Netkiller Cryptography 手札

信息安全与加密

netkiller(陈景峰) BG7NYT

中国广东省深圳市龙华区
518000
+86 755 29812080

<openunix@163(.)com>

文档出处: http://netkiller.sourceforge.net/ | http://netkiller.github.com

文档最近一次更新于 Sat Dec 10 04:05:38 UTC 2011

下面是我多年积累下来的经验总结，整理成文档供大家参考:

Netkiller Architect 手札	Netkiller Linux 手札	Netkiller Developer 手札	Netkiller Database 手札
Netkiller Debian 手札	Netkiller CentOS 手札	Netkiller FreeBSD 手札	Netkiller Shell 手札
Netkiller Web 手札	Netkiller Monitoring 手札	Netkiller Storage 手札	Netkiller Mail System 手札
Netkiller Security 手札	Netkiller PostgreSQL 手札	Netkiller MySQL 手札	Netkiller LDAP 手札
Netkiller Cryptography 手札	Netkiller Intranet 手札	Netkiller Cisco IOS 手札	Netkiller Writer 手札
Netkiller Version 手札	Netkiller Studio Linux 手札		

为什么写这篇文章

有很多想法,不能实现.工作中也用不到,所以想写出来,和大家分享.有一点写一点,写得也不好,就当学习笔记了.

我想到那写到那,你会发现文章没一个中心,今天这里写点,明天跳过本章写其它的.
文中例子决对多,对喜欢复制然后粘贴朋友很有用,不用动手写,也省时间.
理论的东西,网上大把,我这里就不写了,需要可以去网上查.
我爱写错别字,还有一些是打错的,如果发现请指正.
另外本文98%是我亲自编写，另有小部分来自引用网上，但作者不详.

Table of Contents

自述

1. 读者对象
2. 内容简介
3. 作者简介

3.1. 联系作者

1. Office

1. 给文档加密码
2. 数字签名

2. UUID (Universally Unique Identifier)

1. GUID
2. Subversion
3. PHP UUID
4. JAVA UUID
5. PERL UUID
6. Python UUID
7. MySQL uuid()
8. linux command uuid

I. Encode & Decode

3. MIME (BASE64) 专题

1. Linux Command base64
2. PHP Base64

2.1. base64_encode
2.2. base64_decode

3. Python Base64
4. perl base64
5. Java Base64
6. C/C++ Base64

4. Uuencode

1. PHP uuencode

5. Quoted-Printable

1. C Quoted-Printable
2. Java Quoted-Printable
3. Python Quoted-Printable

6. DES crypt() 专题

1. C crypt()
2. PHP crypt()
3. perl crypt
4. mysql crypt
5. Java crypt

II. Message Digest

7. MD5专题

1. md5sum
2. PHP md5()
3. MySQL md5()
4. Java MD5
5. perl md5

8. SHA 专题

1. sha1sum
2. PHP sha1()
3. Java SHA
4. Perl

9. CRC32

1. PHP CRC32

10. 第三方工具

1. htpasswd

1.1. CRYPT
1.2. MD5
1.3. SHA

2. htdigest
3. md5sum
4. sha1sum

11. OpenPGP/OpenGPG(GnuPG)

1. GnuPG(OpenGPG)

1.1. 生成密钥对
1.2. 列出密钥
1.3. 验证签字
1.4. GnuPG For Windows
1.5. EMail-Security
1.6. Smart Card

12. Secure Tunnel

1. OpenSSH Tunnel

1.1. SOCKS v5 Tunnel

2. SSL Tunnel

2.1. 通过SSL访问POP、IMAP、SMTP

13. 硬盘分区与文件系统加密

1. Linux磁盘分区加密
2. Microsoft EFS

14. Email Security using OpenPGP and S/MIME

1. Gpg4win
2. S/MIME

III. 数字证书工具

15. OpenSSL

1. 如何创建一个文件的 MD5 或 SHA1 摘要?
2. 编码/解码
3. web 服务器 ssl 证书
4. 去除私钥的密码
5. 证书转换

5.1. CA证书
5.2. 创建CA证书有效期为一年
5.3. x509转换为pfx
5.4. PEM格式的ca.key转换为Microsoft可以识别的pvk格式
5.5. PKCS#12 到 PEM 的转换
5.6. 从 PFX 格式文件中提取私钥格式文件 (.key)
5.7. 转换 pem 到到 spc

5.8. PEM 到 PKCS#12 的转换
5.9. How to Convert PFX Certificate to PEM Format for SOAP

16. Java - keytool

1. 创建证书
2. Private key generation
3. Public Key Certificate (optional)
4. import your signed certificate
5. Import the certificate and attach it to your server key pair
6. Key pair verification

17. .Net makecert

1. 访问X.509证书

IV. 数字证书开发

18. Java (java.security.*)

1. 访问X.509证书
2. 创建证书

19. SSL Socket

1. Java Socket HTTPS
2. Java SSL Socket Client
3. Java SSL Socket Server

20. Credentials Organization

1. VeriSign

1.1. iTrusChina
1.2. Thawte
1.3. Geotrust

2. UserTrust
3. 境内其他CA机构

3.1. WoSign®、I'm Verified®、WoTrust®、沃通®

4. 生成 .csr 文件

List of Examples

12.1. stunnel.conf

自述

Table of Contents

1. 读者对象
2. 内容简介
3. 作者简介

3.1. 联系作者

一直以来,我对数字证书,CA,PKI,非对称加密,非常感兴趣.很想写一篇这方面的文章,因为工作太忙的关系拖到至今.

对于CA,PKI我也是摸着石头过河,网上文章到是不少,全是理论性很强的文章,关于CA,PKI实现很少有文章提及.不要以为我是大拿.兴趣原因,我才写这篇文章.

我研究过一段时间Java.security.*,也写了一些例子,但工作中一直没用到.前几天我安装Mac OS X for x86,使用System Commander引导.System Commander破坏了我的硬盘分区表,我损失惨重,之前的例子也没了.

对数字证书兴趣也很浓,对于OpenSSL,OpenSSH,OpenPGP,GnuPG,SMIME...所用的证书和它们之间的关系一直搞不清楚,只知道他们可以结构很相似,所用的算法和标准,格式不同.

如果对我所谈的方面感兴趣可以去我的邮件列表讨论

如果您想下载这篇文档,或在其它网站看到这篇文章,请使用md5sum校验.

Procedure 1. 校验步骤:

1. 点击下载 security.tar.gz

   点击下载 security.tar.gz.md5

2. 命令:

   ```
   md5sum -c security.tar.gz.md5
   ```

   security.tar.gz.md5内容类似下面:

   ```
   4d24e2b653e48a74dace9b6648eb8815 *book.html
   ```

3. md5sum -c security.tar.gz.md5

   ```
   book.html: OK
   ```

4. OK表示正确,如果非OK请重新下载,或到我的网站下载 http://netkiller.sf.net/

1. 读者对象

本文档的读者对象:

文档面向有所有读者。您可以选读您所需要的章节,无需全篇阅读,因为有些章节不一定对你有用,用得着就翻来看看,暂时用不到的可以不看.

大体分来读者可以分为几类:

1. 系统管理员, 可以选读(Openssl,OpenSSH,OpenPGP...)

2. 程序开发人员, 可以选读(Java.security.*开发,SSL Socket开发)

3. 系统支持,部署工程师

不管是谁,做什么的,我希望通过阅读这篇文档都能对你有所帮助。

我的目的:

1. 通过阅读本文,你可以学会使用md5,sha,base64...技术

2. 懂得使用OpenSSL,OpenSSH,OpenPGP/GnuPG...

3. 使用Java.security.*来创建,修改删除,删除各种证书如:SMIME邮件签名与加密,Web服务器信认证书,SSL Socket开发...

4. 建立一个基本的PKI系统

---

2. 内容简介

本文档容简介:

文档全篇分为基础应用篇，管理篇，开发篇.

文档内容简介:

1. 基础应用篇，主要讲述一般日常用到的加密方法，如Office与PDF文档数字签名，一般权限设置与加密解密等，适合所有人阅读
2. 管理篇,主要是面向系统管理员，主要讲安装配置等
3. 开发篇,向要开发人员,讲述安全通信,和数据证书相关编程等.

3. 作者简介

主页地址：http://netkiller.sourceforge.net, http://netkiller.github.com/

陈景峰 (ㄔㄣˊ ㄐㄧㄥˇ ㄈㄥ)

Nickname：netkiller | English name: Neo chen | Nippon name: ちんけいほう (音訳) | Korean name: | Thailand name:

IT民工， UNIX like Evangelist, 业余无线电爱好者（呼号：BG7NYT）,户外运动以及摄影爱好者。

《PostgreSQL实用实例参考》，《Postfix 完整解决方案》，《Netkiller Linux 手札》的作者

2001年来深圳进城打工,成为一名外来务工者.

2002年我发现不能埋头苦干,埋头搞技术是不对的,还要学会"做人".

2003年这年最惨,公司拖欠工资16000元,打过两次官司2005才付清.

2004年开始加入分布式计算团队,目前成绩

2004-10月开始玩户外和摄影

2005-6月成为中国无线电运动协会会员

2006年单身生活了这么多年,终于找到归宿.

2007物价上涨,金融危机，休息了4个月（其实是找不到工作）

2008终于找到英文学习方法，， 《Netkiller Developer 手札》，《Netkiller Document 手札》

2008-8-8 08:08:08 结婚,后全家迁居湖南省常德市

2009《Netkiller Database 手札》,年底拿到C1驾照

2010对电子打击乐产生兴趣，计划学习爵士鼓

2011 职业生涯路上继续打怪升级

3.1. 联系作者

Mobile: +86 13113668890

Tel: +86 755 2981-2080

Callsign: BG7NYT QTH: Shenzhen, China

注：请不要问我安装问题!

E-Mail: openunix@163.com
IRC irc.freenode.net #ubuntu / #ubuntu-cn

Yahoo: bg7nyt
ICQ: 101888222
AIM: bg7nyt

TM/QQ: 13721218
MSN: netkiller@msn.com
G Talk: 很少开
网易泡泡：很少开

写给火腿:

欢迎无线电爱好者和我QSO,我的QTH在深圳宝安区龙华镇溪山美地12B7CD,设备YAESU FT-50R,FT-60R,FT-7800 144-430双段机,拉杆天线/GP天线 Nagoya MAG-79EL-3W/Yagi

如果这篇文章对你有所帮助,请寄给我一张QSL卡片,qrz.cn or qrz.com or hamcall.net

Personal Amateur Radiostations of P.R.China

ZONE CQ24 ITU44 ShenZhen, China

Best Regards, VY 73! OP. BG7NYT

Prev | Up | Next
2. 内容简介 | Home | Chapter 1. Office

Chapter 1. Office

Table of Contents

1. 给文档加密码
2. 数字签名

1. 给文档加密码

以Word为例,启动Word -> 工具 -> 选项 -> 安全性

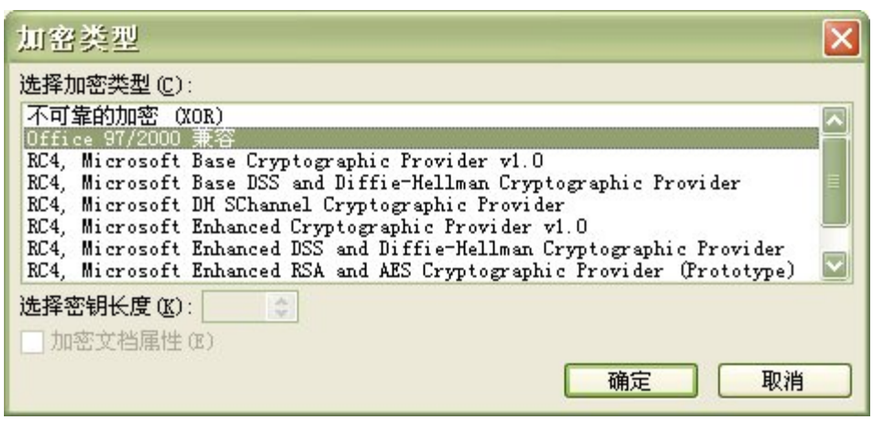
常规操作,在这里输入:

打开密码

修改密码

这样的安全性对一般用户有效,对于电脑高手,他们可以下载一些破解工具,可以在几秒钟内破解你认为安全的密码.

我们现在让密码更安全一些,方法是点击"打开时的文件密码"后面的"高级"按钮



现在你可以看到默认的加密方式是"Offices 97/2000 兼容",这是一种不太安全加密方法

请在列表中选择RAS/DSS算法的加密类型,同时"选择密钥长度"输入大于128的倍数
如:256,1024,2048...

