International Journal of Scientific Research in Computer Science and Engineering
Vol.5, Issue.6, pp.33-38, December (2017)

Research Paper

E-ISSN: 2320-7639

Online Signature Verification using Dynamic Properties

EdigaLingappa[1], Geetavani.B[2], JambulaHareesha[3]

[1*]CSE, Institute of Aeronautical Engineering, JNTUH, Hyderabad, India
[2] CSE, Institute of Aeronautical Engineering, JNTUH, Hyderabad, India
[3] CSE, Institute of Aeronautical Engineering, JNTUH, Hyderabad, India

[*]*Corresponding Author: edigalingappa@gmail.com, Tel.: +919502967317*

Available online at: www.isroset.org

Received: 02/Nov/2017, Revised: 13/Nov/2017, Accepted: 08//Dec/2017, Published: 31/Dec/2017

***Abstract*—** Authentication of persons has been known as a paramount part in society. It is the only means to have a record of the person performing a transaction on his own will. As technology evolved security started to be a main concern as authentication problems increase. There are many authentication methods such as Knowledge based, Biometrics and Ownership based. All these authentication schemes are best at their respective use cases. Knowledge based schemes forces to remember passwords, ownership based requires to carry a thing which performs authentication and finally biometrics methods which are unique to a person requires for the person to safeguard his biological data.The best way of authentication since old times is by using a signature. Verification of the signature is one of the biometric methods used in recognition systems. APIs for verification, storing data and processing coordinates. The user's signature input is given using a touch interface of an Android mobile device. The user has to first register his signature and the properties of this signature are used to verify the signature for later transactions. Dynamic properties such as pressure, button status and the co-ordinates are used to provide a secure authentication. This process is called as online signature verification.This signature based verification is used at banking where there is transaction of high value. For land registration, instead of signing the documents, we can use online signature which cannot be forged easily

***Keywords*—** Dynamic Time Warping (DTW) algorithm; Online Signature; Dynamic properties; Biometric methods; OnlineSignatureVerification

I. INTRODUCTION

Online signatures are captured by social hardware (e.g. smart pens or pressure sensitive tablets) which is capable of measuring dynamic properties of a signature in addition to the shape, which is only used in offline signatures. Dynamic information (e.g. pen pressure) makes the signature unique and more difficult to forge.

Signature is the seal of approval of a person and is said to be unique as the DNA of a person [1]. This fact lays the foundation to develop an authentication system using signature verification. Moreover, in a country like India signature is the primary means to authenticate transactions such as land registrations and bank transactions (cheques). These transactions are on heavy attack of forgery. For such important transactions, online signature verification provides a good level of security. There is also another problem of finding a sign to be legitimate or not, as the forensics lab take their time to give the report on the transaction. This method of finding whether a transaction to be legitimate or not is time consuming and is of heavy investment of money. The importance of online signature verification [1] further adds to the motivation of the authentication system implemented.

Original feature set extraction [2] needs some pre-processing before they can be analysed further. The vector formation [1] also is crucial as it defines an order of how the features are stored and processed. This gives a globalization of the parameters to use across the system there by eliminating any confusion while using the vector. The number of dynamic properties

[1] for a signature are 8, but the stable ones [2] to be used were only implemented. Some of them are not used as there is no support on the platform to retrieve such dynamic properties (azimuthal and altitude angles).

II. RELATED WORK

The verification [2] process uses only the Dynamic Time Warping (DTW) algorithm and furthermore it depends on using a variable VOTE to decide to authenticate a transaction. This part of finding is used to make our decision as layers of comparisons for each dynamic property. The comparison of simple and skilled forgeries [3] provided us an insight that almost the simple forgeries cannot break through the authentication scheme and some skilled forgeries were able to make it through. Note that the scheme was using only DTW algorithm for authentication. This was taken in to consideration and algorithm was implemented as different for different parameters. We have also found that in a real-time scenario, signing on a touch interface was something to the users and was giving results of invalid transactions on forgeries.

Password protection- In a password-protectedsystem, the user is assigned an ID and a password. By verifying these user specific credentials, authentication is done. These credentials are stored beforehand at a database and are cross verified to authenticate the transaction. The first time users will register their details which is a onetime process to perform authentication.

