分组密码的工作模式及流密码

分组密码的工作模式

- 分组密码输入 b 位明文分组，输出 b 位密文分组；
- 若明文长度大于 b，则需要将明文分成 b 位一组的块；
- 每次使用相同的密钥对多个分组加密，会引发安全问题；
- 为了将分组密码应用于各种各样的应用，NIST 定义了五种“工作模式”；
- 本质上，工作模式是一项增强密码算法或者使算法适应具体应用的技术；
- 五种工作模式可使用包括 DES 和 AES 在内的任何分组密码算法。

主要内容

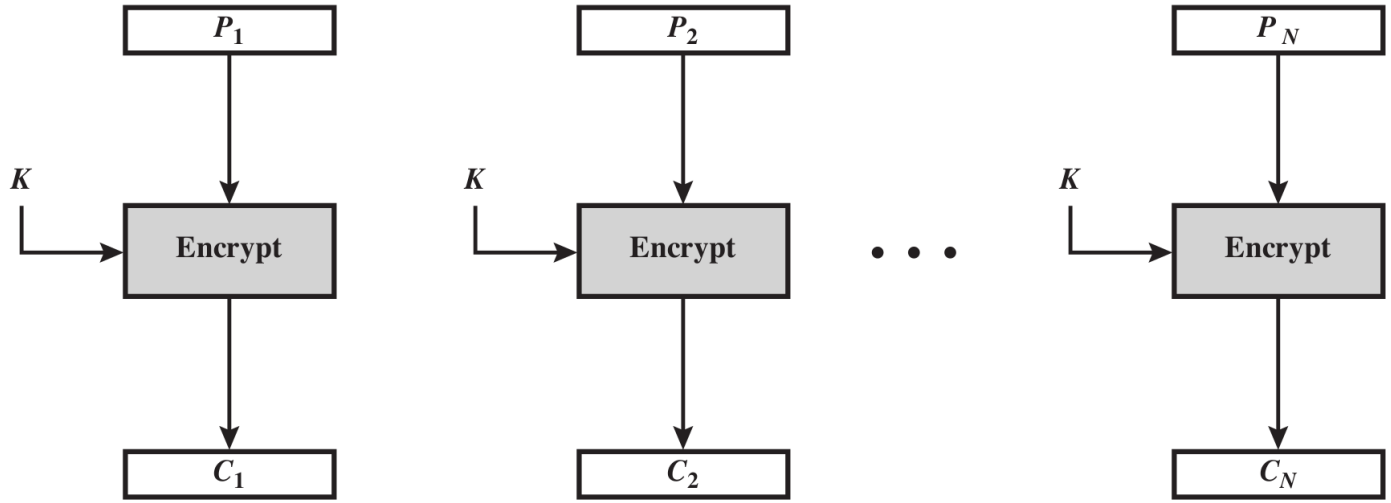
1. 分组密码的工作模式
2. 流密码

主要内容

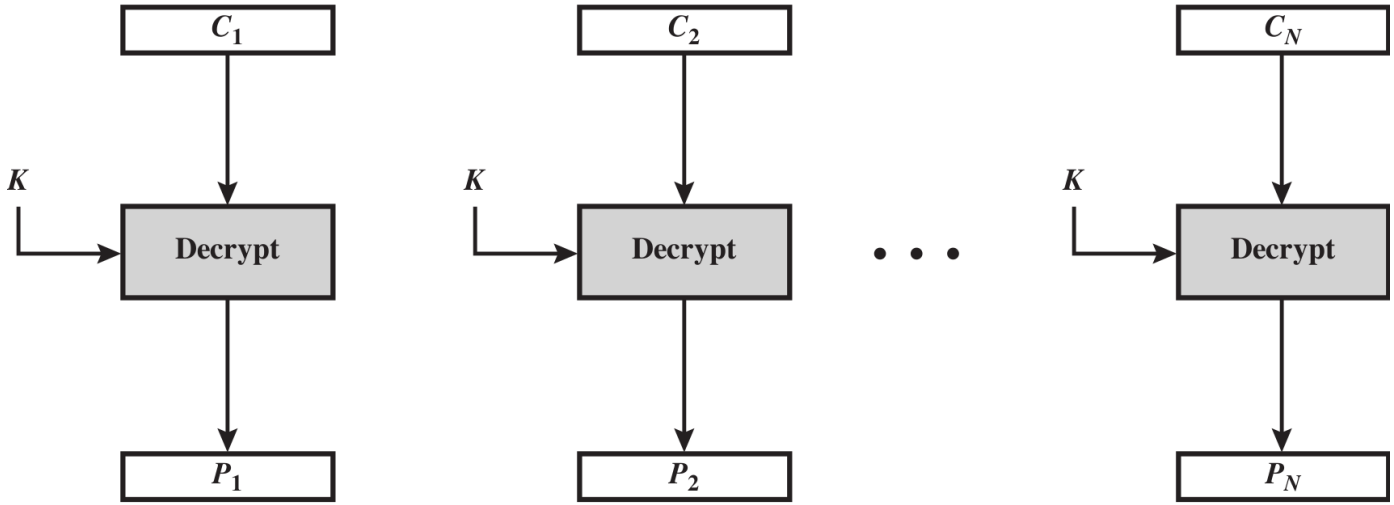
