Security in Mobile Communications: Challenges and Opportunities

Audun Jøsang **Gunnar Sanderud**

Distributed Systems Technology Centre*
QUT, Brisbane, Qld 4001, Australia
Email: ajosang@dstc.edu.au

Ericsson
1371 Asker, Norway
Email: gunnsan@online.no

Abstract

The nature of mobile communication, characterised for example by terminals having poor user interface and limited processing capacity, as well as complex combination of network protocols, makes the design of security solutions particularly challenging. This paper discusses some of the difficulties system architects are faced with as well as some advantages mobile networks offer when designing security solutions for mobile communication.

Keywords: Security, usability, heterogeneous networks, mobile devices

1 Introduction

Over the last few years, a number of mobile communication systems have been developed and numerous service providers and equipment vendors are bringing to market a steady stream of new innovations. Underneath the hype and publicity over these new technologies lie the design specifications and the physical properties that define the capabilities and limitations of mobile communication networks.

When looking at traditional e-commerce, the lack of security and a high level of fraud is seen as the major obstacle to people embracing the possibilities and advantages e-commerce can offer. Considerable effort is therefore put into developing security for e-commerce. One typical example is that Web browsers and servers are enabled to use public-key infrastructures for cryptographic key distribution and to use cryptographic protocols such as SSL (Netscape 1995) (a.k.a. TLS (Dierks & Allen 1999)) for communication security. Unfortunately, communication security alone is not enough. Ensuring system security at both the client and the server end must not be ignored. On the client side, the poor platform integrity, the multitude of default CA certificates and the arcane user interface pose severe security threats. The high level of vulnerability on the server side is best illustrated by the fact that almost all reported hacker attacks are targeted against servers. System security can be addressed by installing firewalls and intrusion detection systems, by monitoring security alerts and prompt implementation of security patches. However this requires skilled system administrators to continuously look after systems, which is relatively labour intensive compared to communication security.

*The work reported in this paper has been funded in part by the Co-operative Research Centre for Enterprise Distributed Systems Technology (DSTC) through the Australian Federal Government's CRC Programme (Department of Industry, Science & Resources)

Flexibility and functionality are key factors for creating successful e-commerce applications. What is often ignored is that there is a trade-off between functionality and security. A typical characteristic of security is that it provides no additional functionality to an application other than security itself, i.e. it does not provide the functionality in which users are primarily interested. When application developers and users have to make a choice they therefore go for functionality rather than security. According to Andersen (Andersen 2001) p.519 the current insecurity of commercial systems on the Internet is thus perfectly rational from the economists' viewpoint, however undesirable from the users'.

For m-commerce, there is a risk that a similar development will take place, but not necessarily so. In m-commerce the limited size and poor user interface of mobile devices pose particular problems for implementing user friendly applications in general and even more so for security. On the other hand, the characteristics of mobile architectures, for example because mobile networks often are strictly controlled, can make it easier to obtain satisfactory security . This paper discusses the feasibility of implementing security in mobile applications. In order to design successful security solutions it is important to recognise the particular aspects of mobile applications and the conditions under which mobile devices will be used. Some aspects of mobile networks make security design hard whereas other aspects provide solid building blocks for security that are normally not found in other networks.

2 Requirements for Communication Security

Communication security is often described in terms of confidentiality, integrity, authentication and non-repudiation of transmitted data. These security services are in turn implemented by various mechanisms that are usually cryptographic in nature. See e.g. IS-7498-2 (ISO 1988) for a concise description of communication security services and mechanisms. In addition there is confidentiality of traffic (i.e. whether or not communication is taking place), of location (where the communicating parties are located) and of the communicating parties' address, all of which are important for privacy. A casual level of security is usually provided implicitly even without taking any extra measures. For example in order to eavesdrop on a particular person's mobile phone conversations the eavesdropper has to be located in physical proximity to the person and carry special radio equipment which in itself represents a certain level of protection. Casual authentication between mobile phone users is indirectly provided by the calling and called party numbers. In case of of voice telephony, authentication results from recognising the other person's voice.

