U-Stroke Pattern Modeling for End User Identity Verification Through Ubiquitous Input Device

Tapalina Bhattasali, Nabendu Chaki, Khalid Saeed, Rituparna Chaki

To cite this version:

Tapalina Bhattasali, Nabendu Chaki, Khalid Saeed, Rituparna Chaki. U-Stroke Pattern Modeling for End User Identity Verification Through Ubiquitous Input Device. 14th Computer Information Systems and Industrial Management (CISIM), Sep 2015, Warsaw, Poland. pp.219-230, 10.1007/978-3-319-24369-6_18 . hal-01444467

HAL Id: hal-01444467

https://hal.inria.fr/hal-01444467

Submitted on 24 Jan 2017

U-Stroke Pattern Modeling for End User Identity Verification through Ubiquitous Input Device

Tapalina Bhattasali[1], Nabendu Chaki[1], Khalid Saeed[2], Rituparna Chaki[3]

[1]Department of Computer Science & Engineering, University of Calcutta, Kolkata, India
tapolinab@gmail.com,nabendu@ieee.org
[2]Faculty of Computer Science, Bialystok university of Technology, Bialystok, Poland
khalids@wp.pl
[3]A.K.Choudhury School of Information Technology, University of Calcutta, Kolkata, India
rchaki@ieee.org

Abstract. Identity verification on ubiquitous input devices is a major concern to validate end-users, because of mobility of the devices. User device interaction (UDI) is capable to capture end-users' behavioral nature from their device usage pattern. The primary goal of this paper is to collect heterogeneous parameters of usage patterns from any device and build personal profile with good-recognition capability. This work mainly focuses on finding multiple features captured from the usage of smart devices; so that parameters could be used to compose hybrid profile to verify end- users accurately. In this paper, U-Stroke modeling is proposed to capture behavioral data mainly from smart input devices in ubiquitous environment. In addition to this, concept of CCDA (capture, checking, decision, and action) model is proposed to process U-Stroke data efficiently to verify end-user's identity. This proposal can draw attention of many researchers working on this domain to extend their research towards this direction.

Keywords: U-Stroke · Smart Device · Touch Screen · Ubiquitous Input Device · Identity Verification

1 Introduction

Nowadays, mobile devices become smarter by offering multiple types of computing services at any place and at any time. Ubiquitous input devices [1] (smart phone, tablet, phablet, PDA, laptop, netbook etc.) become rich source of personal data, due to its support towards "any" paradigm. Sensitive personal data (such as password, financial information, health records [2]), stored in mobile devices are growing day by day. As a result, they are becoming attractive target to be attacked. Accurate identity verification of end-user is becoming a major requirement to preserve confidentiality and integrity in uncontrolled environment.

Traditional authentication mechanism (PIN/Password based) can be easily compromised. Anyone can access all services and can misuse personal information stored in the device, if device is misplaced. Implicit authentication mechanism needs to be

adfa, p. 1, 2011.

considered without affecting normal usage pattern to overcome existing weak authentication methods. Typing on computer keyboard is completely different from typing on smart devices having small keypads [3] and sensors enabled touch screen. Today, user-friendly touch screens are widely used on many devices such as mobile phones, tablets, and computers. As smart devices can perform multiple tasks at the same time, data acquisition only during typing is not efficient to build user profile. Beside this, typing on touch-screen based smart devices becomes more error-prone compared to computer keyboard based typing. Only temporal data based keystroke analysis is not sufficient to build unique user profile for any ubiquitous device. Therefore, keystroke analysis needs to be merged with touch screen based gesture analysis to enhance success rate of identity verification. Keystroke dynamics is mainly considered for desktop computers [2]. Since touch devices comprise sensors to capture environmental changes, they offer more capabilities to authenticate users accurately. Ubiquitous devices consider a different type of Human to Machine (H2M) communication by changing traditional way of human-computer interaction (HCI).

