第 5 章　认证加密算法

5.1　概述

5.1.1　认证加密算法一般性设计原理

认证加密算法是能够同时保护数据机密性、完整性以及进行数据源认证的对称密码算法。早期的认证加密算法多采用组合方式设计，主要的组合方式有先加密后认证（Encrypt-then-MAC，EtM）、先认证后加密（MAC-then-Encrypt，MtE）和加密并认证（Encrypt-and-MAC，E&M）。这 3 种构造方式的优势在于通用性，适用于常见的加密算法和认证算法，但其没有令人信服的安全性[383,384]。Bellare 等人分析了 3 种组合认证加密方案（EtM、MtE、E&M）的安全性，并正式提出了认证加密（Authenticated Encryption，AE）的概念[385]。带有关联数据的认证加密（Authenticated Encryption with Associated Data，AEAD）[386]是认证加密的一种扩展，在为明文数据提供机密性和完整性双重保护的同时，为关联数据提供完整性保护。AEAD 的加密算法以密钥 K、临时值 N、关联数据 A 和明文 M 为输入，输出密文 C 和标签 T；解密算法以密钥 K、临时值 N、关联数据 A、密文 C 和标签 T 为输入，输出明文 M 或者符号 $\perp$（意味着完整性验证没有通过）。AEAD 的典型应用是各种网络协议，协议报头等信息仅需要完整性保护而不需要机密性保护。

目前的认证加密算法基本上都是 AEAD，可以归为两类，一类是分组密码工作模式，另一类是直接设计的认证加密算法。分组密码工作模式类的认证加密算法又称分组密码认证加密工作模式，此类认证加密算法以黑盒方式调用分组密码，优点是可以方便地替换底层分组密码，比较典型的算法有 OCB、GCM、CCM、EAX 等[387-389]。这方面的研究始于十几年前 NIST 对 AES 征集认证加密工作模式。国际标准化组织和 NIST 都颁布了相关标准，收录了 OCB 2.0、Key Wrap、CCM、EAX、Encrypt-then-MAC、GCM 共 6 种认证加密模式[390]，这 6 种模式包含了不同的设计理念和对应用需求的考量，是目前具有代表性和影响力的认证加密算法。

分组密码认证加密工作模式一般分为 One-pass 和 Two-pass 两类。Two-pass 的算法需要对明文进行两次处理，比如 CCM；One-pass 的算法只需要对每个明文分组进行一次处理，比如 OCB。OCB 是最经典的分组密码认证加密工作模式，其设计理念影响了许多认证加密算法。OCB 经历了 OCB1、OCB2 和 OCB3 多个版本，在掩码生成方式等方面不断优化，在安全性证明方面不断改进。如果底层可调分组密码是理想的，则 OCB3 是具有完美保密性的依赖临时值的认证加密算法[391]。

可调分组密码是一个函数 $\widetilde{E}:\{0,1\}^k\times\{0,1\}^w\times\{0,1\}^n\to\{0,1\}^n$，对于任意的密钥 $K\in\{0,1\}^k$ 和调柄 $W\in\{0,1\}^w$，$\widetilde{E}(K,W,\cdot)$是$\{0,1\}^n$ 上的一个置换，通常记为 $\widetilde{E}_K(W,\cdot)$ 或者 $\widetilde{E}_K^W$。可调分组密码是一族分组密码，利用公开参数调柄实现分组密码可变性。和一

般的分组密码相比，可调分组密码多了一个调柄输入，调柄是不需要保密的公开值，意味着即使对手控制了调柄，可调分组密码仍然保持安全性。类似伪随机置换和强伪随机置换，文献[392]定义了伪随机可调置换和强伪随机可调置换以及安全可调分组密码和强安全可调分组密码。

一些分组密码的密钥扩展算法比较复杂，更改密钥需要执行密钥扩展算法。调柄的作用主要是为分组密码提供可变性，因此，可调分组密码应该具有更改调柄比更新密钥代价小的属性。基于分组密码 E，有如下两种构造可调分组密码的通用方法，分别称作 XE、XEX 框架：

(1) $\widetilde{E}_K(W,M)=E_K(W\oplus E_K(M))$。

(2) $\widetilde{E}_{K,h}(W,M)=E_K(M\oplus h(W))\oplus h(W)$。

假设 E 是安全分组密码，则第一种方法构造的可调分组密码是安全的；进一步假定 h 是 ε-AXU_2 泛杂凑函数(即，对于所有满足 $x\neq y$ 的 x、y、z 都有 $\Pr_h(h(x)\oplus h(y)=z)\leqslant\varepsilon$ 成立)，则第二种方法可以构造强安全的可调分组密码。第一种构造方法更改调柄很容易，但是需要调用两次分组密码；第二种构造方法只需调用一次分组密码，比第一种构造方法更有效。

