SHA

- 安全散列算法（Secure Hash Algorithm，缩写为SHA）是一个密码散列函数家族，是FIPS所认证的安全散列算法。
- SHA由美国国家安全局（NSA）所设计，并由美国国家标准与技术研究院（NIST）发布；是美国的政府标准。
- 2005年关于SHA-1安全性的最新研究结果引发了人们对其在未来应用中的担忧。

SHA

Table 11.3 Comparison of SHA Parameters

Algorithm	Message Size	Block Size	Word Size	Message Digest Size
SHA-1	$< 2^{64}$	512	32	160
SHA-224	$< 2^{64}$	512	32	224
SHA-256	$< 2^{64}$	512	32	256
SHA-384	$< 2^{128}$	1024	64	384
SHA-512	$< 2^{128}$	1024	64	512
SHA-512/224	$< 2^{128}$	1024	64	224
SHA-512/256	$< 2^{128}$	1024	64	256

Note: All sizes are measured in bits.

SHA家族的参数比较

SHA-1

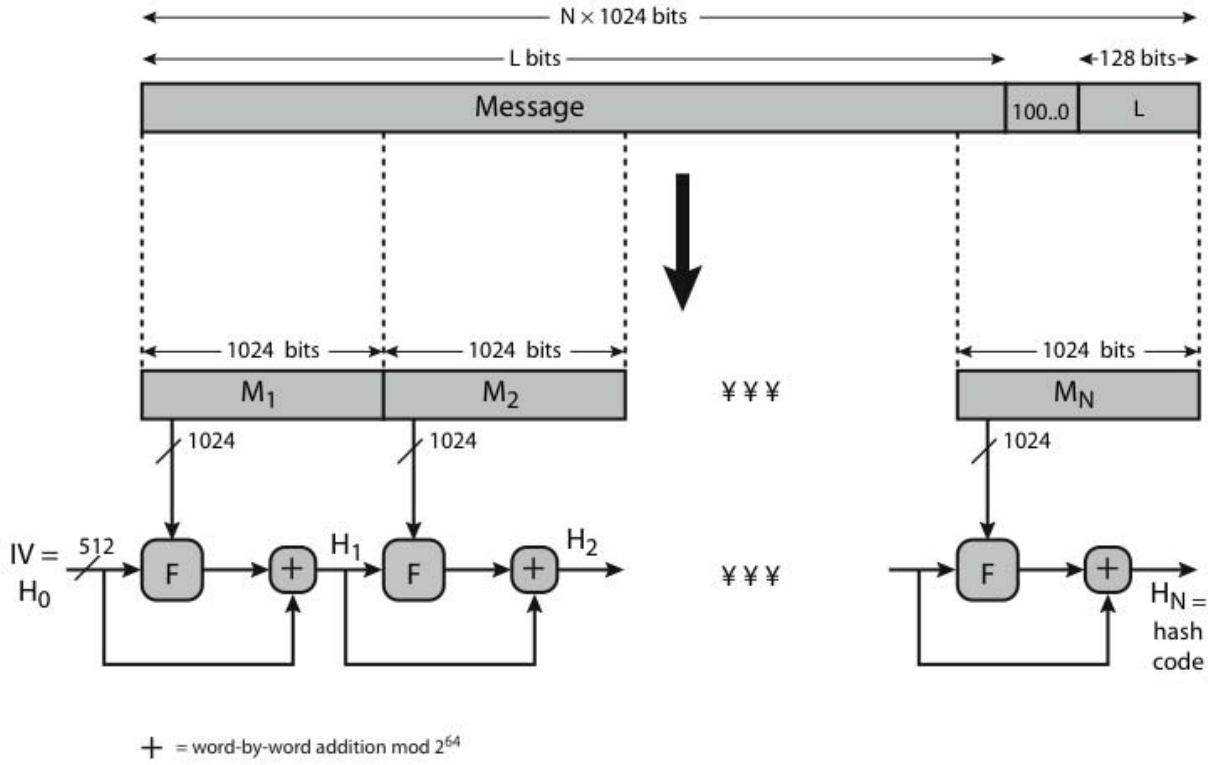
- sha-1散列算法计算出来的hash值达160位，即20字节，比md5多了32位。
- sha-1也是分块计算，每块也是64字节，当最后一块不足64字节也按照md5的方式进行填充。
- 数据块的最后一定要补上表示报文总共位数的8个字节（大端存储位数）。

修订的安全哈希标准

- NIST于2002年发布了修订版FIPS 180-2：
- 添加3个附加版本的SHA
  - SHA-256、SHA-384、SHA-512
- 旨在与AES密码提供的更高安全性兼容
- 结构和细节与SHA-1类似，但安全级别相当高。

