Integrated Technologies for Communication Security on Mobile Devices

Alexandre Melo Braga
Fundação CPqD – Centro de Pesquisa e Desenvolvimento em Telecomunicações
Campinas, Brazil
ambraga@cpqd.com.br

***Abstract*—Communication security, information disclosure and vulnerability exploitation are always a concern and a challenge, especially these days, when everything goes mobile. This short paper describes preliminary results on the construction of an integrated framework of applications for secure communication via mobile devices. Particularly, the paper discusses major design decisions on three topics, namely, framework architecture, mobile applications for communication security, and cryptographic service providers.**

Keywords—mobile security; commuication security; SMS security; Instant Message security

I. INTRODUCTION

A recently reported incident on Android [5][9] brought to light, once again, the issue of blindly relying on a single vendor's defenses. Also, a recent disclosure of top-secret NSA (U.S. National Security Agency) documents to The Guardian [19] exposed U.S. government's surveillance activities on phone and Internet communication. These two incidents suggest more than ever that there is a need, on mobile devices, for security solutions suitable to the regular people and that go beyond the ordinary software for antivirus and e-mail security.

On the other hand, NSA recently started to encourage the use of Commercial-Off-The-Shelf (COTS) mobile devices, in particular smartphones running Android, for communication of classified information [14], and fostering a worldwide improvement of mobile security products.

This short paper presents preliminary results of a Brazilian project that fosters security technologies on mobile environments. The main motivation for the project is to offer technological solutions for the growing demand for security in mobile environments. This demand was caused not only by the significant increase in the use of smart mobile devices (smartphones and tablets), but also by the growing interest of cyber criminals in mobile environments. It is important to note the existence of malicious software specific to the Brazilian context, according to the Computer Security Incident Response Team (CSIRT) for the Brazilian Internet [7]. Furthermore, according to the Brazilian Telecommunications Agency, in 2010 the number of mobile accesses has exceeded the number of people in Brazil [2], and a large portion of it is for data.

The remainder of this text is organized according to the results obtained so far. Section II describes the overall architecture of the proposed framework. Section III outlines related work. Section IV details the set of applications for

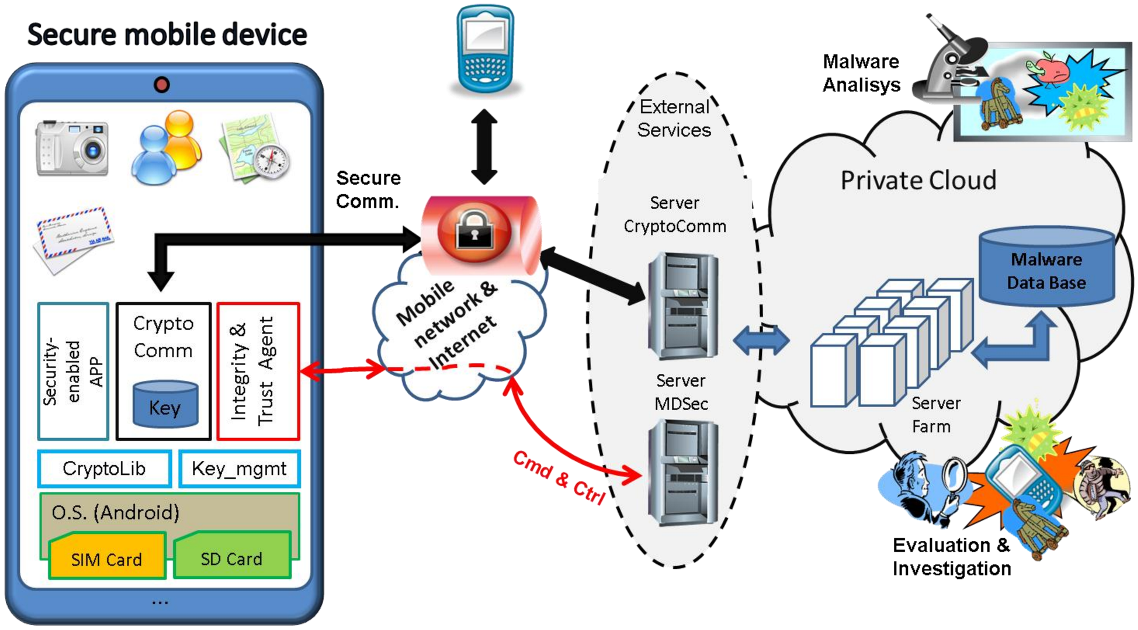

Figure 1. Framework architecture, secure communication, trust management and back office services.

secure communication. Section V explains the main aspects of a cryptographic service provider built for the prototypes. Section VI concludes the paper and points to future directions. For clarification purposes, Table I, at the end, shows a small glossary of cryptographic terms used in this paper.

