



Handshake (computing)

In computing, a **handshake** is a signal between two devices or programs, used to, e.g., authenticate, coordinate. An example is the handshaking between a hypervisor and an application in a guest virtual machine.

In telecommunications, a **handshake** is an automated process of negotiation between two participants (example "Alice and Bob") through the exchange of information that establishes the protocols of a communication link at the start of the communication, before full communication begins.^[1] The handshaking process usually takes place in order to establish rules for communication when a computer attempts to communicate with another device. Signals are usually exchanged between two devices to establish a communication link. For example, when a computer communicates with another device such as a modem, the two devices will signal each other that they are switched on and ready to work, as well as to agree to which protocols are being used.^[2]

Handshaking can negotiate parameters that are acceptable to equipment and systems at both ends of the communication channel, including information transfer rate, coding alphabet, parity, interrupt procedure, and other protocol or hardware features. Handshaking is a technique of communication between two entities. However, within TCP/IP RFCs, the term "handshake" is most commonly used to reference the TCP three-way handshake. For example, the term "handshake" is not present in RFCs covering FTP or SMTP. One exception is Transport Layer Security, TLS, setup, FTP RFC 4217. In place of the term "handshake", FTP RFC 3659 substitutes the term "conversation" for the passing of commands.^{[3][4][5]}

A simple handshaking protocol might only involve the receiver sending a message meaning "I received your last message and I am ready for you to send me another one." A more complex handshaking protocol might allow the sender to ask the receiver if it is ready to receive or for the receiver to reply with a negative acknowledgement meaning "I did not receive your last message correctly, please resend it" (e.g., if the data was corrupted en route).^[6]

Handshaking facilitates connecting relatively heterogeneous systems or equipment over a communication channel without the need for human intervention to set parameters.

Example

TCP three-way handshake

Establishing a normal TCP connection requires three separate steps:

1. The first host (Alice) sends the second host (Bob) a



Example of three way handshaking

- "synchronize" (SYN) message with its own sequence number x , which Bob receives.
2. Bob replies with a synchronize-acknowledgment (SYN-ACK) message with its own sequence number y and acknowledgement number $x + 1$, which Alice receives.
 3. Alice replies with an acknowledgment (ACK) message with acknowledgement number $y + 1$, which Bob receives and to which he doesn't need to reply.
- In this setup, the synchronize messages act as service requests from one server to the other, while the acknowledgement messages return to the requesting server to let it know the message was received.

The reason for the client and server not using a default sequence number such as 0 for establishing the connection is to protect against two incarnations of the same connection reusing the same sequence number too soon, which means a segment from an earlier incarnation of a connection might interfere with a later incarnation of the connection.

SMTP

The Simple Mail Transfer Protocol (SMTP) is the key Internet standard for email transmission. It includes handshaking to negotiate authentication, encryption and maximum message size.

TLS handshake

When a Transport Layer Security (SSL or TLS) connection starts, the record encapsulates a "control" protocol—the handshake messaging protocol (content type 22). This protocol is used to exchange all the information required by both sides for the exchange of the actual application data by TLS. It defines the messages formatting or containing this information and the order of their exchange. These may vary according to the demands of the client and server—i.e., there are several possible procedures to set up the connection. This initial exchange results in a successful

TLS connection (both parties ready to transfer application data with TLS) or an alert message (as specified below).

The protocol is used to negotiate the secure attributes of a session. (RFC 5246, p. 37)^[7]

WPA2 wireless

The WPA2 standard for wireless uses a four-way handshake defined in IEEE 802.11i-2004.

Dial-up access modems

One classic example of handshaking is that of dial-up modems, which typically negotiate communication parameters for a brief period when a connection is first established, and then use those parameters to provide optimal information transfer over the channel as a function of its quality and capacity. The "squealing" (which is actually a sound that changes in pitch 100 times every second) noises made by some modems with speaker output immediately after a connection is established are in fact the sounds of modems at both ends engaging in a handshaking procedure; once the procedure is completed, the speaker might be silenced, depending on the settings of operating system or the application controlling the modem.

Serial "Hardware Handshaking"

This frequently used term describes the use of RTS and CTS signals over a serial interconnection. It is, however, not quite correct; it's not a true form of handshaking, and is better described as flow control.

Mobile device charging

In mobile device chargers offering special quick-charge abilities to supported devices, the charging process will switch up to a higher output voltage for increased power transfer. But this could cause serious damage to an unsupported device or even result in a fire. It is therefore very important for the device and charger to first perform a handshake to "agree" on mutually supported charge parameters. If such a charger can't identify the connected device or determine its compatibility, it will default to normal but much slower charge parameters within the USB standard.

