Continuous Remote Mobile Identity Management Using Biometric Integrated Touch-Display

Conference Paper · December 2012

DOI: 10.1109/MICROW.2012.9

5 authors, including:

Tao Feng
University of Houston
21 PUBLICATIONS **361** CITATIONS
SEE PROFILE

Dainis Boumber
University of Houston
8 PUBLICATIONS **27** CITATIONS
SEE PROFILE

Weidong Shi
University of Houston
121 PUBLICATIONS **1,221** CITATIONS
SEE PROFILE

Continuous Remote Mobile Identity Management Using Biometric Integrated Touch-Display

Tao Feng*, Ziyi Liu*, Bogdan Carbunar†, Dainis Boumber*, and Weidong Shi*
*Computer Science Department, University of Houston, Email: tfeng3@cs.uh.edu
†School of Computing and Information Sciences, Florida International University, Email: carbunar@cs.fiu.edu

***Abstract*—While enriching the user experiences, the development of mobile devices and applications introduces new security and privacy vulnerabilities for the remote services accessed by mobile device users. A trusted and usable authentication architecture for mobile devices is thus in high demand. In this paper, we leverage a unified structure, consisting of transparent TFT-based fingerprint sensors, touchscreen, and display, to propose a novel identity management mechanism that authenticates users of touch based mobile devices for accessing the local devices and remote services. Our solution differs from the previous one-time and enforced authentication approaches through two novel features: (i) user transparent authentication process, requiring neither password nor extra login steps and (ii) continuous identity management based on fingerprint biometric, where each user-to-device touch interaction is used toward authentication. Moreover, we introduce two different security scenarios, one for local identity management, and the second extended solution for remote identity management. Finally we employ TRUST (Trust Reinforcement based on Unified Structural Touch-display) to solve the identity challenge in cyber space.**

I. Introduction

Mobile devices have steadily infiltrated our daily lives for the last two decades. Smartphones, a more recent introduction in this space, not only have the computing capabilities of desktops of a few years back and the ability to maintain connectivity over a variety of network interfaces but also are continuously improving their I/O capabilities. As a result, companies and government agencies have increasingly took the advantage of these features of mobile computing systems and applications to make their employees more productive. Since employees need to connect remotely to the organization's infrastructure, sensitive information (e.g., confidential documents, trade secrets, credentials) is frequently stored or accessed from the networked mobile devices.

Mobile devices are easily lost or stolen, and the data stored and the points of entry offered into organizations make the devices an obvious targets for attack and intrusion. Mobile authentication becomes therefore an issue of primary importance for both the mobile device users and the service provides.

Current solutions do not handle well the existing tension between security and usability required by efficient mobile authentication mechanisms. First, password based access control is not well suited for mobile authentication. As shown in a study of over 6,000,000 passwords, 91% of all user passwords belong to a list of only 1,000 common passwords [1]. Complex passwords and frequent password changes are more secure, but they are difficult to remember. Second, standard biometric authentication techniques are unpopular on mobile devices due to size requirement. Furthermore, most state of the art authentication approaches do not provide continuous protection. Instead,they only provide a trade-off between usability and security. User authentication is periodic (thus requires more user effort), leaving the device vulnerable between authentication sessions.

Instead, we propose a novel mobile user authentication solution that supports constant and implicit user authentication using a touchscreen integrated with multiple TFT fingerprint sensors. Our solution is seamless, touch based, and biometric enabled. It supports both implicit (passive) and explicit (active) mobile user authentication for accessing either a local mobile device or a remote service.

An essential feature of our solution is its ease of use. Fingerprint images are captured without disrupting normal user-mobile device interactions. It supports instant mobile device access and provides transparent fingerprint biometric for mobile authentication and identity management. In addition, it offers continuous, post-login protection after a user gains access to a mobile device or a remote service using touch based fingerprint biometrics. The proposed authentication mechanism is easy to use, secure and cost efficient.

The main contributions of our work are the following:

- Design a biometric integrated touchscreen technology as a solution for continuous mobile user authentication and identity management.
- Design a biometric touch-display module FLock as a solution for remote identity management.
- Propose a Trust Reinforcement based on the Unified Structural Touch-display (TRUST) solution for mobile user authentication and identity management during access of a remote service.

The rest of this paper is organized as follows. Section II describes the challenges of designing a biometric touch panel and background of the related technologies. The details of the hardware design are described in Section III. The TRUST solution is introduced in Section IV. Section V presents the related work. Section VI concludes the paper.

II. Background and Challenges

We first briefly compare three mobile authentication solutions, and then provide required background on touchscreen and fingerprint sensing technologies.

