Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 928039, 8 pages
http://dx.doi.org/10.1155/2014/928039

Research Article

Cubic Bezier Curve Approach for Automated Offline Signature Verification with Intrusion Identification

Arun Vijayaragavan,[1] J. Visumathi,[2] and K. L. Shunmuganathan[3]

[1] *Department of Information Technology, R.M.D. Engineering College, Chennai, Tamil Nadu, India*
[2] *Department of Computer Science and Engineering, Jeppiaar Engineering College, Chennai, Tamil Nadu, India*
[3] *Department of Computer Science and Engineering, R.M.K Engineering College, Chennai, Tamil Nadu, India*

Correspondence should be addressed to J. Visumathi; jsvisu@gmail.com

Received 4 March 2014; Revised 20 May 2014; Accepted 24 July 2014; Published 24 July 2014

Academic Editor: Massimo Scalia

Authentication is a process of identifying person's rights over a system. Many authentication types are used in various systems, wherein biometrics authentication systems are of a special concern. Signature verification is a basic biometric authentication technique used widely. The signature matching algorithm uses image correlation and graph matching technique which provides false rejection or acceptance. We proposed a model to compare knowledge from signature. Intrusion in the signature repository system results in copy of the signature that leads to false acceptance. Our approach uses a Bezier curve algorithm to identify the curve points and uses the behaviors of the signature for verification. An analyzing mobile agent is used to identify the input signature parameters and compare them with reference signature repository. It identifies duplication of signature over intrusion and rejects it. Experiments are conducted on a database with thousands of signature images from various sources and the results are favorable.

1. Introduction

Biometric authentication system involves individual authentication using their traits. Authentication compares iris, fingerprints, voice, and signature with the input. Signature authentication is widely used in various domains such as banks and other government systems. Signature style may vary during course of time and cannot perfectly match with contents in signature repository. Intrusion in the system may involve duplication of the signature which leads to false acceptance.

Mobile agent is a piece of code that migrates from one host to another and can execute code in parallel. It maintains state and can execute to store data. Mobile agent in heterogeneous network is used in distributed applications. It can operate without any active connection between the server and client. Change in network may not affect the migrating nature of the mobile agent since routing algorithm determines the path. In our model, mobile agent plays a vital role to carry out the signature comparison. It can execute on all types of computers since their code is not installed in the host.

Bezier curve is frequently used algorithm to draw an arc in computer graphics. Smooth curves are plotted using Bezier curve algorithm and can be scaled indefinitely. As it is referred to as the curve that moves with velocity over time with controlling points, it suits signature matching. Signature is considered as a linear and curvy drawing with continuous nature and behaviors. Bezier curve is a parametric curve equation used for smooth curve to scale without deformation. Controlling points determine the orientation of the curve. Consider

Bezier Cubic Formula

$$= (1-t)^3 SP_0 + 3t(1-t)^2 P_0 + 3t^2 (1-t) SP_1 + t^3 P_1, \quad \text{where } t \epsilon [0,1], \tag{1}$$

SP_0 and SP_1 are controlling points, and P_0 and P_1 are curve start and end points. Figure 1 defines the controlling points of the Bezier curve.

Our approach uses mobile agent named analyzing mobile agent to match signature in reference signature repository

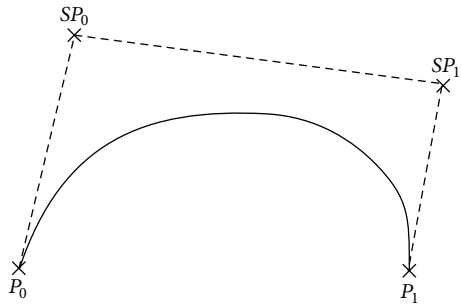


Figure 1: Cubic Bezier curve.

which is not compromised. Earlier approaches involved identification by analyzing (x, y) point. Correlation between the points of the input and source is plotted. Correlated points are directly proportional to the genuine signature. The intruded match for the copy of the source signature may authorize the copied signature. We proposed a novel method for signature matching with intrusion prevention features. Certain algorithm uses Bezier curve over the signature and identifies the point match. We propose an algorithm to analyze each curve using Bezier curve with behavioral parameters and use mobile agent to verify the signature in various sources.