II. Integrated View and Architecture

There are three main objectives that drive the proposed architecture shown in Figure 1. The first is to build prototypes of secure data communication as well as secure voice over data packets (or over IP), both of them through smartphones on public networks (e.g. 3G, 4G, WiFi). The second is to develop tools for integrity checking on smartphones, as well as techniques for active investigation of security incidents and penetration tests on mobile platforms. Finally, the third objective is to build an environment for experimentation, observation and analysis of mobile malware.

The everyday work is supported by a laboratory for mobile security, which is able to carry out assessments on mobile environments, including platforms, applications and communications, as well as security analysis of mobile malware. The knowledge acquired by the lab team feeds the development team with security controls and counter measures. A private cloud provides services to the development team. Not only security services are provided, but also hosting for servers.

On the client side, Android was chosen as the preferred platform for development of prototypes. The preliminary results described in the next sections addresses three main points of these prototypes: (i) design decisions for secure communication, (ii) secure instant messages and secure SMS, and (iii) a cryptographic library for Android.

III. Related Work

This section outlines recent publications related to the work shown in this paper. Enck et al. [22] show a comprehensive study on the general aspects of Android security. Recently, Braga and Nascimento [1] assessed the feasibility of sophisticated cryptographic services on modern smartphones running Android.

Concerning Short Message Service (SMS) encryption, Pereira et al. [6] show an experimental framework for securing SMS-based communications in mobile phones, which encloses a tailored selection of lightweight cryptographic algorithms and protocols, providing encryption, authentication and signature services. Saxena and Chaudhari [12] researched an approach for securing of SMS which is based upon a variant of ECDSA algorithm. Also, Chavan and Sabnees [16] proposed a technique that combines encryption and compression of SMS messages.

The work of Xuefu and Ming [4] shows the use of eXtensible Messaging and Presence Protocol (XMPP) for Instant Messaging on web and smartphones. Massandy and Munir [8] have done experiments on security aspects of communication, but there are unsolved issues, such as strong authentication, secure storage, and implementation of good cryptography, as shown by Schrittwieser et al. [18].

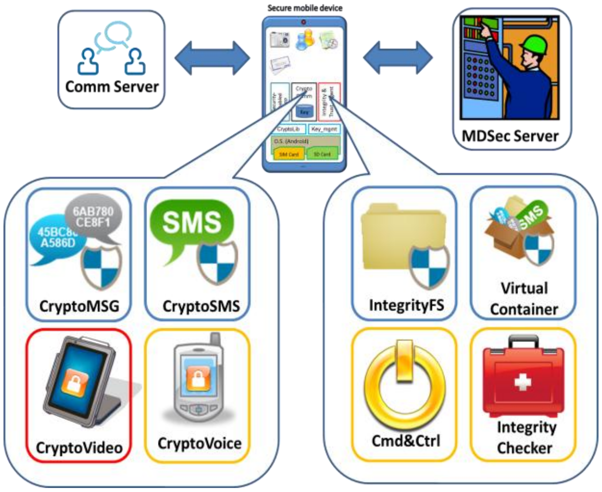

Figure 2. Secure communication and trust management.

Concerning the secure storage of data on mobile devices, a survey by Diesburg and Wang [17] summarizes and compares existing methods of providing confidential storage and deletion of data in personal computing environments, and points that secure deletion of files from flash memory devices is a goal hard to achieve. Wang et al. [23] presented an encrypted file system in user space to protect the removable and persistent storage on smart devices running Android. Reardon et al. [10] addressed the secure deletion of user-space files on Android devices, but with a slow solution.

IV. Secure Communication

Nowadays, secure phone communication does not mean only voice encryption, but encompasses a plethora of security services built over the ordinary smartphone capabilities. To name just a few of them, these are SMS encryption, Instant Message (IM) encryption, voice and video chat encryption, secure conferencing, secure file transfer, secure data storage, secure application containment, and remote security management on the device, including management of cryptographic keys.

