1. 消息认证编码-定义

消息认证编码的目的是为了保护消息在传输过程中不被攻击者修改，接收者能验证消息的完整性。

1.1. 消息认证编码的定义

一个消息认证编码MAC（Message Authentication Code）由三个概率多项式（Gen, Mac, Vrfy）时间的算法构成：

1. Gen: 输入安全参数1^n, 输出密钥k, 要求$|k| \geq n$.
2. Mac: 输入密钥k, 消息$m \in \{0,1\}^*$, 输出标签tag, $t \leftarrow Mac_k(m)$.
3. Vrfy: 输入密钥k, 标签t, 输出比特$b \leftarrow Vrfy_k(m,t)$. 若$b=1$, 则表示"有效"，反之表明"无效"。

正确性要求$Vrfy_k(m, Mac_k(m)) = 1$.

1.2 MAC的安全性定义

给定一个MAC方案Π，考虑如下游戏：

1. 挑战者运行$Gen(1^n)$，获得密钥k.
2. 攻击者输入1^n, 适应性地进行消息查询，将消息记为$m_1, m_2, \cdots, m_q$. 挑战者返回对应的MAC $t_1, t_2, \cdots, t_q$.
3. 攻击者宣布其挑战消息m, 并生成对应的MAC t.
4. 攻击者成功当且仅当（1）$Vrfy_k(m,t) = 1$ 并且（2）$m \notin \{m_1, m_2, \cdots, m_q\}$.

将上述游戏中攻击者A成功的概率记为$Pr[S]$.

一个MAC方案Π在适应性选择明文攻击下是存在不可伪造的EU-CMA（Existentially Unforgeable under an adaptive Chosen-Message Attack）如果对于所有的概率多项式时间算法A, 存在一个可忽略的函数$negl(n)$, 使得：

$Pr[S] \leq negl(n)$.

2. 构造安全的MAC

2.1 固定消息长度的MAC方案

Construction 2.1:

假设F为伪随机函数，定义消息长度为n的MAC方案：

1. $Gen(1^n)$: 选择随机均匀密钥$k \leftarrow \{0,1\}^n$, 并输出。
2. $Mac(m,k)$: 输出标签$t := F_k(m)$.（若$|m| \neq |k|$则不输出任何值。）
3. $Vrfy(k,m,t)$: 若$t = F_k(m)$, 则输出1，反之输出0.

Theorem 2.2: 若F为伪随机函数，则上述构造为安全的定长MAC方案。

Proof: 假设A为一个概率多项式时间的攻击者，定义两个游戏Game 0, Game 1.

Game 0:

1. 挑战者选择密钥$k \leftarrow \{0,1\}^n$.
2. 攻击者适应性地查询$Q = \{m_1, \cdots, m_q\}$ 对应的tag, 挑战者返回$\{t_1, \cdots, t_q\}$, 其中$t_i = F_k(m_i)$.

3. 攻击者输出消息-tag对(m,t), 若$t=F_k(m), m\notin Q$, 则攻击者成功。

将攻击者在Game 0中成功的概率记为S_0.

将伪随机函数改为真随机函数f, 得到Game 1.

Game 1:

1. 挑战者选择密钥$k\leftarrow\{0,1\}^n$.
2. 攻击者适应性地查询$Q=\{m_1,\cdots,m_q\}$ 对应的tag, 挑战者返回$\{t_1,\cdots,t_q\}$, 其中$t_i=f(m_i)$.
3. 攻击者输出消息-tag对(m,t), 若$t=f(m), m\notin Q$, 则攻击者成功。

将攻击者在Game 1中成功的概率记为S_1.

Claim 1: $Pr[S_1]=2^{-n}$.

Proof of Claim 1: 显然当m从未被查询过时，$f(m)$的值为$\{0,1\}^n$上的随机元素，刚好取到t的概率为2^{-n}.

Claim 2: $|Pr[S_0]-Pr[S_1]|\leq negl(n)$.

