布“矢泽久雄的公钥是 3 哦”。这之后当诸位要向笔者发送数据的时候，就可以用这个公钥 3 加密数据了。这样就算加密后的密文被人盗取了，只要他还不知道笔者的私钥就不可能对其解密，从而保证了数据的安全性。而收到了密文的笔者，则可以使用只有笔者自己才知道的私钥 5 对其解密（如图 10.7(2) 所示）。怎么样？这个技术很棒吧！

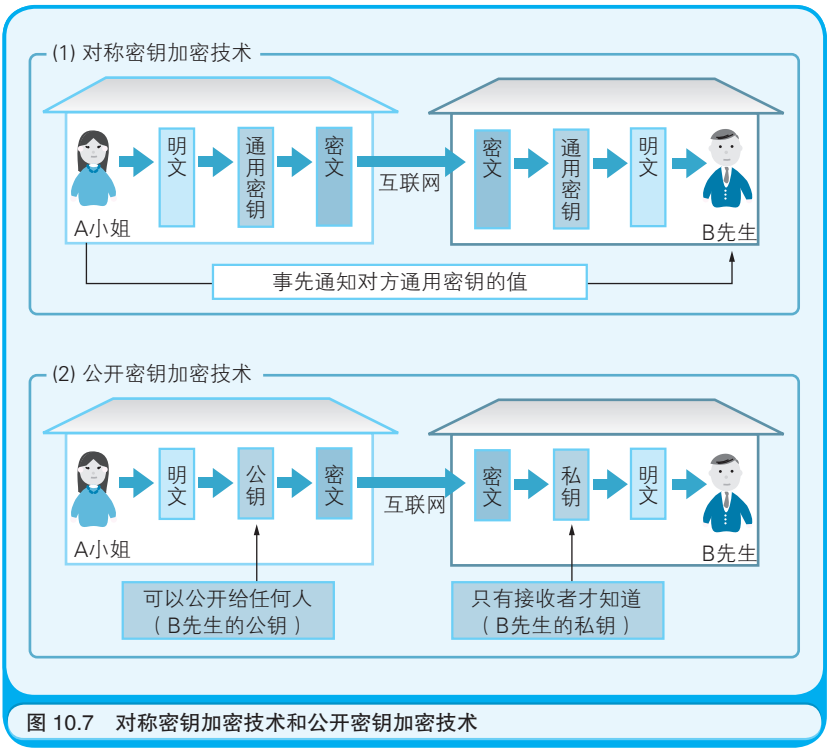

图 10.7 对称密钥加密技术和公开密钥加密技术

可用于实现公开密钥加密技术的算法有若干种，这里笔者将介绍目前广泛应用于互联网中的 RSA 算法。RSA 这个名字是由三位发明者 Ronald Rivest、Adi Shamir 和 Leonard Adleman 姓氏的首字母拼在一起组成的。美国的 RSA 信息安全公司对 RSA 的专利权一直持有到 2000

年 9 月 20 日。使用 RSA 创建公钥和私钥的步骤如图 10.8 所示。无论是公钥还是私钥都包含着两个数值，两个数值组成的数对儿才是一个完整的密钥。

图 10.8　创建公钥和私钥的步骤

由图 10.8 的步骤可以得出：323 和 11 是公钥，323 和 131 是私钥，的确是两个值都不相同的密钥。在使用这对儿密钥进行加密和解密时，需要对每个字符执行如图 10.9 所示的运算。这里参与运算的对象是字母 N（字符编码为 78）。用公钥对 N 进行加密得到 224，用私钥对 224 进行解密可使其还原为 78。

乍一看会以为只要了解了 RSA 算法，就可以通过公钥 c = 323、e = 11 推算出私钥 c = 323，f = 131 了。但是为了求解私钥中的 f，就不得不对 c 进行因子分解，分解为两个素数 a、b。在本例中 c 的位数很短，而在实际应用公开密钥加密时，建议将 c 的位数（用二进制数表示

时）扩充为 1024 位（相当于 128 字节）。要把这样的天文数字分解为两个素数，就算计算机的速度再快，也还是要花费不可估量的时间，时间可能长到不得不放弃解密的程度。

