High Accessible Virtual Keyboards for Preventing Key-Logging

Wooguil Pak
Department of Computer Engineering
Keimyung University
Daegu, Korea
wooguilpak@kmu.ac.kr

Youngrok Cha
Department of Computer Engineering
Keimyung University
Daegu, Korea
zeststyle@kmu.ac.kr

Sunki Yeo
Department of Computer Engineering
Keimyung University
Daegu, Korea
soonsusky@kmu.ac.kr

***Abstract*— Today, most of smart-phones adopt virtual keyboards as input devices for users. However, there are serious secure threats exit such as key-logger attack that leaks key input values of virtual keyboards. Since the smart-phone provides limited virtual keyboards due to its small screen size, it is more vulnerable for such attacks compared to personal computers (PCs). In this paper, we propose a solution for this problem. Our approach has a unique feature to improve security of the keyboard while maintaining high accessibility against existing solutions. Moreover, we can also apply it to various mobile devices with small screens due to its flexibility in terms of the size.**

***Keywords*— Virtual keyboard, key-logger, high accessibility, mobile devices.**

I. INTRODUCTION

We use various mobile devices such as smart-phones in daily life owing to their convenience. According to mobile trend report, about 83% Korean adults use smart-phones [1]. It means that smartphones are the most common IT devices even compared to personal computers (PCs).

The main difference between smart-phones and prior mobile-phones is that development environments of smart-phones are open and free to everyone. Therefore, anyone can make his own program and distribute it through global app stores. Due to these features, the smart-phone becomes the most successful device.

However, open development environments result in serious security problems. For criminals, it is also easy to develop malicious applications and expose them to any user in any country without any cost. Therefore, the number of detecting malicious code increases very fast. For example, it increases by 14.2% and 444% for 2013 and 2014 respectively compared to 2012 [2].

Today, key logging is one of well-known security threats for smart-phones [3][4]. Since smart-phones adopt small screens for high mobility, it is difficult to embed physical keyboards into the smart-phone body. Thereby, a virtual keyboard is very common. Although it is not a real keyboard, it is very similar to the physical keyboard in terms of layout. Due to its similarity, we can precisely estimate the input key value if the malicious app, called 'logger', obtains coordinate data for touch points. Therefore, private data such as passwords and account numbers can be leaked while users cannot notice them.

In this paper, we will propose a new approach to resolve this problem. Although there have been some researches to improve security for virtual keyboards, they just improve the security by sacrificing user accessibility. However, our approach can obtain high security and accessibility simultaneously [5].

II. RELATED WORK

A. QWERTY or ABC Layout

QWERTY and ABC layouts are the most common for initial virtual keyboards. Since users are already familiar with the layouts, they can use the keyboards without practice. However, the fixed key arrangement of the layouts are vulnerable for key-logging attacks. It is possible to estimate input key values exactly from coordinates of touch points if available. Fig. 1 shows QWERTY layout.

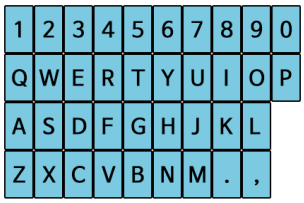

Figure 1. QWERTY layout

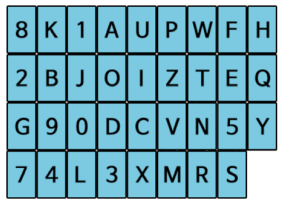

Figure 2. Random layout

B. Random Layout

Random layout was proposed to resolve the vulnerability of ABC or QWERTY layouts as shown in Fig. 2. Whenever the virtual keyboard is displayed, the key arrangement are determined randomly, so it is almost impossible to guess input keys from only coordinates of touch points. It is known to be safest but users cannot be accustomed to the layout due to its randomness, so it always takes long time to input even short messages.

