

信息安全原理与实践第二版课后习题部分解答（素材来源网络）（解答来源于GPT4模型，具体以参考答案为准）

可参考我的个人总结，对答案的总结性描述，能够缩短你看文章的时间，如有侵权，请联系我，我将删除

题号随机，选取的都是我觉得更重要的题目，更具有参考价值的题目~

第一章

1.

站在银行的立场看，其客户数据的完整性和机密性，通常哪一个更重要呢？站在银行客户的立场上，又是哪个更为重要呢？



对于银行来说，客户数据的完整性和机密性都非常重要，不能轻易地说哪一个更重要。如果数据的完整性被破坏，可能会导致错误的交易或者决策，对银行的运营造成严重影响。如果数据的机密性被破坏，可能会导致客户信息的泄露，对银行的声誉造成严重损害，甚至可能触犯数据保护法规，导致法律责任。

对于银行客户来说，他们可能更关心数据的机密性。如果他们的个人信息、账户信息或者交易信息被泄露，可能会导致身份盗用、欺诈或者其他形式的犯罪，对他们个人造成严重损害。然而，这并不意味着数据的完整性对他们来说就不重要。如果数据的完整性被破坏，可能会导致他们的账户余额错误、交易失败等问题，对他们的财务状况造成影响。

总的来说，数据的完整性和机密性都是非常重要的，无论是对于银行还是对于银行的客户。在实际的数据保护工作中，银行需要同时保证数据的完整性和机密性，不能忽视其中任何一个方面。

银行更看重两者，不然会影响其经营

个人更看重机密性，以为个人信息的泄露可能导致一些更严重的后果

但是完整性和机密性都重要

2.

假如Alice提供的不是网上银行服务，而是一种在线国际象棋游戏服务，暂且就称之为Alice在线象棋（AOC吧。玩家们按月支付一定的费用，就可以登录AOC和另外一个水平相当的玩家进行比赛。

- a.请问，在什么情况下，机密性对于AOC和他的客户都会很重要？请解释为什么？
- b.请问，完整性为什么是必需的
- c.请问，为什么可用性会是一个重要的考虑因素？



a. 机密性对于 AOC 和他的客户可能在以下情况下很重要：

客户的个人信息：客户在注册和使用服务时，可能需要提供一些个人信息，如姓名、电子邮件地址、信用卡信息等。如果这些信息的机密性被破坏，可能会导致客户的个人信息被泄露，对客户造成损害。

游戏策略：在一场正在进行的棋局中，如果一方的棋步被对手或第三方获取，那么这将严重影响比赛的公平性。因此，棋步的机密性也是非常重要的。

b. 完整性是必需的，因为：

游戏数据：如果游戏的数据（如棋局状态、玩家的得分等）的完整性被破坏，可能会导致游戏无法正常进行，或者导致比赛结果不公正。

客户信息：如果客户的个人信息或账户信息的完整性被破坏，可能会导致客户无法正常使用服务，或者导致客户的权益受到损害。

c. 可用性是一个重要的考虑因素，因为：

服务可用性：如果 AOC 的服务无法正常提供，那么客户就无法使用这个服务进行游戏，这将直接影响到 AOC 的业务。

数据可用性：如果客户的数据（如棋局状态、个人设置等）无法正常访问，那么客户就无法正常进行游戏，这将影响到客户的使用体验。

a. 个人信息和游戏策略

b. 游戏数据和客户信息

c. 服务可用性和数据可用性

总之就是要保障游戏的可进行性和公平性

3.

验证码往往用于辅助实现一种对于接入访问的约束，该约束针对的是实体的人（相对于其他自动化的接入访问过程）

a.请举一个现实世界的例子，说明在你接入访问某些资源时，会被要求处理验证码。

面对要处理的验证码，你都需要做些什么呢？

b.请讨论：针对你在a中描述的验证码的作用，都有哪些不同种类的技术方法可能会用来破坏其功效？

c.针对你在a中描述的验证码的作用，都有哪些非技术方法，可能会用于破坏其功效！

请列举出来。

d.在a中描述的验证码的实际效果如何？验证码的用户友好性又如何呢？

e.为什么你会讨厌验证码呢？



a. 当我尝试登录我的电子邮件账户时，我可能会被要求处理验证码。这通常发生在我从一个新的设备或位置登录时，或者在我忘记密码并尝试重置密码时。处理验证码通常需要我输入显示在屏幕上的一组字符，或者选择一组图片中符合某种描述的图片。

b. 针对验证码的作用，有一些技术方法可能会用来破坏其功效：

OCR (Optical Character Recognition)：攻击者可能会使用 **OCR** 技术来自动识别和输入字符型验证码。

机器学习：攻击者可能会使用机器学习算法来训练一个模型，使其能够自动识别和处理图片型验证码。

c. 针对验证码的作用，有一些非技术方法可能会用来破坏其功效：

社会工程：攻击者可能会通过欺骗用户，让用户自己提供验证码。

人工服务：攻击者可能会雇佣人工输入验证码，这种服务在一些低工资国家是可行的。

d. 验证码的实际效果通常取决于其复杂性和用户的熟悉程度。一般来说，验证码能够有效地防止自动化的攻击，但可能对用户的体验产生负面影响。对于一些用户来说，验证码可能会显得难以理解或难以操作，从而降低了用户友好性。

e. 我可能会讨厌验证码，因为它增加了我使用服务的复杂性。我需要花费额外的时间和精力来处理验证码，这可能会打断我的工作流程，影响我的使用体验。此外，一些验证码可能设计得过于复杂，使我难以正确输入，这可能会让我感到沮丧。

破坏验证码功效：

技术手段：机器学习

非技术手段：社会工程，人工服务（低工资水平的国家可行）

4.

