A Secure Fingerprint Authentication Protocol

Anongporn Salaiwarakul
Department of Computer Science and Information Technology, Naresuan University, Thailand.
anongporns@nu.ac.th

***Abstract*—This article proposes authentication specifications and a framework for the fingerprint authentication in the circumstance that the presentation of the user's biometric information is not supervised. The specifications of the security properties are to certify that the liveness of the user's fingerprint information is confirmed and that the intention of the user's authentication is not manipulative or illegal. The framework for compliance with the specification of the fingerprint authentication protocol is proposed. Liveness detection by the fingerprint reader is considered to be essential in these situations. Cryptography and the fresh random number, nonce, are included in the framework. Analysis of the authentication framework shows that the proposed security properties are confirmed, the user's biometric data is secured and the user's intention of authentication is preserved.**

***Index Terms*—Biometric; Fingerprint Authentication Protocol; Fingerprint Authentication Specification; Security Protocol.**

I. INTRODUCTION

Biometric authentication uses behavior characteristics of the user and their physical traits to identify the user's identity. Typing patterns or signature styles are examples of user's characteristics that can be used to determine the identity of a user. However, the difficulty of distinguishing a user's characteristics is complicated as the user may present their style differently at different times. A user's physical traits, such as their iris, hand geometry, or fingerprints allow easier recognition each time the user presents them for verification, this being the reason that user physical characteristics are widely used for user authentication.

Biometric authentication in supervised situations is secure and identifiable since the user's biometrics is difficult to steal or fraudulently use without the user being aware, thereby ensuring that the user's identity is correctly determined. In the supervised situation it is implied that the biometric authentication process is monitored by the system's verifier. This prevents the user's biometric data bypassing the authentication device, such as biometric reader. The user's biometric data itself can be used to actually authenticate the user's identity. The security of the biometric data must be of paramount concern in terms of privacy protection and information security.

While the strength of using biometric data as a means of authenticating the user is now well understood, using biometric authentication in unsupervised situations must still be considered as a weakness. Unsupervised biometric authentication situations include, for example, online transactions which include authentication of the user by using that user's biometric data.

During unsupervised biometric authentication, the verifier has no means to guarantee the user's identity and/or their liveness. Liveness in the user's biometric authentication refers to the situation where the user's biometrics are not being reused or replayed from previous transactions. Liveness therefore requires new or 'fresh' biometrics availability, thus confirming that it is the real, correct, user attempting to authenticate themselves to the biometric authentication system on this particular occasion.

An approach to protecting the user's privacy in biometric authentication was proposed by [1], in which the user's identity is hidden during the biometric authentication process. However, this approach cannot be used in the situation that the user's identity must be presented such as in on-line banking transactions where the user's ID links to the account for verification purposes, so must be known.

Research published in [2] proposed an online bank transaction processing approach using biometric authentication which establishes a secure transaction between the online banking software running on the client computer and on the banking server computer. The research proposed that the biometric device at the client end should establish the secure connection to the server; this however is not supported by current technology. Also, the important aspect of the liveness of the user presentation is still questionable and needs further consideration.

Strong protection of the user's biometric data in remote biometric authentication situations is illustrated in [3]. Here, token storage of biometric data is required to secure the biometric data. Hence, it becomes an imperative to manage the token and token reader.

Given the variety of options, and perceived shortcomings of the various approaches discussed so far, our research presents specifications to which the fingerprint authentication protocol should comply, and a secure compliance framework for fingerprint authentication when the fingerprint data is used in the unsupervised situation and should be analysed to confirm its security properties.

This situation is discussed in the two dimensions of the fingerprint authentication framework. First, the background information and knowledge necessary to support the process are presented. Subsequently, the discussion is illustrating the importance of verifying the security properties of the protocol and/or framework. The tools available to be used to analyse the security protocol are shown and discussed. Our conclusions, in part, summarize the contribution of the research and the advantages of the proposed framework.

II. RESEARCH BACKGROUND

A. Biometric Authentication

Biometric Authentication Process: There are two main processes in authentication using the user's biometrics as a means of identification: the enrollment process and the biometric data verification process. For the enrollment process, the user is required to enroll, or register, their

biometric code in the system. This data is stored securely in the system's repository for later use. Once the user has been asked for their biometric authentication action, their biometric data is presented to the system and verified against the stored biometric code. This is the verification process. The result of the biometric verification process will either be a match or a non-match [4,5]. After a match, the system will allow the user to proceed with their request based upon the successful matching result.

