Mobile platform security

Christian Gehrmann and Per Ståhl

End-user expectations regarding features in mobile phones have increased dramatically in recent years, not the least where security functionality is concerned. End users, operators, and content providers take it for granted that the mobile platform will protect all vital information in and about a mobile phone. And Ericsson is proud to say that thanks to robust security mechanisms and rich, basic-platform-security functionality, its new generation of mobile platforms fully lives up to these expectations.

The authors describe the hardware and software security architecture and building blocks implemented in the new mobile platforms. In particular, they describe "core" security functionality such as secure boot, secure reprogramming, and the protection of critical parameters.

Introduction

The term security lacks meaning until one has defined what is to be secure and for whom. Likewise, security is difficult to comprehend without a potential threat. Mobile phones for third-generation mobile systems (3G) have several security stakeholders for which the mobile platform must provide security services. Moreover, the potential threats may differ from stakeholder to stakeholder.

The first class of security stakeholders, users, expects that mobile phones will offer secure and reliable communication – that is, they assume their phones can be trusted to handle sensitive tasks, such as e-commerce transactions. The main threats to this class of stakeholders are malicious software, such as viruses and Trojans, or weak or misbehaving security mechanisms.

The second class of stakeholders, mobile network operators, relies on phone network identification mechanisms (related to billing capability) and network-related software. Users or hostile software must not be allowed to circumvent these mechanisms. Operators thus require that the integrity of software can be guaranteed when the mobile phone is in operation. They also want to be certain that users cannot break SIM lock mechanisms.

A third class of security stakeholders, content providers, wants to be paid for the content (music, pictures, videos and software) that users download. It also wants to know that users cannot (mis)use their phones to illegally copy or distribute content. This is where digital rights management (DRM) functions come into play. However, DRM mechanisms alone cannot provide all necessary security. To provide a DRM solution that meets content provider requirements, the mobile phone platform must contain security functions that guarantee secure execution and code integrity.

Security is usually measured in terms of a set of basic aspects: confidentiality, integrity, authentication and authorization. Confidentiality of data is achieved by cryptographically transforming original data, often called, plaintext, into cipher text, which hides the content of plaintext. This operation is realized as a parameterized transformation that keeps the controlling parameter secret. The controlling parameter is often called a key. The transformation is called encryption. With a key it is easy to perform the inverse transform or decryption. Without the key, decryption should be difficult.

Integrity is about ensuring that data has not been replaced or modified without authorization during transport or storage. This is achieved using cryptographic transforms and a key. Additional information must also be added to the plaintext to verify its integrity.

Authentication is the procedure by which a unit (the claimant) convinces another unit (the verifier) of its (correct) identity. Authentication is different from authorization,

BOX A, TERMS AND ABBREVIATIONS

2G	Second generation
3G	Third generation
3GPP	Third Generation Partnership Project
AES	Advanced encryption standard
AKA	Authentication and key agreement
APDU	Application protocol data units
API	Application program interface
CA	Certificate authority
CBC	Cipher block chaining
CID	Customer ID
CMS	Cryptographic message syntax
CPU	Central processor unit
DES	Data encryption standard
DH	Diffie-Hellman
DRM	Digital rights management
DSP	Digital signal processor
DSS	Digital signature standard
DVB-H	Digital video broadcast - handheld
EAP	Extensible authentication protocol
EICTA	European Information & Communications Technology Industry Association
GPRS	General packet radio system
GSMA	GSM association
HMAC	Hash calculation, MAC calculation
IEEE	Industry of Electrical & Electronics Engineers
IKE	Internet key exchange
IMEI	International mobile equipment identification
IP	Internet protocol
IPDC	IP device control
IPsec	Secure IP
ISMA	Internet Streaming Media Alliance
MAC	Message authentication code
MBMS	Multimedia broadcast/multicast service
MD5	Message digest 5
OCSP	Online certificate status protocol
OMA	Open Mobile Alliance
OS	Operating system
OTA	Over the air
OTP	One-time programmable
PIN	Personal identification number
PKCS	Public key cryptography standard/system
PKI	Public key infrastructure
R&D	Research and development
RAM	Random access memory
ROAP	Rights object acquisition protocol
ROM	Read-only memory
RNG	Random number generator
RSA	Rivest, Shamir & Adleman
SAT	SIM application toolkit
SATSA	Security-and-trust-services API
SHA-1	Secure hashing algorithm 1
SIM	Subscriber identity module
SMS	Short message service
SRTP	Secure real-time transport protocol
SSL	Secure sockets layer
TCP	Transport control protocol
TLS	Transport layer security
UICC	UMTS integrated circuit card
UMA	Unlicensed mobile access
UMTS	Universal mobile telecommunications system
USB	Universal serial bus
USIM	Universal SIM
VPN	Virtual private network
WAP	Wireless application protocol
WCDMA	Wideband cell-division multiple access
WIM	WAP identity module
WLAN	Wireless local area network

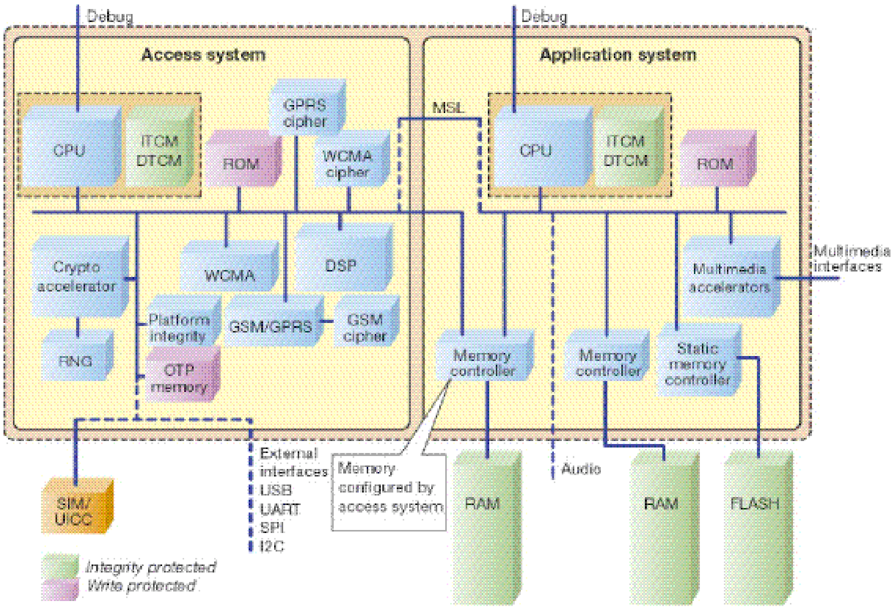


Figure 1
View of platform security hardware.

which is the process of giving a person or entity permission to do or have access to something.

There are two major classes of cryptographic mechanisms: symmetric and asymmetric. In symmetric mechanisms, the same key is used for encryption and decryption. Examples of symmetric confidentiality mechanisms are

- block ciphers, such as DES and AES; and
- stream ciphers, such as the GSM A1, A2 and A3 algorithms.

Integrity is often protected using symmetric mechanisms. Integrity-protection algorithms are also called message authentication codes (MAC). The most popular MAC is the HMAC algorithm.[1] Because the key in symmetric mechanisms can be used to decrypt content, it must be kept secret from all but legitimate users of the encryption scheme.

Asymmetric mechanisms use separate pairs of keys for encryption transform and decryption transform. The public key can be made publicly available, but the private key must never be revealed. Asymmetric mechanisms are typically used for distributing keys (for example, a symmetric key) or for digital signing purposes. A public key can be used to encrypt a symmetric key, which in turn, can only be decrypted by the legitimate receiver using the corresponding private key. A private key may also be used to digitally sign data. The signature can be verified by anyone who knows the corresponding public key. The RSA scheme is widely known example of an asymmetric cryptographic algorithm.