References

1. "What is handshaking? - Definition from WhatIs.com" (<http://searchnetworking.techtarget.com/definition/handshaking>). *SearchNetworking*. Retrieved 2018-02-19.
2. Ware, Peter; Chivers, Bill; Cheleski, Paul (2001). *Jacaranda Information Processes and Technology: HSC Course*. Australia: John Wiley & Sons Australia. pp. 92–93. ISBN 978-0701634728.
3. TCP RFC 793, 2581

4. SMTP RFC 821,5321, 2821, 1869,6531, 2822
 5. FTP 959, 3659 (conversation), 2228,4217 (TLS handshake),5797
 6. "handshaking" (<http://encyclopedia2.thefreedictionary.com/Hand+shake+signal>).
TheFreeDictionary's Encyclopedia.
 7. *The Transport Layer Security (TLS) Protocol, version 1.2* (<https://datatracker.ietf.org/doc/html/rfc5246>). IETF. August 2008. doi:10.17487/RFC5246 (<https://doi.org/10.17487%2FRFC5246>).
RFC 5246 (<https://datatracker.ietf.org/doc/html/rfc5246>).
-

Retrieved from "[https://en.wikipedia.org/w/index.php?title=Handshake_\(computing\)&oldid=1244079204](https://en.wikipedia.org/w/index.php?title=Handshake_(computing)&oldid=1244079204)"

PC-BASED INSTRUMENTATION

CONCEPTS AND PRACTICE

By [N. MATHIVANAN](#) · 2007

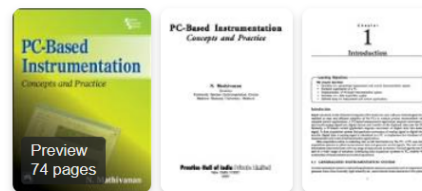
 Preview

 Search inside

[Overview](#)

[Get the book](#)

[More by author](#)



About this edition

ISBN:	9788120330764, 8120330765	Page count:	700
Published:	January 21, 2007	Format:	ebook
Publisher:	PHI Learning	Language:	English
Author:	N. MATHIVANAN		

 Create Citation

 Table of contents

This well-organized book is intended for the undergraduate students of Electrical, Electronics and Communications, Computer, Instrumentation and Instrumentation and Control Engineering; and postgraduate students of science in Electronics, Physics and Instrumentation. Data acquisition being the core of all PC-based measurements and control instrumentation systems engineering, this book presents detailed discussions on PC bus based data acquisition, remote data acquisition, GPIB data acquisition and networked data acquisition configurations. This book also describes sensors, signal-conditioning and principles of PC-based data acquisition. It provides several latest and advanced techniques. This book stresses the need for understanding the use of Personal Computers in measurement and contr...

About the work

Originally published: January 21, 2007
Subject: Technology & Engineering / Electronics / General, [see more](#) ▼

Author

N. MATHIVANAN

Author

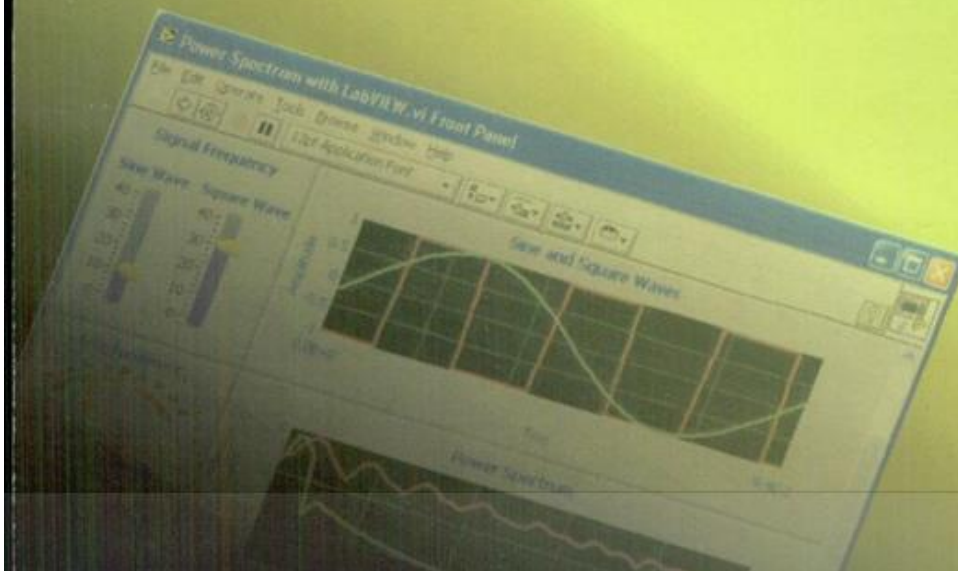


N. MATHIVANAN, Ph.D., is Director, University Science Instrumentation Centre, Madurai Kamaraj University, Madurai. Besides teaching micro-processors and PC hardware, he has designed and developed many PC-based instruments for research applications. Dr. Mathivanan has also published several research papers in national and