The major contribution of this paper is to propose a novel H2UID (human to ubiquitous input device) interaction mechanism U-Stroke (ubiquitous stroke) that can be applied to any computing device. This type of H2M communication is considered here to verify identity of end-user either by using distinct model or authentication model. Here human to human verification (H2HV) is defined by proposed CCDA (capture, checking, decision, and action) processing model. Different types of U-Strokes and their multiple features are discussed here along with collected data through Android device, which may attract researchers to work on this domain in future.

The rest of the paper is organized as follows. Section 2 presents a brief survey of existing works on this domain. Section 3 describes proposed U-Stroke analysis to verify end-user identity through ubiquitous devices. Section 4 presents brief analysis part followed by conclusion in section 5.

2 Literature Survey

Researchers are very much interested to work on human computer interaction to be considered as a means of verification of end-user identity [3, 4, 5, 6, 7, 8, 9]. Nowadays, HCI interaction with ubiquitous computing device becomes popular. Instead of considering end user authentication by means of only PINs or passwords, researchers are working on this area in recent years. Related works can be considered from different aspects such as user's identity verification by keystroke dynamics, by finger movements and tapped information on sensor based touch screens. “Touch Sensor” is the predecessor [10] of modern touch screens. In a few recent studies, touch-based biometrics is proposed for mobile devices instead of keystroke dynamics. There exists several finger gestures based authentication on touch screen of mobile devices. According to literature survey, very few works are based on continuous authentication on smart devices. Software like Touch logger can detect usage pattern of device owner and block unauthorized access to the device. Biometric touch information can be considered to enhance the security by using screen unlock. Interaction data are captured by sensors without affecting normal activities. If it is detected that the current end-user is different from the device owner, explicit access policy needs to be triggered.

PIN authentication method is strengthened with sensor data and timings. Different parameters can be collected like acceleration values, touching pressure, touched area on the screen, different temporal values like key-hold time or inter-key time. Flexible authentication can be implemented without considering predefined text. After a learning phase, end - users are authenticated while entering normal text. Touch dynamics may extract features like priority of usage of left and right hand, one-hand or both hand, use of thumb or index finger, stroke size, stroke timing, stroke speed, and timing regularity. Another way of implicit authentication is through learning behavior of a user- based on sent and received text messages, phone calls, browser history, and location of the smart phone. Instead of entering text into a soft keyboard, gestures like sliding towards a special direction or taps are most efficiently used. Generally, target acquisition tasks are carried out with a stylus that is much smaller than the targets. Among various information processing model, Fitt's model is dependent on finger size and type of stroke [11]. Complexity of Welford's model [12] is high compared to Whiting's model, but Welford model is more efficient to define information processing task. A major challenge is to apply Fitts' law to finger input, which may not be efficient for small-sized targets. This is mainly due to "Fat Finger" problem. Fitts' law fails if targets are small. As finger touch on smart phones and tablets becomes popular, examining Fitts' law [11] for finger touch attracts attention of many HCI researchers. Other information processing models [12] are not utilized effectively to model touch gesture till date.

After studying various existing works on this domain, it can be said there is no suitable model exists till date, which can be applied to any device for end-user verification. Most of the touch screen based authentication techniques consider few parameters or includes the features that are only available to costly devices. Considering few strokes with few parameters may not give accurate result. However, processing too many parameters may slow down the procedure. In order to improve the efficiency and accuracy of end-user identity verification, proposed work mainly focuses on finding physical or virtual UDI parameters of individual users and build hybrid personal profiles for accurate identification of end-users through ubiquitous input devices.

3 Proposed Work

Any type of user to input device interaction is proposed as U-Stroke, which is analyzed to identify end-users to any device in ubiquitous environment. U-Stroke analysis considers typing on numerical keypad or QWERTY keypad of mobile phone, external keyboard of tablet, and physical keyboard of desktop or laptop as physical keystroke; typing on on-screen QWERTY keypad of smart devices or any interaction with touch screen as touch stroke. The concept of U-Stroke pattern is proposed to model any human interaction with ubiquitous input device (H2UID).

