
Practical Cryptology and Web Security

Contents

1. Basic security skills on the World Wide Web

1.1 An introduction to network security

- 1.1.1 Secure and insecure networks
- 1.1.2 Digital cryptography on the web

1.2 The web browser-server dialog

- 1.2.1 The structure and configurations of the web
- 1.2.2 Web browser and server dialog
- 1.2.3 My first page with security
- 1.2.4 Using HTML and migrating to XHTML

1.3 Webpage skills for message manipulation and Security

- 1.3.1 Number Systems Used On The Web
- 1.3.2 The Ascii Character Set
- 1.3.3 Using Unicode On The Web
- 1.3.4 Numerical Representations Of Messages
- 1.3.5 Implementation Of The Caesar Code

1.4 Bitwise Operators And Base64 Encoding/Decoding

- 1.4.1 An Introduction To Bitwise Operators
- 1.4.2 Bitwise Operations and Encryptions on The Web
- 1.4.3 Base64 Encoding And Decoding

1.5 The Xor And Pkzip/Winzip Encryption Schemes

- 1.5.1 Xor Encryption/Decryption
- 1.5.2 Implementation Of The Xor Scheme On The Web
- 1.5.3 Encryption/Decryption Of Pkzip And Winzip
- 1.5.4 Implementation Of The Pkzip/Winzip Encryption Scheme

2. Cryptology, Website Protection And Attacks

2.1 An Overview Of Cryptology

- 2.1.1 What Is Cryptology?
- 2.1.2 Examples On Classic Ciphers And Attacks

2.2 Basic User Authentication And Website Protections

- 2.2.1 The Beginning Of Cipher-Based Authentication
- 2.2.2 Basic HTTP Authentication With Apache
- 2.2.3 Using Access Directives And Group Files
- 2.2.4 Using DBM Database files
- 2.2.5 Inside The Basic Method And Security Attacks

2.3 Digest User Authentication

- 2.3.1 What Is Digest Authentication?
- 2.3.2 Setting Up Digest HTTP Authentication
- 2.3.3 Using Access Directives And Group Files
- 2.3.4 Inside The Digest Method And Security Attacks

2.4 Brute-Force Attacks

- 2.4.1 What Is Brute-Force Attack?
- 2.4.2 The Key-Space Of Brute-Force Attacks

2.5 Implementation And Application Of Brute-Force Schemes

- 2.5.1 The Quotient And Remainder Method
- 2.5.2 The Position Updating Method
- 2.5.3 Attacking Passwords using Brute-Force schemes
- 2.5.4 A Brute-Force Utility To Search MD5 Password

3. One-Way Encryptions, Hash Functions, And Message Digests

3.1 One-Way Function And Encryptions

- 3.1.1 Passwords And One-Way Functions
- 3.1.2 A Step-By-Step Single DES Scheme

3.2 The Single DES Scheme And My First One-Way Encryption: Crypt()

- 3.2.1 Table Lookup Techniques And The DES Sub-Keys
- 3.2.2 DES Encryption On One Chunk (64-Bit) Of Data
- 3.2.3 Performing DES Decryption
- 3.2.4 My First One-Way Encryption: Crypt()
- 3.2.5 Implementation Of Crypt() On The Web

3.3 Hash Functions And The Message Digest: MD5

- 3.3.1 What Are Hash Functions?
- 3.3.2 The 128-Bit MD5 Algorithm
- 3.3.3 The Implementation Of MD5 On The Web
- 3.3.4 Building A Utility From The MD Standard Programs

3.4 Applications Of Message Digests And The MD5crypt Password Scheme

- 3.4.1 The MD5 Checksum And Document Signatures
- 3.4.2 Protecting Downloads Against Viruses And Alterations
- 3.4.3 The MD5crypt Password Algorithm
- 3.4.4 Generating Unix And Apache MD5crypt Passwords

3.5 The secure hash algorithm

- 3.5.1 What is SHA?
- 3.5.2 The Sha-1 algorithm
- 3.5.3 Implementation of Sha-1 on the web
- 3.5.4 Building an Sha-1 utility

4. Some strong symmetric-key ciphers

4.1 An introduction to strong symmetric-key ciphers

- 4.1.1 Strong block ciphers and stream ciphers
- 4.1.2 Operation modes of symmetric-key ciphers