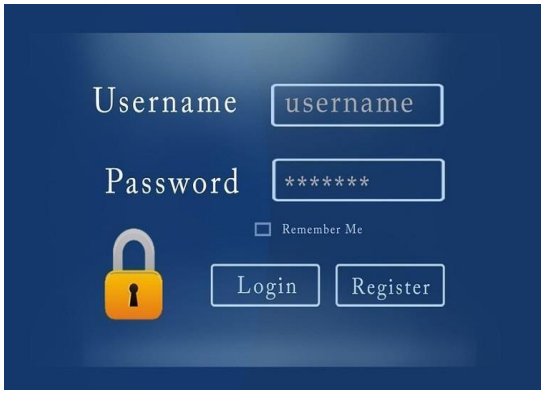

Figure 1.1 Password protection

The issue here is to remember the ID and password or only the password. If we forget the password, then there is another problem of account getting blocked because of invalid attempts. This leads to account recovering procedure and password resetting. We cannot store the password anywhere except our brain as it causes security issues, because someone might misuse or tamper with the account.

Biometric Authentication- As the name says, the authentication is based on biometrics of the user. Biometrics is the utilization of physiological traits like face, iris, and fingerprint. These features of a person are unique and cannot be forged. It is an alternative to password based authentication, as these biometrics cannot be stolen, forgotten or guessed. The procedure of registration is same like the password based authentication scheme, the only change is instead of using password, and we use a biological trait.

Figure 1.2 Biometric authentications

The disadvantage of this authentication system is that if there is any damage to the physiological traits like burnt finger, face scars, etc., then authentication is impossible. There is also a possibility of misusing the biometrics of the user when he/she is unconscious (may be sleeping), in such case the user will authenticate the transaction without his consent.

Signature Verification

Signature is a person‘s name written in a distinctive way as a form of identification in authorizing a document or concluding a letter. This method of authentication is used from the days where there were no technological advancements. Even today, Signature is used as the primary means of authentication for documents and for other transaction.

Signature verification which is done on paper is called as offline signature verification. The problem with signature verification is that it is vulnerable to forgery. It takes weeks of time at a forensics lab to determine whether a signature is legit or not. In this method there is no registration of signature at any database.

Figure 1.3 signature verification

III. METHODOLOGY

The approach for the solution is to have a touch enabled device and the user signs on it either with his finger or using a compatible stylus for authenticating a transaction. So there is question of how the authentication is verified as legitimate. For this, the user has to register his/her signature, this is just like signing up at a site with less details. While registering the user gives his name and other details if required and these are stored along with signature details in a database[2]. These details are used to compare when the user authenticates a transaction for verification.

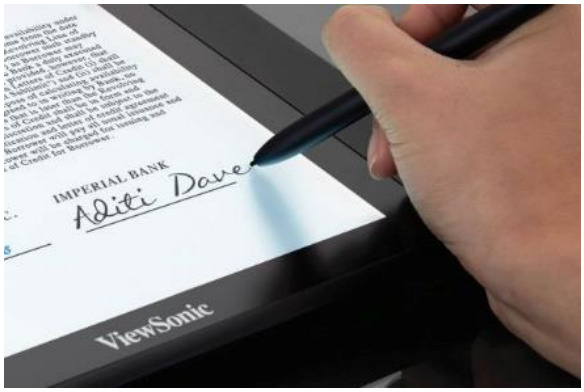

Figure 1.4 Online signature

When the user registers, properties such as the pressure, coordinates and pen-ups are recorded. These properties are called dynamic parameters and can be retrieved only using online signature process. These parameters are subjected to various algorithms and a value is generated as output for each parameter[1]. These values are grouped together, which we call it as a feature vector, this vector later helps in verification process for authenticating a transaction.
APIs for registering, storing and verifying the signatures are developed, so that these can be used in any supported application or to combine with any other authentication scheme to improve efficiency. More about the APIs and its implementation is discussed further.

Dynamic Properties
The advantage of using such parameters is that these cannot be copied or forged easily, as these are not visible to the naked eye. These parameters are like passwords which cannot be communicated to others; it is a pattern which only the user knows which he cannot explain it to others. There are other dynamic properties to consider, but we concentrate on the following properties in our implementation.

Coordinates
The interface is a 2 dimensional space where the user does his/her sign. As the interface is a touch interface, we can know when a person touches a point on the screen and corresponding (x, y) coordinates of the point are retrieved using the predefined functions in the java library packages. The coordinates obtained are processed further for generating a feature vector for the user[3].

Pressure
Whenever the user touches the screen the touch sensors gets activated and a pressure value is obtained. The level of security depends on the user, it is on the user‘s hands to have a complicated pressure pattern which only he can know and difficult to know even for the developer.