Definition (U-Stroke). It is designed for any type of end-user to input device interaction (UDI) on user-friendly interface of smart devices {smart phone, tablet, phablet (phone + tablet)} in ubiquitous environment to validate H2M communication. U-Stroke is implicitly used to create hybrid profile of end-user (HPEU) for identification as well as verification.

Classification of proposed U-Stroke pattern is presented in figure 1.

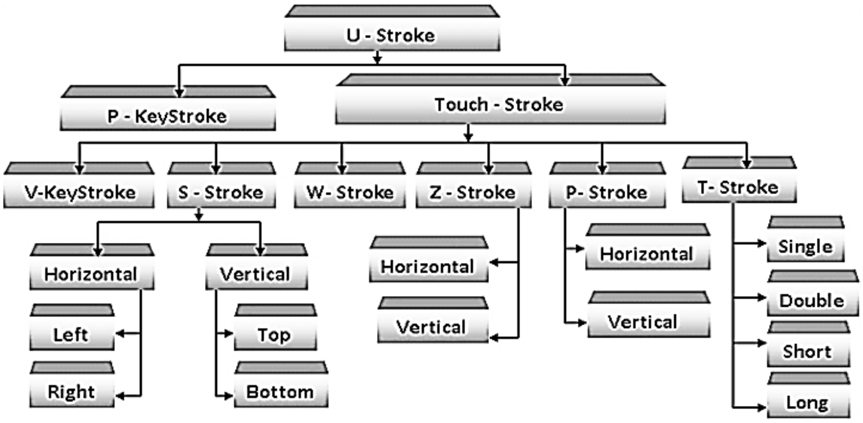

Fig. 1. U-Stroke Pattern

H2UID identity verification triggers multiple times during usage. Detailed classification of proposed U-Stroke pattern is described below.

P- KeyStroke (Physical KeyStroke)–It behaves as keystroke dynamics on physical (hard) keyboards of desktops or laptops or keypads of mobile phones. This type of keystroke mainly considers temporal (key press event, latency, typing speed etc.) data. As keypad of mobile devices may differ with physical keyboards, P-Keystroke patterns may differ.

Touch Stroke– It is based on the usage pattern on touch screen. To recognize valid user, multiple sensor data are integrated to model user variation.

V- KeyStroke (Virtual KeyStroke)–It considers typing on soft keyboard (virtual) on touch screen. Distance between neighbor keys of soft keyboard is much lesser than distance between neighbor keys of hard keyboard. V-Keystroke mainly includes touch_down, and touch_up events.

S-Stroke (Slide Stroke)– If finger movement on touch screen is either in horizontal direction (left or right) or in vertical direction (top or bottom), it is a type of S-Stroke. It is unidirectional and probability of touch_move event is high.

W-Stroke (Write Stroke) –If finger (mainly index) acts just like stylus, used normally for writing or drawing on the screen, it is a type of W-Stroke. This stroke is similar to handwriting. W-Stroke includes touch_down, touch_move and touch_up events.

Z-Stroke (Zoom Stroke)– If two fingers (mainly thumb and index of right hand) start from the same point and move towards opposite directions, it is a type of Z-Stroke. It is considered as open stroke as it moves outwards. Z-Stroke is bi-directional and probability of occurrence for touch_move event is high.

P-Stroke (Pinch Stroke) – If two fingers (mainly thumb and index of right hand) start from two opposite directions and move towards same point, it is a type of P-Stroke. It is considered as close stroke as it moves inwards on touch-screen. P-Stroke is bi-directional and probability of occurrence for touch_move event is high.

T-Stroke (Tap stroke) –If touch_down, and touch_up events occur due to the stroke similar to single click, double lick, long tap or short tap on touch screen and screen unlock, it is a type of T-Stroke. Probability of occurrence for touch_down event is high.