Cryptography on the other hand gives the possibility of designing strong security services but often creates inconveniences when using the application. The use of cryptography therefore makes most sense in case of sensitive

applications. When strong cryptographic security mechanisms are in place the remaining vulnerabilities are usually due to poor management and operation and not by weaknesses in the cryptographic algorithms themselves.

Confidentiality of transmitted data can be provided by encrypting the information flow between the communicating parties, and the encryption can take place end-to-end between the communicating parties or alternatively on separate legs in the communication path. In GSM networks for example, only the radio link between the mobile terminal and the base station is encrypted whereas the rest of the network transmits data in clear-text. Radio link confidentiality in GSM is totally transparent from the user's point of view. Mechanisms for implementing confidentiality of traffic, location and addresses will depend on the technology used in a particular mobile network.

Authentication of transmitted data is an asymmetric service, meaning for example that when A and B are communicating, the authentication of A's data by B is independent from the authentication of B's data by A. The types of authentication available will depend on the security protocol used. In the Internet for example, SSL allows encryption with four different authentication options: 1) server authentication, 2) client authentication, or 3) both server and client authentication or 4) no authentication, i.e. providing confidentiality only.

Non-repudiation is similar to authentication in that it is an asymmetric security service. A simple way to describe the difference between authentication and non-repudiation is that with authentication the recipient himself is confident about the origin of a message but would not necessarily be able to convince anybody else about it, whereas for non-repudiation the recipient is also able to convince third parties. Digital signature is the mechanism used for non-repudiation. Cryptographicly seen a message's authentication code and non-repudiation code can be identical, and the difference between the two services might only depend on the key distribution. In general, if a signature verification key has been certified by a trusted third party the corresponding digital signature will provide non-repudiation, whereas it can only provide authentication if the key has simply been exchanged between the two communicating parties.

Different parties will have different interests regarding authentication and non-repudiation services. Network operators are interested in authenticating the users for billing purposes and to avoid fraud. Users and content service providers are interested in authenticating each other and might also be interested in authenticating the network service provider. How and where in the network authentication services are implemented will depend on the technology used and the business models involved.

3 The Network Operator as Trusted Third Party

Public-key cryptography is the basis of several important security services such as non-repudiation and authentication and is an essential element for SSL that is used for securing Web communication. One public/private key pair is used for authenticating one party by the other, and mutual authentication requires two key pairs. In fact, every entity on the Internet needs a key pair if it shall be possible for an arbitrary entity to authenticate any other entity. It has therefore been predicted that every player on the Internet will have its own public/private key pair which will form the basis for the user's or organisation's digital identity in electronic environments. This requires the secure generation and distribution of potentially hundreds of millions of public/private key pairs, which poses a formidable key management challenge.

A PKI refers to an infrastructure for distributing public keys where the authenticity of public keys is certified by Certification Authorities (CA). A public key certificate basically consists of the CA's digital signature on the public key, usually together with some attributes. If the certificate owner's identity is one of the attributes, then the certificate is called an identity certificate, and the purpose of the certificate is to link the public key and the identity together in an unambiguous way. The CA is a Trusted Third Party (TTP) because it is trusted to correctly verify and certify the identity of the public-key owner before issuing the certificate. The structure of identity public-key certificates is standardised by the ITU X.509 standard (ITU 1997). In order to verify a certificate the CA's public key is needed, thereby creating an identical authentication problem. The CA's public key can be certified by another CA etc., but in the end you need to receive the public key of some CA, usually called the root CA, out-of-band in a secure way. This is difficult to achieve with a handful of global CAs serving the whole Internet community as the case is for the Web PKI. If CAs are either local and/or serve a limited number of relying parties then trust relationships can be much stronger, and out-of-band distribution of CA root public keys and user private keys can be much more secure.

