众所周知，SHA-1不是一个安全的加密哈希函数。研究人员现在认为找到一对冲突的哈希值（即两次不同输入的哈希运算得出了相同的哈希值）是不可避免的，可能未来一个月内就会出现。这就对网络构成了潜在的威胁，因为如今很多网站依旧使用基于SHA-1的数字签名证书。幸运的是，仅仅只是找到一对冲突的哈希值是不足以伪造数字证书并打破网络信任模型的。

本文将讲述如何利用哈希碰撞来伪造数字证书，以及证书颁发机构如何通过例如使用随机证书序列号等措施使得攻击者更加难以伪造数字证书。

数字签名是信任的基石

互联网依赖于信任。无论你是登录网上银行或是浏览Reddit，HTTPS始终通过加密通信数据和验证网站数字证书保护着你。而大多浏览器在访问使用HTTPS的网站时都会非常直观的在地址栏上挂上一把锁。

当目标网站拥有一个包含所有者身份信息和主机名的数字证书时，HTTPS可以向浏览器证实当前网站的真实性。证书是一个由可信的第三方证书颁发机构颁发的包含数字签名的小文件。对浏览器来说，数字签名是信任的来源。如果你浏览器认为证书上面的数字签名是正确的，则浏览器会信任这个证书。这个用于身份认证的系统称为公钥基础设施(Public Key Infrastructure, PKI)。

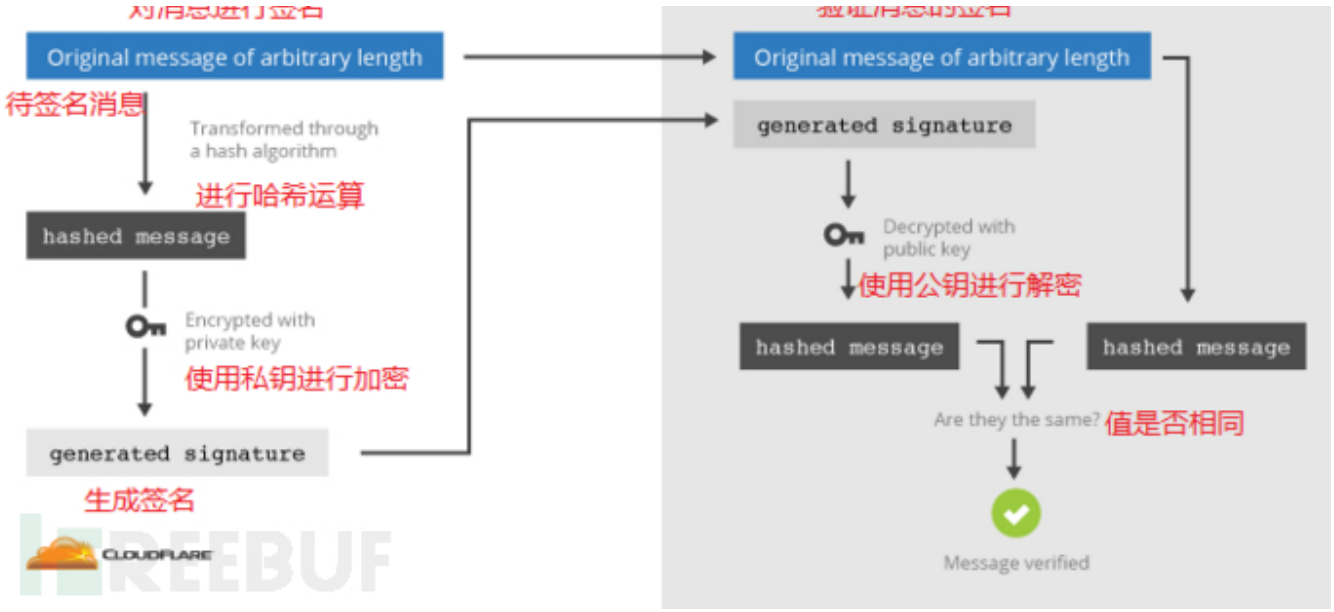
假如数字签名无法在被信任，那么所有依赖数字签名的系统都将崩溃。例如有人可以伪造出一个来自受信任CA颁发的cloudflare.com证书，他就可以伪造成cloudflare.com欺骗浏览者。

