

Architecture for Authentication in Wireless Differentiated Services using Distributed Substring Authentication Protocol (DSAP)

S. Rajeev¹, S. N. Sivanandam², P. Pradeep³, and Santosh G. Rao⁴

^{1,3}Department of Electronics & Communication Engineering

^{2,4}Department of Computer Science & Engineering
PSG College of Technology, Coimbatore, INDIA

Abstract

All the presently available wireless authentication protocols fail to address some of the key critical security [1] issues like the requirement of subscriber to authenticate every service requested at any given time of connection requisition. This is quite cumbersome on the part of the subscriber. Also, most of these protocols give biased access to System Administrators which can lead to illegal use of the user subscribed services. To offset these drawbacks, and adapt to the convergence of the recent wireless differentiated service technologies, a new authentication protocol "Distributed Substring Authentication Protocol (DSAP)" is proposed. In this protocol, the user authentication information is stored in a distributed format as "Sub-strings" at the Wireless Service Providers end that can only be suitably retrieved based on the password provided by the user. The unique feature that provides strength to this scheme is that a single password can be distributed over the entire network. In addition the change of key in every transaction makes it infeasible, even by known mathematical approaches, to break the scheme. This is guaranteed by a hardware that provides a unique mapping between the subscriber and the service provider and which can be seamlessly connected to any mobile device that is tailored for the reception of a particular service.

Key Terms – DSAP, CAS, Distributed Password Authentication, Mobile authentication, Wireless Differentiated Services.

1. Introduction

For Wireless Differentiated Service Networks [2], [3], [4], [5], [18], [19] Conventional cryptographic techniques [6], [7], [8] concentrate only on securing the transmission channel. Although the channel is highly secure, the database is still vulnerable to internal attacks. These observations lead to the ever-challenging task of authenticating the right user without non-repudiation. The Distributed Substring Authentication Protocol (DSAP) is based on the principle of distributed passwords. A password [9], [10] no longer needs to be maintained in a single system, rather, its fragments are distributed over a wide network. Even if the user gets read access to the database where the passwords are stored, makes it impossible even for a cracker to trace the password fragments. This particular feature guarantees secure and singular access to the right user and as well provides homogeneity of access to all users, including administrators.

In addition this scheme includes the provision of a secondary password that is duly incorporated in the removable chip

connected to the mobile device. This implies the need for both the user password and the chip index number to gain access to the service. This drastically reduces masquerades as the chances of a cracker obtaining both the passwords are highly unlikely.

2. Distributed Substring Authentication Protocol (DSAP)

Abbreviations

CAS	Central Authentication System
CCS	Central Control Server
CIN	Chip Index Number
WS	WorkStation
JMS	Java™ Messaging Service
J2EE	Java™ 2 Enterprise Edition
LDAP	Lightweight Directory Access Protocol
MOM	Message Oriented Middleware
IMN	Input Mapping Number
OMN	Output Mapping Number
IPRT	IP Resolution Table
WJMS	Wireless Java Messaging Service
WDSN	Wireless Differentiated Service Network
SSRT	SubString Resolution Table

The DSAP works on a distributed computing environment (suitably a star topology) at the authentication server end.

2.1 Architecture of CAS

The Central Authentication Service (CAS) Fig. 1 has a Central Control Server (CCS) that forms the heart of the entire network. The workstations in the network are

connected to the CAS by star topology. The choice of the topology hinges on the fact that the CAS must have a control over the entire network. The CCS should be informed of each and every transaction that takes place within the network. The number of workstations connected to the CCS is immaterial as they do not find a place in the protocol parameters which gives the flexibility to vary the number of workstations connected as and when required.

