

2/27

crypto, cryptography, crypto-analysis(attack)

代码要求:

- 写注释
- 由文件输入
- 有说明文档
- 可行则尽量参考优秀开源代码

Cryptography is everywhere

Secure communication:

- web traffic:
 - HTTPS:
 - No Eavesdropping (窃听, 被动 (破坏弱) ->难察觉);
 - No Tampering (“鼓捣”/篡改, 主动 (破坏强) ->蜜罐);
 - → Secure communication
- wireless traffic:
 - 802.11i, WPA2 (and WEP):
 - GSM:
 - Bluetooth:

Encrypting files on disk:

EFS: [Portal](#)

TrueCrypt: [Portal](#)

Content protection:

e.g.:

DVD:

Blu-ray:

CSS:

AACS:

User authentication

... and much much more

Secure Sockets Layer / TLS

Two main parts:

1. Handshake Protocol: **Establish shared secret key using public-key cryptography** (2nd part of course)
2. Record Layer: **Transmit data using shared secret key**. Ensure confidentiality and integrity (1st part of course)

Security Elements:

- Confidentiality->Cryptography responsible
- Integrity->Cryptography responsible
- Availability

Protected files on disk

Analogous to secure communication: Alice today sends a message to Alice tomorrow.

Building block: symmetric encryption

Alice → *message* → *Encrypt(key, message)* → *cipher* → *Decrypt(key, cipher)* → *message* → *Bob*

↑ — — — — — Same — — — — — ↑

Single use key: (one time key)

- Key is only used to encrypt one message
 - encrypted email: new key generated for every email

Multi use key: (many time key)

- Key used to encrypt multiple messages
 - encrypted files: same key used to encrypt many files
- Need more machinery than for one-time key

Things to remember

Cryptography is:

- A tremendous tool
- The basis for many security mechanisms

Cryptography is not:

- The solution to all security problems
- Reliable unless implemented and used properly
- Something you should try to invent yourself
 - many many examples of broken ad-hoc designs

Crypto magic

- Privately outsourcing computation: [安全多方计算, paper](#)
- Zero knowledge(proof of knowledge): [Portal](#)

A rigorous science

The three steps in cryptography:

- Precisely specify threat model
- Propose a construction
- Prove that breaking construction under threat mode will solve an underlying hard problem

Few Historic Examples (all badly broken)

1. Substitution cipher: Caesar Cipher(no key),
2. Vigenere cipher,
3. Rotor Machines,
4. Data Encryption Standard:
 - DES: # keys = 256, block size = 64 bits.
 - Today: AES (2001), Salsa20 (2008) (and many others)

Randomized Algorithm

$y \leftarrow A(m, r)$ where $r \xleftarrow{R} 0, 1^n$

and output is a random variable $y \xleftarrow{R} A(m)$.

just like $c = E(m, k)$ where k is random.

The birthday paradox

Let $r_1, \dots, r_n \in U$ be independent. identically distributed random vars.

Thm : when $n = 1.2 \times |U|^{1/2}$ then $Pr[\exists i \neq j : r_i = r_j] \geq \frac{1}{2}$.

notation: $|U|$ is the size of U .

Example: Let $U = \{0, 1\}^{128}$

After sampling about 264 random messages from U , some two sampled messages will likely be the same