一些商业软件是闭源的，意思是用户不能够获得其源代码。另一方面，还有一些软件是开源的，意思是其源代码对用户来说是可获得的。

- a. 请举一个你用的（或者曾经用过的）闭源软件的例子。
- b. 请举一个你用的（或者曾经用过的）开源软件的例子。
- c. 对于开源软件，Trudy 都能做些什么来查找软件中的安全缺陷呢
- d. 对于闭源软件，Trudy 又能做些什么来查找软件中的安全缺陷呢？
- e. 对于开源软件，Alice 都能做些什么来使软件更加安全呢？
- f. 对于闭源软件，Alice 又能做些什么来使软件更加安全呢？
- g. 开源软件和闭源软件，哪一种的固有安全性会更好呢？请说明为什么？



a. 我曾经使用过的一个闭源软件的例子是 **Microsoft Office**，这是一个商业的办公软件套件，它的源代码不对公众开放。

b. 我曾经使用过的一个开源软件的例子是 **Linux** 操作系统，它的源代码是公开的，任何人都可以查看和修改。

c. 对于开源软件，Trudy 可以直接查看和分析源代码，寻找可能存在的安全缺陷。她还可以修改源代码，创建并运行自己的版本，以测试软件的行为和响应。

d. 对于闭源软件，Trudy 可以使用各种黑盒测试技术，如模糊测试、逆向工程等，来查找软件中的安全缺陷。但是，由于源代码不可见，这可能会比查找开源软件的安全缺陷更困难。

e. 对于开源软件，Alice 可以查看和分析源代码，寻找并修复可能存在的安全缺陷。她还可以利用社区的力量，邀请其他人帮助查找和修复安全缺陷。

f. 对于闭源软件，Alice 需要依赖软件供应商来保证软件的安全性。她可以通过各种方式（如购买支持服务、参与 **beta** 测试等）来向供应商反馈可能存在的问题，并等待供应商提供修复。

g. 开源软件和闭源软件的固有安全性并没有绝对的好坏之分。开源软件的源代码可见，这意味着任何人都可以查找和修复安全缺陷，但也意味着攻击者可以更容易地找到利用的方式。闭源软件的源代码不可见，这可能使得找到安全缺陷更困难，但一旦找到，修复的速度和效率可能会受到限制。总的来说，软件的安全性更多地取决于开发和维护的过程，而不仅仅是源代码是否开放。

Microsoft Office 是一个闭源软件的例子，而 **Linux** 是一个开源软件的例子。对于开源软件，用户如 Trudy 可以直接查看和分析源代码以寻找安全缺陷，而 Alice 可以修复这些缺陷并利用社区的帮助。对于闭源软件，Trudy 可以使用黑盒测试技术来寻找安全缺陷，但这可能比查找开源软件的安全缺陷更困难。Alice 需要依赖软件供应商来保证闭源软件的安全性。开源软件和闭源软件的固有安全性没有绝对的好坏之分，软件的安全性更多地取决于开发和维护的过程，而不仅仅是源代码是否开放。

第二章

1.

假设使用凯撒密码加密，请找出与下面密文信息相对应的明文
VSROJHEREVTXDUHSDOWU



具体移动数量未知：以最常见移动三个位置为例：

SPINDECRYPTSAREFUN

"旋转解密很有趣"

2.

假设我们拥有一台计算机，每秒钟能够执行 2^{40} 个密钥的测试。

- 如果密钥空间的大小为 2^{88} ，那么通过穷举式密钥检索找到密钥，预计需要多长时间（以年为单位）？
- 如果密钥空间的大小为 2^{112} ，那么通过穷举式密钥检索找到密钥，预计需要多长时间（以年为单位）？
- 如果密钥空间的大小为 2^{256} ，那么通过穷举式密钥检索找到密钥，预计需要多长时间（以年为单位）？



a. 如果密钥空间的大小为 2^{88} ，那么总的可能性是 2^{88} 。因此，通过穷举式密钥检索找到密钥，预计需要的时间（以年为单位）是 $2^{88} / 2^{40} / 31_536_000$ 。这大约等于 0.00038 年，或者大约 3.33 小时。

b. 如果密钥空间的大小为 2^{112} ，那么总的可能性是 2^{112} 。因此，通过穷举式密钥检索找到密钥，预计需要的时间（以年为单位）是 $2^{112} / 2^{40} / 31_536_000$ 。这大约等于 $97,343$ 年。

c. 如果密钥空间的大小为 2^{256} ，那么总的可能性是 2^{256} 。因此，通过穷举式密钥检索找到密钥，预计需要的时间（以年为单位）是 $2^{256} / 2^{40} / 31_536_000$ 。这大约等于 $1.158e+58$ 年，这是一个极其巨大的数字，远远超过了宇宙的年龄。

3.

加密如下消息：

we are all together

用4行乘4列的双换位密码（即本书中描述的那类双换位密码）进行加密，使用的行置换如下：

$(1,2,3,4) \rightarrow (2,4,1,3)$

使用的列置换如下：

$(1,2,3,4) \rightarrow (3,1,2,4)$



双换位密码是一种加密方法，它使用两次置换（行置换和列置换）来加密消息。首先，我们将消息分割成一个4x4的矩阵，然后按照给定的行置换和列置换进行操作。

给定的消息是 "we are all together"，我们可以将其分割成一个4x4的矩阵如下：

```
w e a r
e a l l
t o g e
t h e r
```

然后，我们按照给定的行置换 $(1,2,3,4) \rightarrow (2,4,1,3)$ 进行操作，得到：

```
e a l l
t h e r
w e a r
t o g e
```

接着，我们按照给定的列置换 $(1,2,3,4) \rightarrow (3,1,2,4)$ 进行操作，得到：

```
l e a l
e t h r
a w e r
g t o e
```

最后，我们将这个矩阵转换回字符串，得到加密后的消息 "leal ethr awer gtoe"。

总之先转为4*4的矩阵再利用规则进行转换即可

4.

考虑“单向”函数 h 。然后，给定值 $y=h(x)$ ，从 y 直接找到 x 是不可行的。请思考如下问题：

- 假设Alice可以计算 $y=h(x)$ ，其中 x 是Alice的薪酬，单位是美元。如果Trudy得到了 y ，那么她如何能够确定Alice的薪酬 x 呢？提示：针对这个问题，使用前向检索攻击。
- 为什么你的攻击没有违背 h 的单向特性？
- 请问，Alice如何能够防止此类攻击？我们假设Trudy能够访问到函数 h 的输出，Trudy也知道输入中包含了Alice的薪酬，并且Trudy还了解输入的格式。另外，这里没有密钥可用，所以Alice不能够对输出的值进行加密。



- Trudy可以通过前向检索攻击来确定Alice的薪酬 x 。这意味着Trudy可以尝试所有可能的薪酬值 x ，计算 $h(x)$ ，并将结果与已知的 y 进行比较。如果 $h(x)$ 等于 y ，那么Trudy就找到了Alice的薪酬 x 。
- 这种攻击没有违背 h 的单向特性，因为Trudy并没有从 y 直接计算出 x 。相反，她尝试了所有可能的 x 值，直到找到一个使得 $h(x)$ 等于 y 的 x 值。这是一个穷举搜索，而不是逆向计算。
- Alice可以通过增加输入的复杂性来防止此类攻击。例如，她可以将其他随机数据（如日期、时间戳等）与她的薪酬一起作为输入。这样，Trudy就不能只通过尝试所有可能的薪酬值来找到正确的 x ，因为她还需要知道其他的输入数据。这大大增加了穷举搜索的复杂性，使得攻击变得不可行。

遍历 $h(x)$

因为是穷尽 x

增加输入的复杂性

5.