单击"确定"按钮完成操作

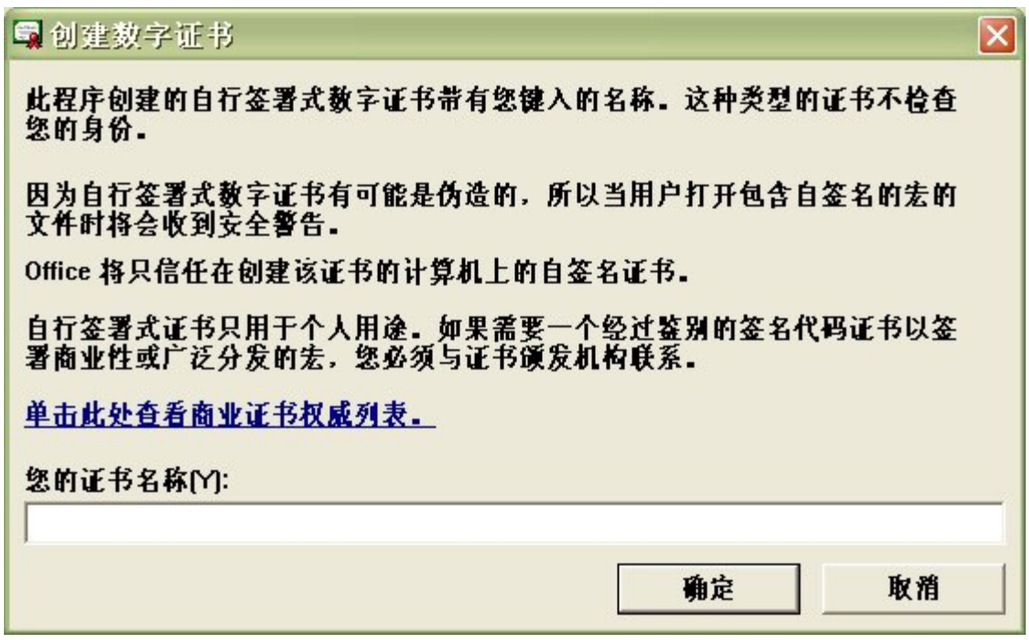
上面的操作也不能保证万无一失,通过穷举算法加黑客字典,只要有足够的时间还可以猜到你的密码,这取决你密码长度和复杂度

---

2. 数字签名

开始菜单 -> 程序 -> Microsoft Office -> Microsoft Office 工具 -> VBA 项目的数字证书

位置:D:\Program Files\Microsoft Office\OFFICE11\SELFCERT.EXE

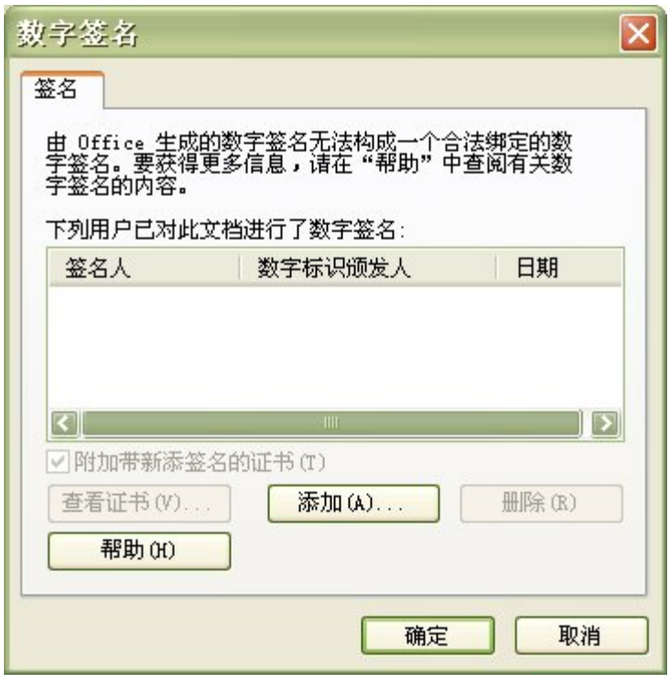


"您的证书名称" 中输入如:netkiller

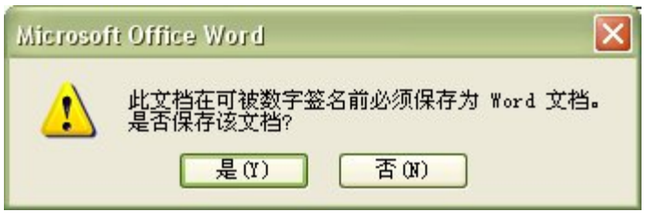
单击"确定"按钮

以Word为例,启动Word 新键文档并输入一些内容或打开一个有内容的文档

Word -> 工具 -> 选项 -> 安全性 -> 数字签名



单击"添加"按钮



单击"是"按钮,同时保存文档

在列表内选择刚才创建是数字证书"netkiller"

单击"OK"按钮

单击"确定"按钮

选项对话框内单击"确定"按钮

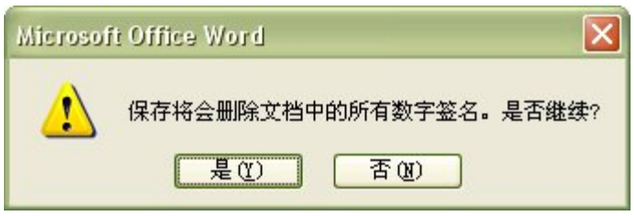
关闭Word 完成操作

验证数字签名是否有效

打开刚刚签名的文档

Word窗口标题栏上(已签名,未验证) 表示签名成功

如果有人修改你的文档,当他再次保存时提示



再打开文档时Word窗口标题栏上"(已签名,未验证)"消失,表示你的文档被其它人撰改

Tip

你同样可以使用第三方数字证书来签名你的文档

Chapter 2. UUID (Universally Unique Identifier)

Table of Contents

1. GUID
2. Subversion
3. PHP UUID
4. JAVA UUID
5. PERL UUID
6. Python UUID
7. MySQL uuid()
8. linux command uuid

以前对UUID的了解很少，只知道是128位整数(16字节)的全局唯一标识符(Universally Unique Identifier)。

UUID 是指在一台机器上生成的数字，它保证对在同一时空中的所有机器都是唯一的。通常平台会提供生成UUID的API。UUID按照开放软件基金会(OSF)制定的标准计算，用到了以太网卡地址、纳秒级时间、芯片ID码和许多可能的数字。由以下几部分的组合：当前日期和时间(UUID的第一个部分与时间有关，如果你在生成一个UUID之后，过几秒又生成一个UUID，则第一个部分不同，其余相同)，时钟序列，全局唯一的IEEE机器识别号（如果有网卡，从网卡获得，没有网卡以其他方式获得），UUID的唯一缺陷在于生成的结果串会比较长。关于UUID这个标准使用最普遍的是微软的GUID(Globals Unique Identifiers)。

其格式为：xxxxxxxx-xxxx-xxxx-xxxxxxxxxxxxxxxx(8-4-4-16)，其中每个 x 是 0-9 或 a-f 范围内的一个十六进制的数字。而标准的UUID格式为：xxxxxxxx-xxxx-xxxx-xxxxxx-xxxxxxxxxx (8-4-4-4-12)

使用UUID的好处在分布式的软件系统中（比如：DCE/RPC, COM+,CORBA）就能体现出来，它能保证每个节点所生成的标识都不会重复，并且随着WEB服务等整合技术的发展，UUID的优势将更加明显。

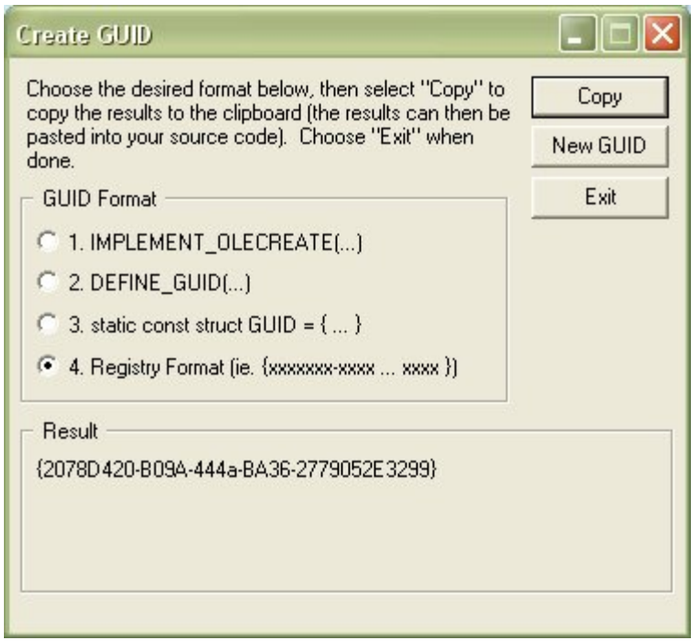
http://en.wikipedia.org/wiki/UUID

RFC

1. GUID

GUID是UUID的windows实现，GUID也是一个128位长的数字，一般用16进制表示。算法的核心思想是结合机器的网卡、当地时间、一个随机数来生成GUID。从理论上讲，如果一台机器每秒产生10000000个GUID，则可以保证（概率意义上）3240年不重复。

到微软件网站下载GUIDGEN.EXE来生成GUID



点击"New GUID"生成新GUID

单击"Copy"复制到剪贴板

生成的GUID：{12466768-64A9-426a-A2E8-ABFDB016B248}

---

2. Subversion

svnlook uuid — 打印版本库的UUID。

```
svnlook uuid REPOS_PATH
```

打印版本库的UUID，UUID是版本库的universal unique IDentifier（全局唯一标示），Subversion客户端可以使用这个标示区分不同的版本库。

```
$ svnlook uuid /usr/local/svn/repos
e7fe1b91-8cd5-0310-98dd-2f12e793c5e8
```

请参考：http://www.subversion.org.cn/svnbook/nightly/index.html

3. PHP UUID

```
<?php
/* Copyright 2006 Maciej Strzelecki

   This program is free software; you can redistribute it and/or modify
   it under the terms of the GNU General Public License as published by
   the Free Software Foundation; either version 2 of the License, or
   (at your option) any later version.

   This program is distributed in the hope that it will be useful,
   but WITHOUT ANY WARRANTY; without even the implied warranty of
   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
   GNU General Public License for more details.

   You should have received a copy of the GNU General Public License
   along with this program; if not, write to the Free Software
   Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA  02110-1301  USA */

function uuid()
{
   // version 4 UUID
   return sprintf(
       '%08x-%04x-%04x-%02x%02x-%012x',
       mt_rand(),
       mt_rand(0, 65535),
       bindec(substr_replace(
           sprintf('%016b', mt_rand(0, 65535)), '0100', 11, 4)
       ),
       bindec(substr_replace(sprintf('%08b', mt_rand(0, 255)), '01', 5, 2)),
       mt_rand(0, 255),
       mt_rand()
```

```
    );
}
?>
```

参考：http://cn.php.net/uniqid

Prev | Up | Next
2. Subversion | Home | 4. JAVA UUID

4. JAVA UUID

```
import java.util.UUID;
public class Test {
        public static void main(String[] args) {
                UUID uuid = UUID.randomUUID();
                System.out.println (uuid);
        }
}
```

编译运行输出：

```
07ca3dec-b674-41d0-af9e-9c37583b08bb
```

参考：http://java.sun.com/j2se/1.5.0/docs/api/java/util/UUID.html

5. PERL UUID

```
#!/usr/bin/perl
use CGI::Carp qw(fatalsToBrowser);

my $uuid_str;
if (@ARGV) {
  $uuid_str = $ARGV[0];
} else {
  eval {
    require Data::UUID;
    my $ug = new Data::UUID;
    $uuid_str = $ug->create_str;
  };
  if ($@) {
    $uuid_str = `uuidgen`;
    $uuid_str =~ s/\r?\n?$//;
  }
}
my @stuff = split /-/, $uuid_str;

print "Content-type: text/html\n\n";
print "<html><head><title>GUID Generator</title></head><body>";
print '<h2><font face="verdana, arial">GUID Generator</font></h2>';
print '<font face="new courier, courier">';
print "{$uuid_str}</font><br>";
print '<h6><font face="verdana, arial"><a
href="http://extensions.roachfiend.com/cgi-bin/guid.pl">Get another
GUID</a></font></h6>';
print '<h6><font face="verdana, arial"><a
href="http://extensions.roachfiend.com/guid.txt">View the source of this
script</a></font></h6>';
print "</body></html>";
exit;
```

