Enhanced Pattern Unlocking in Smartphones

Ayush Gupta
IIIT Delhi
(MT18002)
ayush18002@iiitd.ac.in

Harsh Kumar Jain
IIIT Delhi
(MT18006)
harsh18006@iiitd.ac.in

Wrik Bhadra
IIIT Delhi
(MT18027)
wrik18027@iiitd.ac.in

I. Problem Statement

As per a 2018 report, there are about 2.4 billion smartphone users across the globe, and half of them use a pattern to unlock their phone. However, this unlocking method is prone to smudge attack and over-the-shoulder snooping. A study conducted by researchers of US Naval Academy and the University of Maryland, Baltimore County claims that 2 out of 3 people can easily recreate a pattern just after viewing it once from 5-6 ft. away. In our current project, we propose to address the attacks mentioned above by examining and incorporating additional factors that can significantly improve the unlocking mechanism.

II. Motivation

Improving this system allows us to reach a large base of people who use a pattern as their unlocking mechanism. Our motivation behind developing this system so to enhance the security of smartphones. Our research suggests that this is a relatively unexplored topic and hence working on this might open up new avenues. In regards to this, we recount an experience our friend had when a small girl was able to unlock his phone just by looking at the trace created on the screen; essentially a smudge attack

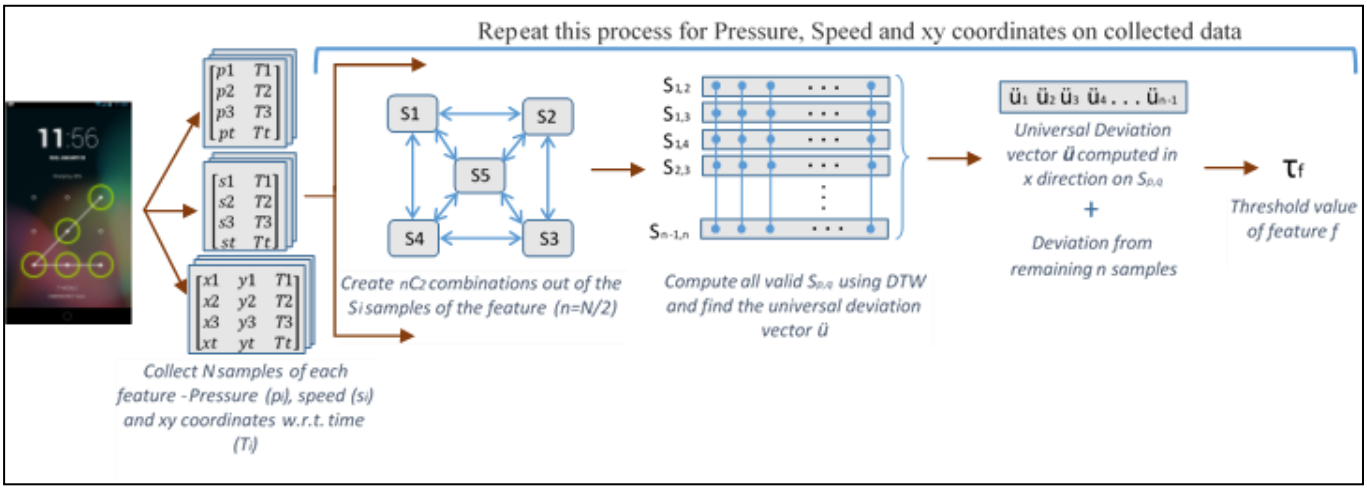

Figure 1 Model

III. Methodology

1. Let the user draw the pattern for 'N' times.
2. The model gets N*3 feature vectors as input.
3. For each feature (pressure, speed and coordinate) 'f'
   a. Choose n=N/2 random vectors.
   b. Create nC2 combination out of n chosen vectors
   c. For each combination of vector (Sp, Sq):
      i. Compute the DTW distance [1] between Sp and Sq using Euclidean distance as the measuring parameter.
      ii. Create a 1-d cost vector (Spq) using the path along the cost matrix produced by DTW.
   d. Compute the Universal Standard Deviation vector (ü) by calculating the standard deviation along 'x' axis across all cost vectors (Spq).
   e. For remaining 'n' vectors (Sp')
      i. Compute the DTW distance using Euclidean distance as the measuring parameter between Sp' and each vector chosen in step 3(a).
      ii. Create a 1-d cost vector (Spq') for each vector of 3(a) using the path along the cost matrix produced by DTW.
      iii. Compute the Standard Deviation vector (ü') by calculating the standard deviation along 'x' axis across all cost vectors (Spq').
      iv. Compute the Standard Deviation (sd) between ü and ü'.
      v. Compute the Standard Deviation (sd') along y-axis on sd.
   f. Now, we have 'n' values of sd' computed from previous steps. Let this vector be (V).
   g. Compute the z-score of V and eliminate the outliers having score greater than 2.0
   h. Compute the threshold (τf)
      τf = min (max(V), 2.5*std(V)+mean(V))
   i. For SVM, train it using values from V.
4. For testing, the input vector of feature f (If') is combined with the n input vector of feature f and the deviation vector (ü'') is created.
5. Compute the standard deviation vector (σ) along x axis between ü and ü''. Then find the standard deviation (sd) of σ.
6. Using DTW:
   If 0<= sd <= τf, the user is genuine, otherwise not.