In case of subscription based mobile networks there exists a formal relationship between users/subscribers on one hand and the network operator on the other. It would therefore be natural to let the network operator play the role of CA. The user's private key as well as the root CA public key can be distributed in a secure way based on the distribution of subscription tokens e.g in the form of the GSM SIM card. Operators who have formed roaming agreements between each other already have a formal relationship which could be extended to cross-certification of each other public keys. Mobile network operators therefore are in a very strong position to establish themselves as CAs, and the mobile device, or more precisely the security token, naturally lends itself to become a secure storage medium for these cryptographic keys. The Web PKI suffers from insecure distribution and storage of cryptographic keys and therefore does not provide a complete chain of trust. To combine the roles of CA and mobile network operator would make it easier to have a complete chain of trust around the PKI because there already exists a trust relationship between mobile network operators and their customers.

Network operators should also explore the possibility of becoming close partners with financial institutions or alternatively establish themselves as independent financial mediators by allowing m-commerce transactions to be billed on subscriptions. This has already happened on a small scale in cases when customers can buy e.g. soft drinks from vending machines by placing a call to a premium rate number linked to the vending machine. An evolution from this primitive type of payment to a more general and flexible form could be driven by the network operators. One of the major problems in e-commerce is the lack of customer authentication. The fact that network operators already have strong subscriber authentication puts them in a natural position to become an intermediary between customers and vendors. This will require a relationship between vendors and network operators similar to the relationship between vendors and credit card companies.

4 Security Across Heterogeneous Networks

Network architectures are based on protocol layers which represent an abstract way of modelling and implementing data transmission between communicating parties. The usual protocol architecture consists of 5 layers as illustrated in Figure 1 below.

In reality, no data are directly transferred between adjacent layers on opposite sides. Instead, data and control information are passed down through the interfaces between the protocol layers on one side and up through the interfaces between the protocol layers on the other side. The physical data transmission actually takes place through a

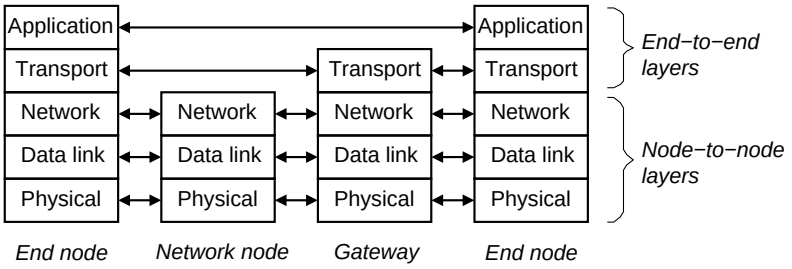

Figure 1: Communication protocol layers

physical medium underneath the physical protocol layer.

The physical, data-link and network layers are node-to-node whereas the transport and application layers usually are end-to-end except when there is a gateway in between. Figure 1 shows an example with a gateway in the transport layer in which case the transport layer no longer can be considered end-to-end.

The network architecture and the security goal together indicate the most appropriate protocol layer where a security service is to be located. Authentication and non-repudiation are for example only meaningful when implemented end-to-end between the parties that need to authenticate each other. Confidentiality and integrity on the other hand can be meaningful by encrypting isolated legs between nodes when it can be assumed that only these legs would be vulnerable to attack. By using Figure 1 as an example, authentication and non-repudiation must thus be implemented on the application layer, whereas confidentiality can be implemented on any layer.

Mobile applications usually span over several networks such as for example a radio network and a fixed network requiring gateways on the transport or application protocol layers. This complicates the implementation of security services because it becomes more difficult to obtain end-to-end security. As a general rule authentication must always be built on top of an end-to-end layer. Whenever confidentiality is based on encryption with a session key obtained through the authentication protocol it is natural to let encryption be end-to-end as well.

An example of a mismatch between desired security service and protocol layer can be seen in the original WTLS protocol (Wireless Transport Layer Security) (WAP-Forum 2000*a*). WTLS is intended to work similarly to SSL for example by providing authentication. However because the WTLS protocol terminates in the transport layer gateway it is not able to provide authentication between the WAP terminal and the WAP service provider, but only between the WAP terminal and the WAP gateway as illustrated in Fig.2 below.

