Security Issues in Mobile Computing

Srikanth Pullela
Department of Computer Science
University of Texas at Arlington
E-mail: pvssrikath@hotmail.com

Abstract

In the present mobile communication environment, lot of research is going on, to improve the performance of issues like handoffs, routing etc. Security is another key issue that needs to be considered, which comes into picture once the communication channel is set-up. Many security protocols are being proposed for different applications like Wireless Application Protocol, 802.11 etc. most of them are based on the public and private key cryptography. This paper provides an insight on these cryptographic protocols and also looks into the current research project going on at Sun Microsystems Lab on wireless security.

1.Introduction

With the rapid growth in the wireless mobile communication technology, small devices like PDAs, laptops are able to communicate with the fixed wired network while in motion. Because of its flexibility and provision of providing ubiquitous infrastructure, the need to provide security increases to a great degree. As wireless communication takes place mainly through the radio signals rather than wires, it is easier to intercept or eavesdrop on the communication channels. Therefore, it is important to provide security from all these threats. There are different kinds of issues within security like confidentiality, integrity, availability, legitimacy, and accountability that needs to be individually taken care off.

One of the key issues of these being, confidentiality and authentication, where the user must be protected from unauthorized eavesdropping. The goal of authentication protocol is to check the identity of other users or network centers before providing access to the confidential information on the users side. When designing any security protocol, there are certain conditions that need to be considered. Firstly, the low computational power of the mobile users and secondly, the low bandwidth available. Therefore, it is important to design the security protocols so as to minimize, the number of message exchanges and the message size. The proposed protocols discussed in this paper try to solve these problems using the limited resources available at the client side in a mobile environment.

In this paper, we concentrate our discussion on few authentication protocols that were proposed to provide security between the users and the network. These protocols are based on the use of certificates, which are built on the concept of security keys (cryptography). Another protocol that is discussed in this paper is the “KiloByte” Secure Socket Layer (KSSL) protocol, which is an extension of Secure Socket Layer (SSL) protocol used for wired networks.

The structure of the paper is as follows: In Section 2,we will discuss the different issues of security. In section 3, the authentication protocols and the KSSL protocol are discussed. In Section 4, the performance of the security protocols is provided, followed by Discussion in section 5 and Conclusion in section 6.

2.Security Issues

Many authors [7] have presented classifications of security issues in communication networks. There are five fundamental goals of security in information system.

- *Confidentiality*, preventing unauthorized users from gaining access to critical information of any particular user.
- *Integrity*, ensures unauthorized modification, destruction or creation of information cannot take place.
- *Availability*, ensuring authorized users getting the access they require.
- *Legitimate*, ensuring that only authorized users have access to services.
- *Accountability*, ensuring that the users are held responsible for there security-related activities by arranging the user and his/her activities are linked if and when necessary.

The way these goals are achieved depends on the security policy adopted by the service providers.

3.Authentication Protocols

In a wireless mobile communication environment, the messages transmitted over wireless medium are more susceptible to eavesdropping than in wired network. Also, it is possible for any user to access the mobile communication system using a false identity. In order to provide security from the above-mentioned situations, we use encryption, which provides confidentiality of the messages sent over wireless channel and to authenticate. There are two types of encryption techniques in cryptosystem, namely symmetric-key cryptosystem and asymmetric-key cryptosystem. The main idea in using these techniques is to conceal the content of the messages before transmitting them in the clear (radio signals). In this system, a common key is shared between the entities before any communication session begins and later these session keys are used to encrypt the data.

3.1 Symmetric-Key and Asymmetric Cryptosystems

In a symmetric-key cryptosystem, the encryption and decryption keys are the same. Since the encryption and decryption transformations are easily derivable from each other, a common secret key is shared between the communicating entities in advance via a secure channel. Therefore, the security of symmetric-key cryptosystems depends on keeping the key secret. Some of the most important symmetric-key cryptosystems that are used presently are the American Data Encryption Standard (DES) and the Japanese Fast Data Enciphering Algorithm (FEAL).

In an asymmetric-key (public-key) cryptosystems, the encryption and decryption keys differ. Each user has a private key and a public key. Let us consider a scenario, where Alice wants to send a message to Bob. Alice encrypts the message M using Bob's public key P_{bob}, which is exchanged before the session started. At Bob's side, this encrypted message is decrypted using Bob's private key S_{bob}, which is known only to Bob.
Another function in public-key cryptosystem is the use of digital signatures. In this process, if Bob wants to send a message to Alice, he first signs the message M with his private key S_{bob} to obtain a digital signature $S= [h (M) S_{bob}]$ of M, where h ( ) is one-way hash function. Here, hash functions like MD5 and SHA are used which accepts a variable size message and outputs a fixed sized representation h (M) of M. Alice decrypts this encrypted message by using Bob's public key. One of the widely used public-key cryptosystem is the RSA public-key cryptosystem proposed by Rivest, Shamir, and Adleman (RSA). The security of the RSA key is based on the difficulty of factoring large integers. Another public-key cryptosystem that is widely used is the Modular Square Root (MSR) public-key cryptosystem. This is a variant of RSA, where the public key is the modulus N, which is a product of two large primes. MSR requires only one modular multiplication for computing the encryption keys, and because of its low computational cost, is preferred over RSA.