Proof of Claim 2: 我们使用归约论证来证明该结论，假设攻击者A能成功伪造MAC, 我们构造新的算法D来区分真随机函数与伪随机函数。

算法D:

算法D的输入为1^n, 并且D能查询预言$O:\{0,1\}^n\rightarrow\{0,1\}^n$, 该算法的作用为区分$O$究竟是 真随机函数还是伪随机函数。

1. 运行$A(1^n)$, 当A用消息m来查询MAC预言时，如下回答:

   (a)用m查询预言O, 返回值记为t, 将t值给A.

2. 当A结束运行并输出(m,t)时:

   (a)用m查询预言O, 返回t'.

   (b)如果$t=t'$并且A之前从来没有查询过m, 则输出1, 反正输出0.

显然，D是多项式时间算法。

算法D的分析:

Case 1: 当D的预言O是伪随机函数时，攻击者A在Game 0中，此时D输出1当且仅当A在Game 0中成功，所以，

$Pr[D^{F_k(\cdot)}(1^n)=1]=Pr[S_0]$.

Case 2: 当D的预言O是真随机函数时，攻击者A在Game 1中，此时D输出1当且仅当A在Game 1中成功，所以，

$Pr[D^{f(\cdot)}(1^n)=1]=Pr[S_1]$.

因为F_k为伪随机函数，所以我们有:

$|Pr[S_0]-Pr[S_1]|=|Pr[D^{F_k(\cdot)}(1^n)=1]-Pr[D^{f(\cdot)}(1^n)=1]|\leq negl(n)$.

Claim 2证明完毕。

结合Claim 1和Claim 2我们有$|Pr[S_0]-2^{-n}|\leq negl(n)$, 即$Pr[S_0]\leq 2^{-n}+negl(n)$是可忽略的。

所以攻击者A成功伪造的概率可忽略，构造2.1是安全的MAC方案。

2.2. 任意消息长度的MAC方案

假设$\Pi' = (Mac', Vrfy')$是定长的MAC方案，将m进行分块获得$m_1, \cdots, m_d$.

思路1: 直接分块计算$t_i := Mac_k^{'}(m_i)$, 输出$< t_1, t_2, \cdots, t_d >$. 但这无法抵抗**块重排序攻击**，假设$(t_1, t_2)$为$(m_1, m_2)$对应的有效MAC. 则攻击者知道$(m_2, m_1)$对应的有效MAC是$(t_2, t_1)$.

思路2: 在每个块上加上块指数i, $t_i := Mac_k'(i||m_i)$, 这就能抵御**块重排序攻击**。但这无法抵御**截断攻击**，攻击者直接将最后一个区块去掉。比如(t_1, t_2)为(m_1, m_2)对应的有效MAC. 则攻击者知道m_1对应的有效MAC是t_1.

思路3: 每个块上加上消息的长度l, $t_i := Mac_k'(l||i||m_i)$, 这能抵御**截断攻击**，因为截断后长度改变。但这无法抵挡**混淆攻击**。比如攻击者已知(m_1, m_2)的MAC值为(t_1, t_2), (m_1', m_2')的MAC值为(t_1', t_2'). 则攻击者能知道(m_1, m_2')的MAC值为(t_1, t_2').

Construction 2.3:

假设$\Pi' = (Mac', Vrfy')$是固定长度n的MAC方案，则如下定义一个新的MAC：

1. $Gen(1^n)$: 选择随机均匀密钥$k \leftarrow \{0,1\}^n$, 并输出。
2. $Mac(m, k)$: 输入的消息$m \in \{0,1\}^*$长度为$l < 2^{n/4}$. 将 m 分成 d 个块$m_1, m_2, \cdots, m_d$, 每个块长度为$n/4$（最后的块用0补足）。选择随机标识符$r \in \{0,1\}^{n/4}$.