Figure 2 illustrates four services of secure communication in scope of this work: CryptoMsg for Instant Messages, CryptoSMS for secure SMS, CryptoVoice and CryptoVideo for secure voice or face-to-face communication, along with a communication server. Figure 2 also shows four secure management services in scope: IntegrityFS for file encryption and integrity, VirtualContainer for application aggregation and containment, Cmd&Ctrl for remote management, and IntegrityChecker for assurance of device's integrity.

By the time of this writing, not all of the services were implemented. In fact, this section describes only two of those cryptographically secure services, namely, encrypted instant messages (CryptoMsg) and encrypted SMS (CryptoSMS).

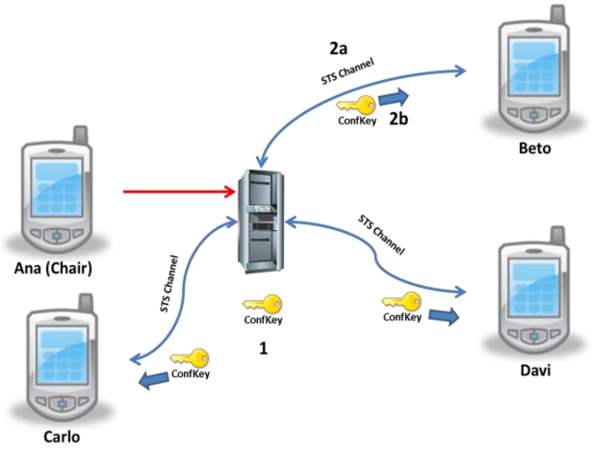

Figure 4. Key agreement for secure conference.

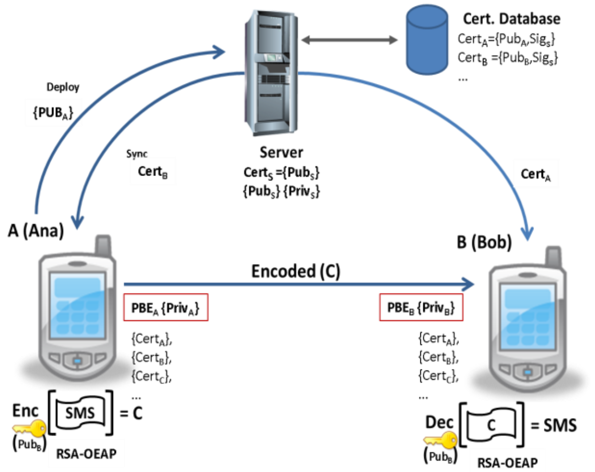

Figure 3. Cryptographically secure SMS.

A. *Cryptographically secure instant message*

The current technology standardized by industry for Instant Messages is the eXtensible Messaging and Presence Protocol (XMPP), the IETF's formalization of base XML streaming protocols for instant messaging and presence, which were originally developed within the Jabber community [15]. The communication architecture supported by XMPP does not allow direct machine-to-machine communication, but requires a server (the XMPP Server) that acts as both a proxy and a mediator among all client applications. This way, the CryptoMsg application is a XMPP client.

An interesting side effect of having chosen XMPP is that CryptoMsg can talk through a proprietary server as well as communicate via Google or Facebook chat servers, as a contingence service, if the primary server is down. Neither Google nor Facebook block encrypted traffic encoded as text, so that two CryptoMsg clients can talk to each other through two Google or Facebook accounts.

CryptoMsg uses a variant of Diffie-Hellman Protocol for key agreement called Station-to-Station (STS). A secret key is negotiated for each chat conversation, and once the key is shared between the two participants, all XMPP messages are encrypted with AES in CBC mode, providing end-to-end encryption. XMPP is actually XML over a TCP communication socket, so TLS can be used instead of regular TCP, for a second layer of point-to-point encryption on the communication channel.

Password-based Encryption (PBE) is the cryptographic technology applied to protect saved conversations. More detail on the cryptographic services provided for this implementation can be found later in this text.

By the time of writing, a secure conference (or group chat) for instant messages was being designed and implemented, as depicted in Figure 4. The Organizer or Chair of the conference requests the conference creation to the Server, and the key agreement for the requested conference proceeds as follows, where $Enc_k(x)$ means encryption of x with key k:

1. Server (S) creates the key for that conference (c_k);
2. For each guest (g[i]), Server (S) does:

   a) Opens a STS channel with key k: $S \leftrightarrow g[i]$, key k;

   b) Sends c_k on time t: $S \rightarrow g[i]$: $Enc_k(c_k; t; C[i])$;

The steps above constitute a point-to-point key transport using symmetric encryption, which is provided by the STS protocol. After that, all guests share the same conference key and the conference proceeds as a multicast of all encrypted messages.