C. QWERTY with Random Space Layout

It overall follows QWERTY but random horizontal spaces are inserted between some keys to randomize key locations in Fig. 3. Since the familiar key layout, user can easily find a key to be typed. Compared to the original QWERTY keyboard, it is safer against key-logging without serious sacrificing accessibility. Therefore, most of banks and governments web sites or applications adopt this virtual keyboard layout.

However, it is recently known that it is possible to find the touched key exactly from multiple coordinate data of touch points for the same key [5].

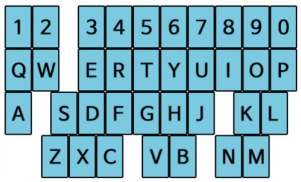

Figure 3. QWERTY with random space

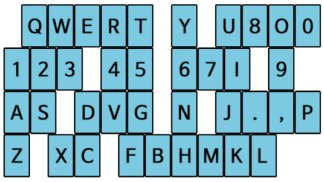

Figure 4. The proposed layout

III. PROPOSED VIRTUAL KEYBOARD LAYOUT

A. Overall Characteristics

Virtual keyboards should satisfy three requirements as following simultaneously.

- High accessibility: the virtual keyboard should adopt well-known layout to allow users for easy adoption and fast typing. It is very important for users to use the keyboard without any prior practice.
- High security: the virtual keyboard should have random layouts whenever displayed to avoid estimating typed keys and leaking private data.
- Flexible size: Since most of virtual keyboards are required for mobile devices such as smart-phones, it should support the small screen size of smart-phones as well as normal PC monitor screens. At least, the size of a new virtual keyboard should not larger than that of existing keyboards.

It is very difficult to develop new virtual keyboard to satisfy three constraints simultaneously. Most of all, accessibility and security generally have trade-off relationship. To resolve the problem, we design our new virtual keyboard as following.

First, we adopt a well-known layout such as QWERTY or ABC to support high accessibility. Second, for improving security of the virtual keyboard, we also adopt the random space technique. An original random space layout uses only horizontal random spaces and incurs vulnerability for key-logger, so we extend this layout for vertical randomness. However, it is difficult to use vertical random spaces since it results in larger keyboard sizes, therefore failing to satisfy the third constraint.

Our solution is exchanging two keys vertically adjacent each other. If we adopt this approach, the size of locatable area for each key increases by three times compared to existing layout. Therefore, it can achieve very high security. One of merits of this solution is that it adds vertical randomness without increasing a keyboard size, so we can satisfy three constraints simultaneously.

Fig. 4 shows a new keyboard layout generated by our proposed approach. It is similar to 'QWERTY with random space layout' but some keys are moved to upper or lower row.

Using existing random layout, a user should search entire keyboard at worst case to find each key, resulting in very slow typing. However, with our layout a user can find each key near the location where the user expect to find the key, so it can improve the typing speed significantly.

The proposed layout provides high security while keeping accessibility high. Above all, it requires the same size of existing layout, therefore, making easy to migrate to our layout without the high cost.

B. Implementation

We should decide each key location with unpredictable randomness to make an entire keyboard layout. For higher security, we use a hash function based random number generator called Hash_DRBG [6]. The random number generator is internally used two times; one for deciding which key is moved upper or lower row, the other for choosing the size and location of random space. The overall algorithm to build virtual keyboard layouts is described in Algorithm 1.

Algorithm 1. Generate the proposed virtual keyboard layout

```
Set n to the total number of keys.
WHILE (--n > 0)
    Randomly select one key which has not been selected yet.
    Randomly select one of actions; move up or down, and don't move.
    Apply the action to selected key.
END_WHILE
FOR i=1 to max row
    Select ith row of keyboard.
    WHILE (total length of ith row < max keyboard width)
        Select location and size of random space.
        Add the space to ith row.
    END_WHILE
END_FOR
```

IV. PERFORMANCE EVALUATION

A. Numerical Analysis

We can calculate the estimation probability of each key [5]. Tables I and II show the results for original and proposed QWERTY with random space layouts. As shown in Table II, the proposed layout decreases the probability at least by half. Therefore we can confirm that it can effective against key-logger attack.

