A Simple Auditable Fingerprint Authentication Scheme Using Smart-Contracts

Xiaohu Zhou, Yousif Hafedh, Yonghao Wang, Vitor Jesus

School of Computing and Digital Technology, Birmingham City University, Birmingham, United Kingdom
vitor.jesus@bcu.ac.uk

Abstract. Biometric authentication, and notably using fingerprints, are now common. Despite its usability, biometrics have however a caveat which is the impossibility of revocation: once the raw fingerprint is breached, and depending on the technology of the reader, it is impossible to stop an illegitimate authentication. This places a focus on auditing both to detect fraud and to have clear indications that the fingerprint has been breached. In this paper we show how to take advantage of the immutability property of Blockchains to design an auditable protocol based on Diffie-Hellman key exchange with applications to fingerprint authentication.

Keywords: biometrics, authentication, security, blockchains, smart-contracts, auditing

1 Introduction

Fingerprint biometrics are nowadays a mature and widespread technology for authentication. It has several advantages being the two most important its usability (by being inherent to virtually any human) and accuracy since two people having the same fingerprint pattern is extremely rare. Fingerprint authentication has also seen commercial use in access control and application security and an example of such is seeing many modern mobile phones supporting it. There is however an important caveat which is the relative easiness of capturing fingerprints and later re-creating (say with wax fingers) and, post-incident, the impossibility of revoking a lost fingerprint in its original form.

In a world where one can only expect to have the identity stolen at some point in the future, biometric authentication is likely to become a reliable, yet non-authoritative (e.g., for legally binding actions), means of authentication. More than with any other security control, once an illegitimate attempt was successful, one can only expect more incidents to follow. Obtaining reliable evidence of the vulnerable step in the authentication flow where the invalid authentication happened is of paramount importance to harden systems, provide security checkpoints (such as multi-factor) and, in general, assess the risk against the sensitivity of the resources accessed. A simple example is with a bank account. If it is known that the fingerprint of a hu-

man has been illegitimately used, allowing bank transfers purely based on it is perhaps not advisable but viewing a bank statement is likely to meet the risk profile.

Audit trails, in the form of reliable and authoritative data on the process, along with real-time notifications, thus play a key role. Following the bank account example, after an incident has been recorded, it is likely that the bank will want to know where the vulnerable step happened; for example, if the authentication was successful because the fingerprint did match or whether it was deeper in the process and a mismatching fingerprint (or any other form of authentication) was accepted.

This paper proposes and discusses how blockchains can be used to provide an authoritative mechanism to prove, beyond any arbitrary level of doubt, that an illegitimate but successful identification took place at the fingerprint scan stage. Blockchains have the inherent property that, if enough time has elapsed (that depends on its size), whatever data is stored in it becomes immutable thus providing an excellent medium to store audit trails and reliable access notifications upon which one can act on.

The challenge we tackle is how to involve a blockchain in the authentication process. We do so by creating a protocol similar to Diffie-Hellman key exchange that, instead of running peer-to-peer, is ran intermediated by a blockchain. Considering that blockchains, whatever its form, typically require private keys to interface it, this scheme inherently provides multi-factor authentication. We envision our work to be, in fact, generic to any type of biometric authentication that relies on pre-acquired templates (such as images of fingers or iris) but we will keep the scope to fingerprints in order make the problem easy to discuss.

In section 2 we present the technical background on fingerprint biometrics and blockchains. In section 3 we present our approach and in section 4 we illustrate with an implementation strategy using Ethereum and the Solidity language, a public blockchain supporting smart-contracts. We conclude our paper in section 5.

2 Background

This section gives a technical background on fingerprint biometrics and blockchains; it further discusses related work in the area.

2.1 Fingerprint biometrics

Fingerprint recognition is based on identifying a user by comparing stored fingerprint data (at enrolment stage) with input fingerprint data obtained in real-time. An automated fingerprint recognition system is concerned with fingerprint acquisition, minutiae extraction, minutiae match and storage [1]. There are two main phases: enrolment and authentication. Before a user can authenticate, s/he needs to record the images on an enrolment module. The minutiae (wiggling patterns virtually unique to everybody) are extracted and stored in a template with an associated user ID. The template is, typically, further transformed as explained below. On authentication, the authentication module extracts a minutiae pattern from the image (the probe) to compare with the corresponding user ID template in the system. A matching algorithm will then

determine a matching score which, given a threshold, will then make a decision to accept or not.

Storing the templates of users is a central problem because, if breached in the raw form, it is impossible to revoke in the sense of revoking a certificate or changing a password. Cancellable biometrics is a key technique to help with this problem. Instead of storing the raw image or template, a distorted version is used either using a non-invertible transformation or biohashes [2] with the latter offering convenient fixed sizes templates.

