Summary:
This report summarizes the outcomes of my participation in the “Keystroke Dynamics Authentication” summer studentship project. In this report, a brief background of the project is initially outlined. Then the report will focus on the main things that I have learned and the contributions I have made during the studentship: keystroke related knowledge, website for collecting experiment data and a program that processes the collected data. It is concluded that I have obtained good knowledge of the basic keystroke authentication mechanisms, good practise of the related knowledge I have gained through previous studies and a brief idea of what a research career may involve.

Table of contents
Summary
1.0 Introduction
2.0 Main keystroke knowledge I have gained
   2.1 Basic algorithms
      2.1.1 Degree of Disorder (DoD)
      2.1.2 n-graph
      2.2 Related works and their pros and cons
      2.3 Brief structure of our algorithm
3.0 The website to collect experiment data
4.0 The program to process data
5.0 Conclusion
6.0 Reference list

1. Introduction
The purpose of this report is to summarise the outcomes of my participation in the “Keystroke Dynamics Authentication” summer studentship project.

Working with another two PhD students and supervised by Jiankun Hu, I have gained a good knowledge of the keystroke authentication area and commonly used authentication algorithms. This studentship also gives me a good practise of the web programming and java programing skills I have obtained from previous studies. I have also got the experience of working within a team and how a research career might look like.

2. Main Keystroke knowledge that I have gained
   2.1.1 Degree of Disorder (DoD)
Given two vectors \( V \) and \( V' \) (each with \( N \) elements), as:

Degree of Disorder (DoD) between \( V \) and \( V' \) can be calculated as the summation of the distances, denoted as ‘\( d_{\text{idx}} \)’, between the position of each element in \( V \) and the position of the same element in \( V' \).

\[
\begin{array}{|c|c|c|c|c|c|}
\hline
V & A & E & I & O & U \\
\hline
V' & O & E & A & U & I \\
\hline
\end{array}
\]

In the above example, ‘\( d_{\text{idx}} \)’ of each corresponding element is computed as:

\[
\begin{align*}
d_{\text{idx}}(A) &= |\text{Idx}(AV) - \text{Idx}(AV')| = |1 - 3| = 2 \\
d_{\text{idx}}(E) &= |\text{Idx}(EV) - \text{Idx}(EV')| = |2 - 2| = 0 \\
d_{\text{idx}}(I) &= |\text{Idx}(IV) - \text{Idx}(IV')| = |3 - 5| = 2 \\
d_{\text{idx}}(O) &= |\text{Idx}(OV) - \text{Idx}(OV')| = |4 - 1| = 3 \\
d_{\text{idx}}(U) &= |\text{Idx}(UV) - \text{Idx}(UV')| = |5 - 4| = 1
\end{align*}
\]

where \( d_{\text{idx}}(x) \) can be calculated as the absolute difference between the index numbers of an element \( x \) in \( V \) and \( V' \). DoD between \( V \) and \( V' \) is computed as:

\[
\text{DoD}(V; V') = \sum d_{\text{idx}}(x)
\]

\[
= d_{\text{idx}}(A) + d_{\text{idx}}(E) + d_{\text{idx}}(I) + d_{\text{idx}}(O) + d_{\text{idx}}(U)
\]

\[
= 2 + 0 + 2 + 3 + 1 = 8
\]

Apparently, DoD increases in direct proportion to the vector size \( N \) and thus the normalisation is needed. It is convenient to normalise DoD via dividing it by the value of the maximum possible DoD value for a given \( N \). The maximum DoD is:

\[
\begin{align*}
&\text{for } N \text{ even: } N^2/2 \\
&\text{for } N \text{ odd: } (N^2 - 1)/2
\end{align*}
\]

After normalisation, the value of DoD fall between 0 (when \( V \) and \( V' \) are exactly the same) and 1 (when \( V' \) is in the reverse order of \( V \)). In the above example, the normalised DoD between \( V \) and \( V' \) is: \( 8/((5^2 - 1)/2 = 8/12 = 0.667. \)

### 2.1.2 n-graph

The duration of a n-graph refers to the elapsed time between the 1st key pressed and the \( N \)th key pressed. For instance, two keys typed one after the other are called a digraph.

### 2.2 Related works and their pros and cons

There are some related research in this area, each research has its pros and cons, summarized as following:

Gunetti et al., classical n-graph based keystroke verification method (GP method)

- pro: reducing the FAR (False Acceptance Rate)
- con: high FRR (False Rejection Rate)
- con: scalability issue and computational efficiency (a query keystroke sample should be compared with every training sample of every subject in the system)

Monrose et al.
pro: experiment was limited to a single password string for all users
con: typing errors did occur during experiment and some forms of error correction were required

Bergadano et al., a trigraph based algorithm
pro: Degree of Disorder (DoD), reduce the effect of variations in the absolute timing data on the authentication mechanism

2.3 Brief structure of our algorithm
Our team built an algorithm called n-graph disorder vector (nGdv) which is correlation-oriented and information loss-less. It has a better performance than all the existing ways of doing user authentication using keystroke.

3.0 The website to collect experiment data
I have built a website for collecting experiment data. The link of this site is: http://yallara.cs.rmit.edu.au/~s3231499/Survey/home.html.
This site consists 3 separate typing tests. Each person taking the test will be given an ID. Only the valid IDs can start the test.

For each letter this person types, the site will record the key code of this letter, the time of the key being pressed down, the time of the key being released and an indicator which indicates whether this person has made a mistake or re-typed the word that s/he is currently typing. When this person mistypes a letter, the wrong letter will still appear on the screen, but the site will stop the user from continuing typing unless this person presses the backspace key to delete the whole word and re-types it, in this case, each letter in this word will be marked as an error (because this person has re-typed it). When this person correctly types a word, but s/he thinks this word is not typed naturally, s/he can press the backspace key to delete the whole word and re-type it, each letter in this word will be marked as error. Clicking the backspace key will cause the whole word being deleted and the data for this word being erased. However, this person can only delete the word that s/he is currently typing, all the words before this one will be locked and cannot be changed anymore. When this person finishes one line, that whole line will be locked and the next line will be activated automatically. All the functions by far is written in javascript. When this person finishes the first page and clicks "next", the next page will appear. The new page will store the data submitted from the previous page and store it into the corresponding files using php. Then this person continues the test on this page. The following pages work the same way. This person keeps typing until finishing the test.

4.0 The program to process data
I have also developed a small java program to process the data.

After all the data being collected in the way described in 3.0, it will be stored in corresponding txt files on the server. This files then will be copied down from the server and sent to the java program. This program will perform calculation on the data according
to the algorithm requirement, formats it in the required way and stores it into the final result files. This files will be processed further more to find out the desired result of this project.

5.0 Conclusion
By doing this project, I have developed a good understanding of the keystroke dynamic algorithms and a good knowledge of the cloud computing area. I have also gained some experience on working within a team to discuss things and schedule things. But most of all, it exercises my knowledge of html, javascript, php and java that I have obtained from the previous studies and makes me realise that a research career is very different from studying the normal course in which everything is so clear. While in a research career, there are a lot of uncertainties that we must find out by ourselves.

6.0 Reference List
1. Kai Xi, "Correlation Keystroke Verification Scheme for User Access Control in Cloud Computing Environment"
2. “User Authentication through Keystroke Dynamics”, FRANCESCO BERGADANO, DANIELE GUNETTI, and CLAUDIA PICARDI University of Torino