Ericsson has designed the security architecture of its mobile platforms to meet the security requirements of different stakeholders. The architecture is built around a combination of hardware and software components that support the implementation of mechanisms that provide security. The main security functions are

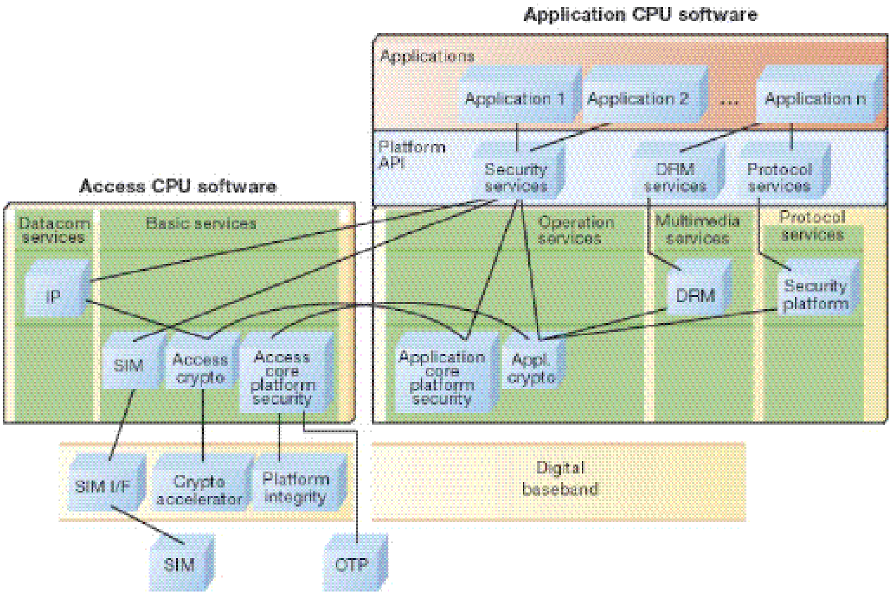
- secure boot and software integrity;
- secure control of debug and trace capabilities;
- digital rights management;
- hardware cryptographic accelerators;
- hardware-based random number generator (RNG);
- cryptographic algorithm service;
- public key infrastructure (PKI) support; and
- secure communication protocols (GSM/GPRS/WCDMA security, TLS/SSL, IPsec, and Bluetooth/WLAN).

These functions either fulfill security expectations or have been included to accelerate cryptographic operations.

Platform and security architecture

The second-generation platform architecture, called A2, interconnects an access CPU and an application CPU through a serial link from Intel. The access CPU handles standard communication protocols; the application CPU handles user application functionality. This division of labor sepa-

Figure 2
Security software function blocks and architecture.



rates real-time control from high-performance execution. Figure 1 gives a schematic view of the hardware configuration.

A digital signal processor (DSP) performs critical, real-time computations for the access system, and a variety of hardware blocks perform application-specific tasks. Five blocks or modules (GPRS cipher, GSM cipher, WCDMA cipher, cryptographic accelerator, and the platform integrity module), for example, perform cryptographic computations for the access system.

The first three modules (GPRS cipher, GSM cipher, WCDMA cipher) handle cryptographic computations for the UMTS and GSM radio access network. The general crypto accelerator

- boosts the performance of critical algorithms, such as SHA-1, MD5, AES and DES;
- provides DES encryption/decryption to and from the block with a unique, chip-independent key; and
- exponentiates large numbers (to support RSA, DH, DSS operations) with shielded private key computation.

An analog random number generator is directly connected to the cryptographic accelerator, supplying it with statistically and cryptographically sound pseudo random sequences.

Secure boot ROM is used on the access and application side to guarantee the integrity of software at startup. This includes code on all volatile and non-volatile storage. The secure ROM functions can also be used to verify integrity at run-time.

The subscriber identity module (UICC or SIM card) provides protected storage and a secure execution environment for user network authentication services and key network management tasks.[2]

Debug interfaces connect to the application and access CPUs. If left unprotected, these interfaces could be used as a backdoor to all platform functionality. The platform provides strong protection for the debug interfaces through particularly secure platform configurations when the debug capabilities have been enabled. These interfaces are always disabled in commercial product configurations.

Figure 2, which is a sketch of the software security architecture, shows the main software modules associated with security and their relationship to the hardware blocks. Note: Only high-level security functions have been depicted; low-level cryptographic functions related to radio access are not shown. The figure reflects the software configuration during normal operation. Additional software can be loaded into the phone to upgrade and configure software (not shown).

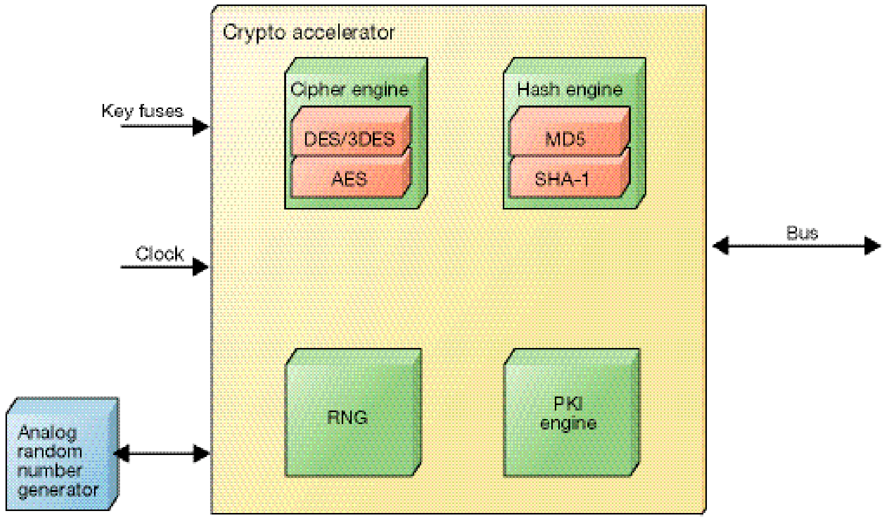


Figure 3
Schematic view of the crypto accelerator.

An IP module contains the IP stack, which includes the secure IP (IPsec) protocol.[3]

A SIM module handles the UMTS integrated circuit card (UICC) data and services, and supports applications on the UICC, for instance, SIM, USIM[2], the SIM application toolkit (SAT), and the WAP identity module (WIM).

An access core platform security module provides the SIM lock, IMEI integrity, and software reprogramming protection at runtime. It also protects data and provides protection during debugging.

An access cryptographic module provides low-level security functionality for platform modules on the access side – for example, the IP module. Note: The reader should not confuse this functionality with low-level network encryption functions, which are handled by dedicated hardware blocks.

An application core platform security module contains mechanisms that protect against software and data reprogramming, for example, secure software upgrading via over-the-air upgrades of firmware.

An application cryptographic module handles cryptographic services on the application side.

A DRM module parses protected DRM files. It also validates the rights associated with protected content, manages decryption and DRM keys, and enforces file usage rights.

A security protocol module handles high-layer security protocols and their mechanisms, such as the extensible authentication protocol (EAP) or transport layer security (TLS).[4-5]

Cryptographic engines

Security in a mobile platform relies heavily on cryptographic techniques. Cryptographic algorithms can be implemented in hardware or software. Hardware implementations often improve performance and better

TABLE 1, ALGORITHMS IMPLEMENTED IN HARDWARE.