   对于$i = 1, 2, \cdots, d$, 计算$t_i \leftarrow Mac_k'(r||l||i||m_i)$, i, l都用$n/4$长度的字符串来表示。输出标签$t = (r, t_1, t_2, \cdots, t_d)$.
3. $Vrfy(k, m, t)$: 输入标签$t = (r, t_1', \cdots, t_{d'}')$，输入的消息$m \in \{0,1\}^*$长度为$l < 2^{n/4}$. 将$m$分成$d$个块$m_1, m_2, \cdots, m_d$, 每个块长度为$n/4$（最后的块用0补足）。首先验证$d = d'$是否成立，其次对于$1 \le i \le d$ 验证$t_i' = Mac_k'(r||l||i||m_i)$. 若验证均通过，输出1，反之输出0.

Theorem 2.4: 假设Π'为长度为n安全的定长MAC方案，那么上述构造是任意长度的安全MAC方案。

Proof: 假设A为概率多项式时间的攻击者，我们把A成功伪造上述MAC方案记为事件S, 我们要证明$Pr[S]$是可忽略的。

符号说明：

令$(m, t =< r, t_1, \cdots >)$为攻击者最后的输出，其中$m = m_1, \cdots$.

Repeat: 攻击者查询MAC预言得到的一系列tag值中，有两个随机标识符一样。

NewBlock: 至少有一个块$r||l||i||m_i$在A的预言查询时从来未被Mac_k'认证过。

$Pr[S] = Pr[S \wedge Repeat] + Pr[S \wedge \overline{Repeat} \wedge NewBlock] + Pr[S \wedge \overline{Repeat} \wedge \overline{NewBlock}]$

$\le Pr[Repeat] + Pr[S \wedge NewBlock] + Pr[S \wedge \overline{Repeat} \wedge \overline{NewBlock}]$

我们证明右边三个概率都是可忽略的。

Claim 1: $Pr[Repeat]$是可忽略的。

Proof of Claim 1: 令事件$H_{i,j}$为第i次查询和第j次查询选择的随机标识符一致。则由于标识符的长度为$2^{n/4}$, 我们有$Pr[H_{i,j}] = \frac{1}{2^{n/4}}$. 假设攻击者查询的次数为$q(n)$, q为某个多项式，那么

$Pr[Repeat] = Pr[\cup_{1 \le i < j \le q(n)} H_{i,j}] = \sum_{i,j} Pr[H_{i,j}] = C_{q(n)}^2 \frac{1}{2^{n/4}} \le \frac{q(n)^2}{2^{n/4}}$

该值为可忽略的值。

Claim 2: $Pr[S \wedge \overline{Repeat} \wedge \overline{NewBlock}] = 0$

我们证明如果攻击者伪造成功且MAC查询时随机标识符都不相同，则**NewBlock**必然发生。

Proof of Claim 2: 再次假设$q = q(n)$为攻击者进行MAC查询的次数。事件**Repeat**没有发生，则$r_1, r_2, \ldots, r_q$各不相同。若$r \notin \{r_1, r_2, \ldots, r_q\}$, 则显然**NewBlock**发生。

若存在j, 使得$r = r_j$, 假设r_j为攻击者第j次查询的标识符，对应的消息为$m^{(j)}$, 长度为l_j.

Case 1: $l \neq l_j$, 则所有块的第二个分量都不一样，显然块$r||l||1||m_1$从未被查询过。**NewBlock**显然发生。

Case 2: $l = l_j$, 则由于攻击者伪造成功了，我们有$m \neq m^{(j)}$. 假设m与$m^{(j)}$的第i个块不一致，即$m_i \neq m_i^{(j)}$. 那么显然$r||l||i||m_i$是新的块。**NewBlock**也发生。

综上所述，$S \wedge \overline{Reapeat}$ 发生则**NewBlock**一定发生。 $Pr[S \wedge \overline{Repeat} \wedge \overline{NewBlock}] = 0.$