Figures 2(a), 2(b), and 2(c) show signatures from various three sources named "test" and are used throughout the paper for illustration.

2. Literature Survey

Many models for signature identification have been proposed earlier by various authors. Bertolini et al. [1] introduced a new graphometric feature that considers the curvature of the most important segments or curves of the signature. Bezier curve is extracted and ensemble of classifiers based on graphometric features to improve the reliability of the classification, hence reducing the false acceptance. The idea was to simulate the most important segments of the signature by using Bezier curves and then extract features from them.

Osadchy et al. [2] proposed a solution for problem of matching images from the same scene that are viewed from different lighting conditions. A mixed strategy of normalized correlation of small windows and comparison of multiscale oriented filters is used to compare images. An experiment over synthetic images and real images is validated effectively.

Ferrer et al. [3] proposed a technique to authenticate a signature in complex background such as cheque or invoice. The signature model is trained with white background and is used to test the input using (x, y) plotting technique. False acceptance occurs when exact signature is given as input.

Bansal et al. [4] exposed a signature matching algorithm with critical regional matching. The signatures generate a region that is required to match the system. A region of area is identified using area plotting technique and is matched with the area generated by the input. But this technique fails when the input is different between the signatures but generates same area of source signatures.

Riskus [5] proposed an algorithmic approach to identify the Bezier curve using circular arcs. His work involves identification of Bezier curve using circular arc matching technique. The existing models do not specify any means of intrusion identification. Intrusion can be prevented in our model when there is an exact match of the input. This reveals that the repository that contains the signature is compromised.

Lakshmi and Nayak [6] implemented an adaptive machine learning technique called a multilayered neural network model (NN model). Using image processing, the quality of images is improved and a huge collection of data is generated from genuine signatures and signatures forgery. Decision making capabilities are improved using NN model and knowledge is continuously updated.

Arun and Shunmuganathan [7] developed a mobile agent that migrates from server to host to identify intrusion in a system. The shielded mobile agent in their proposal migrates to the client host handles the file migration in cloud computing. Any change in memory checksum of the host results in intrusion detection and prevention measures are carried out. Intrusion in signature host can be identified by the HIDS of the system when exact copy of signature is encountered as input.

3. Behavioral Parameters

During signature process, user performs linear and curved drawing with various behavioral methods. Behavioral methods are as follows.

Start Distance. Signer's start position from the margin plays a vital role in verification. The start point is calculated for each sample and an average point is determined and is added as a parameter.

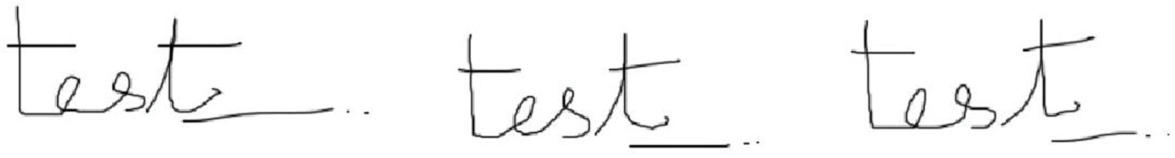
Dark Impression. Signer usually makes an impression over certain parts in the signature. Noise removal approach used for normalization removes those behaviors in prior approaches. These behaviors are used to verify the signer.

Underlined and Dotted. Most signers arrive at a dotted or/and underlined style at the end of the signature. These positions over the margin and the start point of the signature are useful parameters for identification.

4. Analyzing Mobile Agent

Signature verification in one source is insufficient to identify and authorize genuine input. Intrusion in such a system may lead to signature copy, leading to false acceptance. Mobile agent is used to identify the other sources and matches the signature with the input.

Figure 3 describes the architecture of the proposed model. User input (signature) is fed into the signature verification system and compared with the local database. AMA carries the input and compares with other sources.



(a) (b) (c)

Figure 2: (a) Source 1 signature. (b) Source 2 signature. (c) Source 3 signature.

