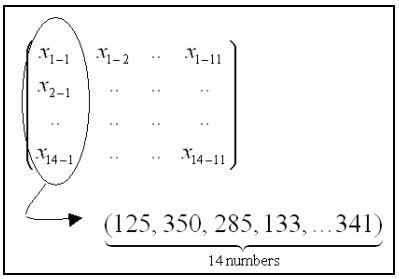

Figure 2 Extracting a samples' constituent parts

A noise is then added to each latency to surround the value, thereby ensuring values inside these bounds are authorised and outside are classified as impostors. Probabilities are then calculated based upon what latencies are observed with predefined boundaries. These probabilities are defined as the weight of that particular interval. Samples are then generated by picking at random, ensuring however the weights are maintained thereby ensuring the impostor samples closely mimic the authorised users data overall, but with the addition of noise. Equation 2 illustrates this approach.

$$\text{Impostor vector} = (\text{Random}(\text{limit\_min}; x_{1-1} - n), \text{Random}(x_{1-1} + n \;;\; x_{2-1} - n), \ldots, \text{Random}(x_{14-1} + n; \text{limit\_max}))$$

Equation 2 Weighted Intervals

This concept is thus very similar to the probability density function, where each interval is assigned a probability of having one of its value represented in the final vector.

3.3 Manipulation of Normal Distribution Parameters

The third algorithm moves away from the raw data to utilising the parameters that describe the authorised users input data. The input data generated from users can be approximated to a normal distribution where the mean and standard deviation parameters can be used to describe the distribution. A noise is again added to the vector components to ensure a desirable distance away from the user distribution. The equation of this algorithm is illustrated below, where the noise is a random coefficient x (0.1<x>2) to preserve a user space, but stay close enough to the distribution.

$$\text{Impostor\_matrix} = (impostor\_vector1(\bar{x}, \sigma, 11, 1) \times noise1 \quad \ldots \quad impostor\_vector14(\bar{x}, \sigma, 11, 1) \times noise14)$$

Equation 3 Normal Distribution Parameter Manipulation

Where $impostor\_vector1(\bar{x}, \sigma, 11, 1)$ is a pseudo random vector of dimension (11×1) based on the standard deviation σ and mean $\bar{x}$ of the authorised user.

In all three approaches an important consideration concerning the amount of impostor data generated must be taken into account. With too much impostor data and the network will respond by rejecting all input samples, but with too little, too many impostors will be able to gain access. With the noise, this gives rise to a second variable that can be altered in order to optimise the performance of the algorithms.

4. Results and Discussion

A comparison of the results achieved by all three approaches, as illustrated in table 2, indicates that in general the *manipulation of normal distribution parameters* technique proved most successful achieving the lowest Equal Error Rate (EER) for both telephone input scenarios, and only 1% off the lowest EER for the 4-digit PIN. A reason for this can be conjectured to be due to the more general classification boundaries that are produced using just two measures of the authorised user's distribution, instead of the large manipulation of actual raw data which the remaining techniques utilise. The manipulation of normal distribution parameters also represents the simpler approach, in terms of both time and computation.

Additionally, a descriptive statistical analysis of the input data unsurprisingly reveals that both telephone input scenarios have a larger intra-user variance (i.e. the spread of input samples within a user's collection of data) than the 4-digit PIN, indicating the classification boundaries created with the telephone inputs scenarios will be more general as the samples vary so much. This also helps to argue the reason as to why the more general normal distribution parameter technique proved more useful – as the other techniques followed user's input data too closely not allowing for the more general pattern.

	Bootstrap Sampling of Vector Components		Weighted Intervals		Normal Distribution	
	EER (%)	Parameters (Noise/# Impostors)	EER (%)	Parameters (Noise/# Impostors)	EER (%)	Parameters (Noise/# Impostors)
4-Digit PIN	18	150-250/ 50	21	0.5-0.6/ 30	19	0.5-0.6/ 80
Varying Telephone #	41	200-250/ 50	44	0.5-0.6/ 30	35	0.4-0.5/ 80
Fixed Telephone #	25	200-300/ 80	24	0.2-0.3/ 30	21	0.5-0.6/ 80

Table 2 Impostor Algorithm Results

The results in table 2 illustrate the best achievable results after both noise and amount of impostor data parameters had been varied. For the bootstrap sampling of vector components, the noise parameters are in milliseconds, however the other two techniques measure noise in terms of standard deviation about the authorised user's mean. The number of impostors is a measure of the amount of artificial data that was utilised, with 1 impostor equating to the number of samples provided by the authorised user (e.g. for the 4-digit PIN, 1 impostor = 16 samples).

A problem with the traditional approach to keystroke analysis is that impostor's used in training the classification engine are also the users that are subsequently used to evaluate the performance. Although the data has never been used by the engine before, the neural network has been specifically trained to reject that particular user's input data. As these artificial impostor algorithms do not use real impostor data during the training procedure, the results given here permit a more accurate representation of the achievable classification that could be expected.