3.2 Protocols based on Symmetric-Key Encryption

3.2.1 Encryption using Symmetric-key function

Because of its negligible computational cost, a symmetric-key encryption is preferred. In this protocol, the home network broadcasts a random number r.

Mobile user $\leftarrow$ Home Network: r
Mobile user $\rightarrow$ Home Network: ID_{ms}, f (k, r)

Then, the mobile user sends its identity ID_{ms} along with function f (k, r), where f () is a symmetric-key function such as DES or FEAL, k is the secret key of the mobile user that it shares with the home network. When the home network receives the secret key from mobile user, it fetches the key in the database and completes the authentication. Once the session keys are exchanged, the messages are encrypted using these keys before transmission.

3.2.2 Encryption using Diffie-Hellman Key Exchange

Diffie-Hellman key exchange is another protocol that is used in Cellular Digital Packet Data (CDPD). This method takes advantage of the ease with which exponentials can be computed in a Galois field GF (q), where q is a prime of elements. As mentioned in paper [1], if y= α^X mod q, for 1 < X < q-1, where α is a fixed primitive element of GF (q), then X= $\log_\alpha$ y mod q is referred to as the discrete logarithm of y to the base α over GF (q). Consider a scenario, where Alice and Bob want to communicate. Here, Alice selects a random number X_a between 1 and q-1, which it keeps as a secret and sends $Y_a=\alpha^{X_a}$ mod q to Bob. Similarly, Bob chooses a random number X_b and sends $Y_b=\alpha^{X_b}$ mod q to Alice. Once the two entities receive the messages, they compute Ks = $\alpha^{X_a X_b}$ mod q and use it as their key. As no one except, Alice and Bob know their keys, any one trying to compute Ks has to do using Y_a and Y_b alone. The security of this system is based on the difficulty of taking the discrete logarithm.

3.3 Protocols based on Public-Key Certificates

In this protocol, a universally trusted certificate authority (CA) is used. This CA can be a single large service provider. Whenever the mobile user registers with a home network, it is provided with a certificate that contains the mobile user's identity, the expiration date of the certificate, the certificate authority's signature and lastly the certificate authority's private key, S_{ca}. Each home network has its own certificate. The certificates of the mobile user ($Cert_{ms}$) and the home network ($Cert_{hn}$) will be as follows:

$Cert_{hn}$ = {ID_{hn}, p_{hn}, $date_{hn}$, [h (ID_{hn}, p_{hn}, $date_{hn}$)] S_{ca},
$Cert_{ms}$ = {ID_{ms}, $date_{ms}$, [h (ID_{ms}, $date_{ms}$)] S_{ca},
where h () is the one-way hash function
date is the expiration date of the certificate
p_{hn} is the private key of the Home Network
S_{ca} is the secret key of the certificate

In this scenario, the home network broadcasts its certificate $Cert_{hn}$ and the mobile user authenticates the home network by verifying the signature with the public key p_{ca} of the certificate.

Mobile user ← Home network: $Cert_{hn}$
Mobile user → Home network: [Ks] p_{hn}, f (Ks, $Cert_{ms}$)

Later, the home network chooses a session key Ks randomly and encrypts it with the public key of the home network. Also the certificate of the mobile user is encrypted with Ks and together both are sent to the home network. When the home network receives the messages, it decrypts the message with its secret key S_{hn}.

Since the security is based on the certificates, any personnel who get to know the certificates has a chance to impersonate the mobile user at home network side. To avoid this kind of security breach, Beller, Chang, and Yacobi suggested using an additional mutual authentication step where another session-key derived from Diffie-Hallman key exchange is used.

In Diffie-Hallman key exchange method, the certificates of the home network and the mobile user contain some additional information as shown below.

$$\mathrm{Cert}_{hn} = \{\mathrm{ID}_{hn}, p_{hn}, Y_{hn}, \mathrm{date}_{hn}, [h\ (\mathrm{ID}_{hn}, p_{hn}, Y_{hn,} \mathrm{date}_{hn})]\ S_{ca},$$
$$\mathrm{Cert}_{ms} = \{\mathrm{ID}_{ms}, Y_{ms,} \mathrm{date}_{ms}, [h\ (\mathrm{ID}_{ms}, Y_{ms,} \mathrm{date}_{ms})]\ S_{ca},$$

$Y_{ms} = \alpha^{X}{}_{ms}$ mod N and $Y_{hn} = \alpha^{X}{}_{hn}$ mod N are the public values for the Diffie-Hellman key exchange of mobile user and home network respectively. X_{ms} and X_{hn} are the secret key values.