UMTS, f8 using Kasumi
UMTS, f9 using Kasumi
GSM, A5/1, A5/2 and A5/3
GPRS, GEA1, GEA2, GEA3 and GEA4
SHA-1 (two different hardware realizations)
MD5
Platform-specific MAC engine
DES and 3DES
AES
RSA and DH

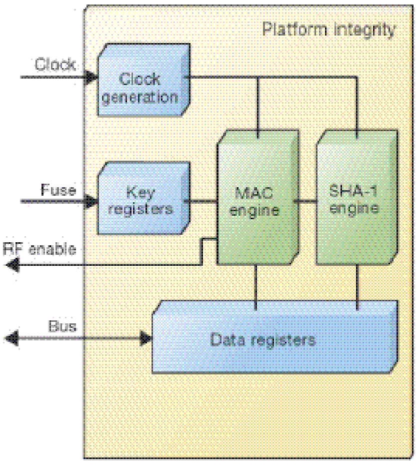


Figure 4
Schematic view of the platform integrity block.

protect sensitive information, such as keys. But they also cost more and limit flexibility. Ericsson's mobile platforms contain a balanced mixture of hardware- and software-based cryptographic algorithm support, giving good performance and security at a reasonable cost.

Hardware cryptographic modules

To offer high-level security and cryptographic performance, the platforms protect and accelerate computations of essential cryptographic operations. Numerous algorithms have been implemented in hardware (Table 1).

Figure 1 shows how the UMTS, GSM and GPRS hardware blocks are connected to the platform baseband. These hardware blocks compute the algorithms that are used to protect the cellular air interface.

The cryptographic accelerator block is dedicated to general cryptographic operations (Figure 3). It has a fairly large set of cryptographic algorithms, supporting shielded symmetric key and asymmetric public-key algorithms. This facilitates secure private key operation with the hardware module. The main security advantage of this design is that the private key can be encrypted using a unique, chip-independent key and the crypto accelerator cipher engine. The private key is thus never exposed outside the accelerator block. The crypto block also makes use of hardware acceleration to compute demanding algorithms, such as 3DES and SHA-1.

The platform also has a dedicated platform integrity hardware block that contains both a SHA-1 engine and a dedicated MAC engine (Figure 4). The block verifies platform code and data integrity. The hardware radio functionality in the platform relies on correct MAC verification of security-critical platform data.

Software cryptographic modules

The platform's access and application cryptographic modules provide applications (DRM, browser security, IPSec, execution environments, DVB-H) with low-level crypto support – for example, cryptographic algorithm support, WIM support, and certificate handling.[6] Most of the functionality is located in the application cryptographic module because most clients of this functionality execute on the application CPU. Figure 5 shows how the supported functionality is partitioned in the application and access cryptographic modules.

Application cryptographic module

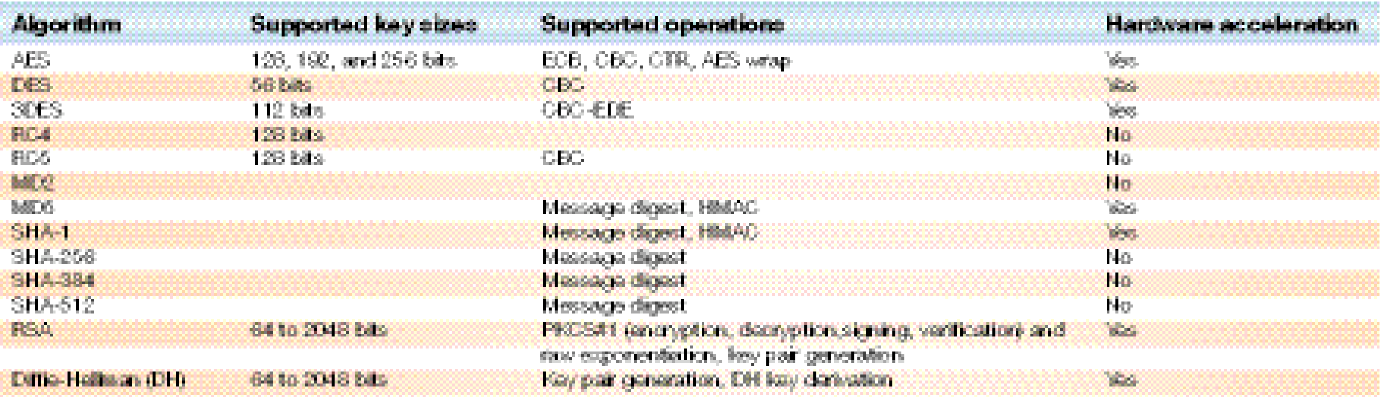
The application cryptographic module provides cryptographic algorithm and WIM support, stores cryptographic objects, and handles certificates. A PKCS#11 interface is used to access cryptographic algorithms and stored cryptographic data.[7] The idea is that the same interface can be used regardless of whether data or algorithms reside in software, hardware, or on a WIM application on a SIM or USIM card.

PKCS#11 is an application program interface (API) to devices that hold cryptographic information and perform cryptographic functions. Each device, called a cryptographic token, can be implemented entirely in software, or with hardware support. A PKCS#11 implementation consists of one or more tokens and a thin, common interface to the tokens. This interface, called Cryptoki, initializes PKCS#11 functionality and manages communication with the cryptographic tokens. The tokens implement the functionality and are responsible for cryptographic data storage. The PKCS#11 functionality is split into three tokens: platform token, SIMWIM token, and DRM token.

The platform token, which is always present, contains every algorithm implemented in the phone. It also contains built-in WIM functionality that supports TLS/SSL client and server authentication.[8-9] In addition, it contains

- symmetric and asymmetric encryption and decryption;
- hash calculation, MAC calculation

TABLE 2. SUPPORTED ALGORITHMS, KEY SIZES, AND OPERATIONS.



Algorithm	Supported key sizes	Supported operations	Hardware acceleration
AES	128, 192, and 256 bits	ECB, CBC, CTR, AES wrap	Yes
DES	56 bits	CBC	Yes
3DES	112 bits	CBC-EDE	Yes
RC4	128 bits		No
RC5	128 bits	CBC	No
MD2			No
MD5		Message digest, HMAC	Yes
SHA-1		Message digest, HMAC	Yes
SHA-256		Message digest	No
SHA-384		Message digest	No
SHA-512		Message digest	No
RSA	64 to 2048 bits	PKCS#1 (encryption, decryption,signing, verification) and raw exponentiation, key pair generation	Yes
Diffie-Hellman (DH)	64 to 2048 bits	Key pair generation, DH key derivation	Yes

(HMAC), digital signature generation and verification[1,10];
- random number generation;
- built-in WIM functionality: TLS, and SSL key exchange (pre-master secret generation, master key derivation, and pseudo random function), and storage of session data and master secrets;
- protected storage of certificates and keys in flash memory; and
- asymmetric key pair generation (RSA, DH); among other things, the RSA private-public key pair can be used for creating signatures for authentication and non-repudiation.

The platform token supports numerous standard algorithms (AES, DES/3DES, RC4, RC5, MD2, MD5, SHA-1, SHA-256, SHA-384, SHA-512, RSA, and DH). As an option, many of these (AES, DES/3DES, MD5, SHA-1, RSA, and DH) can be accelerated in the crypto accelerator hardware block to boost performance and increase security (via shielded key operations). The access cryptographic module contains drivers for the crypto accelerator hardware. If manufacturers waive the hardware-acceleration option, the algorithms are executed in software on the application CPU. The SHA-1 algorithm, however, may always be accelerated in hardware via the integrity block. The other algorithms are always executed entirely in software.

The SIMWIM token provides access to cryptographic algorithms, cryptographic objects of a WIM application on the SIM/USIM card, and cryptographic objects of a WAP provisioning application on the SIM/USIM card.[11] If supported by the SIM/USIM card, the SIMWIM token
- supports PKCS#1 encryption, decryption, signing and verification;
- supports random number generation;
- supports TLS key exchange (pre-master secret generation, master key derivation, and pseudo-random function) and stores session data and master secrets; and
- stores certificates and RSA private-public keys.

The SIM software module on the access side is responsible for communication with the WIM application on the SIM/USIM card. The SIMWIM token uses this module to access the WIM.