可调分组密码的引入使得设计认证加密算法更容易，证明更简洁，可调分组密码已经成为认证加密算法的一种底层模块。TAE 是一个基于可调分组密码的认证加密工作模式，OCB 可以看成 TAE 的一个实例，其中的可调分组密码可以看成是在 XEX 框架下利用分组密码构造的。利用 XEX 框架构造的可调分组密码不是理想的，仅提供生日攻击安全性，因此 OCB 仅提供生日界安全。当临时值 N 重用时，OCB 不安全，因此 OCB 是依赖临时值的认证加密算法。与 OCB 不同，SIV 是抵抗临时值重用的认证加密算法[393]。SIV 基于一个加密方案 E 和一个伪随机函数 F，首先计算标签 $T=F(K_1,N\parallel M)$，然后计算密文 $C=E(K_2,T,M)$，标签 T 作为加密的临时值使用。SIV 是 Two-pass 的认证加密算法，而且需要两个密钥。RFC 5297 给出 SIV 的一个实例，其中，加密方案 E 采用 AES-CTR，伪随机函数 F 采用 CMAC-AES。

直接设计的认证加密算法是从底层模块直接设计的，而且分组密码不再是底层模块的唯一选择，基于伪随机置换、伪随机函数、压缩函数等构建的认证加密算法被陆续提出，其中基于置换的 AEAD 结构多来源于海绵结构。海绵结构采用固定置换作为底层模块，以先吸收再榨取的方式处理消息，是目前杂凑函数设计中备受青睐的一种结构。Bertoni 等人将海绵结构中的吸收与榨取阶段结合，形成边吸收边榨取的 Duplex 结构[394]。如图 5-1 所示，

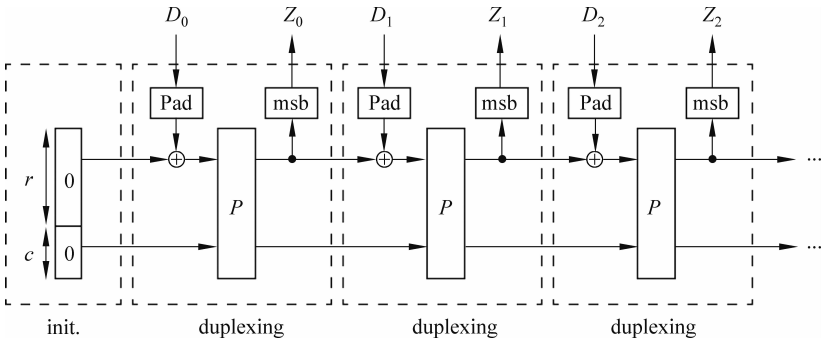

图 5-1　Duplex 结构

Duplex 结构由初始化(init.)和迭代(duplexing)组成。其中，P 是固定长度的置换；Pad 是消息的填充方式；r 和 c 是两个参数，分别称为比率和容量；输出 Z 可用作加密的密钥流或认证的标签。

Duplex 结构可以用于处理各种长度的密钥、关联数据、消息等，具有极强的灵活性。底层固定置换无须进行逆向调用，具有较好的实现性能。基于 Duplex 结构，Bertoni 等人提出了认证加密算法的 MonkeyDuplex 结构[395]，如图 5-2 所示。MonkeyDuplex 结构相对于 Duplex 结构的差异在于：密钥吸入的状态由外部状态扩展到整个状态；增强初始化过程，对吸入密钥的状态进行一次置换；降低 duplexing 模块中置换的迭代轮数。MonkeyDuplex 结构可以使用不同的底层模块。如果采用迭代型置换，初始化和末尾置换的迭代轮数通常大于 duplexing 模块的迭代轮数。如果密钥长度为 k 比特，置换状态为 b 比特，容量为 c 比特，标签为 t 比特，则 MonkeyDuplex 结构的安全界量级如下：机密性为 $\min\{b/2, c, k\}$，完整性为 $\min\{c/2, k, t\}$[396]。