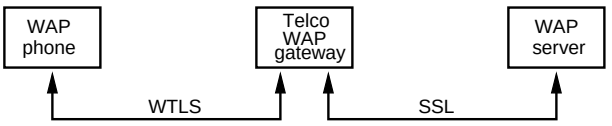

Figure 2: Security architecture using WTLS

In addition to making authentication meaningless, this solution also creates an unavoidable plain text gap in the WAP gateway. WTLS is specified with 3 functionality classes so that the security features can be introduced in steps. Table 1 describes the functionality of each class, where ”M” means mandatory and ”O” means optional.

Class 1 specifies public-key exchange without server or client certificates and is based on the Diffie-Hellman key exchange protocol (Diffie & Hellman 1976) , which provides encryption and confidentiality but no authentication. So far only WTLS Class 1 is being used in mobile WAP applications. Class 2 requires using server certificates and is supposed to provide server authentication. Class 3 requires using client certificates and is supposed to allow user authentication by the service providers.

Feature	**Class 1**	**Class 2**	**Class 3**
Public-key-exchange	M	M	M
Server certificates	O	M	M
Client certificates	O	O	M
Shared-secret handshake	O	O	O
Compression	-	O	O
Encryption	M	M	M
MAC	M	M	M
Smart card interface	-	O	O

Table 1: WTLS Classes (WAP-Forum 2000*a*, p.89)

However, when considering that the original WTLS architecture does not provide end-to-end security it is obvious that the WTLS classes 2 and 3 would be meaningless. The original WTLS can only ever provide authentication of the WAP gateway which is of no value to users or WAP service providers. What users and service providers want is to be able to authenticate each other.

We find it surprising that WTLS originally suffered from this serious design weakness considering that the development of this technology was given top priority by some of the world’s most prestigious IT and telecommunications companies. One possible explanation for introducing the WAP gateway could have been to give the mobile network operators more control of the traffic and transactions and thereby allow specific business models. It could also have been to facilitate legal government interception of traffic contents from the WAP gateway clear text gap. With an end-to-end security architecture this would change, and in fact become similar to the existing security architecture on the Internet.

Because of the deficiencies in the original WTLS protocol the WAP forum has defined a new standard for end-to-end SSL using tunnelling through the WAP gateway (WAP-Forum 2000*b*). This is achieved by implementing a wireless enhanced version of the Internet TCP transport protocol layer in the mobile devices and run SSL on top of that. However, because the origin server will probably not support wireless enhanced TCP, there will be a proxy that acts as the termination point of two TCP sessions, one w-TCP to the client mobile terminal and one TCP to the server. It will just move packets across transparently from one connection to the other.

5 Usability of Security

Details of the security services and mechanisms are often complex and users would quickly be overloaded with information if the details were presented to them. A common design philosophy is therefore to make security services and mechanisms as transparent as possible. How-

ever there is a danger that users receive too little security information. If security is totally hidden from the user he or she would not be able to tell whether it is working the way it was intended, which in turn could allow successful attacks to remain undetected. Obviously, the security evidence provided can not be more than the user can understand and handle but it must be sufficient for the required security level of the application. The challenge is to determine what type of evidence is really necessary and present it to the user in an intuitive and intelligible way.

In the computer network jargon it is sometimes forgotten that communication ultimately goes between human users and organisations, and that some security services only are meaningful if they are designed to suit human users. The interpretation of communication in the human brain can conceptually be described as a semantic protocol layer above the application layer as depicted in Figure 3 below.