The DRM token helps realize OMA DRM support. Access to this token is restricted to the DRM module. The DRM token contains functionality that is specific to DRM, such as RSA-KEM[12], which involves the DRM private key. The crypto accelerator block uses a unique chip-independent key to encrypt and protect the integrity of the DRM private key when stored in flash memory. The DRM private key is always highly protected. When used for decryption and signing operations, for example, it is decrypted and held inside the hardware block; it is thus never exposed outside the block. The private key is installed at the time of production using a special RAM loader.

Table 2 lists supported algorithms, key sizes, and operations.

Certificate management

Digital certificates are used to identify external services or the mobile phone itself. A certificate typically consists of
- an identity value, which identifies the entity that uses the certificate;
- a public key;
- an issuer value, which identifies the certificate authority;
- other information about validity and time; and
- a digital signature.

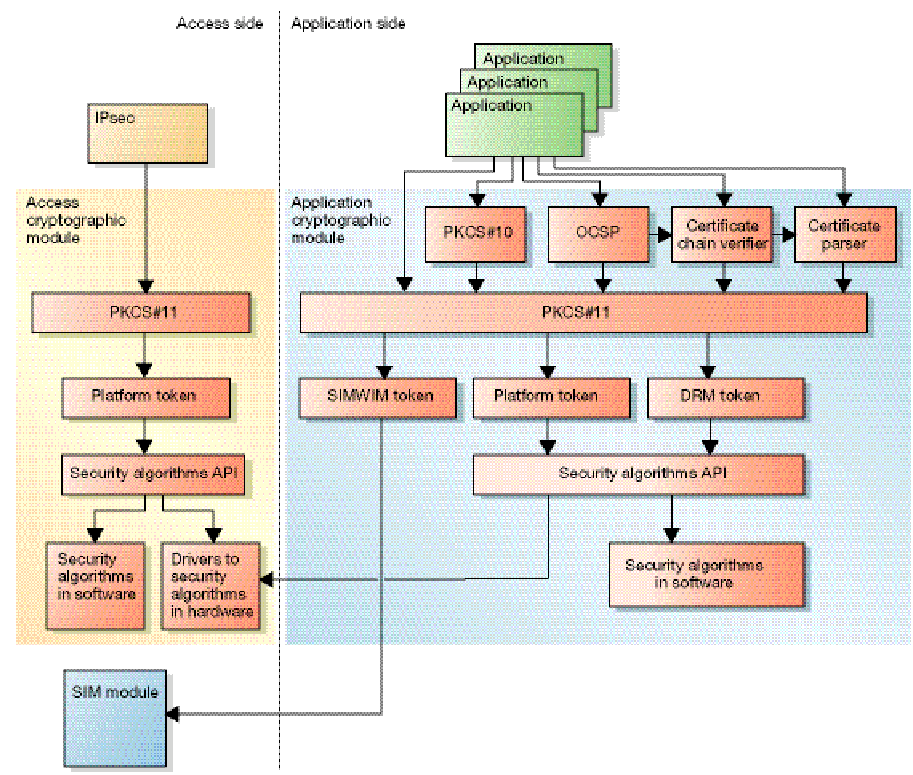
The certificate thus provides a cryptographic binding between an identity and a public key. This is very useful for authentication and key exchange purposes. Certificates can be chained such that the public key in one certificate is used to verify the signature of another certificate (delegated authorization).

The certificate-management function of the application cryptographic module parses certificates, verifies certificate chains, stores certificates, and generates PKCS#10 certificate requests.

Certificates are stored via the PKCS#11 interface either in flash memory via the platform token or on a WIM/WAP provisioning application on the SIM/USIM card.

The certificate parser is used internally to verify certificate chains. Applica-

Figure 5
Block diagram of the software cryptographic modules



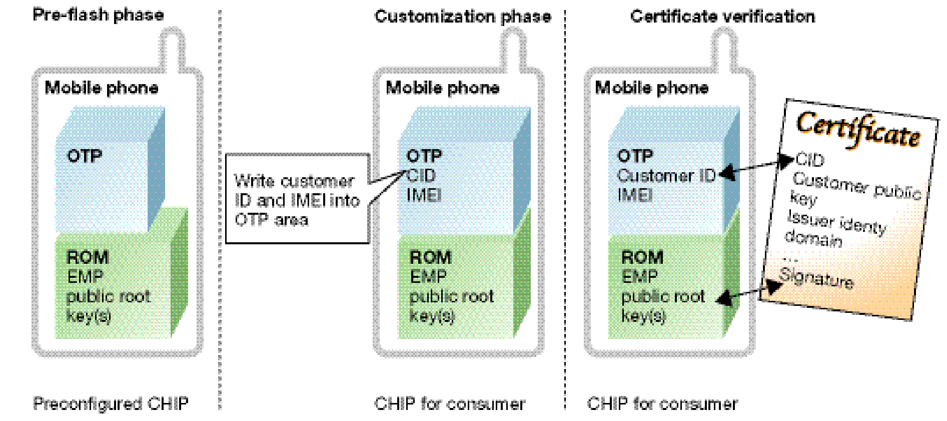


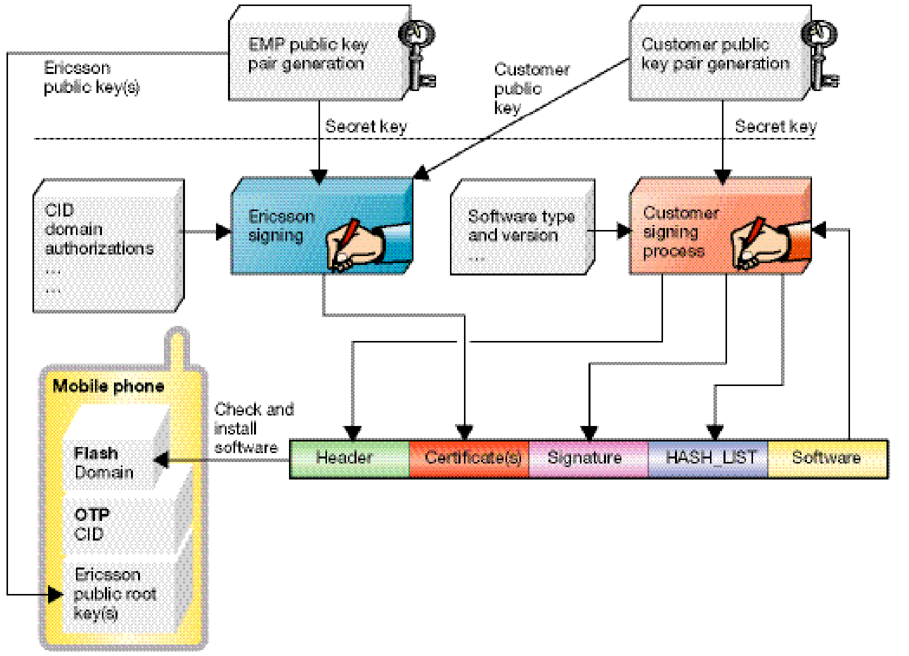
Figure 6
Public key validation and customization.

tions also use it to display parsed information to the user. The parser supports the parsing of X.509 v3 certificates according to the OMA certificate profile.[13] In compliance with this profile, the verifier verifies certificate chains consisting of X.509 certificates. If the certificate chain does not include the root certificate, the verifier can be instructed to look for it and any other missing certificates in the chain on any of the PKCS#11 tokens of the application cryptographic module. The application cryptographic module also supports the generation of online certificate status protocol (OCSP) requests and the validation of OCSP responses that can be used by an application to check the revocation status of certificates in a chain. The platform supports the OMA OCSP mobile profile.[14]

The application cryptographic module also generates PKCS#10 certification requests,[15] which can be sent to a certificate authority to obtain certificates for a public key that is

- generated on the device, or
- available on the WIM on the SIM/USIM card.