Eastern
Economy
Edition

PC-Based Instrumentation

Concepts and Practice



PC-Based Instrumentation

Concepts and Practice

N. Mathivanan

Director

University Science Instrumentation Centre
Madurai Kamaraj University, Madurai

Prentice-Hall of India Private Limited

New Delhi-110001

2007

Table 8.2 GPIB capabilities, their symbols and functional descriptions

<i>Interface capability</i>		<i>Description</i>
<i>Symbol</i>	<i>Function</i>	
T	Talker	Capability of a device to talk.
TE	Extended Talker	Capability of a talker to extended addressing.
L	Listener	Capability of a device to listen.
LE	Extended Listener	Capability of a listener to extended addressing.
SH	Source Handshake	Capability of a device to send command or data byte (multiline message) over GPIB using the three-wire handshake protocol.
AH	Acceptor Handshake	Capability of a device to receive command or data bytes (multiline message) from GPIB using the three-wire handshake protocol.
RL	Remote/Local	Capability of the device to switch between response to its front panel controls and response to commands sent over GPIB.
SR	Service Request	Capability of a device to assert service request to obtain service from controller.
PP	Parallel Poll	Capability of device to indicate that it needs service when the controller requests the status.
DC	Device Clear	Capability of a device to be reset.
DT	Device Trigger	Capability of a device to be triggered by a GPIB command to perform a device dependent function.
C	Controller	Capability of a device to send addresses, universal commands and addressed commands to other devices on the bus. It also includes the capability of the device to conduct polling to determine a device requesting a service.
E	Drivers	It describes the type of electrical bus drivers used in the device.

The IEEE-488 further defines subset of the above capabilities by a number appended to the capability symbol. The standard recommends that the GPIB capabilities of a device be symbolically marked near the GPIB connector on each device to indicate the interface functions the device supports. For example, the symbolic code 'SH1, LH1, T6, L4, SR1, RL1, PP0, DC1, DT0, C0, E2' indicates that the device has the following GPIB capabilities.

- (a) SH1 : Capability to transmit multiline messages across GPIB.
- (b) LH1 : Capability to receive multiline messages from GPIB.
- (c) T6 : Capability to become a talker when talk address with ATN asserted is sent by controller on the bus. Capability to respond to serial poll. The device ceases to be a talker when talk address of another device with ATN asserted is sent by the controller on the bus. The device does not have the talk-only capability.
- (d) L4 : Capability to become a listener when a listen address with ATN asserted is sent by controller on the bus. The device does not have the listen-only capability.
- (e) SR1 : Capability to assert SRQ line when the device requires a service.
- (f) RL1 : Capability to respond to both Go To Local (GTL) and Local Lock Out (LLO) command messages.
- (g) PP0 : Does not have the capability to parallel poll.
- (h) DC1 : Capability to respond to Device Clear (DCL) and Selected Device Clear (SDC) command messages.

Information Security Management Handbook on CD-ROM, 2006 Edition

2006

[Preview](#) [Search inside](#)

[Overview](#)

[Get the book](#)

[Publisher collection](#)



About this edition

ISBN: 9780849385858, 0849385857 Page count: 2,036
Published: April 6, 2006 Format: ebook
Publisher: [Taylor & Francis](#) Language: [English](#)
Editors: Harold F. Tipton, Micki Krause

[Create Citation](#)

[Table of contents](#)

The need for information security management has never been greater. With constantly changing technology, external intrusions, and internal thefts of data, information security officers face threats at every turn.

The Information Security Management Handbook on CD-ROM, 2006 Edition is now available. Containing the complete contents of the Information Security Management Handbook, this is a resource that is portable, linked and searchable by keyword. In addition to an electronic version of the most comprehensive resource for information security management, this CD-ROM contains an extra volume's worth of information that is not found anywhere else, including chapters from other security and networking books that have never appeared in the print editions. Exportable text and hard ...