4.2 Coding optimization of the DES scheme

- 4.2.1 The optimization process for Des
- 4.2.2 Combining Table P with S-boxes to generate SP-Boxes
- 4.2.3 Using permutation sequences for table lookup
- 4.2.4 Optimized DES encryption/decryption
- 4.2.5 Setting up and scheduling all sub-keys

4.3 Optimized DES, triple DES, and some encryption tools

- 4.3.1 A functional optimized DES page
- 4.3.2 Adding operation modes to eliminate block effect
- 4.3.3 Double DES and the meet-in-the-middle attack
- 4.3.4 Implementation of triple DES

4.3.5 Building a Tri-DES utility

4.4 A DES-like cipher: the cast-128

4.4.1 What is CAST-128 ?

4.4.2 The CAST-128 encryption/decryption algorithm

4.4.3 Implementation CAST-128 on the web

4.5 Encryption/decryption tools with CAST-128

4.5.1 A functional page for CAST-128 encryption/decryption

4.5.2 Building a CAST-128 utility

4.5.3 Protecting download directories with CAST-128

4.5.4 CAST-128 encryption/decryption with operation modes

5. Practical software-based stream ciphers

5.1 An introduction to stream ciphers

5.1.1 The main characteristics of stream ciphers

5.1.2 Using block cipher to implement stream cipher

5.2 An unbreakable cipher: the one-time-pad

5.2.1 One-time-pad vs brute-force attack

5.2.2 Generating one-time-pad

5.2.3 One-time-pad encryption/decryption

5.2.4 Two OTP applications on the web

5.3 Techniques to generate random key-stream

5.3.1 Linear feedback shift registers

5.3.2 Linear congruential and related algorithms

5.3.3 Randomness tests and cryptographically secure random numbers

5.3.4 Some implementation discussions on big numbers

5.4 Two fast and compact stream ciphers: The RC4 and ISAAC

5.4.1 The RC4 algorithm and discussions

5.4.2 Implementations of the RC4 scheme

5.4.3 The PRNG used by ISAAC

5.4.4 The ISAAC stream cipher and implementations

5.5 A Heavyweight stream cipher: SEAL2

5.5.1 The software-optimized encryption algorithm

5.5.2 Implementation of SEAL2 on the web

5.5.3 Building a SEAL2 utility

6. Block ciphers with variable key-lengths

6.1 A flexible and adaptive block cipher: Blowfish

6.1.1 A block cipher for small devices

6.1.2 The Blowfish algorithm

6.2 Implementation of the Blowfish scheme

6.2.1 A web page for Blowfish encryption/decryption

6.2.2 Standard results and building a utility

6.2.3 Generating any digit of δ and Blowfish constants.

6.3 A fully parameterized block cipher: RC6

6.3.1 The RC6 algorithm

6.3.2 A web page for RC6 encryption/decryption

6.3.3 Testing vectors and building a utility

6.3.4 The magic numbers of RC6

6.4 A step-by-step advanced encryption standard (AES)

6.4.1 From DES to AES challenge

6.4.2 The mathematics used by AES

6.4.3 The AES encryption/decryption algorithm

6.4.4 A step-by-step demonstration of AES

6.5 An optimised implementation of AES

6.5.1 Using optimised table lookup

6.5.2 The implementation of the AES scheme

6.5.3 Testing vectors and building an AES utility

7. Encryption and server skills for web page protection

7.1 Basic encryption skills for web page protection

7.1.1 Protecting web pages with encryption

7.1.2 Generating encrypted web pages

7.1.3 Protecting the intellectual property rights (IPR)

7.1.4 Licensing your Web page with activation code

7.1.5 Using cookies for security

7.2 Server technologies and security

7.2.1 An introduction to server technologies and CGI

7.2.2 CGI technologies and pre-processors

7.2.3 Passing password information to server

7.2.4 Verifying encrypted passwords using Perl script

7.2.5 Verifying passwords with PHP page

7.3 Using server storage with Perl and Php

7.3.1 Access file storage using Perl and Php

7.3.2 User authentication with password file

7.3.3 Adding new account to password file

7.4 Handling password accounts with MySQL database

7.4.1 Creating an encrypted password table in MySQL

7.4.2 User authentication using MySQL database and ODBC