数字签名是一个基于消息和公钥计算出的一串数字。每个数字签名都需要一个公私钥对和一个哈希函数。其中，哈希函数是用来将消息替换为一个唯一的摘要，私钥用来进行签名，公钥用来对签名进行验证。

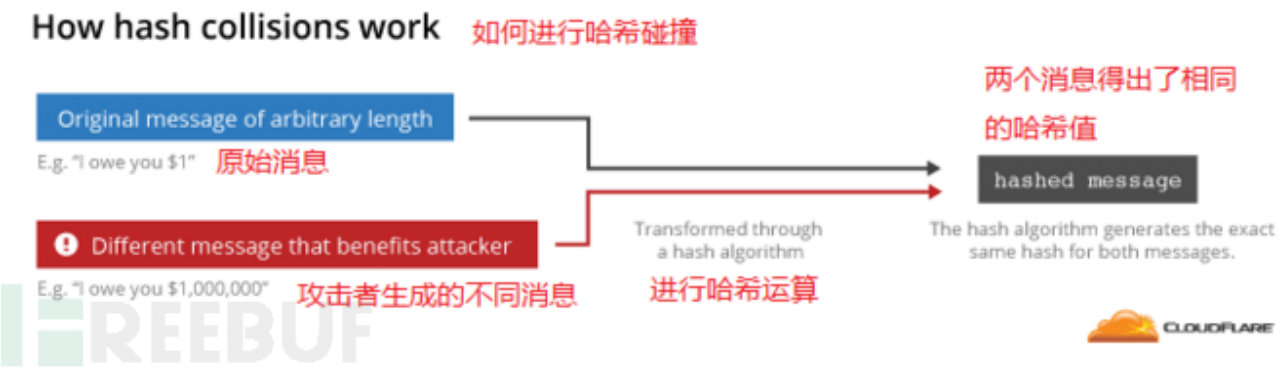
例如，创建一个RSA签名，你需要计算消息的哈希值，随后使用私钥加密这个计算出的哈希值。任何人都可以验证这个签名是否属于你，并可以验证消息的真实性。只需要获取到公钥，然后进行解密运算，将解密运算得出的值与源消息的哈希值进行比较，如果两个值相匹配，我们就认为数字签名是正确的。