B. Experimental Analysis

We measured the total time to type each given message completely for 10 experimenters. We chose 4 messages which have different lengths. Fig. 5 shows the typing time according to the length of messages. As shown in Fig. 5, our layout shows 50% longer time compared to QWERTY with random space but 70% smaller time compared to random layout.

TABLE I. ESTIMATING PROBABILITY OF QWERTY WITH RANDOM SPACE

Row	1		2		3		4		5		6		7		8		9		10		11	
1	1	100.0%	1	10.0%	2	20.0%	3	30.0%	4	40.0%	5	50.0%	6	60.0%	7	70.0%	8	80.0%	9	90.0%	0	100.0%
			2	90.0%	3	80.0%	4	70.0%	5	60.0%	6	50.0%	7	40.0%	8	30.0%	9	20.0%	0	10.0%		
2	q	100.0%	q	10.0%	w	20.0%	e	30.0%	r	40.0%	t	50.0%	y	60.0%	u	70.0%	i	80.0%	o	90.0%	p	100.0%
			w	90.0%	e	80.0%	r	70.0%	t	60.0%	y	50.0%	u	40.0%	i	30.0%	o	20.0%	p	10.0%		
3	a	100.0%	a	20.0%	a	2.2%	s	6.6%	d	13.3%	f	22.2%	j	13.3%	k	6.6%	l	2.2%	l	20.0%	l	100.0%
			s	80.0%	s	35.6%	d	46.7%	f	53.4%	g	55.6%	h	53.4%	j	46.7%	k	35.6%	k	80.0%		
					d	62.2%	f	46.7%	g	33.3%	h	22.2%	g	33.3%	h	46.7%	j	62.2%				
4	z	100.0%	z	14.3%	x	28.6%	c	42.9%	v	57.1%	b	71.4%	n	85.7%	m	100.0%						
			x	85.7%	c	71.4%	v	57.1%	b	42.9%	n	28.6%	m	14.3%								

TABLE II. ESTIMATING PROBABILITY OF THE PROPOSED LAYOUT

Row	1		2		3		4		5		6		7		8		9		10		11	
1	1	50.0%	1	5.0%	2	10.0%	3	15.0%	4	20.0%	5	25.0%	6	30.0%	7	35.0%	8	40.0%	9	45.0%	0	50.0%
	q	50.0%	2	45.0%	3	40.0%	4	35.0%	5	30.0%	6	25.0%	7	20.0%	8	15.0%	9	10.0%	0	5.0%	p	50.0%
			q	5.0%	w	10.0%	e	15.0%	r	20.0%	t	25.0%	y	30.0%	u	35.0%	i	40.0%	o	45.0%		
			w	45.0%	e	40.0%	r	35.0%	t	30.0%	y	25.0%	u	20.0%	i	15.0%	o	10.0%	p	5.0%		
2	1	33.3%	1	3.3%	2	6.7%	3	10.0%	4	13.3%	5	16.7%	6	20.0%	7	23.3%	8	26.7%	9	30.0%	0	33.3%
	q	33.3%	2	30.0%	3	26.7%	4	23.3%	5	20.0%	6	16.7%	7	13.3%	8	10.0%	9	6.7%	0	3.3%	p	33.3%
	a	33.3%	q	3.3%	w	6.7%	e	10.0%	r	13.3%	t	16.7%	y	20.0%	u	23.3%	i	26.7%	o	30.0%	l	0.0%
			w	30.0%	e	26.7%	r	23.3%	t	20.0%	y	16.7%	u	13.3%	i	10.0%	o	6.7%	p	3.3%		
			a	2.2%	a	0.2%	s	0.7%	d	1.5%	f	2.5%	j	1.5%	k	0.7%	l	0.4%	l	3.3%		
			s	8.9%	s	4.0%	d	5.2%	f	5.9%	g	6.2%	h	5.9%	j	5.2%	k	5.9%	k	13.3%		
					d	6.9%	f	5.2%	g	3.7%	h	2.5%	g	3.7%	h	5.2%	j	10.4%				
3	q	33.3%	q	3.3%	w	6.7%	e	10.0%	r	13.3%	t	16.7%	y	20.0%	u	23.3%	i	40.0%	o	45.0%	p	50.0%
	a	33.3%	w	30.0%	e	26.7%	r	23.3%	t	20.0%	y	16.7%	u	13.3%	i	10.0%	o	10.0%	p	5.0%	l	50.0%
	z	33.3%	a	6.7%	a	0.7%	s	2.2%	d	4.4%	f	7.4%	j	4.4%	k	2.2%	l	1.1%	l	10.0%		
			s	26.7%	s	11.9%	d	15.6%	f	17.8%	g	18.5%	h	17.8%	j	15.6%	k	17.8%	k	40.0%		
			z	4.8%	d	20.7%	f	15.6%	g	11.1%	h	7.4%	g	11.1%	h	15.6%	j	31.1%				
			x	28.6%	x	9.5%	c	14.3%	v	19.0%	b	23.8%	n	28.6%	m	33.3%						
					c	23.8%	v	19.0%	b	14.3%	n	9.5%	m	4.8%								
4	a	50.0%	a	10.0%	a	1.1%	s	3.3%	d	6.7%	f	11.1%	j	6.7%	k	3.3%						
	z	50.0%	s	40.0%	s	17.8%	d	23.4%	f	26.7%	g	27.8%	h	26.7%	j	23.4%						
			z	7.2%	d	31.1%	f	23.4%	g	16.7%	h	11.1%	g	16.7%	h	23.4%						
			x	42.9%	x	14.3%	c	21.5%	v	28.6%	b	35.7%	n	42.9%	m	50.0%						
					c	35.7%	v	28.6%	b	21.5%	n	14.3%	m	7.2%								