In this method, the mobile user computes Ks`= $(Y_{hn})^{X}{}_{ms}$ mod N and chooses a random key Ks to encrypt the certificate $\mathrm{Cert}_{ms}$. After receiving the encrypted message the home network computes Ks' = $(Y_{ms})^{X}{}_{hn}$ mod N. Now, both the entities use their session keys Ks` to encrypt the message before sending them on the communication channel. As the session keys are computed using their individual secret keys, any impersonation can be identified. But there are other problems involved in this method. The session keys generated are identical for all sessions, which is not a good sign from a security point of view. To improve on this method, where it is possible to generate variable session keys, a new protocol was proposed.

In this improved method, the secrecy of the certificates is not a priority. This protocol is similar to the one above, the only difference is in the certificates where the value of Y_{ms} is equal to $\alpha^{-X}{}_{ms}$ mod N and Y_{hn} is equal to $\alpha^{-X}{}_{hn}$ mod N.

Mobile user ← Home network: $\alpha^{Rhn+Xhn}$, Cert_{hn}
Mobile user computes Ks`= $(Y_{hn} * \alpha^{Rhn+X\ hn})^{Rms}$ mod N

Mobile user → Home network: $R_{ms}+X_{ms}$, [Ks] p_{hn}, f (Ks, Cert_{ms})
Home network computes Ks` = $(Y_{ms} * \alpha^{Rms+Xms})^{Rhn}$ mod N
Mobile user ← Home network: f (Ks, [ID_{hn}, ID_{ms}])

In this method, a random numbers R_{hn} and R_{ms} are used. The home network calculates $\alpha^{Rhn+Xhn}$ and along with its certificate Cert_{hn} broadcasts the message to mobile users. After receiving the message, the mobile user calculates Ks` using $Y_{hn}$ present in the certificate. Later, the mobile user generates a random session key Ks and encrypts it with public key $p_{hn}$ and sends it to the home network together with f (Ks, $\mathrm{Cert}_{ms}$) and $R_{ms}+X_{ms}$. Then the home network generates its session key Ks` using the public key P_{ms}. Once both the entities establish their session keys they start exchanging the messages using these keys. The advantage of using this method is that each time a session is set-up, a different session key is generated because of the use of random number Rms and Rhn.

There are also situations where communication between two mobile users needs to be considered. To handle this situation an end-to-end security protocol was proposed.
In this scenario, communication between the mobile users should be protected from both outsiders and insiders of the mobile networks, which also includes the home network. In order to support this kind of security, two levels of mutual authentication and session key exchange are used, one between the mobile user and the network and the other between the mobile users.

Consider two mobile users MS (A) and MS (B) that are registered to home networks HN (A) and HN (B) respectively. Also these mobile users are considered to be outside the home networks, in visiting network VN (A) and VN (B) respectively. Initially, both, the mobile users and the visiting network authenticate each other thereby sharing a common session key using the Diffie-Hellman key exchange mechanism. If the visiting network finds the mobile users certificate to be invalid, it checks with the home network to get the valid certificate for that particular mobile user. The main role of the visiting network here is to deliver the messages securely to the other visiting network involved in the communication process. After a call set-up is made, the network is no longer involved in any cryptographic computations. It just passes the encrypted messages to the required destination.

3.4 KSSL Security Protocol

New technologies like Wireless Application Protocol (WAP) and PalmOS, which are used on small mobile devices like mobile phones and Palmtops, do provide some kind of security in a wireless environment. But, the authors of this paper [2] listed out some problems in this WAP protocol, the primary being the use of proxy-based architecture to provide security.

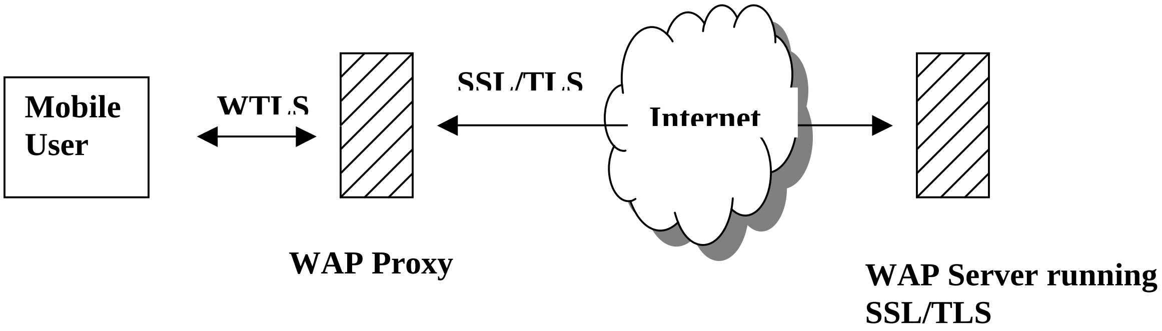


Figure 1: Proxy- based architecture

All the data that the mobile user sends to a particular destination goes through this proxy-based server provided by the service provider. As shown in figure1, the WAP server decrypts the encrypted data using Wireless Transport Layer Security protocol (WLTS) and re-encrypts it using SSL before forwarding to the destination. Some concerns that were raised are, on issues like scalability, where a performance bottleneck comes-up with large number of users using a single service provider besides being a single point of failure, the need to maintain large data buffers to compensate the flow between a low bandwidth wireless channel and a high bandwidth wired channel, and security, where-in the proxy gets to see the process of encryption and decryption, which raises questions on security of sensitive data.