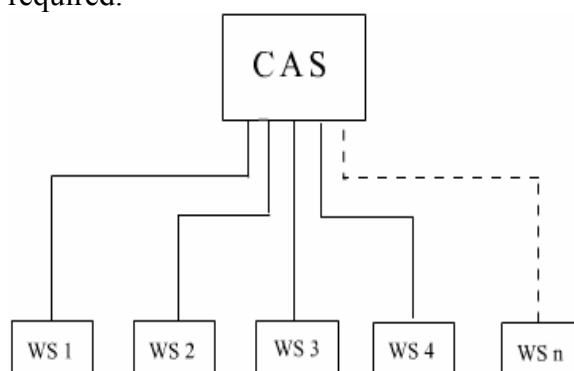


Fig.1.Architecture for Central Authentication Service

2.2 Working of DSAP

The username, password and CIN are sent to the CAS which are encrypted using the key available in the device. Using the user name, CAS identifies the user and retrieves the key from its database. Using this key, it decrypts the user password and CIN. A function in the CAS converts the user name and password to a number called the Initial Mapping Number (S_IMN). This is done in order to get the first input mapping number of the CAS. The structure of the CAS' IPRT is given in Table 1.

S IMN	IP
A3\$yk\N?1#	1
¶♠wiZL=!/↑♦u	12
gXJ<-▼▶●seWH	1

Table 1. IP Resolution Table

Using the IP, the workstation is identified. The S_IMN is given into another function whose parameters include a fraction of the user name and password. The output is the mapping number in the workstation's lookup table namely WS_IMN. The workstation lookup table called the SSRT is as follows.

WS_IMN	SUB STRING	WS_OMN
:+↔↔♫rcUF8)←	routerdevice	↑{m^PB3%y
~paSD6'↓♂ n	567890123456	k]N@l#♣xi[L
E7)♀}o`RC5'	ednetworking	◀♣teWH:,↔☀r

Table 2 Substring Resolution Table

Using the WS_IMN, the sub string and the WS_OMN are retrieved. The sub string is saved in an array, SString. The WS_OMN is input into a function with no other parameters, which gives the next S_IMN. This value gives the next WorkStation's IP and the process is repeated iteratively 'm' times where

m = number of SubStrings.

Each time the sub string retrieved from the workstation, it is concatenated with SString. Finally, the SString is matched with the CIN submitted by the device to authenticate the user.

2.3 DSAP Algorithm

- **Algorithm DSAP_Write**
Writes the User details in to the directory structure.
- **Algorithm DSAP_Read**
Reads the User details from the directory structure.

Policy

```
{
Set MAX_ITER ;
(The MAX_ITER 'should' be greater than or equal to the user name and password)
Set PartString_len;
}
```

Table structures

type

```
celltype = record
    S_IMN : string;
    IP : integer;
end;
server_class = celltype;
```

type

```
celltype = record
    WS_IMN : string;
    PartString : string;
    WS_OMN : string;
end;
WS_class = celltype;
```

```

function DSAP_Write (var server : server_class , var WS : WS_class)
{
    Get U_NAME;
    if(U_NAME:=valid) then
    {
        WriteMessage(“User Name available”);
    }
    Get U_KEY;
        UpdateDatabase(U_NAME,U_KEY);
    }
    else
    {
        WriteMessage(“Try some other name”);
        Call DSAP_Write;
    }
    Get E_U_PASS and E_CIN;
    [U_PASS,CIN]:=Decrypt(E_U_PASS,E_CIN,U_KEY);
    InitNumber:=GenerateInitNumber(U_NAME,U_PASS);
    server.S_IMN:=CalculateS_IMN(InitNumber);
    for j:=0 to MAX_ITER do
    {
    WS.WS_IMN:=CalculateWS_IMN(Server.S_IMN,U_NAME[i mod length(U_PASS)],
    U_PASS[i mod length(U_PASS)]);
    server.ip:=RandomStringGenerator() mod WS_NO;
    MoveControl(server.ip);
        WS.PartString:=WritePartString(WS.WS_IMN,PartString_len,i);
        WS.WS_OMN:= RandomStringGenerator() ;
        WritetoWS(WS.WS_IMN,WS.PartString,WS.WS_OMN,server.ip);
        Server.S_IMN:=CalculateS_IMN(WS.WS_OMN);
    }
    WriteMessage(“User details written into CAS”);
    OldKey:=U_KEY;
        U_KEY:=UpdateKey();
        KeyData:=EncryptU_KEY(OldKey);
        SendToDevice(KeyData);
    }

function DSAP_Read (var server : server_class , var WS : WS_class)
{
    Get U_NAME;
    U_KEY :=RetrieveKey( U_NAME);
    Get E_U_PASS and E_CIN;
    [U_PASS,CIN]:=Decrypt(E_U_PASS,E_CIN,U_KEY);
    InitNumber:=GenerateInitNumber(U_NAME,U_PASS);
        server.S_IMN:=CalculateS_IMN(InitNumber);
    for j:=0 to MAX_ITER do
    {
    WS.WS_IMN:=CalculateWS_IMN(Server.S_IMN,U_NAME[i mod length(U_PASS)],