7. Using SVM:

If the SVM detects sd inside the cluster then the user is genuine, otherwise not.

Data Collection (from Android application):

1. Developed our own Android application that can capture the x-y coordinates and pressure at 't' time instances while drawing the pattern. 't' can vary between different samples.
2. Calculate the speed at 't' instances with distance equal to Euclidean distance between x-y coordinate at ith time instance and (i-1)th time instance and time equal to time difference between the instances.
3. Create buffers of pressure vs time, speed vs time and coordinate vs time, every time the user draws the pattern and send the data to the server using Python Flask Server.

IV. RESULTS

We have tested the following combination of algorithms and models:

1. Using only DTW

TABLE I. CONFUSION MATRIX USING ONLY DTW

Total = 40	Predicted (TRUE)	Predicted (FALSE)
Actual (TRUE)	~85% (17/20)	~ 15% (3/20)
Actual (FALSE)	~10% (2/20)	~90% (18/20)

Precision = 86%
Recall = 85%
F1 Score = 0.85

2. Using only SVM

TABLE II. CONFUSION MATRIX USING ONLY SVM

Total = 40	Predicted (TRUE)	Predicted (FALSE)
Actual (TRUE)	~65% (13/20)	~ 35% (7/20)
Actual (FALSE)	~5% (1/20)	~95% (19/20)

Precision = 92%
Recall = 65%
F1 Score = 0.76

3. Using DTW for first 50 True positives and then SVM on trained model

TABLE III. CONFUSION MATRIX USING BOTH DTW AND SVM

Total = 40	Predicted (TRUE)	Predicted (FALSE)
Actual (TRUE)	~60% (12/20)	~ 40% (8/20)
Actual (FALSE)	~0% (0/20)	~100% (20/20)

Precision = 100%
Recall = 60%
F1 Score = 0.75

V. TAKEAWAY FROM RESULT

As per the results mentioned above, we can observe that combination of SVM and DTW results best in reducing False Positives, which is one of the key factor in authentication systems. However, it also reduces the True Positives by a significant amount. This is because, SVM tries to create strict clusters using the values of True Positives which later on decreases the accuracy (also depicted by F1 score).

DTW and SVM alone perform best in detecting True Positives but lack to gain 100% accuracy in True Negatives. However, certain applications would not require such high accuracy therefore, these methodologies can be used.

SVM alone is very susceptible to noise in less data, hence, the accuracy depends highly on initial model training. User needs to very highly accurate while training the model in SVM only methodology.

VI. SUMMARY

1. The data for speed, x-y coordinates and pressure is collected every time. No such database was available.
2. The algorithm returns result in yes/no format for each of the feature (speed, x-y coordinates and pressure). The final result is obtained using the following approach:
    a. If none of the result is yes, the final output is No.
    b. If only one of the result is yes, the final output is No.
    c. If two of them are yes
        i. If only Pressure is No, the final output is Yes.
        ii. If one of Speed and x-y coordinate is no, then the threshold is relaxed by 10% to check if the result is Yes. In this case, the final result is Yes, otherwise No.
    d. If all three are Yes, the final result is Yes.

    The algorithm gives higher weightage to Speed and x-y coordinates because every smartphone doesn't have pressure sensor which may lead to spurious results.
3. Binary classification was not feasible. It is so because, only the data of valid user is available and no data for negative or invalid user is available for such kind of classification. This motivated us to use one-class SVM in later stages.
4. We used DTW distance to find the difference between the time-series data. We also tried to use Simple-Euclidean distance and Frechet Distance but these are single valued outputs which did not help us to gain good results.
5. Thresholds are found using Z-score among the values. We also tried to use mean of highest frequency histogram but due to outliers this did not help us to gain good results.

6. We also tried using max and mean of values for calculating thresholds, but again, due to certain outliers this was not a good choice.
7. In the initial stages, we took standard deviation among feature vector as a single value and tried to obtain the results. This did not work because it did not give a clear picture about the part which had major difference in values. Later on, we decided to form 1-d vectors of standard deviation which provided higher accuracy in results.
8. Data was collected in 3-dimensions for feature like x-y coordinate. However, a slight time difference amplifies while the user draws the pattern leading to unwanted failure in authentication. We removed the time dimension to overcome this limitation.

VII. CONCLUSION

Enhanced pattern unlocking helped us to achieve results with accuracy more than ~90% in some methodologies. This proves that even simple features like speed and x-y coordinates can help to improve authentication systems for mobile devices. This however, has few limitations, such as, the user should not use very simple patterns because it would not create much difference when other users try to draw the same. Also, to achieve more accuracy, the user may want to train the model on more samples and more accuracy between the samples.

This method can be extended to several other authentication mechanisms like keyboard etc.

The accuracy can be further improved by fine-tuning the parameters of threshold and SVM.

REFERENCES

[1] Stan Salvador, and Philip Chan. “FastDTW: Toward accurate dynamic time warping in linear time and space.” Intelligent Data Analysis 11.5 (2007): 561- 580.

[2] I. Nigam, M. Vatsa and R. Singh. Leap signature recognition using hoof and hot features. IEEE International Conference on Image Processing (ICIP), 2014.