1. 分组密码的工作模式
2. 流密码

电码本（Electronic Codebook, ECB）

加密：



解密：

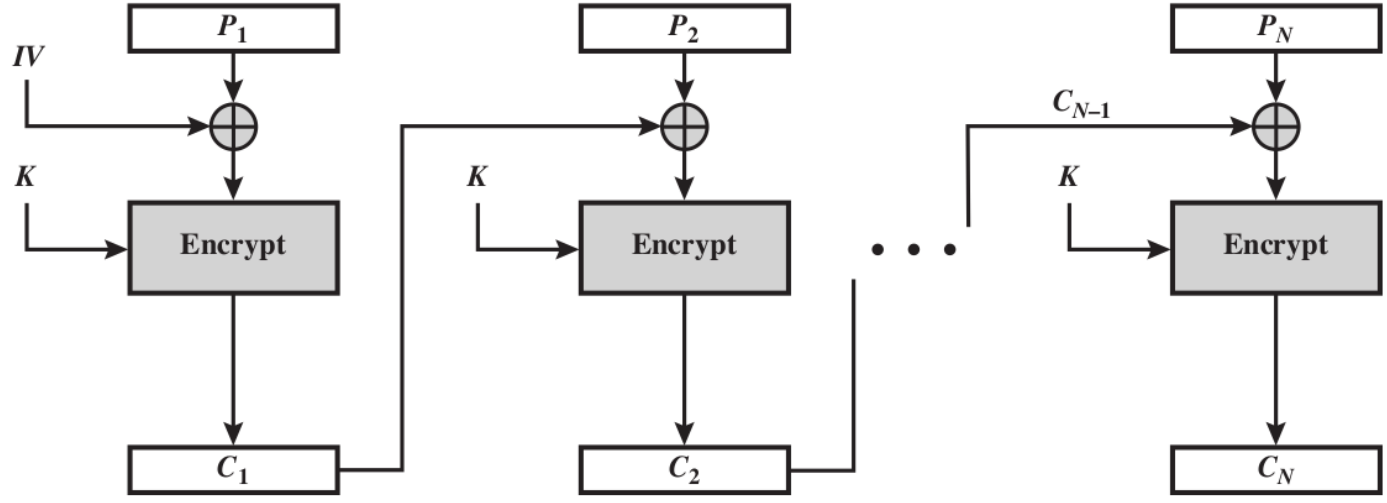


ECB 的局限性

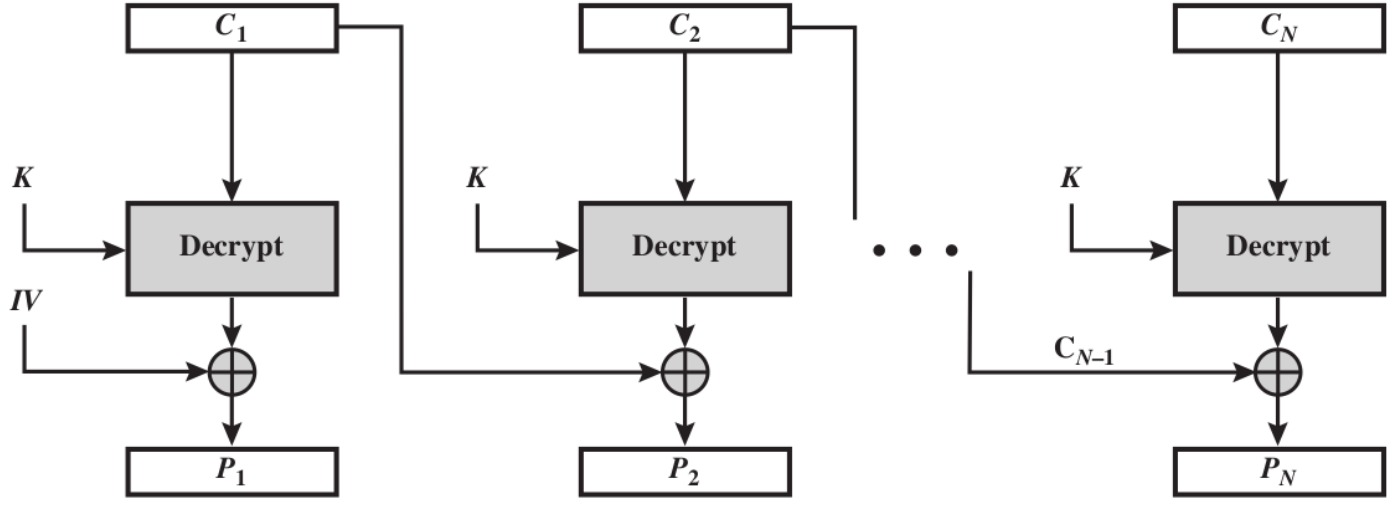
- ECB 模式特别适合数据较少的情况，例如传输 DES 密钥；
- 一段明文消息中若有几个相同的明文组，则密文也将出现几个相同的片段；
- 对于很长的消息，ECB 是不安全的，如果消息是非常结构化的，密码分析可能利用其结构特征来破解；
- ECB 的弱点来源于其加密过的密文分组互相独立。