假设 Alice用一种安全的加密方案加密了一条消息，该方案使用40位的密钥。Trudy知道密文和加密算法，但是她不知道明文和密钥。Trudy计划实施一次穷举式检索攻击，也就是说，她打算尝试每一个可能的密钥，直到她能够找到那个正确的密钥。请思考如下问题：

- 平均而言，Trudy在找到那个正确的密钥之前要尝试多少个密钥呢？
- Trudy 如何才能知道她找到了那个正确的密钥呢？注意：对于 Trudy 来说，因为太多的选择，所以不可能手工去检查每一个密钥，她必须有一些自动化的手段来判定假设的密钥正确与否。
- 在b的测试中，你的自动测试的工作量多大？
- 在b的测试中，你预计会有多少错误警报？也就是说，由不正确的密钥产生的假想的解密结果

能够通过测试的概率多大？



- a. 密钥是 40 位的，所以可能的密钥数量是 2^{40} 。平均而言，Trudy 需要尝试一半的密钥才能找到正确的密钥，所以她需要尝试大约 2^{39} 个密钥。
- b. Trudy 可以通过检查解密的结果是否符合某种预期的模式（例如，是否符合正常的文本格式，是否包含特定的词汇等）来判断她是否找到了正确的密钥。这需要一个自动化的过程，例如使用一个脚本来尝试每个密钥并检查解密结果。
- c. 自动测试的工作量取决于 Trudy 如何实现自动化的检查过程。在最坏的情况下，她可能需要对 2^{40} 个可能的密钥进行解密和检查操作。这是一个非常大的数字，所以工作量将会非常大。
- d. 错误警报的数量取决于 Trudy 的检查过程的准确性。如果她的检查过程非常宽松（例如，只要解密结果包含任何可打印的字符，就认为密钥可能是正确的），那么错误警报的数量可能会非常大。如果她的检查过程非常严格（例如，只有解密结果完全符合某种特定的格式，才认为密钥可能是正确的），那么错误警报的数量可能会非常小。在没有更多信息的情况下，很难准确估计错误警报的数量。

第三章

1.

流密码加密方案可以看成一次性密码本方案的推广。我们在前面曾介绍过，一次性密码本方案是可证明为安全的。那么，我们为什么不能采用之前用于一次性密码本的论据，来证明流密码加密方案也是可证明为安全的呢？

一次性密码本（**One-Time Pad, OTP**）确实是被证明为完全安全的加密方案，但是它的安全性基于两个关键的假设：

密钥是真正随机生成的。

密钥只使用一次。

流密码（**Stream Cipher**）可以看作是一次性密码本的推广，它使用一个密钥和一个初始化向量（**IV**）通过某种算法生成一个伪随机数流，然后将这个伪随机数流与明文进行异或操作来得到密文。

然而，流密码并不能满足一次性密码本的两个关键假设：

在流密码中，密钥并不是真正随机的，而是通过某种确定性的算法生成的。这意味着如果攻击者知道了这个算法，他们就有可能通过穷举搜索或其他方法猜测出密钥。

在流密码中，同一个密钥可能会被用来加密多个消息。如果两个消息使用了同一个密钥和同一个初始化向量，那么攻击者就可以通过分析这两个消息的密文来获取一些关于明文的信息。

因此，我们不能直接将一次性密码本的安全性论证应用到流密码上。流密码的安全性需要通过其他方式来证明，例如通过证明其生成的伪随机数流的随机性和不可预测性。

总结就是密码本加密是真正随机的且使用一次，但是流密码不随机而且多个加密的消息可能会导致消息比对得出加密信息

2.

假如Alice使用流密码加密方案加密明文P，得到了密文C，然后Alice发送C给Bob.

假设Trudy碰巧知道明文P，但是Trudy并不知道该流密码加密方案所使用的密钥K。

a.请说明：Trudy可以非常容易地确定用于加密P的密钥流。

b.请说明：实际上，Trudy能够用她选择的明文，比如P，替换掉P。也就是说，证明Trudy能够创建密文消息C，这样当Bob对C解密之后他将得到P

a. 在流密码中，明文P通过与密钥流K进行异或操作得到密文C。如果Trudy知道明文P和密文C，她可以通过将P和C进行异或操作来得到密钥流K。这是因为在二进制中，异或操作是可逆的。即，如果 $P \oplus K = C$ ，那么 $C \oplus P = K$ 。

b. 一旦Trudy得到了密钥流K，她就可以用它来加密任何她选择的明文P'。她只需要将P'与K进行异或操作，就可以得到新的密文C'。当Bob接收到C'并用K进行解密时，他将得到P'，而不是原始的明文P。这是因为 $P' \oplus K = C'$ ，所以 $C' \oplus K = P'$ 。

这就是为什么在流密码中，如果密钥流被重复使用，或者如果攻击者可以得到明文和对应的密文，那么系统的安全性就会被严重破坏。

再次异或即可获得密钥流

3.

在单独的一轮计算中，DES算法同时运用了扰乱和扩散原则

- a.请给出在DES算法的轮运算中对扰乱原则的应用。
- b.请给出在DES算法的轮运算中对扩散原则的应用。



a. 在DES算法的轮运算中，扰乱原则主要通过置换（**Permutation**）和选择（**Selection**）来实现。置换是将数据块的各个位重新排列，而选择是从输入中选择特定的位。例如，DES的初始置换（**IP**）和逆初始置换（**IP⁻¹**）就是扰乱操作，它们在整个加密和解密过程的开始和结束阶段进行。此外，每轮运算中的S盒（**Substitution boxes**）也是扰乱操作，它们将6位输入转换为4位输出，过程中进行了非线性变换。

b. 在DES算法的轮运算中，扩散原则主要通过扩展置换和异或操作来实现。扩展置换是在每轮运算的开始，将右半部分32位扩展为48位，使其与48位的轮密钥进行异或操作。异或操作确保了输入的每一位变化都会影响到输出的多个位，从而实现了扩散。此外，P盒（**Permutation box**）在每轮运算的末尾进行，它将S盒的输出进行置换，确保了每个S盒的输出都会影响到下一轮的多个S盒的输入，这也是扩散原则的体现。

扰乱原则：

置换+选择

扩散原则：

置换+异或

4.

