Towards a new Cryptographic Protocol for Biometric Security Technologies

Arnab Paul *
Kishore Ramachandran †
College of Computing
Georgia Tech

Abstract

We propose a new cryptographic protocol for remote authentication. This protocol is suitable when the signature key is very large. The emerging paradigm of verification through biometric samples, such as fingerprints, retinal images etc., naturally represents such a scenario. The proposed protocol cuts down the bandwidth requirement by selectively transmitting a very small part of the large signature, instead of the large signature itself. In addition, the protocol offers another extra layer of security if used on top of RSA. The design of the protocol is based on the notion of Probabilistic Checking of Proofs (**P**CP). The computational overhead involved in realizing this scheme would be very small since it consists mostly of arithmetic over polynomial fields which is typically very fast and can be implemented with inexpensive hardware or software.

1 Introduction

Biometric devices are becoming common in security and access control applications. The advantage of using biometrics is obvious; the users need not carry any additional token such as smartcard or need not rely on a small password which can potentially be broken by trial and error without much effort. However, an immediate downside is the fact that a biometric sample is permanently coupled to an individual and can never be changed. Once a malicious user gets hold of someone's biometric signature used to authenticate oneself, the same signature can potentially never be reused again. Some of the biometric samples

*arnab@cc.gatech.edu
†rama@cc.gatech.edu

are indeed easy to obtain without the owner's cognizance. Fingerprints, voice sample etc., fall in this category. This ease of acquiring permanent biometrics and the resulting danger of insecurity led another stream of research that explored dynamic biometric features, such as an individual's keystroke dynamics while typing the password [5]. However, some biometric samples such as retinal image, are hard to get. Regardless of the difficulty of stealing, a biometric sample is usually quite large, typically in the order of a few hundred kilobytes, and therefore cannot be easily generated in an artificial way.

In this paper we consider Remote Authentication using biometric signatures. As of now, biometrics haven't seen much use in remote authentication over a network. However, as technology advances, this may become a common practice. For remote authentication, a user has to send the whole biometric sample over the network. The protocol proposed in this paper uses a trick so that the entire sample need not be transmitted. Instead, a very small fraction of it would be enough for the verifier at the remote end to validate the user's identity. In addition to reducing the bandwidth of such a procedure, the protocol would also offer an additional level of security. However, since we are reducing the transmission by a huge factor, for efficient reconstruction of the biometric sample at the remote end, usually some more computation is necessary. But this computation will add only an insignificant overhead.

The design of the protocol uses the notion of *Probabilistic Checking of Proof* [1]. The paradigm of proof checking has deeper connections with complexity theory and error connections. In a nutshell, any problem in NP [1] can be cast as a problem where a prover is producing some proof and a verifier is verifying the same, and in this process the verifier is looking into only a small fraction of it instead of the whole proof. This is very counter intuitive, and will be discussed with little more details in section 2. Authentication however, seems to readily fit into a prover-verifier framework. But we reformulate it in a slightly different way in section 3. The reason for reformulating the problem is to make it resemble the prover-verifier version of an NP problem. That helps us use the trick of checking only a small fraction of bits in the proof, i.e., in our case, it suffices to transmit only a small fraction of the large biometric sample (the proof of authentication) over the network. It is worth mentioning that **P**CP builds upon the results from Theory of Error Correction which in turn is based on polynomials over mathematical

[1]The class of problems solvable by Nondeterministic Turing Machine in Polynomial Time