	Password	Separate Fingerprint Sensor	Fingerprint Sensors Integrated with Touchscreen
Continuous User Verification	No	No	Yes
User Burden	Memorization	Extra Login Step (Rub/Swipe)	No
Login Speed	Typing Speed	Few Seconds	Instant
Transparent to User	No	No	Yes

TABLE I
COMPARISONS OF THREE MOBILE USER AUTHENTICATION APPROACHES

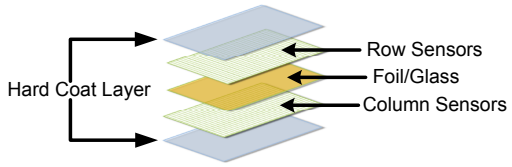

Fig. 1. Touchscreen Sensor

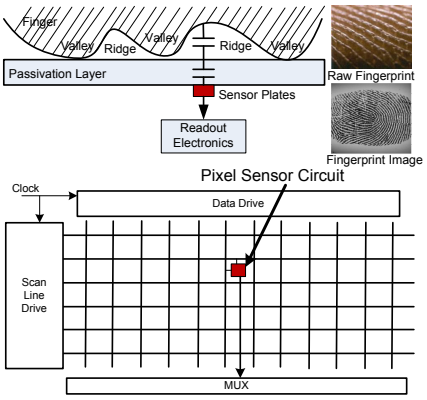

Fig. 2. TFT Fingerprint Sensor with Array of Sensing Cells

A. Comparison of Three Mobile Authentication Solutions

Table I compares three mobile user authentication approaches. Among the three, touchscreen with integrated biometric sensors is the best solution: it supports transparent and continuous user authentication with almost no additional burden or requirements from the users.

B. Touchscreen

Touchscreens, the most widely adopted interacting panel for portable devices such as smartphones, tablets and navigation systems, are mainly add-on types: the touchscreens are separated from the display panel. Touchscreens can be categorized by the used sensing methods as: resistive [3], capacitive [16], acoustic-wave [2], and infrared based touch sensing techniques. Among these, the capacitive based method is increasingly popular due to its sensitivity, durability, and ability to detect multi-touches.

The capacitive touch panel consists of two parallel ITO (indium tin oxide) film layers that are separated from a sensor glass layer, see Figure 1. Each layer contains a pattern of ITO electrodes that can detect capacitive variations caused by the touch of a human finger. The separation of the top and bottom ITO layers supports parallel sensing on both X and Y directions. The top and bottom ITO layers provide row and column touch sensing, respectively. When sensing parallel capacitance of the top ITO layer, the row (X-axis) of a touch can be detected. When sensing parallel capacitance of the bottom layer, the column (Y-axis) of a touch can be detected. The touch points are determined by combining the row and column sensing results. The typical response time of a capacitive touch panel is 4ms.

C. Fingerprint Sensors

Capacitance sensors utilize the principles associated with capacitance in order to form fingerprint images. In this method of imaging, each sensor array pixel acts as one plate of a parallel-plate capacitor. The dermal layer (which is electrically conductive) acts as the other plate. The non-conductive epidermal layer acts as a dielectric.

Among the different fingerprint sensing technologies, the optical and capacitive approaches are the most widely used. Optical fingerprint sensing techniques require a lens system. As such, it is hard to implement in a small package at a low cost, see Figure 3. Similar to the optical sensing techniques, capacitive sensors generate the image of the ridges and valleys of a fingerprint by using electrical current, see top of Figure 2. A capacitive sensor based on the CMOS process is used to capture a fingerprint using a single crystal Si chip by directly touching the surface of the chip. One advantage of this type of fingerprint sensor is that it can be implemented using a thin package. However, CMOS based techniques are difficult to scale when the sensor has to cover a larger area – the sensor is built over a Si substrate. The cost is also prohibitively high when using CMOS based techniques for creating a fingerprint sensor that can cover area as large as a mobile phone display.

One alternative is a TFT (Thin Film Transistor) based fingerprint sensor. The TFT technique is well known for creating large size displays. The technique puts ICs directly onto a glass substrate. It is the most cost effective and scalable way for creating fingerprint sensors that can cover larger areas. In the past, several capacitive fingerprint sensor prototypes and products were developed using poly-Si TFTs [10], [9]. TFTs can be implemented using transparent materials, enabling the electronic manufacturers to build transparent TFT fingerprint readers or displays. For example, Alps Electric demonstrated transparent TFT fingerprint sensors [15]. The response time of capacitive fingerprint sensing devices can be extremely small when optimized and properly designed [24], [20], [21], [10], [9], see Table II.

III. HARDWARE DESIGN

A. Touchscreen with Embedded Transparent Fingerprint Sensors

Our design integrates multiple TFT fingerprint sensors with a touchscreen. A user is authenticated using fingerprint biometrics. The collection of fingerprint data takes two steps. In the first step, the system uses the touchscreen to locate touch