```

```

    U_PASS[i mod length(U_PASS)];
    server.ip:=GetWSIP(WS.WS_IMN);
MoveControl(server.ip);
    WS.PartString:=GetPartString(WS.WS_IMN);
    WS.WS_OMN:=GetWS_OMN(WS.WS_IMN);
SendToServer(WS.PartString,WS.WS_OMN);
StringCat(SString,WS.PartString);
    Server.S_IMN:=CalculateS_IMN(WS.WS_OMN);
}

if SString=CIN then
{
    WriteMessage("User Authenticated.Proceed to access user services.");
    OldKey:=U_KEY;
U_KEY:=UpdateKey();
    KeyData:=EncryptU_KEY(OldKey);
    SendToDevice(KeyData);
}

else
    WriteMessage("Wrong User.Close Connection.");
}
end of the algorithm

```

2.4 Table Structures in DSAP

Table structure of the Service provider's database:

User Name	Current User Key	Services Accessible
Neo	12A^!Slc22rrUF32	1,3,14
Max	53tt@FG9(Tkte	1,7,12

Table structure of the chip manufacturer's database:

Chip Serial Number	Manufacturer Key
8352t45w345	5hdUYE48rkt89
%&%erR6843	Rjt%ou84%2u4J

3. Chip Manufacturing Requirements

The following chip requirements will aid in the effective working of DSAP

1. The manufacturer initially enters a key into the chip which can be re-written. This detail is entered into the manufacturer's database along with its serial number.
2. The serial number of the chip is made unique for every chip.
3. The service provider enters a large string into the chip called the CIN. Any data being sent from the device using the chip to the CAS is encrypted using the key available in the chip.
4. When the user registers himself at the service provider, the service provider simply enters the Serial number of the chip into the program which

looks up into the manufacturer's database and updates the corresponding key into the service provider's database containing the user details.

5. The chip can be any 'Smart Card' which can be connected to the device and has compliance with the above mentioned constraints.

4. Prototype Implementation of DSAP

A simulation model for DSAP was constructed for the Architecture in Fig. 2.

The communication between the mobile user and the Service Provider uses a Asynchronous mode of communication using WJMS (Wireless Java Messaging Service), assuming that only the physical media being wireless, the simulation model uses PC workstations for simulation. The control flow for authentication is shown in Fig. 3.

