Kerberos Assisted Authentication in Mobile Ad-hoc Networks

Asad Amir Pirzada and Chris McDonald

School of Computer Science & Software Engineering,
The University of Western Australia
35 Stirling Highway, Crawley, W.A. 6009, Australia.

email: {pirzada,chris}@csse.uwa.edu.au

Abstract

An ad-hoc network comprises mobile nodes that cooperate with each other using wireless connections to route both data and control packets within the network. As the low transmission power of each node limits its communication range, the nodes must assist and trust each other in forwarding packets from one node to another. However, this implied trust relationship can be threatened by malicious nodes that may fabricate, modify or disrupt the orderly exchange of packets. Security demands that all packets be authenticated before being used.

In this paper we present Kaman, Kerberos assisted Authentication in Mobile Ad-hoc Networks, a new pure-managed authentication service for mobile ad-hoc networks. Kaman is based on the time-tested and widely deployed Kerberos protocol, and provides secure extensions to support the more challenging demands of ad-hoc networks. Kaman migrates a number of features from the traditional, wired Kerberos environments to the ad-hoc environment, including the prevention of node identity forgery, the detection of replay attacks, establishment of secure channels, mutual endpoint authentication, and the secure distribution of provisional session keys amongst replicated servers. Kaman has been specifically designed for hostile environments, in which the presence of malicious nodes and the likelihood of physical node capture is relatively high.[1]

Keywords: Authentication, Security, Ad-hoc, Networks

1 Introduction

Authentication is one of the major security issues affecting the wired and the wireless network community. It is generally accomplished in two ways: direct and indirect authentication (Fox and Gribble 1996). In direct authentication, two parties use pre-shared symmetric or asymmetric keys for verifying each other and the flow of data between them. In indirect authentication, a trusted third party, i.e. a Certification Authority, is made responsible for certifying one party to another party. Most of the secure routing protocols developed for ad-hoc networks, rely on indirect authentication mechanisms using public key infrastructures (PKI) to authenticate communicating nodes (Pirzada and McDonald 2003). PKI, although a very secure system, is based on asymmetric cryptography and hence requires excessive processing and communication resources (Hu, Perrig and Johnson 2002). This resource hungry feature makes PKI based systems more susceptible to Denial of Service attacks. In contrast, Kerberos (Kohl and Neuman 1993) is a symmetric key based indirect authentication mechanism. The security and effectiveness of Kerberos has been proven over a long period of time. It includes many significant features that are non-existent in other authentication mechanisms, including:

1. Prevention of forgery of client or server identity
2. Detection of replay attacks
3. Establishment of secure channels between endpoints
4. Mutual authentication
5. No flow of passwords on the network

In our scheme we use a variant of Kerberos for secure key exchange in ad-hoc networks. Our scheme is applicable to ad-hoc networks in an open-managed environment where there is option for bootstrapping the Kerberos servers and clients. The scheme is modular, secure and reliable due to its distributed architecture.

We present some relevant previous work in Section 2. In Section 3 we describe our proposed scheme in detail and in Section 4 we describe different attacks possible in an ad-hoc network environment and analyze our protocol against them. Concluding remarks are presented in Section 5.

2 Previous Work

2.1 Charon

Charon (Fox and Gribble 1996) provides indirect authentication and secure communication between a lightweight PDA client and a Kerberos Server using an intermediary system called the proxy. Charon uses the Proxy to communicate with the Kerberos Key Distribution Center and the Kerberos Ticket Granting Server to save the computation resources of the client. It operates using two distinct phases. In the first phase known as the Handshake, the client authenticates itself to the proxy and establishes a secure channel with it. In the

[1] Copyright (c) 2004, Australian Computer Society, Inc. This paper appeared at the *27th Australasian Computer Science Conference*, The University of Otago, Dunedin, New Zealand. Conferences in Research and Practice in Information Technology, Vol. 26. V. Estivill-Castro, Ed. Reproduction for academic, not-for profit purposes permitted provided this text is included.

second phase called the Service Access, the proxy accesses the Kerberos servers on the client's behalf for authentication services. The scheme, although very effective for low resource clients, cannot be used for ad-hoc networks where simultaneous access to three servers (Proxy Server, Authentication Server and Ticket Granting Server) may not be possible in every scenario. This scheme is also subject to latency delays in the authentication mechanism.