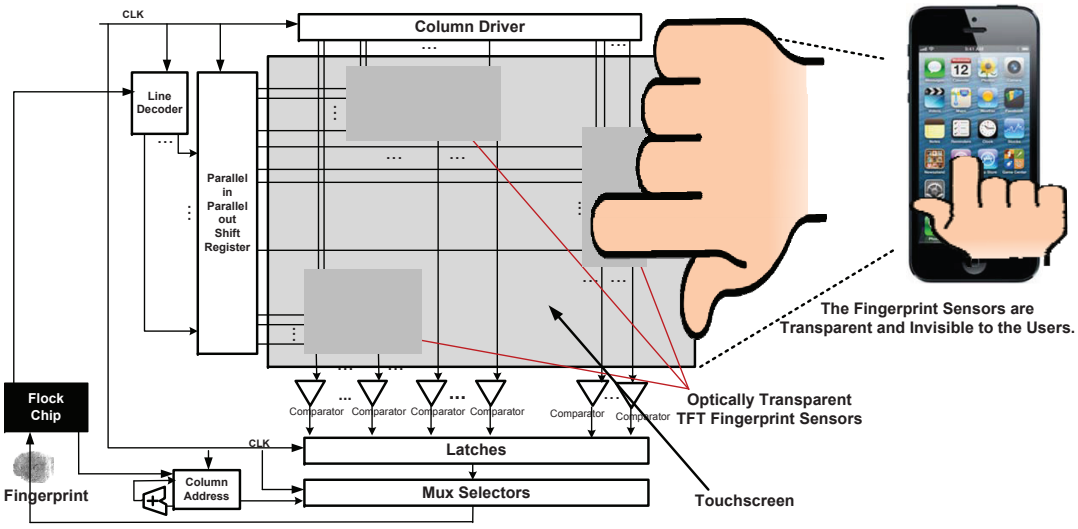

Fig. 4. Touchscreen with Embedded Transparent Fingerprint Sensors

Reference	Cell Size	Resolution	Response	Frequency
[24]	42 μm	64 x 256	3ms	4MHz
[20]	81.6 μm	124 x 166	2ms	Not Mentioned
[10]	60 μm	320 x 250	160ms	500kHz
[9]	66 μm	304 x 304	200ms	250kHz
[21]	50 μm	224 x 256	20ms	Not Mentioned

TABLE II
PERFORMANCE OF SEVERAL FINGERPRINT SENSORS.

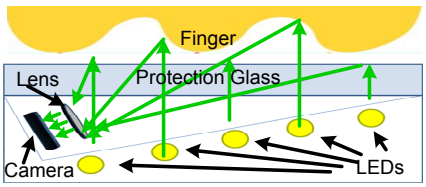

Fig. 3. Optical Based Fingerprint Sensor

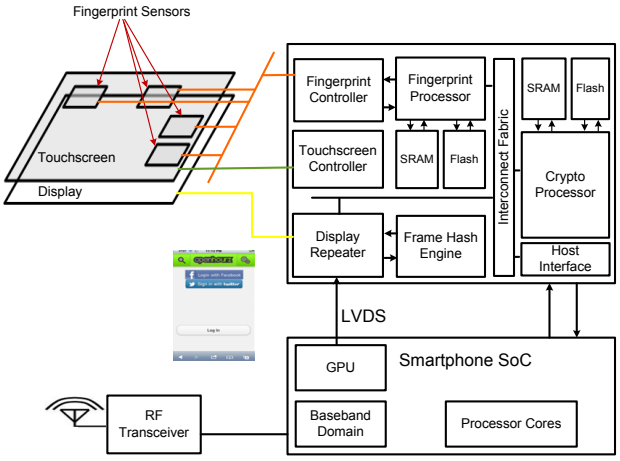

Fig. 5. Hardware Design of FLock

points (taking less than 4ms [4]). The fingerprint sensor around the touch point is activated. In the second step, the fingerprint data is captured by the fingerprint sensors. In order to improve the capture speed, the system applies parallel sensor cell addressing and selective data transfer. For achieving the best trade-off between touch point coverage and cost, one can use a biometric sensor placement approach that chooses the optimal number, places, and sizes of fingerprint sensors. The optimization is based on the observation that during natural usage, touch points made by a mobile user are not evenly distributed and appear more frequently in certain touchscreen regions.

The biometric-touch panel consists of two layers. The touchscreen is at the bottom level and TFT fingerprint sensors are overlaid on the top level by using transparent TFTs [15], (thus look transparent to the user). The placement of TFT fingerprint sensors is optimized. The chance of capturing touch points during user-device interaction is maximized. At the beginning, the touchscreen is in fully powered-on state and fingerprint sensors are idle. The fingerprint sensors are activated after a touch action has been sensed by the touchscreen – the touched region is covered by fingerprint sensors. Such design of opportunistic capture of fingerprint reduces power consumption overhead.

Each fingerprint sensor cell has a unique column and line address. The fingerprint controller translates a touchscreen location (position in touchscreen X-axis and Y-axis) into a pair of fingerprint sensor line and column address. The line address decoder in Figure 4 decodes a line address and send the decoding output to a parallel-in parallel-out shift register. The shift register enables one row of capacitive sensing cells at a time. All the sensing cells in the enabled row are addressed during a clock cycle and disabled after the results of the sensing cells are converted into digital values and fed into the latches that are situated at the end of each column.

For each sensor cell, a comparator compares its voltage output with a reference voltage and stores a binary result into the corresponding latch. The design supports parallel accesses of the sensor cells at an enabled row, therefore maximizing fingerprint scan speed. Pixel results stored in the latches are selected and transmitted to the fingerprint controller. For minimizing the amount of data transferred, the fingerprint controller computes a pair of column addresses at the beginning and the addresses at the end. Only results stored in the latches within the selected columns are transferred to the fingerprint controller. Using parallel addressing and selected data transfer, the fingerprint capture speed can be greatly improved. The described driving and controlling system is compatible with any TFT based capacitive fingerprint sensor that is built using transparent components and transparent TFT electronic fabrication process.