DES算法要对最后一轮运算的输出互换位置，也就是说，最终的密文不是 $C = (L_{16}R_{16})$ ，而是 $C = (R_{16}, L_{16})$ 请问这个互换操作的目的是什么？

为了使得解密过程与加密过程尽可能地对称。

在解密过程中，我们使用相同的步骤，但是密钥的顺序是反向的。如果我们没有在最后一轮进行位置互换，那么在解密过程的第一步，我们需要先进行一次额外的位置互换。通过在加密过程的最后一步进行位置互换，我们可以避免这个额外的步骤，使得加密和解密过程更加对称。

值得注意的是，这个位置互换并不影响DES的安全性，因为它只是改变了密文的两个部分的顺序，而没有改变任何实际的信息。

使得加解密的过程更加对称：毕竟是对称加密（加密密钥和解密密钥一致）

5.

假设我们给出使用168位密钥的三重DES（3DES）的定义如下：

$$C = E(E(E(P, K), K_2), K)$$

假设我们能够计算并存储大小为256的表，而且还可以选择使用明文攻击。请证明，这个三重DES（3DES）并不比常规的3DES更安全，常规的三重DES仅仅使用112位的密钥。

提示：可以使用模拟针对双重DES算法的中间人攻击。

在这个问题中，我们需要证明使用168位密钥的三重DES（3DES）并不比使用112位密钥的常规3DES更安全。这是因为我们可以使用类似于针对双重DES的中间人攻击的方法来攻击这个三重DES。

在双重DES的中间人攻击中，攻击者会创建两个表：一个是使用所有可能的密钥K加密一个已知的明文P得到的中间结果，另一个是使用所有可能的密钥K解密一个已知的密文C得到的中间结果。然后，攻击者会在这两个表中寻找匹配的中间结果，从而找到正确的密钥。

对于这个三重DES，我们可以使用类似的方法。首先，我们可以使用所有可能的密钥K对一个已知的明文P进行两次加密，得到一个中间结果。然后，我们可以使用所有可能的密钥K2对一个已知的密文C进行一次解密，得到另一个中间结果。最后，我们在这两个表中寻找匹配的中间结果，从而找到正确的密钥K和K2。

因此，这个三重DES并不比常规的3DES更安全，尽管它使用了更长的密钥。这是因为我们可以使用类似的中间人攻击来破解它，而这种攻击的复杂性只与密钥的长度有关，而不是加密的次数。

双重DES中间人攻击就是 两个表加解密进行对比，找到密钥

三重DES中间人攻击就是 两个表一个加密两次和另一个解密一次进行对比 从而找到密钥

两种中间人攻击手段几乎一致：所以不一定更安全，安全性取决于密钥的长度

第四章

1.

这道思考题主要针对数字证书（即公钥证书）

- a. 请问，数字证书必须包含的信息有哪些？
- b. 请问，数字证书可能包含的附加信息有哪些？
- c. 为什么说，对于数字证书，最小化其中包含的信息的数量，会是个好主意呢？

a. 数字证书必须包含的信息主要有：

主体的公钥

主体的名称

证书的序列号

签发证书的证书颁发机构（CA）的名称

证书的有效期限

证书颁发机构的数字签名

b. 数字证书可能包含的附加信息主要有：

主体的电子邮件地址

主体的组织名称和单位名称

主体的地理位置信息（如国家、省份、城市等）

证书的用途（如服务器身份验证、客户端身份验证、代码签名等）

公钥的算法和参数

证书策略标识符

证书的扩展字段（如主体备用名称、颁发者备用名称、密钥用途等）

c. 对于数字证书，最小化其中包含的信息的数量是个好主意，主要有以下几个原因：

安全性：减少了可能被攻击者利用的信息，降低了安全风险。

隐私保护：减少了可能泄露的个人或组织的敏感信息，有利于保护隐私。

效率：减少了证书的大小，可以提高网络传输和证书处理的效率。

简洁性：使得证书更加简洁，便于管理和理解。

基础信息就是证书信息

附加信息就是主体（拥有者）的信息

最小化利于安全，隐私和效率以及简洁性

2.

在实施加密时，公开密钥加密系统的操作方式就类似于基于ECB（电子密码本）模式的分组加密方案。也就是说，明文消息会被截成多个分组，再对每个分组实施独立的加密操作。请思考下面的问题

a. 为什么说，采用分组密码加密方案实施加密时，ECB模式不是个好主意呢？为什么使用链模式，如CBC模式，对于分组密码加密方案，其优势有多大呢？当使用公开密钥加密方案时，为什么采用任何类型的链模式都是不必要的呢？你关于b的原因分析是否也适用于分组密码加密方案呢？请说明为什么？

a. ECB模式（电子密码本模式）对于分组密码加密方案来说不是一个好主意，主要原因是它没有提供足够的安全性。在ECB模式中，相同的明文块会被加密为相同的密文块，这使得攻击者可以通过观察密文的模式来推测出一些信息。例如，如果一个图片的某个区域的颜色是一致的，那么在ECB模式下，这个区域的密文也会是一致的，从而泄露出图片的一些信息。

使用链模式，如CBC模式（密码块链模式），对于分组密码加密方案的优势主要在于它提供了更好的安全性。在CBC模式中，每个明文块在加密前会与前一个密文块进行异或操作，这使得即使两个明文块相同，它们的密文块也会不同。这样就避免了ECB模式中的问题，提高了安全性。

当使用公开密钥加密方案时，采用任何类型的链模式都是不必要的，主要原因是公开密钥加密通常用于加密小量的数据，如会话密钥或数字签名，而不是大量的数据。因此，没有必要将数据分成多个块，并进行链式加密。

b. 对于分组密码加密方案，上述的原因分析同样适用。无论是对称密钥加密还是公开密钥加密，只要是在处理大量的数据时，使用链模式都可以提高安全性。而对于处理小量的数据，如公开密钥加密，使用链模式则可能是不必要的。

电子密码本不一定安全

使用公开密钥加密方案时，采用任何类型的链模式都是不必要的

3.

假设Alice的RSA公钥是 (e, N) ，她的私钥是 d 。Alice想要对消息 M 实施签名，也就是说，她想要计算 $[M]_{\text{Alice}}$ 。请列出她会用到的数学表达式

$$S = M^d \bmod N$$

RSA算法最重要的就是取模运算了

4.