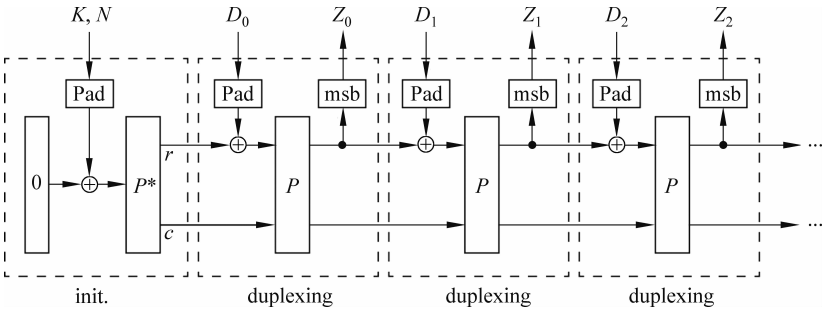

图 5-2　MonkeyDuplex 结构

Chakraborti 等人在 Duplex 结构上增加了一个简单的变换 ρ，得到一种变体——Beetle 结构。变换 ρ 作用在数据吸收和榨取阶段，使得释放的密文和反馈到状态的值不同。Duplex 类结构的密钥嵌入方式有多种形式，但一般都是在初始化和末尾部分添加密钥，底层置换是公开的。还有一些算法的底层置换是带密钥的置换，安全证明中不存在对底层模块的询问，仅考虑整个状态的碰撞，可用更小的内部状态提供相同的安全性。当底层置换公开时，考虑到每个密钥处理的数据量，为了抵抗基于碰撞的状态恢复攻击，内部状态必须足够大。对于内嵌密钥的设备，使用带密钥的底层置换更易于设计轻量级认证加密算法。

5.1.2　认证加密算法研究进展

认证加密算法发展迅速，各种新型算法百花齐放，虽然可以粗略地分为基于置换/分组密码/流密码的直接设计类以及黑盒调用分组密码的认证加密工作模式，但有些算法兼具多类的特点。认证加密算法是功能复合的算法，不同设计在保护机密性和完整性方面有不同的侧重。此外，认证加密算法的应用场景繁杂多样，由此产生了各种设计和评估指标，比如在线加密、多层次的健壮性、增量标签、并行计算、密文保长、不需要底层模块逆运算、较低的实现代价等。

CAESAR 竞赛和 NIST 的轻量级密码标准化项目征集了 100 多个认证加密算法，集中体现了认证加密算法的研究进展和设计趋势：更加注重具体应用需求，努力提高算法的健

壮性，尽量避免因算法实现或使用不当而导致的安全漏洞。分组密码、流密码和杂凑函数等对称密码设计理论在认证加密算法中得到了充分融合，许多候选算法以加密算法和认证算法作为基本单元或者借鉴其设计理念。CAESAR 竞赛的全称是“Competition for Authenticated Encryption: Security, Applicability, and Robustness”，旨在选拔安全、高效的认证加密算法，推动认证加密算法设计与分析的发展[397,398]。CAESAR 竞赛要求候选认证加密算法应满足两个条件：一是具有广泛的兼容性；二是优于 AES-GCM。

2014 年 3 月，CAESAR 竞赛公布了征集的 57 个算法，随后有 9 个算法因安全缺陷宣布退出竞赛。在进入第一轮评估的 48 个候选算法中，将 AES 等密码算法作为整体调用的认证加密工作模式有 18 个：++AE、AES-CMCC、AES-COPA、AES-CPFB、AES-JAMBU、AES-OTR、AVALANCHE、CBA、CLOC、ELmD、Enchilada、HS1-SIV、iFeed、Julius、OCB、POET、SHELL 和 SILC。其中，ELmD 和 SHELL 为了提高性能，部分模块采用缩减轮 AES。考虑到 AES 指令集对算法软件性能的影响，大部分算法是基于 AES 的；Enchilada 和 HS1-SIV 采用黑盒调用 ChaCha 流密码的方法。直接设计的候选算法有 29 个。有些算法采用了多种设计理念，对它们的分类比较困难而且观点各异。

基于算法的典型设计理念，本书尝试将它们归为如下几类。第一类是基于分组密码的认证加密算法，包含 11 个算法：AEGIS、AEZ、Deoxys、Joltik、KIASU、LAC、PAEQ、SCREAM、Silver、Tiaoxin 和 YAES。第二类是基于置换（杂凑函数）的认证加密算法，包含 12 个算法：Artemia、Ascon、ICEPOLE、Ketje、Keyak、Minalpher、NORX、OMD、PRIMATEs、Prøst、π-Cipher 和 STRIBOB。第三类是基于流密码的认证加密算法，包含 6 个算法：ACORN、MORUS、Raviyoyla、Sablier、TriviA-ck 和 Wheesht。此外，还有设计理念比较特殊的 POLAWIS 算法。