參考：http://extensions.roachfiend.com/cgi-bin/guid.pl

6. Python UUID

```
"""UUID (universally unique identifiers) as specified in RFC 4122.

This module provides the UUID class and the functions uuid1(), uuid3(),
uuid4(), uuid5() for generating version 1, 3, 4, and 5 UUIDs respectively.

This module works with Python 2.3 or higher."""

__author__ = 'Ka-Ping Yee <ping@zesty.ca>'
__date__ = '$Date: 2005/11/30 11:51:58 $'.split()[1].replace('/', '-')
__version__ = '$Revision: 1.10 $'

RESERVED_NCS, RFC_4122, RESERVED_MICROSOFT, RESERVED_FUTURE = [
    'reserved for NCS compatibility', 'specified in RFC 4122',
    'reserved for Microsoft compatibility', 'reserved for future definition']

class UUID(object):
    """Instances of the UUID class represent UUIDs as specified in RFC 4122.
    Converting a UUID to a string using str() produces a string in the form
    "{12345678-1234-1234-1234-123456789abc}".  The UUID constructor accepts
    a similar string (braces and hyphens optional), or six integer arguments
    (with 32-bit, 16-bit, 16-bit, 8-bit, 8-bit, and 48-bit values
    respectively).  UUID objects have the following attributes:

        bytes       gets or sets the UUID as a 16-byte string

        urn         gets the UUID as a URN as specified in RFC 4122

        variant     gets or sets the UUID variant as one of the constants
                    RESERVED_NCS, RFC_4122, RESERVED_MICROSOFT, RESERVED_FUTURE

        version     gets or sets the UUID version number (1 through 5)
    """

    def __init__(self, *args):
        """Create a UUID either from a string representation in hexadecimal
        or from six integers (32-bit time_low, 16-bit time_mid, 16-bit
        time_hi_ver, 8-bit clock_hi_res, 8-bit clock_low, 48-bit node)."""
        if len(args) == 1:
            digits = args[0].replace('urn:', '').replace('uuid:', '')
            digits = digits.replace('{', '').replace('}', '').replace('-', '')
            assert len(digits) == 32, ValueError('badly formed UUID string')
            time_low = int(digits[:8], 16)
            time_mid = int(digits[8:12], 16)
            time_hi_ver = int(digits[12:16], 16)
            clock_hi_res = int(digits[16:18], 16)
            clock_low = int(digits[18:20], 16)
            node = int(digits[20:32], 16)
        else:
            (time_low, time_mid, time_hi_ver,
             clock_hi_res, clock_low, node) = args
        assert 0 <= time_low < 0x100000000, ValueError('time_low out of range')
        assert 0 <= time_mid < 1<<16, ValueError('time_mid out of range')
        assert 0 <= time_hi_ver < 1<<16, ValueError('time_hi_ver out of range')
        assert 0 <= clock_hi_res < 1<<8, ValueError('clock_hi_res out of range')
        assert 0 <= clock_low < 1<<8, ValueError('clock_low out of range')
        assert 0 <= node < 0x1000000000000, ValueError('node out of range')
        self.time_low = time_low
        self.time_mid = time_mid
        self.time_hi_ver = time_hi_ver
        self.clock_hi_res = clock_hi_res
        self.clock_low = clock_low
        self.node = node

    def __cmp__(self, other):
        return cmp(self.bytes, getattr(other, 'bytes', other))
```

```
    def __str__(self):
        return '{%08x-%04x-%04x-%02x%02x-%012x}' % (
            self.time_low, self.time_mid, self.time_hi_ver,
            self.clock_hi_res, self.clock_low, self.node)

    def __repr__(self):
        return 'UUID(%r)' % str(self)

    def get_bytes(self):
        def byte(n):
            return chr(n & 0xff)

        return (byte(self.time_low >> 24) + byte(self.time_low >> 16) +
                byte(self.time_low >> 8) + byte(self.time_low) +
                byte(self.time_mid >> 8) + byte(self.time_mid) +
                byte(self.time_hi_ver >> 8) + byte(self.time_hi_ver) +
                byte(self.clock_hi_res) + byte(self.clock_low) +
                byte(self.node >> 40) + byte(self.node >> 32) +
                byte(self.node >> 24) + byte(self.node >> 16) +
                byte(self.node >> 8) + byte(self.node))

    def set_bytes(self, bytes):
        values = map(ord, bytes)
        self.time_low = ((values[0] << 24) + (values[1] << 16) +
                         (values[2] << 8) + values[3])
        self.time_mid = (values[4] << 8) + values[5]
        self.time_hi_ver = (values[6] << 8) + values[7]
        self.clock_hi_res = values[8]
        self.clock_low = values[9]
        self.node = ((values[10] << 40) + (values[11] << 32) +
                     (values[12] << 24) + (values[13] << 16) +
                     (values[14] << 8) + values[15])

    bytes = property(get_bytes, set_bytes)

    def get_urn(self):
        return 'urn:uuid:%08x-%04x-%04x-%02x%02x-%012x' % (
            self.time_low, self.time_mid, self.time_hi_ver,
            self.clock_hi_res, self.clock_low, self.node)

    urn = property(get_urn)

    def get_variant(self):
        if not self.clock_hi_res & 0x80:
            return RESERVED_NCS
        elif not self.clock_hi_res & 0x40:
            return RFC_4122
        elif not self.clock_hi_res & 0x20:
            return RESERVED_MICROSOFT
        else:
            return RESERVED_FUTURE

    def set_variant(self, variant):
        if variant == RESERVED_NCS:
            self.clock_hi_res &= 0x7f
        elif variant == RFC_4122:
            self.clock_hi_res &= 0x3f
            self.clock_hi_res |= 0x80
        elif variant == RESERVED_MICROSOFT:
            self.clock_hi_res &= 0x1f
            self.clock_hi_res |= 0xc0
        elif variant == RESERVED_FUTURE:
            self.clock_hi_res &= 0x1f
            self.clock_hi_res |= 0xe0
        else:
            raise ValueError('illegal variant identifier')

    variant = property(get_variant, set_variant)

    def get_version(self):
        return self.time_hi_ver >> 12

    def set_version(self, version):
        assert 1 <= version <= 5, ValueError('illegal version number')
        self.time_hi_ver &= 0x0fff
        self.time_hi_ver |= (version << 12)

    version = property(get_version, set_version)

def unixgetaddr(program):
    """Get the hardware address on a Unix machine."""
    from os import popen
    for line in popen(program):
        words = line.lower().split()
```

```
        if 'hwaddr' in words:
            addr = words[words.index('hwaddr') + 1]
            return int(addr.replace(':', ''), 16)
        if 'ether' in words:
            addr = words[words.index('ether') + 1]
            return int(addr.replace(':', ''), 16)

def wingetaddr(program):
    """Get the hardware address on a Windows machine."""
    from os import popen
    for line in popen(program + ' /all'):
        if line.strip().lower().startswith('physical address'):
            addr = line.split(':')[-1].strip()
            return int(addr.replace('-', ''), 16)

def getaddr():
    """Get the hardware address as a 48-bit integer."""
    from os.path import join, isfile
    for dir in ['/sbin', '/usr/sbin', r'c:\windows',
                r'c:\windows\system', r'c:\windows\system32']:
        if isfile(join(dir, 'ifconfig')):
            return unixgetaddr(join(dir, 'ifconfig'))
        if isfile(join(dir, 'ipconfig.exe')):
            return wingetaddr(join(dir, 'ipconfig.exe'))

def uuid1():
    """Generate a UUID based on the time and hardware address."""
    from time import time
    from random import randrange
    nanoseconds = int(time() * 1e9)
    # 0x01b21dd213814000 is the number of 100-ns intervals between the
    # UUID epoch 1582-10-15 00:00:00 and the Unix epoch 1970-01-01 00:00:00.
    timestamp = int(nanoseconds/100) + 0x01b21dd213814000
    clock = randrange(1<<16) # don't use stable storage
    time_low = timestamp & (0x100000000 - 1)
    time_mid = (timestamp >> 32) & 0xffff
    time_hi_ver = (timestamp >> 48) & 0x0fff
    clock_low = clock & 0xff
    clock_hi_res = (clock >> 8) & 0x3f
    node = getaddr()
    uuid = UUID(time_low, time_mid, time_hi_ver, clock_low, clock_hi_res, node)
    uuid.variant = RFC_4122
    uuid.version = 1
    return uuid

def uuid3(namespace, name):
    """Generate a UUID from the MD5 hash of a namespace UUID and a name."""
    from md5 import md5
    uuid = UUID(0, 0, 0, 0, 0, 0)
    uuid.bytes = md5(namespace.bytes + name).digest()[:16]
    uuid.variant = RFC_4122
    uuid.version = 3
    return uuid

def uuid4():
    """Generate a random UUID."""
    try:
        from os import urandom
    except:
        from random import randrange
        uuid = UUID(randrange(1<<32), randrange(1<<16), randrange(1<<16),
                    randrange(1<<8), randrange(1<<8), randrange(1<<48))
    else:
        uuid = UUID(0, 0, 0, 0, 0, 0)
        uuid.bytes = urandom(16)
    uuid.variant = RFC_4122
    uuid.version = 4
    return uuid

def uuid5(namespace, name):
    """Generate a UUID from the SHA-1 hash of a namespace UUID and a name."""
    from sha import sha
    uuid = UUID(0, 0, 0, 0, 0, 0)
    uuid.bytes = sha(namespace.bytes + name).digest()[:16]
    uuid.variant = RFC_4122
    uuid.version = 5
    return uuid

NAMESPACE_DNS = UUID('{6ba7b810-9dad-11d1-80b4-00c04fd430c8}')
NAMESPACE_URL = UUID('{6ba7b811-9dad-11d1-80b4-00c04fd430c8}')
NAMESPACE_OID = UUID('{6ba7b812-9dad-11d1-80b4-00c04fd430c8}')
NAMESPACE_X500 = UUID('{6ba7b814-9dad-11d1-80b4-00c04fd430c8}')
```

参考：http://zesty.ca/python/uuid.html

参考：https://svn.n-h.com/svn/exchange4linux/trunk/src/BILL-StorageServer/UUID.py

---

Prev | Up | Next

5. PERL UUID | Home | 7. MySQL uuid()

7. MySQL uuid()

```
mysql> select uuid();
+--------------------------------------+
| uuid()                               |
+--------------------------------------+
| 2f761256-8360-102c-b767-001cc07156cb |
+--------------------------------------+
1 row in set (0.02 sec)
```

8. linux command uuid

```
$ sudo apt-get install uuid

$ uuid
5ce08f58-21ac-11de-a16f-001cc07156cb
```

Part I. Encode & Decode

Table of Contents

3. MIME (BASE64) 专题

1. Linux Command base64
2. PHP Base64

2.1. base64_encode
2.2. base64_decode

3. Python Base64
4. perl base64
5. Java Base64
6. C/C++ Base64

4. Uuencode

1. PHP uuencode

5. Quoted-Printable

1. C Quoted-Printable
2. Java Quoted-Printable
3. Python Quoted-Printable

Chapter 3. MIME (BASE64) 专题

Table of Contents

1. Linux Command base64
2. PHP Base64

2.1. base64_encode
2.2. base64_decode

3. Python Base64
4. perl base64
5. Java Base64
6. C/C++ Base64

什么是Base64?

按照RFC2045的定义，Base64被定义为：Base64内容传送编码被设计用来把任意序列的8位字节描述为一种不易被人直接识别的形式。

为什么要使用Base64?

在设计这个编码的时候，我想设计人员最主要考虑了3个问题：

1. 是否加密？
2. 加密算法复杂程度和效率？
3. 如何处理传输？

加密是肯定的，但是加密的目的不是让用户发送非常安全的Email。这种加密方式主要就是“防君子不防小人”。即达到一眼望去完全看不出内容即可。基于这个目的加密算法的复杂程度和效率也就不能太大和太低。和上一个理由类似，MIME协议等用于发送Email的协议解决的是如何收发Email，而并不是如何安全的收发Email。因此算法的复杂程度要小，效率要高，否则因为发送Email而大量占用资源，路就有点走歪了。

但是，如果是基于以上两点，那么我们使用最简单的恺撒法即可，为什么Base64看起来要比恺撒法复杂呢？这是因为在Email的传送过程中，由于历史原因，Email只被允许传送ASCII字符，即一个8位字节的低7位。因此，如果您发送了一封带有非ASCII字符（即字节的最高位是1）的Email通过有“历史问题”的网关时就可能会出现问题。网关可能会把最高位置为0！很明显，问题就这样产生了！因此，为了能够正常的传送Email，这个问题就必须考虑！所以，单单靠改变字母的位置的恺撒之类的方案也就不行了。关于这一点可以参考 RFC2046。基于以上的一些主要原因产生了Base64编码。

参考邮件正文 Content-Transfer-Encoding: base64

OpenSSL - Base64

1. Linux Command base64

```
$ cat file | base64
```

2. PHP Base64

2.1. base64_encode

base64_encode

(PHP 3, PHP 4, PHP 5)

base64_encode -- 使用 MIME base64 对数据进行编码

说明

string base64_encode (string data)

base64_encode() returns 使用 base64 对 data 进行编码。设计此种编码是为了使二进制数据可以通过非纯 8-bit 的传输层传输，例如电子邮件的主体。

Base64-encoded 数据要比原始数据多占用 33% 左右的空间。

例子 1. base64_encode() 示例

```
<?php
  $str = 'This is an encoded string';
  echo base64_encode($str);
?>
```

此示例将显示：

VGhpcyBpcyBhbiBlbmNvZGVkIHN0cmluZw==

例子 2. stream_filter_append() 示例

```
<?php
$fp = fopen('php://output', 'w');
stream_filter_append($fp, 'convert.base64-encode');
fwrite($fp, "This is a test.\n");
fclose($fp);
/* Outputs:  VGhpcyBpcyBhIHRlc3QuCg==  */
echo "\n============================================\n";

$fp = fopen('php://output', 'w');
stream_filter_append($fp, 'convert.base64-decode');
fwrite($fp, "VGhpcyBpcyBhIHRlc3QuCg==");
fclose($fp);
/* Outputs:  This is a test.  */
echo "============================================\n";

$param = array('line-length' => 8, 'line-break-chars' => "\r\n");
$fp = fopen('php://output', 'w');
stream_filter_append($fp, 'convert.base64-encode', STREAM_FILTER_WRITE, $param);
fwrite($fp, "This is a test.\n");
fclose($fp);
/* Outputs:  VGhpcyBp
         :  cyBhIHRl
         :  c3QuCg==  */
?>
```

2.2. base64_decode

base64_decode

(PHP 3, PHP 4, PHP 5)

base64_decode -- 对使用 MIME base64 编码的数据进行解码

说明

string base64_decode (string encoded_data)

base64_decode() 对 encoded_data 进行解码，返回原始数据，失败则返回 FALSE。返回的数据可能是二进制的。

例子 1. base64_decode() 示例

```
<?php
$str = 'VGhpcyBpcyBhbiBlbmNvZGVkIHN0cmluZw==';
echo base64_decode($str);
?>
```

此示例将显示：

This is an encoded string

---

3. Python Base64

编码：b64encode

```
import base64
base64.b64encode('This is an encoded string')
```

此示例将显示：

’VGhpcyBpcyBhbiBlbmNvZGVkIHN0cmluZw==’

解码：

```
import base64
base64.b64decode('VGhpcyBpcyBhbiBlbmNvZGVkIHN0cmluZw==')
```

此示例将显示：

This is an encoded string

4. perl base64

```
perl -MMIME::Base64 -e 'print encode_base64("netkiller");'

perl -MMIME::Base64 -e 'print decode_base64("bmV0a2lsbGVy");'
```

5. Java Base64

```
import java.io.*;
public class base64Test {

        public static void main(String[] args) {

            try {
                String text = "This is an encoded string";
                //Convert a string to base64 string
                byte[] buf = text.getBytes();
                String encode = new sun.misc.BASE64Encoder().encode(buf);
                System.out.println(encode);

                // Convert base64 string to a string
                buf = new sun.misc.BASE64Decoder().decodeBuffer(encode);
                String decode = new String(buf);
                System.out.println(decode);
            } catch (IOException e) {

            }
        }
}
```

6. C/C++ Base64

Chapter 4. Uuencode

Table of Contents

1. PHP uuencode

Note

uuencode不是MIME标准

application/x-uuencode

Uuencode 是将二进制文件以文本文件方式进行编码表示、以利于基于文本传输环境中进行二进制文件的传输/交换的编码方法之一， 在邮件系统/二进制新闻组中使用频率比较高，经常用于附件二进制文件。

这种编码的特征是：每一行开头用“M”标志。

Uuencode的算法很简单，编码时它将3个字符顺序放入一个 24 位的缓冲区，缺字符的地方补零，然后将缓冲区截断成为 4 个部分，高位在先，每个部分 6 位，用下面的64个字符重新表示：

"`!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_"