假设Alice的RSA公钥是 $(N, e) = (33, 3)$ ，她对应的私钥是 $d=7$ 。请考虑下面的问题。如果Bob使用Alice的公钥加密消息 $M=19$ ，那么请计算对应的密文 C 是什么？

并请证明Alice能够解密密文 C ，得到明文 M 。

b. 假设 S 表示Alice对消息 $M=25$ 实施数字签名计算的结果，那么 S 的值是什么？

如果Bob收到了消息 M 和相应的签名 S ，请说明Bob验证该数字签名的过程，并请证明在这个特定的案例中，数字签名的验证能够成功通过。



a. Bob使用Alice的公钥 $(N, e) = (33, 3)$ 加密消息 $M = 19$ ，他会使用下面的公式来计算密文 C ：

$$C = M^e \bmod N$$

将 $M = 19$ ， $e = 3$ ， $N = 33$ 代入公式，我们得到：

$$C = 19^3 \bmod 33 = 6859 \bmod 33 = 16$$

所以，密文 $C = 16$ 。

Alice可以使用她的私钥 $d = 7$ 来解密密文 C ，她会使用下面的公式来计算明文 M ：

$$M = C^d \bmod N$$

将 $C = 16$ ， $d = 7$ ， $N = 33$ 代入公式，我们得到：

$$M = 16^7 \bmod 33 = 268435456 \bmod 33 = 19$$

所以，Alice能够成功解密密文 C ，得到明文 $M = 19$ 。

b. Alice对消息 $M = 25$ 进行数字签名，她会使用下面的公式来计算签名 S ：

$$S = M^d \bmod N$$

将 $M = 25$ ， $d = 7$ ， $N = 33$ 代入公式，我们得到：

$$S = 25^7 \bmod 33 = 6103515625 \bmod 33 = 4$$

所以，签名 $S = 4$ 。

Bob收到了消息 $M = 25$ 和相应的签名 $S = 4$ ，他可以使用Alice的公钥 $(N, e) = (33, 3)$ 来验证这个数字签名。他会使用下面的公式来计算 M' ：

$$M' = S^e \bmod N$$

将 $S = 4$, $e = 3$, $N = 33$ 代入公式, 我们得到:

$$M' = 4^3 \bmod 33 = 64 \bmod 33 = 25$$

因为 $M' = M$, 所以数字签名的验证成功通过。

公钥加密:

$$C = M^e \bmod N$$

私钥解密:

$$M = C^d \bmod N$$

签名私钥加密:

$$S = M^d \bmod N$$

签名公钥解密:

$$M' = S^e \bmod N = M$$

5.

请回顾一下, 对于RSA公钥系统, 为所有的用户选择相同的加密指数 e 是可能的。出于性能方面的考虑, 有时会选用公共的值 $e=3$ 。假定现在的情况就是如此, 请考虑下面的问题。

- 什么是针对RSA加密体制的立方根攻击? 在什么情况下该攻击能够成功
- 请给出两种不同的防止立方根攻击的方法。需要注意的是, 你的两种提议应该满足: 在未使用公共的加密指数 $e=3$ 的情况下, 仍然能够确保性能获得有效提升



a. 立方根攻击是针对RSA加密体系的一种攻击方式。当公开的加密指数 $e=3$ 时, 如果三个不同的接收者收到了同一条消息的加密版本, 攻击者可以利用中国剩余定理 (Chinese Remainder Theorem) 来恢复原始的明文消息。这种攻击方式在以下情况下能够成功: 发送者发送了相同的消息给三个使用不同公钥的接收者, 而这三个公钥的公开加密指数 e 都是3。

b. 防止立方根攻击的两种方法:

使用更大的公开加密指数 e : 虽然 $e=3$ 可以提高加密和解密的效率, 但是它也使得系统更容易受到立方根攻击。因此, 可以选择一个稍大的质数作为公开加密指数 e , 如 $e=65537$ 。这个数仍然是一个相对较小的质数, 可以保持加密和解密的效率, 同时也可以防止立方根攻击。

使用随机填充: 在加密消息之前, 可以向消息中添加一些随机数据, 使得即使是相同的消息, 加密后的密文也会不同。这样, 攻击者就无法通过比较不同接收者收到的密文来恢复原始的明文消息。这种方法被称为随机填充, 是RSA加密中常用的一种技术, 如OAEP (Optimal Asymmetric Encryption Padding)。

RSA的立方根攻击使用中国剩余定理可以获得原文信息

防范:

- 1.更大的加密指数 e
- 2.向信息中随机填充一些随机数据

即是: 更大指数法和随机数据法

6.

本思考题针对的是Diffie-Hellman密钥交换体制

- a.对于 g 的选择来说, 为什么 $g=1$ 是不允许的?
- b.对于 g 的选择来说, 为什么 $g=p-1$ 是不允许的?



a. 在Diffie-Hellman密钥交换体制中, g 是生成元, 它是在一个大质数 p 的模下的原根。如果选择 $g=1$, 那么所有的 g 的幂都将等于1 (因为1的任何幂都是1), 这意味着所有的可能的密钥都将是1, 这显然是不安全的。

b. 同样的, 如果选择 $g=p-1$, 那么 g 的偶数次幂将等于1 (因为 $(p-1)^2 \bmod p = 1$), g 的奇数次幂将等于 $p-1$ (因为 $(p-1)^1 \bmod p = p-1$)。这意味着所有的可能的密钥只能是1或者 $p-1$, 这同样是不安全的。因此, $g=p-1$ 也是不被允许的。

diffie-hellman加密体系中 g 是生成元: 若 $g=1$ 则所有指数都为1

若 $g=p-1$ 那么所有的偶次幂都将是1, 同样不安全, 故不被允许

7.

