INTERNATIONAL JOURNAL OF ADVANCED SCIENTIFIC RESEARCH AND TECHNOLOGY

ISSUE 2, VOLUME 2 (APRIL 2012) ISSN: 2249-9954

Provably Secure and Blind sort of Biometric Authentication Protocol using Kerberos

S. Nagendrudu [#1] , S. Swarnalatha [#2]

[#1] *Associate Professor, Department of CSE,*

Gurunank Institute of Technology, Ibrahimpatnam,

Andhra Pradesh, India,

nagendraji534@gmail.com, 9392896284

[#2] *Assistant Professor, Department of CSE,*

Aurora's Engineering College, Bhonghir,

Andhra Pradesh, India,

Swarnalathajan14@gmail.com, 9441613462

Abstract

Biometrics authentication has become popular with increase in infrastructure facilities and scope of sensor technologies. They are suited due to high security in applications like remote authentication. We are considering a provably secure and blind sort of biometric authentication protocol combined with the advantages of Kerberos ticket granting. We are using cryptography to make the protocol more secure. It can successfully run over public network for remote access. It can also be implemented to take care of the revoking of registered templates. It is not biometric specific. The main Kerberos part comes in because of the ticket granting mechanism. Kerberos and biometrics are already proven to survive range of attacks. Finally we show a central, already secure, server that can be used for mass authentication and wide range of applications

Keywords- Biometric authentication, encryption and biometrics, biometric Kerberos

Corresponding Author: Nagendrudu

1.1 Introduction

During remote connection we face certain challenges when it comes to security. The system we are going to propose for overcoming these security issues are using biometrics first of all which is cost effective and secure[1][2]. Thus if such technology or hardware is available at the user end it would be very much efficient for the secure authentication as we will see ahead. Though we are not using remote specifications at all points, we are addressing security issued based on it.

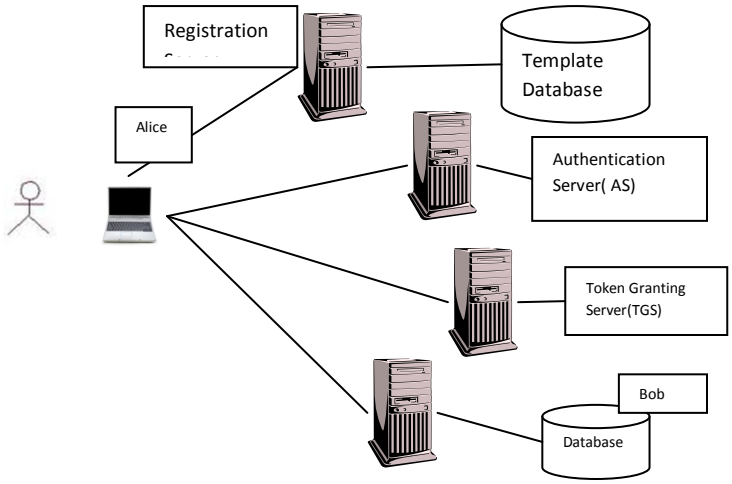
We have to take care about a lot of security issues here[3]. We are implementing a blind authentication protocol[4]. Using Kerberos sort of methodology. This blind authentication crypto- biometric authentication protocol[4] that the authors of the previous work had proposed is the main base and idea of our scheme for remote authentication. We know that how Kerberos has been successful as an authentication protocol. What we wish to do is make it more secure by integrating it with a crypto biometric authentication system in place of the password system that Kerberos implements.

We will see what problems in Kerberos implements. We will see what problems in Kerberos have been addressed and solved by this kind of methodology. Biometrics authentication has become popular with the increase in infrastructure facilities and scope of sensor technologies. There is a unique way of solving problem for each of these two methodologies by combining them. The user's actual biometric data is also not available with the authenticating server. It's only submitted to the registration server. The encryption and biometrics with the registration server, authentication server and the token granting server makes this technique unique. It can guard against almost all kind of possible threats in the scenario. We take care of a) Biometric template security b) privacy of the user c) trust between user and authenticating server and d) network security related issues[5]. The previous works were generally based on system that provided security by securing the secret key by biometrics. The proposed system does not follow this lead. We divide our task into three steps 1) Registration 2) Authentication 3) Ticket Granting. We will target on strong cryptography for user's original data with the registration server, obviously the authentication should be non-reputable and also the user side attacks and the replay attacks should be taken care of, also in the cases where say the key is compromised. The high performance needed by this level of crypto-biometric system is solved by the token granting system of Kerberos while biometrics take care of some of the security issues that Kerberos has not been able to solve. In the proposed method we look towards the design of a classifier that also helps us to improve the performance of biometrics and we use the randomization scheme for this purpose. Our main improvements in the previous methods are thus benefits of ticket granting and single registration and multiple authentication as explained in during the application.

