Pakistan Journal of Applied Sciences 3 (5): 346-359, 2003
ISSN 1607-8926

Secure Banking: Authenticating Peer-to-Peer Ad Hoc Interactions

Khalid Mahmood, Imran Hameed, Khalid Rashid and [1]Amir Qayyum
Department of Computer Science International Islamic University Islamabad, Pakistan
[1]CARE, Islamabad, Pakistan

Abstract: Research on mobile ad hoc networks has been focused primarily on routing protocols, whereas security issues remained mostly unexplored. This paper focuses on security mechanisms and notably on the key management mechanisms. Our authentication protocol is inspired from Resurrecting Duckling policy with a novel diversity. We present a reliable solution for securing bank transactions. In our approach devices exchange a limited amount of public information over location-constrained channel, which will later allow them to complete key exchange over wireless link. After the exchange of public keys, our banking transactions will be secure. This approach does not require any public key infrastructure as is required in MANET. Moreover our approach is secure against passive attacks on location-limited channel and all kinds of wireless attacks on wireless link.

Key words: MANET, authentication, pre-authentication, location-limited channel, elliptic cryptography, threshold cryptography, resurrecting duckling

Introduction

Recent technology has increased the use of laptop and portable computers. These are equipped with megabytes of storage, high-resolution color displays, printing devices and wireless communication adapters. With these advances, wireless computing devices should physically be able to communicate with each other, even when no routers or base stations or internet service provider (ISP) can be found.

An ad hoc network is cooperative engagement of a collection of mobile nodes (wireless computing devices) without the required intervention of any centralized access point. Each mobile node operates as specialized router, which periodically advertises its view of the interconnection topology with other mobile nodes within the network. Fig. 1 shows a small ad hoc network. Initially, MH1 has a direct link with MH2. But when MH1 move out of MH2 range, the link is broken, but the network is still connected, because MH2 can reach MH1 through MH4, MH6 and MH7.

Within the last few years, there has been a surge of interest in mobile ad hoc networks (MANET). A MANET is defined as a collection of mobile platforms or nodes where each node is free to move about arbitrarily. The term MANET (Charles and Perkins, 2001) describes distributed, mobile, wireless, multi hop network that operates without the benefit of any existing infrastructure, except for the nodes themselves. A MANET cloud is composed of autonomous,

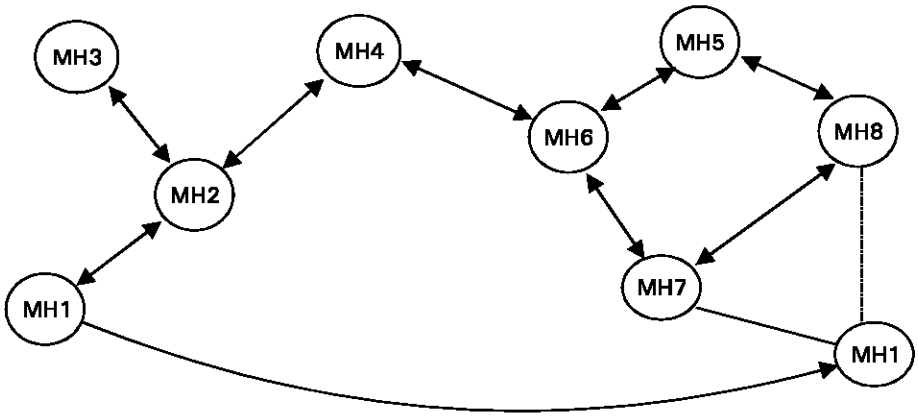

Fig. 1: A small MANET

potentially mobile, wireless nodes that may be connected at the edge of the fixed, wired internet. Police, fire and rescue, disaster relief, robotics, space, distributed sensors and impromptu team communication are a few possible applications of MANET technology.

Confidentiality, availability, integrity and authentication and non-repudiation are core attributes for communication in secure networks. These security requirements are identical for mobile ad hoc networks. Encryption and key exchange mechanism can be used to solve the security problems in ad hoc networks.

Most important characteristics in mobile ad hoc network include dynamic topologies, power constrained operations and bandwidth-constrained variable capacity link.