在RSA公钥加密方案中, 有时会使用通用的加密指数 $e=3$ 或 $e=2^{16}+1$ 。从理论上讲, 如果我们使用公共的解密指数, 比方说 $d=3$, 那么RSA体制也仍将有效运行。请问, 为什么使用 $d=3$ 作为公共解密指数并不是一个好主意? 你是否能够为RSA方案找到安全的公共解密指数 d ? 请详细说明。

在RSA公钥加密方案中，使用公共的解密指数 d 并不是一个好主意，原因如下：

安全性问题：解密指数 d 是基于公钥 (N, e) 和私钥 $\phi(N)$ 的关系计算出来的，其中 $\phi(N)$ 是模数 N 的欧拉函数。如果 d 是公开的，那么攻击者可以通过已知的 (N, e, d) 来计算 $\phi(N)$ ，进而可能分解出模数 N 的因数，从而破解RSA加密。

唯一性问题：对于给定的公钥 (N, e) ，解密指数 d 是唯一的。这意味着，如果我们想要使用一个公共的解密指数 d ，那么我们必须为每一个用户选择一个特定的公钥 (N, e) ，使得对应的解密指数 d 是我们预先设定的值。这在实际中是非常困难的，因为我们不能直接控制解密指数 d 的值。

因此，从理论和实践的角度来看，我们不能为RSA方案找到安全的公共解密指数 d 。在RSA加密方案中，解密指数 d 应该是每个用户独有的，且保密的。

不安全：容易破解

8.

在RSA加密系统中，有可能 $M=C$ ，也就是说，明文消息和密文消息可能相同。

a. 在现实中，这是否是个安全隐患呢？

b. 对于模数 $N=3127$ 以及加密指数 $e=17$ ，请找到至少一条消息 M ，使得对其加密后得到自身。

a. 在RSA加密系统中，明文消息 M 和密文消息 C 可能相同，但这种情况非常罕见。这并不构成一个安全隐患，因为即使攻击者知道存在这样的消息，他们也无法利用这个信息来破解RSA加密。这是因为，要找到这样的消息 M ，需要知道私钥 d ，而私钥 d 是保密的。

b. 对于模数 $N=3127$ 和加密指数 $e=17$ ，我们需要找到一个消息 M ，使得 $M^e \bmod N = M$ 。这需要进行一些试验和计算。在这种情况下，一个简单的方法是尝试一些小的数值。例如，如果我们尝试 $M=2$ ，我们得到：

$$2^{17} \bmod 3127 = 131072 \bmod 3127 = 2$$

所以， $M=2$ 是一个满足条件的消息。

即使机密信息等于原文，也依旧安全，毕竟不知道私钥 d

已知 $(3127, 17)$ 加密指数为17 得 2的加密结果就是2

第五章

1.

请论证如下关于加密哈希函数的陈述。

- a. 具备抗强碰撞特性必然就能够具备抗弱碰撞特性。
- b. 具备抗强碰撞特性并不一定就具备单向特性。



1. a. 这个陈述是正确的。抗强碰撞特性意味着对于所有可能的输入对 (a, b) ，其中 $a \neq b$ ，哈希函数 $H(a)$ 和 $H(b)$ 碰撞的概率极低。抗弱碰撞特性则是指对于给定的输入 a ，找到另一个输入 b ($b \neq a$)，使得 $H(a) = H(b)$ 的概率极低。如果一个哈希函数具备抗强碰撞特性，那么它必然具备抗弱碰撞特性，因为抗强碰撞特性的要求更严格。

b. 这个陈述是正确的。抗强碰撞特性只保证了难以找到两个不同的输入，它们的哈希值相同。而单向特性则是指，给定哈希值 $H(a)$ ，难以找到原始输入 a 。一个哈希函数可能具备抗强碰撞特性，但如果给定哈希值 $H(a)$ ，可以通过某种方法（例如暴力搜索）找到原始输入 a ，那么这个哈希函数就不具备单向特性。

抗强碰撞表示碰撞概率低，那么自然具有抗弱碰撞特性

单向特性表示已知哈希值难以找到其原始输入

2.

假如一个安全的加密哈希函数生成的哈希值是 n 位二进制长度。请说明，一个强力攻击如何才能得以成功实施，其预计的计算开销是多少？



对于一个 n 位的加密哈希函数，一个强力攻击（也称为暴力攻击）通常会尝试所有可能的输入，直到找到一个输入，它的哈希值与目标哈希值相同。因为哈希函数的输出是均匀分布的，所以预计需要尝试 2^{n-1} 次输入才能找到一个碰撞。因此，预计的计算开销是 $O(2^n)$ ，这是指数级的开销，对于大的 n ，这个开销是非常大的。

时间复杂度为2的指数级：开销巨大

3.

请回顾一下，我们在计算数字签名时，既使用了公开密钥系统，也使用了哈希函数。

- 请详细描述数字签名的计算过程和验证过程。
- 假定这里用于计算和验证签名的公开密钥系统是不安全的，但是其中的哈希函数是安全的。请证明，在这种情况下，你能够伪造签名
- 假定这里用于计算和验证签名的哈希函数是不安全的，但是其中的公开密钥系统是安全的。请证明，在这种情况下，你能够伪造签名。



a. 数字签名的计算过程和验证过程如下：

计算过程：

首先，将消息 M 通过哈希函数 H 计算得到哈希值 h 。

然后，使用发送者的私钥 d 对哈希值 h 进行加密，得到数字签名 S 。

验证过程：

首先，将接收到的消息 M 通过同样的哈希函数 H 计算得到哈希值 h' 。

然后，使用发送者的公钥 e 对接收到的数字签名 S 进行解密，得到哈希值 h 。

最后，比较 h 和 h' 是否相同。如果相同，那么签名就是有效的。

b. 如果公开密钥系统是不安全的，那么攻击者可能能够找到发送者的私钥 d 。一旦攻击者找到了私钥 d ，他们就可以伪造任何消息的签名。具体来说，攻击者可以选择任何消息 M ，计算其哈希值 h ，然后使用私钥 d 对 h 进行加密，得到伪造的签名 S 。因此，即使哈希函数是安全的，如果公开密钥系统是不安全的，那么攻击者就能够伪造签名。

c. 如果哈希函数是不安全的，那么攻击者可能能够找到两个不同的消息 $M1$ 和 $M2$ ，它们的哈希值相同。具体来说，攻击者可以选择一个消息 $M1$ ，使用发送者的公钥 e 对 $M1$ 的哈希值进行解密，得到一个签名 S 。然后，攻击者可以将签名 S 和消息 $M2$ 一起发送。因为 $M1$ 和 $M2$ 的哈希值相同，所以接收者在验证签名时会认为签名是有效的。因此，即使公开密钥系统是安全的，如果哈希函数是不安全的，那么攻击者就能够伪造签名。

a, 将 M 哈希用私钥进行加密

然后对接收到的 M 再哈希然后使用公钥解密得到 h' 如果两则相同则表示有效

第六章

偏难：略过

第七章

1.