2.2 M-PKINIT

M-PKINIT (Harbitter and Menasce 2001) is an amalgamation of the Public Key based Kerberos PKINIT (Tung, Neuman and Wray 2001) and Charon for use in mobile networks. It adds Public Key cryptography to the Kerberos protocol to simplify the key management (from the Kerberos perspective) and the ability to use the existing public key certification infrastructures. It aims to enhance the security of the Kerberos protocol by using a minimal number of public key operations along with a proxy for load distribution. This scheme incorporates asymmetric cryptography which in turn slows the overall authentication mechanism. It also requires simultaneous access to three servers for initial authentication, which we have already deemed limiting in such an improvised environment.

2.3 Distributed Public-Key Model

The Distributed Public-Key Model (Zhou and Haas 1999) makes use of threshold cryptography to distribute the private key of the Certification Authority over a number of servers. An (n, t+1) scheme allows any t+1 servers out of a total n servers to combine their partial keys to create the complete secret key. Similarly it requires that at least t+1 servers be compromised to acquire the secret key. The scheme is quite robust but has a number of factors that limit its application to pure ad-hoc networks. Primarily, t+1 servers may not be accessible to any node desiring authentication at any instance and secondly asymmetric cryptographic operations are known to drain precious node batteries.

2.4 PGP Model

In the Pretty Good Privacy Model (Garfinkel 1995) all users act like independent certification authorities and have the capability to sign and verify keys of other users. PGP breaks the traditional central trust authority architecture and adopts a decentralized "web of trust" approach. Each individual signs each other's keys that help build a set of virtual interconnecting links of trust. PGP attaches various degrees of confidence levels from "undefined" to "complete trust" to the trustworthiness of public-key certificates and four levels of trustworthiness of introducers from "don't know" to "full trust". Based on these trust levels the user computes the trust level of the desired party. PGP is suitable for wired networks where a central key server can maintain a database of keys. However, in ad-hoc networks, the inclusion of a central key server creates a single point of failure and also requires uninterrupted access to the nodes. The other option, as in PGP, is for each node to store a subset of the public keys of other users using a subset of the trust graph (Hubaux, Buttyan and Capkun 2001) and to merge these graphs with graphs of other users in order to discover trusted routes. This scheme involves extensive computation and memory requirements and is considered restrictive for ad-hoc networks.

3 Kaman

We present here a secure authentication scheme, Kaman, for ad-hoc networks. In Kaman we have multiple Kerberos servers for distributed authentication and load distribution. As mobile nodes are susceptible to physical possession, in Kaman only the users know the secret key or password and the servers know a cryptographic hash of these passwords. All Kaman servers share a secret key with each other server. In Kerberos, the server is usually a single point of failure as it has the repository of hashed passwords of all users. In Kaman all servers periodically, or on-demand, replicate their databases with each other. In Kerberos, physical compromise of a server means complete failure of the authentication system. In Kaman, an optional availability check mechanism can even disable a captured server. We use an election based server selection mechanism, so the non-availability of a server after some time initiates the server election process. Whenever unicast or multicast communication is required among nodes, the nodes approach the Kaman servers whom in turn allocate a session key for their secure authentic communication.

3.1 Assumptions

In Kaman we have made the following assumptions:

1. All users have a secret key or password known only to them
2. All servers know the hashed passwords of all the users
3. All servers share a secret key with each other server

3.2 Operation

The general operation of Kaman is shown in Figure 3.2. Whenever a node **C1** desires to undertake secure communication with another node **C2** it sends a request to one of the Kaman Servers **S** (1). The server creates a ticket, which contains the session key for the requested communication and sends it back to the client **C1** (2). The client **C1** in turn sends this ticket to the target client **C2** with which communication is desired (3). The recipient client **C2** acknowledges the ticket (4) and a secure session is established between the two clients using the session key provided by the server. The servers also exchange data through an encrypted channel, however, they don't require the ticket as they already possess the session key.

