Blind Recognition of Touched Keys on Mobile Devices

Qinggang Yue
University of Massachusetts Lowell, USA
qye@cs.uml.edu

Zhen Ling
Southeast University, China
zhenling@seu.edu.cn

Xinwen Fu
University of Massachusetts Lowell, USA
xinwenfu@cs.uml.edu

Benyuan Liu
University of Massachusetts Lowell, USA
bliu@cs.uml.edu

Kui Ren
University at Buffalo, USA
kuiren@buffalo.edu

Wei Zhao
University of Macau, China
weizhao@umac.mo

ABSTRACT

In this paper, we introduce a novel computer vision based attack that automatically discloses inputs on a touch-enabled device while the attacker cannot see any text or popup in a video of the victim tapping on the touch screen. We carefully analyze the shadow formation around the fingertip, apply the optical flow, deformable part-based model (DPM), k-means clustering and other computer vision techniques to automatically locate the touched points. Planar homography is then applied to map the estimated touched points to a reference image of software keyboard keys. Recognition of passwords is extremely challenging given that no language model can be applied to correct estimated touched keys. Our threat model is that a webcam, smartphone or Google Glass is used for stealthy attack in scenarios such as conferences and similar gathering places. We address both cases of tapping with one finger and tapping with multiple fingers and two hands. Extensive experiments were performed to demonstrate the impact of this attack. The per-character (or per-digit) success rate is over 97% while the success rate of recognizing 4-character passcodes is more than 90%. Our work is the first to automatically and blindly recognize random passwords (or passcodes) typed on the touch screen of mobile devices with a very high success rate.

Categories and Subject Descriptors

K.4 [COMPUTERS AND SOCIETY]: Public Policy Issues—*Privacy*

General Terms

Human Factors; Security

Keywords

Computer Vision Attack; Mobile Devices; Privacy Enhancing Keyboard

1. INTRODUCTION

Touch-enabled devices are ubiquitously used in our daily life. However, they are also attracting attention from attackers. In addition to hundreds of thousands of malwares [19], one class of threats against mobile devices are computer vision based attacks. We can classify those attacks into three groups: the first group of attacks directly identify text on screen or its reflections on objects [2, 1]. The second group of attacks detect visible features of the keys such as light diffusion surrounding pressed keys [3] and popups of pressed keys [28, 33]. The third group of attacks are able to blindly recognize the text input on mobile devices while text or popups are not visible to the attacker [43].

In this paper, we introduce a novel attack blindly recognizing inputs on touch-enabled devices by estimating touched points from a video of a victim tapping on the touch screen, as shown in Figure 1. In the attack, the deformable part-based model (DPM) is used to detect and track the target device and the optical flow algorithm is used to automatically identify touching frames in which a finger touches the screen surface. We use intersections of detected edges of the touch screen to derive the homography relation mapping the touch screen surface in video frames to the reference image of the software keyboard, as shown in Figure 2. DPM and other computer vision techniques are applied to automatically estimate a tiny touching area. We carefully derive the theory of the shadow formation around the fingertip and use the k-means clustering algorithm to identify touched points in the tiny touching area. Homography can then map these touched points to the software keyboard keys in the reference image. We perform extensive experiments. The target devices include the iPad, Nexus 7 and iPhone 5. Both logon keyboard and QWERTY keyboard are examined. The cameras include webcam, phone camera and Google Glass. The cameras are positioned from different distances and angles. We are able to achieve a per-key success rate of over 97% and a success rate of more than 90% recognizing 4-digit passcodes in various scenarios.

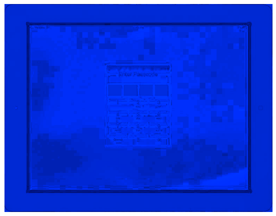
We also show that DPM can be used to recall estimated touched point, which can be mapped to the reference image in order to derive the touched key. This method of direct use of DPM for recognizing touched keys is called the baseline method. However, the baseline method achieves success rate of around 20% since DPM cannot accurately locate touched points.

To the best of our knowledge, we are the first to be able to reliably and blindly recognize passwords (or passcodes) typed on the touch screen of mobile devices of various brands. Since passwords are random and do not contain any meaningful text or pattern, natural language processing techniques cannot be used. This challenges the design of automatic recognition of the password. Our recognition system incorporates recent advancement of object detection techniques and our own analytical model of the touching process, and is able to achieve a very high success rate. We have also extended our work to the scenario of touching with both hands and multiple

CCS'14, November 3–7, 2014, Scottsdale, Arizona, USA.
Copyright 2014 ACM 978-1-4503-2957-6/14/11 ...$15.00.
http://dx.doi.org/10.1145/2660267.2660288.

ingers n re le to recogn e the touch ng nger from 10 n-gers n ch eve h gh success r te of ore th n 95% recogn ng touche es.

F gure 1: Touch ng Fr e



F gure 2 S ft e or

To e t n co puter v s on se tt c s nclu ng the one n th s p per, e es gne n ple ente s ple conte t re r n o e soft re e or for An ro , enote s Pr v c En-h nc ng e o r (PE). PE uto t c ll sho s convent on-l QWERTY e o r for the nor l te t nput n pops up r n o e e o r for nputt ng sens t ve nfor t on such s p ssco es. The rst PE protot pe s e onstr te t n ACM or shop [ ] n Octo er, 2012. To the est of our no le ge, PE s the rst gener c soft re e o r for o le pl for h le s l r pp CodeScrambler for S [23] ppe re n August 2013. PE s full e ge soft re e o r h le Co e cr - ler s es gne onl for the unloc screen n oes not prov e the conte t- re funct on l t . We refer to the ppen for the ple ent t on n ev lu t on of PE.

The rest of the p per s org n e s follo s. e ntro uce the ho ogr ph n DPM n ect on 2. ect on 3 ntro uces the tt c for the c se of t pp ng th one nger. In ect on , e scuss ho to recogn e touche po nts from touch ng fr es. E per ent es gn n ev lu t ons re g ven n ect on 5. e e ten the tt c to t pp ng th ult ple ngers n t o h n s n ect on 6. ect on 7 scusses the rel te or . e conclu e the p per n ect on 8.

2. BACKGROUND

In th s sect on, e ntro uce t o jor co puter v s on tech-n ques e plo e n th s p per: pl n r ho ogr ph n the DPM (Def or le P rt- se o el) o ject etector.