structures called Finite Fields. The techniques used in probabilistic checking of a proof relies on a bunch of other results related to many properties of low degree polynomials [2, 3, 4, 6]. The interesting challenge from a systems point of view is to integrate these theoretical notions into an implementable authentication system. We already mentioned that the scheme proposed here will cut down the bandwidth requirement. In addition, this also adds an extra level of security feature. This additional gain is achieved because our protocol needs the data to be encoded as some polynomial. And only a small fraction of the data gets out in the network. If a malicious eavesdropper accesses this small part, he has to do some additional computation in order to extract the data. Though this in itself cannot be used as an unbreakable code, on top of RSA, this scheme definitely adds an extra level of security feature. In fact, error correction codes have been proposed to be used for designing secure communications [7, 8, 9]. There are different types of Error Correcting Codes. Some of them are not secure enough and are prone to trapdoor attacks [10, 11]. However a very recent result shows that some codes are secure as well [12] and can be used as a backbone of secure communication. An interesting investigation would be to find out if such secure codes (discussed in [12]) can be used as the encryption technique for **P**CP scheme. However, we are assuming that our protocol sits on top of a secure scheme such as RSA. Finally, one should note that computations with low degree polynomials that are relevant to our scheme aren't very expensive in principle. We believe that an efficient implementation is possible in both hardware and software.

We organize the paper in the following way. In section 2 we define the prover-verifier game and PCP and describe how the complexity class NP connects to such a setting. In section 3 we show how remote authentication can be cast as an instance of PCP. This leads us to section 4 where we describe our protocol. However, we haven't yet implemented this protocol. Though in principle the implementation looks quite feasible, a neat and optimal way to do it isn't very obvious to us. In section 5 we discuss why this is so and the possible solution.

2 Preliminaries

We first show that the problem of remote authentication can be reformulated as a two party Prover-Verifier game. In this framework we can immediately apply the notion of PCP and the associated techniques to illustrate how to cut down the number of transmitted bits. As a final assimilation we describe our protocol. We start with defining the infrastructure, viz., the Prover-Verifier game.

2.1 Prover-Verifier Game and PCP

Any computational decision problem essentially means deciding a question like $\mathbf{x} \in \mathbf{L}$? , where x is a string and L is a language, both over the same alphabet [2]. A machine essentially decides this set inclusion problem. The hardness of the problem depends on the lower bound on the time the machine would take in deciding the question. Our two party setting consists of the following : (i) The Prover (**P**r), who has an unlimited computational power, and (ii) The Verifier (**V**f) who has polynomial amount of computational resources. The game is the following: **P**r is trying to prove to **V**f that some string x belongs to some language L. **P**r can produce evidence in the form of bit strings and **V**f will use his limited computational power (in terms of space and time) and verify the question if $\mathbf{x} \in \mathbf{L}$?.

Definition 1 *$\boldsymbol{NP}$ is the class of languages, such that $\forall L \in \boldsymbol{NP}$, and for a string $x \in L$, **P**r can always present a string y of length $\mathcal{O}(poly(n))$ (n is the length of x), such that taking x and y as inputs, **V**f can verify the claim (viz., $x \in L$) in $\mathcal{O}(poly(n)$ time. y is called the certificate for x.*

The above definition is exactly equivalent to the standard definition that **N**P is the set of all languages that can be decided by a Non Deterministic Turing Machines in polynomial time.

PCP has a randomized setting. This is almost identical to the Prover-Verifier game that we described before. However, in addition to the certificate string y (and the original input string x), **P**r presents to **V**f a random bit string r of polynomial length. **V**f is going to take three strings as input, viz., r, y, x but while deciding if $\mathbf{x} \in \mathbf{L}$, it is not going to use all the bits of y but only a selective few (depending on r) and still decides with *high probability* if $\mathbf{x} \in \mathbf{L}$.

Definition 2 *$\boldsymbol{PCP}(r(n), q(n))$ denotes the set of languages for which the random string r has length $r(n)$ and **V**f can look only $q(n)$ number of bits from the certificate string y and still be able to decide with high probability if $x \in L$.*

Having set up the framework, we now introduce the result [1] that we are going to use to design our authentication protocol.

Theorem 1

$$\boldsymbol{NP} = \boldsymbol{PCP}(\mathcal{O}(\log n), \mathcal{O}(1))$$

In other words, **P**r *gives the verifier only* $\mathcal{O}(\log n)$ *random bits (r) and* **V**f *looks*

[2] An alphabet is a set of symbols

into only **c***onstant number of bits into the certificate string, depending on $r(n)$, in addition to x and and r to resolve if $x \in L$.*

PCP theorem stated above enables us to use the trick. Now it is intuitively clear that we are trying to design a Verifier for our remote authentication system that will need only a few bits (constant number - to be exact) instead of the entire sample. Now we need to reformulate our our authentication mechanism as a **P**CP instance, which we do in the next section.

