An alternative approach and attempt to come up with a standard for biometric user authentication in a network based environment

Konstantin Stoychev Tsvetkov\textsuperscript{a}, Teodor Georgiev \textsuperscript{b}

\textsuperscript{a} Konstantin Stoychev Tsvetkov, 115, Universitetska Str., Shumen 9712, Bulgaria
\textsuperscript{b} Teodor Georgiev, 22 Tutolmin Str., Lovech 5500, Bulgaria

Abstract

People who make their first steps into the information technologies are much more vulnerable to various security threats than experienced individuals. They don't know how to protect their passwords, they can't pick up complex passwords, and they cannot type quickly their passwords. During their education they can become a victim of hostile individuals. Their education process can be hindered when it comes to password authentication. Children who are still illiterate cannot take full advantage of all online resources that offer innovative education content. The same matters for adult people with low level of literacy. With some basic reading skills and with the latest touch-screen technologies they can be easily integrated into the IT world. When it comes to writing (typing passwords) however, things can become different for them.

Keywords: fingerprint, biometric authentication, fingerprint hardware vendor, “two-fingers” authentication technique.

1. Introduction

For more than 20 years people are using fingerprint biometric authentication in the IT. It is widely used nowadays, mostly in access-control systems, but also is implemented in some personal equipment like PC USB fingerprint readers, mobile phones with built-in fingerprint readers and so on. Recently fingerprint authentication has found its use in cars and ATMs. Still however we can't authenticate with our fingers on any website, Skype, Google, ssh / telnet service.

In no means we could say that fingerprint authentication is much more secure compared to the conventional typed-password method. It offers however several other obvious advantages that make it one really tempting alternative:

- It is much easier to remember a combination 1 or 2 out of 10 (1 or 2 fingers out of 5 for each hand = 10 in total), rather than a complex secure password that contains small and capital letters + digits. So we won't have to write down our passwords, as most of do now, neither we will have to use that “Password recovery” feature than often. Our fingers are always with us.
- It is faster to swipe a finger or two, rather than to enter a complex keyboard password.
- One does not have to bother typing his password quickly and be afraid that someone is sneaking by his back.
- It is extremely difficult to brute-force (not to mention – guess) a fingerprint password.
- It gives an opportunity for pre-school kids, people with low level of literacy (or certain disabilities) to identify / authenticate and have access to network services.
- The lack of keyboard for authentication allows minimizing the size of embedded devices, public-access systems and in general – to use online services that require authentication merely by a small pointing device with integrated fingerprint reader.
Dr. Manfred Bromba (a German engineer) has created an excellent FAQ that covers in depth all major topics related to fingerprint authentication. It can be found at that URL [1]. Another FAQ about fingerprint authentication is available here [2].

2. Goal achievement

The overall process of implementing (and then mass-adopting) a specification for fingerprint authentication in a TCP/IP network can be separated into three main parts/steps:

2.1. Customer side - That part covers the fingerprint reader hardware, operating system drivers for it, fingerprint data acquisition and intercepting fingerprint data and passing it to the application that requires authentication.

USB fingerprint readers can be nowadays bought in different configurations, usually as standalone devices, but also integrated into other peripheral units - most frequently into laptops, keyboards and mice. The price for an end-user device varies between $30 and $70, which is acceptable. In order to facilitate the widespread use of fingerprint readers on every desktop computer system, hardware vendors should be encouraged to integrate fingerprint readers into PC mice and keyboards. Having yet another device + a cable for it is slightly irritating. Fingerprint readers should be implemented in any laptop, smart-phone or tablet device.

The fingerprint hardware vendors should finally come up with a standard - a low level protocol for acquiring raw images from the fingerprint reader. Right now each vendor develops and offers his own driver and API/SDK. Several solid steps are made however, to overcome the situation. Several libraries offer a unified software interface for interacting with the most widespread devices and more vendors are added to their supported list on a frequent basis. Some of them are:

✔ Libfprint - It is a mature open source (GPL license) library, written in C for use on Linux/UNIX based systems [3]. There is a full-featured Linux graphical application for fingerprint authentication, based on libfprint [4].

✔ Griaule biometrics - This multiplatform biometrics SDK, though commercial, is widely used due to its portability and easiness. It supports a wide range of fingerprint devices and has as well ActiveX support. Flawless support for several programming languages [5].

✔ Silex - It is a commercial library, known by its performance, reliability and optimization. High level of optimization is an authentication data block of 120-360 bytes. Supports both dry and wet fingers. The SDK offers Windows only support [6].

Middle-ware applications (either built-in into the drivers, a web browser plug-in or another application) that will intercept the authentication data entered through the fingerprint reader and pass it to web page / application (like Skype or the World of Warcraft login screen) that awaits authentication to be performed by the user.

2.2. Transit - That part covers the process of transporting the collected fingerprint biometric authentication data from the client-side entity to the server-side entity where authentication against a “database” of biometric passwords is performed. The details of that part are described in chapter “Algorithm implementation”.