A variation of this design can use the Chair to both generate and distribute the conference key. This extra computation over the Chair can be acceptable under extraordinary circumstances when the primary server is off-line and the number of guests is small.

B. *Cryptographically secure SMS*

Despite the increasing popularity of mobile IM applications, SMS is still useful among those users without a reliable data access. Also, secure SMS can be used as a secure communication channel for other applications.

The solution described in this paper utilizes asymmetric encryption in order to simplify key distribution among users, who may not have data access at the very moment of sending a message. Figure 3 depicts the proposed solution. First of all, users receive from the server, during application installation, the digital certificates of all her contacts, along with server's self-signed certificate. As can be deduced by the reader, this step requires data access. Contacts synchronization and software update also require data access.

A secure SMS can be encrypted and digitally signed, as well. Two implementation issues have to be considered in this scenario. First, the text actually typed by the user, after be encrypted and signed, can result in multiple SMS messages. Upon receiving a series of SMS messages, the application has to be able to sequencing and marshaling the segmented text from various SMS messages. Second, SMS is text only, so a sort of encoding has to be used before transmitting cipher text. The current version of CryptoSMS was designed to use OAEP for encryption and PSS for

Copyright (c) IARIA, 2013. ISBN: 978-1-61208-313-1

digital signatures. Multimedia via SMS is not supported in current version.

V. Cryptographic Service Provider

A cryptographic library for Android was built to provide cryptographic services to be used in the protection of secure communication via mobile devices. In order to be useful, the cryptographic library had to accomplish a minimum set of functional requirements. Each functional requirement generates a set of non-functional or supplementary requirements, mostly related to correctness of algorithms, compliance to industry standards, security, and performance of the implementation.

The design of the current version of the cryptographic library is illustrated in Figure 5 and contains the cryptographic algorithms and protocols described in the following paragraphs. This implementation complies with standard cryptographic API of the Java Cryptographic Architecture (JCA), its name conventions, and design principles [11]. A small glossary of acronyms is in Table I.

In order to provide a fully functional Cryptographic Service Provider (CSP) for secure communication, a minimum set of algorithms had to be chosen. This minimalist construction follows, but is not certified by publicly available standards [13] and provides the following set of services:

a) A symmetric algorithm to be used as block cipher, along with the corresponding key generation function, and modes of operation and padding. The AES algorithm was chosen, along with thee modes of operation: ECB, CBC, and the GCM mode for authenticated encryption. The padding technique for block ciphers is the one defined by PKCS#5.

b) An asymmetric algorithm for digital signatures, along with the key-pair generation function. This requirement brings with it the need for some sort of digital certification of public keys. The asymmetric algorithms are RSA-PSS for signatures and RSA-OAEP for asymmetric encryption. Both of them are probabilistic schemes constructed over ordinary RSA, and are supposed to be more secure than RSA. By the time of writing, RSA-OAEP was not fully implemented;

c) A one-way secure hash function, SHA-1, which is an underling hash function in MACs, digital signatures and Pseudo-Random Number Generators (PRNG). SHA1PRNG was chosen to be used by all the key generation functions;

d) Message Authentication Codes (MAC). HMAC with SHA-1 as the underling hash function, and GMAC, directly derived from GCM mode, where chosen;

e) A key agreement mechanism to be used by communicating parties that have never met before, but need to share an authentic secret and communicate securely. The need for key agreement was fulfilled by the implementation of Station-to-Station (STS) protocol [21], which is based on Authenticated Diffie-Hellman (ADH) [20], and provides mutual key authentication and key confirmation;

f) A simple way to keep keys safe at rest and that does not depend on hardware features. The mechanism for Password-based Encryption (PBE) [3] is based on the Password-Based Key Derivation Function 2 (PBKDF2) [3], and provides a simple and secure way to store keys in encrypted form. In PBE, a key-encryption-key is derived from a password.

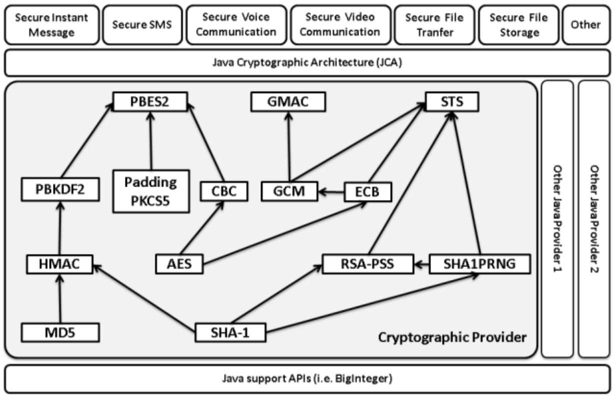

Figure 5. Cryptographic service provider.