Claim 3: $Pr[S \wedge NewBlock]$是可忽略的。

我们用归约来证明Claim 3，假设有多项式时间攻击者A成功伪造MAC，我们构造新的多项式时间攻击者A来伪造方案Π'的MAC，这与我们的假设相反。

Proof of Claim 3:

攻击者A': 输入为1^n, 能用$Mac'_k(\cdot)$进行MAC预言查询。

把攻击者A当成子程序。

1. 当攻击者A用 m（长度为 l）进行$Mac_k(\cdot)$查询时:

   (a)将m分成长度为$n/4$的块$m_1, m_2, \ldots m_d$（假设有d个块），最后不够的用0补足。

   (b)随机选择标识符$r \in \{0,1\}^{n/4}$, 对于$1 \leq i \leq d$, 用 $r||l||i||m_i$ 查询预言$Mac'_k(\cdot)$得到t_i.

   (c)返回$< r, t_1, \ldots, t_d >$.

2. 当攻击者A输出伪造$(m, t =< r, t_1, \ldots, t_d >$时，检查$NewBlock$是否发生:

   (a)若发生，假设$r||l||i||m_i$是新的块从未被认证过，则输出$< r||l||i||m_i, t_i >$.

   (b)若未发生，输出失败。

显然，攻击者A作为A'的子程序时和在原始的攻击游戏中完全一致。

假设$NewBlock$ 发生，则块 $r||l||i||m_i$ 从来未被认证过。

假设S发生，攻击者A伪造成功，则所有块的tag值均有效，$t_i = Mac'_k(r||l||i||m_i)$.

所以如果$S \wedge NewBlock$发生，则攻击者A'成功伪造了$r||l||i||m$ 的MAC值。

假设攻击者A'成功的事件为S', 我们有$Pr[S \wedge NewBlock] \leq Pr[S']$. 根据我们的假设 Π' 为安全的MAC方案，所以$Pr[S']$可忽略，从而得到 $Pr[S \wedge NewBlock]$ 可忽略。

综合Claim 1,2,3, $Pr[S]$可忽略，该方案是安全的MAC方案。

3. CBC-MAC

上一节构造的MAC方案效率低，比如计算消息长度为dn的MAC值，需要进行$4d$次分组计算，最后的tag长度为$4dn$. 本节给出更有效的构造。

3.1. 基本构造

Construction 3.1:

假设F是伪随机函数，固定长度函数$l > 0$. 基本的CBC-MAC方案如下构造：

1. $Gen(1^n)$: 随机选择$k \leftarrow \{0,1\}^n$, 输出作为密钥。
2. $Mac(k,m)$: 输入消息的长度为$l(n) \cdot n$, 进行如下计算：

   (a). 将m分成$m_1, m_2, \ldots, m_l$, 每个m_i的长度为n.

   (b). 令$t_0 = 0^n$, 对于i从1到l, 计算：

   $t_i = F_k(t_{i-1} \oplus m_i)$.

   (c). 输出t_l作为tag.
3. $Vrfy(t,m,k)$: 若m的长度不是$l(n) \cdot n$ 输出0，否则输出1当且仅当$t = Mac(k,m)$.

Theorem 3.2: 假设F是伪随机函数，l为多项式，则上述方案对于长度为$l(n) \cdot n$的消息来说是安全的MAC方案。

下一节证明一个更一般的结论。

*3.2. 安全性证明

本节稍有难度，可以选择跳过不影响后面学习。

定义CBC函数，输入为$(\{0,1\}^n)^*$中的元素（也就是长度为n的倍数）以及长度为n的密钥映射到长度为n的字符串。

$CBC_k(x_1, \ldots, x_l) = F_k(F_k(\cdots F_k(F_k(x_1) \oplus x_2) \oplus \cdots) \oplus x_l)$.

$|x_1| = |x_2| = \cdots = |x_l| = |k| = n$.