B. The FLock Module

We propose FLock , a novel hardware architecture shown in Figure 5 to support touch based biometric authentication for both local and remote identity management. Each FLock module has a unique build-in (public,private) key pair. The FLock module consists of a fingerprint controller, a touchscreen controller, a display repeater, a frame hash engine, a fingerprint processor, a host interface, on-chip storage devices (SRAM and Flash), and a crypto processor. The FLock module enables biometric based trust. It connects with a mobile SoC processor. A smartphone with an integrated FLock module communicates with remote services through its RF support. It displays hyper-text pages using its graphics core. The display output is relayed by the display repeater of FLock module. The main SoC processor can communicate with the FLock module through the host interface. When a mobile user views a displayed hyper-text page and issues a request to the server through a touch, the touchscreen controller first reads touch point from the touchscreen and sends the location information to the fingerprint controller. The fingerprint controller can capture fingerprint opportunistically by activating the fingerprint sensor at the touch point. Then it controls the fingerprint sensor to capture user fingerprint and sends it to the fingerprint processor. The fingerprint processor can authenticate the user identity by matching the input with the stored biometric templates. Meanwhile, the display repeater can intercept displayed contents and sends them to the frame hash engine. The frame hash engine computes a hash value of the displayed frame. The frame hash can be later sent to the server to ensure that the displayed hyper-text page has not been tampered. The crypto processor is used to generate (public,private) key pairs, as well as to encrypt and decrypt. The FLock module can store (public,private) key pairs generated by the crypto processor, and the corresponding information about user account, web server name, and biometric templates in protected non-volatile storage.

IV. The solution: TRUST

In this section we introduce TRUST (Trust Reinforcement based on Unified Structural Touch-display), an authentication solution that considers of two different scenarios. First, we focus on a local, mobile device centric user authentication for local identity management. Second, TRUST addresses the problem of user-to-web authentication and remote user identity management. In order to address these scenarios. TRUST leverages the hardware design introduced in section III.

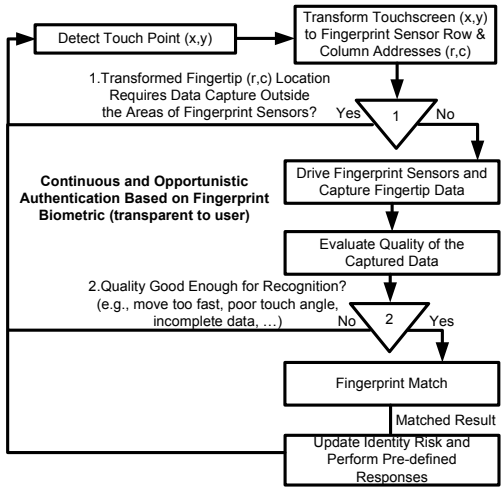

Fig. 6. Continuous and Opportunistic Authentication Based on Fingerprint Biometric

A. Local Identity Security Mechanism

For local identity management, we focus on control of access to local resources and access to information stored in a mobile device. Four assumptions are made in this scenario. First, fingerprint templates are stored in the FLock module to ensure that authentication take place in a protected environment. Biometric verification is applied whenever fingerprints can be captured from normal touch based user-device interactions. Second, we assume that it is not practical to cover an entire touchscreen with transparent fingerprint sensors, due to the cost, performance, and efficiency issues. Third, existing fingerprint match techniques, such as these described in [12], are robust enough to be applied to partial fingerprints. And finally, fingerprint biometric is used not only for unlocking a local mobile device but also for continuously authenticating the user after the one-shot the login process.

Based on the above assumptions, we designed an ambient and opportunistic fingerprint capture solution as shown in Figure 6. The process is the following: When a mobile device is to be unlocked, an unlock button will appear above a fingerprint sensor. The user has to touch the unlock button to unlock the mobile device. Therefore, only an authorized user is allowed to unlock the device. After the device is unlocked, during natural user-device interactions, the touchscreen controller captures each touch made by the user and sends an event to the fingerprint controller. Once the fingerprint controller gets the touch event and its touchscreen coordinates, it calculates the corresponding fingerprint sensor coordinate (x', y') according to each fingerprint sensor's location mapped to the touchscreen space. If the calculated coordination (x', y') is within the data capture range of a fingerprint sensor, the controller enables that specific fingerprint sensor and captures the fingerprint data by

selecting the rows and columns surrounding the touch point. Otherwise, the controller keeps waiting for future touch events. The quality of each captured fingerprint is evaluated before being used for authentication. A variety of reasons may lead to low data quality (e.g. finger moving too fast or pressing with insufficient hardness). Low quality fingerprint data is discarded. If the admitted fingerprint does not match with the stored template, the mobile device can take pre-determined actions as response. These include halting interactions with the user, logging out automatically, etc.

The solution may face two challenges. First, an impostor may try to evade biometric protection by providing only low quality fingerprint data, which will be discarded by the system. The second one is the trade-off between cost of fingerprint sensors and responsiveness of biometric authentication. From both energy consumption and hardware cost aspects, using a large fingerprint sensor to cover the entire touchscreen is not a feasible plan. Additionally, larger size TFT fingerprint sensor will increase the time for capturing a fingerprint due to the increased number of rows and columns. Thus, our strategy is to place multiple small size sensors to achieve optimal performance and reduce overhead.

To tackle the first challenge, we propose three preventive measures. First, a system can display critical buttons or menus over biometric enabled touchscreen regions, that cannot be bypassed by an impostor. Second, for interacting with certain buttons or menus, the system can require a minimal touch time (longer than the required fingerprint capture time). Third, the system can adopt a window based touch authentication mechanism. For instance, at least k out of n consecutive touch inputs need to produce at least one valid fingerprint. Our solution uses identity risk to quantitatively measures the likelihood of identity fraud. Identity risk can be defined as the number of times that fingerprints can be captured and verified out of certain number of touches from a user.

