接收方使用发送方的公钥对签名块进行运算以得到$E(D(\text{hash}))$。这实际上是对解密后的散列进行“加密”，操作抵消，以恢复原有的散列。如果计算后的散列值与签名块中的散列值不一致，则表明：要么文档、要么签名块、要么两者共同被篡改过（或无意中被改动）。这种方法仅仅对一小部分数据（散列）运用了（慢速的）公钥密码体制。请注意这种方法仅仅对所有满足下面条件的x起作用：

$$E(D(x)) = x$$

我们并不能保证所有的加密函数都拥有这种属性，因为我们原来所要求的就是：

$$D(E(x)) = x$$

在这里，E是加密函数，D是解密函数。而为了满足签名的要求，函数运算的次序是不受影响的。也就是说，D和E一定是可交换的函数。而RSA算法就有这种属性。

要使用这种签名机制，接收方必须知道发送方的公钥。有些用户在其Web网页上公开他们的公钥，但是其他人并没有这么做，因为他们担心入侵者会闯入并悄悄地改动其公钥。对他们来说，需要其他方法来发布公钥。消息发送方的一种常用方法是在消息后附加数字证书，证书中包含了用户姓名、公钥和可信任的第三方数字签名。一旦用户获得了可信的第三方认证的公钥，那么对于所有使用这种可信第三方确认来生成自己证书的发送方，该用户都可以使用他们的证书。

认证机构（Certification Authority，CA）作为可信的第三方，提供签名证书。然而如果用户要验证有CA签名的证书，就必须得到CA的公钥，从哪里得到这个公钥？即使得到了用户又如何确定这的确是CA的公钥呢？为了解决上述两个问题，需要一套完整的机制来管理公钥，这套机制叫做PKI（Public Key Infrastructure，公钥基础设施）。网络浏览器已经通过一种特别的方式解决了这个问题：所有的浏览器都预加载了大约40个著名CA的公钥。

上面我们叙述了可用于数字证书的公钥密码体制。同时，我们也有必要指出不包含公钥体制的密码体系同样存在。

9.2.5 可信平台模块

加密算法都需要密钥（Key）。如果密钥泄露了，所有基于该密钥的信息也等同于泄露了，可见选择一种安全的方法存储密钥是必要的。接下来的问题是：如何在不安全的系统中安全地保存密钥呢？

有一种方法在工业上已经被采用，该方法需要用到一种叫做可信平台模块（Trusted Platform Modules，TPM）的芯片。TPM是一种加密处理器（cryptoprocessor），使用内部的非易失性存储介质来保存密钥。该芯片用硬件实现数据的加密/解密操作，其效果与在内存中对明文块进行加密或对密文块进行解密的效果相同，TPM同时还可以验证数字签名。由于其所有的操作都是通过硬件实现，因此速度比用软件实现快许多，也更可能被广泛地应用。一些计算机已经安装了TPM芯片，预期更多的计算机会在未来安装。

TPM的出现引发了很多争议，因为不同厂商、机构对于谁来控制TPM和它用来保护什么有分歧。微软大力提倡采用TPM芯片，并且为此开发了一系列应用于TPM的技术，包括Palladium、NGSCB以及BitLocker。微软的观点是，由操作系统控制TPM芯片，并使用该芯片阻止非授权软件的运行。“非授权软件”可以是盗版（非法复制）软件或仅仅是没有经过操作系统认证的软件。如果将TPM应用到系统启动的过程中，则计算机只能启动经过内置于TPM的密钥签名的操作系统，该密钥由TPM生产商提供，该密钥只会透露给允许被安装在该计算机上的操作系统的生产商（如微软）。因此，使用TPM可以限制用户对软件的选择，用户或许只能选择经过计算机生产商授权的软件。

由于TPM可以用于防止音乐与电影的盗版，这些媒体生产商对该芯片表现出了浓厚的兴趣。TPM同样开启了新的商业模式，如“租借”歌曲与电影。TPM通过检查日期判断当前媒体是否已经“过期”，如果过期，则拒绝为该媒体解码。

TPM还有非常广泛的应用领域，而这些领域都是我们还未涉足的。有趣的是，TPM并不能提高计算机在应对外部攻击中的安全性。事实上，TPM关注的重点是采用加密技术来阻止用户做任何未经TPM控制者直接或间接授权的事情。如果读者想了解更多关于TPM的内容，在Wikipedia中关于可信计算（Trusted Computing）的文献可能会对你有所帮助。