Biometric Authentication Device: In a biometric authentication system, the system involves a biometric reader, the user interface, and storage. The biometric reader is a device to acquire the user's biometric data. The reader and the related software will transform the data into a form that can be stored in the repository or it can be matched against the stored biometric code in the verification process. The user interface will connect the user with the biometric system. This will facilitate the user's connection with the system or with the biometric reader. The biometric data can be stored in the user's token (smart card, for example) or any other database. The user's biometric code can be stored securely in the smart card which the user can carry with them. If the user is required to carry out the biometric user authentication, the user' stored biometric code is read from the smart card and will be matched in the matching process against the user's presented biometric data.

In the networking scenario, the user's biometric data can be stored in the system's database which may be remotely hosted, or hosted in the Cloud. The stored biometric code will be transferred to the matching stage so that the user can be authenticated via the public workstation.

B. Security Properties Verification

The security protocol and/or framework is highly error-prone. Many proposed and well-known security protocols have been found to have weaknesses and to be prone to attacks [6, 7]. An unproven security protocol is a serious threat when it is used for communicating data which needs to be secure and protected.

Verifying the security framework will ensure the soundness of the security properties of the proposed model and ensure that it provides sufficient security properties, according to its claims.

Traditional techniques to verify the security framework, using formal verification methods which transform the processes in the framework into finite states, is error-prone due to the difficulty of correctly and completely identifying flaws in the finite state model. Automated verification, analysis and detection techniques based on programmed rules or symbolic models, are likely to be significantly more successful than human scrutiny. Programmed tools currently available include AVISPA [8], ProVerif [9], or Scyther [10].

III. UNSUPERVISED FINGERPRINT AUTHENTICATION SPECIFICATIONS AND PROTOCOL

A. The Specifications

In order to guarantee the security of the biometric data used in the authentication protocol, the security properties of the protocol should be clarified and, together with the operational parameters of the protocol, should be clearly stated. The protocol should preserve the privacy and the secrecy of the biometric data as well as ensure the authenticity of the user.

The security and verification properties of the device must guarantee that the protocol can not be bypassed in the user's biometric authentication process. Bypassing can be done during user's biometric data presentation or at the time of the matching process, and the result of the matching can be replayed. For example, bypassing the authentication process can be done during presentation of the user's biometric data by using a fake rubber fingerprint [11, 12]. The biometric device must be able to detect the difference between a fake fingerprint and a natural fingerprint of a living person. At the time that the presented biometric data is being matched with the stored biometric code in the matching process, the stolen genuine biometric data could be replaced by the intruder, who can then keep and subsequently replay that data. The security specifications of the protocol must contemplate and prevent all such possibilities for fraudulent access.

Under unsupervised fingerprint authentication, the fingerprint reader is one of the key devices that an intruder can intercept (other devices include the controlling computer and the network connecting the data reader with the computer). A malicious fingerprint reader can capture the legitimate user's biometric data and later use it as the intruder's own data. To secure against such an intrusion, the fingerprint reader must be verified as secure against software intrusion, such as placing of a Trojan or other manipulative software, before the user's biometric data detection takes place.

Moreover, in an unsupervised fingerprint authentication, an intruder can place the captured biometric data of the legitimate user on the fingerprint reader without the authentication administrator being aware. A scenario might be an intruder captures the user's biometric data e.g. a fingerprint a user has left on a public computer or glass from which the intruder can generate a rubber finger with the fingerprint of the legitimate user. A test of this scenario showed successfully biometric authentication [11]. Liveness detection technology in the fingerprint reader, as proposed in our specification of the protocol to prevent success authentication of fake fingerprint data, would ideally detect the temperature or blood pulse of the biometric implement physically placed on the reader. A fake rubber finger, as in the scenario described, would probably be immediately detected and access disallowed. To achieve this, the liveness detection flag is included in the proposed framework (detail is illustrated in section IV) to indicate that this is live data. Hence, even where the legitimate user fingerprint data is captured by an intruder, it cannot be successfully authenticated when compared against the legitimate data own in the proposed biometric authentication protocol.