In wireless ad hoc networks security to be achieved is difficult (Yongguang and Wenke) because:

- Wireless links are susceptible to passive as well as active attacks mainly due to open media.
- Nodes with inadequate physical protection can be compromised.
- Sporadic nature of connectivity.
- Dynamically changing topology.
- Absence of Certification authority.
- Lack of centralized monitoring or management point.
- Cooperative nature of algorithms.
- Sleep deprivation torture.

Threats

The most important threats that mobile ad hoc network have to face are (Hubaux ***et al.***, 2001)

- Attack on basic mechanism of the ad hoc network, such as routing. Prevention of these attacks requires security mechanisms that are often based on cryptographic algorithms. Routing mechanisms are more vulnerable in ad hoc networks than in conventional networks since in ad hoc network each device acts as a relay. This means, that an adversary who hijacks an ad hoc node could paralyze the entire network by disseminating false routing information. A less dramatic but subtler malicious behavior is node selfishness. Moreover, weakness in protocols can be exploited to perform malicious neighbor discovery.

- Attack on security mechanisms and especially on the key management mechanism. Key management is certainly not a problem limited to ad hoc networks; however, because of the peculiarities of ad hoc networks, its solution requires specific attention. Examples of attacks on security mechanism are: Public keys can be maliciously replaced; some keys can be compromised; if there is a distributed trusted server, it can fall under the control of a malicious party.

Security services

To secure an ad hoc network, we consider following services: confidentiality, Authentication, Integrity, Access control, Availability and Non-repudiation (William).

Confidentiality

Confidentiality is protection of transmitted data from passive attacks. It ensures us that our information can't be disclosed to unauthorized entities. The other aspect of confidentiality is the protection of traffic flow from analysis. This requires that an attacker is unable to observe the source and destination, frequency, length or other characteristics of the traffic on a communications facility. The challenge in confidentiality is not only protecting data transported by network but also data stored on device.

Authentication

It enables a node to ensure the identity of the peer node it is communicating with. Without authentication, an adversary could masquerade a node, thus gaining unauthorized access to resource and sensitive information and interfering with the operation of other nodes.

Integrity

It assures that messages are received as sent with no duplication, modification, reordering or replays.

Access control

It is the ability to limit and control the access to host systems and applications via communication links. To achieve this control, each entity trying to gain access must first be identified, or authenticated, so that access rights can be tailored to the individual.

Non-repudiation

It prevents either sender or receiver from denying a transmitted message. Non-repudiation is useful for detection and isolation of compromised nodes.

Availability

It ensures the survivability of network services despite denial of service attacks. A denial of service attack can occur at any layer of ad hoc network. For example on network layer and adversary could disrupt the routing protocol. On higher layer, an opponent could corrupt high

level services too. One such target is the key management service, which is essential for any security framework.

Existing security models in MANET

This section presents various security models for ad hoc networks. Each security model and its scope will be discussed one by one.

Resurrecting duckling policy

Authenticity can be achieved in mobile ad hoc networks (which involve only two devices or less computing devices) by Resurrecting Duckling policy, which was introduced by Frank Stajno and Ross Anderson in (F. Stajano and R. Anderson,1999) and extended in (Stajano, 2000). It is particularly suited to devices without display and embedding a processor too weak for public-key operations. The fundamental authentication problem is a secure transient association between two devices establishing a master-slave relationship. It is secure in the sense that master and slave share a common secret and transient because the master can solve the association only. Also a master can always identify the slave in a set of devices.

The proposed solution is called the Resurrecting Duckling model. The duckling is the slave device while the mother duck is the master controller. The duckling may be in one of the two states, imprinted or imprintable, depending on whether it contains a soul or not; it starts (pre-birth) as imprintable, becomes imprinted at birth when a mother duck gives it a soul and it becomes imprintable again on death, when the soul dissolves. The soul is a shared secret that binds the duckling to its mother: as long as the soul is in the body, the duckling will stay faithful to the mother and obey no one else. Resurrection is allowed, as the name of the policy suggests, but the duckling's metempsychosis works in reverse: instead of one soul inhabiting successive bodies, here we have one body hosting a succession of souls. The soul is originally transferred from mother to duckling over a non-wireless channel (e.g. electrical contact) in order to bootstrap the rest of the protocol. Death, which makes the duckling imprintable by a new mother, may be triggered by the conclusion of the current transaction or by a deliberate order from the mother duck ("commit suicide now!"), but not by one from an outside principal (William). The mother duck should backup the soul with local escrow parties since, if the soul is lost (for example your dog chews on the remote control), the duckling will be unresponsive to any other principal and it will be impossible to reset it to the imprintable state.