SHA-512

- 第 1 步：附加填充位
- 第 2 步：追加长度
- 第 3 步：初始化哈希缓冲区
- 第 4 步：以 1024 位（128 字节）为块处理消息（算法的核心）
- 第 5 步：输出最终状态值作为结果hash值



SHA-3

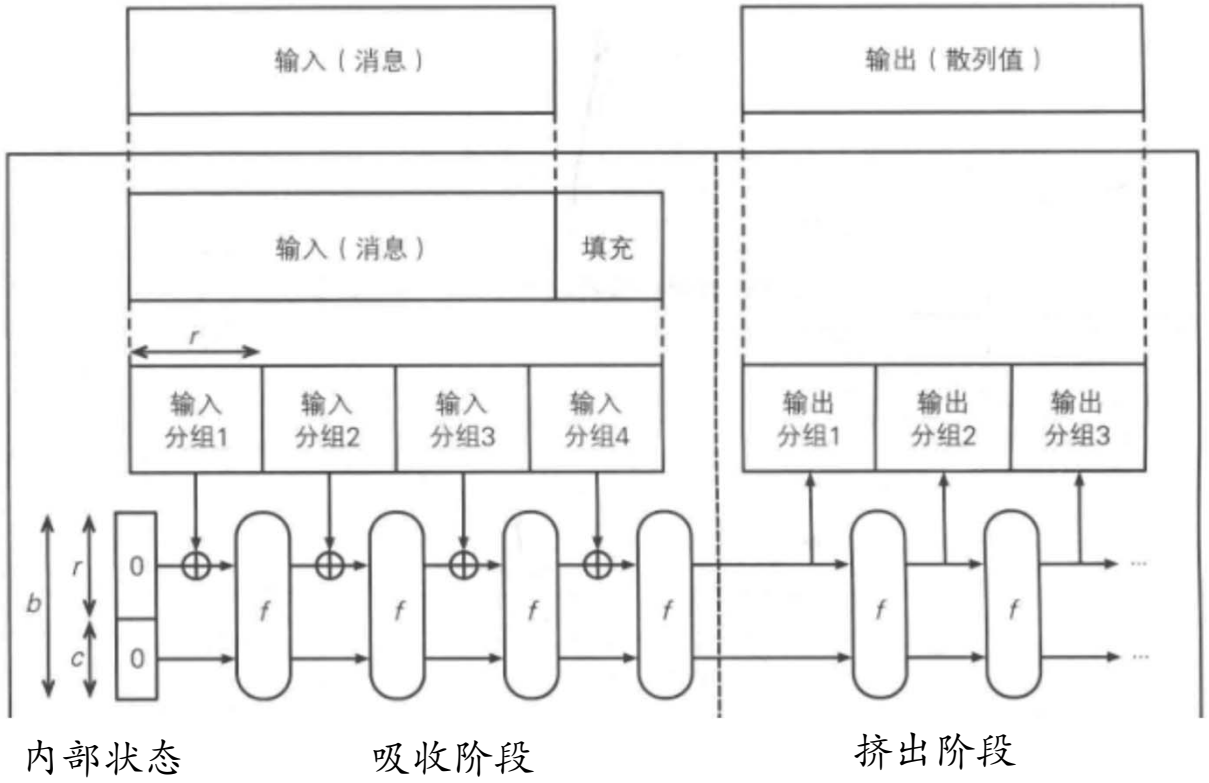
- SHA-1 尚未完全“破解”
  - 但类似于已破译的MD5 和 SHA-0 ，所以被认为是不安全的
- SHA-2（尤其是 SHA-512）目前似乎是安全的
  - 但与前身具有相同的结构和数学运算，令人担忧
- SHA-3是与之前算法不同的，可替换的加密散列算法
  - 2012年10月2日，Keccak 被选为NIST散列函数竞赛的胜利者
  - SHA-3 在2015年8月5日由 NIST 通过 FIPS 202 正式发表

Keccak

- Keccak 是一个加密散列算法，由 Guido Bertoni，Joan Daemen，Michaël Peeters，以及Gilles Van Assche在RadioGatún上设计。
- 2012年10月2日，Keccak 被选为NIST散列函数竞赛的胜利者。
- SHA-2目前没有出现明显的弱点。但由于对MD5、SHA-0和SHA-1出现成功的破解，NIST感觉需要一个与之前算法不同的，可替换的加密散列算法，也就是现在的 SHA-3。

Keccak

- Keccak采用了与SHA-1、SHA-2完全不同的海绵结构，Keccak的海绵结构中，输入的数据在进行填充之后，要经过吸收阶段和挤出阶段，最终生成输出的散列值。先将输入的消息吸收到内部状态，然后再根据内部状态挤出相应的散列值。



Google宣布攻破加密哈希算法SHA-1，从此SHA-1不再安全!