To address the second challenge, we conducted experiments to collect distributions of touches from normal smartphone-user touch interactions, and discover hot-spot touch regions. For example, Figure 7 shows touch distributions of three users collected from an experiment using a HTC smartphone. The touch density results suggest that there are overlaps and hot-spot touch regions among the three users. Based on the results of additional studies, we conclude that it is possible to design a sensor placement solution by analyzing touch distributions and hot-spots so that even limited fingerprint sensor coverage can ensure as many touches to fall within biometric enabled touchscreen regions as possible.

By introducing the novel fingerprint based authentication mechanism, one can achieve three main benefits: (i) The proposed authentication solution based on fingerprint biometric provides stronger access control and identity management exceeding the current approach during the login stage. (ii) Compared with the current practice of no protection in post-login stage, our solution can offer additional security protection even after user login. (iii) The new mechanism is transparent to the user. It incurs neither physical (e.g., extra steps/actions that a user has to take) nor cognitive burden (e.g., remembering a password) on the mobile user, making it more user friendly than the existing approach.

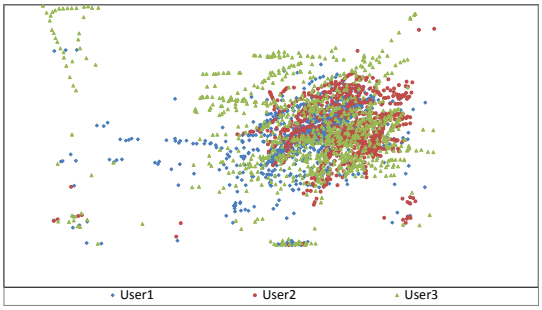

Fig. 7. Distributions of Touches from Three Users

B. Remote Identity Security Mechanism

We now introduce a touchscreen based solution for transparent and efficient remote web service (e.g., bank, e-mail, social networking accounts) authentication. Although the primary focus is the security of accessing web information from a mobile device, by default the local authentication solution (previously described) is considered as well in this scenario.

Assumptions. To deal with the new scenario, five additional assumptions are made: (i) We assume a strong security case that only the FLock module as well as the Web Server are secure, the mobile device software stack and browser executed by the host SoC processor may be monitored or under the control of a remote attacker through malware. The encryptions and authentication steps take place in the FLock module. (ii) The FLock module relies on cookie extensions for exchanging data with a remote server. (iii) The Internet communication between a Web Server and a mobile device is untrusted. Replay and Man-in-the-Middle attacks need to be considered. Furthermore, no fingerprint information should be transferred between the FLock module and the server in cleartext. (iv) An attacker who controls a mobile browser through malware may spoof web connections. For example, an attacker may change the organization of user interface to fool the user to perform wrong operations, or use malicious script to create fake user actions. (v) Devices infiltrated by remote a attacker may leak user passwords: As shown by Miluzzo etc al. [13], with the a 3-axis accelerometer and gyroscope information, touch inputs on the virtual keyboard can be monitored and tracked. This introduce the risk of losing vital password information when the user logs in web sites, such as banks or email box, through the mobile devices.

The primary focus of the proposed solution is to address remote authentication and identity management. Our solution is not designed to prevent denial of service attacks launched by a remote attack through mobile malware, or prevent disclose of information to an attacker through a compromised browser or mobile software stack. Additionally, our solution does not mitigate identity risks when a mobile user is kidnapped and accesses a remote service against his/her will. Using Flock and the proposed approach, a remote server can ensure that, all the requests that it receives, come from an authorized user and the

Fig. 8. Components Involved for Remote Identity Management

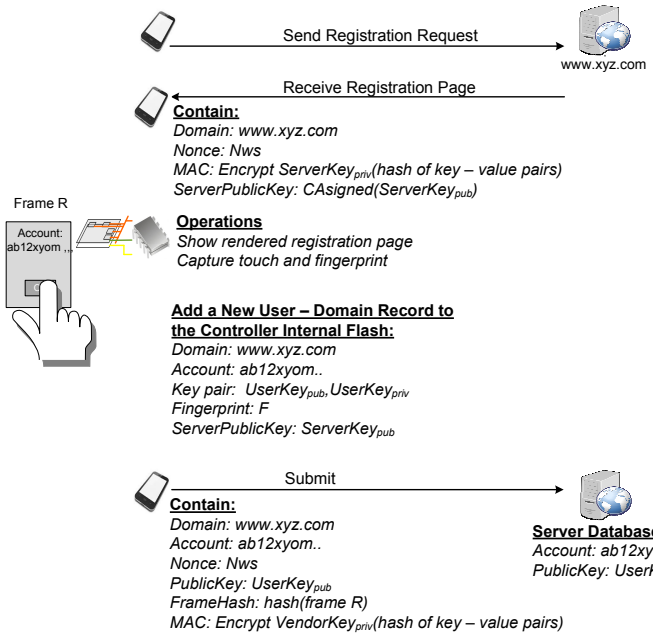

Fig. 9. Process of Registration Using Flock

requests are originated from touch actions from the authorized user. With frame hash support, our solution provides further assurance to the server that the authorized user respond to the true information sent from the server. A remote attacker cannot fool or manipulate a user to gain access to a remote service by showing spoofed information to the user.

Based on the above assumptions, we designed the solution shown in Figure 8. Besides the mobile device and its FLock module, the system consists of a remote web server and a Certificate Authority (CA) server. The public key of the CA is stored on the FLock . We assume that each Web Server and each FLock module of a mobile device, have a public key certificate signed by the CA.