Another insight comes from scenarios where we have a pool of identical devices, such as a bowl of disinfectant containing ten thermometers. The doctor does not really care which thermometer she gets when she picks one up, but she does care that the one her palmtop talks to is the same one she is holding and not any other nearby.

A metaphor inspired by biology will help us describe the behavior of a device that properly implements secure transient association. As Konrad Lorenz (1949) beautifully narrates, a duckling emerging from its egg will recognize as its mother the first moving object it sees that makes a sound, regardless of what it looks like: this phenomenon is called imprinting.

If several entities are present at the device's birth, then the first one that sends it a key becomes the owner: to use another biological metaphor, only the first sperm gets to fertilize the egg. As long as the soul stays in the body, the duckling remains alive and bound to the same mother to which it was imprinted. But this bond is broken by death: thereupon, the soul dissolves and the body returns in its pre-birth state, with the resurrecting duckling ready for another imprinting that will start a new life with another soul. Death is the only event that returns a live device to the pre-birth state in which it will accept an imprinting. We call this process reverse metempsychosis. Metempsychosis refers to the transmigration of souls as proposed in a number of religions; our policy is the reverse of this as, rather than a single soul inhabiting a succession of bodies, we have a single body inhabited by a succession of souls.

With some devices, death can be designed to follow an identifiable transaction. In medicine, a thermometer can be designed to die (and lose its memory of the current patient's temperature history) when returned to the bowl of disinfectant at the nursing station. It is also possible that for some devices we can arrange a simple timeout mechanism, so that the duckling dies of old age. With other devices (and particularly those liable to be stolen) we will arrange that the duckling will only die when so instructed by its mother: thus only the currently authorized user may transfer control of the device. In order to enforce this, some level of tamper resistance will be required: assassinating the duckling without damaging its body.

There are also applications in which only part of the duckling's soul should perish. Our thermometer might be calibrated every six months and the calibration information must not be erased along with the patient data and user key when the device is disinfected, but only when it is plugged into a calibration station. We may consider the device to be endowed with two souls—the calibration state and the user state and a rule that the latter may not influence the former. So the resurrecting duckling security policy may be combined with multilevel security concepts.

During the imprinting phase, as we said, a shared secret is established between the duckling and the mother. One might think that this is easy to do: the mother generates a random secret and encrypts it under the public key of the duckling, from which it gets back a signed confirmation.

In many applications there will only be one satisfactory solution and we advocate its use generally as it is effective, cheap and simple: physical contact. When the device is in the pre-birth state, simply touching it with an electrical contact that transfers the bits of a shared secret constitutes the imprinting. No cryptography is involved, since the secret is transmitted in plaintext and there is no ambiguity about which two entities are involved in the binding.

There is however a number of other equally interesting situations that the model so far described do not adequately cover. All the above cases involve a definite master-slave relationship between the mother and the duckling, but we can envisage cases of ad hoc networks between devices that it would be more natural to consider as peers. If the components of our hi-fi and video system talk to each other, for example because the timer wants to start the satellite TV tuner and the DVD writer in order to record something on air, or because the DVD player wants to tell the TV that it should set the aspect ratio to widescreen

for this program, does it make any sense for the DVD player to become the mother duck of the television?

The new work presented extends the Resurrecting Duckling model to cope with such peer-to-peer cases. The duckling will always obey its mother, who tells it whom to talk to through an access control list. The bond between mother and duckling is broken by death after which the duckling accepts another imprinting. Death may be caused by the mother itself, a timeout or any specific event. The whole security chain corresponds to a tree topology formed of hierarchical master-slave relationships. The root of the tree is a human being controlling all devices and every node controls all devices in its sub tree. However, if one relationship is broken the relationship to the whole sub tree is also broken.