Figure 7
Signing principles and formats.



Access cryptographic module

The access cryptographic module provides cryptographic algorithm support for the access side. IPsec, for example, requires support for AES and DES/3DES in cipher block chaining (CBC) mode, and HMAC with SHA-1 and MD5. The AES, DES/3DES, MD5 and SHA-1 algorithms can be accelerated in hardware to boost performance. The access cryptographic module contains the necessary hardware drivers. If the manufacturer waives this option, the algorithms are executed in software.

Platform software integrity and secure boot

To ensure that the platform behaves correctly, one must guarantee the integrity of core platform software and critical data. This holds true for software that controls the radio, and security software that controls SIMLock, DRM and general security services, such as TLS and IPsec protocols.

The integrity of the platform software is protected by solely allowing digitally signed software to be executed on the platform. This applies to platform software and applications and to RAM loaders; that is, software loaded into RAM over an external interface. RAM loaders are used for customization, verification, and for updating phone software in flash memory.

The software or loader is validated when it is loaded onto the platform and before execution. All access software and core parts of the application software are thus checked each time the platform starts. A first-time check is always performed using digital signatures and certificates when the software is loaded from an external source or when the platform is started after software has been flashed into the phone. To speed up validation, MACs are used to recheck software that was approved during the first-time check. The MACs check software before execution and when a mobile phone is started. They are also used for verifying certain data areas (system settings). MAC verification uses a symmetric encryption key and is thus much faster than the first-time check. The symmetric key is a unique (chip-dependent) code generated in the integrity hardware block.

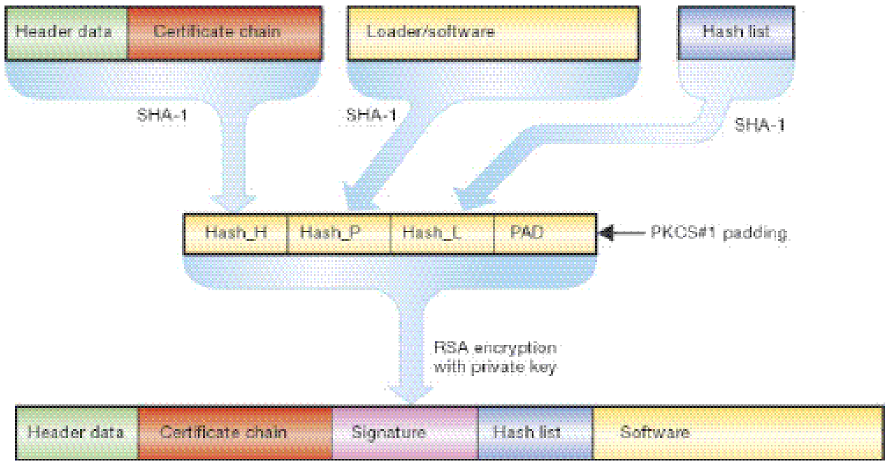


Figure 8
Replacing the signature with a MAC.

Software signing, keys and certificates

The signing of software and RAM loaders is based on a PKI solution for which Ericsson serves as the root certificate authority. Ericsson has generated an RSA root private-public RSA key pair. The public key is stored in the access boot ROM; the private key is kept secret. Ericsson's mobile platforms have three root keys stored in ROM: two 1024-bit keys and one 2048-bit key. Each root key pair is unique for each digital baseband controller. Electrical fuses enable Ericsson to revoke a key pair if the private key has been compromised. All three key pairs are activated from the start, but only one is used at a time.

Protected software must be signed with a private key. The digital signature format is compliant with the RSA PKCS#1 standard.[11] Regardless of who issues the code, the same certification principle and policy apply. A valid RSA signature and associated chain of X.509 certificates means that the software has been approved by Ericsson, by customers of Ericsson, or by trusted third parties, and has not been modified before being downloaded. The platform will only accept software with signatures matching its active public root key. Figure 6 shows the general customer key verification principle.

Ericsson issues unique certificates to its customers (mobile phone manufacturers) who build phones on its mobile platforms. Ericsson's customers may sign software and RAM loaders that are to be accepted by the platform. A certificate for a particular customer contains an RSA public key signed by Ericsson using the Ericsson private key. Customers use their private keys to sign software and RAM loaders for the platform. The RSA key pairs are either generated by Ericsson or a customer. When customers generate key pairs, they must request PKCS#10 certification from Ericsson.[15] This process is surrounded by careful procedures to prevent wrongful issuing of certificates. The certificate format, based on the X.509 v3 standard, uses a defined extension that includes Ericsson-specific information for controlling the rights of certified keys.[13]

Ericsson certifies keys for its customers, who in turn, are allowed to certify keys for their customers or third-party developers. To prevent loaders and software from one customer from being downloaded into other mobile phones, each customer is assigned a unique customer ID (CID). Several CIDs can be assigned to any given customer but no two customers will have the same CID. The CID is written into the customer certificate. It is also written into a one-time programmable (OTP) area during production. When software and RAM loaders are verified, a check is made to ensure that the CID of the customer certificate matches that of the OTP area. The software version is also checked. In keeping with GSMA and EICTA anti-rollback protection requirements, only software with a larger version number than that of current software will be accepted.[16]

The customer digitally signs a software header, the software itself, and optionally, a hash list. If present, the hash list is calculated from hashes of software blocks of the software to be installed. Software blocks may be verified each time the software is loaded or flashed into memory. Figure 7 shows how complete, protected digital signature software packages are created.

The digital signature is replaced with a MAC when software is flashed into memory or during the first boot after it has been flashed into memory (Figure 8). The platform integrity hardware block calculates and checks MACs. It also performs all hash computations (SHA-1).

A mobile equipment domain has been introduced to keep software and RAM loaders intended for use in R&D and at factories from being used in commercial products. The domain is indicated in the customer certificate. The integrity of the domain is protected and stored in flash memory. The platform integrity block protects integrity. The following domains are used:

- Factory – loaders and software intended for use in the factory.
- R&D – loaders and software intended for use in R&D.
- Product – loaders and software for commercial products.
- Service – this domain may only be used by the boot ROM code when the current phone domain cannot be read. This domain makes it possible to restore the phone domain using a special loader signed for the domain. This procedure requires special authorization.

A customer certificate may contain or be issued for several domains. Loaders and software signed by a customer are signed for the domains contained in the customer certificate. Only those loaders and the software associated with the domain to which the mobile phone belongs will pass verification and be accepted by the phone. Ericsson issues several certificates to its customers for use in factory, R&D, or commercial products. The exact list of domains in each certificate depends on the needs of the customer (manufacturer).