2.1 Planar Homography

Pl n r ho ogr ph s 2D project ve tr nsfor t on th t rel tes t o ges of the s e pl n r surf ce [ ]. Assu e $p = (s, t, 1)$ s n po nt n n ge of 3D pl n r surf ce n $q = (s', t', 1)$ s the correspon ng po nt n nother ge of the s e 3D pl n r surf ce. The t o ges e t en the s e c er or f-ferent c ers. There e sts n nvert le 3×3 tr H, enote s ho ogr ph tr ,

$$q = \mathbf{H}p. \quad (1)$$

2.2 Deformable Part-based Model (DPM)

DPM [11] s the st te-of- rt o ject etector n cont ns three n co ponents: ture of st r-structure p rt se o el-s, the ef c ent tch ng processf or o ject etect on n the l tent SVM (support vector ch ne) tr n ng process. DPM u l s ul-t ple st r-structure o els for the o ject of nterest from fferent v e po nts. E ch st r-structure o el h s root o el th t ch r-cter es the o ject s hole n sever l (usu ll s) p rt o els th t ch r cter e e ch p rt of the o ject, the r nchor pos t on rel -t ve to the root n ssoc te efor t on p r eters. The o els

[1] o p per s pu l she on PE .

re represente the H stogr of Or ente Gr ents (HOG) [10] fe ture, h ch s nsens t ve to l ght ng v r t on.

To etect n o ject n n ge, DPM uses sl ng- n o pproach n c lcul tes score $f_{\beta}(x)$ for e ch poss le o ject s ple x t e ch loc t on,

$$f_{\beta}(x) = \max_{z \in Z(x)} \beta \cdot \Phi(x, z), \quad (2)$$

here z s the l tent v lues, β s vector of o el p r eters, n $\Phi(x, z)$ s the fe ture vector of x. A h gh score n c tes the loc -t on of the o ject. D n c progr ng n gener l e st nce tr nsfor s re e plo e for ef c ent tch ng.

Dur ng the tr n ng, oun ng o s use to spec f the o ject of nterest n e ch ge, h le ts p rts re unl ele . DPM tre ts these unl ele p rts s l tent (h en) v r les. It uto t c ll n s n l els these p rts, n e plo s the l tent SVM to tr n the o el. Denote tr n ng t set s $D = (< x_1, y_1 >, \ldots, < x_n, y_n >)$. x_i s the ge p tch n the correspon ng oun ng o . $y_i \in \{-1, 1\}$, n c t ng hether x_i s the o ject of nterest ($y_i = 1$) or not ($y_i = -1$). DPM tr ns β n ng the o ject ve funct on,

$$L_D(\beta) = \frac{1}{2} \parallel \beta \parallel^2 + C \sum_{i=1}^{n} max(0, 1 - y_i f_{\beta}(x_i)), \quad (3)$$

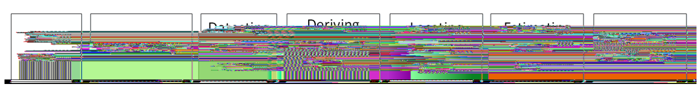
here $max(0, 1 - y_i f_{\beta}(x_i))$ s the st n r h nge loss n the const nt C controls the rel t ve e ght of the regul r t on ter [11]. The purpose of n ng For ul (3) s to cl ss f n o ject x correctl n re uce the o ulus of β.

3. HOMOGRAPHY BASED ATTACK AGAINST TOUCH SCREEN

In th s sect on, e ntro uce the s c e of the tt c n e ch step n et l.

3.1 Basic Idea

F gure 3 sho s the o ch rt of the uto t c n l n recog-n t on of touche e s on o le ev ces.



F gure 3: or o of Bl n Recogn t on of Touche e s

thout loss of gener l t , e often use the four- g t p ssco e n-put on Pho e s the e ple. **Step 1** - T e v eo of the v ct t pp ng on ev ce. e o not ssu e the v eo recor s n te t or popups h le e ssu e the nger ove ent n the t rget e-v ce's screen surf ce re recor e . **Step 2** - Preprocess the v eo n eep onl the touch screen re th ov ng ngers. e s-su e th t the t pe of ev ce s no n or c n e etecte so th t e lso o t n h gh resolut on ge of the correspon ng soft re e o r on the touch screen surf ce, enote s *reference image*, s sho n n F gure 2. **Step 3** - Detect the *touching frames*, n h ch the nger touches the screen surf ce, s sho n n F gure 1. **Step 4** - I ent f fe tures of the touch screen surf ce n er ve the pl n r ho ogr ph tr et een th s touch ng fr es n the reference ge. **Step 5** - E plo DPM n v r ous co puter v s on tech-n ques to o t n l ge o oun ng the touch ng nger t p. Th s s e step of ple ent ng n uto t c touche e recogn -t on. Ho ever, e tr steps re nee e to ctu ll n the touche po nt th t c n e ppe to the reference ge n recogn e the touche e . e enote the rect use of DPM n ng the touche po nt s the s el ne etho . **Step 6** - F n the nger t p contour

n the l e oun n o n tr n t n oun n o roun the fi n er t p top s the ccur te touche r e . Step 7 - Bu l o el of the touch n process, ent f the touche po nts f ro the est t e t n touche r e n p the to the reference e v the ho o r ph . If the touche po nts c n e correctl loc te , e c n sclose the correspon n touche e s. e ntro uce the seven steps n et l elo .

3.2 Step 1 - Taking Videos

The tt c er t es v eo of v ct t pp n on ev ce f ro st nce S uch scen r os nclu e stu ents t n cl sses, rese rcher s tten n conferences, n tour sts o er n n rest n n squ re. th the evelop ent of s rtphones n e c s, stealth t t c t such cro e loc t on s fe s le. For e ple, c er s of phone, *Google Glass* n even s rt tch h ve ecent resolu t on. *Galaxy S4 Zoom* h s 16 e p el ( ) fe r c er th 10 oo lens, e ht n onl 208 . A on sell s e c le plu le 2 B 20 t l croscope th 10 -50 opt c l oo lens [32].

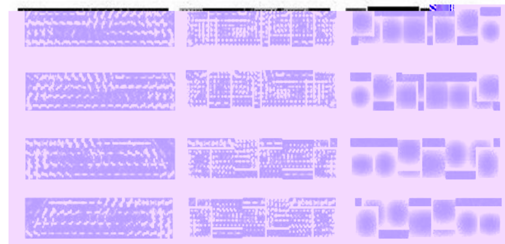
Three f ctors n t n v eos ffect the success of the tt c : c er n le, st nce et een the t et n the c er n l ht n over the t et. The success of the tt c rel es on ccur te ent fi c t on of touche po nts. The c er n le nee s to e ju ste n or er to recor the fi n er ove ent over the touch screen. For e ple, n conference roo , n tt c er n the front c n use the front c er of her phone to recor person t pp n n the c ro . The c er c nnot e too f r f ro the v ct . Other s e, the e s n fi n ers n the e re too s ll to e fferent te . Intu t vel , c er th n opt c l oo lens c n help n such c se. Ho ever, the scenes of nterest n our conte t not llo c er s th lens. ht n ffects the r htness n contr st of the v eo n thus the reco n t on result.

3.3 Step 2 - Preprocessing

S nce e re p rt cul rl nterest e n the fi n ert p re , here the fi n er touches e , our fi rst preprocess n step s to ppl D to etect n loc te the touch ev ce n the v eo. e then crop the v eo n eep the re on of the touch ev ce th ov n fi n ers. Cropp n re oves uch the c roun n es l ter process n s pler.