This security model can be applied to very large ad-hoc networks, e.g. networks consisting of smart dust devices (Warneke ***et al.***, 2001). A possible scenario is a battlefield of smart dust soldiers (acting as slaves or siblings) and their general (acting as the master). The master allows its slaves to communicate by uploading in each of them a highly flexible policy so that sibling entities become masters and slaves for a very short time, enough to perform one operation. The mother duck gives the ducklings credentials that allow them to authenticate themselves.

Scope

The Resurrecting Duckling scheme is an appropriate model for a well-defined hierarchy of trust relationships. It particularly suits cheap devices that do not need a display or a processor to perform public-key operations. It perfectly works for a set of home devices. However, more flexible ad-hoc networks may not contain explicit trust relationships between each pair of nodes or to a centralized entity like the mother duck. Deploying a comprehensive network consisting of a hierarchy of a global mother duck and multiple subsidiary local mother duck is very similar to a public-key infrastructure, where the mother duck correspond to Certification Authority (CA), with all its advantages and drawbacks. Even the battlefield scenario raises some problems. Here the soldiers are siblings and obey their mother, the general. If one soldier device wants to authenticate to another device it has to present its credentials (William). The second device can then check the Credentials by using its policy. But what happens if all soldiers' do not use the same credentials, i.e. the same secret key to prevent it to be stolen by the enemy. If all devices use the same key the other side might invest considerable effort doing some physical attack (Anderson and Kuhn, 1996) to recover the key because it would compromise all nodes. Since the devices cannot hold a list of all valid credentials it seems that a further authentication method is needed.

Password-based key agreement

The work developed in (Asokan and Ginzboorg, 2000) addresses the scenario of a group of people who wants to set up a secure session in a meeting room without any support infrastructure. People physically present in the room know and trust one another personally. However they do not have any prior means of digitally identifying and authenticating one another, such as shared secret or mutually verifiable public key certificate chains or access to

trusted key distribution centers. An attacker can monitor or modify all traffic on the wireless communication channel and may also attempt to send messages purporting to come from those who are inside the room. There is no secure communication channel to connect the computers. Desirable properties of a protocol that solves this problem should be:

- Secrecy: The basic requirement of secrecy is that only those entities who know an initial password should be able to learn the resulting session key. An observer must not be able to get any information about the session key.
- Perfect forward secrecy requires that an attacker who succeeds in compromising one member of the group and learns about his permanent secret information will still be unable to recover the session keys resulting from previous runs of the protocol.
- Contributory key agreement The resulting session key is established by the contribution from all entities participating in the meeting. This ensures that if only one entity chooses its contribution key randomly all other entities will not be able to make the key space smaller.
- Tolerance to disruption attempts The strongest attacker can disrupt any protocol by jamming the radio channel or modifying the contents of messages among legitimate members. The protocol must not be vulnerable to an attacker who is able to insert messages. It is assumed that the possibility of modifying or deleting messages in such an ad-hoc network is very unlikely.

The work describes and introduces several password-based key-exchange methods that meet these requirements. A weak password is sent to the group members. Each member then contributes to part of the key and signs this data by using the weak password. Finally a secure session key to set up a secure channel is derived without any central trust authority or support infrastructure.

Firstly, a well-known two-party protocol for password authenticated key exchange is described, based on the protocol called Encrypted key exchange (EKE) (Steven M. Bellovin and Michael Merrit,1992) and extends the work from two-party to multiple parties by electing a leader. The main drawback of the multiparty version is that the leader chooses the common session key unilaterally: the key agreement scheme is non-contributory. The protocol is then modified to extend it to a contributory multi-party protocol.

Secondly, Diffie-Hellman exchange (DH), a classic two-party key agreement protocol is extended to support multi-party password authenticated key exchange. An extension of unpublished protocol by Michael ***et al.*** (1996) in which each member shares a different password with an authentication server, is used. This new protocol provides perfect forward secrecy to all players. It is also resilient to disruptions: The leader can disrupt the protocol completely. Any other member attempting to disrupt the protocol by sending out a random quantity will not be able to compute the session key.