In a flawed security protocol, an attacker can replace his biometric data with the legitimate data as it is being transmitted via the network. He can intercept messages sent between the fingerprint reader device which is reading his own biometric data, and the authentication matching server. Then the attacker can insert the biometric data of the legitimate user into the protocol in the biometric matching process. It is therefore essential to secure the protocol by introducing encryption methods and nonces.

A positive matching result of the previous user's authentication can be replayed in a defect authentication protocol. A replay attack of the message transmitted can be solved using nonces as well as the user's identity which should be appended to the matching result. Hence, the recipient of the matching result can verify the freshness of the received message.

B. The Secure Fingerprint Authentication Protocol

To preserve the security properties of the protocol, all components involved in the protocol should be considered secure. These include the fingerprint reader, matching server, and all communication channels. As mentioned, a fingerprint reader in an unsupervised environment can be tampered with by an intruder who can insert malicious software and capture the legitimate user's biometric data. Therefore, the fingerprint reader should be able to be proven secure before the user enters their biometric data. A Trusted Platform Module (TPM) can be applied to guarantee the device as tamper-proof [13]. The TPM is applied in the protocol to confirm the integrity of the components involved in the system. Upon the system boot-up, the TPM is responsible to carry out the integrity checking of the platform and all peripheral devices, which includes the attached fingerprint reader if biometric authentication is practiced. This check will verify that the system configuration has, or has not, changed from the previous configuration. If the system has been manipulated by an attacker, that value state will have changed. The user who is about to place their fingerprint on the detection device can be certain that they can trust this system. Hence, the TPM included in the proposed protocol protects the biometric data from being stolen or captured and subsequently misused.

The communication channels are easily interfered with by an attacker. A powerful attacker such as Dolev-Yao style adversary can play with messages. Therefore, to secure the protocol, the messages should be encrypted and the recipient's identity should be included in the messages.

```
I.   USER -> FINGERPRINT READER
     FP, REQUEST
II.  FINGERPRINT READER -> MATCHING SERVER
     {{LD}skBR, {FP}skTPM, ID, n1, TPM, request}pkMS
III. MATCHING SERVER -> SERVICE
     {{result}skMS, ID, request, n2}pkService
```

Figure 1: Communication messages for a secure fingerprint protocol

Figure 1 shows the communication messages sequence for the fingerprint authentication protocol. The scenario of the protocol is that the client wishes to request service and is required to authenticate himself by entering his fingerprint data. The user is requested to place his fingerprint (FP) on fingerprint reader. The user trusts the fingerprint reader he is using by verifying the TPM states. The fingerprint reader has liveness detection so that the live presentation of the user's biometric data is guaranteed; the matching server can verify this before proceeding to the matching process. Upon receiving the verified data, the matching server deciphers the message. It validates the liveness detection (LD) 'flag' from the fingerprint reader via the fingerprint reader's signature. The origin of the fingerprint data is also checked to ensure it was sent from a trusted TPM. The biometric data is matched with the stored biometric code of the user's identity (ID). The freshness of the message is proved by nonce n1. To prevent the replay attack where the matching result message is sent from the matching server (MS) to the service, the nonce n2 is generated. The matching result is signed with the signature of the matching server. The user's identity and the user's request are also appended to the message to confirm the purpose of the user's biometric authentication. The messages are encrypted by public key cryptography to secure their confidentially.

IV. THE PROTOCOL EVALUATION

Evaluating the security protocol is crucial. The result of the evaluation can confirm the properties that the secure protocol promises to the user.

In order to evaluate the correctness and security properties of the proposed protocol, ProVerif is used as an evaluation tool. ProVerif is an automatic protocol verifier for analyzing the security properties of the protocol. When analyzing the protocol, ProVerif provides a Dolev-Yao Style attacker in order to detect flaws or any violations of the security properties in the protocol. A Dolev-Yao Style attacker is named after the original proposers of the idea that an intruder who is likely to attack the security protocol would have the capability to manipulate messages communicating in the network. A Dolev-Yao style attacker can listen, intercept and replay the messages [14]. Analysing the security protocol using the verifier that establishes this style of adversary will guarantee more security measures in the protocol. The ProVerif model of a protocol is assembled in Applied Pi calculus form.