就像本章所讨论的，可以根据一些短语来生成强度相对比较高的口令

a. 请根据短语“Gentlemen do not read other gentlemen's mail”生成两个口令。

b. 请根据短语“Are you who you say you are”生成两个口令



a. 根据短语"Gentlemen do not read other gentlemen's mail", 我们可以生成以下两个口令:

取每个单词的首字母, 将所有字母转为小写, 并在每个字母之间添加一个特殊字符(例如, "@"), 得到: "g@d@n@r@o@g@m"

取每个单词的首字母, 将所有字母转为大写, 并在每个字母之间添加一个数字(例如, "1"), 得到: "G1D1N1R1O1G1M"

b. 根据短语"Are you who you say you are", 我们可以生成以下两个口令:

取每个单词的首字母, 将所有字母转为小写, 并在每个字母之间添加一个特殊字符(例如, "#"), 得到: "a#y#w#y#s#y#a"

取每个单词的首字母, 将所有字母转为大写, 并在每个字母之间添加一个数字(例如, "2"), 得到: "A2Y2W2Y2S2Y2A"

生成口令:

取每个单词首字母大小写以及中间添加其他字符

2.

这题看看了解了解就好了

请针对下面给出的每一个口令, 分别找出有可能生成这些口令的相应的短语。

a. PokeGCTall

b. 4s&7yrSa

c. gimmeliborD

d. lcntgetNOsat

- a. "PokeGCTa11" 可能来自于短语 "Pikachu on keyboard, Eevee goes, Charmander takes all."
- b. "4s&7yrsa" 可能来自于短语 "Four score and seven years ago."
- c. "gimmelibord" 可能来自于短语 "Give me liberty or Death."
- d. "IcntgetN0sat" 可能来自于短语 "I can't get no satisfaction."

3.

研究表明，大部分人都不能根据一幅照片来准确地识别出一个人。例如，有研究发现大部分人都会接受带有照片的ID证件，无论照片是谁，只要照片中的人像和证件持有者具有相同的性别和种族即可。另外，也有证据显示，当将照片引入到信用卡中时，错误接受率就会有很大下降。对于这种明显是自相矛盾的现象，请给出你的解释。作者本人曾经常常光顾一个游乐园，那个游乐园提供一种季度通行证，每个通行证持有者都配发类似于信用卡的塑料卡。这个游乐园拍下每一个用户的照片，但是照片并不出现在卡片上面。相反，当用户要进入游乐园并出示其卡片时，用户的照片就会出现在屏幕上，这时工作人员就可以看到并进行比对。请问，与将照片贴在卡片上的方式相比，为什么这种方案的效果可能会更好？

对于第一个问题，这种现象可能是因为人们在处理身份验证任务时，往往会受到社会和环境因素的影响。例如，他们可能不愿意或者不舒服去仔细检查照片和持卡人的相似度，特别是在公共场合，这可能会被视为不礼貌或者侵犯隐私。然而，当照片被引入到信用卡中时，这种情况就会改变。因为在这种情况下，检查照片成为了正常的、被期望的行为，人们可能会更愿意去仔细检查。

对于第二个问题，将照片显示在屏幕上的方式可能比将照片贴在卡片上的方式更有效，原因可能有以下几点：

照片的大小和清晰度：在屏幕上显示的照片可以比卡片上的照片大，清晰度也可能更高，这使得比对更容易。

实时更新：如果用户的外貌发生了变化，系统可以更新照片，而卡片上的照片则无法做到这一点。

防止欺诈：如果卡片丢失或被盗，别人可能会尝试使用它。但是，如果照片是在屏幕上显示的，那么即使他们有卡片，也无法通过验证。

往往不愿意仔细看持卡人的和照片是否长得一样（我觉得是因为往往都会有很大的变化）

4.

针对基于Webcookie的认证框架都有哪些攻击？请分别给出说明。



基于web cookie的认证框架可能会面临以下几种攻击：

Cookie 窃取：攻击者可能会通过各种方式窃取用户的 **cookie**，例如通过网络嗅探、跨站脚本攻击（XSS）等。一旦攻击者获取了用户的 **cookie**，他们就可以伪装成用户与服务器进行交互。

Cookie 重放攻击：在这种攻击中，攻击者会捕获用户的 **cookie**，然后在另一个会话中重放这个 **cookie**。由于服务器通常会信任 **cookie**，因此攻击者可以利用这个 **cookie** 来伪装成用户。

跨站请求伪造（CSRF）：在 **CSRF** 攻击中，攻击者会诱使用户点击一个链接或加载一个页面，这个链接或页面会包含一个向受害者网站发送请求的脚本。由于浏览器会自动发送与请求相关的 **cookies**，因此如果用户当前已经登录了受害者网站，那么这个请求就会以用户的身份被发送出去。

Cookie 篡改：如果 **cookie** 没有进行适当的保护，攻击者可能会修改 **cookie** 的内容。例如，攻击者可能会尝试修改 **cookie** 中的用户级别字段，以提升他们的权限。

为了防止这些攻击，可以采取一些措施，例如使用安全的连接（如 **HTTPS**）来传输 **cookie**，使用 **HttpOnly** 属性来防止通过脚本访问 **cookie**，使用 **SameSite** 属性来防止 **CSRF** 攻击，以及使用签名或加密来保护 **cookie** 的完整性和机密性。

cookie窃取/重放攻击/伪造/篡改

5.

近年来，人脸识别技术备受追捧，比方说，该技术被认为是在机场进行恐怖分子鉴别的可行方案之一。正如本书中提及的，人脸识别技术在拉斯维加斯赌场作为检测作弊者的一种尝试已被运用。请注意，在这两种情况下，生物特征技术都被用于鉴别（而不是认证），很可能会遇到不配合的情况。