3.1 U-Stroke Pattern Modeling through Ubiquitous Input Device (UID)

This section presents detailed idea about how U-Stroke pattern processing model can be defined in terms of information processing model like Welford's model [12]. Here proposed U-Stroke Pattern Processing Model is defined by CCDA (capture, checking, decision, and action) concept with self-loop to identify end-user. This CCDA concept is mapped to the identity verification of end-user. Capture process is mainly used to collect U-Stroke data. Checking process is mainly used to classify U-Stroke pattern. Decision process is used mainly to take decision. Action process is used to take necessary steps according to final decision. Short term memory is considered as local memory store and long term memory is considered as remote memory store. Self-loop (feedback) is used to update template profile. Figure 2 represents CCDA processing model for UID.

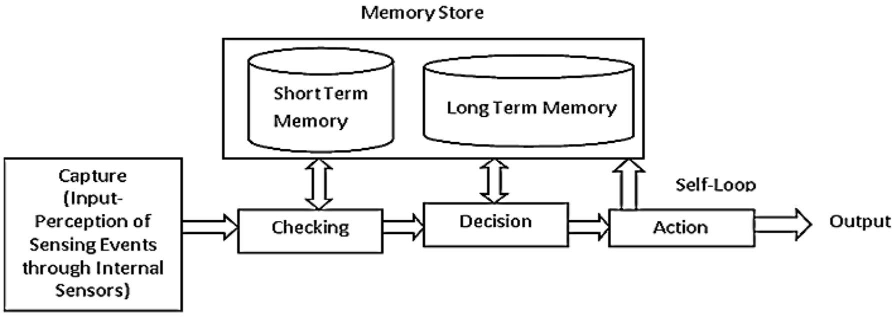

Fig. 2. CCDA Processing Model for UID

CCDA processing model of UID enables effective modeling of end-user's usage pattern from U-Stroke and creation of hybrid profile of end-user after extracting multiple features. Considering more than one feature can enhance accuracy level of classification. End-User's usage pattern is monitored multiple times to avoid malicious use. Less complex computation such as checking only device owner's validity can be performed through short term memory store (local). Short term memory can store device owner's log file. However, other complex computations are performed through long term memory store (remote server) to reduce computational overhead from limited resource device. CCDA process model may slow down if multiple features are processed compositely. For this reason, composite features are separated before processing and then fused as a whole to speed up the process. Significant variation of user's profile is managed by frequent update of template through feedback path of CCDA processing model.

3.2 End-User Identity Verification by U-Stroke Pattern Processing Model

In this section, identity verification procedure of end-users through ubiquitous input devices is briefly presented. Definition of end-user identity verification is given below.

Definition (n class H2HV). A type of H2UID interaction based on U-Stroke pattern processing model CCDA, is classified into two sub-classes- distinction class (1: m verification, where n = m) and authentication class (1:2 verification, where n = 2).

In 1: m distinction logic (where m = number of enrolled end-users), identity of valid end-user is accurately determined among all other end-users. In 1:2 authentication logic (1st class →valid user, 2nd class→ invalid user), identity of claimed owner of the device is determined. U-Stroke can produce raw events at every few milliseconds. One single operation generates a series of raw events to create secure hybrid profile of end-user (HPEU).

1:2 authentication logic can be processed on ubiquitous input device, depending on its capacity. Only owner's usage patterns are stored in short term memory of CCDA for verification. 1: n distinction logic is processed on web server, where usage patterns of n number of enrolled users are stored in long term memory of CCDA for verification. H2HV works in two phases- initial phase and verification phase.

Initial phase includes U-Stroke data acquisition task, multiple features extraction and processing, learning of data-set from environment and template generation. Verification phase includes U-Stroke data collection for claimed identity, multiple features extraction and processing, classification, fusion, match logic, decision logic. Capture process of CCDA model includes data acquisition task, multiple features extraction and processing, learning of data-set from environment and template generation of H2HV. Checking process of CCDA model includes classification, fusion, match logic. Decision process of CCDA includes decision logic and finally action needs to be taken according to security rule. Here main focus is given to data collection from touch screen enabled smart phones or tablets. However, U-Stroke pattern can be collected from any type of devices from traditional desktop computer to smart phone.