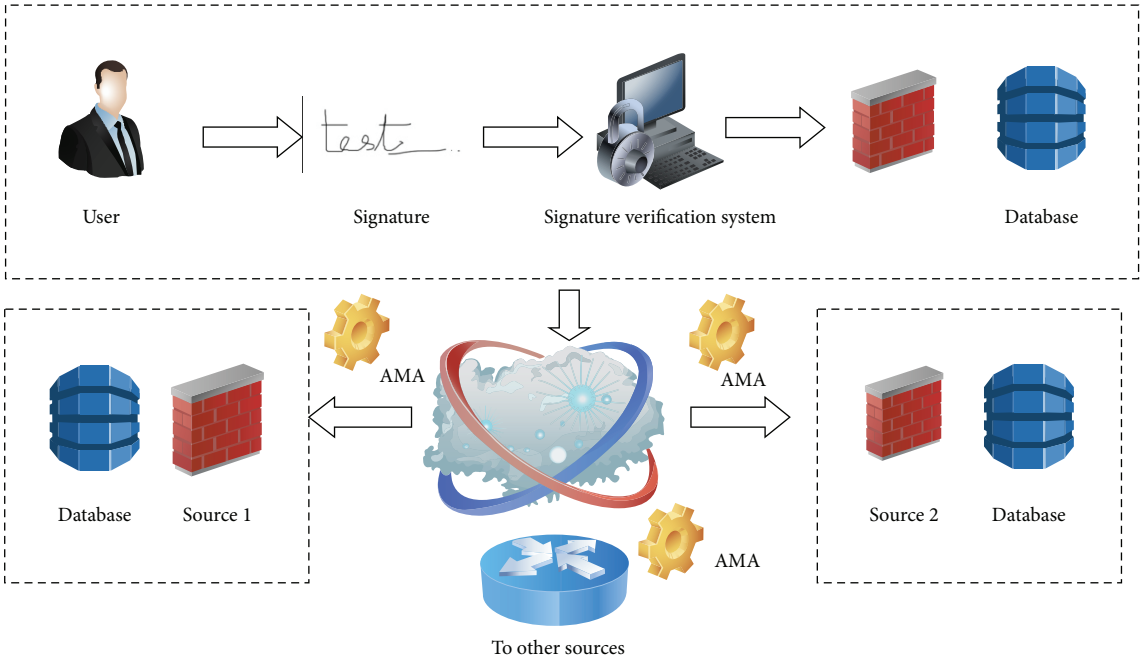


Figure 3: Architecture.

Algorithm 1 demonstrates how AMA is involved in signature verification process.

This algorithm extracts the curve from the signature image. Bezier curves are segmented from the signature by examining the color of the pixel. Change in the color for parameters such as alpha, red, green, and blue is used to identify the curves. Bezier curves are identified by the long deviation in pixel path. Bezier curves are added in an array and each curve is processed in Algorithm 2.

To determine the controlling points and start-end points of the Bezier curve, input curve is plotted on the graph. The start and end points are easily identified by tracking the pixel with change in the pixel color. A new Bezier curve is drawn with the same start-end points and adjusted to the path that matches the input curve. When both curves follow the same pixel path, the controlling curve is identified.

The slope and distance of the Bezier curve are identified using the above algorithm. The Bezier curve and their corresponding start-end points and controlling points are determined; the slope is determined using the formula. The distance between slopes is calculated from the midpoint of the regional and controlling slopes, as shown in Algorithm 3.

```
(1)  Function CURVES[] GETCURVES(IMAGE)
(2)  Begin
(3)      Curves[] BezierCurves;
(4)      Pixel[] SignatureSegments;
(5)      for each(pixel in Image)
(6)          If (Color of pixel is not default color)
(7)              SignatureSegments.Add(pixel);
(8)          End if
(9)      End for
(10)     Arrange pixels in SignatureSegments
(11)     Create new BezierCurve;
(12)     for each(pixel in SignatureSegments)
(13)         If(Change in (x, y) is more than previous pixel)
(14)             BezierCurves.Add(BezierCurve);
(15)             Create new BezierCurve;
(16)         Else
(17)             BezierCurve.Add(pixel);
(18)         End if
(19)     End for
(20)     return BezierCurves
(21) End
```