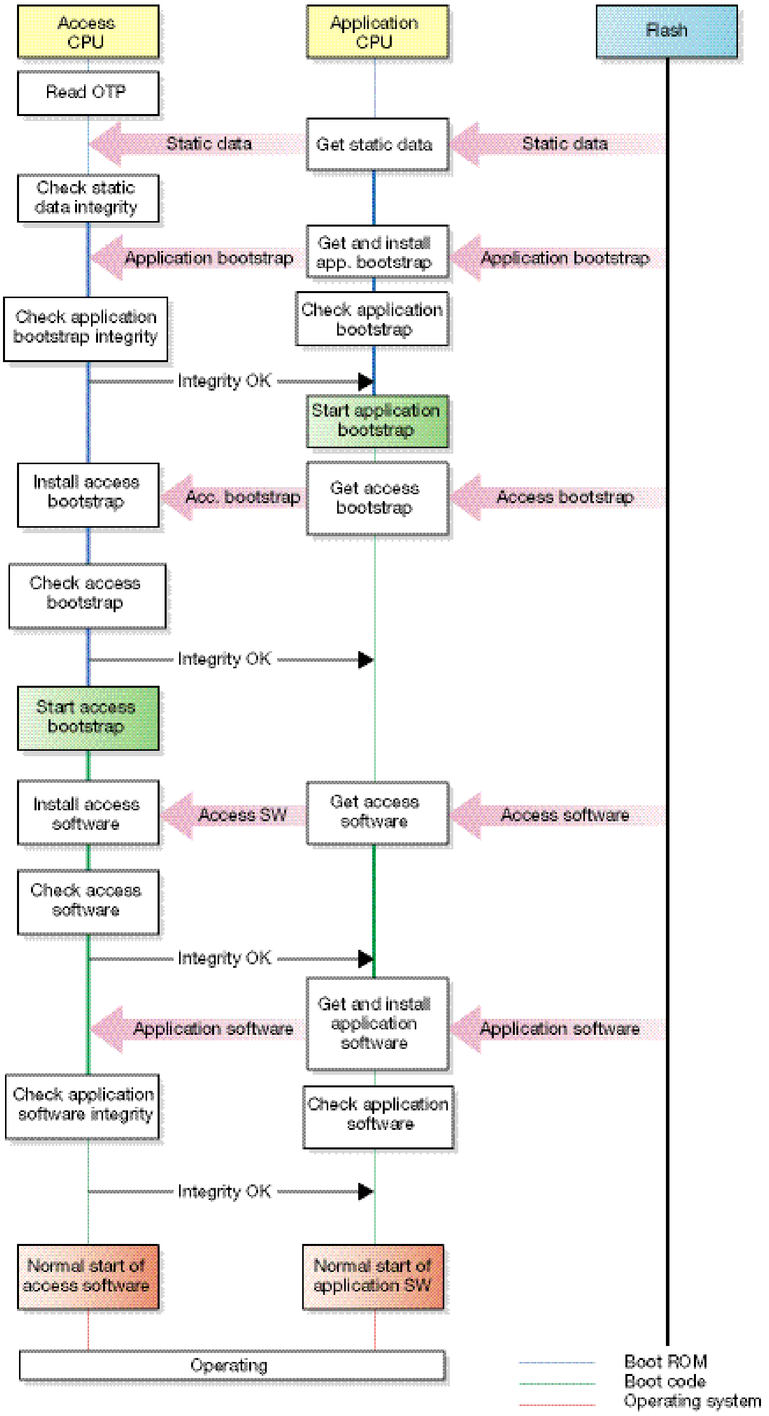
During production, the phone domain of the mobile phone is set to *factory*. Therefore, only factory loaders and software are accepted. At the end of the factory process, the domain is changed to either R&D or product.

Secure boot

The platform can be started in two main modes of operation: service mode and normal mode. At startup in service mode, the

Figure 9
Normal mode boot sequence. The flash used to store access and application software and bootstrap code is connected on the application side.

1. The access ROM reads the one-time programmable (OTP) contents. The OTP configuration only allows software intended for the particular platform to be executed. The customer ID (CID), for example, is part of the OTP. And only software signed for the correct CID is accepted.
2. The phone domain is read from flash memory. The phone domain is stored as static data. The integrity hardware block verifies the integrity of this data. The static data is also compared against the OTP configurations.
3. The application and access bootstrap codes are read from flash memory. The bootstrap codes are stored with a header. They are also protected with a signature or MAC. The integrity block assists the ROM code on the access side by checking the integrity of these two blocks.
4. Complete access and application software images are loaded and verified in exactly the same way as the bootstrap codes. This process ensures that the integrity of all code is checked before it is executed.



platform loads one or more RAM loaders from a fixed connected local interface, such as a serial port or universal serial bus (USB). This mode is used to customize and flash the phone during production. At startup in normal mode, the phone is booted from flash memory. In either case, the software must be verified before it can be executed. The verification principles are similar for both modes of operation. Figure 9 shows the startup sequence for normal mode.

Secure communication

Users want to be able to use their mobile phones to communicate regardless of location or time. And when they do so, they take it for granted that sensitive voice and data messages will not be revealed to hostile third parties. The kind of protection that is applied varies according to the application. For some applications, it is sufficient to protect the wireless communication channel. For other applications, users and content providers require end-to-end protection of information. We distinguish between network access security, local interface security, broadcast security and packet data security. Ordinarily, packet data protection mechanisms solely apply to end-to-end scenarios. The low-level cryptographic support needed for these services (apart from the Bluetooth and WLAN radio interface) are realized using the hardware and software cryptographic modules.

Network access security

Network access security is about giving users secure access to 2G and 3G services; in particular, protecting them against attacks on the radio access link. This entails network and terminal authentication and preserving confidentiality and the integrity of certain control data, user traffic, user identity, and location data. Apart from emergency calls, a valid SIM or USIM is required to access any 2G or 3G service. Theft prevention through the unique terminal identity, IMEI, is also an element of network access security. The IMEI can be used to bar a terminal from accessing 2G and 3G services. Data protection features keep the IMEI secure.

The mobile platform supports all mandatory 3GPP authentication, key exchange, confidentiality, and integrity-protection mechanisms. Hardware support facilitates computation-intensive encryption/decryption algorithms.[3]

A mobility manager software module manages keys and authenticates users with the help of the UICC, which is controlled by the SIM software module. Note: Because this software module mainly handles non-security-related tasks it has not been included in Figure 2.

Depending on the kind of air interface, different software is used to control network cryptographic tasks. The control software is part of the communication software (not shown in Figure 2). In WCDMA, the radio link controller and physical layer software control encryption. In GSM, the physical layer handles encryption; in GPRS, the logical link controller.

Local interface security

In recent years, there has been a growing number of security threats from other radio interfaces. Several Bluetooth products, for example, have shipped with bad security policy settings or faulty access control implementations. Some mobile phones on the market allow unauthenticated devices to access sensitive phone services, such as phone book, calender, business card and mobile phone identity number.[17]

One can considerably reduce the risk of security threats by controlling access and by switching on Bluetooth security. The Ericsson Bluetooth stack supports access control and encryption at the application level. In the end, however, it is up to application developers to decide on the security policy for the interface. The Bluetooth encryption functions are performed on the Bluetooth hardware module outside the platform digital baseband.

Ericsson's mobile platforms also support WLAN access according to the IEEE 802.11 standard.[18] WLAN communcation security is a rather complex area in which a number of security aspects need to be considered. Earlier versions of the standard had several security weaknesses, but WLAN link security has been much improved through the introduction of IEEE 802.11i link-protection mechanisms and IEEE 802.1X port-based access control.[19-20] The WLAN hardware module supports the 802.11i mechanisms. Likewise, the platform data communcation stack supports the IEEE 802.1X framework. The 1X standard allows for a large set of authentication and key exchange options based on the extensible authentication protocol (EAP).[4] Seen in terms of mobile phone communication, some of the most useful authentication mechanisms are EAP-SIM and EAP-AKA, which are based on GSM and UMTS authentication methods.[21-22] The platform supports these mechanisms as part of its security protocol services.

Broadcast traffic protector

Broadcast services are emerging in mobile networks. Two major broadcast standards, multimedia broadcast/multicast service (MBMS) and digital video broadcast – handheld (DVB-H), are finding their way to mobile handsets.[23-24] These two technologies call for separate security solutions.