In contrast to the proxy-based architecture, the authors proposed a new protocol named “Kilobyte” Secure Socket Layer (KSSL), which is currently under research at the Sun Microsystems Lab. This protocol is an extension of Secure Socket Layer (SSL) Protocol, which is widely used in a wired network to provide security. Before discussing about the KSSL protocol, we will discuss about the SSL protocol.

3.4.1 Secure Socket Layer (SSL)

As mentioned in paper [2], SSL provides encryption, source authentication, and integrity protection of application data over insecure public networks. This protocol uses the service of TCP, which provides a bi-directional byte stream service.

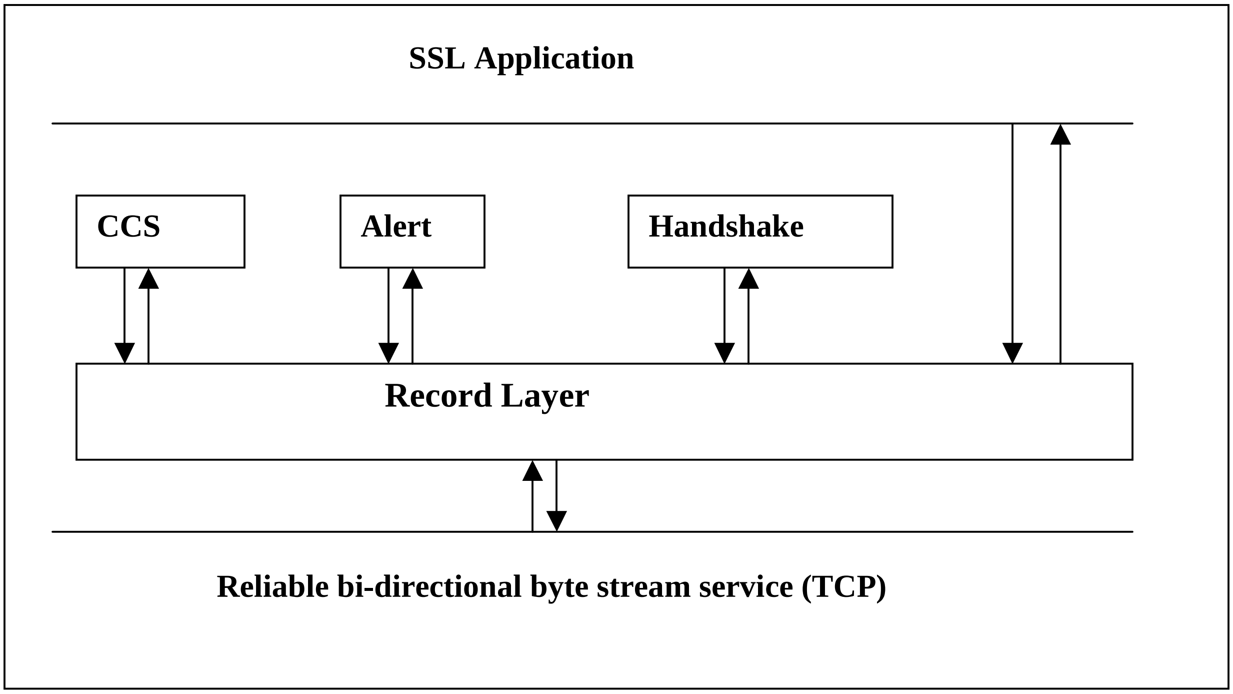


Figure 2: SSL Architecture

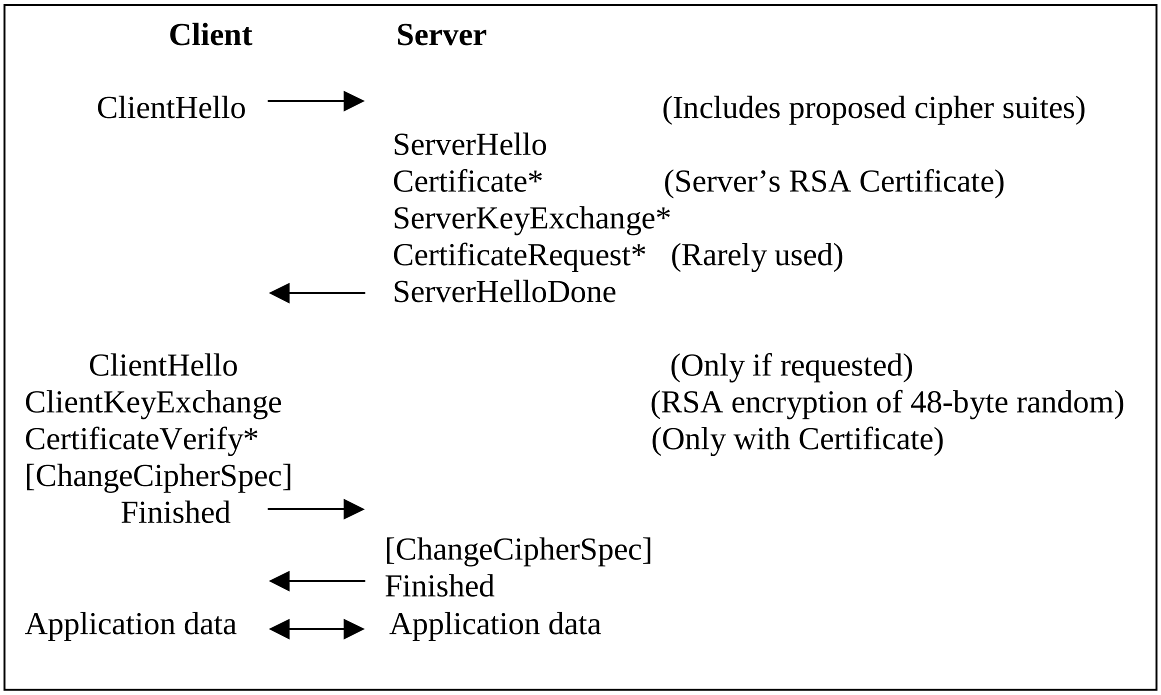
As shown in figure 2, SSL is a layered protocol. The Record layer, placed above the TCP layer, provides encryption and authentication services using symmetric-key algorithm. These keys are established by handshake protocol, which uses public-key algorithms to generate master secret between the SSL client and server. As mentioned in paper [2],this master secret is further used to derive cipher keys, initialization vectors, and message authentication code (MAC) keys for use by Record Layer. The other two protocols that are in the same layer as Handshake are Change Cipher Spec (CCS) and Alert protocols. CCS is used to signal successful completion of the handshake, and start of bulk encryption and authentication and Alert is used to notify if any failures occur.

Because of its flexibility, SSL can support a variety of algorithms, for key agreement (RSA, Diffie-Hallman (DH), etc.), encryption (RC4, 3DES etc.), and hashing (Message Digest (MD5), Secure Hashing Algorithms (SHA), etc.). A standard is been specified that explicitly lists the combinations of these algorithms, together they are called cipher-suites. In our discussion, we use RSA key exchange form. Though SSL protocol supports, client and server side authentication, only server-side authentication is done, as maintaining certificates on the client-side requires maintenance. And the authentication process on the client side is done using passwords sent over an SSL-protected channel.