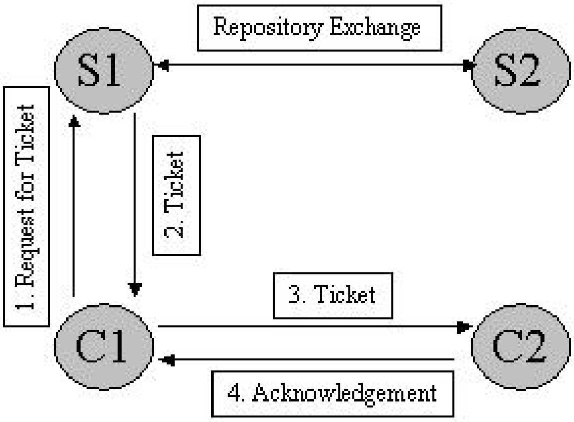

Figure 3.2: Operation of Kaman

3.3 Initialisation

During the initial configuration there may exist only a single server with the repository. In this repository each user's ID and their hashed password is stored along with a priority and a lifetime. All possible users who are trustworthy enough are allocated higher priority than users with lower trust. Based on the usage scenario, the lifetimes of the users are allotted, with lifetimes assigned inversely proportional to the perceived hostility of the network environment. If the network environment is friendly the password lives may be long but in case of insecure environments these lives may be shortened. The repository basically contains four fields: UserID, Password, Priority and Lifetime. The format of the repository is shown in Table 3.3:

User ID	Hashed Password	Priority	Lifetime
Server1	0010101010101010	9	3600
Server2	1010101110011110	8	3000
Client1	1001110111111001	5	1200
Client2	0101010111010101	6	2000

Table 3.3: Format of the Kaman Server Repository

3.4 User Authentication

Kaman uses a modified version of the Kerberos 5 protocol (Stallings 2003) for authentication in ad-hoc networks. A Ticket Granting Server (TGS) is used in the Kerberos 5 protocol to issue tickets to users who have already been authenticated by the Authentication Server (AS). We use a modified version of this scheme known as a four-pass Kerberos protocol (Carman, Kruus, and Matt 2000) that eliminates the use of a TGS so as to make it more viable for use in ad-hoc networks.

Whenever a node wants to establish a secure connection with another node it approaches the Authentication Server and follows the protocol as shown in Table 3.4:

1. Client1 → Server

Options, ID_{C1}, ID_{C2}, Times, Nonce

2. Server → Client1

ID_{C1}, $Ticket_{C2}$, $\{K_{C1,C2}$, Times, Nonce, $ID_{C2}\}K_{C1}$

3. Client1 → Client2

Options, $Ticket_{C2}$, $Authenticator_{C1}$

4. Client2 → Client1

$\{TS, Subkey, Seq\#\}K_{C1,C2}$

$Ticket_{C2} = \{Flags, K_{C1,C2}, ID_{C1}, AD_{C1}, Times\}K_{C2}$

$Authenticator_{C1=}\ \{ID_{C1}, TS\}K_{C1,C2}$

Options: Used to request that certain flags be set in the returned ticket

Times: Used to specify the start, end and renewal time settings in the ticket

Flags: Status of the ticket

Nonce: A random value used as a pseudo-unique transaction identifier to avoid replay attacks

Subkey: Choice for another encryption key for this session instead of $K_{C1,C2}$

Seq#: Starting sequence number to detect replays

ID_{C1}: Identity of Client1

ID_{C2}: Identity of Client2

AD_{C1}:Network Address of Client1

K_{Cn}: Encryption key based on hashed password of user n

$K_{C1,C2}$: Session key between Client1 and Client2

TS: Informs of time when this authenticator was generated

Table 3.4: Kaman Client-to-Client Authentication

During the authentication service exchange, Client1 requests a ticket from the Kaman server for further communication with Client2. The server first checks that if Client1 and Client2 have a valid lifetime associated with their user IDs. If they have valid lifetimes then the server responds by providing Client1 a ticket to access Client2. This ticket, along with other values, contains a session key for communication between Client1 and Client2. Client1 then passes this ticket to Client2 for establishing secure communication using the session key. Client2 acknowledges receipt of this ticket by sending a Time Stamp back to Client1.