2.2 Blockchains and Smart Contracts

Blockchains are a recent, and still maturing, technology having its first appearance to solve the "double-spending problem" in a digital currency and later recognised to solve the more generic "two generals' problem". It has evolved from its first application, Bitcoin, to now support generic scripting (smart contracts) as is the case of Ethereum. Whether it is information, in the sense of a ledger as in Bitcoin, or algorithmic methods and execution state, such as in Ethereum, Blockchains have the disruptive property of immutability: once stored, data is subject to cryptographic operations that are virtually impossible to reverse without abundant computing resources which is made further harder as time passes and as blocks (holding information) are added since they are interlinked. Therefore, if enough time is elapsed (i.e., enough computing effort is spent) it becomes virtually impossible to modify or destroy a record which brings auditing potential [3].

2.3 Related Work

Whereas literature is rich and abundant in biometrics authentication [4][5], and Blockchains are already being analysed in research and academic literature, beyond commercial applications, very few works have approached the combination of the two techniques. Hammudoglu et al [6] propose a mobile biometric-based authentication system for a self-sovereign identity solutions. It integrates a permissionless blockchain with identity and key attestation to be used in mobile phones. This work, however, is focused on how to implement self-sovereignty but storing secret and biometric material in full user control rather than the generic scenario of ours which is enhancing current biometric systems with auditing capabilities of blockchains. A similar remark can be made for Nandakumar et al [7] which design a fairly complete system relying on private blockchains and mixing the blockchain's consensus layer with biometric material so that the decision of matching is distributed. This is further made secure by using secret sharing techniques such as Shamir's.

To the best of our knowledge, this is the first work to combine biometrics with blockchains and especially to use blockchains to enhance the auditability of an authentication process.

2.4 Problem Statement

The trace of the authentication is the central point of this paper. Our attacker model is simple: finger can be stolen or a copy of the fingerprint can be re-created. Our approach does not directly mitigate this problem; rather, it gives trusted means for a user to be notified and later audit the security breach since an adversary cannot delete records. Another possible attack vector is an inside malicious actor: no authentication was done but it is claimed to have happened. Since the insider is able to manipulate the logs, there is no way to credibly dispute or disprove there was, or was not, authentication via fingerprint. If an authentication point has been compromised, there will be no trace left in the blockchain.

3 Approach

Our approach is based on Diffie-Hellman Key Exchange (DHE) which allows the generation of a shared secret between two parties over an insecure channel. In order to make the authentication process auditable, while protecting the biometric material, we use the blockchain as a secure and immutable medium on which messages exchanged cannot be modified once written and enough time has elapsed. **Fig. 1** shows our scheme.

The user, previously enrolled, request authentication by using an agent application. This could be a mobile application. The authenticator responds with the following. It initiates the protocol by sending (random) parameters for the protocol (g, p) and (not shown in the diagram), the location of the smart-contract such as its address. It is assumed the user has secret keys to access the smart-contracts which, effectively, acts as a second-factor for the authentication process. The public key of the authenticator, perhaps obtained in real-time from a certificate sent by the authenticator (e.g., using TLS), is used to encrypt both the remaining material of DHE and also to send a random nounce, r. Shown in dashed lines is the implicit broadcast process of a blockchain: once the block is mined, it is broadcast to all participating nodes. At this point, an audit point **A1** is created proving the user possesses the keys to interact with the blockchain.

The authenticator receives and decrypts the first message of DHE along with the nounce. Still using the blockchain, the authenticator publishes both its component of DHE and the nounce decrypted thus proving it possesses the public key and it is the right endpoint to send authentication material. This creates audit point **A2**. The user and the authenticator have now a shared secret, $k = A^b = B^a$. The user scans the finger (optionally transforming in the sense of cancellable biometrics) and encrypts with shared secret k its fingerprint. It further can encrypt with the receiver's public key for better privacy and forward-secrecy. This creates audit point **A3**. The client then publishes in the blockchain the result S' which holds the fingerprint scan. Upon a match, the authenticator records the result and the user is granted access. At any of the audit points, a notification to the user should be sent. Noting that these points are blockchain-wide, and any client for that blockchain can read it, the impact of a com-

promised agent that is preventing notifications is reduced.

The result of this process is there will now be an undeletable trace on the block-chain of an authentication attempt, whether successful or not.