基于分组密码的 11 个候选算法按模块分类，有 8 个算法直接使用了 AES 轮函数，其他 3 个算法采用了分组密码的设计理念。这 11 个算法按整体框架分为 3 类：AEZ、Deoxys、KIASU、Joltik 和 SCREAM 利用了可调分组密码的理念；PAEQ、Silver 和 YAES 采用分组密码构造大置换并进一步设计认证加密算法的思路，本质上这一类设计和基于置换的设计是相通的，只是置换的设计和分析评估采用了分组密码的理念；AEGIS、Tiaoxin 和 LAC 采用流密码框架，此类算法的明文参与状态更新，而且状态更新函数比较简单，所以一般要求临时值 N 不能重用，而且接收方在解密验证阶段需要先认证再输出密文。

2015 年 7 月，CAESAR 竞赛公布了进入第二轮评估的 29 个候选算法。基于分组密码的认证加密算法有 7 个：AEZ、Deoxys、Joltik、SCREAM、AEGIS、Tiaoxin 和 PAEQ。从实现性能方面考虑，AES 指令是现阶段设计认证加密算法绕不开的点。从设计趋势看，基于可调分组密码设计认证加密算法的思路值得关注。基于置换的认证加密算法有 10 个，其中 Ascon、ICEPOLE、Ketje、Keyak、NORX、PRIMATEs、π-Cipher 和 STRIBOB 都采用了 Sponge 类结构，Minalpher 采用了工作模式类的结构，OMD 采用了 MD 结构。从底层置换分类，Ascon、ICEPOLE、Ketje 和 Keyak 采用了 SHA3，PRIMATEs 采用了 AES 类设计思想，NORX 和 π-Cipher 采用 ARX 类的底层置换，另外有两个算法分别使用了 GOST 和 SHA512 的模块。

2016 年 8 月，CAESAR 竞赛公布了进入第三轮评估的 15 个候选算法。2018 年 3 月，CAESAR 竞赛公布了获胜算法：ACORN、AEGIS、Ascon、COLM、Deoxys-Ⅱ、MORUS 和

OCB。2019年,史丹萍等人发现了MORUS的安全问题[399],随后MORUS被撤出。

2013年,美国NIST启动了一个轻量级密码项目,调研NIST现有密码算法标准在资源受限环境中的适应性以及对轻量级密码算法标准的需求[400]。2017年4月,NIST发布了轻量级密码标准化的流程。首先考虑认证加密算法,一类是面向硬件的带关联数据的认证加密算法,另一类是兼顾软硬件性能的带关联数据的认证加密算法和杂凑函数。2018年8月,NIST发布公告正式启动轻量级密码的征集、评估和标准化工作,旨在推出系列轻量级密码算法,用于现有NIST密码标准不适用的资源受限环境。2019年4月,NIST公布了征集到的56个候选算法,8月宣布了进入第二轮的32个候选算法,包含11个工作模式类的认证加密算法、17个基于置换的认证加密算法、两个基于分组密码的认证加密算法以及两个基于流密码的认证加密算法。

11个工作模式类候选算法包括COMET、ESTATE、ForkAE、GIFT-COFB、HyENA、LOTUS-AEAD&LOCUS-AEAD、mixFeed、Romulus、SAEAES、SKINNY-AEAD和SUNDAE-GIFT。从整体框架分类,ESTATE和SUNDAE-GIFT是Two-pass的算法,其余9个都是One-pass的算法。ESTATE、ForkAE、LOTUS-AEAD&LOCUS-AEAD、Romulus和SKINNY-AEAD采用了可调分组密码的设计理念。SKINNY-AEAD和LOCUS采用了OCB类模式。按采用的分组密码分类,COFB、HyENA、LOTUS-AEAD&LOCUS、SUNDAE-GIFT和ESTATE采用了GIFT分组密码,ForkAE、Romulus和SKINNY-AEAD采用了SKINNY分组密码,mixFeed、ESTATE和SAEAES采用了AES-128分组密码。