1.2 The Authentication Protocol

The overall authentication procedure as explained is divided into the following three major steps i .e, registration, authentication and ticket granting. A remote client first registers himself ,i.e., enrolls himself with the first server. Then it authenticates himself with authentication server proceeding further with the tickets and session keys. Figure1 shows how the process is divided into three servers. Alice is our client that is situated remotely and wants to access Bob from there. We are assuming that she has that hardware required for biometrics. Alice remotely invokes and authenticates herself to access bob initially. Alice has to go to three steps

procedure initially. Then once a ticket is obtained from the ticket granting server she may skip the initial steps for certain time period because the ticket will work until it expires. Also note the authentication scheme that we are going to publish. We will follow the modulo- operations i.e., all the operations like , M operation on N are carried out in the encryption domain using the expression(M operation N) mod P. P will be decided by the encryption scheme we employ. Any encryption method can be used but we must be sure it follows homomorphism for certain operations as explained during authentication.



1.3. Registration Server

This is the basic step of registering the user with the main registration server that has the templates of biometrics provided by all users. The registration server is trusted server here. We are here assuming that this third party server i.e., the registration server is already safe enough for us. K is the public key of Alice that it tell the server. During the registration, the client/ Alice sends samples of her biometric data to the registration server, which generates the classifier for Alice. The Biometric sample from Alice to registration server was digitally marked by the client and encrypted using the public key of the server to protect it hence making it secure. Finally what we send to the authentication server is the Alice's identity , her public key, the encrypted parameters and the threshold value. The algorithm1 shows how the process of registration takes place.

I: Registration

1. Alice collects biometric information on the available hardware device.

2. Alice creates the data x from vectors obtained from biometrics.

3. Alice sends the data x with her identity and her public key K to the registration server.

4. Registration server uses x to compute a parameter (w,t) for the user.

5. These parameters are encrypted using Alice's public key: K(w)

6. K(w) that the generated with Alice's identity, public key K and threshold t are sent to authentication server.

7. Alice is notified about the successful registration process.

8. The connection with the registration server is terminated.(Authentication server is approached for further process)

1.4 Authentication Server

Now we need to compute a value w, x that requires multiplication. We can consider simple scalar multiplication and then addition of the values obtained. Over here we are calculating x based on the vector values y_i that we may obtain from the biometrics. For the sake of simplicity we convert this vector to a finite quantity and single x rather than dealing with I values of a single vector derived from the biometric. x can be the mean of the vectors or note that we are using RSA in this method, we know that it follows homomorphism for multiplication[6]. Hence we can compute K(w x)=K(w)K(x) at the server side because of this property of homomorphism that RSA follows. Though we can not add the results to compute the authentication function making it safe. Sending the product answers to Alice to do the addition actually reveals the classifier parameters to the Alice, which obviously we do not want. We are using a randomization technique for this purpose. We generate the parameter r_i by such randomization. It makes sure that the Alice can do the summation computing while it is not able to decipher any information from the product that she can get hands on. The randomization is done in a way such that the server can compute the final sum to be compared with the value of threshold that was decided earlier. The server here carries out all of its computation in the encrypted domain, and hence does not get any information about the biometric data(x) or classifier parameter(w). No one can guess our classifier parameter from the products as they are randomized when multiplied with r_j. The server is able to compute the final sum S because of the imposed condition on r_j and t_jS.

$$\sum_{j=1}^{a}(kjrj) = 1 \qquad (1)$$

This condition as equation 1 is what we have been able to imply to calculations as shown in the next set of equation. We should note that the ability of the server to generate random number her with actually define the privacy of the server. Substituting the equality in the final sum .i.e., S we get the following

$$S= \sum_{j=1}^{a}(kjsj) \qquad (2)$$

$$= \sum_{j=1}^{a}(kjwxrj)$$

$$= (w\ x) \sum_{j=1}^{a} kjrj = w\ x \qquad (3)$$

The process of authentication follows the steps shown in algorithm2. This products expression is the only thing that the server is able to obtain. This will reveal if the biometric belongs to the Alice or not while does not actually reveal the biometric data which mat scarifies the security. It hence provides complete privacy to the user and the biometric data are not stored at any place temporary for template matching. Whatever is revealed is such that if obtained by an untrusted third party cannot be used in a way that it can harm.