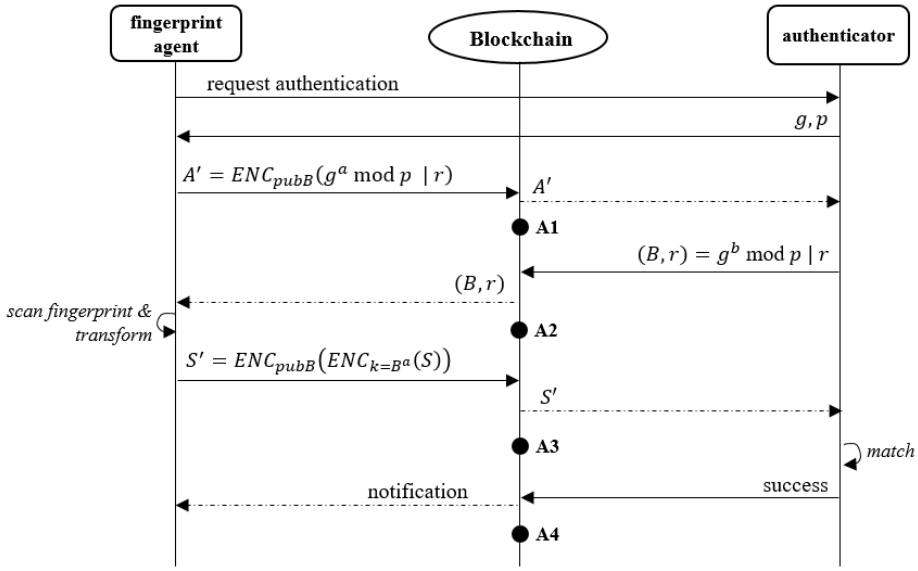

Fig. 1. Signalling diagram and audit points.

4 Evaluation

We used SourceAFIS [8] framework as the fingerprint authenticator. We've built a simple client that is fed with fingerprint images which, when compared the ones in the database produces, a matching score. We've complemented with an interface to Ethereum that executes the protocol in **Fig. 1** and executes the various stages on a Smart-Contract.

4.1 Implementation

The (simplified) smart-contract code is shown in **Fig. 2**. The code simply acts as a medium to exchange messages by implementing two generic methods for message passing of the two peers, called Alice and Bob. We assume, although it is not necessary, that a new contract will be created for every authentication attempt. When the contract is created by the authenticator, the user's address `USER_ADDRESS` will be hardcoded in the contract for control purposes. This provides the claimed 2-factor authentication as only the keys to that address will be able to interface the contract.

```
contract Authentication {
  String message;

  constructor() public {
    supplicant = USER_ADDRESS;
    user_authenticated = false;
  }

  function bob( string msg ) public {
    if (msg.sender == authenticator) {
      message = msg;
    }
  }

  function alice( string msg ) public {
    if (msg.sender == supplicant) {
      message = msg;
    }
  }

  function auth_result() {
    if ( msg.sender == authenticator )
      user_authenticated = true;
  };
}
```

Fig. 2. Simplified smart-contract code.

4.2 Evaluation

As expected, running a protocol over a smart-contract is significantly slow when compared with point-to-point protocols. We deployed and ran the contracts in a local Ethereum test network where no other contracts were being executed. This assured that every block was predictable given the low load in mining and confirmations occurring about every 15 seconds. Running the protocol took minutes as expected and one should note that this is the simplest case where, e.g., no retries exist and the network has no load. A further practical issue is storage which, for the time being, makes this somewhat unfeasible. Since, at certain point, the user sends the fingerprint, regardless of how compressed it may be, it will take up space which, in a public blockchain may be unfeasibly expensive.

5 Conclusions and Outlook

This paper discussed the usage of a blockchain, and smart-contracts, to enable trusted audits of fingerprint biometrics. As seen, any biometric process that requires exchange of media can take advantage of this protocol to guarantee that actions leave an unmodifiable audit trail that can later be analysed.

Our scheme needs improvement, nevertheless, given its impracticalities, notably using the blockchain to store the candidate fingerprint. Furthermore, the time it takes to authenticate a user is also of consideration which may defeat the typical usability of biometrics as an authentication vector. These and other issues are the subject of our current and future work.

References

1. Jain, AK & Nandakumar, K., *Biometric authentication: system security and user privacy,* IEEE Computer, vol 45, pp. 87-92., 2012
2. Ratha, N. K., Chikkerur, S., Connell, J. H. & Bolle, R. M., 2007. *Generating Cancelable fingerprint templates.* IEEE Transactions on pattern analysis and machine intelligence, 29(4), pp. 561-572.
3. Abreu, P., Aparicio, M. & Costa, C., *Blockchain technology in the auditing environment*, Caceres, CISTI, 2018
4. Pakutharivu, P. & Srinath, M. V., *A Compehensive Survey on Fingerprint Recognition Systems,* Indian Journal of Science and Technology, vol 8, issue 35, 2015
5. Weizhi Meng, Duncan S. Wong, Steven Furnell, Jianying Zhou, *Surveying the Development of Biometric User Authentication on Mobile Phones*, IEEE Communications Surveys & Tutorials, Volume: 17 , Issue: 3, 2015
6. Hammudoglu, J.S., Sparreboom, J., Rauhamaa, J.I., Faber, J.K., Guerchi, L.C., Samiotis, I.P., Rao, S.P. & Pouwelse, J.A., *Portable trust: biometric-based authentication and blockchain storage for self-sovereign identity systems.*, June 2017: online available at http://cn.arxiv.org/pdf/1706.03744
7. Nandakumar, K., Ratha, N., Pankanti, S. & Darnell, S., 2017. *Secure one-time biometric tokens for non-repudiable multi-party transactions*. IEEE Workshop on Information Forensics and Security (WIFS), Rennes France, December 2017
8. Vazan, R., *SourceAFIS.* [Online] Available at: https://sourceafis.machinezoo.com [Accessed 26 September 2018].