A. *How those things are tested, anyway?*

When developing a security-aware application, the first thing to ask is how it will be tested for security. Furthermore, cryptographic software usually requires correctness of basic functions, as well as conformance to specifications and standards. Usually, cryptologists require assistance in writing fast and secure code, because doing it from scratch is almost impossible. Also, canonical implementations, based on standard algorithms, always need optimization and other code transformation in order to be useful in real applications. Code transformation can lead to vulnerabilities, requiring fixes, shipped as software patches, in a never ending cycle.

During this implementation, besides regular functional tests and automated unit test, test vectors were used as automated acceptance tests for cryptographic software. Test vectors are usually available for standardized algorithms [13] and meet halfway between customer and developer, because they come from the problem domain (cryptography) and don't look like source code. They have the benefit to be clear to customer (and to cryptologists) and can be used to reach agreement for when the work is finished. Also, they are not completely freeform and can be used to create automated tests. This approach increases customer’s trust in cryptographic software during algorithm implementation, as well as provides good regression tests as an evidence of correctness after many code transformations.

VI. Conclusion and Future Work

This short paper presented design decisions taken during the construction of a framework for cryptographically secure communication on Android devices. The early prototypes still have many challenges to be overtaken, as follows. The use of cryptosystems for short signatures, elliptic curve cryptography and pairings-based cryptography are some of the improvements planned for the near future. Another challenge is to preserve usability in the presence of strong security concerns. Layered protections against common software vulnerabilities, such as secure deletion of sensitive information in flash memory, are planned in the roadmap.

Finally, automatic testability of cryptography has to be improved not only for algorithms, but also for protocols.

ACKNOWLEDGMENT

The author acknowledge the financial support given to this work, under the project "Security Technologies for Mobile Environments – TSAM", granted by the Fund for Technological Development of Telecommunications – FUNTTEL – of the Brazilian Ministry of Communications, through Agreement Nr. 01.11. 0028.00 with the Financier of Studies and Projects - FINEP / MCTI.

REFERENCES

[1] A. M. Braga and E. N. Nascimento, “Portability Evaluation of Cryptographic Libraries on Android Smartphones”, Proc. 4th International Conference on Cyberspace Safety and Security (CSS), Dec. 2012, pp. 459-469.

[2] ANATEL – Agência Nacional de Telecomunicações, www.anatel.gov.br [retrieved: Oct., 2013].

[3] B. Kaliski, RFC 2898, “PKCS #5: Password-Based Cryptography Specification”, Version 2.0, tools.ietf.org/html/rfc2898 [retrieved: Oct., 2013].

[4] B. Xuefu and Y. Ming, “Design and Implementation of Web Instant Message System Based on XMPP”, Proc. 3rd International Conference on Software Engineering and Service Science (ICSESS), Jun. 2012, pp. 83-88.

[5] BBC News, “Master Key to Android Phones Uncovered”. www.bbc.co.uk/news/technology-23179522. [retrieved: Oct., 2013].

[6] C. Pereira et al., “SMSCrypto: A Lightweight Cryptographic Framework for Secure SMS Transmission”, J. Syst. Softw, vol. 86, no. 3, Mar. 2013, pp. 698-706.

[7] CERT.br – Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil. www.cert.br [retrieved: Oct., 2013].

[8] D. T. Massandy and I. R. Munir, “Secured Video Streaming Development on Smartphones with Android Platform”, Proc. 7th International Conference on Telecommunication Systems, Services, and Applications (TSSA), Oct. 2012, pp. 339-344.

[9] J. Forristal, “Uncovering Android Master Key that Makes 99% of Devices Vulnerable”, bluebox.com/corporate-blog/bluebox-uncovers-android-master-key [retrieved: Oct., 2013].

[10] J. Reardon, C. Marforio, S. Capkun, and D. Basin, “User-level Secure Deletion on Log-structured File Systems”, Proc. 7th ACM Symposium on Information, Computer and Communications Security (ASIACCS), May 2012, pp. 63-64.

