Human Authentication Protocol for Distributed Computing Environment

Dang Nguyen Duc and Kwangjo Kim

International Research center for Information Security (IRIS)
Information and Communications University (ICU)
119 Munjiro, Yuseong-gu
Daejon, 305-732, Korea
{nguyenduc, kkj}@icu.ac.kr

Abstract. Human authentication and identification is a process of proving human's identity to a machine. This kind of protocol is useful where electronic devices carried by human like portable devices, smart card cannot be trusted. In this paper, we address the practical use of a human authentication protocol by Hopper and Blum [11] for a distributed computing environment. We argue that, in a distributed computing environment, authentication needs to be done in a proxy manner. That means user information is kept at a single trusted server and this server assists service provider machines scattering around area to authenticate users without revealing user's secret information to them. Online communication between the trusted server and service providers should not be required in order to avoid the server being overloaded. In this paper, we present a solution by adapting HB protocol to three-party setting where a service provider acts as a proxy between a user and the server.

Key words: human authentication, LPN problem, proxy authentication.

1 Introduction

Authentication and identification are among the most important cryptographic primitives. After the seminal work on ID-based cryptography by Fiat and Shamir [1], many protocols have been proposed in the literature. However, most of traditional protocols assume that two parties, a prover and a verifier, have significant computational and memory capacity, far beyond that of human's brain. For example, a provable secure protocol often bases its security on some hard number theoretic problems like factoring or discrete logarithm problem. And as a consequence, it requires addition, multiplication and exponentiation of numbers with roughly a hundred digit long. While validating authenticity of a human is a more natural problem in real life (than authenticating a device held by the human), such computational burden forces people to depend on machines in order to prove their identities to another machine, regardless of whether it is really applicable. With the increasing number of computer viruses, Trojans and hacking tools these days, it is even more necessary not to trust personal electronic devices, especially for end users. In the attempt to develop human-friendly cryptography, it is not surprising that researchers have been looking for new foundations for provable secure protocols and hopefully these alternative foundations would put less burden on the prover side (human) [1]. The earliest works in human authentication was done by Matsumoto and Imai [2]. Their scheme

[1] Biometric is also a non-cryptographic approach to human authentication and identification. Arguably, it still requires expensive hardware and is not reasonable reliable in some case.

was later improved by Wang *et al* in [3]. However, their schemes look quite complicated for normal users and more seriously are usable for only a limited number of times for a single secret key. The reason that discourages using Matsumoto and Imai's scheme is its lack of a firm security foundation. Instead of relying on a well-studied hard problem, the security of their protocol depends on a sophisticated trick to reduce the chance of discovering the secret key by an eavesdropper. Matsumoto himself also proposed another scheme in [4]. Unfortunately, his scheme suffers from the same problem in [2] and [3]. The first formal treatment of human authentication and identification appeared in [6]. Along with model and rigorous definitions of security, Naor and Pinkas also suggested several schemes based on the 2-out-of-2 visual secret sharing scheme. In their scheme, a human (as a prover) and (a machine) as a verifier can share a secret key as a physical object called transparency. This transparency is therefore subject to stealing and copying.

The first human authentication scheme based on a well-studied hard problem is due to Hopper and Blum (HB for short) [11]. The hard problem they used comes from machine learning field called *Learning Parity in the Presence of Noise* (LPN). Several researches have shown that LPN is likely very hard problem in general [5, 7, 9]. An advantage of HB human authentication scheme over the previous works includes multiple usage of a single secret key with low complexity. Here, we will extend the original HB scheme to three-party setting which is specially applicable in a distributed computing environment.

In a distributed system, the whole system consists of networked machines located in different geographical locations and yet it appears to users as a single physical system. When a user tries to log into the system (using a human authentication protocol), the local machine might have a copy of user database or needs to contact the so-called authentication server in order to verify the user. In this paper, we consider a scenario in which user database is centralized at the authentication server and this server is fully trusted while other machines in the network are considered less secure (*i.e.,* subject to be hacked). Keeping a single copy of user database have the following advantages:

- Avoiding the job of replicating user database to multiple machines across the network.
- Reducing the chance of revealing user information to hacker by only keeping it at a secure place.