解码时它将4个字符分别转换为4个6位字符后，截取有用的后六位放入一个 24 位的缓冲区，即得3个二进制代码。

1. PHP uuencode

编码：convert_uuencode()

```
<?php
$some_string = "test\ntext text\r\n";
echo convert_uuencode($some_string);
?>
```

解码：convert_uuencode()

```
<?php
        $some_string = "This is an encoded string";
        $encode = convert_uuencode($some_string);
        echo convert_uudecode($encode);
?>
```

Chapter 5. Quoted-Printable

Table of Contents

1. C Quoted-Printable
2. Java Quoted-Printable
3. Python Quoted-Printable

Quoted-Printable也是MIME邮件中常用的编码方式之一。同Base64一样，它也将输入的字符串或数据编码成全是ASCII码的可打印字符串。

Quoted-Printable编码的基本方法是：输入数据在33-60、62-126范围内的，直接输出；其它的需编码为“=”加两个字节的HEX码(大写)。为保证输出行不超过规定长度，可在行尾加“=\r\n”序列作为软回车。

1. C Quoted-Printable

```
int EncodeQuoted(const unsigned char* pSrc, char* pDst, int nSrcLen, int
nMaxLineLen)
{
    int nDstLen;         // 输出的字符计数
    int nLineLen;        // 输出的行长度计数

    nDstLen = 0;
    nLineLen = 0;

    for (int i = 0; i < nSrcLen; i++, pSrc++)
    {
        // ASCII 33-60, 62-126原样输出，其余的需编码
        if ((*pSrc >= '!') && (*pSrc <= '~') && (*pSrc != '='))
        {
            *pDst++ = (char)*pSrc;
            nDstLen++;
            nLineLen++;
        }
        else
        {
            sprintf(pDst, "=%02X", *pSrc);
            pDst += 3;
            nDstLen += 3;
            nLineLen += 3;
        }

        // 输出换行?
        if (nLineLen >= nMaxLineLen - 3)
        {
            sprintf(pDst, "=\r\n");
            pDst += 3;
            nDstLen += 3;
            nLineLen = 0;
        }
    }

    // 输出加个结束符
    *pDst = '\0';

    return nDstLen;
}
```

Quoted-Printable解码很简单，将编码过程反过来就行了。

```
int DecodeQuoted(const char* pSrc, unsigned char* pDst, int nSrcLen)
{
    int nDstLen;        // 输出的字符计数
    int i;

    i = 0;
    nDstLen = 0;

    while (i < nSrcLen)
    {
        if (strncmp(pSrc, "=\r\n", 3) == 0)        // 软回车，跳过
        {
            pSrc += 3;
            i += 3;
        }
        else
        {
            if (*pSrc == '=')        // 是编码字节
            {
                sscanf(pSrc, "=%02X", pDst);
                pDst++;
                pSrc += 3;
                i += 3;
            }
            else        // 非编码字节
            {
                *pDst++ = (unsigned char)*pSrc++;
                i++;
            }

            nDstLen++;
        }
    }

    // 输出加个结束符
    *pDst = '\0';

    return nDstLen;
}
```

参考:http://dev.csdn.net/develop/article/19/19205.shtm

---

2. Java Quoted-Printable

3. Python Quoted-Printable

Chapter 6. DES crypt() 专题

Table of Contents

1. C crypt()
2. PHP crypt()
3. perl crypt
4. mysql crypt
5. Java crypt

Tip

CRYPT_MD5 是Unix like Shadow密码

1. C crypt()

crypt是个密码加密函数，它是基於Data Encryption Standard(DES)演算法。

crypt基本上是One way encryption，因此它只适用於密码的使用，不适合於资料加密。

char *crypt(const char *key, const char *salt);

key 是使用者的密码。salt是两个字，每个字可从[a-zA-Z0-9./]中选出来，因此同一密码增加了4096种可能性。透过使用key中每个字的低七位元，取得 56-bit关键字，这56-bit关键字被用来加密成一组字，这组字有13个可显示的 ASCII字，包含开头两个salt。

```
[root@linux root]# cat crypt.c
/*
Netkiller 2003-06-27 crypt.c
char *crypt(const char *key, const char *salt);
*/
#include <unistd.h>
main(){
    char key[256];
    char salt[64];
    char passwd[256];
    printf("key:");
    scanf("%s",&key);
    printf("salt:");
    scanf("%s",&salt);
    sprintf(passwd,"passwd:%s\n",crypt(key,salt));
    printf(passwd);
}
[root@linux root]# gcc -o crypt -s crypt.c -lcrypt
[root@linux root]# ./crypt
key:chen
salt:salt
passwd:sa0hRW/W3DLvQ
[root@linux root]#
```

2. PHP crypt()

将字符串用 DES 编码加密。

语法: string crypt(string str, string [salt]);

返回值: 字符串

函数种类: 编码处理

内容说明

本函数将字符串用 UNIX 的标准加密 DES 模块加密。这是单向的加密函数，无法解密。欲比对字符串，将已加密的字符串的头二个字符放在 salt 的参数中，再比对加密后的字符串。

更详细的资料请参考 UNIX Manual (man) 中的 crypt。

在一些较新的 UNIX 版本中，除了 DES 之外还提供了其它的加密模块，如 MD5。甚至有些系统还用 MD5 取代 DES。在 salt 参数还有一些变化，端看传给 salt 参数的字符串长度而定:

* CRYPT_STD_DES - 标准的 DES 编码，输入 2 字符的 salt。
* CRYPT_EXT_DES - 延伸的 DES 编码，输入 9 字符的 salt。
* CRYPT_MD5 - MD5 编码，输入 12 字符加上 1 的 salt。
* CRYPT_BLOWFISH - 延伸的 DES 编码，输入 16 字符加上 2 的 salt。

此外，若不使用 salt 参数，则程序会自动产生。

```
# cat crypt.php

<html>

<p>DES 密码</p>

<form method=post action=crypt.php>

<p>password:<input name=passwd type=text size=20></p>

<input type=submit value=submit>

</form>

<?

$enpw=crypt($passwd);

echo "password is: $enpw";

?>



[root@linux root]# wget http://netkiller.hikz.com/linux/download/myphp/site-
2.1.0.tar.gz
[root@linux root]#tar zxvf site-2.1.0.tar.gz
[root@linux root]#cp -r site /usr/local/apache/htdocs
[root@linux root]#lynx http://localhost/site
```

3. perl crypt

```
perl -e 'print("userPassword: ".crypt("secret","salt")."\n");'
```

4. mysql crypt

```
select encrypt('password');


mysql> select encrypt('password');
+---------------------+
| encrypt('password') |
+---------------------+
| WXvvG0CWY7v5I       |
+---------------------+
1 row in set (0.00 sec)


mysql>
```

5. Java crypt

第一种方法：

Crypt.java

Import netkiller. Security;

Crypt pw = new Crypt();

String passwd = pw.crypt(“passwd”,” salt”);

System.out.println(passwd);

关于JAVA的Crypt包请与我联系

第二种方法：

使用PostgreSQL JDBC中提供的org.postgresql.util.UnixCrypt产生crypt。

Class postgresql.util.UnixCrypt

java.lang.Object

|

+----postgresql.util.UnixCrypt

公共类 UnixCrypt 扩展 Object

这个类为我们提供了在通过网络流传输口令时的加密的功能

包含静态方法用于加密口令和与 Unix 加密的口令比较.

参阅 John Dumas 的 Java Crypt (加密)页面获取原始代码.

http://www.zeh.com/local/jfd/crypt.html

方法

public static final String crypt(String salt, String original)

加密给出了明文口令和一个"种子"("salt") 的口令.

参数:

salt - 一个两字符字串代表的所用的种子，用以向加密引擎说明加密的不同方式．如果你要生成一个新的密文那么这个值应该是随机生成的.

original - 待加密口令.

返回:

一个字串, 先是 2 字符的种子, 然后跟着密文口令.

方法：

1. 安装PostgreSQL JDBC，请到http://www.postgresql.org 下载
2. 将JDBC的.jar文件加到JAVA 的CLASSPATH中
3. 新建JAVA文件。
4. 编译javac crypt.java
5. 运行JAVA CLASS文件 java your-package.your-class
java crypt

```
import org.postgresql.util.UnixCrypt;
import java.io.InputStreamReader;
import java.io.BufferedReader;
import java.io.IOException;

public class crypt {
  public static void main(String[] args) throws IOException {
    String password;
    BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
    System.out.println("Enter the password to encrypt. Your password"+
                " will be echoed on the screen,");
    System.out.println("please ensure nobody is looking.");
    System.out.print("password :>");
    password=br.readLine();
    System.out.println(UnixCrypt.crypt(password));
  };
};
```

Part II. Message Digest

Table of Contents

7. MD5专题

1. md5sum
2. PHP md5()
3. MySQL md5()
4. Java MD5
5. perl md5

8. SHA 专题

1. sha1sum
2. PHP sha1()
3. Java SHA
4. Perl

9. CRC32

1. PHP CRC32

10. 第三方工具

1. htpasswd

1.1. CRYPT
1.2. MD5
1.3. SHA

2. htdigest
3. md5sum
4. sha1sum

Chapter 7. MD5专题

Table of Contents

1. md5sum
2. PHP md5()
3. MySQL md5()
4. Java MD5
5. perl md5

1. md5sum

MD5 为当前常用的 hash function,一般用来计算资料的杂凑值,俾利资料正确性之验证; md5sum 则为用来检查计算hash function 的的工具程序，具体的参数用法可去man md5sum 的用法。

生成杂凑值,有些文章叫指纹

md5sum file.txt

```
C:\GnuWin32\neo>md5sum file.txt
7012acbb1d394b20567dffbf0992b677 *file.txt

C:\GnuWin32\neo>md5sum file.txt > file.txt.md5

C:\GnuWin32\neo>md5sum -c file.txt.md5
file.txt: OK
```

生成指纹并重订向到文件

md5sum file.txt > file.txt.md5

```
C:\GnuWin32\neo>md5sum file.txt
7012acbb1d394b20567dffbf0992b677 *file.txt

C:\GnuWin32\neo>md5sum file.txt > file.txt.md5

C:\GnuWin32\neo>md5sum -c file.txt.md5
file.txt: OK
```

生成一组文件

```
md5sum file0.txt > file.txt.md5
md5sum file1.txt >> file.txt.md5
md5sum file2.txt >> file.txt.md5
```

使用通配符

```
C:\GnuWin32\neo>md5sum *
7012acbb1d394b20567dffbf0992b677 *file.txt
d9226d4bd8779baa69db272f89a2e05c *message.txt

C:\GnuWin32\neo>md5sum * >file.txt.md5
```

验证文件是否被人更改过

md5sum -c file.txt.md5

```
C:\GnuWin32\neo>md5sum file.txt
7012acbb1d394b20567dffbf0992b677 *file.txt

C:\GnuWin32\neo>md5sum file.txt > file.txt.md5

C:\GnuWin32\neo>md5sum -c file.txt.md5
file.txt: OK
```

2. PHP md5()

```
# cat md5.php
<html>
<p>MD5 密码产生器</p>
<form method=post action=des.php>
<p>password:<input name=passwd type=text size=20></p>
<input type=submit value=submit>
</form>
<?
$enpw=md5($passwd);
echo "password is: $enpw";
?>
```

3. MySQL md5()

select md5('password');

```
[chen@linux chen]$ mysql

Welcome to the MySQL monitor.  Commands end with ; or \g.

Your MySQL connection id is 11947 to server version: 4.0.13-log


Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> select md5('chen');

+----------------------------------+
| md5('chen')                      |
+----------------------------------+
| a1a8887793acfc199182a649e905daab |
+----------------------------------+

1 row in set (0.00 sec)

mysql>


mysql> select md5('chen') as passwd;

+----------------------------------+
| passwd                           |
+----------------------------------+
| a1a8887793acfc199182a649e905daab |
+----------------------------------+

1 row in set (0.00 sec)

mysql>
```