一个字符串集合$P \subset (\{0,1\}^n)^*$是无前缀的如果它不包含空串以及任何字符串 $X \in P$ 都不是其它字符串 $X' \in P$ 的前缀。

Theorem 3.3: 对于任意的概率多项式时间算法D, D 能进行预言查询，但所有查询的字符串构成的集合是无前缀的。存在可忽略的函数$negl$, 使得，

$|Pr[D^{CBC_k(\cdot)}(1^n) = 1] - Pr[D^{f(\cdot)}(1^n) = 1]| \leq negl(n)$.

我们可以用一个编码函数$encode$, 将任意长度的字符串m映射到 $encode(m) \in (\{0,1\}^n)^*$, 然后输出$CBC_k(encode(m))$. 该编码方案需要是无前缀的。

我们证明当CBC的“密钥”是一个随机函数g时的安全性，也就是说定义

$CBC_g(x_1, \ldots, x_l) = g(g(\cdots g(g(x_1) \oplus x_2) \oplus \cdots) \oplus x_l)$.

我们证明$CBC_g(\cdot)$与$(\{0,1\}^n)^*$到$\{0,1\}^n$上的随机函数就是不可区分。

Theorem 3.4: 固定$n \geq 1$, 取随机函数$g : \{0,1\}^n \rightarrow \{0,1\}^n$, f :$(\{0,1\}^n)^* \rightarrow \{0,1\}^n$, 对于任意（无时间限制）算法$D$, D 能进行q次预言查询，但这q次查询的字符串构成的集合是无前缀的，并且所有的查询中最长的消息包含l个区块。则

$|Pr[D^{CBC_g(\cdot)}(1^n) = 1] - Pr[D^{f(\cdot)}(1^n) = 1]| \leq \frac{q^2 l^2}{2^n}$.

Proof of Theorem 3.4: 假设$P = \{X_1, \ldots, X_q\}$为无前缀的q次查询，每个$X_i \in (\{0,1\}^n)^*$, 且P中最长的消息包含l个区块。对于任意的$t_1, \ldots, t_q \in \{0,1\}^n$, 有$Pr[\forall i : f(X_i) = t_i] = 2^{-nq}$.

我们称CBC是$(q,l,\delta) - smooth$的如果对于每个无前缀集合$P = \{X_1, \ldots, X_q\}$, 以及每个$t_1, \ldots, t_q \in \{0,1\}^n$,有

$Pr[\forall i : CBC_g(X_i) = t_i] \geq (1-\delta) \cdot 2^{-nq}$.

也就是说 $(q,l,\delta) - smooth$ 的CBC是 $\delta - close$ 真随机函数 f 的。

Claim 3.5: CBC_g是$(q,l,\delta)-smooth$的，其中 $\delta=q^2l^2\cdot 2^{-n}$.

下面假设Claim 3.5成立。

定义函数$\alpha(X_1,\ldots,X_q;t_1,\ldots,tq)=1$当且仅当$D$查询$X_1,\ldots,X_q$时返回$t_1,\ldots,t_q$, 且$D$最后输出1.

令$\overrightarrow{X}=\{X_1,\ldots,X_q\}$, $\overrightarrow{t}=\{t_1,\ldots,t_q\}$, 我们有

$\operatorname{Pr}\left[D^{\mathrm{CBC}_g(\cdot)}\left(1^n\right)=1\right]$

$=\sum_{\vec{X} \text{ prefix-free; } \vec{t}} \alpha(\vec{X},\vec{t})\cdot \operatorname{Pr}[\forall i: \mathrm{CBC}_g\left(X_i\right)=t_i]$

$\geq \sum_{\vec{X} \text{ prefix-free; } \vec{t}} \alpha(\vec{X},\vec{t})\cdot(1-\delta)\cdot \operatorname{Pr}[\forall i: f\left(X_i\right)=t_i]$

$=(1-\delta)\cdot \operatorname{Pr}\left[D^{f(\cdot)}\left(1^n\right)=1\right]$,

这证明了

$\operatorname{Pr}\left[D^{f(\cdot)}\left(1^n\right)=1\right]-\operatorname{Pr}\left[D^{\mathrm{CBC}_g(\cdot)}\left(1^n\right)=1\right]\leq\delta\cdot\operatorname{Pr}\left[D^{f(\cdot)}\left(1^n\right)=1\right]\leq\delta.$

完成**Theorem 3.4**的证明。

下面进行Claim 3.5的证明。

Proof of Claim 3.5:

令$X\in(\{0,1\}^n)^*$, 且$X=x_1,\ldots$并且, $|x_i|=n$.