In the context of broadcasted content, one usually distinguishes between traffic protection and content protection. Content-protection mechanisms (also called digital rights management, DRM) control how broadcasted content is used. Traffic protection, on the other hand, is about controlling access to the broadcast service. At present, the standardization of traffic protection is incomplete. What is clear, however, is that basic security mechanisms cannot easily be combined in DVB-H and MBMS. The mobile platform currently supports the DVB-H broadcast standard. The security options in DVB-H protect traffic via IPsec, secure real-time transport protocol (SRTP) and ISMAcryp according to the IP device control (IPDC) standard.[25-28] Among these, IPsec and ISMAcryp are part of the platform's security protocol services. The key material for this protection (encryption key and integrity key) must be given to the platform, for example, at the application level on top of the platform. Once the MBMS security standard and OMA-based broadcast security mechanisms have been standardized, the platform will also support them.[29]

Packet data protector

Two major protection protocols are used to protect TCP/IP connections: secure IP (IPsec) and transport layer security (TLS).[26, 8]

TLS and its predecessor, secure sockets layer (SSL), offer secure browsing services. The TLS and SSL protocol suites contain authentication and key exchange mechanisms and transport security. TLS is also very flexible when it comes to client authentication options. Server authentication is based on (X.509) certificates. The optional client authentication is also based on certificates. Platform support for TLS and SSL is provided on the application system as part of the protocol services.

The platform also offers IPsec as part of the IP stack on the access subsystem. In contrast to TLS, the IPsec protocol is solely an IP packet-protection mechanism. Therefore, it must be complemented with key management support on another layer. For virtual private network (VPN) applications, the obvious choice is internet key exchange (IKEv1 or IKEv2).[5,30] IKEv2 will be added to the platform protocol suite in 2007.

The TLS and IPsec protocols require low-level support of several cryptographic algorithms. The cryptographic modules, which implement the algorithms in software, hardware, or both, provide this support.

Packet data services are important and must be protected in a WLAN setting. In the near future, the platform will support universal mobile access (UMA) security mechanisms based on IPsec and IKEv2.[31] Futhermore, there is an apparent need for additional protection at the packet level. Ericsson thus foresees a firewall, a virus protection mechanism, or both as part of the platform security offering.

IMEI protection and SIM lock

IMEI protection

Originally intended as a unique identity to prevent non-type-approved mobile phones from connecting to GSM networks, the IMEI is today used to identify mobile phones on mobile networks, to combat the use of stolen equipment. Network operators can blacklist stolen phones to prevent them from being used in their networks. For this mechanism to be effective, users must not be able to modify a phone's IMEI value. Ericsson supports the 3GPP requirement that to resist tampering the IMEI cannot be changed after production. The Ericsson mobile platform thus solely allows manufacturers to store the IMEI in an OTP memory area (Figure 1) at the time of production.[32] Once the IMEI has been written, the memory area is locked and cannot be modified or rewritten. Furthermore, the integrity of platform software that is responsible for reading the IMEI value and sending it to the network is protected and verified at every startup. This implementation complies with the GSMA and EICTA security principles related to handset theft.[16]

SIM lock

The SIM lock feature uses information stored in mobile phones to list the number of SIM/USIM cards with which the equipment will operate. The function can also be used to lock mobile phones to a range of SIM/USIM cards. This range can vary from one SIM/USIM card to any SIM/USIM card that belongs to one particular or several cooperating networks.

Ericsson's mobile platforms support the 3GPP standard for locking mobile phones to a SIM/USIM.[33] Several different locks are available. In addition, an Ericsson-specific lock extends locking functionality to fulfill operator requirements (Box B).

The platform thus offers flexible customization of SIM locks for the manufacturer during production. The customized SIM lock settings are stored in flash memory. The platform integrity hardware block prevents SIM lock cloning. Each time the phone is started, the platform software checks that the inserted SIM/USIM card matches the SIM lock settings. The integrity of the SIM lock software is also checked at every start-up.

The SIM locks can be unlocked via dedicated platform interfaces or over the air (OTA), via SMS.[33]

Ericsson's mobile platforms also support an anti-theft feature that enables users to stipulate that a personal identification number (separate from the regular PIN protecting the SIM/USIM card) must be entered each time the phone is powered on. This security PIN prevents stolen phones from being used even after the SIM/USIM card has been replaced.

BOX B, SUPPORTED SIM LOCK TYPES

- Network lock – locks the phone to a specific network defined by the mobile country code and the mobile network code.
- Network subset lock – locks the phone to a specific subset of the network.
- Service provider lock – uses the group identifier on the SIM/USIM to lock the phone to a specific service provider.
- Corporate lock – uses the group identifier to lock the phone to SIM/USIM cards that belong to a specific company.
- Cooperative network list – locks the phone to a specific group of networks. This lock can be a combination of any or all the locks listed above.
- SIM lock – locks the phone to a specific SIM/USIM card.
- Extended SIM lock – an Ericsson-specific lock that adds greater flexibility for handling special operator requirements.

The Ericsson mobile platforms also support auto-locking to a SIM/USIM card; that is, the phone locks to the first SIM/USIM card inserted into it. The lock type can be one or more of the lock types described above.

Application security and DRM

Application security

The platform allows Java MIDlets and dynamically linked native applications to be downloaded and executed at runtime. To be accepted by the platform, applications or MIDlets must first be digitally signed via a scheme that is based on RSA signatures with X.509 certificates. When a native application is installed, the platform installer verifies the signature of the application and the certificate chain that is downloaded with it. The application cryptographic module handles certificate parsing and verification.

The platform supports the downloading and verification of Java MIDlets according to Java MIDP 2.0.[34] The MIDlets are either trusted or untrusted. Untrusted MIDlets may operate in a "sandbox" without access to sensitive platform APIs. Trusted MIDlets must be digitally signed and verified before they can be granted access to more sensitive platform APIs. The platform APIs they may access is determined by the Java MIDP security domain to which they belong. The platform supports operator, manufacturer, and third-party domains.[34] The signing of Java MIDlets is based on RSA signatures with X.509 certificate support. The Java MIDlets are verified in the same way as native applications. In addition, the platform checks the security domain to which the Java MIDP belongs. This is determined by the root certificate in the certificate chain associated with the MIDlet.

The platform also supports the Java security-and-trust-services API (SATSA, JSR 177)[35], which defines a collection of APIs that provides the following security and trust services:

- Secure storage, to protect sensitive data, such as private user keys, public key (root) certificates, personal information, and so on.
- Cryptographic operations, to support payment protocols, data integrity, and data confidentiality.
- A secure execution environment, for deploying custom security features. Java MIDlets rely on these features to handle many value-added services, such as user identification and authentication, banking, payment, loyalty applications, and so on.

The above features can be obtained via a smart card. They can also be implemented in software.

The APIs are grouped into four categories: SATSA-APDU, SATSA-JCRMI, SATSA-PKI and SATSA-CRYPTO.

The SATSA-APDU and SATSA-JCRMI categories handle communication with a smart card. The implementation of SATSA-APDU allows a Java MIDlet to communicate with applications on a smart card that is attached to the platform via the application protocol data unit (APDU) protocol.

SATSA-JCRMI allows a Java MIDlet to invoke a method of a Java object on a Java card attached to the platform.

SATSA-PKI gives Java MIDlets the functionality to generate digital signatures that conform to the cryptographic message syntax (CMS) format[36] and manage user certificates and public or private keys.

SATSA-CRYPTO gives Java MIDlets basic cryptographic operations, such as message digest, digital signature verification, encryption, and decryption. The cryptographic operations enable a Java MIDlet to provide secure data communication, protect data, and manage content.

The implementation of SATSA-CRYPTO and SATSA-PKI is based on basic cryptographic services, including WIM support, from the application cryptographic module.

Phone manufacturers can employ the basic security services of the application cryptographic module to write their own security applications, such as secure e-mail clients, e-commerce applications, and so on.

DRM

The platform supports DRM, which is a complete system designed to limit access to digital media content to people who have acquired a proper license to play or view it. The content provider (the party who created the content) "packages" the content according to the DRM specification and establishes one or more sets of usage rights (or rules) and associated usage costs. Consumers can buy and download content (usually encrypted) and associated rights from, say, the website of a content distributor. The DRM part of the phone makes certain that a proper license has been obtained and enforces rules for playing content.