7.4.3 Adding new password account to database

7.4.4 Updating and changing password accounts

8. Practical public-key security and digital signatures

8.1 Security with public-key technology

8.1.1 Key distribution problems and public-key encryption

8.1.2 Data integrity, digital signature, and non-repudiation

8.2 The Diffie-Hellman key exchange scheme

8.2.1 The Diffie-Hellman (DH) key exchange

8.2.2 Using arbitrary precision mathematics (APM) package

8.2.3 The discrete logarithm problem and brute-force attack

8.3 The Elgamal public-key algorithm and digital signatures

8.3.1 The Elgamal public-key algorithm APM package

8.3.3 Implementing message encryption/decryption

8.3.4 Using elgamal scheme for digital signature and data integrity

8.3.5 Implementation of the digital signature scheme

8.4 The RSA scheme, digital signature and hybrid encryption

- 8.4.1 The RSA public-key algorithm and challenge
- 8.4.2 Generating RSA public and secret keys
- 8.4.3 Message encryption/decryption with RSA scheme
- 8.4.4 Sending and receiving secure message with RSA digital signature
- 8.4.5 Building a hybrid encryption scheme: RSA + AES

8.5 Elliptic curves and public-key encryption/decryption

- 8.5.1 What are elliptic curves?
- 8.5.2 Elliptic curve cryptography (ECC)
- 8.5.3 Adding two points on an elliptic curve
- 8.5.4 Scalar multiplication and generating the keys for ECC
- 8.5.5 Encryption/decryption using elliptic curves

9. Security applications with GnuPG, Winpt and server techniques**9.1 An introduction to Gnu privacy guard (GnuPG)**

- 9.1.1 What are PGP, OpenPGP, And GnuPG?
- 9.1.2 The installation and set up of GnuPG

9.2 Using GnuPG for information security

- 9.2.1 Generating and handling public and secret keys
- 9.2.2 Exporting and importing public-keys
- 9.2.3 Message encryption and decryption using GnuPG
- 9.2.4 Signing documents using GnuPG

9.3 Using Winpt On Windows, And Outlook Express

- 9.3.1 The Installation And Set Up Of Winpt
- 9.3.2 Generating keys and key management with Winpt
- 9.3.3 Encryption/decryption with files, clipboard and current window
- 9.3.4 Sending and receiving secure emails with Outlook Express

9.4 Secure emailing with server technologies

- 9.4.1 Calling GnuPG functions using Perl
- 9.4.2 Using Sendmail and Smtplib with Perl
- 9.4.3 Sending encrypted email using GnuPG and Perl
- 9.4.4 Using GnuPG And PHP for secure emailing

9.5 Sending secure attachments with server technologies

- 9.5.1 Mime format: the construction of attachment
- 9.5.2 Sending secure attachments with Perl
- 9.5.3 Programming mime format and secure attachments with PHP

10. SSL security, applications, and XML contracts**10.1 Digital certificates, contracts, and SSL security**

- 10.1.1 The Legal Status Of Digital Contracts
- 10.1.2 Certificates and certificate authorities
- 10.1.3 The SSL Security and HTTPS protocol

10.2 Basic security applications with OpenSSL

- 10.2.1 The installation of OpenSSL
- 10.2.2 Generating hash values
- 10.2.3 Encryption and decryption
- 10.2.4 Generating digital keys

10.3 Generating and signing certificates with OpenSSL

- 10.3.1 Certificate and certificate signing request
- 10.3.2 Signing certificates as a CA with OpenSSL
- 10.3.3 Certificate Formats: X509 And Pkcs#12
- 10.3.4 Importing and exporting certificate with browsers

10.4 Integrating OpenSSL and Apache to build a secure HTTPS site

- 10.4.1 Basic requirements for a secure website
- 10.4.2 Integrate Apache and OpenSSL using Mod_SSL
- 10.4.3 Configure Apache to use SSL
- 10.4.4 Configure Apache for secure and insecure connections
- 10.4.5 Some examples to use the secure site

10.5 XML Security And XML Digital Contracts

- 10.5.1 An introduction to XML
- 10.5.2 Using XSLT and parser on XML documents
- 10.5.3 The XML security library: adding OpenSSL To XML
- 10.5.4 XML encryption and decryption
- 10.5.5 XML signatures and contracts