To ppl D to the etect on of the t et ev ce n e ch v eo fr e, e fi rst nee to ener te pos t ve t (such s) n ne t ve t (c roun) to tr n t et ev ce o el. To et the pos t ve t , e t e 100 es of the t et ev ce such s f ro fferent v e po nts, n nu ll l el the ev ce th oun n o . To et t ht oun n o of the ev ce n n e, e fi rst er ve the ho o r ph rel t on et een the ev ce e n the reference e n F ure 2, n then p the four corners of the ev ce (n th s e ple) n the reference e to the tr n n e. The up-r ht oun n rect n le of the four po nts ccur tel l ts the ev ce n the tr n n e. To e r ve the ne t ve t , e r n o pl o 900 c roun es f ro the S t se [2] n l el o jects th t h ve s l r sh pe to the t et ev ce. D ll lso ener te the ne t ve t tself us n ts o n t n n etho s.

A t et ev ce ppe rs fferent n es f ro fferent v e po nts. Thus, e nee to tr n ult -co ponent o el. F ure sho s the four co ponent o el of . The fi rst ro o els v e e f ro the r ht, n the secon ro o els v e e f ro the left, n the th r n fourth ro s o el v e e f ro the r ht front n left front of . The fi rst colu n sho s the root o el (the co rse o el ch r cter n s hole), the sec on colu n sho s ts p rts f ro fferent v e po nts, n the th r colu n v su l es the sp t l o el of the loc t on of e ch p rt rel t ve to the hole o ject. Th s ture o el effect vel ch r c ter es the structure n fe tures of . After tr n n , e ppl the le rne o el to the v eo fr es, n the ev ce s ccur tel loc l e s sho n n F ure 5.



F ure : Tr ne D o el

F ure 5: Detecte (n e)

D s ver t e-consu n o ject etector n s not co put t on efi c ent. If the t et ev ce s st t c n the v eo, e just nee to etect the t et ev ce n the fi rst fr e n crop the s e re of the t et ev ce n ll the v eo fr es. Other se, e h ve to use D n tr c the t et ev ce n ever fr e.

The secon preprocess n step s to t ll enh nce the e resolut on of the t et ev ce. e t ll n f the croppe v eo fr es. For e ple, e res e e ch croppe fr e to f our t es ts or n l s e.

The th r preprocess n step s to o t n the reference e of the soft re e o r on the t et ev ce. e ssu e the t et ev ce r n s no n n the tt c er c n et h h qu l t e of the soft re e o r on the touch screen. Th s e s the "reference e", s sho n n F ure 2. The e sh ll sho et le fe tures of the ev ce, p rt cul rl the touch screen sur f ce. For e ple, for , e choose l c ll p per so th t the touch screen h s h h contr st th ts h te fr e. It s not ffi cult to reco n e ost t lets n s rtphones s nce e ch r n h s s l ent fe tures. For e ple, l n p st the v ct , the t t c er c n no the ev ce r n . The tt c er lso reco n e the ev ce r n f ro the v eo.

3.4 Step 3 - Detecting Touching Frames

Touch n fr es re those v eo fr es n h ch the fi n er touch es the screen surf ce. To etect touch n fr es, e nee to n l e the fi n er ove ent p ttern of the touch n process. Here e n l e the c se of people us n one fi n er to t p on the screen n nput the p ssco e hile e ten our or to t pp n th ul t ple fi n ers n t o h n s n ect on 8.

Dur n the touch n process, the fi n ert p fi rst oves o n r to r s the touch screen, stops, n then oves up r f ro the touch screen. The fi n er lso ove left or r ht hile ov n o n r or up r . e efi ne the rect on of ov n to r the ev ce s pos t ve n the oppos te rect on s ne t ve. In the process of e e n touche , the fi n ert p veloc t s fi rst pos t ve hile ov n o n r , then ero hile stopp n on the

screen n i n ll n t ve h le ov n up r . h s process re- pe ts or e ch touch e . here ore, touch n r e s the one here the i n ert p veloc t s ero o et es the i n er oves so st th t there s no r e here the i n ert p h s ero veloc t . In such c se, the touch n r e s the one here the i n ert p veloc t ch n es ro pos t ve to n t ve.

he ch llen e to er ve the i n ert p veloc t s to ent the i n ert p . he n le ro h ch e t e the v eo ects the sh pe o the i n ert p n the v eo. he i n ert p sh pe lso ch n es hen the so t i n ert p touches the h r touch screen sur ce. eople lso use erent r e s o the i n ert p to t p the screen. e i n th t hen people touch e s th the i n ert p, the hole h n ost l el eeps the s l r esture n oves n the s e rect on. Inste o tr c n the i n ert p to ent touch n r e, e tr c the h n , h ch h s enou h nu er o e ture po nts or n uto t c tr c n .

e e plo opt c l o theor [38] to er ve the veloc t o e - ture po nts on the ov n h n . pt c l o co putes o ect o- t on et een t o r es. he spl ce ent vector o the po nts e- t een su sequent r es s c lle the e veloc t or the opt c l o t th t po nt. e e plo the L l or th [ ], h ch c n tr c sp rse po nts. o e the L l or th e ect ve, e select un que e ture po nts, h ch re o ten corners n the e. he S h o s corner etector [35] s ppl e to o t n these po nts. e tr c sever l po nts n c se so e po nts re lost ur n the tr c - n . ur e per ents sho th t e ch touch th the i n ert p pro uce ult ple touch n r es. h s s re son l e s nce the i n- er t p s so t. hen i n ert p touches the screen, t or s n th s or n process t es t e. eople lso ntent on l- l stop to e sure th t e s touche . Dur n the nter ct on et een i n ert p n touch screen, so e tr c e po nts lso ove up r n cre te no se or etect n touch n r es. e use s ple l or th to e l th ll the no se: the veloc t o ost tr c e po nts n r e oves ro pos t ve to n t ve, th t r e s touch n r e. ur e per ents sho th t i ve e tures po nts re re l le or etect n ll touch n r es.

3.5 Step 4 - Deriving the Homography Matrix

In co puter v s on, uto t c ll er v n the ho o r ph tr H o pl n r sur ce n t o es s ell stu e pro le [ ]. F rst, e ture etector such S IF, S c le-Inv r nt Fe ture r ns or) [2] or S R (Spee e Ro ust Fe tures) [ ] s use to etect e ture po nts. tch n etho s such s F A (F st r r or Appro te Ne rest Ne h ors) [3] c n e use to tch e ture po nts n the t o es. he p rs o tch e po nts re then use to er ve the ho o r ph tr v the l or th o AC (o S A ple Consensus) [8].