Algorithm 1

```
(1)  Function GETCONTROLLINGANDSTART-ENDPOINTS(BEZIERCURVE BC1)
(2)  Begin
(3)    Get Start-End points of BezierCurve BC1
(4)    Plot a new BezierCurve BC2 with Start-End points of
       BezierCurve BC1
(5)    Adjust controlling points of BezierCurve BC2 to
       match BezierCurve BC1
(6)    if (BC1 == BC2)
(7)      GetControllingPoint of BezierrCurve BC2
(8)      Return the Start-End points and ControllingPoint
(9)    End if
(10) End function
```

Algorithm 2

```
(1)  Function GET PARAM ETERS(Image s1)
(2)  Begin
(3)    Get Curves[] BezierCurves = GetCurves(s1)
(4)    for each (Curve BC in BezierCurves)
(5)    Plot the BC in the graph
(6)    end for
(7)    ControllingPoints(SP1, SP2)andStart-EndPoint(P0, P1) = GetControllingAndStart-EndPoint(BC);
(8)      Plot Start-End points and Controlling Points
(9)      Find Regional Slope (m1)P0 = m1P1 + c  (c constant)
(10)     Find Controlling Slope (m2)SP1 = m2SP2 + c
(11)     Find Midpoint of Regional Slope: (PX, PY)
(12)     Find Midpoint of Controlling Slope: (SX, SY)
(13)     Find Distance: d = sqrt((PX − SX)^2 − (PY − XY)^2)
(14) End
```

Algorithm 3

5. Illustration

5.1. Bezier Points Identification. Normalized signature is used to identify lines perfectly to avoid noises in older systems. But this may remove the behavioral parameters. Initially, behavioral parameters are identified from the input signature such as dark spot. Dark spots (x, y) points are identified and are added as input. The signature is normalized to get clear curve path to identify curves. It is segmented into cubic Bezier curves to identify the Bezier parameters. Controlling points and start-end points are identified for each curve. The array of parameters for each curve is identified using pixel identification technique which scans the input signature with color change identification to identify signature path. Figure 4 is the input signature to determine the slope and distance.

The Bezier curves that are identified are segmented to identify the Bezier points. Figure 5 shows the signature is segmented into Bezier curves. Hough transform algorithm is used to detect curves in the signature. It uses two-dimensional array known as accumulator to detect shapes. Scanning process was carried out in every pixel to identify the evidence of a curve with an immediate change in the curve (x, y) coordinate. Immediate change in the angle of the curve determines the separation of curve points in the signature.

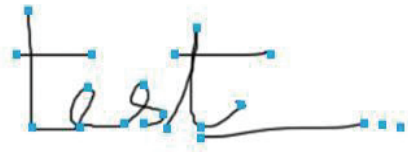


Figure 4: Bezier initial curve and linear identification.

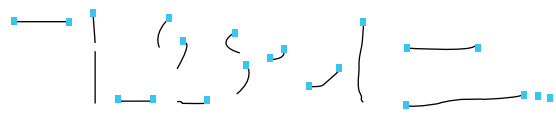


Figure 5: Segmented signature.

5.2. Controlling Points Identification. Segmented curves are subjected to identify controlling points and start-end points. Each segment is an input to the Bezier identification block. Start-end points are plotted over the segment and a Bezier curve is plotted over it. The controlling points are adjusted to identify the exact controlling points. When the drawn curve matches the segment, the points of the controlling system

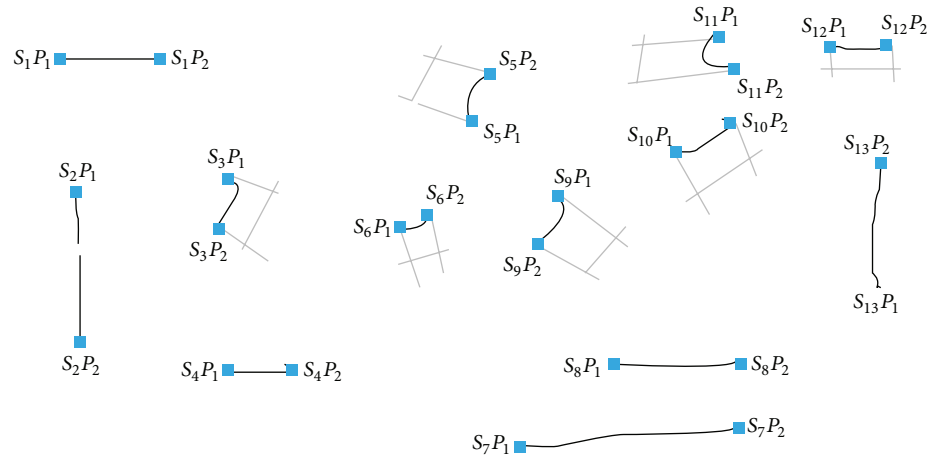


Figure 6: Controlling point determination.