Multiple Features Collected for Capture Process of CCDA Model.
At first, H2UID interaction captures U-Stroke data according to device usage. Event e in U-Stroke consists of multiple parameters. Capture process of CCDA processing model includes raw data collection. U-Stroke raw data includes {activity, timestamp,

X-coordinate, Y-coordinate, pressure, area covered}. Figure 3 represents U-Stroke analysis framework to verify identity of end-user in ubiquitous environment.

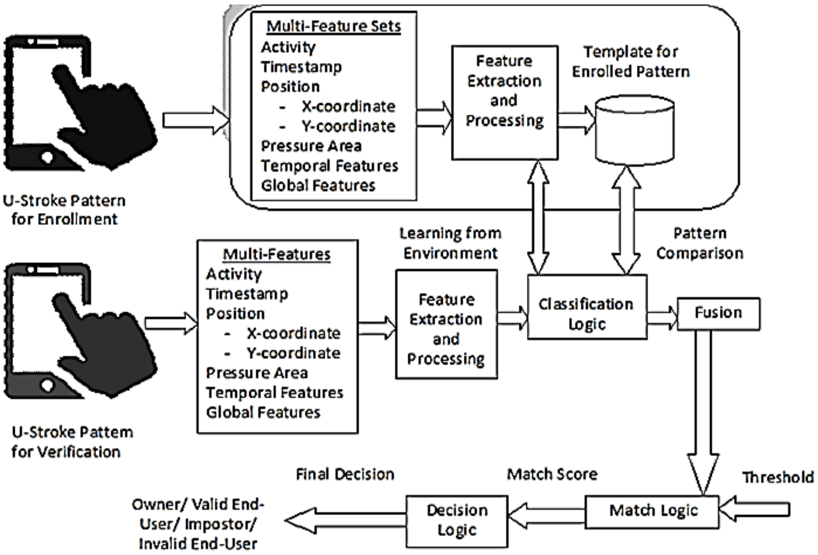

Fig. 3. U-Stroke Analysis for End-User Identity Verification (H2HV)

Features of U-Stroke pattern are presented below.

activity- P- KeyStroke, Touch-Stroke, V-KeyStroke, S-Stroke, W-Stroke, Z-Stroke, P-Stroke, T-stroke - all are under activity category. Three types of activity are touch_down (0), touch_move (0.5) and touch_up (1).

timestamp - Absolute time of recorded action is measured in milliseconds. Timestamp depends on clock resolution of used devices [13].It is measured by Android API Date.getTime()[14]. Device clock is characterized by sampling rate of 1ms.

position (X-coordinate, Y-coordinate) - On screen X-axis and Y-axis values of location of each touch point are recorded. When end-user touches the screen, one finger_touch point (part of fingertip) is created. It is measured by Android API ImageView.getLocationOnScreen()[14].

pressure- Finger pressure ranges from 0 to 1. 0 implies no pressure and 1 implies normal pressure on the screen. However, few touch points hold the same value of finger pressure. Finger pressure force can be obtained by multiplying pressure and size of each touch point. It is measured by Android API MotionEvent.getPressure()[14]. To enhance usability of finger pressure data, values of pressure are expanded 1000 times.

covered area- Part of fingertip touched on the screen is considered as size of finger_touch point. It is measured by Android API MotionEvent.getSize().The area covered by fingertip varies with the size of the finger or type of the finger.

Mapping of H2HV procedure with CCDA model is presented in table 1.

Table 1. Mapping of H2HV Procedure with CCDA Model

```
begin
   check H2HV phase
   if H2HV phase:= initial then,
      enable CCDA(capture)
   else if H2HV phase:= verification then,
      enable  CCDA (capture, checking, decision, action)
    if n=m in H2HV then, // n $\leftarrow$ number of end-users needs to be
                                  checked for H2HV and m $\leftarrow$ number of enrolled user
       activate 1:m distinction model
   else  if n=2 in H2HV then,
       activate 1:2 authentication model
end
```

Table 2 represents procedural logic of CCDA model to identify end-users in ubiquitous environment through any device.