令$C_g(x)$是 计算$CBC_g(X)$过程中g的输入构成的集合。例如假设$X\in(\{0,1\}^n)^m$, 那么

$\mathcal{C}_g(X)\stackrel{\text{def}}{=}\left(x_1, \mathrm{CBC}_g\left(x_1\right)\oplus x_2,\ldots,\mathrm{CBC}_g\left(x_1,\ldots,x_{m-1}\right)\oplus x_m\right).$

令$X\in(\{0,1\}^n)^m$, $X'\in(\{0,1\}^n)^{m'}$.

$C_g(X)=(I_1,\ldots,I_m), C_g(X')=(I'_1,\ldots,I'_{m'})$.

称X内部存在非平凡碰撞如果存在 $I_i=I_j$ 但是 $i\neq j$.

称X,X'之间存在非平凡碰撞如果存在 $I_i=I'_j$ 但是$(x_1,\ldots,x_i)\neq\left(x'_1,\ldots,x'_j\right)$.

称字符串集合$P=\{X_1,\ldots,X_q\}$ 存在非平凡碰撞若存在 $X\in P$ 有内部碰撞或者存在$X,X'\in P$之间存在碰撞。

定义事件 $Coll$ 为集合P存在非平凡碰撞。

我们分两部来完成证明：

第一步：如果 $Coll$ 未发生，则 $\operatorname{Pr}\left[\forall i: \mathrm{CBC}_g(X_i)=t_i|\overline{\mathrm{Coll}}\right]=2^{-nq}$.

第二步：证明 $Coll$ 发生的概率小于$\delta\leq q^2l^2\cdot 2^{-n}$.

对于X, 随机均匀选择 $g(I_1)$ 的值，确定$I_2=g(I_1)\oplus x_2$, 随机均匀选择$g(I_2)$的值，如此重复，直到选择 $g(I_{m-1})$ 的随机值（无需选取$g(I_m)$的值，因为$g(I_m)\notin C_g(X)$）。用上述方式计算出字符串$X_1,\ldots,X_q$对应的$C_g(X_1),\ldots,C_g(X_q)$，然后就可检查事件 $Coll$ 是否发生。

假设事件$Coll$未发生，那么$C_g(X_1),\ldots,C_g(X_q)$中的最后一项必不相同。我们证明g在这些最后一项上的值未被确定，假设$C_g(X)$的最后一项I_m的g值已经被确定，那么肯定存在I'_j, $I_m=I'_j$并且 I'_j 不是$C_g(X')$的最后一项。但是由于事件$Coll$未发生，上述事件只有当$(x_1,\ldots,x_m)=(x'_1,\ldots,x'_j)$发生时才发生，但由于$X\neq X'$, 我们得到$X$是$X'$的前缀，这与$P$无前缀相矛盾。

$CBC_g(X_i)$的值就是g在$C_g(X_i)$上最后一项的值。根据上述分析，g在最后一项上的值未被确定，所以从$\{0,1\}^n$上独立均匀选出，选择到某个固定值的概率为2^{-n}. 选择任意的$t_1,\ldots,t_q\in\{0,1\}^n$, 有

$\Pr\left[\forall i:\mathrm{CBC}_g(X_i)=t_i|\overline{\mathrm{Coll}}\right]=2^{-nq}.$