信息论

1. 熵与疑义度

1.1 自信息

文字、图片、声音是**消息**，**信息**是消息的有价值内容。

定义：事件 x_i 的自信息，记作 $I(x_i)$ ，定义为

$$I(x_i) = -\log p_i$$

注意：自信息的定义没有规定对数的底！

- 对数底为2时，自信息单位为比特 (bit)；
- 对数底取为e时，自信息单位为奈特 (nat)；
- 对数底为10时，自信息单位为哈特 (hart)。

自信息度量了一个随机事件 x_i 未出现时所呈现的不确定性，也度量了该事件 x_i 出现后所给出的信息量。

事件的不确定性越大，则一旦出现给出的信息量也就越大。

1.2 熵

自信息描述了事件集 X 中一个事件出现给出的信息量，整个集 X 的平均信息量是该集所有事件自信息的统计平均值（数学期望），称作**集 X 的熵**。

集 X 的熵，记作 $H(X)$ ，定义为

$$H(X) = -\sum_{i=1}^N p_i \log p_i$$

定义中，规定 $0 \log 0 = 0$ 。

$H(X)$ 度量了集 X 中各个事件未出现时所呈现的平均不确定性(疑义度)，也度量了集 X 中一个事件出现时所给出的平均信息量。

疑义度：消息的熵同时也可衡量其不确定性（疑义度），即将消息隐藏在密文中时，要破译它所需的明文比特数（即当消息被加密成密文时，为了获取明文需要解密的明文的位数）。

例子

1. 甲任意取一个不超过15的整数，由乙来猜，但允许乙提K个问题，甲只回答“是”或者“非”，问K多大时可以确定猜到该数。

解：

若令乙猜想作为事件 V , V 可能有16种结果, 假定这16种结果是等概率的, V 的熵为：

$$H(V) = \log_2 16$$

令事件 $A_k = U_1 U_2 U_3 \dots U_k$ 为提问 k 个问题, 但 U_i 的熵不超过 $\log_2 2 = 1$, (因为只有“是”或者“非”), 故 A_k 的熵为不超过 k 比特, 则：

$$\log_2 16 \leq k \cdot \log_2 2 = k, k \geq 4$$

故 $k = 4$.

如何保证每次可以获得1位的信息呢？

最直接的四个问题：

1. 这个数被表示为四位二进制后, 第一位是0吗？
2. 这个数被表示为四位二进制后, 第二位是0吗？

.....

这样, 我们可以确保每次都可以得到一位信息。

2. 有25个外表完全相同的硬币, 其中24个重量完全一样, 有一个较轻的伪币, 用无砝码的天平, 试问要做多少次的比较, 可以找到这枚伪币？

解：

事件 V 为找出伪币, 可能有25个结论, 他们是等概率, 故：

$$H(V) = \log_2 25,$$

事件 U 为天平称的结果, 可能有3种情况：1.左右平衡；2.左边重；3.右边重. 故：

$$H(U) = \log_2 3,$$

令 $A_k = U_1 U_2 U_3 \dots U_k$ 为连续用 k 次天平的事件,

$$\begin{aligned} k \cdot \log_2 3 &\geq \log_2 25 \\ k &\geq (\log_2 25) / \log_2 3 = 2.93 \end{aligned}$$

故 k 最少为3次.

一种解决方案：

1. $25=8+8+9$ (第一次)
 - 天平两端各放8个, 如果平衡, 则伪币在剩余的9个之中, 跳到ii;

- 如果不平衡，则伪币在较轻的8个之中，跳到iii。

2. $9=3+3+3$ (第二次)

- 天平两端各放3个，如果平衡，则从剩下3个中寻找伪币。否则，从较轻的3个中寻找伪币。

3. $8=3+3+2$ (第二次)

- 天平两端各放3个，如果平衡，则从剩下2个中寻找伪币。否则，从较轻的3个中寻找伪币。

若不知道轻重呢？

GPT:

这个问题是“小球称重”问题的一个变种，其目的是在外观上相似的物体中，通过有限次称重找出异常的一个。在这个问题中，异常的物体是一枚重量未知的伪币。解决这种问题的关键是每次称重后都要有效地缩小搜索范围。

对于这类问题，由于我们并不知道异常物体更重还是更轻，数学上最少需要的称重次数，需要通过公式 $(\lceil \log_3(2N + 1) \rceil)$ 计算得出，这里的 (N) 代表物体的总数， $(\lceil \dots \rceil)$ 是向上取整的意思。在本题中，由于 $(N = 25)$ ，因此至少需要 $(\lceil \log_3(25 * 2 + 1) \rceil = 3)$ 次称重。所以，我们可以通过三次天平称重来找出那个重量未知的伪币。