Finally, a fault-tolerant Diffie-Hellman exchange on a d-cube is extended to handle failures which were based on the idea proposed by Becker and Wille.

Scope

Password-Based key agreement model perfectly works for small groups. Authentication is done outside the IT system, e.g. the group members authenticate themselves by showing their passport or common knowledge. This model does not suffice anymore for more complicated environments, though, groups of people who do not know each other or number of people who want to have confidential exchanges without bringing in knowledge of the rest of the group be able to eavesdrop on the channel, are two examples. Another problem arises for large groups or groups at different locations. The secure channel to distribute the initial password is not available anymore. It seems that existing support infrastructure is required to set up a secure channel.

Distributed public-key management

Key management for established public-key systems requires a centralized trusted entity called Certificate Authority (CA). The CA issues certificates by binding a public key to a node's identity. One constraint is that the CA should always be available because certificates might be renewed or revoked. Replicating the CA improves availability. However, a central service goes against the distributed structure of ad-hoc networks.

Zhou and Haas (1999) proposes to distribute trust to a set of nodes by letting them share the key management service, in particular the ability to sign certificates. This is done using threshold cryptography (Desmedt and Frankel, 1990). An (n, t + 1) threshold cryptography scheme allows n parties to share the ability to perform a cryptographic operation so that any t + 1 parties can perform this operation jointly whereas it is infeasible for at most t parties to do so. Using this scheme the private key k of the CA is divided into n shares (S1, S2, Sn), each share being assigned to each special node. Using this share a set of t + 1 special node is able to generate a valid certificate. As long as t or less special nodes are compromised and do not participate in generating certificates the service can operate. Even if compromised nodes deliver incorrect data the service is able to sign certificates. Threshold cryptography can be applied to well use signature schemes like the Digital Signature Standard (DSS) (Gennaro ***et al.***, 1996).

Another approach introduced in (Hubaux ***et al.***, 2001) presents a self-organized public-key infrastructure. The system replaces the centralized CA by certificate chains. Users issue certificates if they are confident about the identity, i.e. if they believe that a given public key belongs to a given user. Each user stores a list of certificates in its own repository. To obtain the certificate of another entity the user builds a certificate chain using his repository list and implicitly trusted entity's lists, until a path to an entity that has the desired certificate in its repository is found.

Scope

The Threshold Key Management system is a way to distribute a public-key system. For high-value transactions public-key systems are certainly the only way to provide a satisfactory and legal security framework. Trust should, as much as possible, be based on knowledge. Since two entities that never met before cannot have common knowledge - a shared secret - they both

have to trust a central entity, e.g. a CA to substitute this knowledge. When a user wants to prove his identity to CA, he goes there with his public key and shows his passport. The CA proves his identity and then binds his identity to his public key and signs the certificate. How it can be done when the CA is distributed? The user must prove his identity to all special nodes to prevent that a compromised node passes on faulty information. In this case friendship or just knowing each other is considered as common knowledge. The CA signs certificates without proving the identity, that's why; this model cannot be used for high-value transactions.

The self-organized public-key infrastructure (Hubaux ***et al.***, 2001) shows similar problems. To make the system bullet-proven, the entity's identity had to be checked in real world before users issue certificates. Furthermore it is assumed that the certificate requester trusts each node in the recommendation chain. Finally a significant computing power and time is consumed to obtain a certificate going through the certificate chain. Each node in the chain has to perform public-key operations, first to check the received certificate for authentication (signature verification) and then to sign it before forwarding it (signature generation). This cannot be done in parallel but only one after the other.

Despite its centralized nature a central CA is preferable for applications with high-security demand. To ensure high availability the CA can be replicated. The replicated CA's are as secure as the original CA as long as the replication process is not vulnerable to attacks. The private key of the CA does not get weaker after replication. Much research has been done about efficient public-key systems - for example, a public-key system for mobile systems is presented in (G. Horn and B. Preneel, 1998).