Pen-ups
It is the count of number of times the user breaks the sign in the whole signature, this property is important as it helps in determining whether a sign is forged or not. The count is obtained in the same way the coordinates are obtained.

Pseudo code

```
(Assumes the input is a one-based array)
functionDouglasPeucker (Point List[], epsilon)
// Find the point with the maximum distance dmax = 0
index = 0
end = length(Point List) fori = 2 to ( end - 1) {
d = perpendicular Distance (Point List[i], Line (Point List [1], Point List[end])) if ( d >dmax ) {
Index = i dmax = d
}
}
// If max distance is greater than epsilon, recursively simplify if (dmax> epsilon) {
// Recursive call
recResults1[]= DouglasPeucker (Point List[1...index], epsilon)recResults2[]=DouglasPeucker(Point List[index...end], epsilon)
// Build the result list
ResultList [] = {recResults1 [1...length (recResults1)-1], recResults2 [1...length (recResults2)]}
} else {
ResultList [] = {Pointlist[1], Point List[end]}
}
 // Return the result returnResultList [] end
```

DTW Algorithm
After RDP algorithm the coordinates are interpolated to a set of points before giving them as input to DTW. We use linear interpolation to improvise the accuracy. In time series analysis, dynamic time warping (DTW) is an algorithm for measuring similarity between two temporal sequences which may vary in speed. For instance, similarities in walking could be detected using DTW, even if one person was walking faster than the other, or if there were accelerations and decelerations during the course of an observation. DTW has been applied to temporal sequences of video, audio, and

graphics data — indeed, any data which can be turned into a linear sequence can be analyzed with DTW.

The goal of dynamic time warping (DTW for short) is to find the best mapping with the minimum distance by the use of DP. The method is called "time warping" since both xx and yy are usually vectors of time series and we need to compress or expand in time in order to find the best mapping. We shall give the formula for DTW in this section.

Let tt and rr be two vectors of lengths mm and nn, respectively. The goal of DTW is to find a mapping path {(p1,q1),(p2,q2),...,(pk,qk)}{(p1,q1),(p2,q2),...,(pk,qk)} such that the distance on this mapping path ∑ki=1|t(pi)−r(qi)|∑i=1k|t(pi)−r(qi)| is minimized, with the following constraints:

- Boundary conditions:(p1,q1)=(1,1)(p1,q1)=(1,1), (pk,qk)=(m,n)(pk,qk)=(m,n). This is a typical example of "anchored beginning" and "anchored end".
- Local constraint: For any given node (i,j)(i,j) in the path, the possible fan-in nodes are restricted to (i−1,j)(i−1,j), (i,j−1)(i,j−1), (i−1,j−1)(i−1,j−1). This local constraint guarantees that the mapping path is monotonically non-decreasing in its first and second arguments. Moreover, for any given element in tt, we should be able to find at least one corresponding element in rr, and vice versa.

Pseudo code

```
IntDTWDistance(s: array [1..n], t: array [1..m]) { DTW := array [0..n, 0..m]
For i: = 1 to n
DTW [i, 0] := infinity fori := 1 to m DTW[0, i] := infinity DTW[0, 0] := 0
For i: = 1 to n for j: = 1 to m
Cost: = d(s[i], t[j])
DTW [i, j] := cost + minimum(DTW[i-1, j ], // insertion DTW[i , j-1], // deletion
DTW [i-1, j-1]) // match return DTW[n, m]
}
```

Algorithms for Pressure and Pen-ups

The pressure values are averaged and also the maximum pressure value from the set of pressure values is found. Using these two values, the verification using pressure is done.

For the pen-ups, the numbers of pen-ups are counted while registering and is not subjected to a change that is no algorithm is applied to it. This value is directly used to compare the number of pen-ups when the user performs authentication.

III. RESULTS AND DISCUSSION

The screenshots of our implemented application are taken and are shown as follows

Figure 5.1 Home screen

This is the home screen of the application, which has three options as we can see in the diagram, which are register, verify and database.

Figure 5.2 Register1

On clicking the register button this is the first screen we get where the user enters the name and sign in the box.

Figure 5.3 Register2

After submitting the first signature, we get the screen as shown in figure 5.3, where the name is already present and the user signs for the second time and clicks next.

Figure 5.4 Register3

This is the final sign done by the user for registration process and on clicking submits we get the next screen.