4. Java MD5

1.2版之前的JDK没有实现md5;

```
/*********************************************
MD5 算法的Java Bean
@author:Topcat Tuppin
Last Modified:10,Mar,2001
*********************************************/
package netkiller.security;
import java.lang.reflect.*;
/*********************************************
md5 类实现了RSA Data Security, Inc.在提交给IETF
的RFC1321中的MD5 message-digest 算法。
*********************************************/

public class MD5 {
       /* 下面这些S11-S44实际上是一个4*4的矩阵，在原始的C实现中是用#define 实现的，
       这里把它们实现成为static final是表示了只读，切能在同一个进程空间内的多个
       Instance间共享*/

        static final int S11 = 7;
        static final int S12 = 12;
        static final int S13 = 17;
        static final int S14 = 22;

        static final int S21 = 5;
        static final int S22 = 9;
        static final int S23 = 14;
        static final int S24 = 20;

        static final int S31 = 4;
        static final int S32 = 11;
        static final int S33 = 16;
        static final int S34 = 23;

        static final int S41 = 6;
        static final int S42 = 10;
        static final int S43 = 15;
        static final int S44 = 21;



        static final byte[] PADDING = { -128, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
        0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 };
        /* 下面的三个成员是MD5计算过程中用到的3个核心数据，在原始的C实现中
           被定义到MD5_CTX结构中
         */

        private long[] state = new long[4];  // state (ABCD)
        private long[] count = new long[2];  // number of bits, modulo 2^64 (lsb
first)
        private byte[] buffer = new byte[64]; // input buffer

       /* digestHexStr是MD5的唯一一个公共成员，是最新一次计算结果的
          16进制ASCII表示.
       */

        public String digestHexStr;

        /* digest,是最新一次计算结果的2进制内部表示，表示128bit的MD5值. */

        private byte[] digest = new byte[16];

       /*
          getMD5ofStr是类MD5最主要的公共方法，入口参数是你想要进行MD5变换的字符串
          返回的是变换完的结果，这个结果是从公共成员digestHexStr取得的.
```

```
*/
 public String getMD5ofStr(String inbuf) {
         md5Init();
         md5Update(inbuf.getBytes(), inbuf.length());
         md5Final();
         digestHexStr = "";
         for (int i = 0; i < 16; i++) {
                 digestHexStr += byteHEX(digest[i]);
         }
         return digestHexStr;

 }
 // 这是MD5这个类的标准构造函数，JavaBean要求有一个public的并且没有参数的构造函数
 public MD5() {
         md5Init();
         return;
 }
 /* md5Init是一个初始化函数，初始化核心变量，装入标准的幻数 */
 private void md5Init() {
         count[0] = 0L;
         count[1] = 0L;
         ///* Load magic initialization constants.

         state[0] = 0x67452301L;
         state[1] = 0xefcdab89L;
         state[2] = 0x98badcfeL;
         state[3] = 0x10325476L;

         return;
 }
 /* F, G, H ,I 是4个基本的MD5函数，在原始的MD5的C实现中，由于它们是
 简单的位运算，可能出于效率的考虑把它们实现成了宏，在java中，我们把它们
实现成了private方法，名字保持了原来C中的。 */

 private long F(long x, long y, long z) {
         return (x & y) | ((~x) & z);

 }
 private long G(long x, long y, long z) {
         return (x & z) | (y & (~z));

 }
 private long H(long x, long y, long z) {
         return x ^ y ^ z;
```

```
    }

    private long I(long x, long y, long z) {
            return y ^ (x | (~z));
    }

   /*
      FF,GG,HH和II将调用F,G,H,I进行近一步变换
      FF, GG, HH, and II transformations for rounds 1, 2, 3, and 4.
      Rotation is separate from addition to prevent recomputation.
   */

    private long FF(long a, long b, long c, long d, long x, long s,
            long ac) {
            a += F (b, c, d) + x + ac;
            a = ((int) a << s) | ((int) a >>> (32 - s));
            a += b;
            return a;
    }

    private long GG(long a, long b, long c, long d, long x, long s,
            long ac) {
            a += G (b, c, d) + x + ac;
            a = ((int) a << s) | ((int) a >>> (32 - s));
            a += b;
            return a;
    }
    private long HH(long a, long b, long c, long d, long x, long s,
            long ac) {
            a += H (b, c, d) + x + ac;
            a = ((int) a << s) | ((int) a >>> (32 - s));
            a += b;
            return a;
    }
    private long II(long a, long b, long c, long d, long x, long s,
            long ac) {
            a += I (b, c, d) + x + ac;
            a = ((int) a << s) | ((int) a >>> (32 - s));
            a += b;
            return a;
    }
    /*
```

```
 md5Update是MD5的主计算过程，inbuf是要变换的字节串，inputlen是长度，这个
 函数由getMD5ofStr调用，调用之前需要调用md5init，因此把它设计成private的
*/

private void md5Update(byte[] inbuf, int inputLen) {

        int i, index, partLen;

        byte[] block = new byte[64];

        index = (int)(count[0] >>> 3) & 0x3F;

        // /* Update number of bits */

        if ((count[0] += (inputLen << 3)) < (inputLen << 3))

                count[1]++;

        count[1] += (inputLen >>> 29);

        partLen = 64 - index;

        // Transform as many times as possible.

        if (inputLen >= partLen) {

                md5Memcpy(buffer, inbuf, index, 0, partLen);

                md5Transform(buffer);

                for (i = partLen; i + 63 < inputLen; i += 64) {

                        md5Memcpy(block, inbuf, 0, i, 64);

                        md5Transform (block);

                }

                index = 0;

        } else

                i = 0;


        ///* Buffer remaining input */

        md5Memcpy(buffer, inbuf, index, i, inputLen - i);

}

/*
  md5Final整理和填写输出结果
*/

private void md5Final () {

        byte[] bits = new byte[8];

        int index, padLen;

        ///* Save number of bits */

        Encode (bits, count, 8);


        ///* Pad out to 56 mod 64.

        index = (int)(count[0] >>> 3) & 0x3f;

        padLen = (index < 56) ? (56 - index) : (120 - index);

        md5Update (PADDING, padLen);

        ///* Append length (before padding) */

        md5Update(bits, 8);

        ///* Store state in digest */

        Encode (digest, state, 16);
```

```
    }
    /* md5Memcpy是一个内部使用的byte数组的块拷贝函数，从input的inpos开始把len长度的
字节拷贝到output的outpos位置开始
    */
    private void md5Memcpy (byte[] output, byte[] input,
            int outpos, int inpos, int len)
    {
            int i;

            for (i = 0; i < len; i++)
                    output[outpos + i] = input[inpos + i];
    }
    /*
       md5Transform是MD5核心变换程序，有md5Update调用，block是分块的原始字节
    */
    private void md5Transform (byte block[]) {
            long a = state[0], b = state[1], c = state[2], d = state[3];
            long[] x = new long[16];

            Decode (x, block, 64);

            /* Round 1 */
            a = FF (a, b, c, d, x[0], S11, 0xd76aa478L); /* 1 */
            d = FF (d, a, b, c, x[1], S12, 0xe8c7b756L); /* 2 */
            c = FF (c, d, a, b, x[2], S13, 0x242070dbL); /* 3 */
            b = FF (b, c, d, a, x[3], S14, 0xc1bdceeeL); /* 4 */
            a = FF (a, b, c, d, x[4], S11, 0xf57c0fafL); /* 5 */
            d = FF (d, a, b, c, x[5], S12, 0x4787c62aL); /* 6 */
            c = FF (c, d, a, b, x[6], S13, 0xa8304613L); /* 7 */
            b = FF (b, c, d, a, x[7], S14, 0xfd469501L); /* 8 */
            a = FF (a, b, c, d, x[8], S11, 0x698098d8L); /* 9 */
            d = FF (d, a, b, c, x[9], S12, 0x8b44f7afL); /* 10 */
            c = FF (c, d, a, b, x[10], S13, 0xffff5bb1L); /* 11 */
            b = FF (b, c, d, a, x[11], S14, 0x895cd7beL); /* 12 */
            a = FF (a, b, c, d, x[12], S11, 0x6b901122L); /* 13 */
            d = FF (d, a, b, c, x[13], S12, 0xfd987193L); /* 14 */
            c = FF (c, d, a, b, x[14], S13, 0xa679438eL); /* 15 */
            b = FF (b, c, d, a, x[15], S14, 0x49b40821L); /* 16 */

            /* Round 2 */
            a = GG (a, b, c, d, x[1], S21, 0xf61e2562L); /* 17 */
            d = GG (d, a, b, c, x[6], S22, 0xc040b340L); /* 18 */
            c = GG (c, d, a, b, x[11], S23, 0x265e5a51L); /* 19 */
            b = GG (b, c, d, a, x[0], S24, 0xe9b6c7aaL); /* 20 */
```

```
    a = GG (a, b, c, d, x[5], S21, 0xd62f105dL); /* 21 */
    d = GG (d, a, b, c, x[10], S22, 0x2441453L); /* 22 */
    c = GG (c, d, a, b, x[15], S23, 0xd8a1e681L); /* 23 */
    b = GG (b, c, d, a, x[4], S24, 0xe7d3fbc8L); /* 24 */
    a = GG (a, b, c, d, x[9], S21, 0x21e1cde6L); /* 25 */
    d = GG (d, a, b, c, x[14], S22, 0xc33707d6L); /* 26 */
    c = GG (c, d, a, b, x[3], S23, 0xf4d50d87L); /* 27 */
    b = GG (b, c, d, a, x[8], S24, 0x455a14edL); /* 28 */
    a = GG (a, b, c, d, x[13], S21, 0xa9e3e905L); /* 29 */
    d = GG (d, a, b, c, x[2], S22, 0xfcefa3f8L); /* 30 */
    c = GG (c, d, a, b, x[7], S23, 0x676f02d9L); /* 31 */
    b = GG (b, c, d, a, x[12], S24, 0x8d2a4c8aL); /* 32 */

    /* Round 3 */
    a = HH (a, b, c, d, x[5], S31, 0xfffa3942L); /* 33 */
    d = HH (d, a, b, c, x[8], S32, 0x8771f681L); /* 34 */
    c = HH (c, d, a, b, x[11], S33, 0x6d9d6122L); /* 35 */
    b = HH (b, c, d, a, x[14], S34, 0xfde5380cL); /* 36 */
    a = HH (a, b, c, d, x[1], S31, 0xa4beea44L); /* 37 */
    d = HH (d, a, b, c, x[4], S32, 0x4bdecfa9L); /* 38 */
    c = HH (c, d, a, b, x[7], S33, 0xf6bb4b60L); /* 39 */
    b = HH (b, c, d, a, x[10], S34, 0xbebfbc70L); /* 40 */
    a = HH (a, b, c, d, x[13], S31, 0x289b7ec6L); /* 41 */
    d = HH (d, a, b, c, x[0], S32, 0xeaa127faL); /* 42 */
    c = HH (c, d, a, b, x[3], S33, 0xd4ef3085L); /* 43 */
    b = HH (b, c, d, a, x[6], S34, 0x4881d05L); /* 44 */
    a = HH (a, b, c, d, x[9], S31, 0xd9d4d039L); /* 45 */
    d = HH (d, a, b, c, x[12], S32, 0xe6db99e5L); /* 46 */
    c = HH (c, d, a, b, x[15], S33, 0x1fa27cf8L); /* 47 */
    b = HH (b, c, d, a, x[2], S34, 0xc4ac5665L); /* 48 */

    /* Round 4 */
    a = II (a, b, c, d, x[0], S41, 0xf4292244L); /* 49 */
    d = II (d, a, b, c, x[7], S42, 0x432aff97L); /* 50 */
    c = II (c, d, a, b, x[14], S43, 0xab9423a7L); /* 51 */
    b = II (b, c, d, a, x[5], S44, 0xfc93a039L); /* 52 */
    a = II (a, b, c, d, x[12], S41, 0x655b59c3L); /* 53 */
    d = II (d, a, b, c, x[3], S42, 0x8f0ccc92L); /* 54 */
    c = II (c, d, a, b, x[10], S43, 0xffeff47dL); /* 55 */
    b = II (b, c, d, a, x[1], S44, 0x85845dd1L); /* 56 */
    a = II (a, b, c, d, x[8], S41, 0x6fa87e4fL); /* 57 */
    d = II (d, a, b, c, x[15], S42, 0xfe2ce6e0L); /* 58 */
```

```
                c = II (c, d, a, b, x[6], S43, 0xa3014314L); /* 59 */
                b = II (b, c, d, a, x[13], S44, 0x4e0811a1L); /* 60 */
                a = II (a, b, c, d, x[4], S41, 0xf7537e82L); /* 61 */
                d = II (d, a, b, c, x[11], S42, 0xbd3af235L); /* 62 */
                c = II (c, d, a, b, x[2], S43, 0x2ad7d2bbL); /* 63 */
                b = II (b, c, d, a, x[9], S44, 0xeb86d391L); /* 64 */

                state[0] += a;
                state[1] += b;
                state[2] += c;
                state[3] += d;

        }
        /*Encode把long数组按顺序拆成byte数组，因为java的long类型是64bit的，
          只拆低32bit，以适应原始C实现的用途
        */
        private void Encode (byte[] output, long[] input, int len) {
                int i, j;
                for (i = 0, j = 0; j < len; i++, j += 4) {
                        output[j] = (byte)(input[i] & 0xffL);
                        output[j + 1] = (byte)((input[i] >>> 8) & 0xffL);
                        output[j + 2] = (byte)((input[i] >>> 16) & 0xffL);
                        output[j + 3] = (byte)((input[i] >>> 24) & 0xffL);
                }
        }

        /*Decode把byte数组按顺序合成成long数组，因为java的long类型是64bit的，
          只合成低32bit，高32bit清零，以适应原始C实现的用途
        */
        private void Decode (long[] output, byte[] input, int len) {
                int i, j;
                for (i = 0, j = 0; j < len; i++, j += 4)
                        output[i] = b2iu(input[j]) |
                                (b2iu(input[j + 1]) << 8) |
                                (b2iu(input[j + 2]) << 16) |
                                (b2iu(input[j + 3]) << 24);

                return;
        }
        /*
          b2iu是我写的一个把byte按照不考虑正负号的原则的" 升位" 程序，因为java没有unsigned运
算
        */
        public static long b2iu(byte b) {
```

```
                return b < 0 ? b & 0x7F + 128 : b;
        }
       /*byteHEX(), 用来把一个byte类型的数转换成十六进制的ASCII表示,
        因为java中的byte的toString无法实现这一点, 我们又没有C语言中的
         sprintf(outbuf,"%02X",ib)
       */
        public static String byteHEX(byte ib) {
                char[] Digit = { '0','1','2','3','4','5','6','7','8','9',
                'A','B','C','D','E','F' };
                char [] ob = new char[2];
                ob[0] = Digit[(ib >>> 4) & 0X0F];
                ob[1] = Digit[ib & 0X0F];
                String s = new String(ob);
                return s;
        }

        public String getMD5String(String md5){
                        return getMD5ofStr(md5).toLowerCase();
        }
        public static void main(String args[]) {
                MD5 m = new MD5();
                if (Array.getLength(args) == 0) {   //如果没有参数, 执行标准的Test
Suite

                    System.out.println("MD5 Test suite:");
                       System.out.println("MD5(\"\"):"+m.getMD5ofStr(""));
                       System.out.println("MD5(\"a\"):"+m.getMD5ofStr("a"));
                       System.out.println("MD5(\"abc\"):"+m.getMD5ofStr("abc"));
                       System.out.println("MD5(\"message
digest\"):"+m.getMD5ofStr("message digest"));
                       System.out.println("MD5(\"abcdefghijklmnopqrstuvwxyz\"):"+
                        m.getMD5ofStr("abcdefghijklmnopqrstuvwxyz"));

System.out.println("MD5(\"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789\"):

m.getMD5ofStr("ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789"));
                }
                else
                           System.out.println("MD5(" + args[0] + ")=" +
m.getMD5ofStr(args[0]));


        }

}
```