The ProVerif model is assembled in the same way that messages are communicated between components. Messages being sent in a public channel are the reason to verify whether an attacker could acquire the transmitted data. The components are modeled as processes in ProVerif, and each of which is responsible for both creating and obtaining messages. The process verifies the received message to check whether it is the original and was sent from the claimed participant. The detail data in the message, such as encryption, nonces and signature, are verified to ensure that the messages are not inserted by an intruder.

A. ProVerif Model

There are four processes involved in the ProVerif model corresponding to the proposed protocol: client process, fingerprintReader process, matchingServer process and service process, each of which corresponds to activities of the client, fingerprint reader, matching server, and requesting service respectively.

The user process generates the user's fingerprint data and the user's request and sends this out to the public channel, Ch. The private key of the fingerprint reader is received securely via the private channel, privCh, which represents that it is distributed from the trusted key distributor. It also expects to receive the message sent from the user which includes the user's biometric data. In order to demonstrate that the biometric reader is reading the live presentation of the biometric data, the LD is generated. The authenticity of the user is guaranteed via the user's ID. Nonce n1 is included to confirm the freshness of the message. The whole message is encrypted by the public key of the matching server and transmitted via the public channel. In this scenario, the matching server is responsible for verifying the stored biometric code against the presented biometric data and revealing the matching result. The matching server process receives its private key securely from the key distributor. It also looks for the received message which includes the biometric data read from the fingerprint reader. Once the message is received, it is deciphered. The matching result is presumably generated and encrypted with the matching server's public key. The service process receives the message from the public channel and deciphers the message. The main process is responsible for generating the keys and distributing

them to other processes as well as running all processes concurrently. The ProVerif model of the protocol is shown in Figure 2.

When all processes run concurrently, ProVerif will generate a Dolev-Yao adversary according to the designed model. In the analysis part of ProVerif, the verifier will ask if an intruder could reach the secret that the protocol wishes to keep.

```
let user =
   new FP
   new request
   out(Ch,(FP,request))

let fingerprintReader =
   in(privChFR,skFR)
   in(Ch,m1)
   new LD
   new ID
   new TPM
   new n1
   out(Ch,enc((sign((LD),skFR),
       sign((FP),skTPM),ID,n1,TPM,request),pkMS))

let matchingServer =
   in(privChMS,skMS)
   in(Ch,m2)
  let(m3,m4,IDx,nx1,=TPM,request)=dec(m2,skMS) in
   out(Ch,enc((sign(result,skMS),IDx,request,n2),
       pkService))

let service =
   in(privChS,skService)
   in(Ch,m5)
  let(resultReceived,ID,request,nx2) =
      dec(m5,skService) in

process
   new skFR
   new skMS
   new skService
   let pkFR = pk(skFR) in
   let pkMS = pk(skMS) in
   let pkService = pk(skService) in
   out(Ch,pkFR)
   out(Ch,pkMS)
   out(Ch,pkService)
   !(user) || !(fingerprintReader) || !(service)
     || !(matchingServer)
```

Figure 2: ProVerif model for the communication messages

B. Analysis

Analysing the security properties of the proposed protocol is essential. The intended security properties of the protocol are evaluated to guarantee the proposed security features. The proposed protocol is analysed using ProVerif which analyses attacks on the proposed protocol based on Dolev-Yao style adversary and the defined attack is handled in the query attacker command (as illustrated below). Based on the designed protocol, the analysis part is intended to interpret the secrecy property and replay attack property.

To verify the secrecy of the biometric data, *query attacker: FP* is analysed. Here, FP represents fingerprint data in the authentication process. The consequence of this analysis shows that an attacker cannot reach FP. Hence, the user's fingerprint data is kept secret in the security protocol.

To be able to verify whether an intruder presenting his biometric data to the system is actually inserting the captured positive matching result of the previous transaction of the legitimate user, leads to success in the authentication process in the protocol. The *query attacker : result* is illustrated to verify if an intruder could be able to obtain the matching result and replay it as if they are the legitimate users. A positive result from the analysis shows that a replay attack of the matching result cannot be successful.