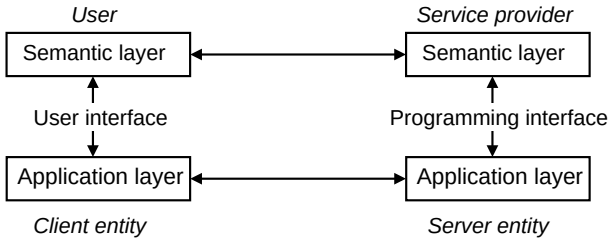

Figure 3: Semantic protocol layer between human users and organisations

Between the application and the semantic layers lies the user interface, and this is not something that can be specified and implemented in a software module as it is done for the other protocol layer interfaces. A good user interface represents an intricate combination of multimedia and optimal terminal design adapted to the human physical and mental capabilities. The study of how security information should be handled in the user interface forms part of *security usability*. This field of research has mostly been ignored by researchers as well as application and hardware developers. Whitten and Tygar (Whitten & Tygar 1999) argue that effective security requires a different usability standard, and that it can not be achieved through the user interface techniques appropriate to other types of consumer software.

One example illustrating the subtleties of security is the padlock icon on Web browsers where an open padlock indicates insecure communication whereas a closed padlock indicates secure communication. This is seemingly a very neat and intuitive way of indicating that a Web server has been authenticated with SSL and that transmitted and received data are being encrypted. However a closed padlock only tells the user that some Web server has been authenticated but not which Web server in particular. As long as the user does not do the extra mouse clicks to view the server certificate he or she has not authenticated anything at all. Despite the appearance of being very simple, the padlock hides crucial aspects of security, without which authentication becomes meaningless.

The Web browser does allow viewing the server public-key certificate by clicking on the padlock icon, but users hardly ever do this, and even security aware users who view the certificate when accessing a secure Web site can have difficulty in judging whether the certificate really is what it claims to be. The browser usually checks that the domain name in the certificate is the same as the domain name pointed to by the browser, and aware users might notice when an intruder's domain name is different from the expected domain name. However, users do not usually inspect the URL for the domain name when browsing the Internet Web, and many companies' secure Web sites have URLs with non-obvious domain names that do not correspond to the domain names of their non-secure Web sites. One example is the Norwegian bank Nordea with the URL: `http://www.nordea.no` and where its secure on-line banking has URL: `https://ibank.bbsas.no/iBank/Dispatcher`. Another vulnerability is the fact that distinct domain names can appear very similar, for example differing only by a single letter so that a false domain name may pass undetected. How easy is it for example to distinguish between the following URLs: `http://www.bellabs.com`, `http://www.belllabs.com`, and `http://www.bell-labs.com`?

In order to make authentication on the Internet Web more meaningful some familiar elements from the physical world could be used. Jøsang *et al.* (Jøsang, Patton & Ho 2001) have proposed to display a digitally certified company logo in the Web browser to allow meaningful authentication at a glance and bridge the gap between the cryptographic mechanisms and the human user. This idea is presently discussed in the IETF and may become a standard feature in the future (see (Santesson 2001)).

For mobile devices the relatively small visual display will make it virtually impossible to inspect public-key certificates for authentication. Cryptographic authentication by identity certificates such as X.509 will be unreliable because of the difficulty of comparing an Internet site name with the identity stored in the digital certificate. Figure 4 shows a typical hand-held WAP device with which server authentication will be meaningless.

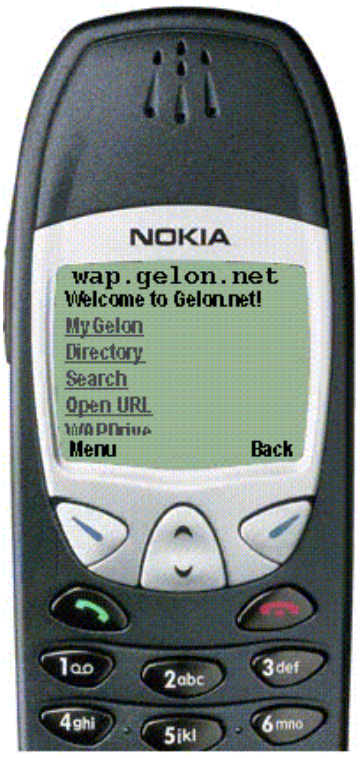

Figure 4: Interface that is unable to provide meaningful WAP server authentication

By typing the correct URL of a WAP or Web site, authentication is not really needed as long as the integrity of the network is preserved, i.e. you will access the right site as long as you type the URL correctly. WAP sites are more likely to be accessed through portals than by typing URLs, which makes other forms of authentication the more important. However, cryptographic authentication mechanisms are only meaningful if the interface is able to provide authentication information in a secure way. For mobile devices with small display, certified company logos seem to provide a good solution.