- 谷歌宣布，完成了真实世界的第一次碰撞攻击，创造了两个SHA-1完全相同的PDF文件。
- 此举意味着，SHA-1从2005年开始就被质疑到如今正式要摇摇欲坠。
- 2013年，Marc Stevens曾发表一篇论文，专门介绍了创建SHA-1碰撞的理论性方法。谷歌首先创建了一份专门制作的PDF前缀，用以生成两份拥有任意不同内容的文档，但二者同时具备相同的SHA-1摘要。此后，谷歌利用谷歌的技术专长与云基础设施计算碰撞情况，这也是谷歌截至目前已完成的规模最大的计算任务之一。
- 本次碰撞攻击的计算量相当于单CPU连续处理6500年再加上GPU连续处理110年。

SHATTERED

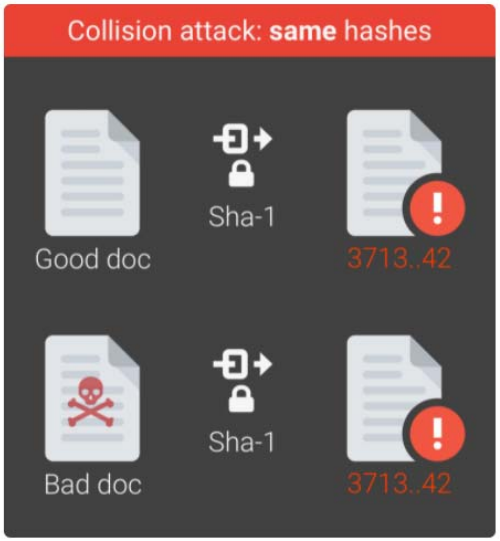
We have broken SHA-1 in practice.

This industry cryptographic hash function standard is used for digital signatures and file integrity verification, and protects a wide spectrum of digital assets, including credit card transactions, electronic documents, open-source software repositories and software updates.

It is now practically possible to craft two colliding PDF files and obtain a SHA-1 digital signature on the first PDF file which can also be abused as a valid signature on the second PDF file.

For example, by crafting the two colliding PDF files as two rental agreements with different rent, it is possible to trick someone to create a valid signature for a high-rent contract by having him or her sign a low-rent contract.