II : Authentication

1. Alice computes K(x) and sends to the server

2. Authentication server computes ***a*** random numbers, r_j and k_j such that they satisfy the condition $\sum_{j=1}^{a}(kjrj) = 1$

3. Authentication server computes $K(w\ x\ r_j) = K(w)\ K(x)\ K(r_j)$ (because of the homomorphism)

4. The products obtained are sent to the Alice

5. Alice decrypts the product to obtain $w\ x\ r_j$

6. Alice return $s_j = wx\ r_j$ to the server.

7. Authentication server computes $S= \sum_{j=1}^{a}(kjsj)$ and checks if S>t, if true then server issues to Alice K_{rg} (Alice, K_s) and K_s encrypted with K_A. Where K_A is Alice's key K_s is the session key and K_{rg} is the ticket granting server's key.

1.5 Token Granting Server

TGS or the ticket granting server issues a ticket for the real server(Bob) that Alice wants to access. It provides with the session key K_{AB} between Alice and Bob. Ticket granting adds to performance factor of Kerberos environment with our combination. The Kerberos ticket is a certificate issued by an authentication server, encrypted using the server key. Among other information, the ticket contains the random session key that will be used for authentication of the principal to the verifier , the name of the principal to whom the session key was issued, and an

expiration time after which the session key is no longer valid. The ticket is not sent directly to the verifier , but is instead sent to the client who forwards it to the verifier as part of the application request. Because the ticket is encrypted by the server key, known only by the authentication server and intended verifier, it is not possible for the Alice to modify the ticket without detection. We already have an authentication system that quite space and time consuming and this ticket granting will help us reduce that factor. Although Alice verifies the ID just once with the authentication server, she can contact TGS multiple times for different servers and alternatively access the same server again and again. This benefit covers some part of the tag that we may face in the biometric authentication technique that we are suggesting.

III : Token Granting

1. Now TG server sends two tickets containing

2. K_s (Bob, K_{AB}) and K_b(Alice, K_{AB})

3. Alice sends Bob's ticket timestamp encrypted by K_{AB} i.e., K_{AB}(T) and K_B(Alice , K_{AB}) it received from TG.

4. Bob confirms with Alice by a response such as K_{AB} (T+1) and confirms the success in ticket granting.

Hence once this ticket is granted no need to authenticate again and again and we can thus increase the performance of the biometrics. Figure2 shows how the overall process of authentication is done and summarizes the same. It gives an idea about the steps that are carried on the client's side on the side of three servers.

1.6. Security and Privacy Provided by the Authentication

We analyze all the scenarios to see how the security risks are handled by this authentication technique. First of all the client is to be verified and then the user. The user also has the risk of identity theft due to an unidentified or unsecure server. The database containing the user's information and authentication templates is at risk[3], having the critical information. And finally the network security is to be kept in mind because we will using an unsecure network

a) *The hacker gains access to the template database* – In this case we know that the templates are encrypted by the public key of the respective clients. Hence it's hard to crack the public key algorithm. Moreover if the template is leaked then new can be created from the new public – private key encryption algorithm. Even brute force for this would be almost impossible given the chance of getting a hit.

b) Hacker is in the database server during authentication- hence the hacker has total view of the protocol and how things are working. But the hacker cannot learn anything from the w x or x

values. We can only obtain the S_j values from which it is almost impossible to derive the original biometric data. It may reveal some information about w x but still most part of biometric will remain protective. Even if the hacker is in the server over multiple authentication trial of S_j. However the values of x will slightly change during multiple tries. Now this problem is approximate calculation of w x. thus the two points cover how the server will be protected.