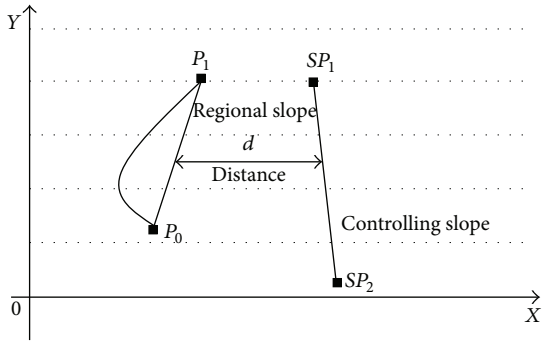


Figure 7: Slope and distance determination.

are saved for next steps. Figure 6 describes the segmented particles of the signature with controlling and start-end points.

5.3. Slope Creation. Points may vary when start point of signature varies. Even the slope of the start-end points and controlling point may not vary but the start point of the signature varies. Regional slope is the slope of the start-end points and controlling slope is the slope between controlling points. Distance between slopes may vary; midpoint of the slope is identified as d. Figure 7 defines the identification of slope and distance between the slopes. Consider the following.

Regional slope: $P_0 = mP_1 + c$ (c constant).

Controlling slope: $SP_1 = mSP_2 + c$ (c constant).

Midpoint of regional slope: (PX, PY).

Midpoint of controlling slope: (SX, SY).

So, distance: $d = \sqrt{(PX - SX)^2 - (PY - XY)^2}$.

5.4. Bezier Parameter Comparison. AMA recognizes signature as the following data:

Bezier segment count (n),

Segment start-end points (S-E),

Segment controlling points (C),

Segment regional slope (RS),

Segment controlling slope (CS),

Distance between regional and controlling slopes (d).

AMA compares signature parameters with the self-signature repository. And when a proper match is found, it searches for reference signature repository system.

AMA performs asynchronous encryption technique to encrypt parameter with public key of the service. AMA duplicates itself and performs the check in reference signature repository. Successful parameter matches authenticate the user signature. Threshold range is a range in which the calculated compared results of signatures in sources lie for acceptance of the input. It is calculated depending upon the security level which ranges from 85% to 99%. Values that lie outside the range are assigned as invalid input.

6. Result and Experiment

We implemented simulation for analyzing mobile agent and used three signature sources. C number framework and WCF services are used for mobile agent transmission. WCF services are mounted for AMA propagation to signature sources and compare the data. 837 signature samples are loaded and proposed; signature matching algorithm is executed. It shows an effective result compared to image correlation and graph matching algorithm.

Input 1 to input 3 are online input signatures captured from electronic signature pen. Input 4 to input 10 represent offline signature and are scanned as image. Figures 8 and 9 describe compared result of false rejection and false acceptance of the signature input. Proposed algorithm shows an effective result over existing algorithm.

Figure 10 shows ten sample inputs to demonstrate how each parameter is used to determine the signature comparison. Table 1 shows the acceptance criteria based upon the parameters value false on the threshold range.

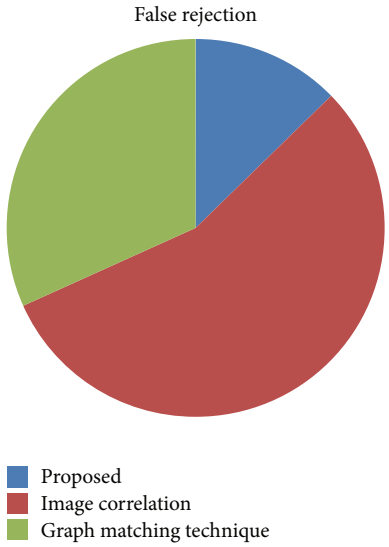


Figure 8: False rejection comparison result.