17个基于置换的候选算法为ACE、ASCON、DryGASCON、Elephant、Gimli、ISAP、KNOT、ORANGE、Oribatida、PHOTON-Beetle、SPARKLE(SCHWAEMM and ESCH)、SPIX、SpoC、Spook、Subterranean 2.0、WAGE和Xoodyak。从整体结构分类,除Elephant采用先加密后认证的方式之外,其余16个算法都采用了Duplex结构及其变体。从底层置换分类,可将这17个候选算法分为3类:6个算法采用了杂凑函数的设计理念,主要参考的杂凑函数是SHA3和PHOTON,其中,Gimli、ISAP、Subterranean 2.0和Xoodyak的底层置换类似SHA3,ORANGE和PHOTON-Beetle采用了PHOTON256;9个算法的底层置换采用了分组密码设计理念,其中,ACE、Oribatida、SPIX和SpoC采用或借鉴了SIMON或Simeck分组密码,ASCON、DryGASCON、KNOT和Spook采用了比特切片技术,SPARKLE采用了ARX类底层置换;WAGE的底层置换采用了流密码的设计思想。

两个基于分组密码的候选算法是Pyjamask和Saturnin。Pyjamask采用了OCB类模式,专门设计了底层分组密码Pyjamask-96和Pyjamask-128。Saturnin结合CTR模式和MAC算法,采用先加密后认证的方式,底层分组密码Saturnin的设计借鉴了3D AES算法。两个基于流密码的候选算法是Grain-128AEAD和TinyJAMBU。Grain-128AEAD基于流密码Grain-128,采用基于泛杂凑函数的MAC。TinyJAMBU的整体框架类似Duplex结构,基于带密钥的置换,其中置换的设计采用了NFSR。

可证明安全性是衡量工作模式类认证加密算法的重要指标。可证明安全是在一定的安全假设下,针对确定的攻击者能力,证明相应的安全目标。常用的安全假设包括伪随机置换、伪随机函数、不可预测函数和理想密码。刻画攻击者能力和手段通常需考虑密码算法运行的环境,目前存在多种对攻击者能力和手段的刻画。分组密码认证加密工作模式的可证

明安全通常不限制攻击者的计算能力，但是限制攻击者向随机预言机查询的次数、查询消息的总长度等。结合不同的安全假设和攻击者能力，形式化定义了多种安全模型：选择明文攻击下的不可区分性(IND-CPA)、选择密文攻击下的不可区分性(IND-CCA)、明文完整性(INT-PTXT)、密文完整性(INT-CTXT)等。如果一个认证加密算法能够同时达到 INT-CTXT 和 IND-CPA，则认为该认证加密算法是安全的。随着临时值与关联数据等概念的加入，相应的使用唯一值与带关联数据的认证加密算法的安全性定义陆续被提出。同时，针对认证加密算法的一些新性质与功能，也提出了一些特殊的安全模型，比如抵抗临时值重用的认证加密算法、依赖临时值的认证加密算法、超越生日界的认证加密算法、在线认证加密 RUP 安全的认证加密算法等[401-403]。

由于轻量级分组密码的分组长度比较短，考虑到量子计算安全性，超越生日界安全成为一些密码算法设计的指标。对于认证加密算法而言，结合 TAE 模式和超越生日界安全的可调分组密码，可以很自然地设计超越生日界安全的认证加密算法，比如 CAESAR 竞赛的获胜算法 Deoxys-Ⅱ和 SKINNY-AEAD 等。问题是如何设计高效的超越生日界安全的可调分组密码。TWEAKEY 是一种可调分组密码的设计框架[200]，将密钥和调柄看成调柄密钥(tweakey)，设计调柄密钥扩展算法以替换分组密码的密钥扩展算法。利用 TWEAKEY 框架的可调分组密码没有理论安全性证明，只能评估其对差分分析、线性分析等分析方法的安全性。王磊等通过两次调用底层分组密码，构造了达到全安全性的可调分组密码[404]。

对于直接设计的认证加密算法，已有的可证明安全理论显示出局限性。由于没有合理化的安全假设，因此不能从可证明安全角度论证其安全性，只能评估其对各种分析方法的安全性，差分分析、线性分析、碰撞攻击、相关密钥攻击、故障攻击等都被用来分析认证加密算法的安全性[405-407]。直接设计的认证加密算法种类多、效率高，是未来发展的趋势，但是安全性基础不够坚固，安全性分析方法与评估技术有待深入研究。

5.2 分组密码认证加密工作模式

在本节中，主要使用如下符号与记号：

A	关联数据。
M	明文。
N	临时值。
K	密钥。
C	密文。
E	分组密码加密算法。
D	分组密码解密算法。
$\widetilde{E}$	可调分组密码的加密算法。
m	M 划分后的分组个数。
n	分组密码的分组长度。
t	标签的比特长度。
S	状态。
T	标签。