Signing an encrypted message

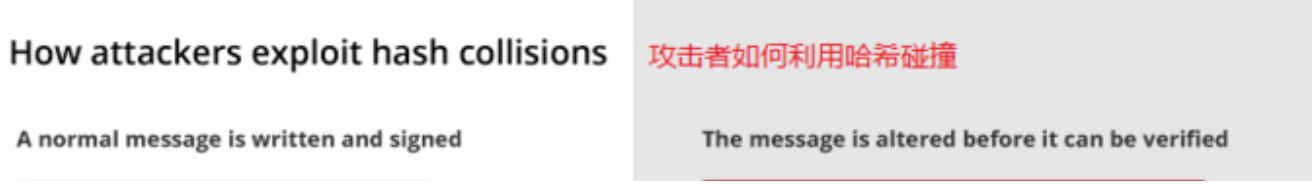
Verifying an encrypted message

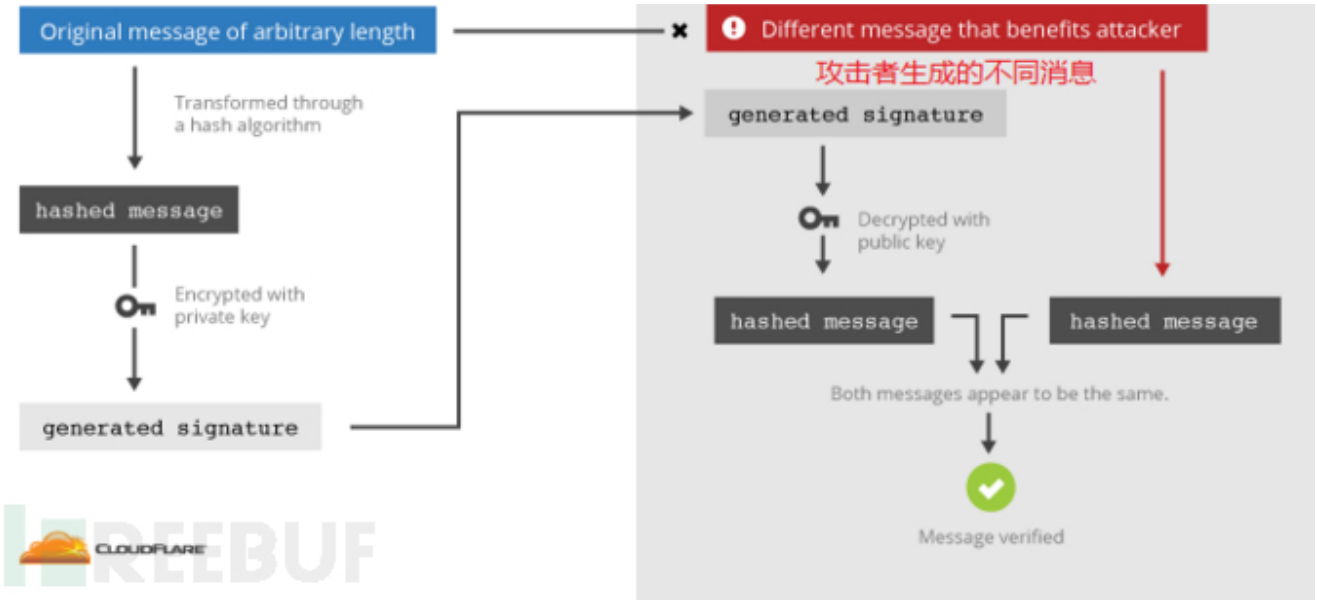


网站有多重途径可以获取到一个进过签名的证书。一般做法是从像GlobalSign或Comodo这类证书颁发机构购买证书。另一种方法就是窃取CA的私钥。排除潜在的违法行为，这种方法本身就非常困难：CA的密钥通常存储在一个用于防止窃取的专用安全设备中，这个安全设备称为硬件安全模块(Hardware Security Modules, HSM)。还有第三种更为有趣的方法获取证书，即伪造一个同可信证书具有相同哈希值的证书。



如果你找到了一个跟签名消息具有相同哈希值的消息，你就可以替换原有的消息，而最终验证签名也不会出错。这就可以用来伪造任何人签署他们本没有签署过的消息。





加密哈希函数的设计初衷就是抵御这类攻击事件，但是他们并不总能成功。

哈希函数的安全性

哈希函数设计时必须满足一下三个安全要求：

```
1.强抗碰撞性(Collision  Resistant)
2.弱抗碰撞性或抗第二原像性(second  pre-image  resistance)
3.单向性或抗原像性(pre-image  resistance)
```

强抗碰撞性是指没有比暴力破解更快的方法找到不同的两个输入进过哈希运算后得到相同的输出。这是最强的安全要求，但通常也是最先被打破的。

弱抗碰撞性是指给定一个值和它的哈希值，很难计算出另一个值的哈希值跟其相同。表面上看，这似乎保证了数字证书上面数字签名的安全性。然而，正如我们下面将讨论的，我们仍然可以伪造出一个具有弱抗碰撞性哈希函数的证书。

单向性是指给定一个哈希值，攻击者无法通过逆运算计算出其初始值。如果函数具有单向性，则很容易在满足强抗碰撞性和弱抗碰撞性。

```
译者注：
强抗碰撞性：找到x1和x2，使得f(x1)=f(x2)，非常困难

弱抗碰撞性：给定x1找x2，使得f(x1)=f(x2)，非常困难
单　　向　　性：给指定y找x，使得f(x)=y，非常困难
```