Ho ever, those co on co puter v s on l or th s or er v n ho o r ph H re not e ect ve n our conte t. Bec use o the n le o t n v eos n re ect on the touch screen, there re e oo e ture po nts n the v eo r es or the l or th s ove to or e ect vel . Intu t vel , touch screen corners re potent l oo e tures, ut the re l urr n our conte t s nce the v eo s t en re ot el n the resolut on s poor. here ore S IF or S R c nnot correctl etect these corners.

e er ve the ho o r ph tr H n For ul ( ) s ollo s. H h s 8 e rees o ree o . here ore, to er ve the ho o r ph tr , e nee p rs o tch n po nts o the s e pl ne n the touch n r e n re erence e. An three o the shoul not e coll ne r [ ]. In our c se, e tr to use the corners o the touch screen s sho n n F ure n F ure 2. Bec use the corners n the e re l urr , to er ve the coor n tes o these corners, e i rst etect the our e es o the touch screen. he ntersect ons o these e es re the es re corners. e ppl the C nn e e etector [9] to e tr ct the e es n use the Hou h l ne etector [29] to er ve c n te l nes n the e. e nu ll choose the l nes l e to the e es. h s s the onl nu l proce ure n our ent re s ste o l n l reco n n touch e s. A ter e es re er ve , no e c n c lcul te the ntersect on po nts n er ve the coor n tes o the our corners o nterest. th these our p rs o tch n po nts, e c n er ve the ho o r ph tr th the D (D rect ne r r ns or) l or th [ ]. the ev ce oes not ove ur n the touch n process, th s ho o r ph tr c n e use or ll the v eo r es. ther se, e h ve to er ve H or ever touch n r e n the re erence e.

3.6 Step 5 - Locating the Touching Fingertip

In th s step e loc te the touch n i n ert p n the touch n r e to ent here the i n ert p touches the screen. hen e c n p the touche po nt to the re erence e the ho o r ph tr n or er to et the touche e . A n, e turn to the D o ect etector to loc te the touch n i n ert p n touch n r es.

he process o e plo n D to loc te the touch n i n er- t p s s l r to the process o ppl n D to the etect on o the t et ev ce n v eo r e. e i rst ener te pos t ve - t (touch n i n ert p) n n t ve t (non touch n i n ert p) to tr n o el or the "touch n " i n ert p. o et the pos t ve t , e t e v eos n v r ous scen r os n o t n the touch n r es. For e ch touch n r e, e l el the touch n i n ert p th n ppropr te oun n o cente e t the center o the touche e . e er ve the center o e n touch n r e n the ollo n . Dur n the tr n n process, e no the touche e s n c n er ve the r pos t on pp n the re o e ro the re - erence e to the touch n r e th the pl n r ho o r ph . As e no , D nee s oun n o th t s l e enou h to per or ell lthou h e nt oun n o s s ll s poss - le. e ev lu te oun n o es o erent s e. he opt l oun n o n our conte t s the one oun n the i n ert p, cen- tere t the touche e n h s s e o 0 × 30 p els. erent oun n o s es re use or tr n n es, D res es the oun e re to un or s e. o et the n t ve t , e use the oun n o roun the non-touch n i n ert p. D lso ener-

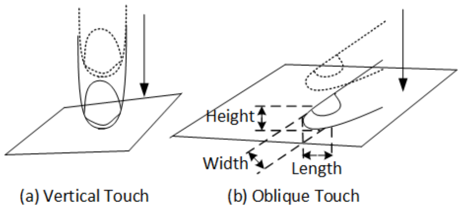


Figure 11: Touching Gestures

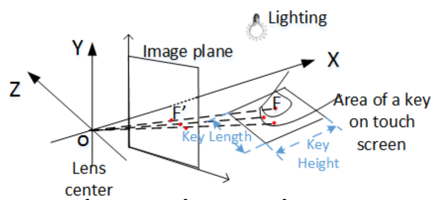


Figure 12: Fingertip Image Formation

on the fingertip will project to the point F' on the image plane. Its brightness in the image will be determined by the lighting and the fingertip shape. Because of the lighting difference, points on the side of the finger facing the touch screen are dark in the image. Adjacent to the dark area is the gray area where lighting is weak. There is also the bright area on the finger that is well illuminated.

Figure 13 shows the longitudinal view of a finger touching the surface. We use Figure 13 to discuss our basic principle of inferring touched keys. K_f and K_b are the front and back of the touched key $\mathcal{K}$ respectively. T is the touched point. Apparently T is on the line segment $\overline{K_fK_b}$. T and $\overline{K_fK_b}$ are projected onto the image plane as T' and $\overline{K'_fK'_b}$. If we can identify T' in the image, our problem is solved. However, as we can see from Figure 13, since the human finger has a curved surface, the camera may not be able to see the touched point. OT_o is the tangent line to the curved finger surface and it intersects the touch screen surface at T_o. The camera can see T_o, which is the closest point to the touched point on the touch screen surface. T_o is projected as T'_o on the image plane. If T'_o is on the line segment $\overline{K'_fK'_b}$, then we just need to find T'_o in the image and T'_o can be used to determine the touched key.

We argue that T'_o generally lies in the area of a key. Table 1 shows the key size of the *unlock screen software keypad* for iPad, iPhone and Nexus 7 tablet. Figure 12 gives the definition of key height and length. Table 2 gives the average size of fingertips of index and middle fingers of 1[illegible] students of around 27 years old, including [illegible] females and 10 males. The fingertip height is the distance from the fingertip pulp to the fingernail. The fingertip length is the distance between the fingertip pulp to the far front of the finger. When people touch the screen, they generally use the upper half of the fingertip to touch the middle of the key so that the key can be effectively pressed. We can see that half of the fingertip is around 4.5mm, less than the key height for all devices in Table 1. Moreover, according to Tables 1 and 2, the fingertip width is smaller than the key length. Therefore, the fingertip generally lies inside the key area, as shown in Figure 13. That is, the far front of the fingertip F in Figure 13 is in the range of the key and the touched point is inside the key area. Based on the perspective projection, T_o is on the segment of $\overline{K_fK_b}$ so that T'_o is on the segment of $\overline{K'_fK'_b}$ whenever the fingertip is in the view of the camera.

In the QWERTY keyboard of iPhone and other small smartphones, keys are very small. In these scenarios, people often use vertical touching or touch with the fingertips in order not to touch wrong keys. That is, the fingertip top lies inside the key area. The analysis above is still valid. Our experiments on the QWERTY keyboard also validate this analysis.