The integrity of the evidence presented to the user can be assured by having a reserved area for certified content on the interface which is never used for other types of content. Because of limited size of visual displays this might seem to be an expensive sacrifice. We therefore recommend using the normal display for displaying security information, but in a special security mode, and instead to reserve a small exclusive area to indicate that the display is in security mode. The exclusive security display area and the security display mode should not be accessible by content applications. This security mode should be easy to invoke and be distinguishable from the other display modes. The security mode of the interface then represents a separate interface channel that can be distinguished from the normal information content channel.

What represents the most suitable type of certified information to be displayed will depend on the application. A simple solution from an implementation point of view is to link the authentication directly to the logical network address used such as e.g. a telephone number or Internet domain name, and display the certified address in the separate control field. The user would then be required to know exactly which network address he or she wants to contact, but this can be problematic as mentioned above.

Certified company logos, pictures of persons and sound files can be easier to perceive as distinguished qualifiers than simple names. There are however new problems that need to be solved before such solutions can be implemented.

Image and sound can only be used for strong authentication if the image and sound files are certified and included in digital certificates. This requires the CA to verify their authenticity before issuing certificates. A company logo must for example be sufficiently different from all other company logos and this requires the CA to perform a similarity check, but this is likely to create new problems. What are the criteria for a similarity check? If similar logos or names are used by companies in totally different businesses, is that OK? According to Stubblebine and Syverson (Stubblebine & Syverson 2000) hierarchies adequate to issue certificates are not by themselves adequate to ensure global uniqueness. Suppose that a company obtains a certificate for a logo and then another company applies for a certificate for a much too similar logo, but it owns that logo as a registered trademark? More generally, what about revocation of a logo because of previously unrecognised problems? Does every little shop need to hire a graphic artist? What is the size of the space of meaningfully discernible logos? The authenticity of pictures of persons can best be assured by taking the photos on the CA's premises. Similar requirements apply to sound files, i.e. they must be recorded in person on the CA's premises.

Verifying these additional elements will require the CA's to be physically more local to users or organisations, making it difficult for one CA to serve the whole world. In Sec.3 we described how PKI operations can be simplified by letting the mobile network carrier act as CA.

6 Securing Active Contents

Before active content was available Web pages were mainly static displays of information coded in the Hyper Text Markup Language (HTML). Active content allows sound and image animation and provides the user with the ability to interact with the server side during a Web session. Active content exists in many forms. Java applets and ActiveX controls are some of the best known but there are also JavaScripts, VBScripts, MSWord Macros and even images. All these basically consist of mobile code that is sent from the Web server and loaded into the client machine for execution there.

All this is very appealing from a functionality and flexibility point of view but it poses a formidable threat to the integrity of the client machine. Active content can cause damage by intent or by simply being poorly designed. A discussion of threats and risks posed by active contents can be found in (Jansen 2001). An attack using malicious applets is described in (Lefranc & Naccache 2002). Firewalls offer little protection because they are usually configured to let http traffic and active content through. Unless the active content can be controlled, all files and network connections can be accessed and (mis)used, making it impossible to operate any secure applications on the client machine. Sandboxing and certification can be used to counter threats from active content.

Sandboxing basically means that the active content is constrained in what resources it can access on the host system. The advantage is that it is always active and completely transparent to the user. The disadvantage is that it severely limits the capabilities of active contents.