在下一节中，我们将使用一个用于数字证书但不抗冲突的哈希函数来描述上述要求。

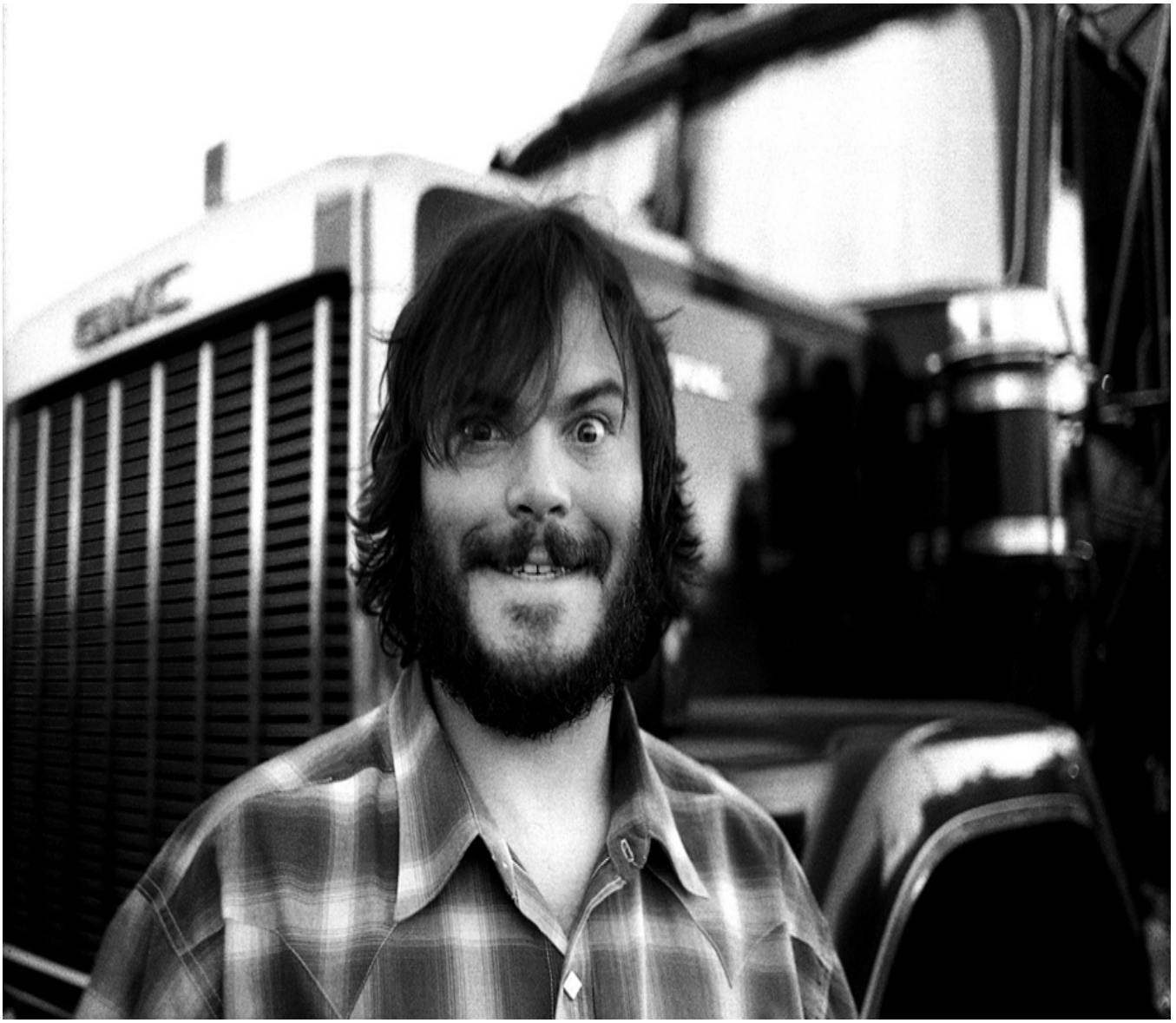
破解MD5

破解MD5

MD5是一个非常流行的加密哈希函数，但它也完全打破了强抗碰撞性。如今MD5依旧被广泛使用，不单单是在数字签名应用中。

在2004年由来自中国的研究人员(PDF)进过超级计算机近一个小时大量的数学运算后首次发现了MD5的碰撞。今天，在我们的笔记本上计算MD5的碰撞也仅需数秒钟的时间。这由Nat McHugh进行了巧妙的展示，他运用选择前缀碰撞法(chosen-prefix collisions)找出两张具有相同MD5值的图像，即下面的Barry White和James Brown(之后进行的三次碰撞又找到了Jack Black)。