Table 1: Unlock Screen Keypad Key Size - Height × Length (mm)

	iPad	iPhone 5	Nexus 7
Height (mm) × Length (mm)	9 × 17	8 × 16	10 × 16

Table 2: Fingertip Size (σ: Standard Deviation)

	Index Finger		Middle Finger	
	Average	σ	Average	σ
Height (mm)	9.4	1.2	10.[illegible]	1.3
Length (mm)	12.9	1.4	13.1	1.7
Width (mm)	13.1	1.9	13.7	1.7

There are cases that T'_o is not on the line segment $\overline{K'_fK'_r}$, corresponding to the touched key $\mathcal{K}$. Figure 14 illustrates such a case. Please note we intentionally draw a large finger for clarity. In this case, the key, such as one on the keyboard for non-unlock screen, is so small. The camera is too close to the finger and takes such a wrong angle that T_o lies outside $\overline{K_fK_r}$. Therefore, T'_o is not on the line segment $\overline{K'_fK'_r}$. In such cases, our observation is that T'_o generally lies into the rear part of the key $\mathcal{K}'$ in front of $\mathcal{K}$. We define the percentage α. If an estimated touched point lies in the rear α of $\mathcal{K}'$, the touched key is $\mathcal{K}$.

We now derive the size of a key in an image and investigate its impact. The camera focus length is f. The height from the camera to the touch screen surface is h. The physical key height $|\overline{K_fK_b}| = w$. The distance between the key front K_f and the lens center is d. By geometric operations, we have

$$|K'_fK'_b| = \frac{fh}{d(1+d/w)}. \quad (4)$$

If the physical key length is l, the key length l' in the image is,

$$l' = \frac{fl}{d}. \quad (5)$$

From Formulas (4) and (5), the farther the touch screen from the camera, the smaller the size of the key in the image. The smaller the physical key size, the smaller of the key in an image. Table 3 gives the camera specifications of the cameras used in our experiments: Logitech HD Pro Webcam C920 [24], the iPhone 5 camera and the Google Glass camera. If the camera is around 2 meters and half meter away from the target, according to Formula (4) and our experiments, the key height is only a few pixels. Therefore, in our experiments, we often need to zoom the fingertip image for accurate localization of touched points. We can also derive the key size in the touching frames practically by using the homography from the reference image to the touching frames. The key area in the reference image is known, thus the key size in the touching frames can be derived.

4.2 Clustering-based Recognition of Touched Points

Based on the model of the touching finger in an image, we now introduce the clustering-based strategy recognizing touched keys. If we can derive the position of the touched point T'_o in Figure 15, we can infer the corresponding key by applying the homography. The problem is how to identify this touched point[2]. Intuitively, since T'_o is far below the fingertip, which blocks lighting rays, T'_o should be in the darkest area around the fingertip in the image.

We now analyze the brightness of the area around the fingertip. The fingertip is a very rough surface at the microscopic level and can be treated as an ideal diffuse reflector. The incoming ray of light is reflected equally in all directions by the fingertip skin. The

[2] Touched points actually form an area under the fingertip.

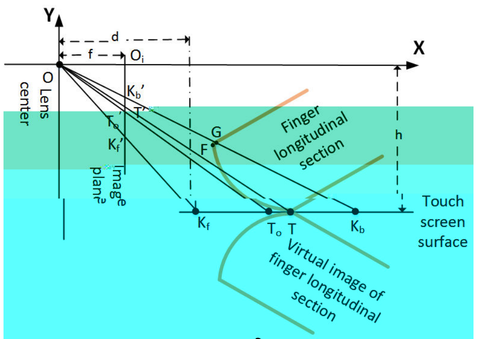


Figure 13: Touching Points ins the the

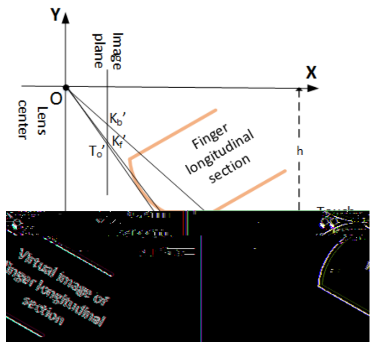


Figure 1 : Touching Points outs the the

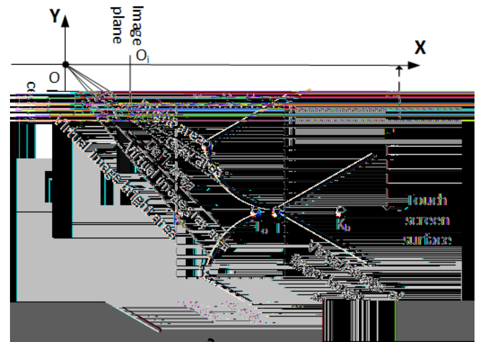


Figure 15: Five Groups of Fingertip

reflection conforms to the Lambert's Cosine [38]: the reflected energy from a small surface area in a particular direction is proportional to cosine of the angle between the particular direction and the surface normal. Therefore, for the lower part of the fingertip arc facing the touch screen, denoted as the inner side of the fingertip, the angle is larger and less energy will be reflected so th trorr

ﬀerent evces s ll s the pctof ﬀerent n sof e o r - s. In ll e per ents, e tr to recogn e - g t or -ch r cter p sscod es, h ch re r n o l gener te . The success r te s e- ﬁ ne s the pro l t th t the p sscod es re correctl recogn e .

In t on to ﬀerent c er s n t rget evces, e lso con- s er the pctfro the follo ng f ctors: users, the st nce e- t een the c er n t rget evce, n the c er ngle.

Users: Dﬀerent people h ve ﬀerent ﬁ nger sh pe, ﬁ ngern l n touch ng gestures. F ve fe les n s les th the e pe- r ence of us ng t lets n s rtphones p rt c p te n the e per - ents. The ere sep r te to three groups: 3 people n the ﬁ rst group, n people n the secon group. These t o groups per- for e e per ents th . The l st group help us ev lu te the success r te versus the st nce et een the c er n the t r- get, ﬀerent c er s versus ﬀerent evces, n the e c er versus ﬀerent n s of e o r s. For the ﬁ rst group, e too 10 v eos for ever person t e ch ngle (front, left n r ght of the t rget evce). For the secon group, ﬁ ve v eos ere t en for ever person per ngle. D sc r ng 3 v eos not recor ng the hole touch ng process, e o t n 192 v eos tot ll . Dur ng the e per ents, users t p n the r o n thout n ntervent on.

Angles and Distance: To e sure the pct of the ngle, e pl ce the t rget n front, on the left (3 o'cloc) n on the r ght (9 o'cloc) of the c er . In the ﬁ rst t o groups of e per ents, the c er s 2.1 eters ( ) to 2. fro n roun 0 .5 ove the evce. To test ho the st nce ﬀects the recogn t on results, e lso pos t one the c er , the og tech HD ro e - c C920 , n front of the t rget evce, n , t st nce of 2 , 3 , n 5 , n ppro t el one eter ove the t rget.

Lighting: The l ght ng ﬀects the r ghtness n contr st of the ge. The e per ents re perfor e n cl ssroo th le l ps on the ce l ng. The ﬁ rst group of v eos ere t - en un er nor l l ght ng n the secon group of e per ents ere t en un er strong l ght ng. All other e per ents ere perfor e un er nor l l ght ng. D r ness ctu ll helps the tt c s nce the touch screen s r ghter n the r . e not cons er these e s r scenes n our e per ents.