Table 2. Procedural Logic of CCDA for H2HV

```
begin
 process capture()
   begin
     collect_UStroke(P-KeyStroke, T-KeyStroke)         // data collection
     aggregate X (activity, time, position$_x$, position$_y$, pressure, area) //  data of
                                                                        raw event
      fragment X into {$X_1$, $X_2$, ...,$X_n$} based on time // feature extraction and
                                                          processing
        construct feature vector F
         fragment F into {$f_1$, $f_2$, ...,$f_n$}
        map X {$X_1$, $X_2$, ...,$X_n$} to F {$f_1$, $f_2$, ...,$f_n$}
        learn_context ( CCDA self loop)
        find frequency (used pattern)
       store_template(F)
       update(CCDA feedback)
      if template_size<= local_storge then,
          encode and  store (short_term_memory)
      else
        encode, transmitted through secure protocol  and store (long_term_memory)
    end
   process checking()
   begin
    calculate_closeness (U, F) // U $\rightarrow$ enrolled user's feature vector during veryfi-
                                     cation, F $\rightarrow$ feature vector stored at template
    evaluate similarity_score() and assign to to s
    fusion()
```

```
  begin
  assign w to each factor of U-Stroke  / /w→weight
  calculate confidence_level(w,s)and assign to c
  end
  evaluate match score(s, c, th) and assign to rank   // th→tolerance level
 end
 process decision()
  begin
    if rank=1 then,
      set validity = true   // valid end-user
     assign valid identity of end-user
   else
     set validity = false // invalid end-user
  end
  process action()
   begin
    generate_warning() or block_user() // based on security context
   end
```

4 Analysis

Overall data collection task is performed here by Spice Dual Core Phablet (screen size 5 inches or 12.7 cm), having easy to use touch based user interface with Android version 4.2 Jelly Bean. Here Android data acquisition device id is 37d4d7ddd0b99bdf. As U-Stroke contains heterogeneous strokes, multiple features are not processed as a whole; because it may consume more energy in low-cost devices. As for example, S-Stroke or W-Stroke with one finger move or a T- Stroke with one finger hit is quite different from P-Stroke or Z-Stroke with two fingers. It is seen from the collected raw data that there exists relation between two sets. Set 1 includes {activity, position (X-coordinate, Y-coordinate), timestamp} and set 2 includes {pressure, area}. Set 1 is considered as fine-grained set, whereas set 2 is considered as coarse-grained set. Coarse grained set member “area” is proportional to finger-tip size of end-user. Fingertip size of thumb is greater than index. Frequency of use of thumb and index fingers are higher compared to other fingers. As multiple features are considered, FTA (failure to acquire) and FTE (failure to enroll) rate for H2UID interaction are almost equal to zero compared to other behavioral usage pattern analysis. However, probability of error for device usage (like V-Keystroke, W-Stroke, T-Stroke) is directly proportional to the size of fingertip. Fat finger also affects FTE and FTA parameters. Table 3 represents sample of U-Stroke raw data. Table 4 represents few strokes of U-Stroke pattern for user U1. To build strong profile of user U1 all raw data are collected such as T-Stroke, S-Stroke, and then are grouped according to type of stroke and are processed separately. Finally outputs of all features are fused to generate final decision.