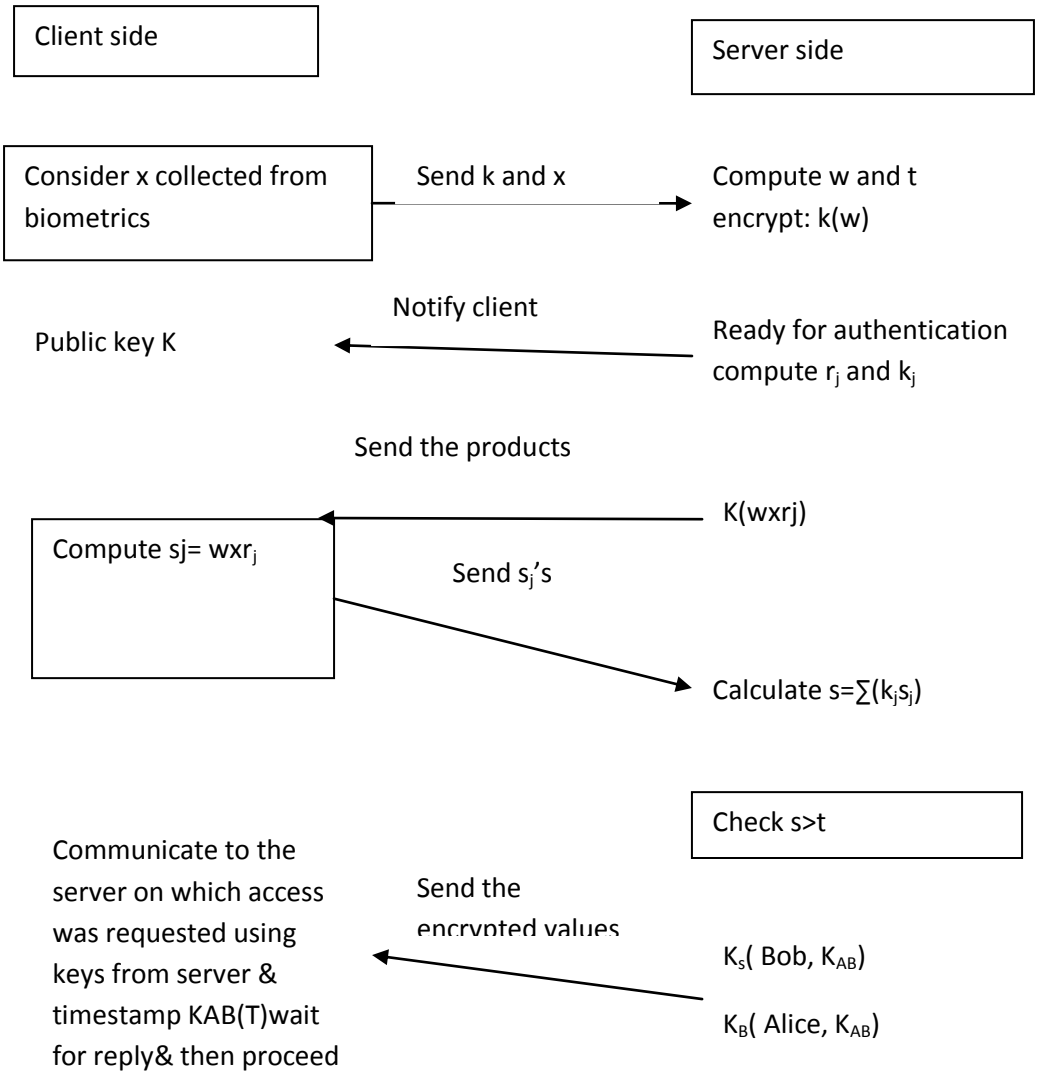


Fig 2: sequential representation of the process

c) *on the client side if the Hacker gains access to the user's biometric or private key*- over here we should note that we Are considering the advantages of not only the biometric authentication but also the security of PKC. He needs private key of the user to understand the biometric information if somehow he gets his hands on the user's biometric. In practice the private key can be stored on a smart card or such a hardware device to increase security and it is very rare to get both these. Even then if the hacker is successful then it will only affect one user and doesn't mean a threat to the whole system.