Proposed system model

Imagine a situation in which bank customers are standing in a queue and two or three bank employees are dealing them in making transactions. The customers have to spend a lot of time and also have to wait for their turn in the queue. The bank needs more employees for giving better service to its customers. This scenario, both for the bank and the customers, can be improved by deploying ad hoc networking using Resurrecting Duckling Policy (Stajano and Anderson,1999) and (Stajano, 2000), with some extensions in our idea.

Our idea is elaborated as: a user (bank customer) enters the bank, with his laptop or PDA, to perform some transactions. The bank has mounted the Infrared Bar Code at one side in the bank and also owes a wireless radio link network (e.g. Bluetooth (www.bluetooth.com) or 802.11). The user or customer enters the bank premise, would walk up to the Infrared bar code and briefly establish physical contact between infrared bar code and his laptop or PDA. This process is termed as Pre-authentication, which is done using a link-constrained channel. During Pre-authentication, their public keys will be exchanged. Now the user can sit in the bank premises and his laptop or PDA can then perform standard SSL or TLS (Dierks and Allen, 1999) key exchange with the bank server over the wireless link (e.g. Bluetooth(www.bluetooth.com) or 802.11), since he owns a secret key to perform secure transactions and can establish an authenticated and secret communication channel.

Such an exchange of pre-authenticated data ensures the user that he wants to communicate with the device (infrared bar code) by using a special, link-constrained side channel to exchange a small amount of cryptographic information. That information can be used to authenticate standard key exchange protocols performed over wireless link.

Once the pre-authentication is completed, the devices proceed to establish a secure connection between them over the main wireless link. To this end, they can use any established public-key-based key exchange protocol which requires them to prove possession of a particular private key (e.g., SSL/TLS, SKEME IKE, etc.), which will correspond to the public key committed to in the pre-authentication step.

After secret key exchange, the customer or user will send his/her account number along with the password or secret code (which can be issued at the time of opening an account in the bank and can be changed at any time through bank website) to the bank server. The A/C No. plus code or password will be encrypted with secret key (Session key) exchanged during pre-authentication. After this phase, the user can perform his transactions securely.

We can summarize the basic scheme for pre-authentication as follows.

A) Pre-authentication over location-limited channel.

$$1\text{- } X \rightarrow Y\text{: } H\ (KU_X)$$
$$2\text{- } Y \rightarrow X\text{: } H\ (KU_Y)$$

B) Authentication over wireless channel with SSL/TLS.

$$X \rightarrow Y\text{: TLS\_CLIENT\_HELLO..... and so on.}$$

The various symbols denote:

X: Customer's Laptop

Y: Bar code device

KU_X, KU_Y: Public key belonging to X and Y respectively.

$H(KU_X)$, $H(KU_Y)$: one-way hash of encoding of corresponding keys.

Advantages of our approach

This idea of pre-authentication has been generalized to secure arbitrary peer-to-peer ad hoc interactions using a wide variety of key exchange protocols. We have introduced the use of public-key cryptography and can remove the secrecy requirements on link-constrained channels used to authenticate key exchange protocols. More importantly, it allows us to expand the range of key-exchange protocols which can be authenticated in this manner to include almost any standard public-key-based protocol. As a result, our approach can also be used with an enormous range of devices, protocols and applications (as one of them is discussed).

At the same time, our approach is significantly more secure than previous approaches, as we force an adversary to mount an active attack on the location-limited channel itself in order

to successfully subvert an ad-hoc exchange. Previous approaches (e.g., use of unauthenticated Diffie-Hellman key exchange) are either vulnerable to either active attacks in the main wireless channel, or, in the case of Anderson and Stajano, to passive (eavesdropping) attacks in the location-limited side channel.

Advantages of Location-limited Channel

Communication technologies that have inherent physical limitations in their transmissions are good candidates. For example, audio (both the audible and ultrasonic range), which has limited transmission range and broadcast characteristics, can be used by a group of PDA's in a room to demonstratively identify each other. For situations that require a single communication endpoint, channels with directionality such as infrared are natural candidates.