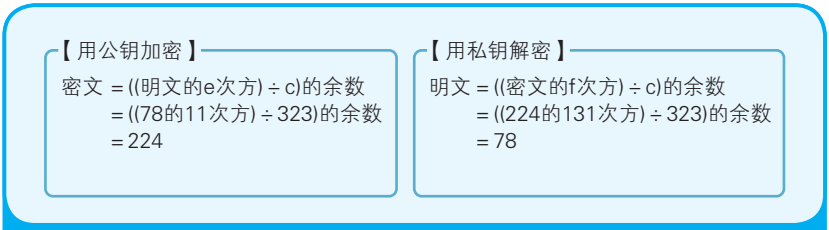

图 10.9 用公钥加密，用私钥解密

10.5 数字签名可以证明数据的发送者是谁

在本章的最后，先来介绍一种公开密钥加密技术的实际应用——数字签名。在日本的商界有盖章的习惯，而在欧美则是签字。印章和签名都可以证明一个事实，那就是某个人承认了文件的内容是完整有效的。而在通过网络传输的文件中，数字签名可以发挥出与印章和签名同样的证明效果。通常可以按照下面的步骤生成数据签名。步骤中所提及的“信息摘要”（Message Digest）可以理解为就是一个数值，通过对构成明文的所有字符的编码进行某种运算就能得出该数值。

【文本数据的发送者】

（1）选取一段明文

例：NIKKEI

（2）计算出明文内容的信息摘要

例：(78 + 73 + 75 + 75 + 69 + 73) ÷ 100 的余数 = 43

（3）用私钥对计算出的信息摘要进行加密

例：43 → 66（字母 B 的编码）

（4）把步骤（3）得出的值附加到明文后面再发送给接收者

例：NIKKEIB

【文本数据的接收者】

（1）用发送者的公钥对信息摘要进行解密

例：B = 66 → 43

（2）计算出明文部分的信息摘要

例：(78 + 73 + 75 + 75 + 69 + 73) ÷ 100 的余数 = 43

（3）比较在步骤（1）和（2）中求得的值，二者相同则证明接收的信息有效

例：因为两边都是 43，所以信息有效

请诸位注意，这里是使用私钥进行加密、使用公钥进行解密，这与之前的用法刚好相反（如图 10.10 所示）。而且这里所使用的是信息发送者（图 10.10 中的 A 小姐）的密钥对儿，而之前所使用的则是信息接收者（B 先生）的密钥对儿。

本例中信息摘要的算法是把明文中所有字母的编码加起来，然后取总和的最后两位。而在实际中计算数字签名时，使用的是通过更加复杂的公式计算得出的、被称作 MD5（Message Digest5）的信息摘要。由于 MD5 经过了精心的设计，所以使得两段明文即使只有略微的差异，计算后也能得出不同的信息摘要。

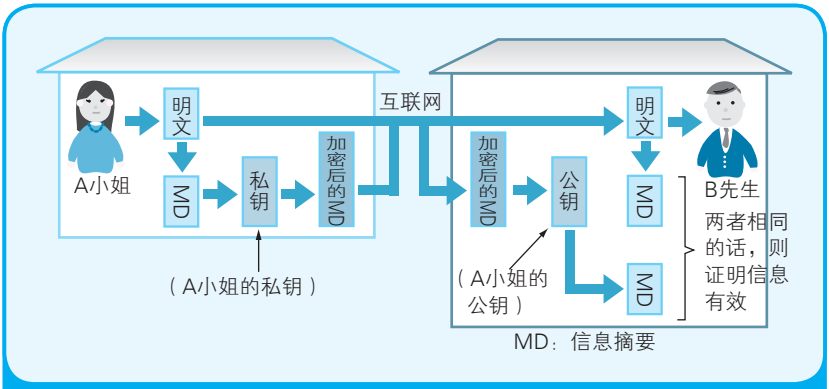

图 10.10 创建数字签名的步骤

也许诸位会认为把文件发送者的名字，比如“矢泽久雄”这个字符串用私钥加密，然后让对方用公钥解密也能代替印章或签字。但是如果这样做就不算是数字签名了，因为印章或签字有两层约束。其一是发送者承认文件的内容是完整有效的；其二是文件确实是由发送者本人发送的。发送者用构成文件的所有字符的编码生成了信息摘要，就证明发送者从头到尾检查了文件并承认其内容完整有效。如果接收者重新算出的信息摘要和经过发送者加密的信息摘要匹配，就证明文件在传输过程中没有被篡改，并且的确是发送者本人发送的。正因为数据是用发送者的私钥加密的，接收者才能用发送者的公钥进行解密。