5.2 Detecting Touching Frames via Optical Flow

As scusse n Sect on 3. , e tr c h n 's fe ture po nts n use the r veloc t ch nge to etect touch ng fr es. ur e per - ents sho th t 5 or ore fe ture po nts re st le for tr c ng touch ng fr es th true pos t ve of 100 %, s sho n n T le . The opt c l ﬂ o l or th lso pro uce f lse pos t ves, f lse recogn ng fr es n h ch ﬁ nger oes not touch the screen sur- f ce s touch ng fr es. The f lse pos t ve r te s ver lo , less th n 1 % s sho n n T le . ne to re uce the f lse pos t ve s to use D . ur e per ents sho th t D s le to etect touch- ng fr es s nce no ﬁ ngers touch the screen n non-touch ng fr es n D onl recogn es touch ng ﬁ ngers n touch ng fr es. e e clu e the non-touch ng fr es D f the nu er of etecte touch ng fr es s ore th n opt c l ﬂ o .

T le : Perfor nce of Detect ng Touch ng Fr es

	Front	eft	ght	Aver ge
True os t ve	100 %	100 %	100 %	100 %
F lse os t ve	0 .91 %	0 .88%	0 .88%	0 .89%

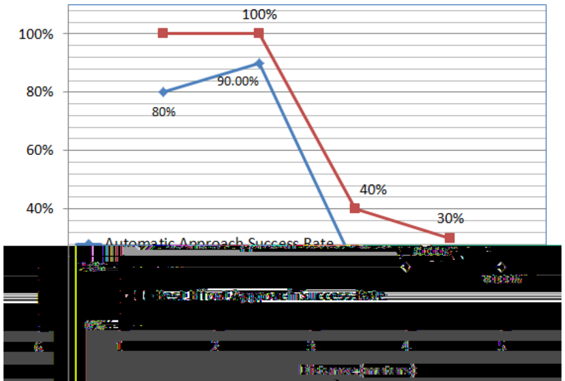
5.3 Recognizing Touched Keys on iPad via Webcam

T le 5 sho s the result of the s l ne etho for v eos t - en fro ﬀerent v e po nts. Its over ll success r te s less th n 30 %. Therefore, the s l ne etho s not ver ﬀect ve s nce D c nnot ccur tel loc te the touche po nts.

T le 5 S uccess te B s l ne etho

	Front	eft	ght	Aver ge
S uccess te	2[illegible]%	290 3%	22.22%	2[illegible] 3%

Fro no on, e present e per ent results us ng the seven-step recogn t on etho , referre to s Auto t c Apprp ch (AA), n- tro uce n Sect on 3. e lso use etr c c lle the Best Eﬀort Appro ch (BEA) success r te, h ch s er ve g v ng sec- on tte pt for correct ng rong recogn t on th so e hu n ntervent on. The BEA s perfor e n the follo ng . e of- ten see one or t o rong e s n the f le e per ents. So e of these rong e s re c use D th t f ls to etect the touch- ng ﬁ ngert pS o et es, even f the touch ng ﬁ ngert p s etecte , the ge c n e so lurr th t p els roun the touch ng ﬁ ngert p h ve l ost the s e color n t s ﬁ cult to er ve the ﬁ nger- t p contour n F gure 9. ther ﬁ ngers lso loc the touch ng ﬁ nger n ncur rong recogn t on of the touch ng ﬁ ngert p top. Therefore, e often no h ch e ght e rong n g ve the secon tr . e nu ll select the s ll oun ng o of the ﬁ n- gert p n F gure 8 or the touche re n F gure 10 to correct such errors. As n l e n Sect on , for e ch touch e lso pro- uce t o c n tes. s ng one of the t o cho ces, e c e co uc ero of t n( )2(l)-3(l)l 3(9) -3( )2(5(e)2(s)-2( ) 29(An-295e)99l [(e)2A)-3(n2)5(c)2(e)-2



F gure 1 S uccess te v.s. D st nce

t o eters fro n out 0. ove the ev ce. o n- vest e te the p ct of fferent c er s, e con ucte 3 e per- ents us n hone 5 to recor p ssco e nputs on , fro s r st nce n t s r he ht. 3 e p r ents th the Goo le l ss recor n p ssco e nputs on ere per or e t o eters n t hu n he ht. F ure 18 presents the re- sults. he AA success r te s ore th n 80 %, n the BEA success r te s ore th n 90 % n ll c ses. h h h success r te or ll the c ses e onstr tes the sever t of our tt c .

F ure 18: S uccess te Co p r son

e lso teste the effect of our tt c on fferent n s of e- o r : the E Y e o r n hone E Y e- o r . he E Y e o r e s l er th n the hone E Y e o r e . 3 e p r ents ere one respect vel th the e c er fro the front of the t et fro out 2.2 eters n t he ht of 0. eters. F ure 18 presents the results. It c n e o serve th t the AA success r te s over 80 % n the BEA success r te s over 90 %!

6. DISCUSSION: TOUCHING WITH MULTIPLE FINGERS AND TWO HANDS

As sho n n F ure 19, people t pe the p ssco es, or the r n ccount p ss or s on l e e o r (n F ure 21) th ult ple f n ers n t o h n s. In th s sect on, e e ten our or n the prev ous sect ons n scuss the reco n t on of touch nput t o h n s n ult ple f n ers.

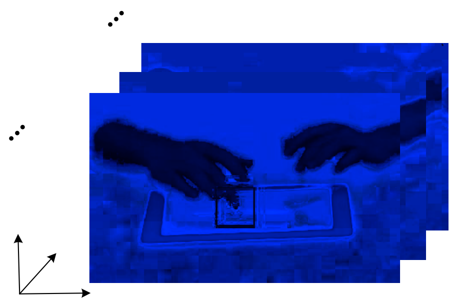
F ure 19: ouch n Fr e th ult ple F n ers (n e)

In the c se of ult ple f n ers, the ch llen e s to reco n e h ch f n er s the touch n f n er. In [3], touch n s e ne s f n er- t p hover n t cert n pos t on or su en ch n n the ov n rect on f or nother e . In the c se of touch n th ult ple f n ers, th s s not true n ore. As sho n n F ure 19, the non- touch n f n ert ps lso f ollo s l r p ttern ur n the touch n process f oth the l ttle f n er of the r ht h n n le f n er of the left h n re use f or t p n . he str te n [3] f or er v n the f n ert p's ov n tr jector c nnot or rect l f ult ple f n ert ps re nvolve .