Full Handshake

The process begins with the client sending a ClientHello message containing a random number, a session ID and a set of supported cipher-suites to the server. The server accepts the message and checks whether it can support the proposed cipher-suite. If no, it aborts the handshake and sends a failure message back to the client. Otherwise, it generates a random number, and along with a session ID and the selected cipher-suite sends them in a ServerHello message to the client.

Figure3: Full SSL Handshake



The ServerHello message is followed by a Certificate message containing the server's RSA public key in an X.509 certificate. At the client side, if this certificate is not able to generate a ClientKeyExchange message then the server sends another RSA public key in a ServerKeyExchange message. The client then verifies the server's public key and generates a 48-byte random number, called pre-master secret and encrypts it with the server's public key. This result is then sent in ClientKeyExchange message. The client then computes a master secret using the pre-master secret, and the client and server random numbers exchanged previously. This master secret is later used for encrypting and authenticating the messages containing data. Once both, client and server establish their master secret keys a ChangeCipherSpec message is sent to signal the end of in-the-clear communication. After this, a Finished message is sent, which is the first message to be completely secured using negotiated cipher-suite. Once the server receives the ClientKeyExchange message, it decrypts the message using the client's public key and generates a master key and stores it in the Record Layer in the same manner as at the client side. Later the server too sends a ChangeCipherSpec message followed by a Finished message to complete the Handshake process. From here on, all the data that is exchanged is securely encrypted as negotiated during the handshake process. Each direction of the traffic's flow uses distinct encryption and MAC keys.

Abbreviated SSL Handshake

Another feature that SSL Handshake supports is the session reuse, where the client and server are allowed to reuse the master key derived in a previous session.