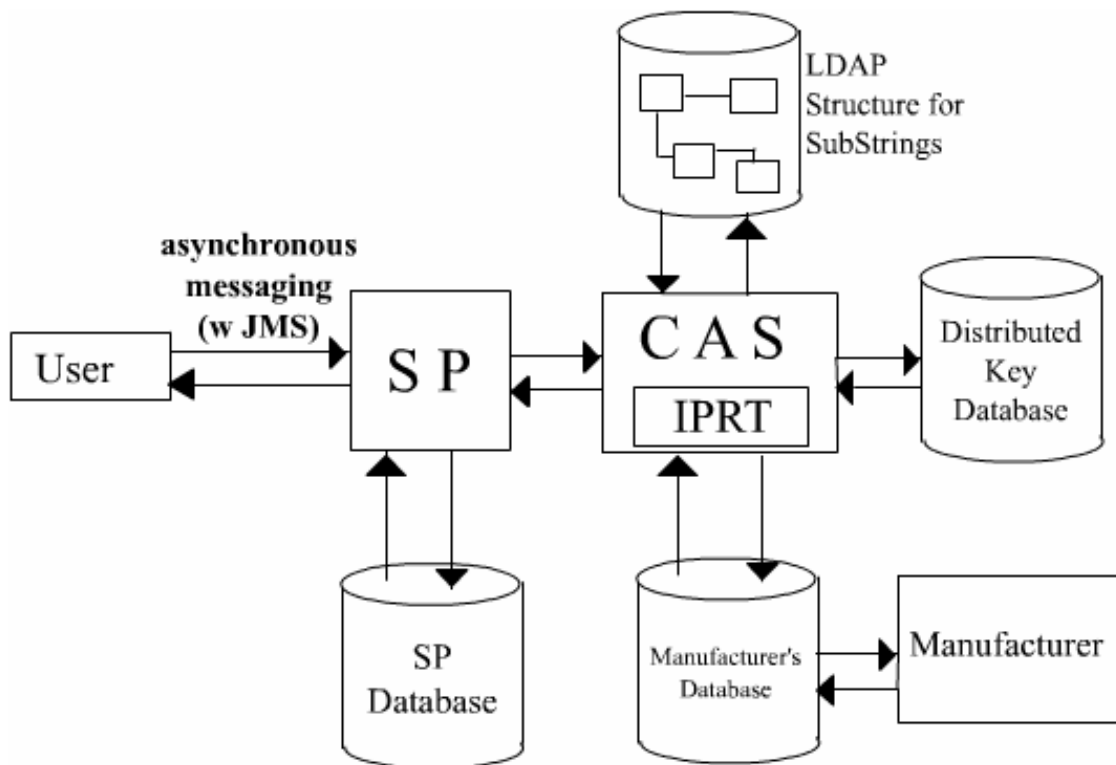


Fig. 2. Architecture for Wireless Differentiated Service Network

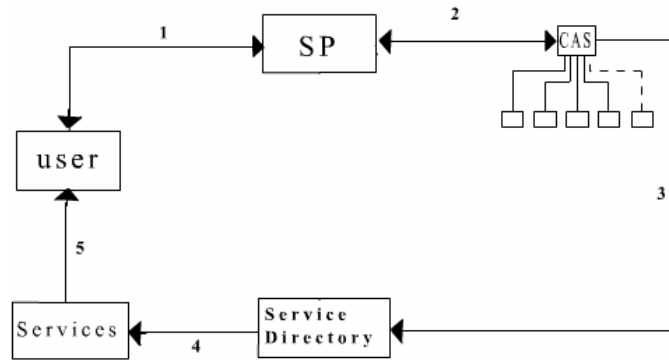


Fig. 3. Control Flow Diagram

- 1 - User sends the user name and password to the SP.
- 2 - Service Provider forwards the details to the CAS. After the user is authenticated, the CAS sends a report to the SP. The wrong user is disconnected.
- 3 - If the user is valid, the CAS forwards the request to the Service Directory.
- 4 - Service Directory identifies the user services.
- 5 - Services are now gives to the user for access.

As the protocol focuses primarily on authentication of mobile devices [11], actual implementation must take into consideration, the resource constraints [12] at the client side like memory, processor etc. This, in fact limits the choice of encryption algorithms that can be applied.

The following parameters determine the choice of implementation:

4.1. Choice of programming language

Mobile Devices need to communicate over multiple communication platforms. This requirement enforces them to make use of common methods and protocols.