以下使用JDK 1.5.x版实现;

```
import java.security.*;

public class md5Test {

        private static String dumpBytes(byte[] bytes) {
            int i;
            StringBuffer sb = new StringBuffer();
            for (i = 0; i < bytes.length; i++) {
              if (i % 32 == 0 && i != 0) {
                sb.append("\n");
              }
              String s = Integer.toHexString(bytes[i]);
              if (s.length() < 2) {
                s = "0" + s;
              }
              if (s.length() > 2) {
                s = s.substring(s.length() - 2);
              }
              sb.append(s);
            }
            return sb.toString();
        }
        public static void main(String[] args) {

                String passwd = "netkiller";
            MessageDigest md = null;
                try {
                        md = MessageDigest.getInstance("MD5");
                        md.update("chen".getBytes());
                } catch (NoSuchAlgorithmException e) {
                        e.printStackTrace();
                }

            System.out.println(dumpBytes(md.digest()));
        }

}
```

编译运行,输入字符串:a1a8887793acfc199182a649e905daab

5. perl md5

```
# Functional style
use Digest::MD5 qw(md5 md5_hex md5_base64);

$digest = md5($data);
$digest = md5_hex($data);
$digest = md5_base64($data);

# OO style
use Digest::MD5;

$ctx = Digest::MD5->new;

$ctx->add($data);
$ctx->addfile(*FILE);

$digest = $ctx->digest;
$digest = $ctx->hexdigest;
$digest = $ctx->b64digest;
```

Chapter 8. SHA 专题

Table of Contents

1. sha1sum
2. PHP sha1()
3. Java SHA
4. Perl

1. sha1sum

```
$ sha1sum /etc/passwd
c144c5cc8d5d3b90ad74a1dcf6af9e6c935e2a2a  /etc/passwd

$ sha1sum about/*
905a26de0f2fd6fcb53bf8db75d76c538d094237  about/index.html
d0aeb4409808b6afded0522964bed6b795d30fc0  about/index.tpl

$ sha1sum about/* > about.sha1
$ cat about.sha1
905a26de0f2fd6fcb53bf8db75d76c538d094237  about/index.html
d0aeb4409808b6afded0522964bed6b795d30fc0  about/index.tpl

$ sha1sum -c about.sha1
about/index.html: OK
about/index.tpl: OK
```

2. PHP sha1()

string sha1 (string str [, bool raw_output])

```
<?php
        $str = 'netkiller';
        echo sha1($str);
?>
```

运行输出字符串:eb673aa189c814d2db9fb71f162da1c81b4eba1c

3. Java SHA

```
import java.security.*;

public class shaTest {

        private static String dumpBytes(byte[] bytes) {
            int i;
            StringBuffer sb = new StringBuffer();
            for (i = 0; i < bytes.length; i++) {
              if (i % 32 == 0 && i != 0) {
                sb.append("\n");
              }
              String s = Integer.toHexString(bytes[i]);
              if (s.length() < 2) {
                s = "0" + s;
              }
              if (s.length() > 2) {
                s = s.substring(s.length() - 2);
              }
              sb.append(s);
            }
            return sb.toString();
        }
        public static void main(String[] args) {

                String passwd = "netkiller";
            MessageDigest md = null;
                try {
                        md = MessageDigest.getInstance("SHA");
                        md.update("chen".getBytes());
                } catch (NoSuchAlgorithmException e) {
                        e.printStackTrace();
                }

            System.out.println(dumpBytes(md.digest()));
        }

}
```

编译运行,输入字符串:8a89798cf0878e37bb6589ae1c36b9d8a036275b

4. Perl

```
# Functional style
use Digest::SHA1  qw(sha1 sha1_hex sha1_base64);

$digest = sha1($data);
$digest = sha1_hex($data);
$digest = sha1_base64($data);

# OO style
use Digest::SHA1;

$ctx = Digest::SHA1->new;

$ctx->add($data);
$ctx->addfile(*FILE);

$digest = $ctx->digest;
$digest = $ctx->hexdigest;
$digest = $ctx->b64digest;
```

Chapter 9. CRC32

Table of Contents

1. PHP CRC32

CRC校验实用程序库在数据存储和数据通讯领域，为了保证数据的正确，就不得不采用检错的手段。在诸多检错手段中，CRC是最著名的一种。CRC的全称是循环冗余校验，其特点是:检错能力极强，开销小，易于用编码器及检测电路实现。从其检错能力来看，它所不能发现的错误的几率仅为0.0047%以下。从性能上和开销上考虑，均远远优于奇偶校验及算术和校验等方式。因而，在数据存储和数据通讯领域，CRC无处不在:著名的通讯协议X.25的FCS(帧检错序列)采用的是CRC- CCITT，ARJ、LHA等压缩工具软件采用的是CRC32，磁盘驱动器的读写采用了CRC16，通用的图像存储格式GIF、TIFF等也都用CRC作为检错手段。

1. PHP CRC32

```
<?php
$checksum = crc32("The quick brown fox jumped over the lazy dog.");
printf("%u\n", $checksum);
?>
```

Chapter 10. 第三方工具

Table of Contents

1. htpasswd

1.1. CRYPT
1.2. MD5
1.3. SHA

2. htdigest
3. md5sum
4. sha1sum

1. htpasswd

```
$ sudo apt-get install apache2-utils
```

1.1. CRYPT

```
neo@master:~$ htpasswd -d -n neo.chen
New password:
Re-type new password:
neo.chen:Tyr60pyBFo0ng
```

1.2. MD5

```
neo@master:~$ htpasswd -m -n neo.chen
New password:
Re-type new password:
neo.chen:$apr1$CbZkN...$QzT7LwjRpQCKr4IkryM3Z.
```

1.3. SHA

```
neo@master:~$ htpasswd -s -n neo.chen
New password:
Re-type new password:
neo.chen:{SHA}iol5jPCHjje7ZYmuHDa52KA2J1s=
```

2. htdigest

htdigest 與 htpasswd 不同的地方在於對密碼的加密方式，htdigest 是使用 md5 來加密而 htpasswd 則是使用 crypt 來加密

```
htdigest -c /home/neo/trac/conf/passwd.digest localhost netkiller
htdigest /home/neo/trac/conf/passwd.digest localhost neo
```

3. md5sum

```
$ md5sum /etc/passwd
325b7229c82c90c8a1823f5d939156bc  /etc/passwd
```

4. sha1sum

```
$ sha1sum /etc/passwd
f7a5582dd42ce0411bc2c59e2f1d8e89adcf0f81  /etc/passwd
```

Chapter 11. OpenPGP/OpenGPG(GnuPG)

Table of Contents

1. GnuPG(OpenGPG)

1.1. 生成密钥对
1.2. 列出密钥
1.3. 验证签字
1.4. GnuPG For Windows
1.5. EMail-Security
1.6. Smart Card

下载PGP: http://www.pgp.com/

下载OpenPGP: http://www.pgpi.org/

1. GnuPG(OpenGPG)

下载OpenGPG: http://www.gnupg.org/

Note

GnuPG (OpenGPG)安装时可以选择语言，支持简体中文．但对中文支持不是很好，如真实姓名输入：王老五,系统提示"姓名至少要有五个字符长"

1.1. 生成密钥对

使用 gpg --gen-key 生成密钥对

```
C:\GNU>gpg --gen-key
gpg (GnuPG) 1.4.3; Copyright (C) 2006 Free Software Foundation, Inc.
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it
under certain conditions. See the file COPYING for details.

请选择您要使用的密钥种类:
   (1) DSA 和 ElGamal (默认)
   (2) DSA (仅用于签字)
   (5) RSA (仅用于签字)
您的选择?
DSA 密钥对会有 1024 位。
ELG-E 密钥长度应在 1024 位与 4096 位之间。
您想要用多大的密钥尺寸? (2048)
您所要求的密钥尺寸是 2048 位
请设定这把密钥的有效期限。
         0 = 密钥永不过期
      <n>  = 密钥在 n 天后过期
      <n>w = 密钥在 n 周后过期
      <n>m = 密钥在 n 月后过期
      <n>y = 密钥在 n 年后过期
密钥的有效期限是? (0)
密钥永远不会过期
以上正确吗? (y/n)y

您需要一个用户标识来辨识您的密钥；本软件会用真实姓名、注释和电子邮件地址组合
成用户标识，如下所示:
    “Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de>”

真实姓名: neo chen
电子邮件地址: openunix@163.com
注释: netkiller
```

```
您选定了这个用户标识:
    “neo chen (netkiller) <openunix@163.com>”

更改姓名(N)、注释(C)、电子邮件地址(E)或确定(O)/退出(Q)? O
您需要一个密码来保护您的私钥。

我们需要生成大量的随机字节。这个时候您可以多做些琐事(像是敲打键盘、移动
鼠标、读写硬盘之类的),这会让随机数字发生器有更好的机会获得足够的熵数。
+++++++++++++++++++++++++++++++++.+++++++++++++++++++++++++++++++++.++++++++
++++++++++++.++++++++++++++++++++.++++++++++++++++++++++++++...>+++++.....
..+++++
我们需要生成大量的随机字节。这个时候您可以多做些琐事(像是敲打键盘、移动
鼠标、读写硬盘之类的),这会让随机数字发生器有更好的机会获得足够的熵数。
+++++.++++++++++++++++..++++++++++.+++++.++++++++++.++++++++++.+++++.+++++++
++++++++.+++++....+++++..++++++++++.++++++++++++++++++..++++++++++++++++++.+
++++>++++++++++...>.+++++.............................>+++++<+++++.............
.............+++++^^^^^
gpg: 密钥 C9441A1A 被标记为绝对信任
公钥和私钥已经生成并经签字。

gpg: 正在检查信任度数据库
gpg: 需要 3 份勉强信任和 1 份完全信任, PGP 信任模型
gpg: 深度: 0 有效性:   2 已签字:   0 信任度: 0-, 0q, 0n, 0m, 0f, 2u
pub   1024D/C9441A1A 2006-06-02
密钥指纹 = EFDC A97C C711 E9C9 FAC1  3EA9 33C1 1FB2 C944 1A1A
uid                  neo chen (netkiller) <openunix@163.com>
sub   2048g/B713326C 2006-06-02

C:\GNU>
```

1.2. 列出密钥

列出密钥使用 gpg --list-keys

```
C:\GNU>gpg --list-keys
C:/Documents and Settings/neo.chen/Application Data/gnupg\pubring.gpg
------------------------------------------------------------------------
pub   1024D/C9441A1A 2006-06-02
uid                  neo chen (netkiller) <openunix@163.com>
sub   2048g/B713326C 2006-06-02
```

列出密钥和签字使用 gpg --list-keys

```
C:\GNU>gpg --list-sigs
C:/Documents and Settings/neo.chen/Application Data/gnupg\pubring.gpg
------------------------------------------------------------------------
pub   1024D/C9441A1A 2006-06-02
uid                  neo chen (netkiller) <openunix@163.com>
sig 3        C9441A1A 2006-06-02  neo chen (netkiller) <openunix@163.com>
sub   2048g/B713326C 2006-06-02
sig          C9441A1A 2006-06-02  neo chen (netkiller) <openunix@163.com>
```

列出并检查密钥签字 gpg --check-sigs

```
C:\GNU>gpg --check-sigs
C:/Documents and Settings/neo.chen/Application Data/gnupg\pubring.gpg
------------------------------------------------------------------------
pub   1024D/C9441A1A 2006-06-02
uid                  neo chen (netkiller) <openunix@163.com>
sig!3        C9441A1A 2006-06-02  neo chen (netkiller) <openunix@163.com>
sub   2048g/B713326C 2006-06-02
sig!         C9441A1A 2006-06-02  neo chen (netkiller) <openunix@163.com>
```

1.3. 验证签字

检查 PGP 签名 与 md5sum 作用类似:

```
bash$ gpg --verify gnupg-x.x.x.tar.gz.sig gnupg-x.x.x.tar.gz
bash$ md5sum gnupg-x.x.x.tar.gz
```

1.4. GnuPG For Windows

GnuPG

```
```

1.5. EMail-Security

EMail-Security using GnuPG for Windows

gpg4win

1.6. Smart Card

http://www.gnupg.org/howtos/card-howto/en/smartcard-howto-single.html

---

Chapter 12. Secure Tunnel

Table of Contents

1. OpenSSH Tunnel

1.1. SOCKS v5 Tunnel

2. SSL Tunnel

2.1. 通过SSL访问POP、IMAP、SMTP

1. OpenSSH Tunnel

mysql tunnel

```
$ ssh -L 3306:127.0.0.1:3306 user@example.org
```

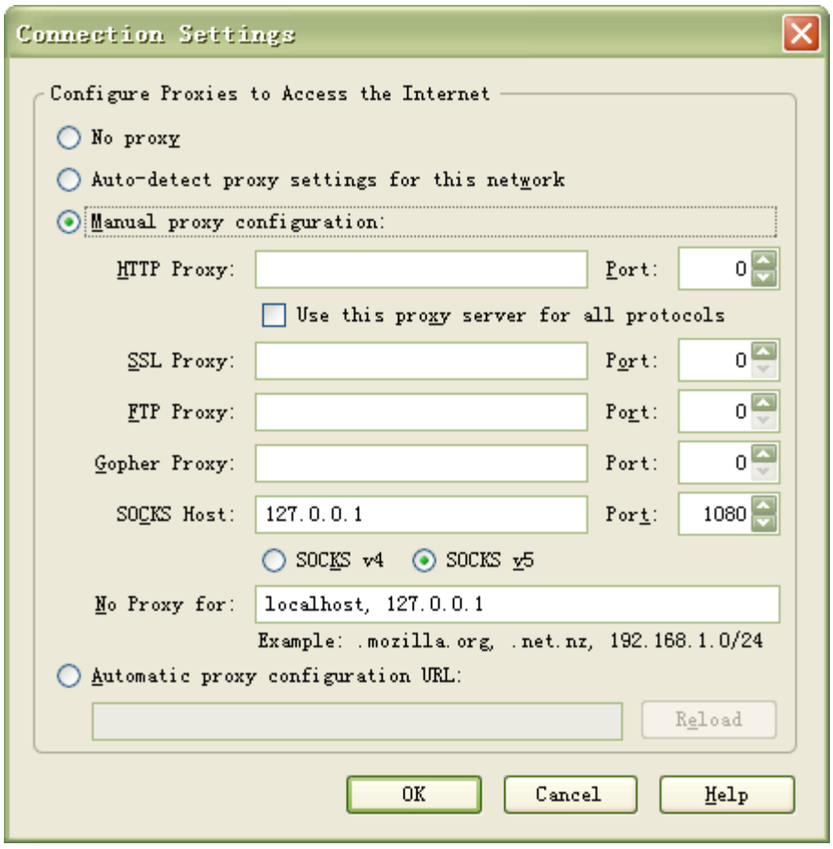
testing

```
$ mysql -h 127.0.0.1 -uroot -p test
```