Infographic | Paper



Attack complexity

9,223,372,036,854,775,808
SHA-1 compressions performed

Shattered compared to other collision attacks

MD5
1 smartphone
30 sec

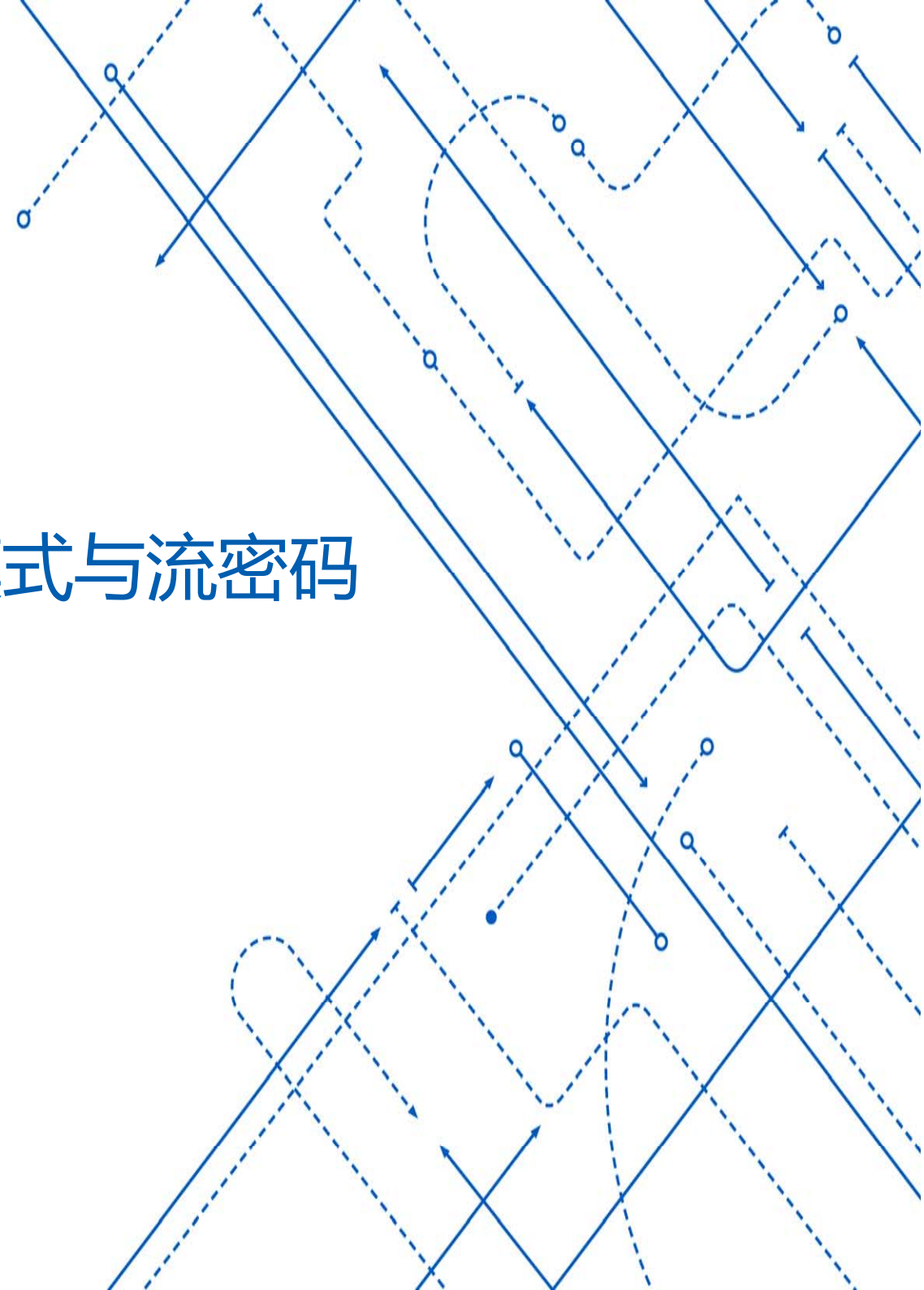
SHA-1 Shattered
110 GPU
1 year

SHA-1 Bruteforce
12,000,000 GPU
1 year

标志性成果

- 王小云教授带领的研究小组于2004年、2005年先后破解了被广泛应用于计算机安全系统的MD5和SHA-1两大密码算法。
- 对于这项十几年来国际上首次成功破解全球广泛使用的密码算法与标准的工作，整个国际密码学界为之震惊。
- 密码学领域最权威的两大刊物Eurocrypto与Crypto将2005年度最佳论文奖授予了这位中国女性，其研究成果引起了国际同行的广泛关注。
- 2006年6月8日，中国科学院第13次院士大会和中国工程院第8次院士大会上以“国际通用Hash函数的破解”获颁陈嘉庚科学奖信息技术科学奖。