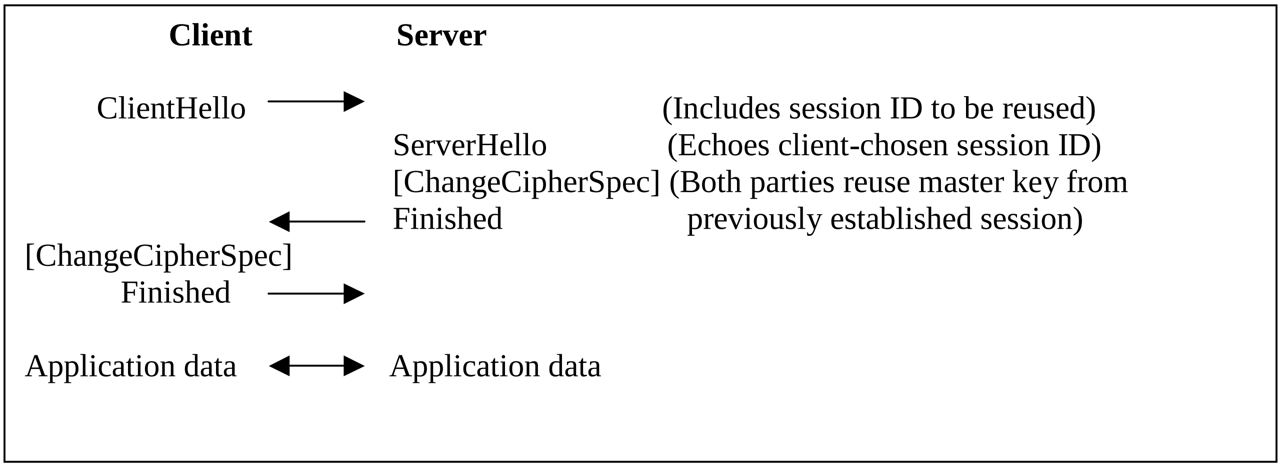

Client
Server
ClientHello
(Includes session ID to be reused)
ServerHello
(Echoes client-chosen session ID)
[ChangeCipherSpec]
(Both parties reuse master key from previously established session)
Finished
[ChangeCipherSpec]
Finished
Application data
Application data

Figure5: Abbreviated SSL Handshake

In the abbreviated handshake protocol, the communication starts with client sending a ClientHello message containing a random number, a non-zero ID of a previously negotiated session, and the proposed cipher-suite. After receiving, the server checks whether it has the session information with it and is ready to use the corresponding master key. If yes, it echoes back the session ID in the ServerHello message. Otherwise, it sends a new session ID signaling the client that a Full handshake process needs to be initiated. As the abbreviated handshake protocol doesn't involve certificates or public key cryptographic operations, fewer messages are exchanged and as a result the process is faster compared to a full handshake process.

3.4.2 KSSL and KSecurity

“KiloByte” SSL (KSSL) is a protocol implemented on the SSL client side for the Mobile Information Device Profile (MIDP) of Java2 Micro Edition. The overall architecture and relationship to the base J2ME is shown in Figure6. This model works in similar terms as a SSL protocol, the only extension being that, all the critical information such as certificates, secret key are stored in a smart card. The different classes present in the model use this information to provide security by encrypting and decrypting the content present in the messages.