密文分组链接（Cipher Block Chaining, CBC）

加密：



解密：

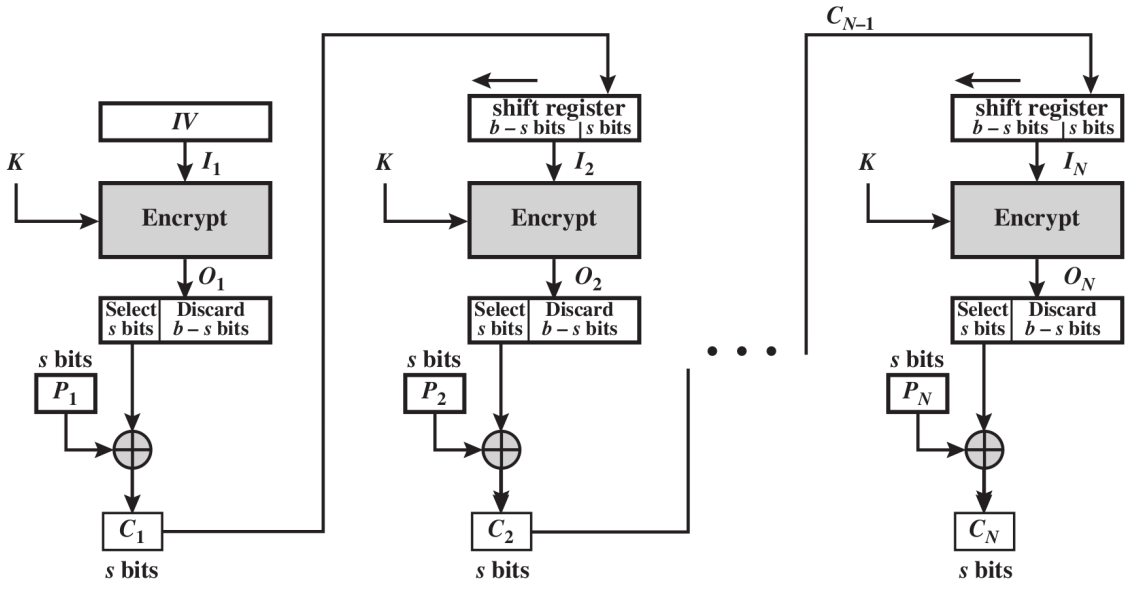


CBC 的优缺点

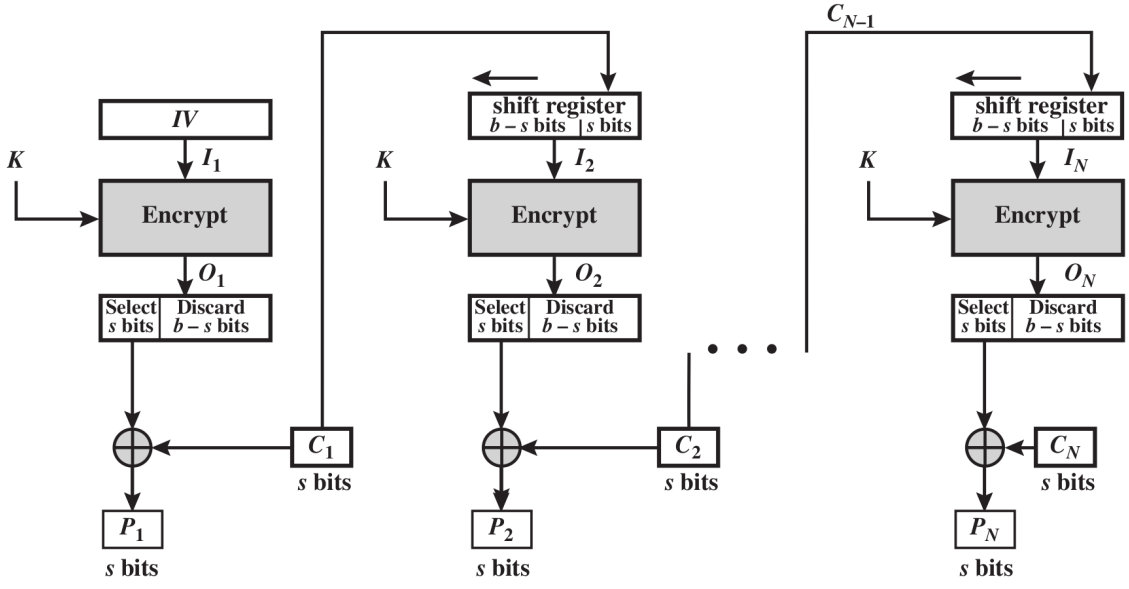
- 每个密文分组依赖于所有之前的明文分组；
- 明文消息中的任何一点变化都会影响所有的密文分组；
- 发送方和接收方需要共享初始向量（Initial Value, IV）：如果 IV 被明文传送，则攻击者可以改变第一个分组的某些位，然后预先改变 IV 中的某些位，则接收者收到的 P1 也就相应改变了；因此，IV 必须是一个固定的值 (as in EFTPOS) 或者必须用 ECB 方式在消息之前加密传送。
- 如果最后一个分组不是完整的分组，则需要填充：可以填充已知非数据值，或者在最后一块补上填充位长度，eg., [b1 b2 b3 0 0 0 0 5], i.e., 3 data bytes, then 5 bytes pad+count