下面是三幅具有相同MD5值的照片：



你可以通过使用curl共md5来进行验证：

```
$ curl -s https://blog.cloudflare.com/content/images/2015/08/white.jpg | md5
b69dd1fd1254868b6e0bb8ed9fe7ecad
$ curl -s https://blog.cloudflare.com/content/images/2015/08/brown.jpg | md5
b69dd1fd1254868b6e0bb8ed9fe7ecad
$ curl -s https://blog.cloudflare.com/content/images/2015/08/black.jpg | md5
b69dd1fd1254868b6e0bb8ed9fe7ecad
```

(译者注：由于CDN可能会将EXIF数据进行压缩导致验证出错，译者已将三张图片上传至云盘 http://yunpan.cn/cu3uxRUjgjeGa 访问密码 8a94)

尽管不像发现MD5碰撞那样重要，但我们还是不知道如何针对MD5进行弱抗碰撞性攻击。比如，给定任意一个数字签名证书，没人可以伪造出另一个具有相同哈希值的证书。尽管如此，研究人员还是制造出了一个跟现有受信任证书具有相同签名的证书。

他们使用的技术就是选择前缀攻击(chosen-prefix attack)，这一技术最初是由Marc Stevens在其硕士论文中提出的。选择前缀攻击被用于伪造证书，但其前提是证书颁发机构颁发的证书起始为可预测的值。

选择前缀攻击

如果你能找到一对冲突，你就可以将两个不同的值追加上一段数据然后进行哈希运算得到相同的哈希值。在最初的MD5碰撞攻击中，研究人员对两个消息M1和M2进行计算得到H(M1)=H(M2)。Stevens扩展了这项研究，找到了一种方法对已知的两个值P1和P2通过追加字节的方式进行碰撞。即对于前缀P1和P2，他演示了如何找到S1和S2使得H(P1|S1)=H(P2|S2)。

这就使得攻击者可以创造出两个具有相同哈希值的证书。下面我们先来看一下证书的简化结构：

```
序列号(Serial  number)
有效期(Validity  period)
域名(Domain  name)
公钥(Public  key)
扩展(X.509  extensions)
签名(Signature)
```

如果攻击者事先知道除域名外的所有值，则：

```
P1  =  序列号  |  有效期  |  真实域名
```

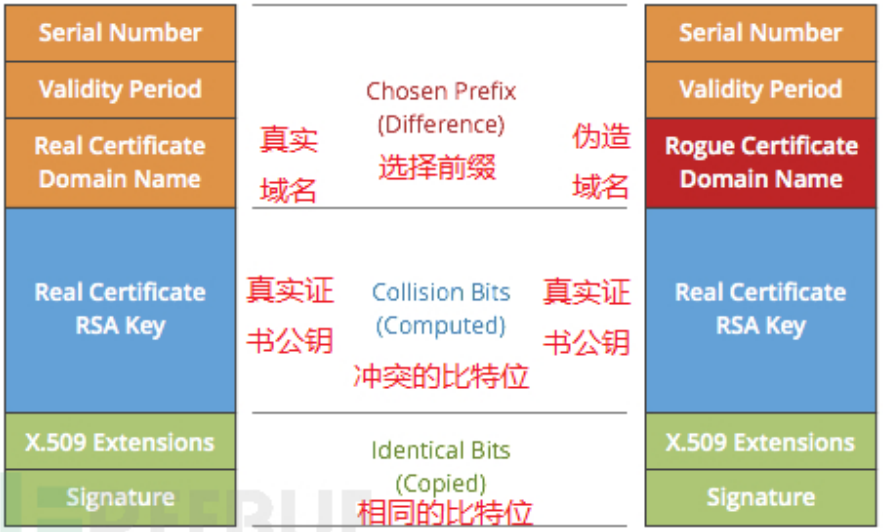
并且：

```
P2  =  序列号  |  有效期  |  伪造域名
```