d) *A Passive kind of attack on the user's computer* – Hacker is present in user's computer during the login process. But the private key is on the hardware and has no direct access to it. He only thus come to know the intermediate computation values. He will have ***a*** values with more variables. An effort equivalent to brute force will be needed in this case. Though multiple login attempts can help the hacker to succeed in this way. Though he would not be able to perform an authentication without the private key.

e) *Network Security*- in this case the network can easily be secured using standard cryptographic methods like symmetric cipher and digital signature. All traffic is encrypted either by clients public key or random number by the server. Thus no information will be deciphered. No replay attack is possible due to the use of random number generation.

f) *risks that Kerberos faces*- Kerberos makes assumptions that the servers are secured and the password guessing attack is not possible. Kerberos implicitly relies on the servers being secure and software being non- malicious [7][8][12].

g) The concerns of being tracked at any case during the authentication and revealing personal information to the intruder are secured by the fact that we use different keys for all the three application servers.

h) *Loose synchronization* – The loose synchronization [7][9] that needs to be done for Kerberos to avoid replay attack is also not a problem when it comes to our model. Replay attack is taken care by itself as explained earlier.

i) *The password theft problem*- This problem that Kerberos authentication is vulnerable to [8][9][11] is solved by the method because of the use of crypto biometric data. Kerberos does not protect against the theft of password for example say through a Trojan horse login program on the user's workstation.

j) *DSA and Fast RSA* – The RSA algorithm applied in the protocol can be substituted by either of the DSA or the Fast RSA, using Montgomery algorithm. This will enhance the security provided by the mechanism by the several folds.

1.7 Applications

Applications using multiple AS, one RS – As we learned from the proposal we can establish a central registration server and once the user is registered the templates are safe with this server. We can now have a lot of remote connections all being authenticated at this single server. We can include a wide area of authentication like a state or even a country. What we need is one time registration on the users end and then through whosever's remote link he is connecting his biometrics can be authenticated. Many companied and organizations can share such servers and save a huge amount of spending on such a secure server and use the benefits of such a scheme. This will be both economical and effective. Though this main server will need to have a very high tolerance and performance curves, but it is achievable. Also it will give a great amount of security that many small firms may not be able to implement due to economic and other reasons. This feature can be enhanced in many ways and gives a lot of possibilities. As shown in Figure3 there is one trusted central server. We have many Authentication (AS) and ticket granting Server(TGS) pairs linked to such central server. The central registration server will need to be something like a server farm. Different authentication servers hold access to the different databases and other application servers that the user needs remotely. He will have to request for the specific authentication server he wants to access and based on that its request will proceed. This has real time applications like unique identity management and mass authentication systems at public places. Various corporate level authentication mechanisms for employees using remote connections through mobile devices. The algorithm 4 shows how this is applied. Keys and variables are explained in the table itself.

IV : Multiple AS. One RS application

1. Users ***1….n*** send their data for registration.

2. User ***I*** calls for authentication.

3. It sends it data collected from biometric samples, ***x*** with its authentication server $\mathbf{A_i}$, both encrypted by the registration server's public key **R**.

4. The registration server computes (W_i, t_i)

5. Encrypts $\mathbf{w_i}$, $\mathbf{t_i}$ and **K** with the $\mathbf{A_i}$'s public key and send to **Ai** for authentication.

6. Now the authentication server takes over and direct communication between the user and authentication server takes place and the relative ticket server $\mathbf{T_i}$ will do the work of assigning the token.