Table 3. U-Stroke Raw Data Collected from Two End-Users

User-ID	Fine Grained Features				Coarse Grained Features	
	Activity	Timestamp (ms)	X-coordinate	Y-coordinate	Pressure	Area
U1	0	4893336544	272	269	0.21	0.04444445
U1	0.5	4893336790	262	271	0.32	0.04444445
U1	0.5	4893336795	123	327	0.28	0.04444445
U1	0.5	4893336952	98	336	0.17	0.13333336
U1	0	4893337798	108	339	0.48	0.04444445
U2	0	4626934695	138	242	0.71	0.15555558
U2	0.5	4626934713	140	241	0.71	0.1777778
U2	0.5	4626934727	180	225	0.71	0.20000002
U2	0.5	4626934744	230	220	0.71	0.1777778
U2	1	4626934790	293	216	0.55	0.13333336

Table 4. Few Strokes of U-Stroke Pattern for User U1

Stroke	Timestamp(ms)	X	Y	Pressure	Area
T-Stroke	78864458	73	541	0.1	0.070588
S-Stroke (right)	79073654	125	656	0.133333	0.109804
S-Stroke (left)	79137343	433	595	0.133333	0.129412
S-Stroke(up)	79107024	392	600	0.141176	0.133333
S-Stroke (down)	79168990	405	351	0.2	0.160784
W-Stroke	79203117	244	271	0.233333	0.156863
Z-stroke	59021345	240	481	0.266667	0.133333
P-Stroke	37581872	394	55	0.137255	0.2

Compared to existing works [15, 16], proposed model works well by collecting heterogeneous data, reduces resource consumption by using CCDA model, and works well for low-cost devices for considering only basic sensor features. Here only 1:2 authentication model is considered, where user U1 is treated as valid end-user (owner) of the device, whereas user U2 is treated as invalid one. Figure 4 represents end-user identification through coarse-grained features. It is seen that inter-user variability between user U1 and U2 is much higher than intra-user variability that can identify owner U1 of Spice Dual Core Phablet efficiently.

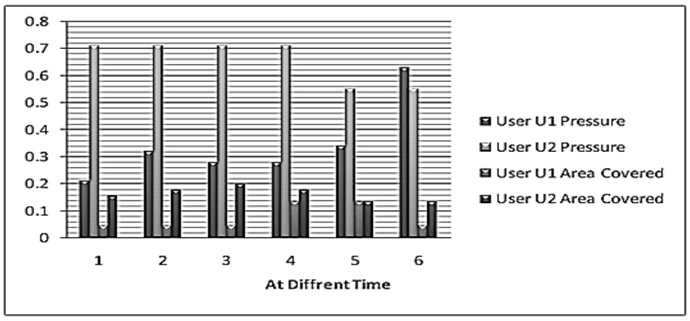

Fig. 4. Coarse-grained Features for End-User Identification H2HV

Remote healthcare application is considered as a use case [2] of our proposed work. In this case, data owner (patients or local caregivers) can use smart device to upload health related data towards remote web server or cloud server. Similarly healthcare professionals from remote locations can access health related data of patients through smart devices at any time. Here authentication through ubiquitous input device is most important step to validate remote health service. Our proposed model works on this direction.

5 Conclusion

Existing works focus on either traditional keystroke logic or touch-stroke logic. We are designing a system model, which is capable to identify user at any place. This paper determines various patterns to build strong hybrid profile of end-users that is capable to reduce false detection, which is very harmful in uncontrolled environment. For touch stroke based authentication, multiple features are collected through user-friendly interface of Android device. This is useful to enhance the flexibility of end-user verification. Multiple features are processed separately and then fused to build user profile. Requirement of computational power is minimized by proposing CCDA model running on the background of limited resource mobile device. Proposed logic can process multiple features in a way that end-user can be identified accurately with less time and computational overhead. User profiles are frequently updated by self-loop. Procedural logic of proposed CCDA model is given in paper and result shows that owner of Spice Dual Core Phablet can be successfully identified through coarse grained features (along with fine grained features consideration) with high accuracy and less resource consumption. Detailed analysis is not given here due to shortage of space. Target of our proposed work is to apply end-user verification (H2HV) on cheap smart devices that can be used by any people within budget.

Work is on to evaluate the performance of proposed model compared to other well-known works on this domain. Various security features, scalability, processor and memory overhead, battery consumption, and timeliness are also need to be considered to implement U-Stroke analysis in ubiquitous environment. Hopefully, direction of this research will be valuable for the further research on this domain.