Certification means that a trusted party has validated and digitally signed the active content and that the platform verifies the digital signature before it can execute. The advantage is that the active content can access all system resources. The disadvantage is that certification is not equivalent with trustworthiness. A Web browser can for example be tuned so that any piece of certified active content is accepted by default or alternatively so that only active content certified by certain parties is accepted by default and that any other trigger a dialog box. The dialog box basically asks the user whether he or she wants the active content to be executed. Experience shows that users almost always accept active content when asked by a dialog box simply because they want the functionality and because most active content is benign anyway. This means that should the user receive a piece of malicious active content he or she will almost certainly make the wrong decision and accept it. The user simply does not have sufficient evidence to make an informed decision.

A similar development is taking place in the market for downloadable executables in mobile terminals, but the maturity of this technology is still behind that of the Web, and due to technical constraints is likely to follow a slightly different path.

Presently there is no standard protocol for downloading executables to mobile terminals such as http on the Web. Neither is there a standard execution environment for running executables on mobile terminals such as the Java Virtual Machine in Web browsers. WAP was originally perceived as a method by which all types of content would be downloaded, including WMLScripts which allow minimally executable applications to be run on mobile terminals. However, WAP never achieved its envisioned acceptance in the market, and was not designed to provide an execution environment for programs written in Java or other rich programming languages.

Several stakeholders have started to roll out new technology to correct these deficiencies, and the question to ask is whether these solutions will provide the necessary security to protect the mobile terminals against harmful active contents.

Sun Microsystems has introduced Java 2 MicroEdition (J2ME) and the Kilobytes Virtual Machine (KVM) to provide an application execution environment for constrained devices such as mobile terminals. A complementary and compatible technology for downloading content and executables to mobile terminals called "Download Fun" has been introduced by Openwave Systems. According to the Download Fun FAQ (OpenwaveSystems 2002) it provides a mechanism to download binary objects from a content site to a mobile device in a secure manner. What that really means is that the Download Fun Client supports a variety of security protocols including SSL and WTLS Class II, meaning for example that the executables can be digitally signed, and that the signature must be verified before the executable can run on the mobile terminal. The network operator will normally sign the executable, and a revenue sharing scheme will make sure that third party application

developers get included in the revenue stream.

Qualcomm has also entered the market with Binary Runtime Environment for Wireless (BREW) (Qualcomm 2002). BREW encompasses both an application execution environment and a mechanisms for downloading executables. Qualcomm has also introduced a scheme for digitally signing executables, and this forms part of BREW's particular business model. Qualcomm will be the only authority to validate BREW applications, and the digital signature will be applied by Qualcomm, the application developer and the network operator in concert. In that way Qualcomm makes sure it always gets included in the revenue stream. The reason why Qualcomm believes the market will accept this type of monopolistic business model is that they control the production of chips for CDMA[1] phones, and will make sure that every CDMA chip is BREW-enabled at no extra cost to the mobile phone manufacturers.

What remains to see is whether these two (and other) technologies will restrict users to only download end run digitally signed executables.That would require the mobile terminals to be designed so that the network operator controls what executables the terminals can run. Our guess is that users will want solutions that give greater flexibility and allow running any executable if they so desire, which necessarily will create security vulnerabilities. In fact mobile phones have already been the target of various types of malicious active content as for example reported in (Krane 2002).

As a result additional security mechanisms are needed to protect against harmful executables. Using sandboxing and dialog boxes obviously comes to mind, but unfortunately these mechanisms seem even less suitable in mobile terminals than they are in Web browsers. The challenge is therefore to come up with alternative and better solutions.

Present mobile phones do not have enough memory to run traditional anti-virus software. A solution suggested by Hoshizawa (Hoshizawa 2002) is to run anti-virus software at the network level so that network operators and ISPs can block virus outbreaks and thereby prevent them from spreading.

On a non-technical level it is of course always a good idea to improve user awareness and hygiene in download habits. This may seem like daunting task given that it would require educating the global mass consumer market. Nevertheless network operators should have an obligation to make an effort towards greater awareness about mobile phone security.