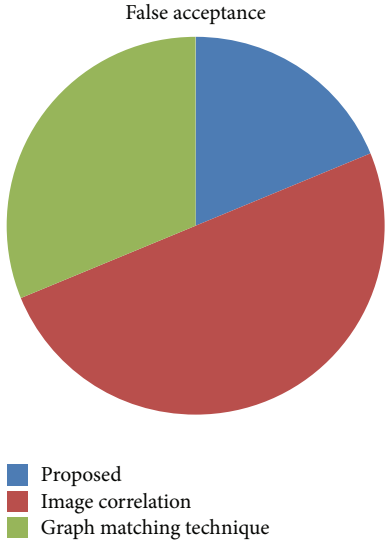


Figure 9: False acceptance comparison result.

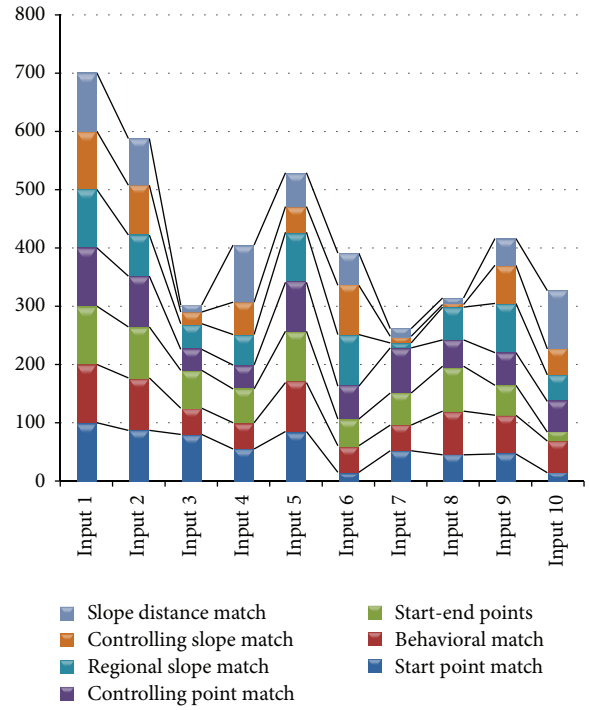


Figure 10: Signature compared result.

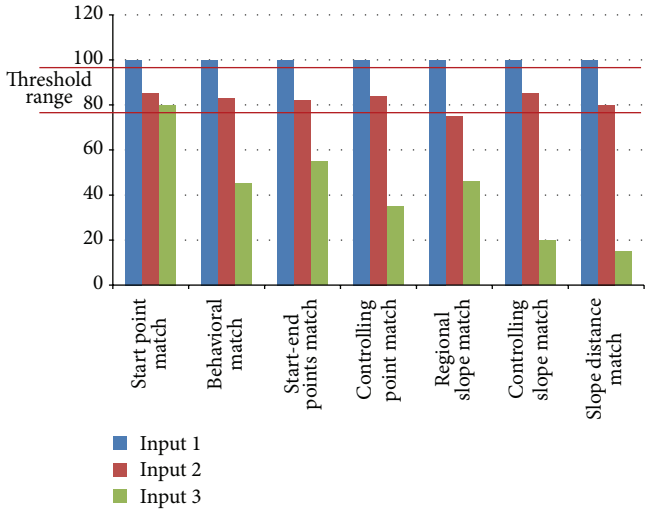


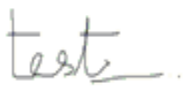
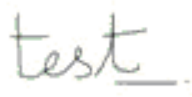
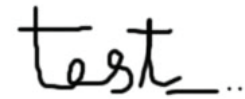
Figure 11: Threshold range illustration.

Threshold range determines the value at which the input signature is selected. Threshold value ranges from 74.3% to 94.55% determined from series of experiments over all possible inputs. Exact copy result is 100% exact match and may not fall between threshold regions. Small change in the copy of signature creates significant change in the controlling point. A change in the controlling and starting points of a single curve may result in significant change. For example, minor change in above Bezier curve's starting point may result in significant change in other parameters since slope distance depends on it. Thus, experimental result determines the minor change may fall parameter value below 72.5%.