第一步证明完毕，下面进行第二步证明。

给定$X_i,X_j\in P$, 令$Coll_{i,j}$为X_i,X_j中内部存在碰撞或者X_i,X_j之间存在碰撞，则

$Coll=\vee_{i,j}\ \mathrm{Coll}_{\ i,j}$,

$\Pr[\mathrm{Coll}]\leq\sum_{i,j:i<j}\Pr[\mathrm{Coll}_{i,j}]=\begin{pmatrix}q\\2\end{pmatrix}\cdot\Pr[\mathrm{Coll}_{i,j}]\leq\frac{q^2}{2}\cdot\Pr[\mathrm{Coll}_{i,j}].$

令$X=X_i,X'=X_j$, 两者的块长度都是l. $X=(x_1,\ldots,x_l),X'=(x'_1,\ldots,x'_l)$, t是最大的整数使得

$(x_1,\ldots,x_t)=(x'_1,\ldots,x'_t)$.（其中 $t<l$ 否则 $X=X'$）

注意到此时$(I_1,\ldots,I_t)=(I'_1,\ldots,I'_t)$.

用下面 $2l-t-2$ 步来确定 g 的值:

i 从1到 $t-1$:

随机均匀选择$g(I_i)$的值从而确定 I_{i+1}和I'_{i+1}.（注意这步两者相等）

$i=t$:

随机均匀选择$g(I_t)$的值从而确定 I_{t+1}和I'_{t+1}.（注意这步两者不等）

i 从 $t+1$ 到 $l-1$:

随机均匀选择 $g(I_{t+1}),g(I_{t+2}),\ldots,g(I_{l-1})$ 的值从而确定 $I_{t+2},I_{t+3},\ldots,I_l$.

i 从 l 到 $2l-t-2$:

随机均匀选择 $g(I'_{t+1}),g(I'_{t+2}),\ldots,g(I'_{l-1})$ 的值从而确定 $I'_{t+2},I'_{t+3},\ldots,I'_l$.

假设事件$Coll(k)$表示第k步发生了非平凡碰撞，那么

$\Pr[\mathrm{Coll}_{i,j}]=\Pr[\vee_k\ \mathrm{Coll}(k)]\leq\Pr[\mathrm{Coll}(1)]+\sum_{k=2}^{2l-t-2}\Pr[\mathrm{Coll}(k)|\overline{\mathrm{Coll}(k-1)}]$.

若 $k<t$, $Coll(k-1)$未发生，则$(I_1,..,I_k)$各不相同，$I_{k+1}=g(I_k)\oplus x_{k+1}$与前$k$个值发生碰撞的概率为$\frac{k}{2^n}$, 所以$\Pr[\mathrm{Coll}(k)|\overline{\mathrm{Coll}(k-1)}]=k/2^n$.

若 $k=t$, $\Pr[\mathrm{Coll}(k)|\overline{\mathrm{Coll}(k-1)}]\leq 2t/2^n$, 因为这一步生成了两个值$I_{t+1},I'_{t+1}$且不相同。

若 $k>t$, $\Pr[\mathrm{Coll}(k)|\overline{\mathrm{Coll}(k-1)}]\leq(k+1)/2^n$, 因为第$t$步多生成了一个值，所以加上1.

我们得出

$\Pr[\mathrm{Coll}_{i,j}]\leq 2^{-n}\cdot\left(\sum_{k=1}^{t-1}k+2t+\sum_{k=t+1}^{2l-t-2}(k+1)\right)$

$=2^{-n}\cdot\sum_{k=2}^{2l-t-1}k=2^{-n}\cdot(2l-t+1)\cdot(2l-t-2)/2<2\ell^2\cdot 2^{-n}$.

所以，我们有

$Pr[Coll]\leq\frac{q^2}{2}Pr[Coll_{i,j}]<q^2l^2\cdot 2^{-n}=\delta.$

综合所有，我们有

$Pr[\forall i: CBC_g(X_i)=t_i] \geq Pr[\forall i: CBC_g(X_i)=t_i|\overline{Coll}]\cdot Pr[\overline{Coll}] \geq 2^{-nq}\cdot Pr[\overline{Coll}] \geq (1-\delta)\cdot 2^{-nq}$.