In Sect on 3. hen e ress the c se of touch n th one f n er, e use the f n er's veloc t rect on ch n e to etect the touch n f r es. he touch n f n er n other f n ers f ollo s l r p ttern of veloc t rect on ch n e. he touch n f r e s the one n h ch the jor t of those f n ers ch n e the veloc t fro pos t ve to ne t ve. In the c se of ult ple f n ers, n or er to use the s l r str te , e h ve to fferent te n tr c the t o h n s. he co pl c te c roun nvolv n the t o h n s s s re t ch llen e f or ener l sol ut on.

e ress the ch llen es ove loo n ore closel t h t s touch n . he f n er ove ent re sters touch nput onl hen the f n ert p ctu ll touches the screen. heref ore, e etect touch n f r es etect n touch n ct ons fro the perspec- t ve of ct on etect on [5] (lso terme ct on loc l t on [2], or event etect on [20]). A v eo c n e o ele s sequence of fr es c pture lon the t e, s sho n n F ure 20. e c n etect e ch touch ct on th S D- $ p t ote por l D for le r n o el [39]). ne touch event usu ll nvolves sever l touch- n f r es. heref ore, e c n tre t the touch n ct on etect on pro le s the pro le of etect n sets of touch n f r es.

e use D- to etect touch n f r es n loc l e the touch- n f n er n the c se of t p n th ult ple f n ers. As scusse n Sect on 3. D- c n loc l e the touch n f n ert p n touch- n f r es eff ect vel . heref ore, ppl n D- to ll v eo fr es, e c n etect n loc l e ll the touch n f n ert ps n the touch n f r es, e clu n non touch n f n ert ps. he oun n o n F ure 20 sho s the etecte touch n f n ert p n the c se of t p n th ult ple f n ers. It s lso o serve th t the touch event nvolves f e consecut ve touch n f r es lon the t- s s sho n n F ure 20.



F ure 20: ne ouch Event Involv n ult ple Fr es n V eo

For ever touch n ct on, e use the touch n f r e n the le s the ctu l touch n f r e. G ven the touch n f r e, e opt the s e steps ntro uce n Sect on to er ve the touche e . F ure 21 sho s the ppe result (reen ot) f or the touch event n F ure 20, here "" s the touche e . Fro the scus- s on ove, e c n lso see th t touch n th one f n er (or h n) s spec l c se of touch n th t o h n s n ult ple f n ers.

o v l te the tt c nst t p n th ult ple f n ers, e perf or e 21 e per ents th the e c er sp n on the - ch r cter e o r , fro st nce out 2.2 eters n

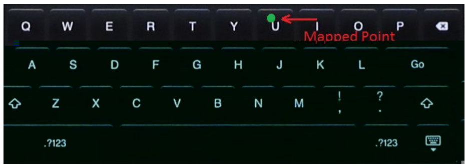


Figure 2: Multi-touch Mapping Result

0.45 meters above the device. We achieve the AA success rate of 95.2%. In these experiments, only one touch event is not correctly detected. For the manually retrieved touching finger for that touch event as introduced in previous sections, we get the BEA success rate of 100% to retrieve the character pressed even when there is multiple fingers. This demonstrates the correctness and severity of the attack introduced in this paper.

7. RELATED WORK

In this paper, we exploit the movement of the touching finger to infer the input on a touch screen. It is one type of side channel attack. There are various side channel attacks on touch-enabled devices. We exploit different channels than others. Because of the space limit, we discuss the most related work on side channels using computer vision knowledge. Backes et al. [2, 1] exploit the reflections of computer monitor on glasses, tea pots, spoons, plastic bottles, and eyes of the user to recover what is displayed on the computer monitor. Their tools include a SLR digital camera Canon EOS 400D, a refractor telescope and a Newton reflector telescope, which can successfully spy from 30 meters away.

Balzarotti et al. propose an attack retrieving text typed on a physical keyboard from a video of the typing process [3]. When keys are pressed on a physical keyboard, the light diffusing surrounding the key's area changes. Contour analysis is able to detect such a key press. They employ a language model to remove noise. They assume the camera can see fingers typing on the physical keyboard. Maggi et al. [28] implement an automatic shoulder-surfing attack against touch-enabled mobile devices. The attacker employs a camera to record the victim tapping on a touch screen. Then the stream of images are processed frame by frame to detect the touch screen, rectify and magnify the screen images, and ultimately identify the popping up keys. Raguram et al. exploit reflections of a device's screen on a victim's glasses or other objects to automatically infer text typed on a virtual keyboard [33]. They use inexpensive cameras (such as those in smartphones), utilize the popup of keys when pressed and adopt computer vision techniques processing the recorded video in order to infer the corresponding key although the text in the video is illegible.

Xu et al. extend the work in [33] and track the finger movement to infer input text [43]. Their approach has five stages: in Stage 1, they use a tracking framework based on AdaBoost [13] to track the location of the victim device in an image. In Stage 2, they detect the device's lines, use Hough transform to determine the device's orientation and align a virtual keyboard to the image. In Stage 3, they use Gaussian modeling to identify the "fingertip" (not touching points as in our work) by training the pixel intensity. In Stage 4, RANSAC is used to track the fingertip trajectory, which is a set of line segments. If a line segment is nearly perpendicular to the touch screen surface, it implicates the stopping position. In Stage 5, they apply image recognition techniques to determine the keys which were most likely pressed given the stopping positions. In Stage 6, they apply a language model to optimize the result given the candidate sets and associated confidence values from the previous stage. They use two cameras: Canon VIXIA HG21 camcorder with 12x optical zoom and Canon 60D DSLR with 400mm lens.

In comparison with [43] on recognizing input passwords, we can achieve a much higher success rate. We extend our work to the scenario of touching with one hand and multiple fingers while such scenario is not addressed in [43].

8. CONCLUSION

In this paper, we present a computer vision based attack that blindly recognizes inputs on touch screen from a distance automatically. The attack exploits the homography relationship between the touching images (in which fingers touch the screen surface) and the reference image of a soft keyboard. We use the optical flow algorithm to detect touching frames. Deformable Part-based Model (DPM) and various computer vision techniques are utilized to track the touching fingertip and identify the accurate touched area. We carefully analyze the image formation of the touching fingertip and design the k-means clustering strategy to recognize the touched points. Homography is then applied to recognize the touched keys. Our extensive experiments show that the AA success rate of recognizing touched keys is more than 80%, while the BEA success rate is more than 90%. We have also extended the attack to the case of typing with one hand and multiple fingers and achieve a high success rate of more than 95%. As countermeasure, we design a context aware Randomized Virtual Keyboard (RVK) which pops up a randomized keyboard on Android systems for sensitive information input such as passwords. Our future work includes further refinement of the attack and design of alternative authentication strategies for mobile devices.

9. ACKNOWLEDGEMENTS

This work was supported in part by National Key Basic Research program of China under grant 2010CB32810[illegible], MOE Key Lab FDCA, 0[illegible]-2011-A3, International S&T Cooperation Program of China under grant 2013DFA10[illegible]90, US NSF Grants 111[illegible], CNS-131[illegible]89[illegible]8 and 12[illegible]22[illegible]5, National Science Foundation of China under grant [illegible]12[illegible]20[illegible]5[illegible]. Any opinions, findings, conclusions, and recommendations in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

10. REFERENCES

[1] M. Backes, T. Chen, M. Duermuth, H. Lensch, and M. Welk. Tempest in a teapot: Compromising reflections revisited. In *Proceedings of 30th IEEE Symposium on Security and Privacy*, pages 315–327, 2009.