0^i　　由 i 个 0 构成的比特串。

1^i　　由 i 个 1 构成的比特串。

10^*　　由 1 个 1 和若干 0 构成的比特串，在常见的填充形式 $X10^*$ 中，星号代表 $n-|X|-1$，即将任意的比特串 $X(|X|<n)$ 填充成长度为 n 比特的分组。

$\leftarrow_n$　　划分运算，$M_1M_2\cdots M_m\leftarrow_n M$ 表示将 M 划分为长度为 n 比特的分组，其中，$|M_i|=n(i=1,2,\cdots,m-1)$ 且 $|M_m|\leqslant n$。

$|X|$　　比特串 X 的比特数。

$|X|_B$　　比特串 X 的字节数。

$[x]_a$　　数值 x 的长度为 a 的比特串表示，其中 $0\leqslant x<2^a$。

$X[a..b]$　　比特串 X 的(左数)第 a 个比特到第 b 个比特。

$\mathrm{lsb}_a(X)$　　比特串 X 的右 a 比特。

$\mathrm{msb}_a(X)$　　比特串 X 的左 a 比特。

Pad　　填充函数，对于任意长度的比特串输入，通过添加比特使比特串的长度为分组整数倍。如无特殊说明，则默认的填充函数为 1-0 填充。对于 $X\in\{0,1\}^*$ 且 $|X|\neq 0$，

$$\mathrm{Pad}_r(X)=\begin{cases}X\parallel 10^{(-|X|-1)\bmod r}, & |X|\bmod r\neq 0\\ X, & |X|\bmod r=0\end{cases}$$

5.2.1　OCB

OCB 是 ISO/IEC、IETF 等采用的标准算法之一。OCB 利用掩码伪装明文和密文，最后一块消息分情况处理，标签的生成利用校验和的方式。OCB 具有可并行计算、消息长度任意、加密保长、底层分组密码调用次数接近最优等特点。OCB 经历了 OCB1、OCB2 和 OCB3 这 3 个版本。OCB1 不具有处理关联数据的操作；掩码生成采用格雷码，通过异或一个预计算值的方式进行掩码更新。OCB2 增加了处理关联数据，构成带关联数据的认证加密算法；采用 Doubling 技术替代格雷码，以期获得更快的计算速度，实现时使用 1 次移位和 1 次或 0 次异或。然而，实验证明 Doubling 的实现速度较慢，因此 OCB3 改回用格雷码生成掩码，利用预计算提升初始化操作的实现速度。设计者曾尝试在 OCB3 中用 LFSR 生成掩码，实验结果显示 LFSR 实现的速度仍略低于 OCB1 中异或预计算值的速度。此外，OCB 在末尾消息的处理等也存在部分优化。作为分组密码工作模式，OCB 在调用底层分组密码以外的代价极低，当使用轻量级分组密码时，OCB 可以实现轻量级的认证加密。

1. OCB 的加密算法

OCB 的加密算法由初始化、处理关联数据、加密、生成标签 4 个部分组成，如图 5-3 所示。记临时值为 N，关联数据为 A，任意长度的明文为 M。加密过程如下：