接下来是具体的称重步骤：

首次称重：

把25个硬币分成三组，两组各8个，另一组9个，随机选两组中的两组进行称重。

情况A: 如果天平平衡，那么问题硬币在未称重的那组中。

情况B: 如果不平衡，那么问题硬币在称重的两组中。

对于情况A (问题硬币在未称重的那组中)：

第二次称重：把这9个硬币分成三组，每组3个，任选两组进行称重。

情况A: 如果天平平衡，问题硬币在未称重的那组中。

情况B: 如果不平衡，问题硬币在称重的两组中。

第三次称重：从疑似的3个硬币中选一个，和1个正常硬币称重。

情况A: 如果天平平衡，未称重的那1个硬币是伪币。

情况B: 如果不平衡，则天平上称重的那1个硬币是伪币。

对于情况B (问题硬币在称重的两组中)：

假设称重时重的一组是A组，轻的一组是B组。

第二次称重：分别从A组拿2个，B组拿1个，放在天平两端，进行称重。

情况A: 如果平衡，问题硬币在B组剩下的2个中，且我们可以知道，伪币更轻。

情况B: 如果不平衡，则伪币有两种可能性：

可能性B1: 天平上，较重的那端里，属于A组的两个硬币之一是伪币，并且，它比正常硬币重。

可能性B2: 天平上，较轻的那端里，属于B组的1个硬币，并且，它比正常硬币轻。

第三次称重：参考第二结果的两种情况，进行称重：

情况A: 如果第二次称重时，天平平衡，则从剩余2个疑似伪币中任选一个，和一个正常硬币称重。

情况A.1: 如果天平平衡，未称重的那个是伪币。

情况A.2: 如果不平衡，称重的那个是伪币。

情况B: 如果第二次称重时，天平不平衡，选择较重端，来自A组的2个硬币进行称重。

情况B.1: 如果天平平衡，则第二次称重中，可能性B2正确，说明伪币是，第二次称重中，天平较轻一端，来自B组的1个硬币，且伪币更轻。

情况B.2: 如果不平衡，则第二次称重中，可能性B1正确，说明伪币在本次称重的A组两个中，由之前分析可知，它比正常硬币更重，因此，伪币是本次称重中的较重一个。

通过这种方法，最多三次称重就能准确找出问题硬币。

2. 自然语言率和绝对语言率

自然语言率：对于给定的一种语言，其自然语言率为

$$r = H(M)/N$$

其中 N 为消息长度。

- 英语的自然语言率：1.0比特/字母~1.5比特/字母
- 它是一个语言系统的实际表现力，实际上是一个语言系统的实际熵。

绝对语言率：每个字符编码的最大比特数，这里假设每个字符序列出现的机会相等。

若语言中有 L 个字母，则绝对语言率为：

$$R = \log_2 L$$

为单个字母的最大熵。

- 英语的绝对语言率： $\log_2 26 \approx 4.7$ 比特/字母
- 它是一个语言系统理论上的最大表现力。当每个字符出现的概率相同时，其具有最大表现力。实际上是语言系统的最大熵。

冗余度：语言的冗余度记为 D ，定义为：

$$D = R - r$$

其中， R 为绝对语言率， r 为自然语言率。

英语： $r = 1.3$ 比特/字母，则 $D = 3.4$ 比特/字母。

3. 密码系统的安全性

绝对安全的密码系统：

- M : 明文空间; K : 密钥空间; C : 密文空间;

$$c = E(m, k). E : M \rightarrow C.$$

- $H(M), H(K)$
- 绝对保密的密码系统的必要条件:

$$H(K) > H(M)$$

密码系统的熵: 衡量密钥空间 K 的大小的一个标准, 通常是密钥数以 2 为底的对数。

$$H(K) = \log_2 k$$

4. 确定性距离

对于长度为 n 的消息, 能够将一段密文消息解密成与原始明文同种语言的可懂文本的密钥个数为: $2^{H(K)-nD} - 1$

确定性距离: 能够唯一地确定密钥的最短的密文长度的近似值。

- 对称密码系统的确定性距离: 定义为密码系统的熵除以语言的冗余度。

$$U = H(K) / D$$

理想安全的密码系统: 确定性距离无限大的密码系统。

5. 混乱与扩散

混乱: 在加密变换中, 让密钥与密文的关系尽可能复杂的做法。

- 实现混乱的方法: **代替**

扩散: 在加密过程中, 尽可能将明文的位置统计特性在密文中消除。

- 实现扩散的方法: **换位**