Device-to-user-account binding. During an initial device bind process, the Web Server sends its public key certificate signed by the CA. FLock then verifies the signature to ensure that the public key belongs to the Web Server. We assume that a user can create an account with the Web Server. Let A be the user. The binding process is shown in Figure 9 and is described in the following steps:

- **Step 1.** The user A sends a registration request to the Web Server through a mobile device. The Web Server (shown as www.xyz.com) sends to the mobile device its public key certificate (signed by the CA), the contents of the web page, a cookie comprising a fresh nonce, N_{WS}, and a message authentication code (MAC) – computed over these values and signed by the Web Server (using its private key).
- **Step 2.** FLock verifies the Web Server's public key certificate and verifies the integrity of the transferred information. It then generates a key pair (pk_A, pr_A). The mobile device displays the content of the web page received from the Web Server – including a registration button. When the user touches the registration button, FLock captures the user's fingerprint, processes it as template for fingerprint authentication and stores it along with the Web Server's domain, the user's account on it, the key pair (pk_A, pr_A), and the public key of the Web Server, in protected non-volatile memory. The Display Repeater reads the frame information from the mobile device SoC processor (typically with integrated graphics core) and creates a frame hash (e.g., apply MD5 or SHA-256 to a displayed frame). The frame hash can be sent back to the server as a cookie field. The server can store it to a log file. During future audit event, the log can be investigated to discover how the user interacted with the service.
- **Step 3.** FLock sends its public key certificate, the nonce N_{WS}, A's account id, the Web Server's domain, the frame hash code, A's public key, pk_A, and a MAC, computed over this message and signed with the FLock 's private key, to the mobile device.
- **Step 4.** The mobile device submits to the Web Server the information from FLock . The mobile device can read this information but it can not tamper it without being detected.
- **Step 5.** The Web Server uses the public key of A's device FLock module to verify the MAC against the transferred information to ensure it has not been modified. If the verification succeeds, it stores the account information and A's public key, pk_A, in the Server Database.

Continuous user authentication. Following the initial binding process, whenever user A needs to login to and access the Web Server through the mobile device, the authentication process and continuous identity management takes place as shown in Figure 10, and is described in the following steps:

- **Step 1.** A accesses the Web Server through the mobile device and sends a login request. The Web Server sends to the mobile device the contents of the web page, using a cookie containing a fresh nonce N_{WS1}, and a MAC over these values, encrypted by the Web Server's private key.
- **Step 2.** The mobile device displays the content of the web site by forwarding information from the Web Server to FLock . When the user touches the button of the page's login, FLock captures the user's fingerprint and authenticates it. Meanwhile, Flock reads the frame information and creates a frame hash. FLock sends to the mobile device, the Web Server domain, A's account id, the nonce N_{WS1}, a freshly generated session key encrypted with the

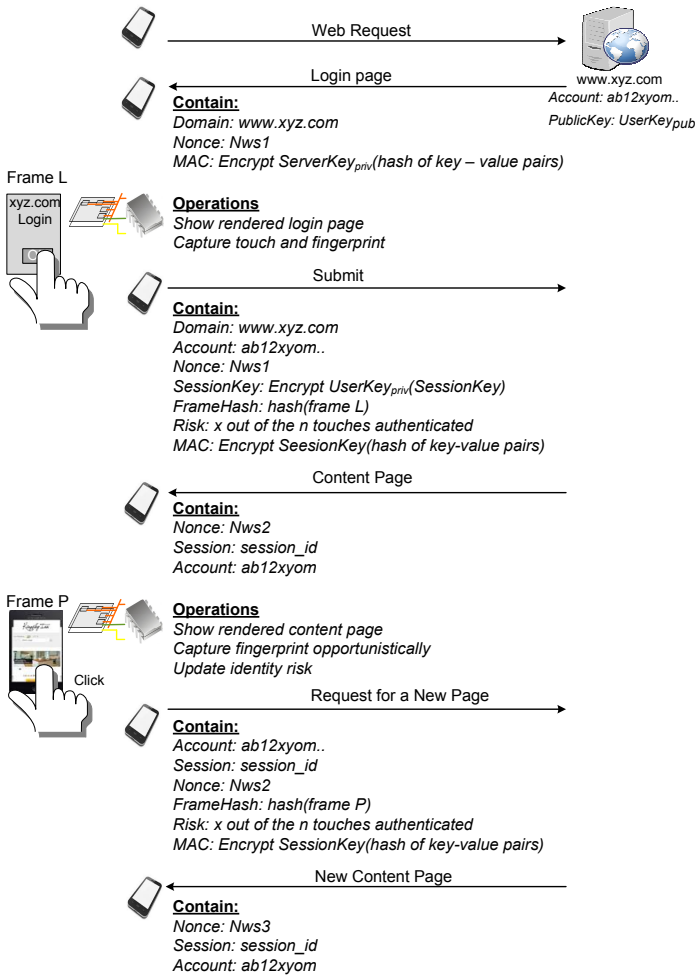

Fig. 10. Process of Continuous Authentication Using Flock

Web Server's public key, the frame hash, the risk factor and a MAC over these fields, encrypted with the session key.

- **Step 3.** The mobile device forwards the information to the server. The Web Server uses its private key to recover the session key. It then uses the session key to extract the MAC and verify the integrity of the message. Then the web server returns a concept page to the mobile device.
- **Step 4.** For each subsequent user-to-Web Server interraction, the above process is repeated, with the difference that the communication is encrypted using the session key.