7 Conclusions

We have seen that the aspects of mobile networks can make it both harder and easier to implement communication security as compared to for example the Internet. Communication between mobile and fixed networks create particular problems regarding security protocol design. Mobile devices usually have a poor user interface thereby creating problems for the usability of security. On the positive side, the mobile network operators are well placed to become trusted third parties and thereby be able to support security applications. There is also a potential for the network operators to control downloadable executables and thereby be able to filter out harmful executables, but it is questionable whether that will be accepted by the market. As a general rule in the development of e-commerce technology, functionality and flexibility always gets highest priority because they form the basis for new business models. Security usually serves as a remedy to solve whatever vulnerabilities emerge, and not as a primary goal in its own right.

[1] CDMA (Code Division Multiple Access) is a technology for wireless communication used in all 3G and in some 2G mobile networks.

References

Andersen, R. (2001), *Security Engineering*, Wiley.

Dierks, T. & Allen, C. (1999), *RFC2246 - The TLS (Transport Layer Security) protocol, Version 1.0*, IETF. URL: http://www.ietf.org/rfc/rfc2246.txt.

Diffie, W. & Hellman, M. E. (1976), 'New directions in cryptography', *IEEE Transactions on Information Theory* **22**(6), 644–654.

Hoshizawa, Y. (2002), Are Java-Enabled Mobile Phones Secured?, *in* U. Gattiker, ed., 'EICAR Conference Best Paper Proceedings', European Institute for Computer Anti-Virus Research (EICAR), pp. 141–151.

ISO (1988), *IS 7498-2. Basic Reference Model For Open Systems Interconection - Part 2: Security Architecture*, International Organisation for Standardization.

ITU (1997), *Recommendation X.509, The Directory: Authentication Framework* (also ISO/IEC 9594-8, 1995), International Telecommunications Union, Telecommunication Standardization Sector(ITU-T).

Jansen, W. A. (2001), Guidelines on Active Content and Mobile Code – NIST Special Publication 800-28, Technical report, National Institute of Standards and Technology.

Jøsang, A., Patton, M. & Ho, A. (2001), Authentication for Humans, *in* B. Gavish, ed., 'Proceedings of the 9th International Conference on Telecommunication Systems (ICTS2001)', Cox School of Business, Southern Methodist University.

Krane, J. (2002), 'As mobile devices get 'smarter', they become prone to viruses', SiliconValley.com - Mercury News, URL: http://www.siliconvalley.com/mld/siliconvalley/2833740.htm.

Lefranc, S. & Naccache, D. (2002), 'Cut and paste attacks with java', Cryptology ePrint Archive, Report 2002/010. http://eprint.iacr.org/.

Netscape (1995), *The SSL (Secure Sockets Layer) 3.0 Protocol*, Netscape Communications Corp.

OpenwaveSystems (2002), 'Download Fun FAQ', URL: http://developer.openwave.com/prod_tech/faq-df.html.

Qualcomm (2002), 'BREW Whitepaper', http://www.qualcomm.com/brew/about/brewwhitepaper.pdf.

Santesson, S. (2001), *Logotypes in X.509 certificates*, IETF PKIX Working Group INTERNET-DRAFT. URL: http://www.ietf.org/internet-drafts/draft-ietf-pkix-logotypes-00.txt.

Stubblebine, S. G. & Syverson, P. F. (2000), Authentic Attributes with Fine-Grained Anonymity Protection, *in* 'Proceedings of Financial Crypto'.

WAP-Forum (2000*a*), *WAP-199, WTLS (Wireless Transport Layer Security), Version 18-Feb-2000*, URL: http://www.wapforum.org/what/technical.htm.

WAP-Forum (2000*b*), *WAP-219, WAP TLS Profile and Tunneling, Prototype Version 04-Dec-2000*, URL: http://www1.wapforum.org/tech/documents/WAP-219-TLS-20001204-t.pdf.

Whitten, A. & Tygar, J. (1999), Why Johnny Can't Encrypt: A Usability Evaluation of PGP 5.0, *in* 'Proceedings of the 8th USENIX Security Symposium'.