☆ ☆ ☆

其实，绝对无法破解的加密技术也是存在的。首先密钥的位数要与文件数据中的字符个数相同，其次每次发送文件时都需要先更换密钥，最后为了防止密钥被盗，发送者还要亲手把密钥交给接收者。诸位明白为什么说这样做就绝对无法破解了吗？原因在于这样做等同于发送完全随机并且没有任何意义的数据。可是这种加密技术是不切实

际的。合理的密钥应该满足如下条件：长短适中、可以反复使用、可以通过某种通信手段交给接收者，并且通信双方以外的其他人难以用它来解密。公开密钥加密技术就完全满足上述条件，笔者在这里要对发明了这项技术的工程师们表达由衷的敬意。

在接下来的第 11 章中，笔者将介绍作为通用数据格式的 XML。敬请期待！

第 11 章 XML 究竟是什么

热身问答

在阅读本章内容前，让我们先回答下面的几个问题来热热身吧。

初级问题

XML 是什么的缩写？

中级问题

HTML 和 XML 的区别是什么？

高级问题

在处理 XML 文档的程序组件中，哪个成为了 W3C 的推荐标准？

怎么样？被这么一问，是不是发现有一些问题无法简单地解释清楚呢？下面，笔者就公布答案并解释。

答案

初级问题：XML 是 Extensible Markup Language（可扩展标记语言）的缩写。

中级问题：HTML 是用于编写网页的标记语言。XML 是用于定义任意标记语言的元语言。

高级问题：DOM（Document Object Model，文档对象模型）。

解释

初级问题：所谓标记语言，就是可以用标签为数据赋予意义的语言。

中级问题：通常把用于定义新语言的语言称作元语言。通过使用 XML 可以定义出各种各样的新语言。

高级问题：本章将会介绍使用了 DOM 的示例程序。

本章重点

在计算机行业，没听说过 XML 这个词的人恐怕不存在吧。诸位也一定都知道 XML 这个词，而且也应该能深切地体会到，XML 作为一种诞生不到 10 年的新技术，却不断地渗透到了计算机的各个领域。例如，这个应用程序能够把文件保存成 XML 格式；那个 DBMS（数据库管理系统）的下一个版本将支持 XML；而那个 Web 服务是基于 XML 实现的……

本章的主题将围绕“XML 究竟是什么”来展开。XML 其格式本身就是既简单又通用的。也正因为如此，XML 才会被扩充成各种各样的形式，应用于各种各样的场景。而且今后对 XML 的利用方式也将不断地进化下去。为了不至于对进化后的 XML 形态感到吃惊，趁着现在我们就先来整理一下 XML 的基础知识吧。

11.1 XML 是标记语言

本章就从 XML 这个词的含义开始讲起吧。XML 是 eXtensible Markup Language 的缩写，译为可扩展标记语言。下面先介绍什么是“标记语言”，接着再说明何谓“可扩展”。

其实诸位已经在享用标记语言所带来的便利了。例如用于编写网页的 HTML（Hypertext Markup Language，超文本标记语言）就是一种标记语言。请看图 11.1，这个网页实际上是一个名为 index.html 的 HTML 文件，部署在日经 BP 公司的 Web 服务器上。一般情况下，HTML 文件的扩展名是 .html 或 .htm。

図 11.1　日经软件的首页，这个页面的本质是个 HTML 文件

只要从 Internet Explorer Web 浏览器的“查看”菜单中选择“源文件”，就会自动打开浏览器所附带的“原始源”窗口，上面显示的正是 index.html 的内容（如图 11.2 所示）。可以看到里面有很多用“<”和“>”括起来的单词，例如 <html>、<head>、<title>、<body> 等。通常把它们称作“标签”。<html> 是用于表示这是 HTML 文件的标签。同样，其他标签也分别被赋予了意义，<head> 表示网页的头部，<title> 表示网页的标题，<body> 表示网页的主体。除此之外还有很多标签，例如使文字加粗显示的 <b>、在网页中插入图片的 <img>，等等。

通常把通过添加标签为数据赋予意义的行为称为“标记”。为这种给数据赋予意义的行为定义规则的语言就是“标记语言”。HTML 是用于编写网页的标记语言，更简单地说法就是 HTML 决定了可用于编写网页的标签。

也可以这样说，可使用的标签的种类决定了标记语言的规范。Web 浏览器会对 HTML 的标签进行解析，把由它们标记的信息渲染成在视觉上可以阅读的网页。