However, the other machines in the network should still be able to verify user's identity. Obviously, the authentication server should send some data related to user's secret information to the other machines so they have some knowledge to assure that only genuine users are granted access to computing resources. To preserve the advantage of securing user information in one place, it should be infeasible to extract secret information from data sent by the authentication server. In addition, to avoid the authentication server being flooded, data from the authentication server could be reused without compromising security of the authentication process.

In a typical distributed system like the automated teller machine (ATM) network, user information is stored at a centralized secure server, say account server. Whenever a user needs to access to an ATM machine, the ATM machine contacts the account server through a secure communication line in order to authenticate

the user. Even though it is true that ATM machines should have online connection to the authentication server to handle a bank transaction, it is a good practice to verify user identity before actually initiating a connection to account server. In case all authentication information is stored on a smart card carried by the user, this makes smart card a very attractive target for stealing and cloning. Recent incidents show that credit and debit card fraud keeps increasing. Therefore, we think that secret information like PIN should not be kept on a physical device as well as any ATM machine which is located in a public area. Instead, user should memorize it and prove his owner of PIN using a human authentication protocol in a way that no online connection from an ATM machine to the account server is required and only some pre-computed data sent by the account server is needed.

In this paper, we realize the aforementioned scenario by proposing a variant of the HB human authentication protocol. In our protocol, there are three parties, namely user, machine and authentication server (AS for short). The AS is fully trusted and keeps all user information. We assume that the machine acts honestly but it is not secure. To authenticate a given user, the machine should first get some appropriate pre-computed data from the AS, not necessary through a secure channel. Using the pre-computed data, the machine can verify the user authenticity on its own will. We show that our modified protocol remains secure against impersonation under the assumption that the adversary is passive. We also show that the pre-computed data sent by the authentication server cannot be used to reveal user's secret information. Last but not least, our scheme does not incur any more complexity on human side. The amount of pre-computed data from the authentication server is also reasonable.

2 Background and Previous Work

2.1 Binary Inner Product

The HB protocol makes use of binary inner product of two k-bit numbers. We briefly review the concept including its property. Given two k-bit number $a = (a_0 a_1 ... a_{k-1})_2$ and $b = (b_0 b_1 ... b_{k-1})_2$, the binary inner product of a and b, denoted as $a \cdot b$ is computed as follows:

$$a \cdot b = (a_0 \wedge b_0) \oplus (a_1 \wedge b_1) \oplus \cdots \oplus (a_{k-1} \wedge b_{k-1})$$

This binary inner product operation can be carried out relatively easy by human as well as on extremely low-cost hardware (like RFID tag). The distributive law also applies for binary inner product operation. Therefore, given three k-bit integers a_1, a_2 and b, we have $(a_1 \oplus a_2) \cdot b = (a_1 \cdot b) \oplus (a_2 \cdot b)$. This law can be easily generalized for more than two a_i's. One final observation we want to state is that $a \cdot b$ produces the parity bit for $a \wedge b$.

2.2 HB Human Authentication Protocol

In [11], Hopper and Blum proposed two human authentication protocols. The second protocol makes use of an error-correcting code. In this paper we refer HB protocol to the first protocol which is based on a hard machine learning problem. HB protocol repeats a basic challenge-response protocol r times either in a sequential or concurrent manner [13] (r is a security parameter). Each time, a different challenge

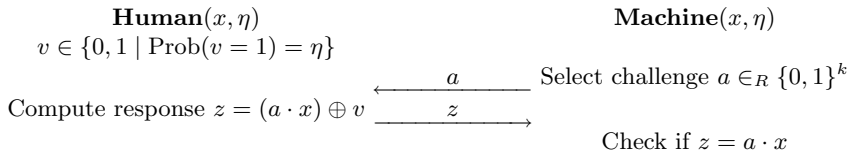

Fig. 1. One round of HB Protocol

should be used. One basic round of HB protocol is depicted in Fig.1. Hopper and Blum showed that any cheater attempting to play random guess of the response z has success probability at most $e^{-c_0 r}$ where c_0 is a constant depending on η and greater than $\frac{2}{3}$. Note that HB protocol is not secure against active adversaries since any attacker who is capable of modifying the challenge a sent by $\mathcal{C}$ might replace the random a with a fixed a for $\Omega((1-2\eta)^{-2})$ rounds of the basic protocol. It is likely that he can learn a noise-free response z and thus come up with one valid equation with k unknown bits of x.