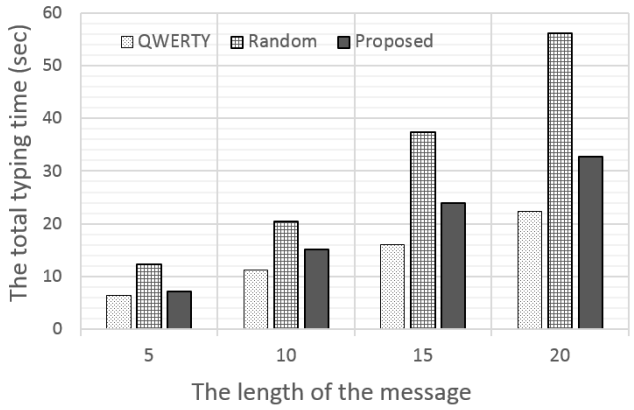

Figure 5. The total typing time according to the length of the message

V. CONCLUSION

The proposed layout for virtual keyboards adds randomly exchanging of vertically adjacent keys to existing QWERTY with random space for 2 dimensional randomness. Therefore, it can provide high accessibility and high security simultaneously.

ACKNOWLEDGMENT

This work (Grants No. C0333038) was supported by Business for Cooperative R&D between Industry, Academy, and Research Institute funded Korea Small and Medium Business Administration in 2015.

REFERENCES

[1] Digieco mobile trend report, http://www.digieco.co.kr.

[2] Ahnlab, http://www.ahnlab.com.

[3] Daily secu, http://www.dailysecu.com.

[4] Roland M., Langer J. and Scharinger J., "Practical Attack Scenarios on Secure Element-Enabled Mobile Devices," 2012 4th International Workshop on Near Field Communication, pp. 19-24, March 2012.

[5] Yunho Lee, "An Analysis on the Vulnerability of Secure Keypads for Mobile Devices," Journal of Korean Society for Internet Information, vol.14, no.3, pp. 15-21, June 2013.

[6] Recommendation for Random Number Generation Using Deterministic Random Bit Generators, Nist SP 800-90A, http://www.nist.gov.