密文反馈（Cipher Feedback, CFB）

加密：



解密：

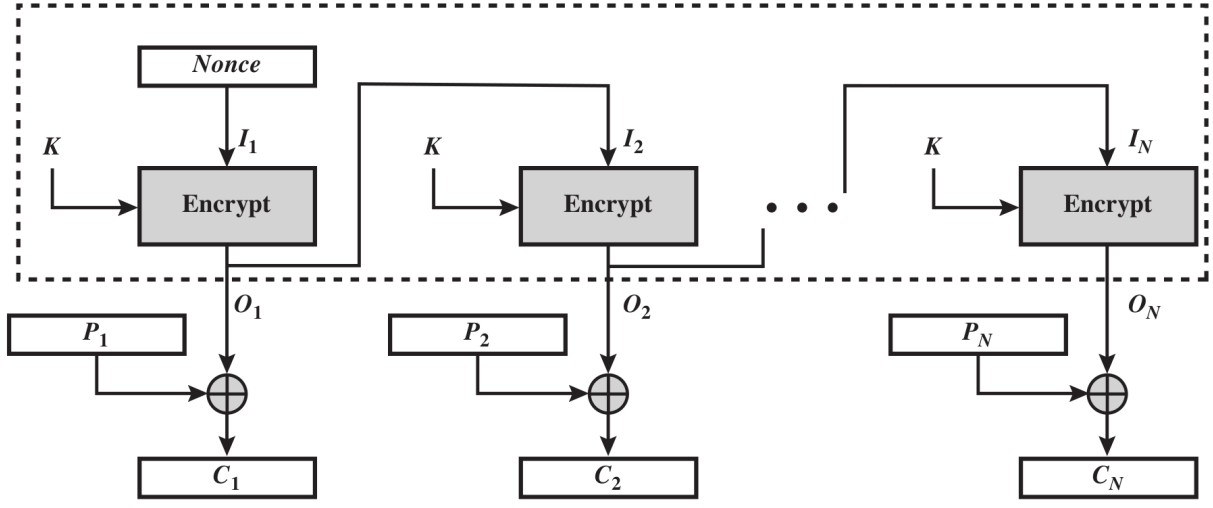


CFB 的优缺点

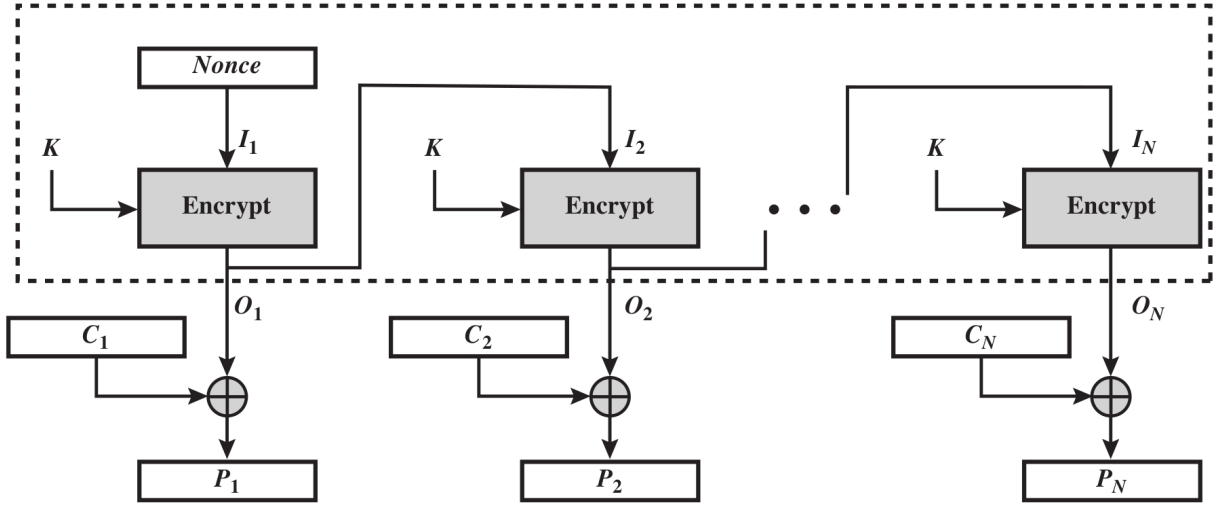
- 可以将分组密码转化成流密码的技术；
- 不再要求报文被填充成整个分组, 数据以位或字节形式到达时都适用；
- 加解密使用相同方案，注意解密时仍使用加密函数；
- 密文在传输过程中发生错误时，会得到错误明文，错误会传播几个分组。