However, the principle objective of this research was to artificially created impostor data that performed as well as, if not better than, utilising real impostor input samples. Table 3 illustrates a comparison of the best artificial impostor results against the traditional technique of using actual impostor data.

	Traditional Approach			Artificial Impostor Algorithms			
	FAR	*FRR*	*EER*	FAR	FRR	EER	Technique
4-Digit PIN	*9*	*39*	*24*	27	9	18	Bootstrap sampling of vector components[*2]
Varying Telephone #	*9*	*71*	*40*	28	41	35	Manipulation of Normal Distribution Parameters[*1]
Fixed Telephone #	*10*	*38*	*24*	23	18	21	Manipulation of Normal Distribution Parameters[*1]

Table 3 Artificial Impostor Algorithm Results (with a comparison versus the traditional approach)

[*1] Noise parameters set at +/- 0.4-0.5 of standard deviation about mean with the equivalent of 80 impostors worth of input data.

[*2] Noise parameters set at +/- 150-250 mS with the equivalent of 50 impostors worth of input data.

As the results show, the artificially created impostor data has outperformed the traditional approach in all three input scenarios, with a 25% improvement in the 4-digit PIN and 12.5% improvement in both telephone input scenarios.

5. Conclusions & Future Work

The use of artificially created impostor data over real impostor data has a number of advantages; a true representation of the performance, optimised neural networks for all compatible authorised users, no requirement for a database of users to be used as impostor data and a self-contained authentication technique with small storage footprint – as only the authorised user's data need be kept. Moreover the use of artificially created impostor data has improved the performance of the technique over the traditional approach, indicating stronger classification boundaries have been created using impostor data generated directly based on the authorised user's input samples.

However, with a view of improving the performance still further, a number of areas have been identified for further research. The first minor improvement would be to dynamically adapt the noise level on an individual user basis thereby optimising the performance, as the current approach sets the noise level of all users to the same level, which might on average be the best level but might not be the case for individual users.

The basic assumption throughout this paper, and used explicitly in the manipulation of normal distribution parameters algorithm, has been the approximation of user's input samples to a normal distribution. Although this stands true, the approximation can be quite general in a number of users, so an argument exists for implementing another more complex distribution to model the input data. Figure 3 below illustrates theoretically a user's distribution with the dotted line and the attempt to model the distribution more accurately through the use of multiple normal distributions.

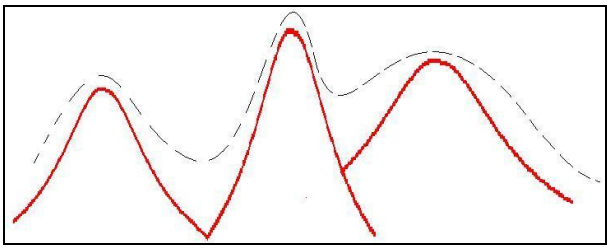

Figure 3 Model of a Complex Distribution using a mixture of Normal Distributions

Finally, it would be interesting to study more advanced algorithms to create impostor data. For instance, some biometrics utilise Hidden Markov Models to create impostor files (Rabinier, 1989). They are used to model events that have a probability to occur depending on a previous event. This technique could be used to more intelligently create impostor samples.

Given previous research projects have identified the usefulness and promise of keystroke analysis, this research has successfully identified a means of solving the issue of creating a classification engine using only data supplied by the authorised user.

6. References

BBC. 2002. "Huge surge in mobile phone thefts", BBC News Report, http://news.bbc.co.uk/1/hi/uk/1748258.stm 8th January 2002.

Cellular Online, Stats Snapshot 8/2003, http://www.cellular.co.za, Sep 2003. http://www.cellular.co.za/stats/stats-main.htm.

Clarke, N.L., Furnell, S.M., Lines, B., and Reynolds, P.L., 2003. "Using Keystroke Analysis as a mechanism for Subscriber Authentication on Mobile Handsets", *Proceedings of the IFIP SEC 2003 Conference*, Athens, Greece, May, pp97-108.

Clarke, N.L., Furnell, S.M., Lines, B., and Reynolds, P.L., 2004. "Application of Keystroke Analysis to Mobile Text Messaging", *Proceedings of the IsOneWorld 2004 Conference*, April 2004.

Giussani, B., 2001. *Roam; Making Sense of the Wireless Internet*. Random House Business Books.

Jain, A. K., Robert P.W. Duin, Jianchang Mao, 1999. "Statistical Pattern Recognition: A Review". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp4-37.

Rabinier, L., 1989. "A Tutorial on Hidden Markov Models & Selected Applications in Speech Recognition". *Proceedings of the IEEE*, vol. 77, no. 2, pp257-285.