第4章 分组密码工作模式与流密码



浙江大学
ZHEJIANG UNIVERSITY

目录

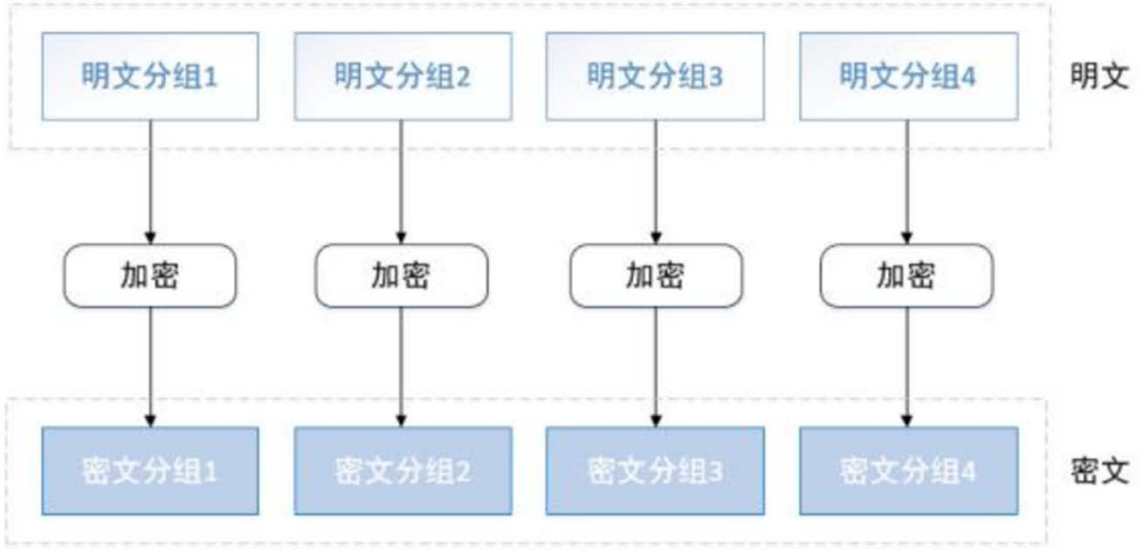
CONTENTS

分组密码

- 分组密码（Block cipher），又称分块加密或块密码，是一种对称密钥算法。
- 它将明文分成多个等长的模块（block），使用确定的算法和对称密钥对每组分别加密解密。
- 分组加密是极其重要的加密协议组成，其中典型的如AES和3DES作为美国政府核定的标准加密算法，应用领域从电子邮件加密到银行交易转帐，非常广泛。
- 现代分组加密创建在迭代的思想上产生密文。作为一种通过简单操作如替代和排列等以有效改善保密性的方法。迭代产生的密文在每一轮加密中使用不同的子密钥，而子密钥生成自原始密钥。

分组密码的基本原理

- 设n是一个分组密码的分组长度，k 是密钥，$x = x_0x_1x_2\cdots x_{n-2}x_{n-1}$ 为明文，其中 $x_i \in \mathrm{GF}(2), 0 \leqslant i \leqslant n-1$，$y = y_0y_1y_2\cdots y_{m-2}y_{m-1}$ 为相应的密文，其中 $y_j \in \mathrm{GF}(2), 0 \leqslant j \leqslant m-1$ 则 $y = E_k(x)$，$x = D_k(y)$ 分别表示在密钥k控制下的加密和解密变换。
- 如果n < m，则分组密码对明文加密后有数据扩展；如果n > m，则分组密码对明文加密后有数据压缩；如果n = m，则分组密码对明文加密后既无数据扩展也无压缩。



分组密码的基本原理-置换

- 定义：设S 是一个有限集合，φ 是从S 到S 的一个映射。如果对任意u；v ∈ S，当u ≠ v 时，$\varphi(u) \neq \varphi(v)$，则称$\varphi$ 为S 上的一个置换(permutation)。

- 对于一个分组长度为n 的分组密码，不同的密钥应该对应不同的加密和解密变换. 给定密钥k，对于任意的 $u, v \in \mathrm{GF}(2)^n$，如果u ≠ v，则一定有 $E_k(u) \neq E_k(v)$
- 这是因为如果 $E_k(u) = E_k(v)$ 则在解密时将难以准确地恢复明文. 因此，对于给定的密钥k，加密变换E_k是 $\mathrm{GF}(2)^n$ 的一个置换，解密变换D_k是 E_k 的逆置换。

分组密码的基本原理-扩散与混淆

- 扩散(diffusion) 和混淆(confusion) 是C. E. Shannon 提出的设计密码体制的两种基本方法, 其目的是为了抵抗对手对密码体制的统计分析。
- 扩散：让明文中的每一位影响密文中的许多位，或者说让密文中的每一位受明文中的许多位的影响，这样可以隐蔽明文的统计特性。
- 混淆：将密文与密钥之间的统计关系变得尽可能复杂，使得对手即使获取了关于密文的一些统计特性，也无法推测密钥。

分组密码工作模式（mode of operation）

- 密码学中，分组密码的工作模式（mode of operation）允许使用同一个分组密码密钥对多于一块的数据进行加密，并保证其安全性。
- 分组密码自身只能加密长度等于密码分组长度的单块数据，若要加密变长数据，则数据必须先被划分为一些单独的密码块。通常而言，最后一块数据也需要使用合适填充方式将数据扩展到符合密码块大小的长度。
- 一种工作模式描述了加密每一数据块的过程，并常常使用基于一个通常称为初始化向量的附加输入值以进行随机化，以保证安全。

分组密码工作模式（mode of operation）

- 工作模式主要用来进行加密和认证。
- 对加密模式的研究曾经包含数据的完整性保护，即在某些数据被修改后的情况下密码的误差传播特性。后来的研究则将完整性保护作为另一个完全不同的，与加密无关的密码学目标。部分现代的工作模式用有效的方法将加密和认证结合起来，称为认证加密模式。
- 虽然工作模式通常应用于对称加密，它亦可以应用于公钥加密，例如在原理上对RSA进行处理。