(1) 初始化：

$\Sigma_{M,0}\leftarrow 0^n$

$C\leftarrow\varepsilon$

$\mathrm{Nonce}\leftarrow 0^{n-|N|}1N$

$\mathrm{Top}\leftarrow\mathrm{Nonce}\wedge 1^{n-6}0^6$

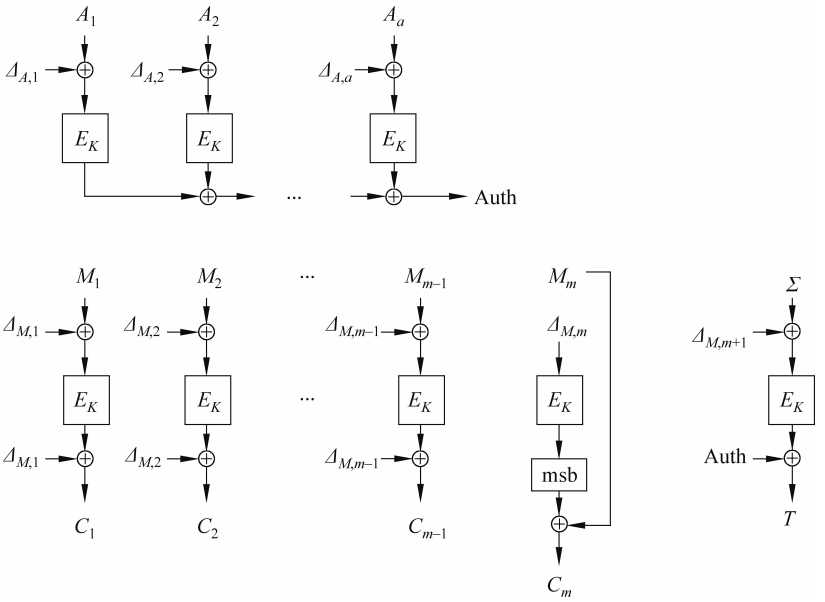

图 5-3　OCB 算法

Bottom←Nonce $\wedge 0^{n-6}1^6$

Ktop←E_K(Top)

Stretch←Ktop ‖ (Ktop⊕(Ktop $\ll 8$))

$\Delta_{M,0}$←msb_n(Stretch $\ll$Bottom)

(2) 处理关联数据：

$A_1A_2\cdots A_a \leftarrow {}_nA$

$\Sigma_{A,0}\leftarrow 0^n$

$\Delta_{A,0}\leftarrow 0^n$

对 i←1,2,…,$a-1$,计算

$\Delta_{A,i}\leftarrow\Delta_{A,i-1}\oplus L[\mathrm{ntz}(i)]$

$\Sigma_{A,i}\leftarrow\Sigma_{A,i-1}\oplus E_K(A_i\oplus\Delta_{A,i})$

根据$|A_a|$分两种情况：

若$|A_a|=n$,则

$\Delta_{A,a}\leftarrow\Delta_{A,a-1}\oplus L[\mathrm{ntz}(a)]$

$\Sigma_{A,a}\leftarrow\Sigma_{A,a-1}\oplus E_K(A_a\oplus\Delta_{A,a})$

若$|A_a|<n$,则

$\Delta_{A,a}\leftarrow\Delta_{A,a-1}\oplus L_*$

$\Sigma_{A,a}\leftarrow\Sigma_{A,a-1}\oplus E_K(\mathrm{Pad}_n(A_a)\oplus\Delta_{A,a})$