In contrast to the original Kerberos 5 protocol, in Kaman we only use the Authentication Server to provide the requesting client with a ticket for communication with another client. This mechanism provides us with the following benefits in an ad-hoc network environment:

1. Single server needs to be accessed
2. Faster authentication
3. Reduced client side processing
4. Reduced battery consumption

3.5 Key Revocation

The Kaman server only authorizes users until the expiry of their password lifetimes. The ticket automatically expires when this period expires. According to a pre-agreed set of rules, the session between two clients is also terminated when the ticket expires. All clients requiring extended authorization must apply for a new password before expiration of the last ticket. The protocol for this extension is shown in Table 3.5:

1. Client1 → Server

Options, ID_{C1}, Nonce

2. Server → Client1

ID_{C1}, {K_C, Times, Nonce, ID_{C1}}K_{C1}

Options: Used to request that certain flags be set in the returned ticket

Times: Used to specify the start, end and renewal time settings in the ticket

Nonce: A random value used as a pseudo-unique transaction identifier to avoid replay attacks

ID_{C1}: Identity of Client1

AD_{C1}: Network Address of Client1

K_{C1}: Encryption based on users old hashed password

K_C: New password for Client1

Table 3.5: Extended Key Exchange

3.6 Server Elections

Based on the area of coverage and range of the nodes, there may need to exist one or more servers. Server elections are triggered in the following situations:

1. When the number of servers available increases or decreases
2. When the server lifetime expires
3. When a server fails the optional availability check mechanism

Servers periodically use secure BEACON and ECHO packets to discover the availability of other servers (Corson, Papademetriou, Papadopoulos, Park, and Qayyum 1999). In the event that a server is not available either due to its wireless transmission range, geographical position or if its lifetime has expired an election is triggered. During these elections, the servers check their repositories for users with the highest priority levels. If more than one node has the same priority then their lifetimes are taken into account. These nodes with the greatest priority and lifetime are automatically upgraded to servers by securely transferring the repository to them. Similarly if the number of servers is increased (an unavailable server comes up) the server with the lowest priority and lifetime is downgraded to a client. The periodicity of these elections is dependent on the area of dispersal, node density and severity of the situation.

3.7 Replication of Repository

Regular replication of the repository is vital for the overall synchronization and continual operation of the authentication mechanism. Replication ensures that the accounts database is spread over a number of servers so as to safeguard from node capture and compromise. It also ensures that all user accounts are kept up to date by reflecting any changes to all the servers. This mechanism ensures that all accounts that have been added, modified or revoked since the last replication are updated in the repository in a timely and orderly manner. Each replication is associated with a replication sequence number and can be either periodic or implemented using a push-pull mechanism (Fife and Gruenwald 2003). The frequency of replication is dependent upon the area of dispersal, node density, situation severity and energy constraints (Gruenwald, Javed and Gu 2002). In order to maintain global consistency of the user database it is recommended that the Extended Static Access Frequency (E-SAF) method suggested by Hara (2003) be employed. The E-SAF aims at lowering traffic overhead by taking into account the data access probability and replica relocation period of frequently accessed data.

1. Server1 → Server2

{TS, R, Seq# }$K_{S1,S2}$

Seq#: Replication Sequence number

R: Repository

$K_{S1,S2}$: Session key between Server1 and Server2

TS: Informs of time when this replication was carried out

Table 3.7: Repository Replication

If R is the current repository of Server S1, TS is the replication time and N is the current replication sequence number, then the replication package would be {TS, R, N)$K_{S1,S2}$.

3.8 Optional Availability Check

The optional availability check mechanism is used for insecure environments where chances of node capture are relatively high. This mechanism may be based on biometric devices for user authentication or may be a simple prompt for a password. When this availability option is selected the user is either prompted periodically or as per demand of other servers for a password or a secret code so as to assure that the node is in safe hands. In case the current user is unable to provide the password, the node informs the other Kaman servers regarding its status, deletes the user database repository (in case of a server) or the user's account (in case of a client) and stops functioning. The frequency of this security prompt is dependent on the severity of the situation in which the ad-hoc network has been established.