分组密码工作模式（mode of operation）

- 分组密码加密固定大小的块
  - 例如，DES 使用 56 位密钥加密 64 位数据块
- 在实践中需要某种方法来加密/解密任意数量的数据
- 常用模式：
  - 1.电子密码簿模式（Electronic Codebook Book (ECB)）
  - 2.密码块链接模式（Cipher Block Chaining (CBC)）
  - 3.密文反馈模式（Cipher FeedBack (CFB)）
  - 4.输出反馈模式（Output FeedBack (OFB)）
  - 5.计数器模式（Counter (CTR)）
  - 6. XTS

电子密码簿模式（ECB）

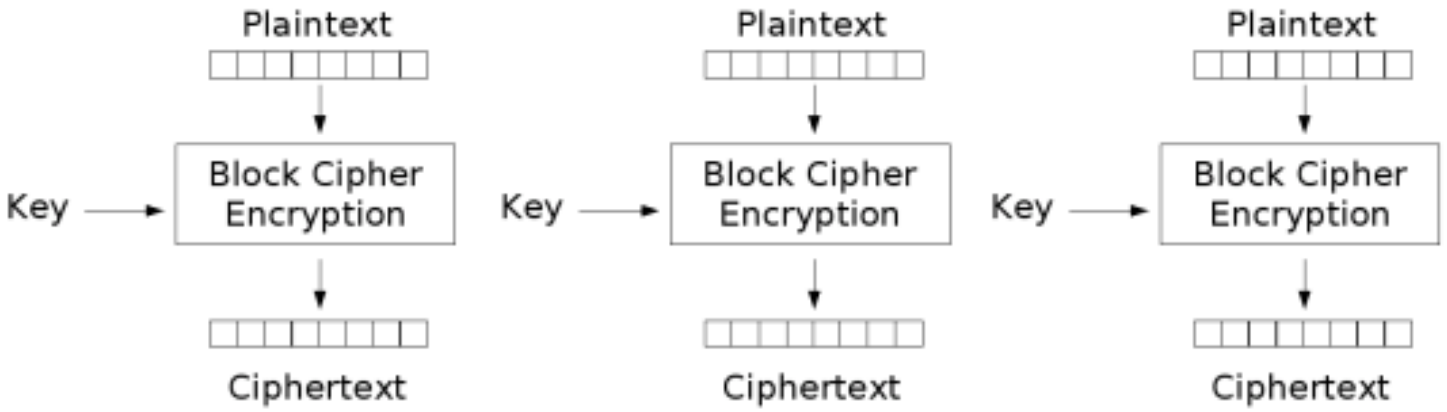
- 需要加密的消息按照块密码的块大小被分为数个块，并对每个块进行独立加密。

- 典型应用：
  - 用于随机数的加密保护。
  - 用于单分组明文的加密。

- 为了解决统一明文产生相同密文的问题，提出了其它的加密模式。

电子密码簿模式（ECB）- 加密

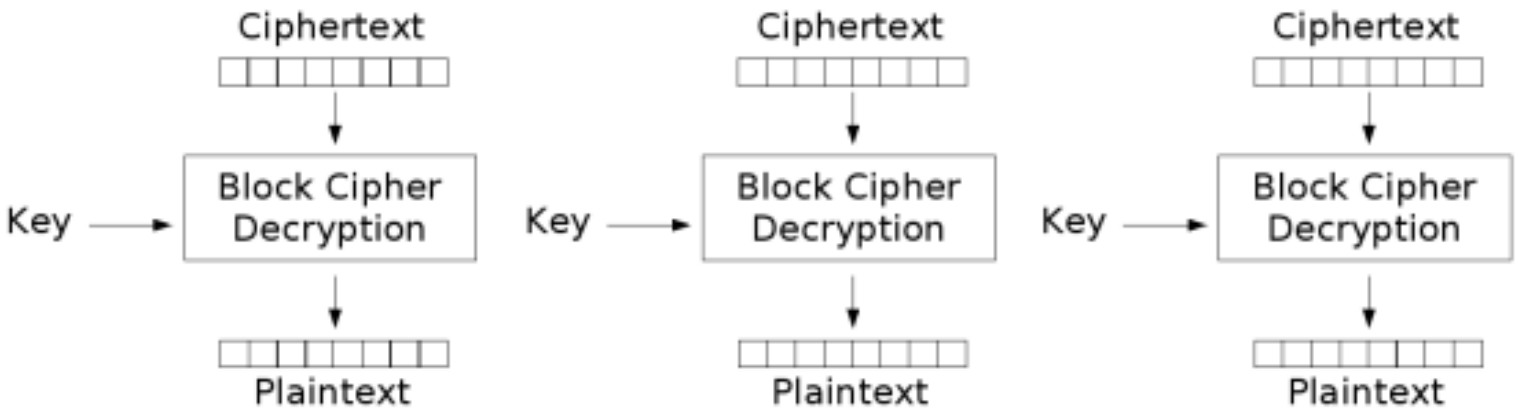
- 加密过程：$C_j = E_k(P_j)$