The channel be impervious (or resistant) to eavesdropping. For example Anderson and Stajano use secret data, such as a symmetric key, exchanged across the location-limited channel to allow participants to authenticate each other. As a result, that authentication protocol is vulnerable to a passive attacker capable of eavesdropping on the location-limited channel, thereby obtaining the secrets necessary to impersonate one of the legitimate participants. A location-limited channel used to exchange such secret pre-authentication data must therefore be very resistant to eavesdropping.

We therefore propose that any physically limited channel suitable for demonstrative identification, on which it is difficult to transmit without being detected by at least one legitimate participant (human or device), is a candidate for use as a pre-authentication channel. Such candidates include: contact, infrared, near-field signaling across the body and sound (both audible and ultrasound). The amount of data exchanged across the pre-authentication channel is only a small fraction of that sent across the main wireless link and so we can use channel media capable only of low data rates.

Advantages of pre-authentication

Because legitimate participants would only communicate with entities from which they had received pre-authentication data, we would now require an attacker to perform an active attack - to be able to transmit -not only in the main wireless medium, but also in the location-limited channel. Because of the physical limitations of transmission on location-limited channels, it is significantly harder for an attacker to passively eavesdrop on them, not to mention to actively transmit.

For such an active attack to succeed, the attacker must not only transmit on the location-limited channel, but must do so without being detected by any legitimate participant.

The difficulty of monitoring a pre-authentication for such unwanted participation depends on the type of channel used and the number of legitimate parties involved. The more directed the channel and the smaller the number of parties, the easier it is to monitor. Note that, because of the physical limitations of the channels used and this monitoring requirement, it is only possible to use our techniques to pre-authenticate devices that are physically co-located at the time of first introduction.

Usage of Elliptic curve cryptography

How do we remove the requirement that pre-authentication data be kept secret? We can do this very simply through the use of public key cryptography. If the participants use the location-limited channel to exchange their public keys as pre-authentication data, it doesn't matter whether an attacker manages to eavesdrop on the exchange. The participants will authenticate each other over the wireless link by proving possession of their corresponding private keys; as the attacker does not know those private keys, he will not be able to impersonate any of the legitimate participants.

If we accept the existence of cryptographically secure hash functions (e.g., SHA-1), we can further limit the size of the pre-authentication data exchanged. The participants do not actually need to exchange their complete public keys as pre-authentication data, they merely need to commit to those keys (e.g., by exchanging their digests).

Instead of using traditional public key algorithms like RSA, we have included Elliptic cryptography (A. Murat Fiskiran and Ruby B) in our approach for generating public/private key pair because it is more suitable for constrained environments (low memory wireless devices).

In this paper, we have proposed a secure, efficient and user-friendly solution to our bank transaction problem (and to the problem of authentication in local ad hoc wireless networks in general, for which our bank scenario merely serves as example.)

Implementation

We have divided our work in to three major areas. The first step is to implement exchange of pre-authentication data. We have exchanged hash of public keys. Hash code is generated by using SHA. Public/Private key pair is generated by using RSA but later we have implemented ECC (A. Murat Fiskiran and Ruby B) for this purpose, as it is more suitable for small devices.

In second step, we have implemented SSL for complete exchange of keys and these keys are authenticated by using pre-authentication data.

For these two steps we have used sockets. The server sockets listens for a connection on both location-limited channel and the primary link, but only admits primary-links connection from clients, who have performed pre-authentication on location-limited channel. Currently we have used serial cable for location-limited channel.

The third step is to transfer data from client to server. For this purpose we have utilized Microsoft Access for database and this whole framework is implemented in VC++6.

Future work

The scheme presented has distinct advantages over traditional authentication and security models. We are focusing our future work on different scenarios relating to security and authentication problems, keeping in view the bandwidth-constrained media and computation limited devices.

Now we are going to implement our work by utilizing Bluetooth devices. To show actual transactions we are using PHP and MySql. Our work will mainly focus on upper layers (e.g. RFCOMM layer, SDP, L2CAP etc.) of Bluetooth by using open source code of Bluetooth device driver in Linux.

Conclusion

There is no any single authentication model which ensures authentication in every environment. While for a group meeting in small conference room the Password-based key exchange will work perfectly, for a network defined by hierarchy of trust relationships, the Resurrecting-Duckling police are an ideal alternative. For guaranteed secure transaction certainly the choice is to use public key system involving the hassle of group certificates.