4.2. Proper messaging mechanism

The CAS should have the feature of upward scalability. This dictates the need for Asynchronous messaging where the CAS need not spend precious time in establishing

the connection. An optimum solution is obtained by using wJMS (Wireless Java™ Messaging Service) [13] which is an API (Application Programming Interface) in the J2EE (Java™ 2 Enterprise Edition). The JMS also serves as a good Message Oriented Middleware (MOM)[14].

4.3. Suitable data retrieval system

LDAP (Lightweight Directory Access Protocol) [15],[16] is the most desirable data retrieval system. It offers compatibility supporting a Directory Enabled Network (DEN)[17]. As DSAP does not require transaction processing and heavy retrieval, a lightweight directory is much suited. It also supports for storing user policy in the directory structure.

The Simulation Result (Fig. 4) is done for all SSRTs and are plotted separately for number of SubStrings in each CIN, showing the Probability of hack in each case.

The function used to calculate the Probability of hack = $\left(\frac{1}{x}\right)^r$

where x is the total number of substrings in the various SSRTs distributed over the network

r is the number of SubStrings in each CIN

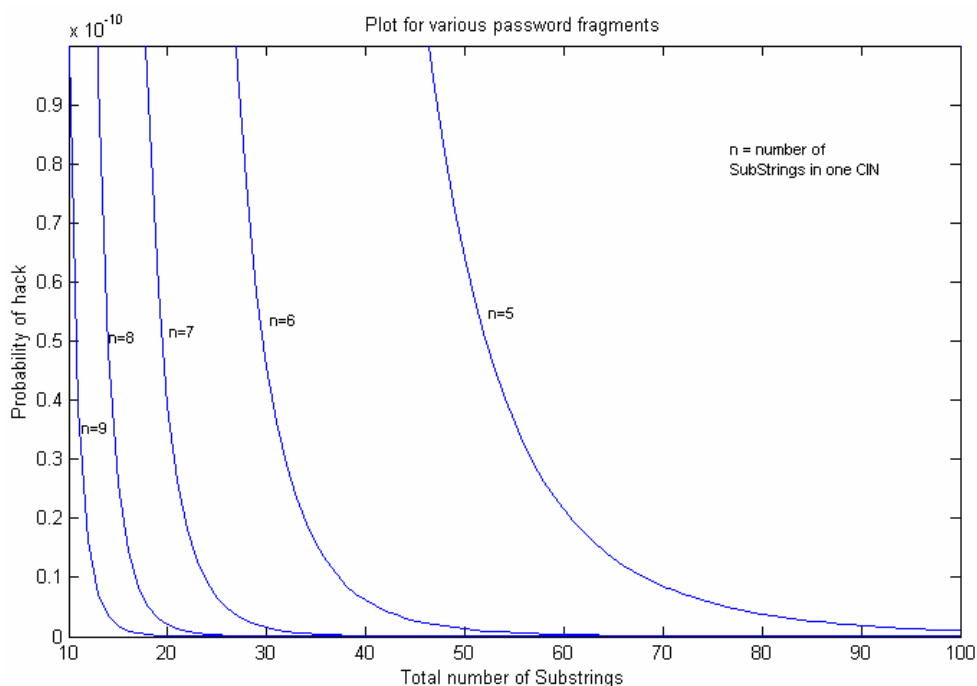


Fig. 4. Plot for Password Fragment Sizes

5. Advantages of DSAP

- There are no lookup tables of passwords; hence unauthorized entry by users who may be capable of illegally accessing the table is completely eliminated.
- Even the System Administrator of CAS cannot access the user password and CIN without knowing the actual user name and password.
- The use of CIN allows the user access from any compatible device which has provision to use the chip.
- The change in the encryption key for every transaction eliminates chance of a crypto-analytic attack by identifying multiple messages.
- The workstations used need not be dedicated systems, as their task remains to fetch the server's queries for the substring.
- The Protocol offers flexibility to increase the number of Workstations without major changes in the policy.
- As the algorithm has less time and space complexity, quick authentication is resulted.
- The star topology ensures that all the data sensitive computations are done at the CCS and no data could be processed outside.