Source: Publisher

About the work

Originally published: April 6, 2006

Editors: [Harold F. Tipton](#), [Micki Krause](#)

Genre: [Reference work](#)

Subject: Business & Economics / Commerce,
[see more](#) ▼

Publisher

Taylor & Francis



[taylorandfrancis.com](#)

[Search Taylor & Francis](#) ▼

[More from the publisher collection](#)

Information Security Management Handbook

2006 Edition

Library of Congress Cataloging-in-Publication Data

Information security management handbook / Harold F. Tipton, Micki Krause, editors.—2006 ed.

p. cm.

AU8585

ISBN 0-8493-8585-7 (CD)

1. Computer security—Management—Handbooks, manuals, etc. 2. Data protection—Handbooks, manuals, etc. I. Tipton, Harold F. II. Krause, Micki.

QA76.9.A25I54165 2003

658'.0558—dc22

2003061151

This book contains information obtained from authentic and highly regarded sources. Reprinted material is quoted with permission, and sources are indicated. A wide variety of references are listed. Reasonable efforts have been made to publish reliable data and information, but the author and the publisher cannot assume responsibility for the validity of all materials or for the consequences of their use.

Neither this book nor any part may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, microfilming, and recording, or by any information storage or retrieval system, without prior permission in writing from the publisher.

All rights reserved. Authorization to photocopy items for internal or personal use, or the personal or internal use of specific clients, may be granted by CRC Press LLC, provided that \$1.50 per page photocopied is paid directly to Copyright Clearance Center, 222 Rosewood Drive, Danvers, MA 01923 USA. The fee code for users of the Transactional Reporting Service is ISBN 0-8493-8585-7 /03/\$0.00+\$1.50. The fee is subject to change without notice. For organizations that have been granted a photocopy license by the CCC, a separate system of payment has been arranged.

The consent of CRC Press LLC does not extend to copying for general distribution, for promotion, for creating new works, or for resale. Specific permission must be obtained in writing from CRC Press LLC for such copying.

Direct all inquiries to Taylor & Francis Group, LLC, 6000 Broken Sound Parkway, Suite 300, Boca Raton, Florida 33487.

Trademark Notice: Product or corporate names may be trademarks or registered trademarks, and are used only for identification and explanation, without intent to infringe.

Visit the CRC Press Web site at www.crcpress.com

© 2006 by CRC Press LLC

Auerbach is an imprint of CRC Press LLC

No claim to original U.S. Government works

International Standard Book Number 0-8493-8585-7



FIGURE 6.1 TLS Protocol stack.

TABLE 6.1 Security Parameters

Parameter	Description
Session identifier	This value is chosen by the server and is an arbitrary value to identify an active or resumable session state.
Peer certificate	This is the X509v3 [X509] certificate of the peer.
Compression method	The algorithm used to compress data prior to encryption.
Cipher spec	This identifies the bulk encryption algorithm, the MAC algorithm, and any other specific cryptographic attributes for both.
Master secret	This is a 48-byte secret shared between the client and server.
Is resumable	A flag indicating whether the session can be used to initiate new connections.

The shared secret is used to generate the key for the symmetric cryptography used in the Record Protocol. However, of importance here is the high level of protection placed on the secret. During the negotiation, because the secret is protected by asymmetric cryptography, it is not possible for an eavesdropping attacker to recover the secret. Second, the manner in which the negotiation occurs means any attempt by an attacker to modify the communication will be detected. These features provide a high level of security and assurance of the privacy and data integrity of the connection.

Additionally, the Handshake Protocol also provides other sub-protocols to assist in the operation of the protected session. The entire protocol stack is presented in Figure 6.1. This chapter presents the protocol stack and operation of a TLS session.

Understanding the TLS Handshake Protocol

The TLS Handshake Protocol allows two peers to agree upon security parameters for the TLS Record layer, authenticate, initiate those negotiated security parameters, and report errors to each other.

During the session negotiation phase, the Handshake Protocol on each peer negotiates the security parameters in Table 6.1.

Once the session is initiated, however, the application can request a change in the cryptographic elements of the connection. The change is handled through the “change cipher spec protocol,” which sends a message to the peer requesting a change to the cipher properties. The change itself is encrypted with the current cipher values to ensure the request and associated information cannot be deciphered if intercepted.

How the Protocol Works

For TLS to properly protect a session using cryptographic features, it must negotiate the cryptographic parameters. Figure 6.2 illustrates establishing the session.

Upon initiating a TLS connection, the two nodes must establish a “handshake” and negotiate the session parameters. These parameters include the cryptographic values, optional authentication, and generated shared secrets.