4 Security Analysis

In this section we discuss how Kaman defies certain attacks possible in an ad-hoc network. As discussed earlier, the basis of a security infrastructure is primarily dependent on the initial key exchange providing authentication. The other security services like confidentiality, integrity and non-repudiation all rely on the accurateness of the authentication service. The strengths and weakness of Kerberos have been studied in detail by Bellovin and Merritt (1991) and the protocol has been found to be robust against a number of attacks. If the initial session key exchange using Kaman is secure then the proposed solution is to use encrypted IP packets with MAC layer broadcasts for route discovery and maintenance. Once secure routes have been established then the data packets need to be encrypted between endpoints. The fields being encrypted in IP packets are shown in Figure 4.

Routing Packets

MAC	IP	DATA
Clear	Encrypted	Encrypted

Data Packets

MAC	IP	DATA
Clear	Clear	Encrypted

Figure 4 : Encryption of Routing and Data Packets

In the following sections, we describe how the Kaman protocol can protect against a number of frequently cited attacks (Dahill, Levine, Royer and Shields 2002 & Hu, Perrig and Johnson 2002) against ad-hoc networks.

4.1 Active Attacks

4.1.1 Modification Attacks

Attacks using modification are generally targeted against the integrity of routing computations and so by modifying routing information an attacker can cause network traffic to be dropped, redirected to a different destination, or to take a longer route to the destination increasing communication delays.

4.1.1.1 Black Hole or Grey Hole attack

The black and grey hole attacks are launched by modifying the routing packets to point to a particular node, which in turn drops or forwards packets at its own discretion.

4.1.1.2 Routing Loops

Routing loop attacks modify routing packets in such a manner that the packets traverse a cycle, and don't reach their intended destination.

4.1.1.3 Increase in route length

In this attack routing packets are modified to create excessively long routes to the destination, typically by including other compromised nodes.

4.1.1.4 Battery Exhaustion Attack

In this attack routing packets are modified in such manner that the network traffic is concentrated towards a single target node. This node's battery will be consumed in receiving excess packets.

Solution: The nodes of an ad hoc network that are executing the Kaman protocol can use the session keys for encrypting the traffic flow of data and control packets. Thus, including the hash of the message contents in every transmitted packet, guarantees the integrity of the contents along with confidentiality.

4.1.2 Fabrication Attacks

Fabrication attacks are performed by generating false routing messages. These attacks are difficult to recognize as they are received as genuine routing packets. The rushing attack is a typical instance of malicious attacks using fabrication. This attack is targeted against on-demand routing protocols that use duplicate containment at each node. An attacker quickly disseminates routing messages throughout the network, suppressing any later genuine routing messages when nodes drop them due to the duplicate suppression. Similarly an attacker can nullify a working route to a destination by fabricating routing error messages claiming that a neighbour can no longer be contacted.

Solution: If the network is implementing the Kaman protocol then the authenticity of the received control and data packets can be verified using the session keys. As the session keys are unique, fabricated packets can easily be verified and hence discarded.

4.1.3 Impersonation Attacks

A malicious node can launch many attacks in a network by masquerading as another node (spoofing). Spoofing occurs when a malicious node misrepresents its identity by altering its MAC or IP address in order to alter the vision of the network topology that a benign node can gather. As an example, a spoofing attack allows the creation of loops in routing information collected by a node with the result of partitioning the network.

Solution: The encryption of all point-to-point traffic indirectly ensures the verification of packets, as the session keys are only held by the previously authenticated end points. As a consequence, the legitimacy of all packets is automatically verified during the decryption phase, ensuring that any packets that were spoofed are discarded.

4.2 Passive Attacks

In passive attacks the attacker does not perturb the routing protocol. Instead, it only eavesdrops on the routing traffic and tries to extract valuable information like node hierarchy and network topology from it. For example, if a route to a particular node is requested more frequently than to other nodes, the attacker might expect that the node is significant for the operation of the network, and disabling it could bring down the entire network. Likewise, even when it might not be possible to isolate the exact position of a node, one may be able to find out information about the network topology by analysing the contents of routing packets. This attack is virtually impossible to detect in the wireless environment and hence also extremely difficult to prevent.

Solution: As all the routing and data packets are encrypted before transmission, it is quite challenging for the passive eavesdropper to use the information effectively in the brief time span of the ad-hoc network.