Electronic Codebook (ECB) mode encryption

电子密码簿模式（ECB）- 解密

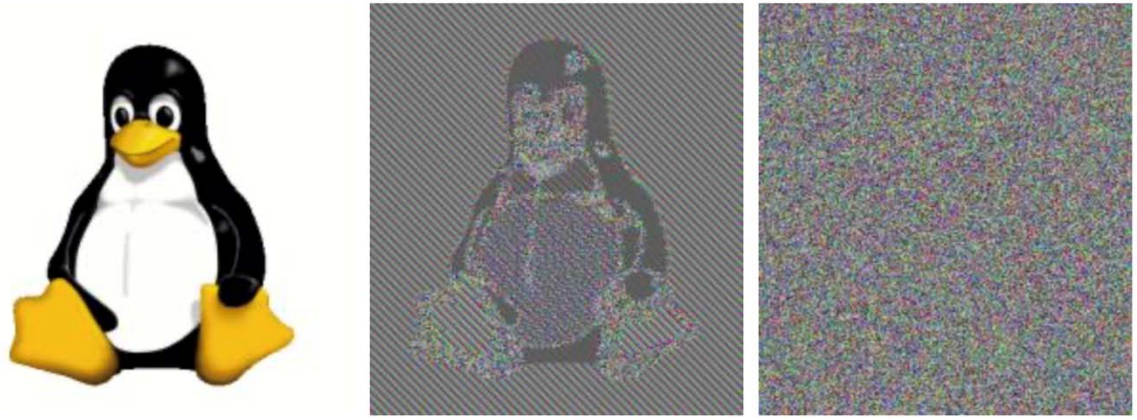
- 解密过程：$P_j = D_k(C_j)$



Electronic Codebook (ECB) mode decryption

电子密码簿模式（ECB）

- 优点：
  - 实现简单。不同明文分组的加密/解密可以并行计算，速度很快。
- 缺点：
  - 同样的明文块会被加密成相同的密文块，不会隐藏明文分组的统计规律。



原图　　使用ECB模式加密　　提供了伪随机性的非ECB模式

密文块链接模式（CBC）

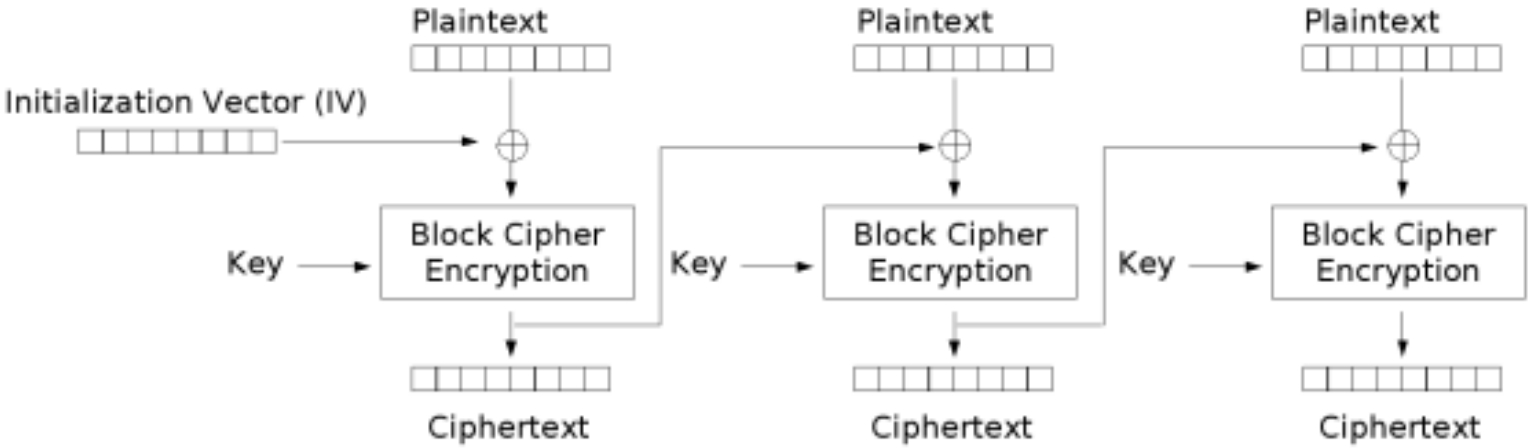
- 1976年，IBM发明了密文块链接（CBC，Cipher-block chaining）模式。为了克服 ECB 中的重复和顺序独立性问题，需要某种方式使密文依赖于它之前的所有块。
- 在CBC模式中，每个明文块先与前一个密文块进行异或后，再进行加密。
- 在这种方法中，每个密文块都依赖于它前面的所有明文块。
- 为了保证每条消息的唯一性，在第一个块中需要使用初始化向量（IV）。
- 用途：
  - 批量数据加密
  - 常见的数据加密和 TLS 加密
  - 完整性认证和身份认证