[2] M. Backes, M. Durmuth, and D. Unruh. Compromising reflections or how to read lcd monitors around the corner. In *Proceedings of IEEE Symposium on Security and Privacy*, pages 158–169, 2008.

[3] D. Balzarotti, M. Cova, and G. Vigna. Clearshot: Eavesdropping on keyboard input from video. In *Proceedings of the 2008 IEEE Symposium on Security and Privacy*, SP '08, pages 170–183, 2008.

[4] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool. Speeded-up robust features (surf). *Comput. Vis. Image Underst.*, 110(3):3[illegible]–359, June 2008.

[5] H. Benko, A. D. Wilson, and P. Baudisch. Precise selection techniques for multi-touch screens. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '0[illegible], pages 12[illegible]3–1272, 200[illegible].

[6] R. Biddle, S. Chiasson, and P. van Oorschot. Graphical passwords: Learning from the first twelve years. In *ACM Computing Surveys*, 2012.

[33] Y. Xu, J. Heinly, A. M. White, F. Monrose, and J.-M. Frahm. Seeing double: Reconstructing obscured typed input from repeated compromising reflections. In *Proceedings of the 20th ACM Conference on Computer and Communications Security (CCS)*, 2013.

[34] Q. Yan, J. Han, Y. Li, J. Zhou, and R. H. Deng. Designing leakage-resilient password entry on touchscreen mobile devices. In *Proceedings of the 8th ACM Symposium on Information, Computer and Communications Security (AsiaCCS)*, 2013.

[35] Y. Hu, L. Cao, F. Lv, S. Yan, Y. Gong, and T. S. Huang. Action detection in complex scenes with spatial and temporal ambiguities. *ICCV*, 2009.

[36] J.-Y. Bouguet. Pyramidal implementation of the Lucas Kanade feature tracker. *Intel Corporation, Microprocessor Research Labs*, 2000.

[37] Y. Zhang, P. Xia, J. Luo, Z. Ling, B. Liu, and X. Fu. Fingerprint attack against touch-enabled devices. In *Proceedings of the second ACM workshop on Security and privacy in smartphones and mobile devices*, SPSM '12, pages 57–68, 2012.

Appendix

In this appendix, we discuss countermeasures to computer vision based attacks introduced in this paper and related work. There are many other authentication approaches immune to these attacks to some extent, including using biometric-rich gesture-based authentication [3 , , 21] and graphic password schemes [, 3 , 8]. The idea of randomized keyboard has been proposed for legacy keypad and touch-enabled devices [1 , 15, 30, 1 , 3 , 25, 22]. We have designed and developed the context-aware Privacy Enhancing Keyboard (PEK) for Android for the first time. We have implemented the PEK as a third party app and are also able to change the internal system keyboard to implement the PEK.

Design and Implementation of PEK

A software keyboard contains three sub-keyboards. The primary sub-keyboard is the QWERTY keyboard, which is the most common keyboard layout. The second sub-keyboard is the numerical keyboard that may also contain some symbols. The last sub-keyboard is a symbol keyboard that contains special symbols. The layout for these three sub-keyboards is stored in a XML file, which records the positions of keys and corresponding key codes. The system generates its keyboard in this way: the keys will be read from the XML file and put in a right position.

PEK changes the key layout to implement randomized keys. When a keyboard is created, we first generate a random sequence of key labels for each of the three different keyboards. When a key is read from the XML file, we randomly choose an integer number between one and the size of the key sequence. We use this number to pick the specific key label from the randomized key sequence and also remove this label from the key sequence. This randomly selected key replaces the current key. In this way, we can shuffle the key positions on normal keyboard. Another version of PEK is to make each key move within the keyboard region in a Brownian motion fashion by updating each key's position repeatedly according to the Brownian motion. In this way, the keys are moving all the time. Even if the same key is pressed a few times, their positions are different. This is an improvement compared with PEK with shuffled keys, in which the keyboard does not change in one session of password input. Figure 22 shows PEK with shuffled keys and Figure 23 shows PEK with the Brownian motion of keys.

Figure 22: PEK Shuffled Keys

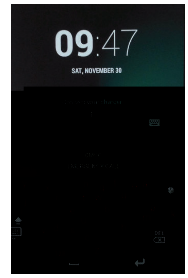


Figure 23: PEK-Brownian Motion

PEK is aware of the context and can pop up the randomized keyboard only if the input box is for sensitive information. The Android class "EditorInfo" can be used to detect the input box type. In our case, TYPE_NUMBER_VARIATION_PASSWORD, TYPE_TEXT_VARIATION_PASSWORD, TYPE_TEXT_VARIATION_VISIBLE_PASSWORD and TYPE_TEXT_VARIATION_WEB_PASSWORD are used to identify the password input box. The first type is the variation of TYPE_CLASS_NUMBER, while the other three types are the variations of TYPE_CLASS_TEXT. Once the password input box is triggered by the user, a new randomized keyboard will be constructed. As a result, the user can have a different key layout every time she presses the password input box.

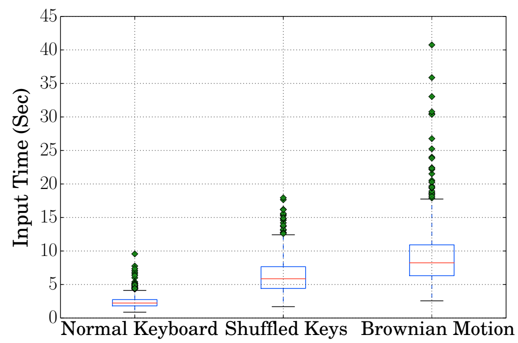


Figure 24: Input Time of Three Distinct Keyboards

Evaluation of PEK

To measure the usability of the PEK, we recruit 20 students, 5 female students and 15 male students, whose average age is 25 years old. We implemented a test password input box and generated 30 random four-letter passwords. The students are required to input these 30 passwords using shuffled keyboard and Brownian motion keyboard, and the test app records the user input time. Table 5 shows the results of our evaluation and Figure 24 gives a box plot of the input time of three different keyboards. The median input time is around 2.2 seconds on the normal keyboard, 5.9 seconds on the shuffled keyboard and 8.2 seconds on the Brownian motion keyboard. The success rate is the probability that a user correctly inputs four-letter passwords. The success rate of all three keyboards are high while the rate is a little lower for the Brownian motion keyboard. The participants in our experiments feel PEK is acceptable for PEK only pops up the randomized keyboard for sensitive information input.

Table 5: Usability Test

	Normal Keyboard	Shuffled Keys	Brownian Motion
Median Input Time (Second)	2.235	5.859	8.2
Success Rate	98.50%	98.83%	9 .7%