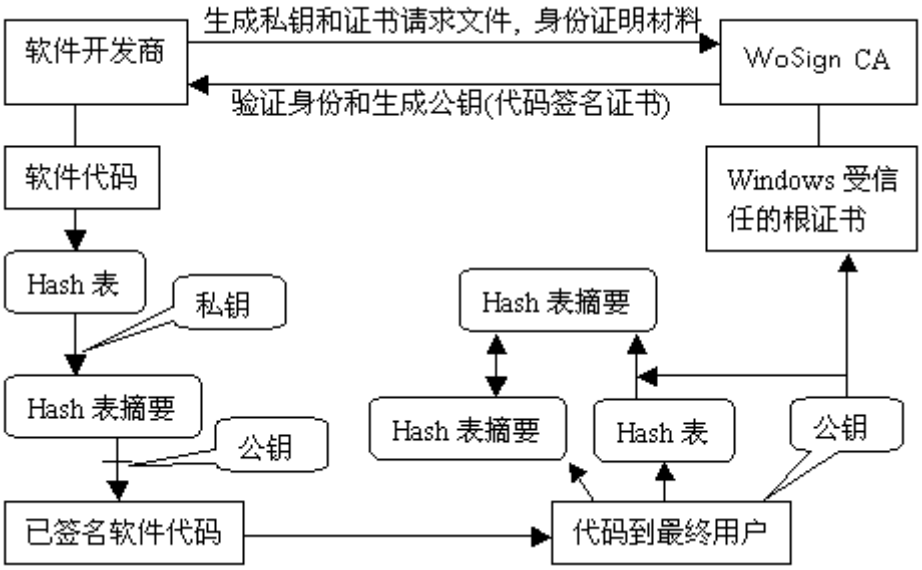
软件代码数字签名基本原理

在通过精美的包装盒销售软件的时代，大家使用什么防伪标志等来让用户识别什么是正版软件。但在当今的网络时代，有利的一面是软件开发商可以通过网络不受时间、地域的限制而快速发行软件，但不利的一面，则是用户无法辨认软件的真伪，根本无法确认软件代码的真实身份。在没有间谍软件和木马程序之前，大家可能还都信任某个软件就是软件中声称的开发商开发的软件，但是在今天就不应该这样认为了，互联网的匿名性使得用户根本无法确认此软件是否真的是软件中声称的开发商开发的软件！如何保证软件代码在网络传输过程中不会被非法修改，同时还能让用户非常清楚地识别软件发行者的真实身份（软件真实来源），答案就是代码签名。

以微软代码为例，为了保证微软 Windows 系统的安全和用户安全，微软推出了 Microsoft Authenticode 技术，即微软认证码技术，此技术保证了只有使用了 Windows 的受信任的根证书颁发机构颁发的代码签名证书对软件代码数字签名后才允许在 Windows 上运行，从而保证了软件代码来自真实的发行者和保证软件代码没有被非法篡改。

软件代码数字签名仍然采用 PKI 双钥技术，整个数字签名过程如下图所示：



软件开发商在自己电脑上生成私钥（.pvk）和证书请求文件（CSR）提交给 GDCA， 同时提交有关身份证明文件（ 如营业执照和第三方证明文件等 ）给 GDCA 查验，GDCA 验证身份后用自己的私钥给 CSR 文件签名后生成代码签名证书，也就是公钥（.spc）给软件开发商。这样就完成了证书的申请和颁发。

软件开发商用代码签名工具给要签名的代码生成一个 Hash 表，再用其私钥加密 Hash 表产生认证摘要，接着就把认证摘要连同其公钥与软件代码一起打包生成签名后的新的软件代码，软件开发商就可以把已经签名的代码放到网上发行了。

最终用户从网上下载已经签名的代码时，浏览器会从签名代码中解读出其签名证书（公钥）和 Hash 表摘要，并与 Windows 的受信任的根证书相比较查验公钥证书的有效性和合法性，验证签名证书正确后，就可以确认此代码确实是来自真实的软件开发商。

接着，再使用签名时使用的同样算法对软件代码生成一个 Hash 表，并使用公钥也同样生成一个 Hash 表认证摘要，比较从代码中解包出来的 Hash 表认证

摘要与生成的 Hash 表认证摘要是否一致，如果一致，则表明此代码在传输过程中未有任何修改，从而可以确认代码的一致性。

从以上整个过程的简单介绍，可以看出：

(1) 购买代码签名证书一定要从 Windows 内置的受信任的根证书颁发机构购买（如：GDCA)，否则无法通过验证。而通过人为的添加根证书到 Windows 受信任的根证书存储区，一来不可能要求所有网上用户在使用代码之前先下载和安装某个根证书，更重要的是，谁都可以人为添加的根证书不能保证签名证书的唯一性和权威性。

(2) 代码签名后不仅保证了软件开发商的真实身份，而且还保证了代码的完整性，以免代码被病毒干扰和被非法篡改。

(3) 只有使用了 Windows 受信任的证书颁发机构（如 GDCA）颁发的代码签名证书签名的代码才允许下载，所以，如果您要让您的代码能让用户放心地下载，就一定要申请 GDCA 的代码签名证书。