The DRM functionality in Ericsson's mobile platforms provides a secure, efficient and robust solution for implementing DRM support in mobile devices. The platform DRM functionality is not a complete DRM client solution, however. The platform contains basic mechanisms that can be used to build complete DRM agents, but the functionality, such as that required to download DRM content, is handled by the applications (with some support from the platform). The platform currently supports OMA DRM v1.0 and v2.0 agents. In addition, a plug-in framework is available in the platform to support other DRM standards. Manufacturers may thus implement other DRM standards without having to modify the platform software.[12,37] The platform supports each of the different DRM implementations in parallel, including application-level DRM solutions implemented using the plug-in concept.

The platform DRM solution provides a uniform rendering interface for non-DRM and DRM content. Platform rendering services might also be used, for example, by DRM solutions implemented by a mobile phone manufacturer (using the plug-in concept). The platform uses a common interface for generic DRM operations. Likewise, it provides automatic DRM support for every supported media type. To protect it, decrypted DRM content is kept inside the platform domain. Tight coupling to the platform cryptographic services also ensures overall high security.

Support for OMA DRM v1.0 includes forward lock (a mechanism that prevents con-

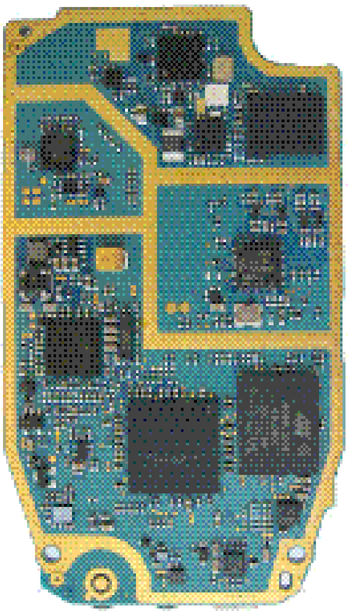


Figure 10
Reference printed circuit board (PCB) for Ericsson's U350 mobile platform.

TRADEMARKS

- Java is a trademark or registered trademark of Sun Microsystems, Inc. in the United States and other countries.
- Intel is a trademark or registered trademark of Intel Corporation or its subsidiaries in the United States and other countries.
- Microsoft is a registered trademark of Microsoft Corporation.
- Open Mobile Alliance (OMA) is a trademark of Open Mobile Alliance Ltd.

tent and messages from being transferred out of the mobile phone), combined delivery of content and rights, and separate delivery of content and rights. The platform also supports every mandatory part of OMA DRM v2.0, including the rights object acquisition protocol (ROAP). In addition, Ericsson is adding support for OMA DRM domains, progressive download, and streaming to the platform.

The DRM module, which plays a central role in the platform DRM solution, decrypts content while rendering downloaded and streaming DRM content. It also downloads (ROAP) and parses DRM files, stores and reads license files, validates rights associated with protected content, and manages DRM keys.

Conclusion

Ericsson's mobile platforms contain a variety of security mechanisms and an advanced model for issuing and checking the integrity of software. Manufacturers of mobile phones can thus build secure applications and a secure process for issuing platform software. Apart from communication security and basic platform integrity control, mobile phone developers can decide how and when they want to use these mechanisms. Therefore, additional security mechanisms might be needed.

The mobile platforms do not use an open software environment that permits end users to install new software modules or applications. This limits the need for further software-protection mechanisms. In the future, however, we might see a more PC-like situation, where new native software, such as non-Java software, can easily be installed and executed. In that case, non-trusted software will have to be isolated from trusted software. Advanced operating systems isolate non-trusted software in such a way that it is never allowed to interfere with the execution of trusted software. In a large and flexible OS, however, isolation is difficult to achieve. Therefore, other means, such as hardware mechanisms or more stringent security requirements (in the OS) will have to be employed.

REFERENCES

1. HMAC: Keyed-Hashing for Message Authentication, IETF RFC 2104, http://www.ietf.org/rfc/rfc2104.txt
2. 3GPP TS 31.101: 3rd Generation Partnership Project (3GPP); Technical Specification Grou Terminals; UICC-terminal interface; Physical and logical characteristics, http://www.3gpp.org
3. V. Niemi and K. Nyberg, UMTS Security, Wiley, 2003
4. Extensible Authentication Protocol, IETF RFC 3748, http://www.ietf.org/rfc/rfc3748.txt
5. The Internet Key Exchange (IKE), IETF RFC 2409, http://www.ietf.org/rfc/rfc2409.txt
6. OMA, Wireless Identity module v1.1, http://www.openmobilealliance.org
7. RSA Laboratories, PKCS#11 v2.20, http://www.rsasecurity.com/rsalabs/pkcs
8. Transport Layer Security v1.0, IETF RFC 2246, http://www.ietf.org/rfc/rfc2246.txt
9. Secure Socket Layer v3.0, http://wp.netscape.com/eng/ssl3/draft302.txt
10. RSA Laboratories, PKCS#1 v2.1, http://www.rsasecurity.com/rsalabs/pkcs
11. OMA, Provisioning Smart Card Specification v1.1, http://www.openmobilealliance.org
12. OMA, DRM Specification v 2.0, http://www.openmobilealliance.org
13. OMA, Certificate and CRL Profiles v1.1, http://www.openmobilealliance.org
14. OMA, Online Certificate Status Protocol Mobile Profile v1.0, http://www.openmobilealliance.org
15. RSA Laboratories, PKCS#10 v1.7, http://www.rsasecurity.com/rsalabs/pkcs
16. GSMA/EICTA, Security Principles Related to Handset Theft v3.0.0
17. C. Gehrmann, J. Persson and B. Smeets, Bluetooth Security, Artech House, 2004
18. IEEE 802.11 1999 Edition,. http://www.ieee.org
19. IEEE 802.11i -2004, http://www.ieee.org
20. IEEE 802.1X-2004, http://www.ieee.org
21. Extensible Authentication Protocol Method for Global System for Mobile Communications (GSM) Subscriber Identity Modules (EAP-SIM), IETF RFC 4186, http://www.ietf.org/rfc/rfc4186.txt
22. Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA), IETF RFC 4187, http://www.ietf.org/rfc/rfc4187.txt
23. ETSI EN 300 744 Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for digital terrestrial television, http://www.etsi.org
24. 3GPP TS 223.246: Multimedia Broadcast/Multicast Service, Stage 1, http://www.3gpp.org
25. The Secure Real-time Transport Protocol (SRTP), IETF RFC 3711, http://www.ietf.org/rfc/rfc3711.txt
26. Security Architecture for the Internet Protocol, IETF RFC 2401, http://www.ietf.org/rfc/rfc2401.txt
27. Internet Streaming Media Alliance Encryption and Authentication Version 1.1, http://www.isma.tv/technology/ISMACryp1.1.html
28. IPDC Services Purchase and Protection Specification, http://www.etsi.org
29. 3GPP TS 33.246: Security of Multimedia Broadcast/Multicast Service, http://www.3gpp.org
30. Internet Key Exchange (IKEv2) Protocol, IETF RFC 4306, http://www.ietf.org/rfc/rfc4306.txt
31. Unlicensed Mobile Access (UMA); Protocols (Stage 3), http://www.umatechnology.org/specifications/index.htm
32. 3GPP, International Mobile station Equipment Identities (IMEI), TS 22.016 v5.0.0
33. 3GPP, Personalization of Mobile Equipment (ME); Mobile Functionality Specification, TS 22.022 v5.0.0
34. Java Micro Edition (JME), Mobile Information Device Profile v 2.0
35. Java Micro Edition (JME), Security and Trust Services API (SATSA) v 1.0
36. Cryptographic Message Syntax, IETF RFC 2630, http://www.ietf.org/rfc/rfc2630.txt
37. OMA, DRM Specification v 1.0, http://www.openmobilealliance.org