输出反馈（Output Feedback, OFB）

加密：



解密：

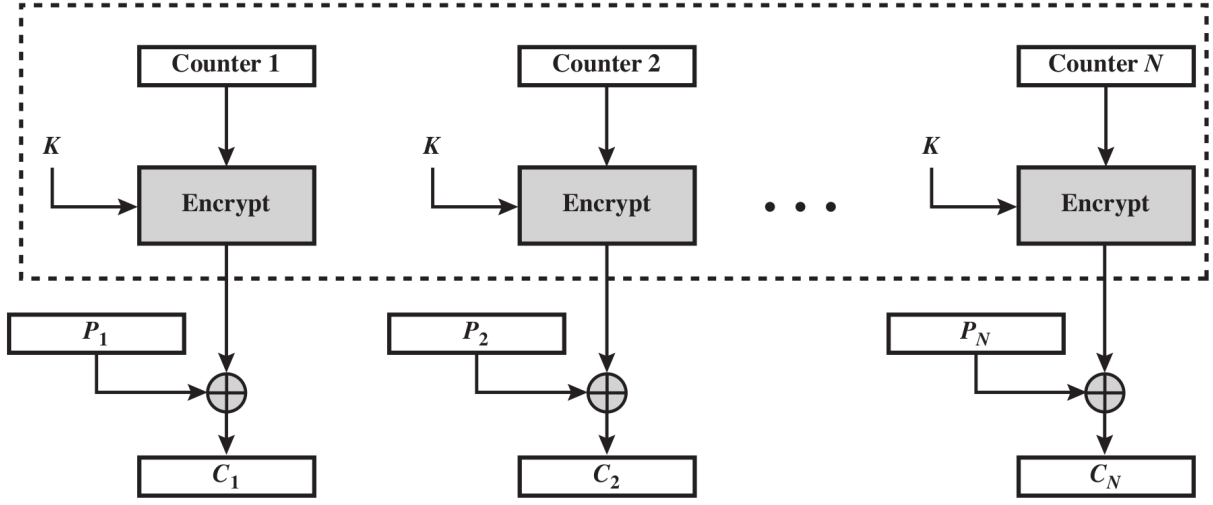


OFB 的优缺点

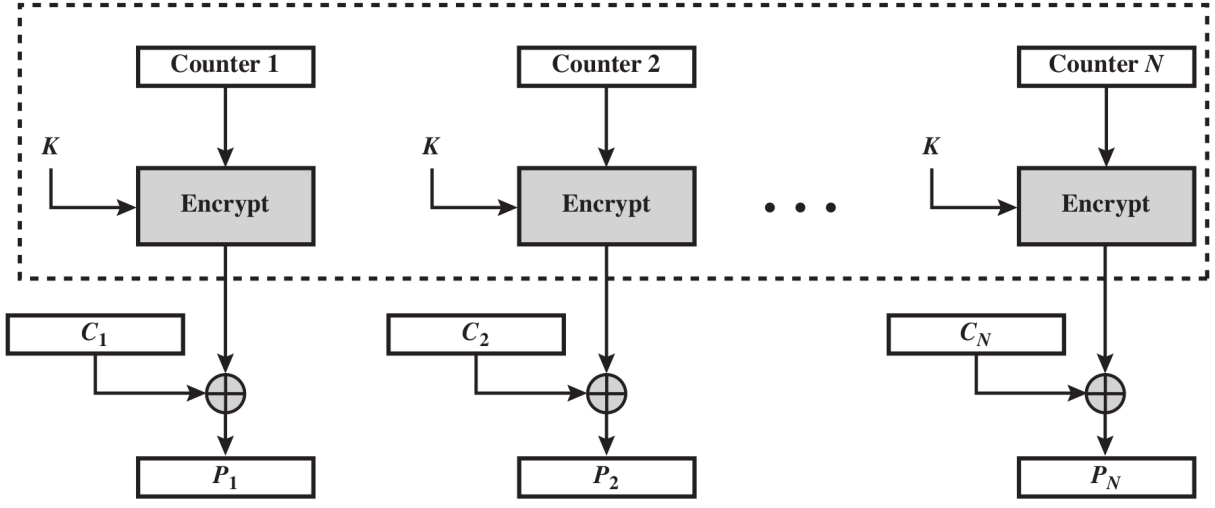
- 优点：密文传输过程中某位上发生的错误不会影响其他明文的恢复，例如，C1 中有一位发生了错误，只会影响 P1 的恢复，不会影响后续明文的恢复。
- 缺点：抗消息流篡改能力不如 CFB，即如果密文某位取反，则恢复出来的明文相应位也取反。

计数器（Counter, CTR）

加密：



解密：



CTR 的优缺点

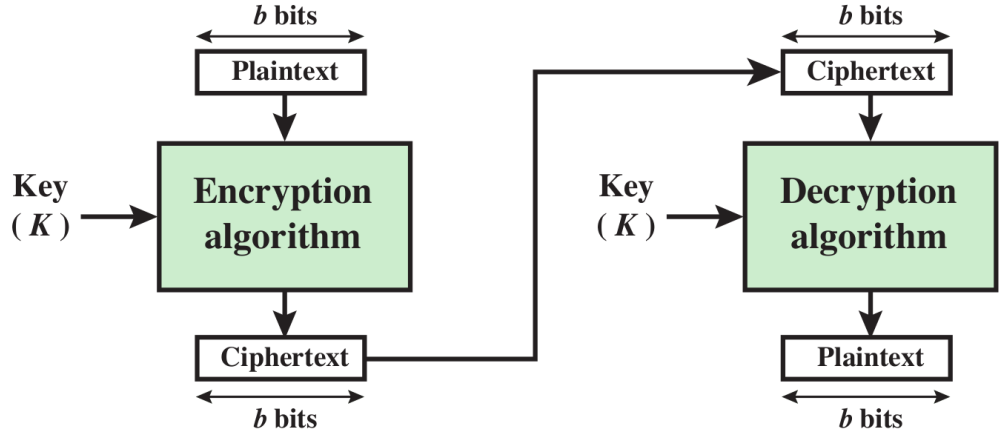
- 高效，可以做并行加密，可以用于高速网络加密中；
- 可以对被加密的分组进行随机存取；
- 相当安全；
- 简洁；
- 必须决不重复使用密钥和计数器值。