Security Analysis. For identity management, fingerprint based approach is more secure (has more entropy) than passwords generated by the users. Users no longer have to remember unique passwords for each of their remote site accounts. Because only the generated public key is disclosed to the web servers, TRUST avoids exposure of user's fingerprint. Another benefit of TRUST is that CAPTCHA based mechanisms are no longer needed: the use of real finger touches prove that the user is human. For the post login stage, since fingerprint authentication takes place with touch events, cookie expiration control is no longer needed. Post-login remote hijack attacks to the browser or mobile software stack are handled during touch interaction. Unlike the existing cookie based approach, a web server can constantly verify the identity of a remote user. Based on our solution, the identity management is secure even when a mobile device is under the control of a malware because a server can always verify user operations by checking the frame hash codes sent from TRUST. Because a user can change the displayed view of a web page through touch gestures such as zoom-in or zoom out, the frame hash code of a displayed frame may vary and be different from the hash code computed according to the original hyper-text page. However, displayed view of a web page can only belong to a finite set of all the possible views of the original page. It is feasible to match the corresponding frame hash code against a finite set of all the possible hash codes. To avoid expensive computation, a server can store the returned frame hash code in a log and perform verification during off-line audit process. Further, with the usage of fresh nonce, session keys and risk factors, we can prevent replay attacks.

Identity Reset. When a user loses her mobile device, all her identity information stored in the old mobile device is lost. The user needs then to reset her identity information for all her web services, from a new mobile device. The user can rely on her old passwords in order to login on her web services accounts using her new mobile device. Each web service needs to provide users with the option to reset their identity information. The identity reset service enables the server to remove the user's previous public key binding to the account. The user can then bind her new mobile device to each web service in a manner similar to the registration processes.

Identity Transfer. When a user buys a new mobile device, she has the option to transfer her identity from the old mobile device to the new mobile device. To achieve this, the user connects the old and new mobile devices to a computer, e.g., a desktop. TRUST then creates a channel allowing the two mobile devices to communicate: The user sends an identity transfer request from the new mobile device along with its build-in public key certificate to the old mobile device. Upon receipt of the request, the old mobile device posts an authentication request for the transfer. The user can authorize the operation by verifying her fingerprint. When the authentication process is completed, the old mobile device encrypts – using the new device's public key – all the web service information and the corresponding (public,private) key pairs along with the user's biometric identity, and transfers the resulting information to the new mobile device. The new mobile device decrypts the received information and initializes the data structures maintained for each web service.

V. Related Work

In this section, we survey works that are most related to user identification, especially implicit authentication on mobile devices and authentication using biometrics information.

For local mobile implicit authentication, previous study by Clarke et al. [5] on users' perceptions of authentication in mobile devices, showed that a system that can implicitly and continuously perform user authentication in the background

without disrupting the normal user-mobile device interaction is a desired solution by mobile device users. While in [7], Denning et al. proved text passwords had been known to impose a cognitive burden on users that resulted in selection of weak passwords. Hwang et al. [17] and Maiorana et al. [11] tried to improve the security of mobile devices by performing implicit authentication using the keystroke dynamics. De Luca et al. also proposed an implicit authentication [6]. It still focused on the login mechanism which was not as user-transparent as our solutions. For device-to-web authentication, Kai et al. [25] presented a bio-cryptographic security protocol designed for client-server authentication in mobile computing environment. Also in [18], Shi et al. presented implicit authentication-authenticating users based on behavior patterns. In [19], the authors described a multi-sensor based user authentication scheme that includes touchscreen as one of the continuous user identification modality. Furthermore, in [8], Feng et al. proposed an implicit authentication mechanism applying machine learning algorithms on the features of touch gestures.

In [23], [14], [22], the authors presented an application for fingerprint-based security system, called fingerprint fuzzy vault. The general idea is extracting the keys from the biometric information such as fingerprint and then using this key to encrypt secret information in the database. The security strength of the fuzzy vault is based on the complexity of the polynomial reconstruction issue. However, the fuzzy vault is not suitable in our scenario for two reasons: first, its 10 percent false reject rate will put the users in trouble when employed it in a continuous authentication solution; second, the touch areas of fingers vary each time the user touches the mobile devices, which will make the accuracy of the fingerprint fuzzy vault even lower.

VI. Conclusions

This paper presents a novel unified solution integrating transparent TFT-based fingerprint sensors, touchscreen, and display. Based on the hardware design we then introduced two security and identity management scenarios, one for local identity management, and the second extended solution for remote identity management. The new opportunistic fingerprint based authentication is able to achieve a good balance between security and usability with continuous and transparent user verification during access to a remote service. It maintains totally transparent user interactions, and increases access control protection when compared with the solutions without post-login authentication. By its nature the strength of fingerprint based authentication is far beyond the current mobile authentication systems that do not use fingerprints and is comparable to the current biometric login approaches, while the transparent login design provides for a much better user response.

References

[1] More top worst passwords. http://xato.net/passwords/more-top-worst-passwords#more-269, 2010.

[2] R. Adler and P. Desmares. An economical touch panel using saw absorption. *Ultrasonics, Ferroelectrics and Frequency Control, IEEE Transactions on*, 34(2):195 –201, march 1987.