It is straightforward that HB protocol is secure against impersonation only if a computationally bounded eavesdropper observing messages exchanged between $\mathcal{H}$ and $\mathcal{C}$ has a negligible chance of being accepted by $\mathcal{C}$. More specifically, an eavesdropper $\mathcal{A}$ obtains r pairs of (a, z) and tries to deduce a k-bit number x' (in the best case, $x' = x$) such that using x' to carry out HB protocol, $\mathcal{A}$ would get accepted by $\mathcal{C}$. And we call such problem is *Learning Parity with Noise* (LPN for short) problem.

The LPN problem has been extensively studied in [5, 7, 9, 11]. Their results showed that LPN problem is very likely to be an intractable problem. The best known algorithm to solve LPN problem is due to Blum *et. al.* [9]. It has sub-exponential complexity of $2^{O(\frac{k}{\log k})}$.

3 HB Protocol in Three-Party Setting

We now proceed to describe our variant of HB protocol in three-party setting. Let us denote $\mathcal{H}$ to be human user, $\mathcal{C}$ to be machine and S to be authentication server. $\mathcal{H}$ and $\mathcal{C}$ share a k-bit secret x. In the original HB protocol, $\mathcal{C}$ is capable of checking the correctness of $\mathcal{H}$'s response z based on its knowledge of the random challenge a and the secret x. Indeed, $\mathcal{C}$ can compute the pair $(a, a \cdot x)$ before actually executing a session of the protocol. Therefore, in three-party setting in which $\mathcal{C}$ does not know x, $\mathcal{S}$ can pre-compute a set of random challenge-response $(a, a \cdot x)$ and send it to $\mathcal{C}$. Using that set, $\mathcal{C}$ can authenticate $\mathcal{H}$ just like in the original HB protocol. However, as we stated before, the loss of pre-computed data from $\mathcal{S}$ should not reveal the secret x. Using the same technique for $\mathcal{H}$ to prevent eavesdroppers from discovering x, $\mathcal{S}$ can intentionally falsify $(a, a \cdot x)$ by replacing it with $(a, (a \cdot x) \oplus u)$ where u is determined by another noise factor μ such that $u \in \{0, 1|\text{Prob}(u = 1) = \mu\}$. $\mathcal{C}$ also counts the number of correct responses from $\mathcal{H}$ in order to decide whether to accept $\mathcal{H}$. However, due to the effect of the additional noise parameter from $\mathcal{S}$, the threshold value is now different. Suppose that a legitimate $\mathcal{H}$ has sent a response z to $\mathcal{C}$. Then, because v and u are generated independently, $z \neq z*$ with probability $\eta(1-\mu)+\mu(1-\eta) = \eta+\mu-2\eta\mu$. This leads to the threshold value being $(\eta+\mu-2\eta\mu)q$ where q is the number of times which the basic protocol is repeated. To conclude, $\mathcal{C}$

accepts $\mathcal{H}$ only if it receives less than $(\eta + \mu - 2\eta\mu)q$ unmatched responses from $\mathcal{H}$. Note that, the interaction between $\mathcal{C}$ and $\mathcal{S}$ is essentially the same as the interaction between $\mathcal{H}$ and $\mathcal{C}$ in the original HB protocol, *i.e.,* both a and $z*$ are available to eavesdroppers. Therefore, we do not even a secure channel between $\mathcal{C}$ and $\mathcal{S}$. $\mathcal{S}$ can just publish (a, z^*) for any machine which wishes to authentication users.