主要内容

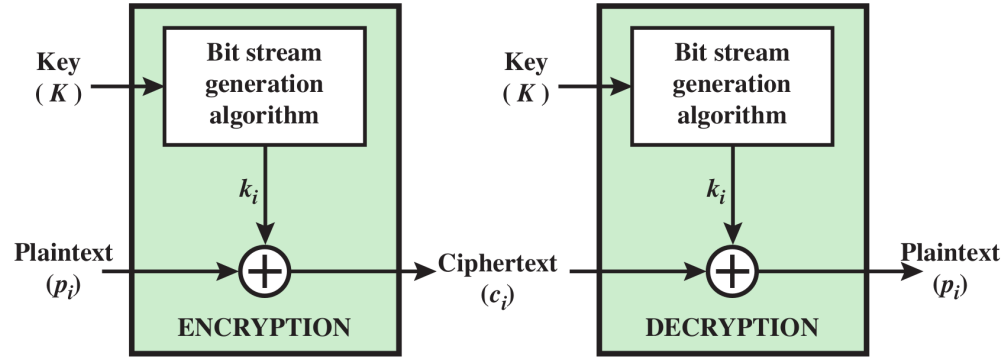
1. 分组密码的工作模式
2. 流密码

流密码与分组密码

- 分组密码将一个明文分组作为整体加密，输出等长密文分组。

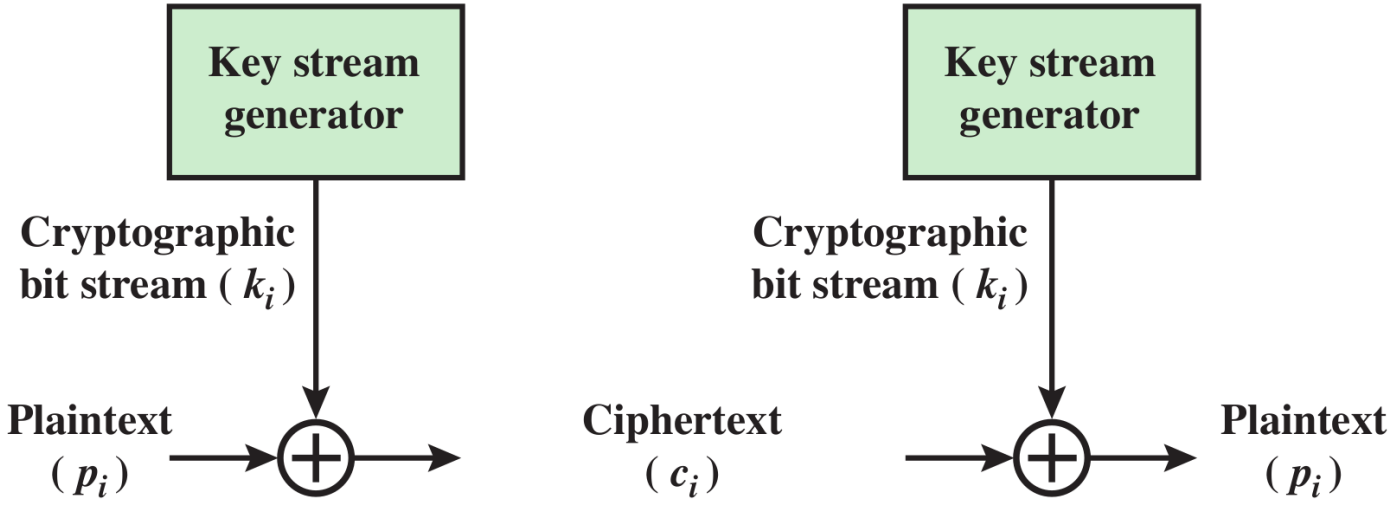


- 流密码每次加密数据流的一位或一个字节。例如古典的 Vigenère 密码与 Vernam 密码。

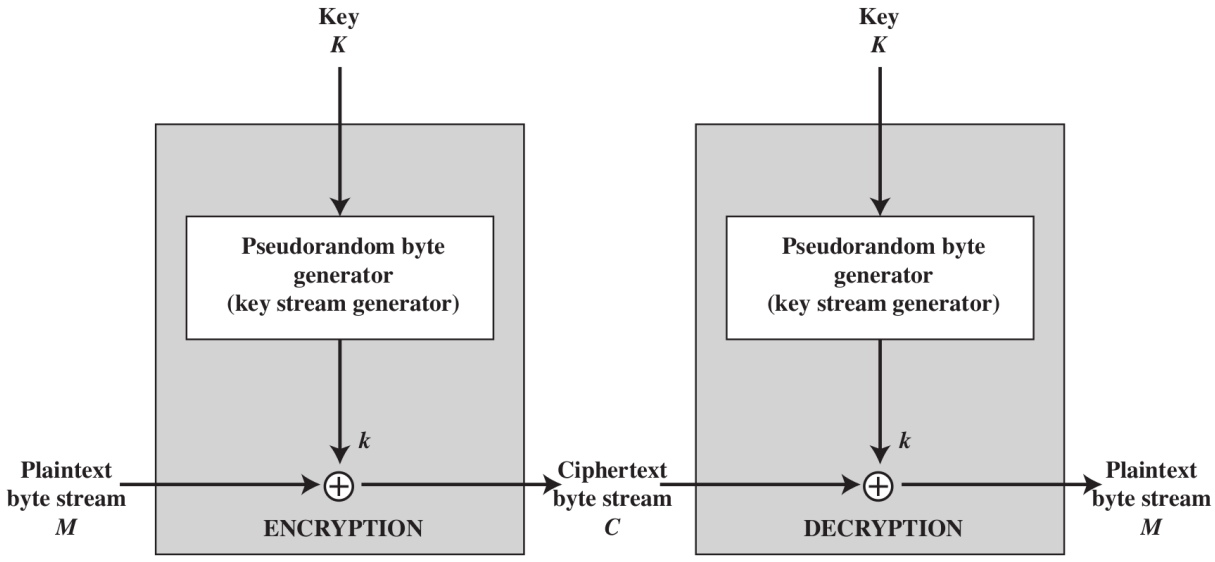


流密码与“一次一密”

- “一次一密”使用的是真正的随机数流，而流密码使用伪随机数流。



流密码设计时考虑的主要因素



- 加密序列周期要长；
- 密钥流应尽可能地接近真正的随机数流；
- 伪随机数发生器的输出受密钥 K 调节，为防止穷举攻击，密钥应该足够长。

RC4 算法

- RC4 算法是 Ron Rivest 在 RSA 公司于 1987 年设计的一种流密码[1]。
- 很长一段时间内，RC4 作为 RSA 公司的商业机密并没有公开，直到 1994 年，RC4 才在匿名邮件列表上公开。
- 密钥长度可变，面向字节操作，简单高效。
- 使用广泛，包括 WEP, WPA, BitTorrent, SSH, Remote Desktop Protocol, PDF, Skype 等。

[1] While RC4 is officially termed “Rivest Cipher 4”, the RC acronym is alternatively understood to stand for “Ron’s Code”.