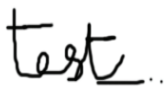
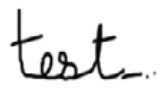
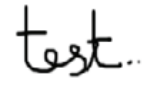
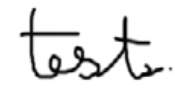
Figure 11 shows comparison between various signature inputs. Input 1 represents the copy of intruded source, input 2 represents original authenticated signature, and input 3 represents forensic signature. Since input 1 is a copy of original signature in the repository result in full match that overflows from threshold value, each and every match for input 1 is full that results in rejection and AMA concludes that the corresponding source is compromised. Input 1 matches all parameters exactly. This illustrates that the signature repository is compromised and input is rejected. Input 2 matches all parameters above the threshold range. Thus, it is accepted as valid signature. Input 3 does not match any parameter above threshold values and it is rejected and determined as invalid input. Our approach detects the

Table 1: Acceptance criteria based on threshold range.

(a)

	Input 1	Input 2	Input 3	Input 4
Sample				
Start Point Match	100%	87%	80%	80%
Behavioral Match	100%	90%	45%	45%
Start-end Point Match	100%	87.5%	52%	65%
Controlling Points Match	100%	87.1%	36%	36%
Regional Slope Match	100%	70.6%	44%	44%
Controlling Slope Match	100%	85.4%	20%	20%
Slope Distance Match	100%	80.21%	12%	12%
Result	Rejected	Accepted	Rejected	Rejected

(b)

Input 5	Input 6	Input 7	Input 8	Input 9	Input 10
					
55%	85%	15%	52%	78%	14.3%
45%	84%	45.6%	44%	75%	54.3%
58%	88%	45.8%	55%	77%	15.4%
41%	84%	56.4%	77.5%	82%	52.6%
52.4%	86%	88.5%	8.5%	79.5%	45.6%
55.5%	82%	84.6%	11.4%	99%	44.5%
98%	77%	55.2%	13.5%	75.8%	100%
Rejected	Accepted	Rejected	Rejected	Accepted	Rejected

behavior of signature as simple Bezier curve with controlling point's results in efficient authentication.

Input 2 carries proper behavior and Bezier curve parameters and lies between the threshold values. As a result, the input is accepted. Input 3 has improper Bezier curve parameters due to improper plotting of the controlling and regional slopes, leading to rejection.

7. Conclusion

We proposed a new signature verification technique with Bezier curve approach to match input data. We also proposed a mobile agent called analyzing mobile agent that propagates to various sources and compares the inputs. Moreover, comparing the signature undergoes efficient steps to avoid false acceptance or rejection. From the experimental results, it has been observed that signature has been compared efficiently, compared to old graph matching technique. If an intrusion occurs in the signature, system will not affect our verification technique. Future work could be use of AMA in other biometric authentication processes such as iris and fingerprint matching.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] D. Bertolini, L. S. Oliveira, E. Justino, and R. Sabourin, "Reducing forgeries in writer-independent off-line signature verification through ensemble of classifiers," *Pattern Recognition*, vol. 43, no. 1, pp. 387–396, 2010.

[2] M. Osadchy, D. W. Jacobs, and M. Lindenbaum, "Surface dependent representations for illumination insensitive image comparison," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 1, pp. 98–111, 2007.

[3] M. A. Ferrer, J. F. Vargas, A. Morales, and A. Ordóñez, "Robustness of offline signature verification based on gray level features," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 966–977, 2012.

[4] A. Bansal, B. Gupta, G. Khandelwal, and S. Chakraverty, "Offline signature verification using critical region matching," *International Journal of Signal Processing, Image Processing and Pattern*, vol. 2, no. 1, 2009.

[5] A. Riskus, "Approximation of a cubic bezier curve by circular arcs and vice versa," *Information Technology and Control*, vol. 35, no. 4, pp. 371–378, 2006.

[6] K. V. Lakshmi and S. Nayak, "Off-line signature verification using Neural Networks," in *Proceedings of the 3rd IEEE International Advance Computing Conference (IACC '13)*, pp. 1065–1069, February 2013.

[7] V. Arun and K. L. Shunmuganathan, "Secure sandbox for mobile computing host with shielded mobile agent," *Indian Journal of Applied Research: Information Technology*, vol. 3, no. 9, pp. 47–48, 2013.