In this paper we have presented new approaches for peer-to-peer authentication in mobile ad hoc networks. We have constructed our schema on previous work by Anderson, Stajano and others and presented to perform pre-authentication over location-limited channels. Our schema is public key infrastructure less and resolves naming problem that plagues traditional authentication systems.

Unique location-limited channels

Instead of limiting the concept of imprinting a duckling device with its mother's secret key, we propose to use location-limited channel to bootstrap a wide range of key-exchange protocols. This approach can be experimented to audio, infrared and contact-based channels, but other media are certainly imaginable.

Composed pre-authentication protocols

We provide a composed recipe for enhancing existing key exchange protocols with a pre-authentication step. And we explained how passive as well as active attacks can be detected by human user or by the system.

No dependency on public key infrastructure

Key exchange and key agreement protocols depend on authentication step to verify the user. We have suggested a way to solve this problem. A reliance on pre-existing third party naming and trust infrastructures is unnecessary if one can briefly bring communicating parties within close physical proximity. In such a case, our pre-authentication protocols can be used in place of a PKI.

References

Anderson, R. and M. Kuhn, 1996. Tamper resistance-a cautionary note. 2nd USENIX Workshop on Electronic Commerce.

Asokan, N. and P. Ginzboorg, 2000. Key Agreement in Ad-hoc Networks. Computer Communications.

Charles, E. Perkins and Ad hoc Networking, 2001, Addison-Wesley, London, ISBN: 0-201-30976-9.

Desmedt Y. and Y. Frankel, 1990. Threshold cryptosystems. Advances in Cryptology-Crypto '89, LNCS 435, Springer-Verlag.

Dierks, T. and C. Allen, 1999. The TLS Protocol Version 1.0. IETF Network Working Group, the Internet Society, RFC 2246.

Gennaro, R., S. Jarecki, H. Krawczyk and T. Rabin, 1996. Robust threshold DSS signatures. Advances in Cryptology-Eurocrypt '96, LNCS 1070, Springer-Verlag.

Horn, G. and B. Preneel, 1998. Authentication and Payment in Future Mobile Systems. Computer Security - ESORICS '98, LNCS 1485, Springer-Verlag.

Hubaux, J.P., L. Buttyan and S.I. Capkun, 2001. The quest for security in mobile ad hoc networks. In Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking and Computing MobiHoc).

Klaus Becker and Uta Wille. Communication complexity of group key distribution. In 5th ACM Conference on Computer and Communications Security, pp: 1-6.

Konrad Lorenz. Er redete mit dem Vieh, den V¨ogeln und den Fischen (King Solomon's ring). Borotha-Schoeler, Wien, 1949.

Michael Steiner, Gene Tsudik and Michael Waidner, 1996. Private communication. Unpublished work described in slides of presentation made at the ACM CCS conference.

Murat Fiskiran, A. and Ruby B. Elliptic Curve Cryptography. Lee Princeton Architecture Laboratory for Multimedia and Security Princeton University

Stajano, F. and R. Anderson, 1999. The Resurrecting Duckling: Security Issues for Ad hoc Wireless Networks. The 7th International Workshop on Security Protocols, LNCS 1796, Springer-Verlag.

Stajano, F., 2000. The Resurrecting Duckling-what next? The 8th International Workshop on Security Protocols, LNCS 2133, Springler-Verlag.

Steven M. Bellovin and Michael Merrit, 1992. Encrypted key exchange: Password-based protocols secure against dictionary attacks. In proceedings of the IEEE Symposium on Research in Security and Privacy.

The official Bluetooth SIG website. www.bluetooth.com.

Warneke, B., M. Last, B. Leibowitz and K.S.J. Pister, 2001. Smart Dust: Communicating with a Cubic-Millimeter Computer. Computer Magazine, IEEE.

William Stallings. Network and Internet work Security Principles and practice.

Yongguang Zhang, Wenke Lee, Intrusion Detection in ad hoc Networks.

Zhou, L. and Z.J. Haas, 1999. Securing Ad Hoc Networks. IEEE Network Magazine.