密文块链接模式（CBC）

- 初始化向量（IV）：
  - 发送方和接收方必须知道的信息
- 攻击者可能可以通过改变初始化向量来伪造第一块数据，因此：
  - IV必须是一个固定值
  - 或者以难以操纵、不可预测的方式衍生出来
  - 或者在发送消息之前以ECB模式加密发送
  - 否则必须检查消息完整性

密文块链接模式（CBC）- 加密

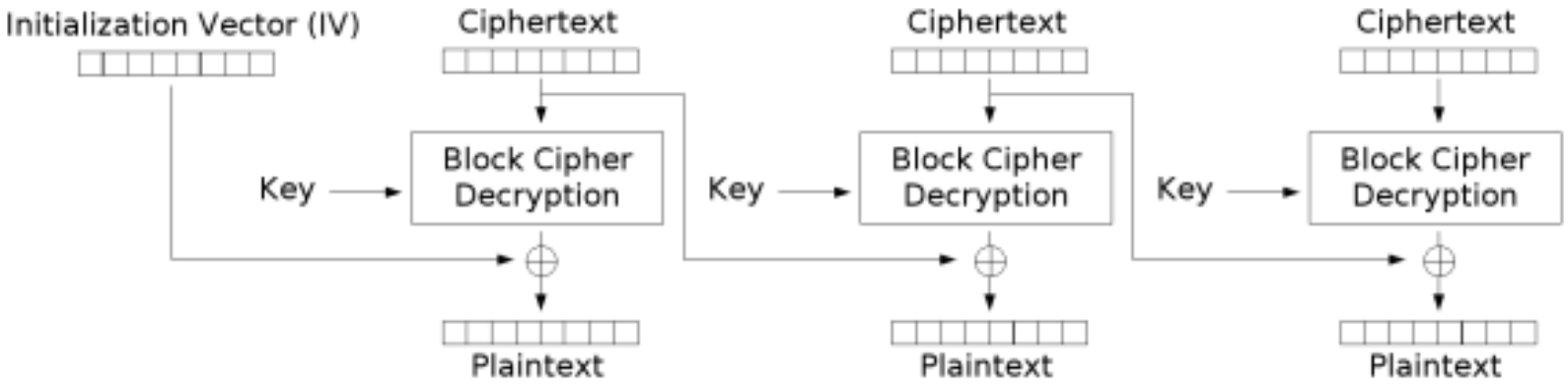
- 加密过程： $C_j = E_k(P_j \oplus C_{j-1})$



Cipher Block Chaining (CBC) mode encryption

密文块链接模式（CBC）- 解密

- 解密过程： $P_j = D_k(C_j) \oplus C_{j-1}$



Cipher Block Chaining (CBC) mode decryption

密文块链接模式（CBC）

- 优点：
  - 密文块不仅和当前明文块相关，而且和前一个密文块或 IV 相关，隐藏明文的统计特性。
  - 具有有限的两步错误传播特性，密文块中的一位变化只会影响当前密文块和下一密文块。
  - 在加密时，明文中的微小改变会导致其后的全部密文块发生改变（雪崩效应）。
  - 而在解密时，从两个邻接的密文块中即可得到一个明文块，解密过程可以被并行化。
  - 而解密时，密文中一位的改变只会导致其对应的明文块完全改变和下一个明文块中对应位发生改变，不会影响到其它明文的内容。具有自同步特性，即第 k 块起密文正确，则第 k+1 块就能正常解密。
- 缺点：
  - 加密不能并行，只能串行。解密可以并行。
  - 可能可以伪造明文的第一块。

密文块链接模式（CBC）- 密文窃取（CTS）

- 密文窃取（CTS）是一种分组密码操作模式使用的通用方法，该操作模式允许处理不能均匀分割成块的消息，而不会导致密文的扩展，代价是稍微增加了复杂性。
- ECB 模式下的密文窃取在前两个块内引入了一种块间依赖关系，导致后两个块的错误传播行为发生改变。