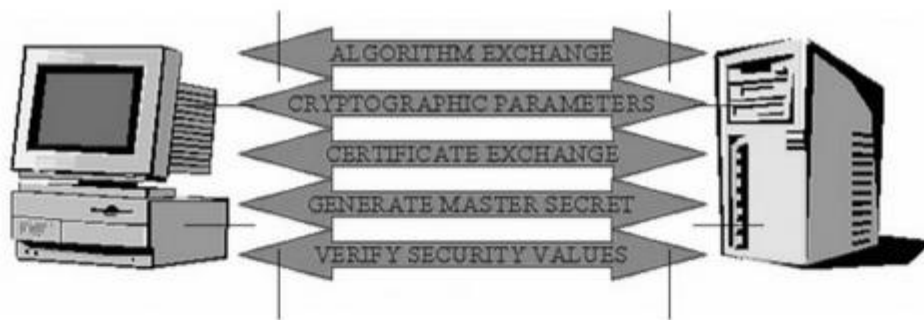


FIGURE 6.2 Handshake setup.

The process breaks down as follows:

1. Each node exchanges a "hello" message to communicate supported cryptographic algorithms, select one that is mutually acceptable, exchange random values used for session initialization, and finally to check to see if this is the resumption of a previous session.
2. Both nodes then exchange the needed cryptographic parameters to agree on a "pre-master" secret.
3. Both nodes exchange their certifications and appropriate cryptographic information to authenticate.
4. Both nodes use the pre-master secret from Step 2 to generate a master value, which is then exchanged.
5. Each node provides the agreed security parameters to the TLS record layer.
6. Verifies the other has calculated the same security parameters and the session was not tampered with by an attacker.

While TLS was designed to minimize the opportunity an attacker has to defeat the system, it may be possible according to RFC 2246 for an attacker to potentially get the two nodes to negotiate the lowest level of agreed encryption. Some methods are described later in this chapter.

Regardless, the higher-level protocols should never assume the strongest protocol has been negotiated and should ensure whatever requirements for the specific connection have been met. For example, 40-bit encryption should never be used, unless the value of the information is sufficiently low as to be worth the effort.

Dissecting the Handshake Protocol

When the client contacts the server to establish a connection, the client sends a client hello message to the server. The server must respond with a server hello message or the connection fails. This is extremely important, as the hello messages provide the security capabilities of the two nodes.

Specifically, the hello message provides the following security capabilities to the other node:

- TLS protocol version
- Session ID
- Available cipher suite
- Compression method

As mentioned, both nodes compute a random value r that is also exchanged in the hello message.

Exchanging the keys can involve up to four discrete messages. The server first sends its certificate, provided the server is to be authenticated. If the certificate is only for signing, the server then sends its public key to the client. The server then sends a "server done" message, indicating that it is waiting for information from the client.



FIGURE 6.3 Handshake exchange.

TABLE 6.2 Supported Certificate Types

Key Type	Description
RSA	This is the RSA public key, which must support use of the key for encryption.
RSA_EXPORT	This is an RSA public key with a length greater than 512 bits used only for signing. Alternatively, it is a key of 512 bits or less that is valid for either encryption or signing.
DHE_DSS	DSS public key.
DHE_DSS_EXPORT	DSS public key.
DHE_RSA	This is an RSA public key used for signing.
DHE_RSA_EXPORT	This is an RSA public key used for signing.
DH_DSS	This is a Diffie-Hellman key. The algorithm used to sign the certificate should be DSS.
DH_RSA	This is a Diffie-Hellman key. The algorithm used to sign the certificate should be RSA.

Note: Due to current restrictions documented in U.S. export laws, RSA values larger than 512 bits for key exchanges cannot be exported from the United States.

The server can send a request to the client for authentication, whereby the client sends its certificate, followed by the client's public key and the "client done" message. The client done message is sent using the agreed-to algorithm, keys, and secrets. The server then responds with similar information and the change to the new agreed-to cipher is complete. This exchange is illustrated in Figure 6.3. At this point, the handshake between the two devices is complete and the session is ready to send application data in the encrypted session.

Resuming an Existing Session

When the client and server agree to either duplicate an existing session to continue a previous session, the handshake is marginally different. In this case, the client sends the "hello" message using the Session ID to be resumed. If the server has a match for that session ID and is willing to re-establish the session, it responds with a "hello" message using the same Session ID. Both the client and server then switch to the previously negotiated and agreed-to session parameters and transmit "done" messages to the other.

If the server does not have a match for the Session ID, or is not willing to establish a session based on the previous parameters, a full handshake must take place.

Certificates

The TLS Protocol is meant to be extensible and provide support in a wide variety of circumstances. Consequently, the certificate types³ in Table 6.2 are supported.