5 Conclusion

In this paper we have presented a secure key exchange scheme for use in ad-hoc networks. Our scheme is based on the reliable Kerberos protocol. We have introduced certain changes to the original protocol for its viability for ad-hoc networks. For inter-client communication, each node approaches one of the servers for a session key. The server generates the key and encapsulates it in a ticket and sends it to the requesting client. The client can then use this ticket to create a secure session with the intended party. Due to the mobility and short range of the nodes, we have introduced measures like replication and elections, so as to ensure maximum connectivity of the clients with the servers. To protect against physical capture and tampering we have introduced an optional availability check feature that minimizes the risk of malicious attacks from within the network.

6 References

Bellovin, S. M. and Merritt, M. (1991): Limitations of the Kerberos authentication system, *Proc. of USENIX Winter Conference*, 253-267.

Carman, D. W. Kruus, P. S. and Matt, B. J (2000): Constraints and approaches for distributed sensor network security, Technical Report 00-010, NAI Labs.

Corson, S. Papademetriou, S., Papadopoulos, P., Park, V. and Qayyum, A. (1999): An Internet MANET encapsulation protocol (IMEP) specification, Internet draft (work in progress).

Dahill, B., Levine, B. N., Royer, E. and Shields, C. (2002): A Secure Routing Protocol for Ad Hoc Networks, *Proc. of International Conference on Network Protocols (ICNP)*, 78- 87.

Fife, L. D. and Gruenwald, L. (2003): Research issues for data communication in mobile ad-hoc network database systems, *ACM SIGMOD Record*, 32(2):42-47.

Fox, A. and Gribble, S. D. (1996): Security on the move: Indirect Authentication using Kerberos, *Proc. of the Second Annual International Conference on Mobile Computing and Networking*, 155-164.

Garfinkel, S. (1995): *PGP:Pretty Good Privacy*, O'Reilly & Associates, Inc.

Gruenwald, L., Javed, M., and Gu, M. (2002): Energy-Efficient Data Broadcasting in Mobile Ad-Hoc Networks, *Proc. of the International Database Engineering and Applications Symposium*, 64-73.

Hara, T. (2003): Replica allocation methods in ad hoc networks with data update, *Journal of Mobile Networks and Applications*, 8(4):343-354.

Harbitter, A. and Menasce, D. A. (2001): The performance of public key enabled Kerberos authentication in mobile computing applications, *Proc. of the 8th ACM conference on Computer and Communications Security*,78-85.

Hu, Y-C, Perrig, A. and Johnson, D. B (2002): Ariadne A secure On-Demand Routing Protocol for Ad Hoc Networks, *Proc of the eighth Annual International Conference on Mobile Computing and Networking*, 12-23.

Hu, Y-C, Perrig, A. and Johnson, D.B. (2002): SEAD Secure Efficient Distance Vector Routing for Mobile Wireless Ad Hoc Networks, *Proc of IEEE Workshop on Mobile Computing Systems and Applications*, 3-13.

Hubaux, J. P., Buttyan, L. and Capkun, S. (2001): The Quest for Security in Mobile Ad Hoc Networks. *Proc. of ACM Symposium on Mobile Ad Hoc Networking and Computing*, 146-155.

Kohl, J. and Neuman, S. (1993): The Kerberos Network Authentication Service (V5), RFC 1510.

Pirzada, A. A. and McDonald, C. (2003): A Review of Secure Routing Protocols for Ad hoc Mobile Wireless Networks, *(To be published in) Proc. of 2nd Workshop on the Internet, Telecommunications and Signal Processing (DSPCS'03 & WITSP'03).*

Royer, E. M. and Toh, C. -K. (1999): A Review of Current Routing Protocols for Ad-Hoc Mobile Wireless Networks, *IEEE Personal Communications Magazine*, 6(2):46-55.

Stallings, W. (2003): *Network Security Essentials 2nd Edition*, Prentice Hall.

Tung, B., Neuman, C. and Wray, J. (2001): Public key cryptography for initial authentication in Kerberos, Internet draft (work in progress).

Zhou, L. and Haas, Z. J. (1999): Securing Ad Hoc Networks, *IEEE Network Magazine*, 13(6).