- 请讨论，当用于赌场中检测作弊者的场景时，对于人脸识别技术都会有哪些攻击？
- 请讨论，针对你在a中提出的攻击方式，赌场可以采取何种对策来降低其功效。
- 请讨论，对于你在b中提出的对策，攻击者可以采取何种反对策来削弱其功效。



a. 在赌场中使用人脸识别技术检测作弊者时，可能会遇到以下攻击：

遮挡面部：作弊者可能会通过戴帽子、墨镜或者其他方式遮挡面部，使得人脸识别系统无法准确识别。

面部变形：作弊者可能会通过化妆、面部整形或者使用面部贴纸等方式改变面部特征，以欺骗人脸识别系统。

欺骗攻击：作弊者可能会使用高质量的面部图片、视频或者3D面部模型，试图欺骗人脸识别系统。

b. 针对上述攻击，赌场可以采取以下对策：

增强人脸识别技术：使用更先进的人脸识别算法，例如深度学习算法，可以提高对遮挡和变形面部的识别能力。

多模态生物特征认证：除了人脸识别，还可以使用其他生物特征，如指纹、虹膜或者行为特征等，提高识别的准确性和鲁棒性。

活体检测：通过检测面部的微小运动（如眨眼、嘴唇运动等）或者反射光等，可以防止欺骗攻击。

c. 对于上述对策，攻击者可能会采取以下反对策：

更复杂的面部遮挡和变形：攻击者可能会使用更复杂的遮挡和变形技术，例如使用专业的化妆和整形技术，或者使用更先进的面部贴纸等。

欺骗多模态生物特征认证：如果赌场使用多模态生物特征认证，攻击者可能会尝试欺骗所有的生物特征。

更复杂的欺骗攻击：对于活体检测，攻击者可能会使用更复杂的欺骗攻击，例如使用动态的面部图片或者视频，或者使用具有真实反射光的3D面部模型。

人脸欺骗 -> 增强面部识别 -> 更复杂的检测技术

第八章

1.

这个思考题针对的是视觉CAPTCHA技术。请描述一个本书中尚未提及的现实世界中的视觉CAPTCHA技术的例子，并说

明这个CAPTCHA实例的工作原理，也就是说，请解释程序是如何生成CAPTCHA验证码的，又是如何评判结果的。要通过这个测试，需要用户做些什么？

b.对于a中所述的CAPTCHA技术实例，攻击者有哪些可以利用的信息？



a. 一个常见的视觉CAPTCHA技术实例是谷歌的reCAPTCHA v2，它要求用户识别一组图片中的特定对象。

工作原理如下：

生成CAPTCHA：系统会随机选择一组图片，并选择一个主题（例如，“选择所有包含交通信号灯的图片”）。这些图片会以网格的形式显示给用户。

评判结果：用户需要点击所有包含指定对象的图片。一旦用户完成选择，系统会根据用户的选择来评判结果。如果用户正确地选择了所有相关的图片，那么测试就通过了。

b. 对于这种CAPTCHA技术，攻击者可能会利用以下信息：

图片内容：攻击者可以尝试使用图像识别技术来自动识别图片中的对象。

用户行为：reCAPTCHA v2还会分析用户的行为（例如，鼠标移动的速度和路径）来判断用户是否为人类。攻击者可能会尝试模拟人类的行为来欺骗系统。

谷歌的图片reCAPTCHA -> 图像识别技术

用户行为reCAPTCHA（鼠标移动速度） -> 模拟人类的行为

2.

假设一个包过滤防火墙对其所允许通过的每一个数据包，都将TTL字段的值重置为255。那么本章中所描述的Firewalk端口扫描工具将会无效。

a. 请问，为什么在这种情况下Firewalk会失效呢？

b. 请问，这里所提出的解决方案是否会引发什么新的问题？

c. 请问，是否可以对Firewalk加以修改，使其可以对付这样的防火墙呢？



a. Firewalk工作原理是通过发送具有特定TTL值的数据包，并观察它们是否到达目标或被中途丢弃，从而推断出防火墙的规则。如果防火墙将所有通过的数据包的TTL字段重置为255，那么Firewalk就无法通过观察TTL值的变化来推断出防火墙的规则，因此Firewalk会失效。

b. 尽管这种解决方案可以防止Firewalk端口扫描，但它可能会引发新的问题。例如，它可能会干扰那些依赖于TTL值的网络协议和服务。此外，它可能会增加防火墙的处理负载，因为防火墙需要修改每一个通过的数据包的TTL字段。

c. 对于这种防火墙，可能很难直接修改Firewalk来对付它。因为Firewalk的基本原理就是依赖于观察TTL值的变化来推断防火墙的规则，如果防火墙将所有通过的数据包的TTL字段重置，那么Firewalk就无法工作。然而，可能可以通过其他方式来推断出防火墙的规则，例如通过观察数据包是否被丢弃，或者通过尝试不同的数据包类型和端口号等。

第九章

1.

我们想要基于一个共享的对称密钥来设计一个安全的相互认证协议。我们还想要建立一个会话密钥，并且获得完全正向保密特性。

- a. 请利用三条消息设计这样的一个协议
- b. 请利用两条消息设计这样的一个协议

a. 三条消息的协议可以设计如下：

A → B: A, {Na, A}Kab

B → A: {Na, Nb, B}Kab

A → B: {Nb}Kab

这里，A和B是通信的两方，Na和Nb是A和B生成的随机数，用作一次性的会话密钥，Kab是A和B共享的对称密钥。{}表示加密操作。

b. 两条消息的协议可以设计如下：

A → B: A, {Na, A}Kab

B → A: {Na, Nb}Kab

这个协议的工作原理与上面的三条消息协议类似，但是它省略了第三步。在这个协议中，B在接收到A的消息后，直接返回一个包含A的随机数Na和自己的随机数Nb的加密消息。然后，A和B可以使用Na和Nb的组合作为会话密钥。这个协议的缺点是，它没有明确的方式来确认B已经接收到A的第三条消息，因此可能会对重放攻击更加敏感。

第十章

1.

SSL协议和IPSec协议的设计目的都是为互联网上的应用提供安全性保护

- a. 请问，SSL较之IPSec的主要优点是什么？
- b. 请问，IPSec较之SSL的主要优点是什么

a. SSL (Secure Sockets Layer) 相比IPSec的主要优点:

端到端安全: SSL在应用层提供安全性保护, 可以为特定的应用提供端到端的安全连接, 而不需要改变中间网络设备的配置。

灵活性: SSL可以用于各种应用, 如HTTP (HTTPS)、FTP、Email等, 而IPSec主要用于VPN等网络层的安全应用。

兼容性: 由于SSL在应用层工作, 因此它可以穿越防火墙和NAT设备, 而IPSec可能会遇到兼容性问题。

b. IPSec (Internet Protocol Security) 相比SSL的主要优点:

网络层安全: IPSec在网络层提供安全性保护, 可以为所有通过IPSec隧道的流量提供安全性保护, 而不仅仅是特定的应用。

透明性: 对于应用程序来说, IPSec是透明的, 应用程序不需要进行任何修改就可以利用IPSec提供的安全性保护。

完整性: IPSec提供了数据源认证、数据完整性和重放攻击防护等安全性保护, 而SSL主要提供了数据的机密性和服务器的认证。

ssl应用于各层 而ipsec用于网络层