- Asynchronous messaging implementation of JMS and LDAP gives the user, easier access to the CAS.
- From the chip requirements, it is clear that no one has complete knowledge of the chip details. This adds-on to the amount of security provided by the chip.

6. Conclusion

This authentication protocol, DSAP, highlighting the need for a better architecture in WDSN that counters attacks even to the database containing the passwords was designed and implemented. From the implemented protocol simulation test results it was observed that no presently available architecture for wireless communication provides the advantages mentioned. Future work on the Protocol is to design a policy framework that the system administrator can use to provide various levels of security, depending on sensitivity of data, and to generate new CINs every time the user logs on to the CAS.

References

- [1] Miller K.M, "Facing the Challenge of Wireless Security", *IEEE Computer*, July 2000.
- [2] S. Blake, D. Black, M. Carlson, E. Davies, Z.Wang and W.Weiss, "An architecture for differentiated services", *IETF RFC 2475*, 1998.
- [3] K. Kilkki, "Differentiated Services for the Internet", Macmillan Technical, Indianapolis, IN, 1999.
- [4] Mahadevan I., Sivalingam K.M., "Architecture and Experimental Framework for Supporting QoS in Wireless Networks Using Diferentiated Services", *ACM, Mobile Networks and Applications*, Vol.6, Issue 4, August 2001, pg. 385 – 395, ACM Press.
- [5] Mahadevan I. and Sivalingam K.M., "Differentiated Services for Wireless Networks: Architecture and Experimental Implementation", *International Conference on Computer Communication Networks*, 1999.
- [6] Ronald Rivest, Adi Shamir and Len Adelman "A method for obtaining Digital Signatures and Public Key Cryptosystems" *ACM* 1978
- [7] Koblitz N, "Elliptic Curve Cryptosystems", *Mathematics of Computation* Vol. 48, pp 203-209, 1987
- [8] Jeffrey Hoffstein, Jill Pipher, Joseph H. Silverman, "NTRU: A Ring Based Public Key Cryptosystem", *Algorithmic Number Theory (ANTS III)*, Portland, OR, 1998.
- [9] N. Haller, C. Metz, P. Nesser, M. Straw "A One-Time Password System", *RFC2289*, 1998
- [10] J. Archer Harris, "OPA: A One-Time Password System", *International Conference on Parallel Processing Workshops (ICPPW'02)*, 2002
- [11] Boertien N, Janssen W, Middelkoop E.M, "Virtuele Haven- Multichannel Architecture, Enschede", CMG Telematica Institute, 2001.
- [12] Boertien N, Middelkoop E.M, "Virtuele Haven-Authentication in Mobile Applications", CMG Telematica Institute, 2002
- [13] Nigel Thomas, "Building Asynchronous Applications Using Java Messaging", White Paper, SpiritSoft, Inc 2002.
- [14] Vivien Quéma, Luc Bellissard, Philippe Laumay, "Application-Driven Customization of Message-Oriented Middleware for Consumer Devices", Workshop on Software Infrastructures for Component-Based Applications on Consumer Devices, 2002.
- [15] Heinz Johner, Larry Brown, Franz-Stefan Hinner, Wolfgang Reis, Johan Westman, "Understanding LDAP", White Paper, IBM Corporation.

- [16] Rob Weltman, Tony Dahbura, *LDAP Programming with java*, Addison-Wesley, 2002
- [17] Li Wei, “Directory Enabled Services in IP RAN Base Station”, Masters thesis, Graduate School of Informatics, Kyoto University, 2001.
- [18] S. Rajeev, S. N. Sivanandam, “Policy Based Provisioning System for Wireless Differentiated Service Networks Using Graph-Node Coloring”, International Conference on Wireless Communication Networks (ICWCN), 2003.
- [19] S. N. Sivanandam , S. Rajeev, D. Kiran Swaroop, “Distributed Computing and Network Provisioning In Wireless LANs”, International Conference on Wireless Communication Networks (ICWCN), 2003.

—