Acknowledgement

This research work was partially supported by University of Calcutta grant no. UGC/452/DST Purse and Bialystok University of Technology grant no. S/WI/1/2013.

References

1. Ballagas, R., Borchers, J., Rohs, M., Sheridan, J.G. :The smart phone: a ubiquitous input device. IEEE Journal of Pervasive Computing, vol. 5, no. 1, pp. 70-77 (2006).
2. Bhattasali, T., Saeed, K., Chaki, N., Chaki, R.: Bio-authentication for layered remote health monitor framework. Journal of Medical Informatics & Technologies, vol. 23, pp. 131-140 (2014).
3. Maxion, R., Killourhy, K.: Keystroke biometrics with number-pad input. In: Proceedings of IEEE International Conference on Dependable Systems & Networks, pp.201-210(2010).
4. Xu, H., Zhou, Y., Lyu, M., R.: Towards Continuous and Passive Authentication via Touch Biometrics: An Experimental Study on Smartphones. In: Proceedings of Symposium on Usable Privacy and Security, pp. 187-198 (2014).
5. Shahzad, M., Liu, A., X., Samuel A.: Secure Unlocking of Mobile Touch Screen Devices by Simple Gestures – You can see it but you cannot do it. In: Proceedings of International Conference on Mobile Computing and Networking, pp. 39-50 (2013).
6. Feng, T., Liu,Z., Kwon, K-A., Shi, W., Carbunar,B., Jiang, Y., Nguyen, N. : Continuous mobile authentication using touchscreen gestures. In: Proceedings of IEEE International Conference on Biometrics: Theory, Applications and Systems, pp. 451-456 (2013).
7. Xu, Z., Bai, K., Zhu, S.:TapLogger: Inferring User Inputs On SmartphoneTouchscreens Using On-board Motion Sensors. In: Proceedings of International conference on Security and Privacy in Wireless and Mobile Networks, pp. 113-124. ACM (2012).
8. Frank, M., Biedert, R., Ma, E., Martinovic, I., Song, D.: Touchalytics: On the Applicability of Touchscreen Input as a Behavioral Biometric for Continuous Authentication. IEEE Transactions on Information Forensics and Security, vol. 8. no.1, pp. 136-148 (2013).
9. Luca, A.D., Hang, A., Brudy, F., Lindner, C., Hussmann, H.: Touch me once and I know it's you! Implicit Authentication based on Touch Screen Patterns. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 987-996. ACM (2012).
10. Touch Screen. http://inventors.about.com/library/inventors/bltouch.htm.
11. Bi, X., Li, Y., Zhai, S.: FFitts law: Modeling Finger Touch with Fitts' Law. In: Proceedings of SIGCHI Conference on Human Factors in Computing Systems, pp. 1363-1372. ACM(2013).
12. Welford, A. T., Norris, A. H., Shock, N. W.: Speed and Accuracy of Movement And Their Changes with Age. Elsevier Journal Acta Psychologica. vol. 30, pp. 3–15 (1969).
13. Killourhy, K., Maxion R.: The Effect of Clock Resolution on Keystroke Dynamics. In: Proceedings of International Symposium of Recent Advances in Intrusion Detection. Lecture Notes in Computer Science (LNCS), vol. 5230, pp. 331-350. Springer(2008).
14. Rogers, R.: Android Application Development. O'Reilly(2009).
15. Zheng, N., Baiy, K., Huangy H., Wang, H.: You are how you touch: User verification on smartphones via tapping behaviors. In: Proceedings of IEEE International Conference on Network Protocols (ICNP), pp. 221-232 (2014) .
16. Bo, C., Zhang, L., Li, X..Y., Huang, Q., Wang, Y.: SilentSense: Silent user identification via touch and movement behavioral biometrics. In Proceedings of Annual International Conference on Mobile Computing & Networking (MobiCom), pp. 187-190 (2013).