But one problem still remains. That is $\mathcal{C}$ needs to keep getting new (a, z^*) pairs for next authentication sessions. If so, it will limit all advantages of keeping secret information at the authentication server. To resolve this issue, first we observe that for 2 pairs (a_1, z_1^*) and (a_2, z_2^*), we can compute a new challenge-response pair $(a_1 \oplus a_2, z_1^* \oplus z_2^*)$ where $z_1^* \oplus z_2^*$ is a valid response probability $\mu(1-\mu) + \mu(1-\mu)$ (either z_0^* or z_1^* is falsified). If we want to produce a new challenge-response pair from t original pairs in the same way, according to **Lemma 3** of [11], the probability of the new response to be incorrect is

$$\gamma = 1 - \left(\frac{1}{2} + \frac{1}{2}(1-2\mu)^t\right) = \frac{1}{2} - \frac{1}{2}(1-2\mu)^t$$

For $\mu \in \left(0, \frac{1}{2}\right)$, we also have $\gamma \in \left(0, \frac{1}{2}\right)$. However, since γ is varied with n, it is difficult to find a correct threshold value for accepting $\mathcal{H}$. To resolve this problem, we fix n for one session of the protocol. We now describe a new variant of HB protocol which allows $\mathcal{C}$ to reuse a set of challenge-response pairs from $\mathcal{S}$. Let us assume $\mathcal{C}$ gets a set G of n challenge-response pairs, $G = \{(a_1, z_1^*), (a_2, z_2^*), \cdots, (a_n, z_n^*)\}$. As we know, there are $2^n - 2$ non-trivial subsets of G (excluding the empty set and G itself). At first, $\mathcal{C}$ randomly chooses a positive integer number t from $\{1, 2, \cdots, n-1\}$. To select a random challenge a to send $\mathcal{H}$, $\mathcal{C}$ selects a subset of G of order t at random and computes $\gamma = \frac{1}{2} - \frac{1}{2}(1-2\mu)^t$. Let the chosen subset be $g = \{(a_1, z_1^*), (a_2, z_2^*), \cdots, (a_t, z_t^*)\}$. The challenge a is computed by $a = a_0 \oplus a_1 \oplus \cdots \oplus a_t$ and the referenced response z^* is computed similarly, $z^* = z_1^* \oplus z_2^* \oplus \cdots \oplus z_t^*$. The authentication protocol can now proceed normally. $\mathcal{C}$ accepts $\mathcal{H}$ only less than

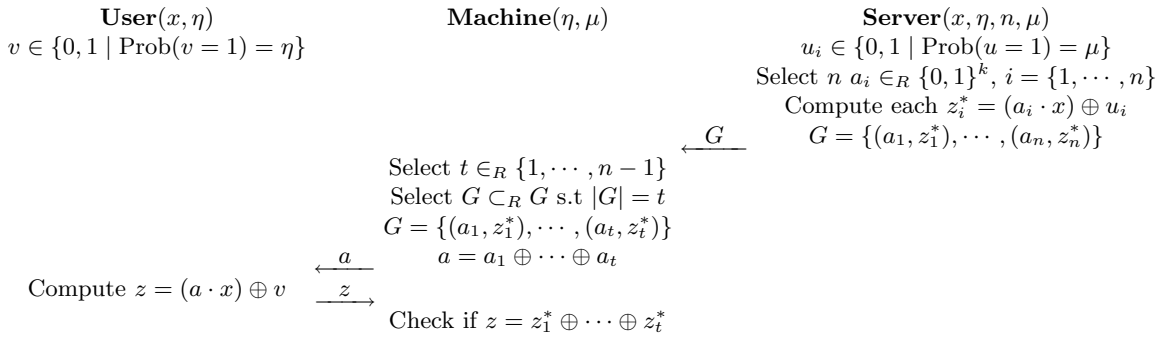

Fig. 2. One round of three-party HB Protocol

$[\eta(1-\gamma) + \gamma(1-\eta)]r = (\gamma + \eta - \gamma\eta)r$ responses from $\mathcal{H}$ are unmatched (to be safe, μ and η should be chosen so that $(\gamma + \eta - \gamma\eta) < \frac{1}{2}$, *e.g.*, $0 < \gamma, \eta < \frac{1}{4}$). In the original HB protocol, the random challenge a is k-bit long so the sample space has cardinality of 2^k. To maintain the same magnitude of sample space, we can let $n = k$. In addition, $\mathcal{S}$ needs to choose a_i so that all of them are pairwise linear

independent. This will ensure that the sample space in our modified HB protocol has cardinality of exactly 2^k. $\mathcal{C}$ may also periodically update its challenge-response from the authentication server.