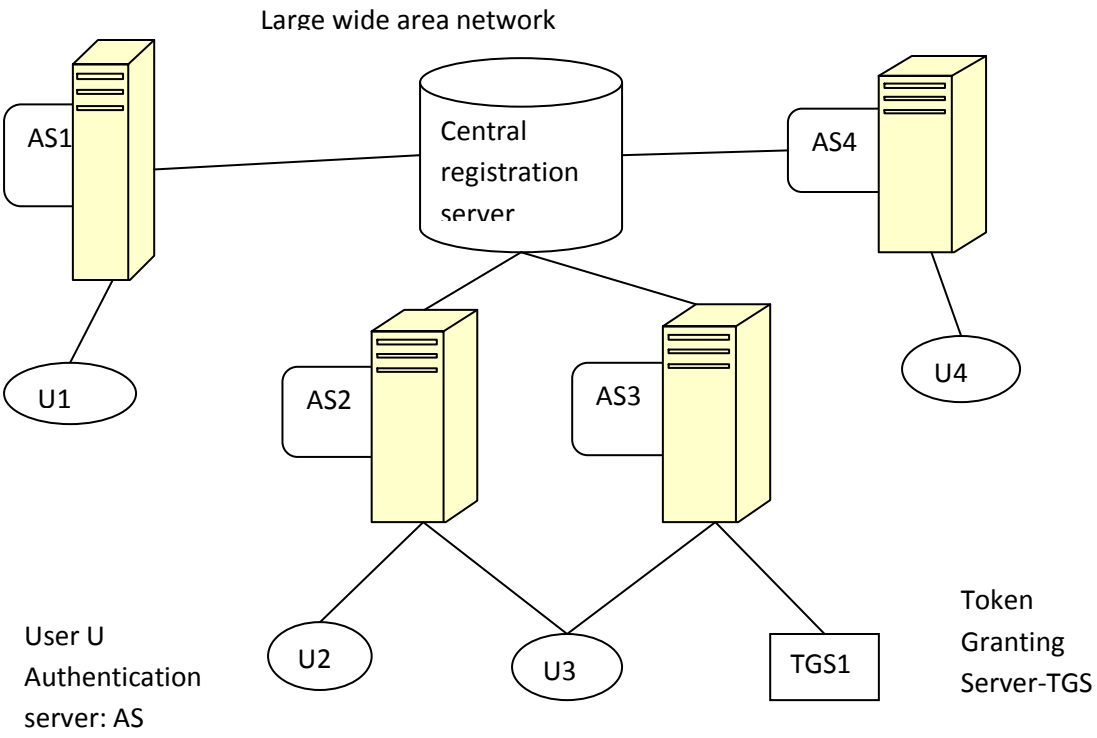


Fig 3 : Application in Large Networks

Conclusion and Future Work

Though the system is very apt for security we can increase the factor of authentication by adding a smart card methodology. This can help us improve the mechanism in many ways. This method gives rise to a very unique idea of a "Central Registration Server" as explained that can act a biometric template matcher for a large number of hardware and efficiency costs can be handled. Also many groups together need just one very secure central server. Once such can be configured. The system is still vulnerable to Denial of Service attack at some points and is one of the drawback that needs to be handled.

References

[1] A.K. Jain , A. Ross and S. Prabhakr " An introduction to Biometric recognition", IEEE trans. Circuit systems , Video Technol., Vol 14, nol,pp 4-20,jan2004

[2]Lawrence O' Gorman, Avaya Labs, Basking Ridge " Comparing Passwords, Tokens, and Biometrics for User Authentication" Proceedings of the IEEE, Vol. 91, No 12,Dec 2003

[3] N.K. Ratha , J.H Connell and R.M Bolle, "enhancing security and privacy in biometric based authentication systems", IBM syst. J. ,vol 40, no.3, pp.614-634,mar 2001.

[4] Upmanyu, M.; Namboodiri, A.M.; Srinanthan, K.; Jawahar, C.V. " Blind Authentication: A Secure Crypto Biometric Verification Protocol" ;; Information Forensic s and Security, IEEE Transactions on. Vol 2. Issue 2, June 2010:pp 255

[5] Cryptography and Network Security, William Stallings 4th edition.

[6] R. Rivest, A. Shamir, and L. Adelman, " A method for obtaining digital signatures and public key cryptosystems", Commun ACM, Vol 21, no 2 , pp. 120-126, 1978.

[7] MIT's online documentation for Kerberos

[8] B. Clifford Neuman and Theodore Ts'o. " Kerberos: An Authentication Service for Computer Netwotks", IEEE Communications Magazine, Volume 32, Number 9, pages 33-38, September 1994.

[9] USC/ISI inline material for Kerberos.

[10] S.P. Miller, C. Neuman., J.I.Schiller, " Kerberos authentication System" Project Athena Technical Plan: Section E.2.1 Cambridge University Press,1987

[11] S. Yeqin, T. Zhongqun, Z. Xiaorong. "Security Analysis of Kerberos 5 Protocol" Computer Knowlwdge and Technology. Vol 6, no 6, pp.1319,june 2010