[3] R. Aguilar and G. Meijer. Fast interface electronics for a resistive touchscreen. In *Sensors, 2002. Proceedings of IEEE*, volume 2, pages 1360 – 1363 vol.2, 2002.

[4] Atmel. Touchscreen Controllers - Parameters. http://www.atmel.com/products/touchsolutions/touchscreens/default.aspx, 2012.

[5] N. L. Clarke and S. M. Furnell. Authenticating mobile phone users using keystroke analysis. *Int. J. Inf. Secur.*, 6:1–14, December 2006.

[6] A. De Luca, A. Hang, F. Brudy, C. Lindner, and H. Hussmann. Touch me once and i know it's you!: implicit authentication based on touch screen patterns. In *CHI '12 Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*, pages 987–996, 2012.

[7] T. Denning, K. Bowers, M. van Dijk, and A. Juels. Exploring implicit memory for painless password recovery. In *CHI '11 Proceedings of the 2011 annual conference on Human factors in computing systems*, pages 2615–2618, New York, NY, 2012.

[8] T. Feng, Z. Liu, K. Kwon, W. Shi, B. Carbunar, and Y. Jiang. Enhancing mobile security with continuous authentication based on touchscreen gestures. In *IEEE Conference on Technologies for Homeland Security 2012 (HST)*, 2012.

[9] H. Hara, M. Sakurai, M. Miyasaka, S. W. B. Tam, S. Inoue, and T. Shimoda. Low temperature polycrystalline silicon 'tft' fingerprint sensor with integrated comparator circuit. In *Solid-State Circuits Conference - ESSCIRC 2004*, pages 403–406, 2004.

[10] R. Hashido, A. Suzuki, A. Iwata, T. Okamoto, Y. Satoh, and M. Inoue. A capacitive fingerprint sensor chip using low-temperature poly-si tfts on a glass substrate and a novel and unique sensing method. *IEEE Journal of Solid-State Circuits*, 38, 2003.

[11] E. Maiorana, P. Campisi, N. González-Carballo, and A. Neri. Keystroke dynamics authentication for mobile phones. In *Proceedings of the 2011 ACM Symposium on Applied Computing*, pages 21–26, 2011.

[12] S. Malathi. Improved partial fingerprint matching based on score level fusion using pore and sift features. In *Process Automation, Control and Computing (PACC), 2011 International Conference*, pages 1–4, 2011.

[13] E. Miluzzo, A. Varshavsky, S. Balakrishnan, and R. Choudhury. Tapprints: your finger taps have fingerprints. In *MobiSys '12 Proceedings of the 10th international conference on Mobile systems, applications, and services*, pages 323–336, 2012.

[14] K. N, S. Member, A. K. Jain, S. Pankanti, and S. Member. Fingerprint-based fuzzy vault: Implementation and performance, 2007.

[15] K. Nomura, H. Ohta, A. Takagi, T. Kamiya, M. Hirano, and H. Hosono. Room-temperature fabrication of transparent flexible thin-film transistors using amorphous oxide semiconductors. *Nature*, 432(7016):488–492, 2004.

[16] J.-Y. Ruan, P.-P. Chao, and W.-P. Chen. A multi-touch interface circuit for a large-sized capacitive touch panel. In *Sensors, 2010 IEEE*, pages 309 –314, nov. 2010.

[17] S. seob Hwang, S. Cho, and S. Park. Keystroke dynamics-based authentication for mobile devices. *Computers & Security*, 28(1C2):85 – 93, 2009.

[18] E. Shi, Y. Niu, M. Jakobsson, and R. Chow. Implicit authentication through learning user behavior. In M. Burmester, G. Tsudik, S. Magliveras, and I. Ilic, editors, *Information Security*, volume 6531 of *Lecture Notes in Computer Science*, pages 99–113. Springer, 2011.

[19] W. Shi, J. Yang, Y. Jiang, F. Yang, and Y. Xiong. Senguard: Passive user identification on smartphones using multiple sensors. In *IEEE 7th International Conference on Wireless and Mobile Computing, Networking and Communications*, pages 141–148, 2011.

[20] S. Shigematsu, H. Morimura, Y. Tanabe, T. Adachi, and K. Machida. A single-chip fingerprint sensor and identifier. *J. Solid-State Circuits*, 34(12):1852–1857, 1999.

[21] T. Shimamura, H. Morimura, S. Shigematsu, M. Nakanishi, and K. Machida. Capacitive-sensing circuit technique for image quality improvement on fingerprint sensor lsis. *J. Solid-State Circuits*, 45(5):1080–1087, 2010.

[22] U. Uludag. Securing fingerprint template: fuzzy vault with helper data. In *in Proceedings of CVPR Workshop on Privacy Research In Vision*, page 163, 2006.

[23] U. Uludag, S. Pankanti, and A. K. Jain. Fuzzy vault for fingerprints. In *in Proc. AVBPA, Lecture Notes in Computer Science 3546*, pages 310–319. Springer, 2005.

[24] J. woo Lee, S. Member, D. jin Min, J. Kim, and W. Kim. A 600-dpi capacitive fingerprint sensor chip and image-synthesis technique. *IEEE Journal of Solid-State Circuits*, 34:469–475, 1999.

[25] K. Xi, T. Ahmad, F. Han, and J. Hu. A fingerprint based bio-cryptographic security protocol designed for client/server authentication in mobile computing environment. *Security and Communication Networks*, 4(5):487–499, 2011.