2.3. Server side - Our experience and observations show that ~90% of the authentications of an average Internet user are performed against web based services (a regular HTML pass). The rest 10% goes to network-based applications and protocols like Skype (and other instant messengers), ssh (for Linux and UNIX based systems remote access), RDP (Microsoft Windows remote access), various entertainment applications (game client interfaces). Therefore, in order for the newly created protocol to gain popularity, first it has to be implemented to support authentication for web-based services.

There are several possible ways to collect a biometric password at the web page:

✔ A proposal should be applied to W3C to implement an input type field for biometric passwords in the new HTML5 standard. Something like <INPUT TYPE=B-PASSWORDNAME="my password">. In that way a web page that requires authentication will have two alternative/self-excluding password input forms -- a regular one and a biometric one. It
will be up to the user to decide which one he desires to use in that specific case. This implementation however is doubtful to succeed, because W3C has still not come to a decision for implementing PKI authentication support in HTML5. A proof for this can be found in these two discussions at the IETF forum [7, 8].

✓ A custom AJAX method that will be able to switch between regular password-based authentication and biometric authentication. It shall collect the biometric password (if that authentication type has been chosen by the user) and pass it to the web back-end application.

✓ The simplest way is to transmit the fingerprint authentication data as a regular string data into the existing password input form. That can be controlled by the biometric middle-ware application (like the above mentioned browser plug-in). How the entered biometric password will be handled at the server side is up to the specific web server and web application. Authentication modules should be written for the most popular web servers (Apache, IIS, ng-inx) that allow matching a biometric password against an SQL/LDAP database.

It should be worth mentioning that there are already several attempts to offer such services. Eye Net Watch is an British company that offers a "proxy" web server (called BIO Web-Server) for fingerprint authentication. Here is their product description webpage [9].

3. Algorithm implementation

The presented here algorithm relies on several techniques that offer improved security, robustness and flexibility compared to what is offered currently as fingerprint authentication in the IT.

3.1. It authenticates by the fingerprints of two fingers, instead of one. It is not unknown that an ill-wisher could obtain and make a copy of someone's finger print with some skills. At this URL can be seen a video demonstration by Chaos Computer Club of stealing fingerprint data [10].

The “two-finger” authentication technique will increase dramatically the difficulty to “copy” and forge a fingerprint. Our 10 hand-fingers produce 10 different fingerprints. Stealing fingerprint from two different hands is tremendously difficult.

3.2. Unlike some other widely used fingerprint authentication techniques, this one does NOT store and operate with an image bitmap file of a fingerprint. Instead, it obtains the fingerprint image from the sensor, and then it extracts and optimizes the minutiae data out of the image, without saving/ storing the fingerprint image anywhere. In that way it is impossible to reconstruct back an image of the fingerprint.

The size of the extracted and optimized minutiae data shall vary in a range of 180 to 420bytes for one finger and all that is pretty enough to perform flawless fingerprint authentication. That data could be even further squeezed via several possible methods:

✓ Compression algorithm - one nice possibility is Smaz- a lightweight compression library registered under the BSD license that is suitable for compressing very short strings. It can be found at[11].

✓ MD5 (or another) hash – applying an MD5 hashing against the data will transform it into a 16 byte hash.

✓ PBKDF2 – key derivation function for public-key cryptography. Widely used nowadays in technologies like wireless WPA/WPA2, Microsoft Windows data protection, WinZip AES encryption technique and so on. More information about PBKDF2 can be found at[12].

3.3. In order to avoid capturing authentication data, fingerprint biometric authentication process will be always refused either by the client or server side if there is no encryption in the path – SSL in example.

The entire processin general can be described in the diagram shown below [Fig.1]:

4. Conclusions

Our daily life depends more and more on the information technologies. The various online services we use increases with each single day. Our children who make their first social steps into this world are compelled to gain vast IT skills quite early. The above described “feature”, if implemented, will make it possible to give the new generation the possibility to start their IT education quite earlier than usual. Elderly people and even these with some mental disabilities will be able to use the Internet quite more efficiently. In general – that feature will make life much easier for simply everyone.

References

http://www.bromba.com/faq/fpfaqe.htm
http://www.ask.com/faqcentral/BIOMETRIC_PASSWORD.html#5
http://www.freedesktop.org/wiki/Software/fprint/libfprint
http://www.n-view.net/Appliance/fingerprint/
http://www.ietf.org/mail-archive/web/pkix/current/msg01682.html
http://www.imc.org/ietf-pkix/mail-archive/msg05521.html
http://www.eyenetwatch.com/Biowebserver/fingerprint_authentication.htm
http://www.youtube.com/watch?v=aGjGzqYgz0c
https://github.com/antirez/smaz

http://www.ietf.org/rfc/rfc2898.txtc


The article was designed РД -07-304/14.03.2011.of Konstantin Preslavsky University of Shumen