3 Remote Authentication as a Prover Verifier game

In this section we cast the problem of remote authentication as a two party game. In particular we consider the scenario of a user remotely authenticating himself to a server, using a biometric sample. The user provides a user id, say U and a biometric sample b. The server can access a database which has a sample b_U known to be the identifier for U. Once presented with b, the server computes some distance $d(b, b_U)$ between the two samples, and if $d(b, b_U)$ is small enough, it approves U.

Now, we can think of the aforementioned scenario as a language recognition problem as follows :

Let L be the set of all possible biometric samples that can be generated by U. Usually all the elements of L are almost identical differing very little from each other. However, L is not necessarily a singleton set. Consider Finger Print recognition for example. All the thumb imprints generated by a person are almost identical, but they may vary by small amounts depending on the physical condition of the hand and the sampling device. When U requests the server for authentication, the question the server really asks is - if $\mathbf{b_U} \in \mathbf{L}$?. The answer is obviously affirmative if the user is really U. However if the person is an imposter and not U then he would potentially generate another language $L_v \neq L$ so that $b_U \notin L_v$. To prove the identity, the user gives one biometric sample of his own, which plays his certificate to the server. If the user is really U, then he can give a sample b such that $d(b, b_U)$ is quite small. However for some other person $V \neq U$, it won't be possible to give such a biometric sample. To summarize, the process of remote authentication can be thought of as a two party game, where the server plays the role of **V**f , the user plays the role of **P**r , the sample b_U in the server database is the input string (called x in Definition 1). The user (or the prover **P**r) provides the certificate string b (denoted by y in Definition 1).

Now we can present the above scenario in a **P**CP setting. The question asked is - if $b_U \in L$? The work required to resolve the question is computing the distance $d(b, b_U)$. This computation is typically a finger print recognition or a scanned retinal image matching or something similar depending upon what kind of biometric sample is being used. There are polynomial time algorithms available for these pattern matching problems. So the decision question is in **N**P . Given this, We can now appeal to Theorem 1 which asserts the existence of a randomized algorithm A such that given a logarithmically long random string r , **V**f (the server) can run A on r, b_U and only $\mathcal{O}(1)$ number of bits of b and decide with very high probability if $b_U \in L$, i.e., approve (or disapprove) the user at the remote end. This is the focal point of this paper. Once **P**r and **V**f agrees on the random string r, **P**r needs to present to **V**f only a very small number of bytes from its biometric sample. And by the strength of Theorem 1, **V**f will still be able to decide whether **P**r is an authorized person or not. Normally the biometric sample will be quite large in size, usually a few hundreds of kilobytes. Instead, in our setting only a very small fraction of that needs to be communicated over the network. The string r is also logarithmically small compared to b resulting in a very small overhead of communication. This leads to the final protocol which we summarize in the following section.

4 The Remote Authentication protocol

All the previous discussion culminates in constructing our protocol. We stick to the same notation used so far to denote the parties and input strings of our problem.

1. **P**r sends his user-id U to **V**f .
2. **V**f generates a small number of random bits (logarithmic in the size of the expected biometric sample) and sends this string (r) to **P**r .
3. **P**r computes from r and b, a small string b_c (of $\mathcal{O}(1)$ length) and sends this to **V**f .
4. **V**f runs a computation with r, b_u and b_c as inputs and decides the authorization. This becomes possible by the result of **P**CP theorem already stated.

The above protocol offers the following advantages.

- The communication overhead associated with a remote authentication is reduced by a significant amount.
- Since an additional level of encryption of information takes

place in order to transmit the data, this will offer at least one extra level of protection on top of the normal one offered by a public key infrastructure.