V. Discussion and Conclusion

We propose a fingerprint authentication protocol in the unsupervised situation. The objective of the protocol is to protect the secrecy of the authentic user's biometric data; fingerprint data, ensuring that the user's purpose is legitimate, and authentication should be assured. The Trusted Platform Module is introduced to protect the system from being tampered with by an intruder. The liveness detection of the fingerprint reader guarantees that the fingerprint data presented to the protocol comes from a live presenter. Such a configuration overcomes one of the major complications in unsupervised fingerprint authentication. The transmission of the data is secured using public key cryptography.

Additionally, the protocol confirms that the positive biometric authentication matching result can not be successfully replayed and used by an intruder, thus guaranteeing the authenticity of the data, as being original from a live source.

The advantage of the proposed protocol over previous solutions of the biometric authentication protocol is in the way that the proposed protocol can detect and guarantee the liveness of the biometric data being used in the authentication process. Previous attempts to preserve the privacy of the user could not be applied in situations where the identity of the user must be presented, such as in remote bank transaction. The proposed protocol confirms the authenticity of the user and provides protection against authentication attempts by an impostor.

References

[1] M. Barbosa, et al., "Secure biometric authentication with improved accuracy" in *ACISP*, New York:Springer, vol. 5107, pp. 21-36, 2008.

[2] D. Hartung, C. Busch, "Biometric Transaction Authentication Protocol, in *Proc. of int. Conf. on Emerging Security Information, Systems, and Technologies*, 2010, pp. 207–215.

[3] E. Syta,et al., "Private Eyes: Secure Remote Biometric Authentication," in *Proc. 12th Int. Joint Conf. on e-Business and Telecommunications (ICETE)*, Colmar, France 2015, pp. 243-250.

[4] A. K. Jain, et al., "Biometrics: A grand challenge," in *Proc. 17th Int. Conf. on Pattern recognition*, Cambridge, UK, 2004, pp. 935–942.

[5] S.M. Mudholkar, P.M. Shende, M.V. Sarode, "Biometrics authentication technique for intrusion detection systems using fingerprint recognition," *Int. J. Computer Science, Engineering and Information Technology*, vol.2, no.1, pp. 57-65, 2012.

[6] G. Lowe, "Breaking and fixing the Needham-Schroeder public-key protocol using FDR," in *Proc. 2nd Int. Workshop on Tools and Algorithms for the Construction and Analysis of Systems*, London, UK, 1996, pp. 147–166.

[7] G. Lowe, "Towards a completeness result for model checking of security protocols," *J. Computer Security*, vol. 7, no.2-3, pp.89-146, 1999.

[8] A. Armando, et al., "The AVISPA tool for the automated validation of Internet security protocols and applications," in *Proc. 17th Int. Conf. Computer Aided Verification*, Scotland, UK, 2005, pp.281–285.

[9] B. Blanchet, B. Smyth, "ProVerif 1.93: Automatic cryptographic protocol verifier, user manual and tutorial," [Internet] [cited June 2016], Available from : https://www.bensmyth.com/publications/2010-ProVerif-manual-version-1.93/.

[10] C. Cremers, "The Scyther tool", [Internet] [cited June 2016], Available from : https://www.cs.ox.ac.uk/ people/cas.cremers/scyther/.

[11] T. Matsumoto, H. Matsumoto, K. Yamada, S. Hoshino, "Impact of Artificial Gummy Fingers on Fingerprint Systems," in *Proc. SPIE Vol.4677. Optical Security and Counterfeit Deterrence Techniques IV*, CA, USA, 2002, pp. 1-18.

[12] A. Ross, A.K. Jain, "Biometrics : When Identity Matters," in *Advance in biometric person authentication*, 1st ed., Guangzhou: Springer Berlin Heidelberg, 2004, pp. 1-2.

[13] Tusted Computing Group. TPM main specification [Internet] [cited June 2016], Available from : http://www.trustedcomputinggroup.org/tpm-main-specification/.

[14] D. Dolev, A.C. Yao, "On the Security of Public Key Protocols," *IEEE Trans.Information Theory*, vol. 29, no.2, pp. 198-208, 1983.