RC4 算法概述

- 一个 256 字节的状态向量 $S : S[0], \ldots, S[255]$；
- 在生成密钥流的过程中，S 反复被置换，置换后的 S 始终包含 $0 \sim 255$ 所有 8 位整数；
- 密钥流的每个字节是从 S 的 255 个元素中按照一种系统化的方法选出的一个元素生成的；
- 每产生密钥流中的一个字节，S 就被重新置换一次。

初始化 S

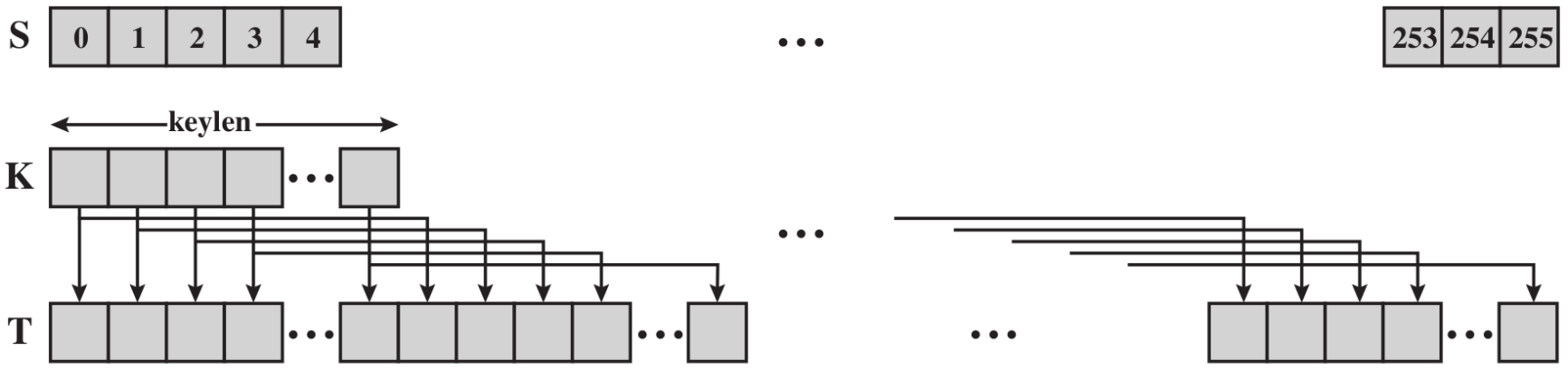
- S 中元素的值按升序被置为 0 到 255；
- 同时，建立一个 256 字节的临时向量 T：若输入密钥 K 长度为 256 字节，则将 K 赋给 T；否则，循环使用密钥赋值 T。

```
for i in range(256):
  S[i] = i
  T[i] = K[i mod keylen]
```

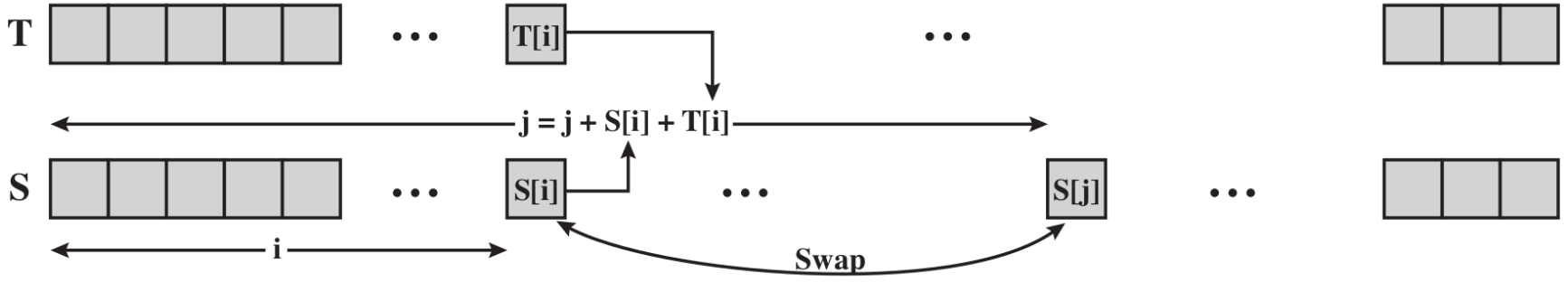
- 然后用 T 产生 S 的初始置换：对每个 $S[i]$，根据由 $T[i]$ 确定的方案，将 $S[i]$ 置换为 S 中的另一个字节。

```
j = 0
for i in range(256):
  j = (j + S[i] + T[i]) mod 256
  Swap(S[i], S[j])
```