Auth←$\Sigma_{A,a}$

(3) 加密：

$M_1M_2\cdots M_m \leftarrow {}_nM$

对 i←1,2,…,$m-1$,计算

$\Delta_{M,i} \leftarrow \Delta_{M,i-1} \oplus L[\mathrm{ntz}(i)]$

$C_i \leftarrow E_K(M_i \oplus \Delta_{M,i}) \oplus \Delta_{M,i}$

$\Sigma_{M,i} \leftarrow \Sigma_{M,i-1} \oplus M_i$

根据$|M_m|$分两种情况：

若$|M_m| = n$，则

$\Delta_{M,m} \leftarrow \Delta_{M,m-1} \oplus L[\mathrm{ntz}(m)]$

$C_m \leftarrow E_K(M_m \oplus \Delta_{M,m}) \oplus \Delta_{M,m}$

$\Sigma_{M,m} \leftarrow \Sigma_{M,m-1} \oplus M_m$

若$|M_m| < n$，则

$\Delta_{M,m} \leftarrow \Delta_{M,m-1} \oplus L_*$

$C_m \leftarrow M_m \oplus \mathrm{msb}_{|M_m|} E_K(\Delta_{M,m})$

$\Sigma_{M,m} \leftarrow \Sigma_{M,m-1} \oplus \mathrm{Pad}_n(M_m)$

最后，再令 $\Delta_{M,m+1} \leftarrow \Delta_{M,m} \oplus L_\$$。

(4) 生成标签：

$\mathrm{Final} \leftarrow E_K(\Sigma_{M,m} \oplus \Delta_{M,m+1})$

$\mathrm{Tag} \leftarrow \mathrm{Final} \oplus \mathrm{Auth}$

$T \leftarrow \mathrm{msb}_t(\mathrm{Tag})$

最后，输出 $C_1 C_2 \cdots C_m \| T$。

在以上加密过程中，ntz(i)是 i 的二进制表示中结尾 0 的个数。

参数 L 由密钥 K 生成，定义如下：

$L_* \leftarrow E_K(0^{128})$

$L_\$ \leftarrow \mathrm{double}(L_*)$

$L[0] \leftarrow \mathrm{double}(L_\$)$

对 $i \leftarrow 1,2,3,\cdots$，计算

$L[i] \leftarrow \mathrm{double}(L[i-1])$

其中，double 为有限域上的乘 2 运算。在 OCB3 中，double 的定义如下：

$\mathrm{double}(X) \leftarrow (X \gg 1) \oplus (\mathrm{msb}_1(X) \cdot 135)$

注意，图 5-3 和 OCB 的加密算法并不完全对应，只给出了部分情况。本章其他一些算法的结构图示也存在类似情况，不再一一解释。

2. OCB 的解密算法

OCB 的解密算法由初始化、处理关联数据、解密和验证标签 4 个部分组成。记临时值为 N，关联数据为 A，密文为 C，标签为 T。解密过程如下：

(1) 初始化(与加密算法相同)。

(2) 处理关联数据(与加密算法相同)。

(3) 解密：

$C_1 C_2 \cdots C_m \leftarrow {}_n C$

对 $i \leftarrow 1,2,\cdots,m-1$，计算

$\Delta_{M,i} \leftarrow \Delta_{M,i-1} \oplus L[\mathrm{ntz}(i)]$

$M_i \leftarrow D_K(C_i \oplus \Delta_{M,i}) \oplus \Delta_{M,i}$

$\Sigma_{M,i} \leftarrow \Sigma_{M,i-1} \oplus M_i$

根据$|C_m|$分两种情况：

若$|C_m| = n$，则

$\Delta_{M,m} \leftarrow \Delta_{M,m-1} \oplus L[\mathrm{ntz}(m)]$

$C_m \leftarrow E_K(M_m \oplus \Delta_{M,m}) \oplus \Delta_{M,m}$

$\Sigma_{M,m} \leftarrow \Sigma_{M,m-1} \oplus M_m$

若$|C_m| < n$，则

$\Delta_{M,m} \leftarrow \Delta_{M,m-1} \oplus L_*$

$M_m \leftarrow C_m \oplus \mathrm{msb}_{|C_m|}(E_K(\Delta_{M,m}))$

$\Sigma_{M,m} \leftarrow \Sigma_{M,m-1} \oplus \mathrm{Pad}_n(M_m)$

最后，再令$\Delta_{M,m+1} \leftarrow \Delta_{M,m} \oplus L_{\$}$。

(4) 验证标签。

用与加密算法相同的方式生成标签T'。

如果$T = T'$，则输出$M_1 M_2 \cdots M_m$；否则，输出$\perp$。

3. OCB 的安全性分析

当底层分组密码为 PRP 或 SPRP 时，OCB 在 CPA 和 CCA 下都可以达到较强的安全性。Ferguson 指出，当出现碰撞时，OCB1 会丧失其认证安全性，并给出了针对 OCB 的碰撞攻击。因此，需要限制 OCB 在同一密钥下处理数据的总量，以保证其安全性[408]。Sun 等人论证了 OCB1、OCB2、OCB3 等都不能抵御碰撞攻击[409]。OCB2 的安全性证明归约为 XEX* 可调分组密码的安全性。2019 年，Inoue 和 Iwata 等人发现 OCB2 不满足 XEX* 安全性证明的前提条件，进一步给出了 OCB2 的伪造攻击和明文恢复攻击[410]。

5.2.2 JAMBU

JAMBU 是 CAESAR 竞赛的候选算法，其设计目标是仅利用 AES 和异或运算的轻量级认证加密算法。相比于其他工作模式类的认证加密算法，JAMBU 具有最小的状态和最小的增量。JAMBU 经历了两个版本：JAMBU v1 要求临时值不可重用；JAMBU v2 增加了临时值重用情况下的安全性说明，JAMBU v2.1 补充了临时值重用情况下的认证安全性证明与硬件实现。

1. JAMBU 的加密算法

JAMBU 的加密算法由初始化、处理关联数据、加密、生成标签 4 个部分组成，如图 5-4 所示。JAMBU 要求总数据量小于$2^{n/2}$比特，临时值N和标签T均为$n/2$比特。状态S由 3 个$n/2$比特的子状态S_1, S_2, S_3组成。记临时值为N，关联数据为A，任意长度的明文为M。加密过程如下：

(1) 初始化：

$(S_{1,0}, S_{2,0}) \leftarrow (0^{n/2}, N)$

$(S_{1,1}, S_{2,1}) \leftarrow E_K(S_{1,0}, S_{2,0}) \oplus ([0]_{n/2}, [5]_{n/2})$

$S_{3,1} \leftarrow S_{1,1}$