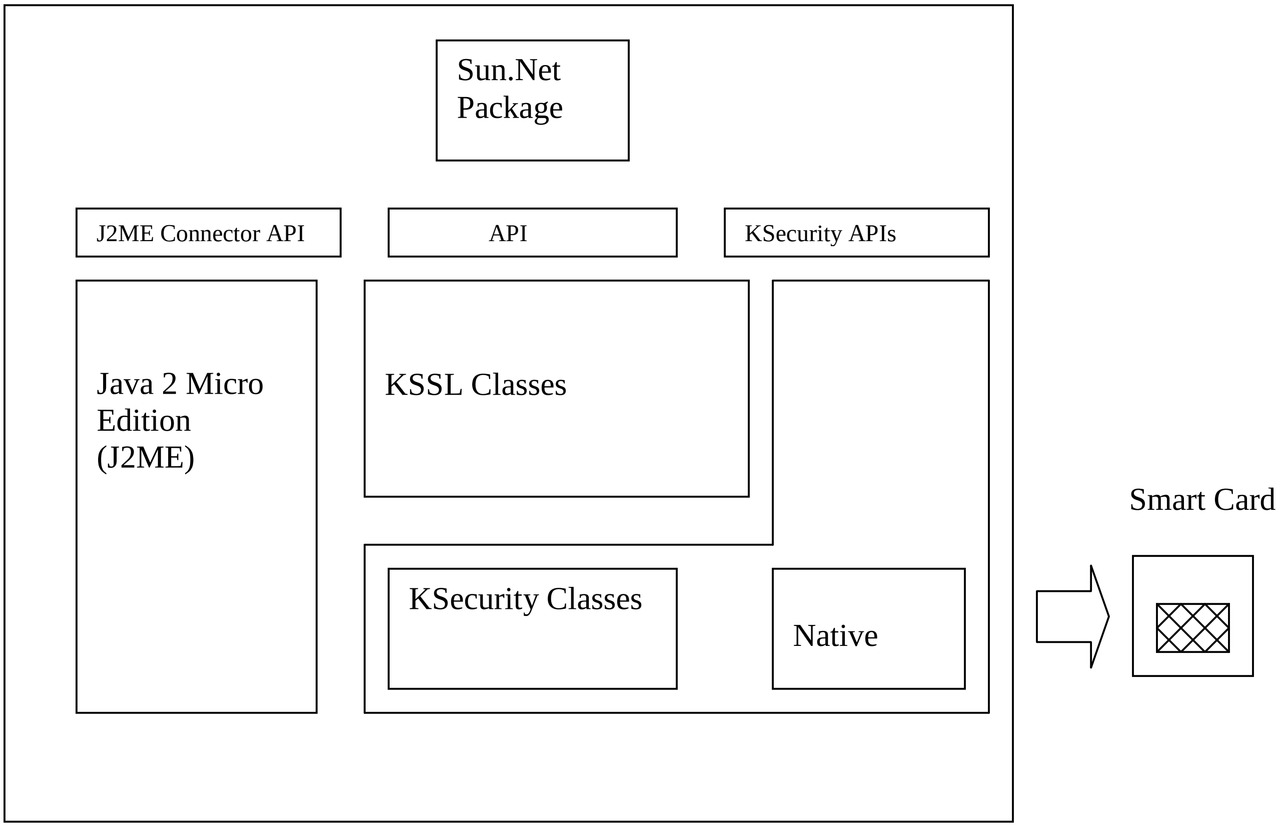


Figure6: KSSL Architecture

The KSecurity classes provide all the basic cryptographic functions such as random number generation, encryption, and hashing. It also contains a Java Card API, which on some occasions is used as a hardware crypto accelerator with minimum changes to the KSSL code. In this model the SSL protocol (KSSL) is written purely in Java and its functions can be accessed using J2ME Connector API. The Connector API in addition to KSSL is used to support HTTPs and URLs. Another API is also used, which is yet to be standardized, for offering greater control over the SSL protocol like seeking user input upon encountering problematic certificates.

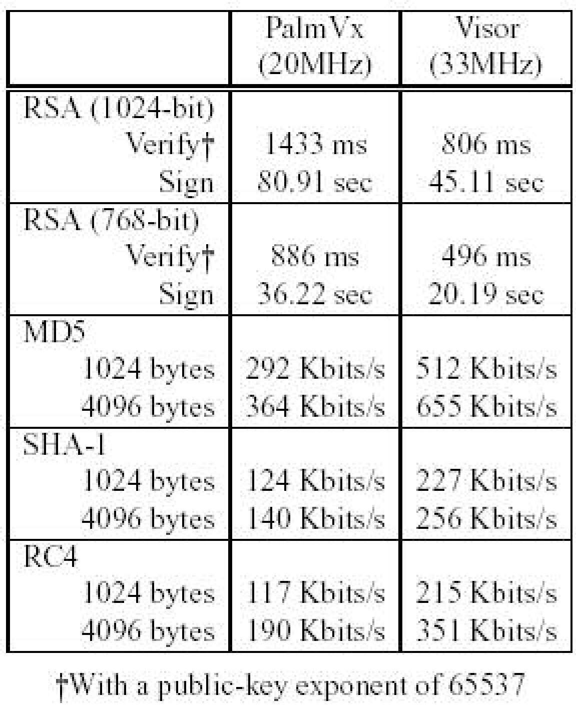
Some of the features that KSSL and KSecurity offer are discussed below.
They support symmetric keys of different lengths and RSA public/private keys with modulus lengths up to and including 1024 bits. For ciphers, they use RSA for key exchange and RC4 for bulk encryption. For Signatures, they support RSA with both MD5 and SHA. Only X.509 certificates containing RSA keys and signed using RSA with MD5 or SHA are supported. The client offers only two cipher-suites, RSA_RC4_128_MD5 and RSA_RC4_40_MD5, as they are fast and universally accepted by the SSL servers. Also client SSL supports session reuse and works on J2ME running on PalmOS, Solaris, and Windows.

4. Performance

The certificate-based security protocol is considered to be more secure than symmetric key protocol in terms of key management. Because of its computational complexity, public key cryptosystem is considered to be a burden on a mobile user with limited resources. Instead, a MSR with RSA system and low component can be used for mobile users. Once a smart card containing the secret value, the certificate and the public key of the CA is issued to a mobile user, no secret information ever leaves the smart card. As specified in the paper [1], the recently announced smart card chip contains an 8-bit CPU as a standard smart card controller and an additional arithmetic coprocessor optimized for modular exponentiation of long operands.