- The whole process of computing the small subset of bits to be communicated is usually done through techniques of Error Correcting Codes which heavily uses polynomials and an associated mathematical structure called Galois field. It has been found that these Galois field operations are very easy to implement in hardware in terms of some cheap circuitry, which means the shift from a straightforward communication to this complicated one, doesn't really add any extra computational overhead.

5 Implementation and Future Work

In this section we describe the main challenge in the implementation of our protocol. As we've already seen, this protocol is based on the idea of **P**CP which gives us a novel polynomial time algorithm for verifying proofs. However, this algorithm is meant for a specific problem called 3-CNF [3] satisfiability. This being an NP-complete problem, ensures that for any other problem in NP, an equivalent algorithm exists. In our case the verification algorithm typically does some pattern recognition in polynomial time. And all we know for sure by the strength of **P**CP theorem is that there exists another equivalent algorithm that performs the same task, but looks only into a small fraction of the pattern. However, no one to our knowledge has ever constructed such an algorithm. One straightforward way would be to reduce the problem of pattern recognition into 3CNF and then apply the algorithm known for 3CNF on the new reduced form of the problem. [4]. However, as such this reduction is quite complicated for implementation. Our future work consists of modifying a standard algorithm (such as fingerprint recognition) into one that our protocol can use. And we believe that not only is such an algorithm constructible but also implementable through *inexpensive* polynomial computations over finite fields. In fact modern communication routinely uses these easy polynomial computations in order to do error correction. Moreover, the error correction techniques used by **P**CP are quite similar. While implementing our protocol seems to be quite feasible, a parallel open question is

[3] Conjunctive Normal Form

[4] This is doable because anything in NP is reducible to 3CNF

whether any *secure* error correction code [12] can be used as the backbone for our protocol. This is completely a theoretical question for which we don't have any answer yet.

References

[1] S. Arora *Probabilistic Checking of Proofs and Hardness of Approximation Problems*. Ph.D Thesis, Computer Science Devision, University of California at Berkeley, 1994.

[2] R. Rubinfeld, M. Sudan. Testing Polynomial functions efficiently and over rational domains. *Proceedings of the Third Symposium on Discrete Algorithms*, ACM, 1994.

[3] R. Rubinfeld, M. Sudan. Robust characterizations of polynomials with applications to program testing. *TR RC-19156, IBM Research Division*, T. J. Watson Research Center, Yorktown Heights, NY 10598, September 1993.

[4] k. Friedel and M. Sudan. Some improvements to total degree tests *Proceedings of the Third Israel Symposium on Theory and Computing Systems*, 1995.

[5] F. Monrose, M. K. Reiter, and S. Wetzel. Password hardening based on keystroke dynamics. *Proceedings of the 6th ACM Conference on Computer and Communications Security* November 1999.

[6] C. Lund, L. Fortnow, H. Karloff and N. Nissan. Algebraic methods for interactive proof systems *Proceedings of the Thirty First Annual Symposium on the Foundations of Computer Science* , IEEE, 1990.

[7] R. McEliece, E. Berlekamp, and H. Van Tilborg. On the inherent intractability of certain coding problems. *IEEE Trans. on Information Theory* IT-24(3), May 1978

[8] R. McEliece, A public-key cryptosystem based on algebraic coding theory. Jet Propulsion Lab. DSN Progress Report 1978.

[9] E.M Gabidulin, A. V. Paramonov and O.V Tretjakov, Ideals over a non-commutative ring and their application in cryptology. *Advances in Cryptology - EUROCRYPT91* ed. D.W. Davies, vol. 547 LNCS Springer Verlag 1991.

[10] N. Sendrier, On the structure of a randomly permuted concatenated code. *EUROCODE 94*, 1994.

[11] On cryptosystem based on generalized reed-solomon codes, *Diskret. Maths*, (4), 1992.

[12] A. Canteaut and N. Sendrier, Cryptanalysis of the original mceliece cryptosystem. *Advances in Cryptology - ASIACRYPT'98, LNCS, Springer Verlag* 1998.