4 Concluding Remarks

In this paper, we have presented a variant of a human authentication protocol by Hopper and Blum. Our variant addresses a real life issue that secret information should not be lying around together with computing equipments. However, keeping secret information at one secure location does prevent machines from authenticating users before letting them access computing resources. In addition, the centralized user database server should not be a bottleneck in the system. This can be done by pre-computing data required for authentication at the server and distribute it to other machines (using an insecure channel). Our proposed scheme is as secure as the original HB scheme and does not produce any significant overhead, especially on human side. We also suggest that our idea could be applied to other authentication system as well, especially RFID system.

References

1. Amos Fiat and Adi Shamir, "How to Prove Yourself: Practical Solutions to Identification and Signature Problems", *In the Proceedings of CRYPTO'86*, Odlyzko A. M. (Ed.), Springer-Verlag, LNCS 263, pp. 186-194, 1987.
2. Tsutomu Matsumoto and Hideki Imai, "Human Identification through Insecure Channel", *In the Proceedings of EUROCRYPT'91*, Davies D. W. (Ed.), Springer-Verlag, LNCS 547, pp. 409-421, 1991.
3. Chih-Hung Wang, Tzonelih Hwang and Jiun-Jang Tsai, "On the Matsumoto and Imai's Human Identification Scheme", *In the Proceedings of EUROCRYPT'95*, Guillou L. C. and Quisquater J. J. (Ed.), Springer-Verlag, LNCS 921, pp. 382-392, 1995.
4. Tsutomu Matsumoto, "Human-Computer Cryptography: an Attempt", *In the Proceedings of the Third ACM Conference on Computer and Communications Security*, Neuman C. (Ed.), ACM Press, pp. 68–75, 1996.
5. Johan Hastad, "Some Optimal Inapproximability Results", *In the Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing*, ACM Press, pp. 1–10, May, 1997.
6. Moni Naor and Benny Pinkas, "Visual Authentication and Identification", *In the Proceedings of CRYPTO'97*, Kaliski Jr. B. S. (Ed.), Springer-Verlag, LNCS 1294, pp. 322–336, 1997.
7. Michael Kearns, "Efficient noise-tolerant learning from statistical queries", *In the Journal of ACM* Volume 45, Issue 6, ACM Press, pp. 983–1006, November, 1998.
8. Xiang-Yang Li and Shang-hua Teng, "Practical Human-Machine Identification over Insecure Channels", *In the Journal of Combinatorial Optimization*, Volume 3, Kluwer Academic Publishers, pp. 347–361, 1999.
9. Avir Blum, Adam Kalai and Hal Wasserman, "Noise-tolerant Learning, the Parity Problem, and the Statistical Query Model", *the Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing*, ACM Press, pp. 435–440, 2000.
10. Nicholas Hopper and Manuel Blum, "A Secure Human-Computer Authentication Scheme", Technical Report CMU-CS-00-139, Carnegie Mellon University, May, 2000.
11. Nicholas Hopper and Manuel Blum, "A Secure Human-Computer Authentication Scheme", *In the Proceedings of ASIACRYPT'01*, Bart Preneel (Ed.), Springer-Verlag, LNCS 2248, pp. 149-153, 2001.
12. Ari Juels and Stephen Weis, "Authenticating Pervasive Devices with Human Protocols", *In the Proceedings of CRYPTO'05*, Victor Shoup (Ed.), Springer-Verlag, LNCS 3261, pp. 293-308, 2005.
13. Jonathan Katz and Ji Sun Shin, "Parrallel and Concurrent Security of the HB and HB+ Protocols", Available at `http://eprint.iacr.org/2005/461.pdf`.
14. CAPTCHA Project, `http://www.captcha.net/`.