1.1. SOCKS v5 Tunnel

```
ssh -D 1080 <远程主机地址>
```

Firefox 配置



为了防止所访问网站的DNS被窥探，可以在Firefox的地址栏中输入about:config 把 network.proxy.socks_remote_dns 改为true

---

2. SSL Tunnel

http://www.stunnel.org/

2.1. 通过SSL访问POP、IMAP、SMTP

Example 12.1. stunnel.conf

```
# Sample stunnel configuration file
# Copyright by Michal Trojnara 2002

# Comment it out on Win32
cert = /etc/stunnel/stunnel.pem
# chroot = /usr/var/run/stunnel/
# PID is created inside chroot jail
pid = /stunnel.pid
#setuid = nobody
#setgid = nogroup

setuid = root
setgid = root

# Workaround for Eudora bug
#options = DONT_INSERT_EMPTY_FRAGMENTS

# Authentication stuff
#verify = 2
# don't forget about c_rehash CApath
# it is located inside chroot jail:
#CApath = /certs
# or simply use CAfile instead:
#CAfile = /usr/etc/stunnel/certs.pem

# Some debugging stuff
debug = 7
output = stunnel.log

# Use it for client mode
#client = yes

# Service-level configuration

[pop3s]
accept  = 995
connect = 110

[imaps]
accept  = 993
connect = 143

[ssmtp]
accept  = 465
connect = 25

#[https]
#accept  = 443
#connect = 80
#TIMEOUTclose = 0

[nntps]
accept  = 563
connect = 119
```

```
# SMTP
/sbin/iptables -A INPUT -p tcp --dport 25 -j ACCEPT
# SMTPS
/sbin/iptables -A INPUT -p tcp --dport 465 -j ACCEPT
# POP3
/sbin/iptables -A INPUT -p tcp --dport 110 -j ACCEPT
# POP3S
/sbin/iptables -A INPUT -p tcp --dport 995 -j ACCEPT
# IMAP
/sbin/iptables -A INPUT -p tcp --dport 143 -j ACCEPT
# IMAPS
/sbin/iptables -A INPUT -p tcp --dport 993 -j ACCEPT
```

```
[root@linuxas3 stunnel]# nmap localhost

Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Interesting ports on linuxas3.9812.net (127.0.0.1):
(The 1582 ports scanned but not shown below are in state: closed)
Port       State       Service
22/tcp     open        ssh
25/tcp     open        smtp
80/tcp     open        http
110/tcp    open        pop-3
111/tcp    open        sunrpc
119/tcp    open        nntp
143/tcp    open        imap2
443/tcp    open        https
465/tcp    open        smtps
563/tcp    open        snews
631/tcp    open        ipp
783/tcp    open        hp-alarm-mgr
993/tcp    open        imaps
995/tcp    open        pop3s
3306/tcp   open        mysql
5000/tcp   open        UPnP
5001/tcp   open        commplex-link
8009/tcp   open        ajp13
8080/tcp   open        http-proxy

Nmap run completed -- 1 IP address (1 host up) scanned in 3 seconds
[root@linuxas3 stunnel]#
```

Chapter 13. 硬盘分区与文件系统加密

Harddisk Partition & File System

Table of Contents

1. Linux磁盘分区加密
2. Microsoft EFS

1. Linux磁盘分区加密

Procedure 13.1. cryptsetup - configures encrypted block devices

1. 安装 cryptsetup

```
# apt-get install cryptsetup dmsetup
```

2. 硬盘分区

添加一块新硬盘，使用cfdisk /dev/sdb 对他进行分区

```
sapnu-melencio:~# fdisk -l

Disk /dev/sda: 8589 MB, 8589934592 bytes
255 heads, 63 sectors/track, 1044 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0x0004287b

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1         993     7976241   83  Linux
/dev/sda2             994        1044      409657+   5  Extended
/dev/sda5             994        1044      409626   82  Linux swap / Solaris

Disk /dev/sdb: 4294 MB, 4294967296 bytes
255 heads, 63 sectors/track, 522 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
Disk identifier: 0x7256cdce

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1               1         522     4192933+  83  Linux
```

/dev/sdb1 就是我的分区

注意：分区操作要小心加小心，马虎不得，否则你将数据全失。

3. 创建加密分区

```
# cryptsetup --verbose --verify-passphrase -c aes-cbc-plain luksFormat
/dev/sdb1

WARNING!
========
This will overwrite data on /dev/sdb2 irrevocably.

Are you sure? (Type uppercase yes): YES (输入大写的YES来确定创建加密分区)
Enter LUKS passphrase:  (输入密码)
Verify passphrase:              (确认密码)
Command successful.
```

这将把不可逆转地改写/dev/sda2 上的数据。

注意：也要小心加小心，马虎不得，否则你将数据全失。一定不要搞错分区。

4. 挂载的逻辑分区

```
# cryptsetup luksOpen /dev/sdb1 sdb1
Enter LUKS passphrase:
key slot 0 unlocked.
Command successful.
```

如下命令将显示/dev/mapper路径中的隐藏设备

```
# ls -l /dev/mapper
```

5. 格式化加密分区

现在将该分区格式化为ext3文件系统.

```
 mkfs.ext3 /dev/mapper/sdb1
```

6. 挂载

接下来我们创建一个用于挂载的挂载点并挂载.

```
# mkdir /mnt/secret
# mount /dev/mapper/sdb1 /mnt/secret
```

7. 使用加密分区

好了,现在你可以使用你的加密分区了.

cd /mnt/secret

touch file

8. 卸载

使用完毕后为了保护数据的隐密,我们需要取消挂载并关闭加密分区.

```
# umount /mnt/secret
# cryptsetup luksClose sdb1
```

"Disconnect" ƆoǴ`

2. Microsoft EFS

http://www.microsoft.com/china/technet/security/sgk/protect_data_EFS.mspx

Chapter 14. Email Security using OpenPGP and S/MIME

Table of Contents

1. Gpg4win
2. S/MIME

1. Gpg4win

http://www.gpg4win.org/

2. S/MIME

数字签名、加密与证书颁发机构

Part III. 数字证书工具

Table of Contents

15. OpenSSL

1. 如何创建一个文件的 MD5 或 SHA1 摘要?
2. 编码/解码
3. web 服务器 ssl 证书
4. 去除私钥的密码
5. 证书转换

5.1. CA证书
5.2. 创建CA证书有效期为一年
5.3. x509转换为pfx
5.4. PEM格式的ca.key转换为Microsoft可以识别的pvk格式
5.5. PKCS#12 到 PEM 的转换
5.6. 从 PFX 格式文件中提取私钥格式文件 (.key)
5.7. 转换 pem 到到 spc
5.8. PEM 到 PKCS#12 的转换
5.9. How to Convert PFX Certificate to PEM Format for SOAP

16. Java - keytool

1. 创建证书
2. Private key generation
3. Public Key Certificate (optional)
4. import your signed certificate
5. Import the certificate and attach it to your server key pair
6. Key pair verification

17. .Net makecert

1. 访问X.509证书

Chapter 15. OpenSSL

Table of Contents

1. 如何创建一个文件的 MD5 或 SHA1 摘要?
2. 编码/解码
3. web 服务器 ssl 证书
4. 去除私钥的密码
5. 证书转换

5.1. CA证书
5.2. 创建CA证书有效期为一年
5.3. x509转换为pfx
5.4. PEM格式的ca.key转换为Microsoft可以识别的pvk格式
5.5. PKCS#12 到 PEM 的转换
5.6. 从 PFX 格式文件中提取私钥格式文件 (.key)
5.7. 转换 pem 到到 spc
5.8. PEM 到 PKCS#12 的转换
5.9. How to Convert PFX Certificate to PEM Format for SOAP

不多说了。

1. 如何创建一个文件的 MD5 或 SHA1 摘要?

摘要创建使用 dgst 选项.

```
# MD5 digest
openssl dgst -md5 filename

# SHA1 digest
openssl dgst -sha1 filename
```

Note

MD5 信息摘要也同样可以使用md5sum创建

```
C:\GnuWin32\neo>echo "Hello World!" > message.txt

C:\GnuWin32\neo>type message.txt
"Hello World!"

C:\GnuWin32\neo>openssl dgst -md5 message.txt
MD5(message.txt)= d9226d4bd8779baa69db272f89a2e05c

C:\GnuWin32\neo>openssl dgst -sha1 message.txt
SHA1(message.txt)= 423988b040f83a66d1b981735d4ef8933ce6fac0

C:\GnuWin32\neo>
```

其它可用摘要

```
C:\GnuWin32\neo>openssl list-message-digest-commands
md2
md4
md5
mdc2
rmd160
sha
sha1
```

2. 编码/解码

使用 base64-encode 编码/解码?

使用 enc -base64 选项

```
# send encoded contents of file.txt to stdout
openssl enc -base64 -in file.txt

# same, but write contents to file.txt.enc
openssl enc -base64 -in file.txt -out file.txt.enc
```

命令行

```
C:\GnuWin32\neo>openssl enc -base64 -in file.txt
SGVsbG8gV29ybGQhDQo=

C:\GnuWin32\neo>openssl enc -base64 -in file.txt -out file.txt.enc

C:\GnuWin32\neo>type file.txt.enc
SGVsbG8gV29ybGQhDQo=

C:\GnuWin32\neo>
```

通过管道操作

```
C:\GnuWin32\neo>echo "encode me" | openssl enc -base64
ImVuY29kZSBtZSIgDQo=

C:\GnuWin32\neo>echo -n "encode me" | openssl enc -base64
LW4gImVuY29kZSBtZSIgDQo=

C:\GnuWin32\neo>
```

使用 -d (解码) 选项来反转操作.

```
C:\GnuWin32\neo>openssl enc -base64 -d -in file.txt.enc
Hello World!

C:\GnuWin32\neo>openssl enc -base64 -d -in file.txt.enc -out file.txt
```

快速命令行

```
C:\GnuWin32\neo>type file.txt.enc | openssl enc -base64 -d
Hello World!

C:\GnuWin32\neo>type file.txt.enc
SGVsbG8gV29ybGQhDQo=

C:\GnuWin32\neo>echo SGVsbG8gV29ybGQhDQo= | openssl enc -base64 -d
Hello World!
```

可用的编码/解码方案

```
# or get a long list, one cipher per line
openssl list-cipher-commands

C:\GnuWin32\neo>openssl list-cipher-commands
```

```
aes-128-cbc
aes-128-ecb
aes-192-cbc
aes-192-ecb
aes-256-cbc
aes-256-ecb
base64
bf
bf-cbc
bf-cfb
bf-ecb
bf-ofb
cast
cast-cbc
cast5-cbc
cast5-cfb
cast5-ecb
cast5-ofb
des
des-cbc
des-cfb
des-ecb
des-ede
des-ede-cbc
des-ede-cfb
des-ede-ofb
des-ede3
des-ede3-cbc
des-ede3-cfb
des-ede3-ofb
des-ofb
des3
desx
idea
idea-cbc
idea-cfb
idea-ecb
idea-ofb
rc2
rc2-40-cbc
rc2-64-cbc
rc2-cbc
rc2-cfb
rc2-ecb
rc2-ofb
rc4
rc4-40
rc5
rc5-cbc
rc5-cfb
rc5-ecb
rc5-ofb
```

3. web 服务器 ssl 证书

```
$ sudo openssl req -new -x509 -keyout server.pem -out server.pem -days 365 -nodes
```

4. 去除私钥的密码

```
$ openssl rsa -in neo.key -out nopassword.key
Enter pass phrase for neo.key:
writing RSA key
```

5. 证书转换

PKCS 全称是 Public-Key Cryptography Standards ，是由 RSA 实验室与其它安全系统开发商为促进公钥密码的发展而制订的一系列标准，PKCS 目前共发布过 15 个标准。 常用的有：
PKCS#7 Cryptographic Message Syntax Standard
PKCS#10 Certification Request Standard
PKCS#12 Personal Information Exchange Syntax Standard
X.509是常见通用的证书格式。所有的证书都符合为Public Key Infrastructure (PKI) 制定的 ITU-T X509 国际标准。
PKCS#7 常用的后缀是： .P7B .P7C .SPC
PKCS#12 常用的后缀有： .P12 .PFX
X.509 DER 编码(ASCII)的后缀是： .DER .CER .CRT
X.509 PAM 编码(Base64)的后缀是： .PEM .CER .CRT
.cer/.crt是用于存放证书，它是2进制形式存放的，不含私钥。
.pem跟crt/cer的区别是它以Ascii来表示。
pfx/p12用于存放个人证书/私钥，他通常包含保护密码，2进制方式
p10是证书请求
p7r是CA对证书请求的回复，只用于导入
p7b以树状展示证书链(certificate chain)，同时也支持单个证书，不含私钥。

5.1. CA证书

用openssl创建CA证书的RSA密钥(PEM格式)：

```
openssl genrsa -des3 -out ca.key 1024
```

5.2. 创建CA证书有效期为一年

用openssl创建CA证书(PEM格式,假如有效期为一年)：

```
openssl req -new -x509 -days 365 -key ca.key -out ca.crt -config openssl.cnf
```

openssl是可以生成DER格式的CA证书的，最好用IE将PEM格式的CA证书转换成DER格式的CA证书。

5.3. x509转换为pfx

```
openssl pkcs12 -export -out server.pfx -inkey server.key -in server.crt
```