[11] JCA Providers Documentation for Java Platform Standard Edition 7, docs.oracle.com/javase/7/docs/technotes/guides/security/SunProviders [retrieved: Oct., 2013]

[12] N. Saxena, N. S. Chaudhari, “Secure Encryption with Digital Signature Approach for Short Message Service”, Proc. World Congress on Information and Communication Technologies (WICT), Nov. 2012, pp. 803-806.

[13] NIST CAVP, Cryptographic Algorithm Validation Program, csrc.nist.gov/groups/STM/cavp/index.html [retrieved: Oct., 2013].

[14] NSA, Mobility Capability Package - Secure VoIP, v. 2.1, www.nsa.gov/ia/_files/Mobility_Capability_Pkg_Vers_2_1.pdf [retrieved: Oct., 2013].

[15] P. Saint-Andre, K. Smith, and R. Tronçon, “XMPP: The Definitive Guide - Building Real-Time Applications with Jabber Technologies”, O’reilly, 2009.

[16] R. Chavan and M. Sabnees, “Secured Mobile Messaging”, Proc. International Conference on Computing, Electronics and Electrical Technologies (ICCEET) , Mar. 2012, pp.1036-1043.

[17] S. Diesburg and A. Wang. “A Survey of Confidential Data Storage and Deletion Methods”. ACM Comp. Surveys, vol. 43, no. 1, Nov. 2010.

[18] S. Schrittwieser et al., “Guess Who's Texting You? Evaluating the Security of Smartphone Messaging Applications”. Proc. 19th Network & Distributed System Security Symposium, Feb. 2012.

[19] The Guardian, Hot site on “Edward Snowden’s disclosure of NSA top-secret information”, www.guardian.co.uk/world/edward-snowden [retrieved: Oct., 2013].

[20] W. Diffie and M. Hellman, "New Directions in Cryptography," IEEE Transact. on Inform. Theory, vol. 22, no. 6, Nov. 1976, pp. 644-654.

[21] W. Diffie, P. C. van Oorschot, and M. J Wiener, “Authentication and Authenticated Key Exchanges”, Designs, Codes and Cryptography, vol. 2, no. 2, 1992, pp. 107–125.

[22] W. Enck, D. Octeau, P. McDaniel, and S. Chaudhuri, “A Study of Android Application Security”, Proc. 20th USENIX conference on Security (SEC), 2011, pp. 21-21.

[23] Z. Wang, R. Murmuria, and A. Stavrou, “Implementing and Optimizing an Encryption Filesystem on Android”, Proc. 13th International Conf. on Mobile Data Management, 2012, pp. 52-62.

TABLE I. OVERVIEW OF CRYPTOGRAPHIC ACRONYMS

Acronym	**Brief Description of the Acronym**
ADH	Authenticated Diffie-Hellman is a way of exchanging cryptographic keys among mutually autenticated parties.
AES	Advanced Encryption Standard is a specification for data encryption established by the U.S. government.
CBC	Cipher Block Channing is an operation mode for block ciphers, in symmetric-key cryptography.
CSP	Cryptographic Service Provider is a collection of cryptographic implementations.
ECB	Electronic Code Book is an operation mode for block ciphers, in symmetric-key cryptography.
GCM	Galois/Counter Mode is an operation mode for block ciphers, in symmetric-key authenticated encryption.
GMAC	An authentication-only variant of GCM.
HMAC	keyed-Hash MAC is a function for calculating a MAC involving a secure hash function and a secret key.
MAC	Message Authentication Code is a small piece of data that provides integrity and authenticity for a message.
OAEP	Optimal Asymmetric Encryption Padding is a padding scheme often used together with RSA encryption.
PBE	Password-Based Encryption is a method of deriving a cryptographic key from a password.
PBKDF2	Password-Based Key Derivation Function 2 is a specific technique of implementing a PBE.
PKCS#5	Public-Key Cryptography Standards 5 is a specification devoted to Password-based Encryption.
PRNG	Pseudo-Random Number Generator is an algorithm for generating sequences of numbers that approximate the properties of truly random numbers (a. g. SHA1PRNG).
PSS	Probabilistic Signature Scheme is a secure way of creating digital signatures with RSA.
RSA	RSA is an algorithm for public-key cryptography that is based on the difficulty of factoring large integers.
SHA-1	Standard Hash Algorithm 1 is a cryptographically secure hash function.
STS	Station-to-Station protocol is a key agreement scheme based on ADH that provides mutual authentication.
TLS	Transport Layer Security is a cryptographic protocol for communication security over the Internet.