然后，通过匹配公钥部分攻击者可以得到S1和S2。如果可以预测到序列号和有效期的数值，则可进过以下步骤进行碰撞攻击：

```
1.预测证书何时被发布
```

2. 预测该时间段内的证书序列号
3. 计算出几个包含序列号和有效期的前缀



攻击者首先对每一个取值预先进行选择前缀碰撞，直到碰撞出一个包含攻击者掌握的域名(如：attacker.com)与目标域名(如：google.com)相匹配的前缀。计算冲突比特位(collision bits)是因为我们伪造的域名跟原域名长度不一致从而导致前缀的末尾无法对齐，这是就会与公钥的初始几位比特值发现冲突。

在计算完冲突后，攻击者就必须想办法让证书颁发机构在正确的时间以正确的序列号给目标域名颁发证书，并且还要使用经过碰撞的公钥。如果幸运的话，最终证书颁发机构会颁发一张跟伪造证书拥有相同哈希值的证书。此时，攻击者就可以使用伪造的证书替代可信的证书进行使用了。

伪造MD5证书

2008年，一些研究人员(包括Marc Stevens和Alex Sotirov)使用选择前缀碰撞法伪造了一个受浏览器信任的证书。其POC在2008年的混沌通信大会上面进行了展示，而为了得到这个POC研究人员也仅花了几天的计算时间。

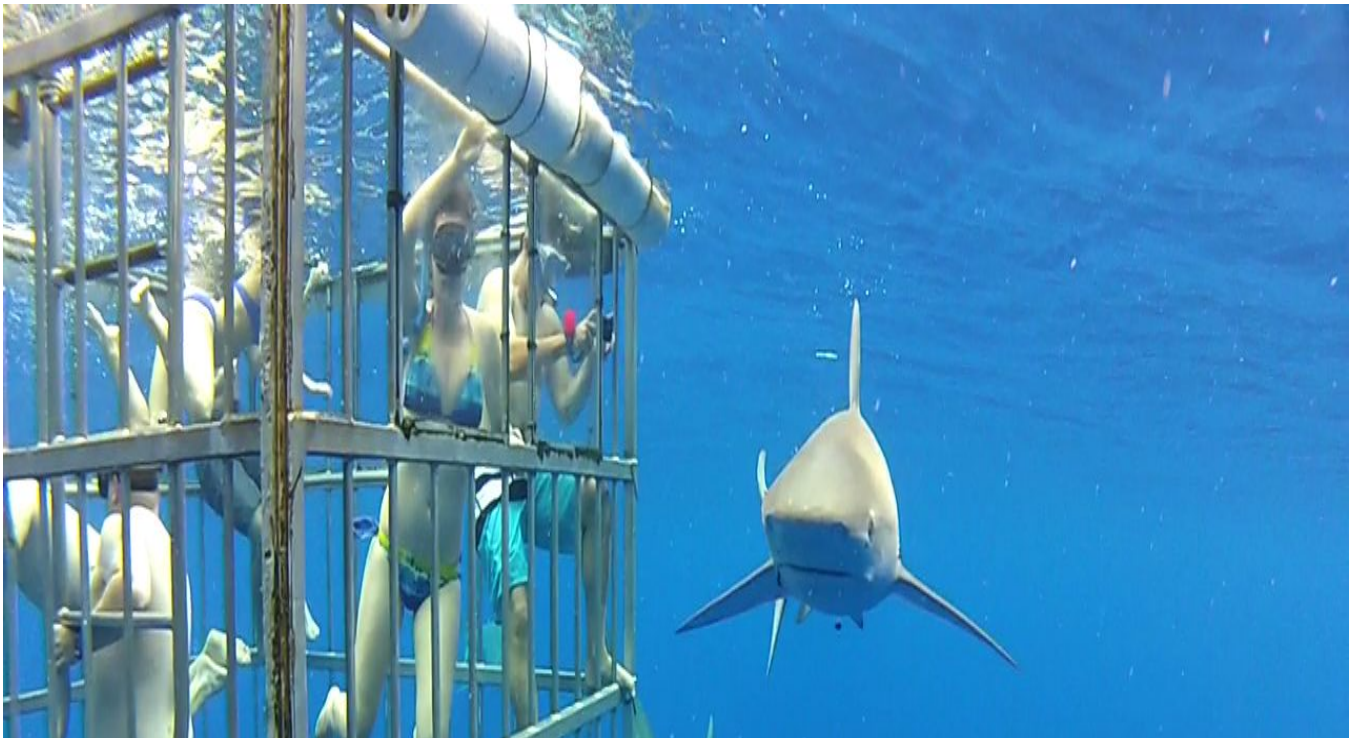
2012年，一个名为Flame的恶意软件被发现。Flame能通过劫持微软Windows Update服务来感染计算机。当时，Windows Update验证更新文件正是通过验证文件基于MD5数字签名得到的数字证书。Flame的作者伪造了微软的数字证书。通过对伪造证书的分析，发现作者很有可能就是基于选择前缀碰撞技术伪造的证书。