5.4. PEM格式的ca.key转换为Microsoft可以识别的pvk格式

```
pvk -in ca.key -out ca.pvk -nocrypt -topvk
```

5.5. PKCS#12 到 PEM 的转换

```
openssl pkcs12 -nocerts -nodes -in cert.p12 -out private.pem
验证
openssl pkcs12 -clcerts -nokeys -in cert.p12 -out cert.pem
```

5.6. 从 PFX 格式文件中提取私钥格式文件 (.key)

```
openssl pkcs12 -in mycert.pfx -nocerts -nodes -out mycert.key
```

5.7. 转换 pem 到到 spc

```
openssl crl2pkcs7 -nocrl -certfile venus.pem  -outform DER -out venus.spc
```

用 -outform -inform 指定 DER 还是 PAM 格式。例如：

```
openssl x509 -in Cert.pem -inform PEM -out cert.der -outform DER
```

5.8. PEM 到 PKCS#12 的转换

```
openssl pkcs12 -export -in Cert.pem -out Cert.p12 -inkey key.pem
```

IIS 证书

```
cd c:\openssl
set OPENSSL_CONF=openssl.cnf
openssl pkcs12 -export -out server.pfx -inkey server.key -in server.crt
```

server.key和server.crt文件是Apache的证书文件，生成的server.pfx用于导入IIS

5.9. How to Convert PFX Certificate to PEM Format for SOAP

```
$ openssl pkcs12 -in test.pfx -out client.pem
Enter Import Password:
MAC verified OK
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
```

Chapter 16. Java - keytool

Table of Contents

1. 创建证书
2. Private key generation
3. Public Key Certificate (optional)
4. import your signed certificate
5. Import the certificate and attach it to your server key pair
6. Key pair verification

1. 创建证书

```
keytool -genkey -keyalg RSA -keystore keys/server.keystore
Enter keystore password:  changeit
What is your first and last name?
  [Unknown]:  www.caucho.com
What is the name of your organizational unit?
  [Unknown]:  Resin Engineering
What is the name of your organization?
  [Unknown]:  Caucho Technology, Inc.
What is the name of your City or Locality?
  [Unknown]:  San Francisco
What is the name of your State or Province?
  [Unknown]:  California
What is the two-letter country code for this unit?
  [Unknown]:  US
Is <CN=www.caucho.com, OU=Resin Engineering,
  O="Caucho Technology, Inc.", L=San Francisco, ST=California, C=US> correct?
  [no]:  yes

Enter key password for <mykey>
        (RETURN if same as keystore password):  changeit
```

2. Private key generation

```
keytool -genkey -keyalg RSA -alias myserverkeypair \
                -storepass YourPasswordHere -keystore private.keystore
What is your first and last name?
  [Unknown]:  www.myserver.com
What is the name of your organizational unit?
  [Unknown]:  Foo Dept
What is the name of your organization?
  [Unknown]:  Bar
What is the name of your City or Locality?
  [Unknown]:  Paris
What is the name of your State or Province?
  [Unknown]:  France
What is the two-letter country code for this unit?
  [Unknown]:  FR
Is <CN=www.myserver.com, OU=Foo Dept, O=Bar, L=Paris,
                ST=France, C=FR> correct?
  [no]:  yes

Enter key password for <myserverkeypair>
        (RETURN if same as keystore password):
```

3. Public Key Certificate (optional)

```
>keytool -certreq -alias myserverkeypair -storepass YourPasswordHere \
                -keystore private.keystore
-----BEGIN NEW CERTIFICATE REQUEST-----
MIIBqjCCARMCAQAwajELMAkGA1UEBhMCRlIxDzANBgNVBAgTBkZyYW5jZTEOMAwGA1UEBxMFUGFy
... cut ...
KDYZTklbg1NOiXTdXIhPHb3+YOgZ+HoeDTxOx/rRhA==
-----END NEW CERTIFICATE REQUEST-----
```

4. import your signed certificate

```
keytool -import -alias servertest -storepass YourPasswordHere \
                -keystore private.keystore -file servertest.crt
```

5. Import the certificate and attach it to your server key pair

Import the certificate and attach it to your server key pair by typing the command

```
keytool -import -alias myserverkeypair -storepass YourPasswordHere \
                -keystore private.keystore -file myserver.cer
Certificate reply was installed in keystore
```

6. Key pair verification

```
keytool -list -v -alias myserverkeypair -storepass YourPasswordHere \
                -keystore private.keystore
```

Chapter 17. .Net makecert

Table of Contents

1. 访问X.509证书

1. 访问X.509证书

Java访问X.509证书

Part IV. 数字证书开发

Table of Contents

18. Java (java.security.*)

1. 访问X.509证书
2. 创建证书

19. SSL Socket

1. Java Socket HTTPS
2. Java SSL Socket Client
3. Java SSL Socket Server

Chapter 18. Java (java.security.*)

Table of Contents

1. 访问X.509证书
2. 创建证书

1. 访问X.509证书

Java访问X.509证书

```
/*
 * Created on 2005-7-1
 *
 * Author: neo chen <openunix@163.com>
 * Nickname: netkiller
 */
import java.io.*;
import java.security.cert.*;
import java.security.cert.CertificateFactory;

public class CertInfo {
        static String issue,after,before,subject;
        static String serialno,signalg;
        static int version;
        public void Init() throws Exception{
        CertificateFactory certFactory = CertificateFactory.getInstance("X.509");
        FileInputStream fis=new FileInputStream("e:/Java/chen.cer");
        X509Certificate cert =
(X509Certificate)certFactory.generateCertificate(fis);

        fis.close();
        issue=cert.getIssuerDN().toString();
        subject=cert.getSubjectDN().getName();
        after=cert.getNotAfter().toString();
        before=cert.getNotBefore().toString();
        version=cert.getVersion();
        serialno=cert.getSerialNumber().toString();
        signalg=cert.getSigAlgName();
        }
        public String getIssue(){
        return issue;
        }

        public String getAfter(){
        return after;
        }

        public String getBefore(){
        return before;
        }

        public String getSerial(){
        return serialno;
        }

        public String getsignalg(){
        return signalg;
        }

        public String getsubject(){
        return subject;
        }

        public String getversion(){
        return ("ver:"+version);
        }
```

```
        public static void main(String[] args) throws Exception
        {
        CertInfo c=new CertInfo();
        c.Init();
        System.out.println(c.getBefore());
        System.out.println(version);
        System.out.println(c.getversion());
        System.out.println(issue);
        System.out.println(c.getsubject());
        System.out.println(c.getsignalg());
        }
}
```

---

2. 创建证书

Chapter 19. SSL Socket

Table of Contents

1. Java Socket HTTPS
2. Java SSL Socket Client
3. Java SSL Socket Server

1. Java Socket HTTPS

```
package netkiller;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.net.MalformedURLException;
import java.net.URL;

import javax.net.ssl.HttpsURLConnection;
import javax.net.ssl.SSLContext;
import javax.net.ssl.TrustManager;
import javax.net.ssl.X509TrustManager;

public class HTTPS {

        public static void main(String[] args) {
            // Create a trust manager that does not validate certificate chains
            TrustManager[] trustAllCerts = new TrustManager[]{
                new X509TrustManager() {
                    public java.security.cert.X509Certificate[]
getAcceptedIssuers() {
                        return null;
                    }
                    public void checkClientTrusted(
                        java.security.cert.X509Certificate[] certs, String
authType) {
                    }
                    public void checkServerTrusted(
                        java.security.cert.X509Certificate[] certs, String
authType) {
                    }
                }
            };

            // Install the all-trusting trust manager
            try {
                SSLContext sc = SSLContext.getInstance("SSL");
                sc.init(null, trustAllCerts, new java.security.SecureRandom());

HttpsURLConnection.setDefaultSSLSocketFactory(sc.getSocketFactory());
            } catch (Exception e) {
            }

            // Now you can access an https URL without having the certificate in
the truststore
            try {
                //Create a URL for the desired page
                URL url = new URL("https://java.sun.com/");

                // Read all the text returned by the server
                BufferedReader in = new BufferedReader(new
InputStreamReader(url.openStream()));
                String html;
                while ((html = in.readLine()) != null) {
                    // str is one line of text; readLine() strips the newline
character(s)
                        System.out.println(html);
```

```
                }
                in.close();

            } catch (MalformedURLException mue) {
            } catch (IOException ioe) {
            }

        }

}
```

Prev
2. 创建证书

Up
Home

Next
2. Java SSL Socket Client

2. Java SSL Socket Client

```
package netkiller;

import java.io.*;
import java.net.*;
import javax.net.SocketFactory;
import javax.net.ssl.*;

public class SSLClientSocket {

        public static void main(String[] args) {
            try {
                int port = 443;
                String hostname = "java.sun.com";

                SocketFactory socketFactory = SSLSocketFactory.getDefault();
                Socket socket = socketFactory.createSocket(hostname, port);

                // Create streams to securely send and receive data to the server
                InputStream in = socket.getInputStream();
                OutputStream out = socket.getOutputStream();

                BufferedReader socketReader = new BufferedReader(new
InputStreamReader(in));
                PrintWriter socketWriter = new PrintWriter(out);

                socketWriter.println("GET /");
                socketWriter.flush();
                String line=null;
                StringBuffer html = new StringBuffer();
                while((line=socketReader.readLine())!=null){
                        html.append(line+"\n");
                }
                // Read from in and write to out...
                System.out.println(html.toString());

                // Close the socket
                socketReader.close();
                socketWriter.close();
                in.close();
                out.close();
            } catch(IOException e) {
            }

        }

}
```

3. Java SSL Socket Server

这里实现一个简单的SSL Echo服务器

创建证书

keytool -genkey -keyalg RSA -alias mycert -keystore mySrvKeystore

```
C:\workspace\test>keytool -genkey -keyalg RSA -alias mycert -keystore
mySrvKeystore
输入keystore密码:  13721218
您的名字与姓氏是什么?
  [Unknown]:  陈景峰
您的组织单位名称是什么?
  [Unknown]:  中国无线电运动协会
您的组织名称是什么?
  [Unknown]:  无线电运动协会
您所在的城市或区域名称是什么?
  [Unknown]:  深圳
您所在的州或省份名称是什么?
  [Unknown]:  广东省
该单位的两字母国家代码是什么
  [Unknown]:  CN
CN=陈景峰, OU=中国无线电运动协会, O=无线电运动协会, L=深圳, ST=广东省, C=CN 正确
吗?
  [否]:  Y

输入<mycert>的主密码
        (如果和 keystore 密码相同, 按回车):  13721218
```

C:\workspace\neo>javac netkiller\SSLServerSocket.java

java -Djavax.net.ssl.keyStore=mySrvKeystore -Djavax.net.ssl.keyStorePassword=13721218 netkiller.SSLServerSocket

Client

C:\workspace\neo>javac netkiller\SSLClientSocket.javajava -Djavax.net.ssl.trustStore=truststore -Djavax.net.ssl.trustStorePassword=13721218 netkiller.SSLClientSocket

Chapter 20. Credentials Organization

Table of Contents

1. VeriSign

1.1. iTrusChina
1.2. Thawte
1.3. Geotrust

2. UserTrust
3. 境内其他CA机构

3.1. WoSign®、I'm Verified®、WoTrust®、沃通®

4. 生成 .csr 文件

1. VeriSign

http://www.verisign.com/

http://www.verisign.com/cn/

VeriSign (Nasdaq: VRSN) is the trusted provider of Internet infrastructure services for the networked world. Billions of times each day, VeriSign helps companies and consumers all over the world engage in communications and commerce with confidence. VeriSign offerings include SSL, SSL Certificates, Extended Validation (EV SSL), VeriSign Trust Seal, two-factor authentication, identity protection, malware scan, public key infrastructure (PKI), DDoS mitigation and Domain Name Services.

1.1. iTrusChina

http://verisign.itrus.com.cn/

1.2. Thawte

http://www.thawte.com/

Thawte is a leading global Certification Authority. Our SSL and code signing digital certificates are used globally to secure servers, provide data encryption, authenticate users, protect privacy and assure online identifies through stringent authentication and verification processes. Our SSL certificates include Wildcard SSL Certificates, SGC SuperCerts and Extended Validation SSL Certificates.

thawte 是全球领先的认证机构。我们的 SSL 和代码签名数字证书在全球范围内提供服务器的安全保护，可以进行数据加密、可以验证用户，通过严格的验证和认证程序保护个人隐私，确保在线识别过程的安全。我们的 SSL 证书包括通配符 SSL 证书、SGC SuperCerts 和扩展验证 SSL 证书。

1.3. Geotrust

http://geotrust.itrus.com.cn/

2. UserTrust

http://www.usertrust.com/

Comodo offers essential infrastructure to enable e-merchants, other Internet-connected companies, software providers, and individual consumers to interact and conduct business via the Internet safely and securely. Our PKI solutions including, SSL Certificates, Extended Validation Certificates, Code Signing Certificates as well as Secure E-Mail Certificates, increase consumer trust in transacting business online, secure information through strong encryption, and satisfy many industry best practices or security compliance requirements with SSL.

3. 境内其他CA机构

3.1. WoSign®、I'm Verified®、WoTrust®、沃通®

http://www.wosign.com/

上级是 UserTrust

4. 生成 .csr 文件

```
# openssl genrsa -des3 -out example.com.key 2048
Generating RSA private key, 2048 bit long modulus
.....................................................................................+++
.....................................................................+++
e is 65537 (0x10001)
Enter pass phrase for example.com.key:
Verifying - Enter pass phrase for example.com.key:


# openssl req -new -key example.com.key -out example.com.csr
Enter pass phrase for example.com.key:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [GB]:CN
State or Province Name (full name) [Berkshire]:Guangdong
Locality Name (eg, city) [Newbury]:Shenzhen
Organization Name (eg, company) [My Company Ltd]:XXX CO.LTD.,
Organizational Unit Name (eg, section) []:Technical Support Center
Common Name (eg, your name or your server's hostname) []:*.example.com
Email Address []:webmaster@example.com

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```