初始化 S



(a) Initial state of S and T



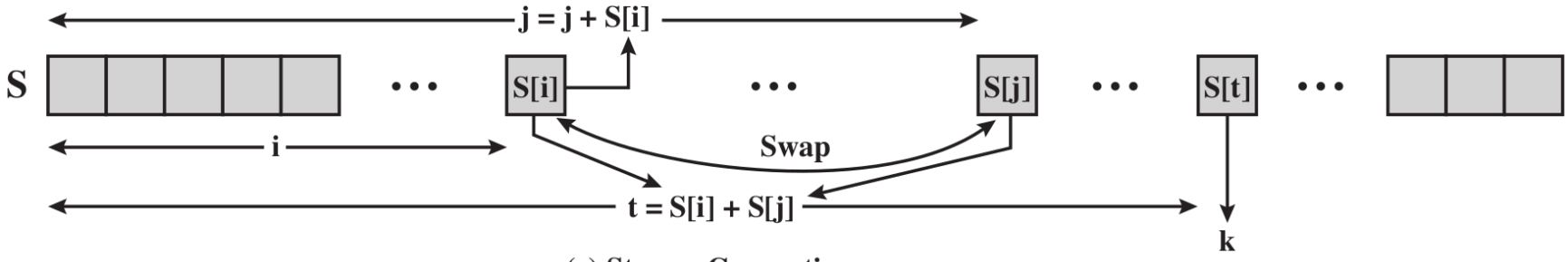
(b) Initial permutation of S

密钥流生成

- 状态向量 S 初始完后，输入密钥 K 就不再被使用。
- 从 $S[0]$ 到 $S[255]$，对每个 $S[i]$，根据 S 当前配置，将 $S[i]$ 与 S 中的另一个字节置换。
- 当 $S[255]$ 完成置换后，操作重复从 $S[0]$ 开始。

```
i, j = 0
while True:
  i = (i + 1) mod 256
  j = (j + S[i]) mod 256
  Swap(S[i], S[j])
  t = (S[i] + S[j]) mod 256
  k = S[t]
```

密钥流生成



(c) Stream Generation

RC4 的安全性

- RC4 的加密结果是非常“非线性”的；
- 当密钥足够长时，可以抵御很多攻击；
- 用于为 802.11 无线局域网提供安全性的 WEP 协议，易受一种特殊攻击，但问题不在 RC4 本身，而是 RC4 的输入密钥的产生方法有漏洞；
- The use of RC4 in TLS is prohibited by RFC 7465 published in February 2015.
- Mozilla and Microsoft have issued similar recommendations.

RC4 的安全性

[Docs] [txt|pdf] [draft-ietf-tls-...] [Tracker] [Diff1] [Diff2]

PROPOSED STANDARD

Internet Engineering Task Force (IETF) — A. Popov
Request for Comments: 7465 — Microsoft Corp.
Updates: 5246, 4346, 2246 — February 2015
Category: Standards Track
ISSN: 2070-1721

Prohibiting RC4 Cipher Suites

Abstract

This document requires that Transport Layer Security (TLS) clients and servers never negotiate the use of RC4 cipher suites when they establish connections. This applies to all TLS versions. This document updates RFCs 5246, 4346, and 2246.

Version	Editor	Changes
5.0.1	April King	Add note about IE 11 on Windows Server 2008 R2
5.0	April King	Server Side TLS 5.0
4.2	April King	Updated cipher suite table
4.1	Julien Vehent	Clarify Logjam notes, Clarify risk of TLS Tickets
4	Julien Vehent	Recommend ECDSA in modern level, remove DSS ciphers, publish configurations as JSON
3.8	Julien Vehent	redo cipher names chart (April King), move version chart (April King), update Intermediate cipher suite (ulfr)
3.7	Julien Vehent	cleanup version table (April King), add F5 conf samples (warburtron), add notes about DHE (rgacogne)
3.6	Julien Vehent	bump intermediate DHE to 2048, add note about java compatibility
3.5	alm	comment on weakdh vulnerability
3.4	Julien Vehent	added note about session resumption, HSTS, and HPKP
3.3	Julien Vehent	fix SHA256 prio, add POODLE details, update various templates
3.2	Julien Vehent	Added intermediate compatibility mode, renamed other modes
3.1	Julien Vehent	Added non-backward compatible ciphersuite
3	Julien Vehent	Remove RC4 for 3DES, fix ordering in openssl 0.9.8 (1024430), various minor updates
2.5.1	Julien Vehent	Revisit ELB capabilities
2.5	Julien Vehent	Update ZLB information for OCSP Stapling and ciphersuite
2.4	Julien Vehent	Moved a couple of aes128 above aes256 in the ciphersuite
2.3	Julien Vehent	Precisions on IE 7/8 AES support (thanks to Dobin Rutishauser)
2.2	Julien Vehent	Added IANA/OpenSSL/GnuTLS correspondence table and conversion tool
2.1	Julien Vehent	RC4 vs 3DES discussion. r=joes r=tinfoil
2.0	Julien Vehent, kang	Public release.
1.5	Julien Vehent, kang	added details for PFS DHE handshake, added nginx configuration details; added Apache recommended conf

Security Advisory 2868725: Recommendation to disable RC4

Security Research & Defense / By swiat / November 12, 2013 / Schannel

In light of recent research into practical attacks on biases in the RC4 stream cipher, Microsoft is recommending that customers enable TLS1.2 in their services and take steps to retire and deprecate RC4 as used in their TLS implementations. Microsoft recommends TLS1.2 with AES-GCM as a more secure alternative which will provide similar performance.

小结

1. 分组密码的工作模式
2. 流密码