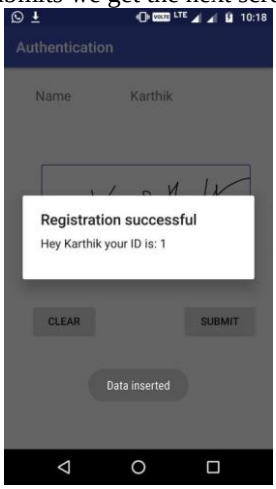

Figure 5.5 Registration

This is the output screenshot when a user registers his signature using the application. After submitting the third signature, the user sees the screen in Figure 5.1. This is confirmation to the user that his registration is successful and an ID is generated on his name.

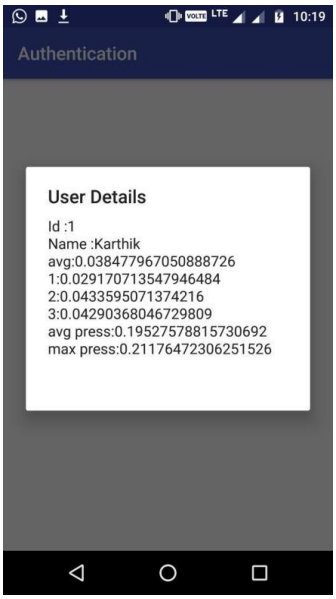

Figure 5.6 Feature vector

The screen output shows the user details when a user accesses the database. All the user‘s ID and the corresponding details are listed here, where we can search and get specified user details. The screen shows a user named _Karthik‘ and his details.

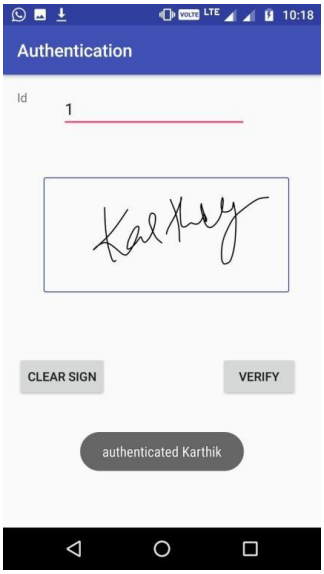

Figure 5.7 Verification

The figure shows a screenshot of verification of signature, where the user signs using his ID and submits for authenticating the transaction and he is prompted back with a message saying authenticated.

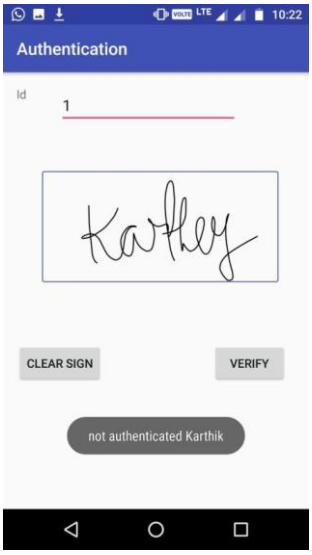

Figure 5.8 Forgery

As the figure shows, the transaction is not authenticated as in this case the sign was forged by some other person and the app denies the authentication request.

IV. CONCLUSION AND FUTURE SCOPE

We conclude that our work on dynamic properties for verification process has yielded promising results and thus helps in improving the efficiency in authenticating a transaction. We believe that every problem needs a unique solution and we have provided one for this.

In our work, the pressure values are averaged for a signature and this value is used in the feature vector. Instead of averaging or finding the maximum of the values, the pressure values can be analysed. This analysis requires a dataset and this project also provides a way to perform analysis. By analysing these values, a better algorithm can be implemented which can improve the efficiency of the algorithm.

There is another important dynamic property which we can implement, that is the orientation angles. Due to the lack of hardware support on the smart phones during this project implementation, this dynamic property couldn‘t be implemented. In future, if there is any support to obtain the orientation angle while signing, this can be added to the implementation, which will add another layer of security to the authentication method.

V. REFERENCES

[1] Madasu Hanmandlu, Farrukh Sayeed, Shantaram Vasikarla, ‖Online Signature Verification using the Entropy Function‖, 2015 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)

[2] Xiaoyu Song, Xinghua Xia and Fangjun Luan, ―Online Signature Verification Based on Stable Features Extracted Dynamically‖, 2016 IEEE Transactions on Systems, Man, and Cybernetics: Systems

[3] Michał Lech, Andrzej Czyżewski ―A handwritten signature verification method employing a tablet‖, 2016 Signal Processing: Algorithms, Architectures Arrangements, and Applications (SPA)