The performance of the KSSL protocol was tested using PalmOS. The results are as follows: the bulk encryption and authentication algorithms are adequately fast on Palm’s CPU. On a 20MHz chip (found in Palm Vx, Palm IIIc, etc.) RC4, MD5, and SHA all run at over 100Kbits/s. When measuring SSL Handshake Latency, a typical key with size of 768 or 1024 bits using RSA takes 0.5-1.5 seconds on a 20MHz Palm CPU.

The table shown below gives the performance of KSSL cryptographic primitives on PDAs. This information is taken from paper [2].



	PalmVx (20MHz)	Visor (33MHz)
RSA (1024-bit)		
Verify†	1433 ms	806 ms
Sign	80.91 sec	45.11 sec
RSA (768-bit)		
Verify†	886 ms	496 ms
Sign	36.22 sec	20.19 sec
MD5		
1024 bytes	292 Kbits/s	512 Kbits/s
4096 bytes	364 Kbits/s	655 Kbits/s
SHA-1		
1024 bytes	124 Kbits/s	227 Kbits/s
4096 bytes	140 Kbits/s	256 Kbits/s
RC4		
1024 bytes	117 Kbits/s	215 Kbits/s
4096 bytes	190 Kbits/s	351 Kbits/s

†With a public-key exponent of 65537

5. Discussion

Presently many wireless technologies are being used with each having their own approach to provide security. In this section we will discuss some of the current approaches and industry standards that are being followed.

The IEEE 802.11 wireless LAN uses a wired equivalent privacy protocol (WEP) mechanism to provide security. Here, the wireless network administrator provides a WEP-algorithm-based key for each authorized user. Any user without an assigned key is denied access.

The WAP application provides security, using a Wireless Transport Layer Security protocol (WTLS). This protocol uses RSA-based cryptography. However another protocol is also under consideration called Elliptic-Curve Cryptography (ECC). This protocol provides high level of security and using less memory resources and computation.

Another widely used authentication protocol is the wireless public-key-infrastructure mechanism (PKI), which is based on the wired PKI mechanism. Some of the products that use this mechanism are Certicom, eTrust, VeriSign.

Some of the new wireless-security standards that are under development are:
Pre-IKE Credential (PIC), where IKE stands for Internet Key Exchange: It is a protocol proposed by the IETF's IP Security Remote Access Working Group. This protocol

provides additional features like flexibility and ease of configuration to the IPSec (IP Security) standard.

Open Multimedia Applications Protocol (OMAP): This protocol was developed by Texas Instruments. "It is a library of software from various vendors that will permit secure transactions on wireless devices that use TI's digital signal processors"[3].

Biometrics: It is a new system that identity's authorized user using their unique physical characteristics like finger prints, voice patterns, facial geometry, or retinal images.

6. Conclusion

Initially, when the wireless mobile environment came into existence security was not given a priority. But, as the time passed by, the extent to which this technology is used increased. This created a need to protect the information from unauthorized users and control the fraud. In the beginning, many security protocols where proposed, which where based on cryptographic techniques. With new loopholes coming up each time, a new protocol was proposed based on the existing one, to answer the problem. Presently, many researchers are concentrating on using the wired based security protocols over the wireless mobile communication. One such research is taking place at Sun Microsystems Labs, where the KSSL protocol is being tested within the corporate campus, using concepts like smart card and certificates.

According to me, the present research that is going on, that is trying to extend the security protocols used in wired networks to wireless mobile environment is a good step in providing high-end security. As the security protocols used in wired network have undergone heavy scrutiny over the years from various ends using these protocols in the mobile environment, would help in achieving good performance results.

Also, many wireless communication service vendors are developing new protocols and standards to provide a secured medium for the mobile users. With these efforts relatively new and not yet developed to its full extent, service providers are hoping to keep security development in pace with other developmental aspects of wireless technology.

References:

[1] Chang-Seop Park, " On Certificate-Based Security Protocols for Wireless Mobile Communication Systems."IEEE Network 1997

[2] Vipul Gupta and Sumit Gupta "Securing the Wireless Internet" IEEE Communications 2001.

[3] S.K Miller, "Facing the Challenges of Wireless Security", IEEE Computer, 2001.

[4] Gopal Racherla, Debashis Saha, " Security and Privacy issues in Wireless and Mobile Computing " IEEE Proceedings, 2001.

[5] Jun-Zhao Sun, Douglas Howie, Antti Koivisto, Jaakko Sauvola, " A Hierarchical Framework Model of Mobile Security", IEEE 2001.

[6] Subramanyam, Anupam Joshi, “ Security in Mobile Systems”, IEEE Proceedings 1998.
[7] Asokan, "Security Issues in Mobile Computing," Univ. of Waterloo, Dept. of Computer Science, Technical Report CS690B, Apr. 1995.
[8] Charlie Perkins, “ Mobile IP and Security Issue: An Overview”
[9] Mavridis Pangalos, ”Security Issues in a Mobile Computing Paradigm.”