福布斯宣布这是2012年最令人担忧的安全发现，因为当用户端信任基于MD5签名的证书时，攻击者可能通过MD5碰撞来伪造任意网站的数字证书。

SHA-1 vs MD5

在数字签名应用中SHA-1是一个比MD5更加安全的算法。美国国家安全局在1995年发布SHA-1作为一个

安全散列标准。其最初设计认为进行碰撞代价为2^80。SHA-1已经使用了相当长的时间了，尽管还没有一个碰撞出现，但是一些理论攻击(最早出现于2005年，改进版出现于2012年)表明SHA-1已经不在安全。最新的研究表明，在容许攻击者决定部分原消息的条件之下仅需2^65计算复杂度即可找到一对碰撞。实际上，今年早些时候研究人员又发表一种称为“freestart碰撞”的SHA-1碰撞方法。



学术上的加密就像鲨鱼，一旦水中有血，就会吸引成群的鲨鱼。从第一次发布MD5的缺陷(1996)到第一次出现MD5的碰撞(2005)总共花费了9年时间，随后又进过三年的时间才伪造一个可信的数字证书(2008)。尽管这一过程可能非常缓慢，但始终是不间断的。

攻击者的技术在不断提升，电脑的运算能力也在不断增强。从提出MD5的缺陷到出现碰撞花费了9年时间，而SHA-1已经出现有10年时间了。如果以史为鉴，一个公开的SHA-1碰撞随时都可能出现。

使用随机序列号来抵御选择前缀攻击

一种提升数字证书抵御选择前缀攻击的方法就是使得前缀不可预测。

如上所述，选择前缀攻击需要攻击者能够实现预测出序列号和有效期。网络上使用的数字证书长度为160 bit。要求证书颁发机构使用无序不可预测的数字代替有序的序列号，将加大选择前缀碰撞攻击证书的代价。

例如，如果证书颁发机构确保序列号包含20位有效数据，则攻击者需要计算2^20次前缀以猜测出正确的序列号。这将使得攻击代价增加百万倍。

计算出SHA-1碰撞可能就会出现在2016年，可能就是使用选择前缀碰撞攻击。然而，使用部分随机序列就可以使得攻击者更加难以伪造证书。除非在哈希碰撞技术上出现重大突破，否则在长时间内我们不太可能看到一个冲突的数字证书。

总结

广泛运用于文件完整性校对和数字签名的SHA-1算法已经到了使用的寿命。一些浏览器甚至对使用基于SHA-1数字签名的网站直接提示安全警告。虽然SHA-1已经不安全，但这并不影响到整个W

HA-1数字签名的网站直接提示安全警告。尽管SHA-1已经不在安全，但这并不影响到整个Web的安全。在网络中，伪造数字签名是对信任体系的一大威胁。证书颁发机构使用随机的序列号将增大这种伪造证书的成本，从而使得像SHA-1这种存在潜在安全问题的算法可以使用更长的时间。这也就是为什么我们向CA/Browser Forum插图保留已颁发的SHA-1证书，但**强制**要求证书颁发机构使用包含20位有效数据的序列号。

***原文：cloudflare，FB小编xiaix编译，转自须注明来自FreeBuf黑客与极客（FreeBuf.COM）**