这就证明了 CBC_g 是$(q,l,\delta)-smooth$的。

4. 认证加密

如何设计加密方案使其既保证安全性又保证消息的完整性?

4.1. 定义

我们规定密文必须满足某种条件，攻击者无法伪造出有效的密文。

考虑攻击者与挑战者A之间不可伪造的加密游戏:

1. 挑战者运行$Gen(1^n)$获得私钥k.
2. 攻击者A输入1^n以及可以用消息m_i进行加密查询，挑战者计算$c_i=Enc_k(m_i)$并返回。
3. 攻击者A最后输出一段密文c.
4. 令Q表示攻击者进行的加密查询集合，挑战者计算$m=Dec_k(c)$, 若密文无效输出$\perp$.攻击者在该游戏中成功当且仅当$m\neq\perp$且$m\notin Q$.

把上述攻击者成功的事件记为事件S.

定义4.1: 我们称一个对称加密方案 Π 是不可伪造的，如果对于任意的概率多项式时间攻击者，存在一个可忽略的函数 $negl$ 使得$Pr[S]\leq negl(n)$.

定义4.2: 如果一个加密方案是CCA安全并且不可伪造的，那么称该方案是一个认证加密。

4.2. 通用构造

假设$\Pi_E=(Enc,Dec)$为CPA安全的加密方案，$\Pi_M=(Mac,Vrfy)$ 为MAC方案，k_E,k_M为对应方案的密钥。考虑下面三种方案:

1. **加密并认证:** 密文为$<c,t>$, 其中$c\leftarrow Enc_{k_E}(m), t\leftarrow Mac_{k_M}(m)$. 接收者先解密得到$m$, 再验证$Vrfy(m,t)=1$是否成立，若成立，输出$m$, 反之输出错误。
2. **先认证，再加密:** 密文为c, 先计算$t\leftarrow Mac_{k_M}(m)$, 再计算$c\leftarrow Enc_{k_E}(m||t)$. 接收者先解密得到$m||t$, 再验证$Vrfy(m,t)=1$是否成立，若成立，输出$m$, 反之输出错误。
3. **先加密，再认证:** 密文为$<c,t>$, 先计算$c\leftarrow Enc_{k_E}(m)$, 再计算$t\leftarrow Mac_{k_M}(c)$, 接收者先验证$Vrfy(c,t)=1$是否成立，若成立解密出m, 反之输出错误。

注意MAC方案不保证不泄露原文的一些消息，所以直接计算明文的MAC值并作为密文的一部分并不安全，方案1，2不适用。

考虑第3种方案，先加密，后认证:

Construction 4.3:

假设$\Pi_E=(Enc,Dec)$为私钥加密方案，$\Pi_M=(Mac,Vrfy)$ 为消息认证编码。定义如下的私钥加密方案:

1. Gen': 随机均匀选择密钥$k_E,k_M\in\{0,1\}^n$.
2. Enc': 输入消息m以及密钥k_E,k_M, 计算$c\leftarrow Enc_{k_E}(m)$, 再计算$t\leftarrow Mac_{k_M}(c)$, 输出密文$<c,t>$.
3. Dec': 输入密文$<c,t>$, 密钥k_E,k_M, 首先验证$Vrfy(c,t)=1$是否成立，若成立，输出$m=Dec_{k_E}(c)$, 反之输出$\perp$.

Theorem 4.4: 假设Π_E为CPA安全的私钥加密方案，Π_M是安全的消息认证编码，则上述构造为一个认证加密方案。

Proof of Theorem 4.4: 非形式化地说，由于密文包含 c 的MAC值，而由于我们假设Π_M是不可伪造的MAC方案，所以攻击者无法伪造出c的MAC值，也就是说无法伪造出有效的密文。这就说明了该方案是不可伪造的方案。并且由于密文不可伪造性，攻击者进行解密查询的密文都是无效的（有可忽略的概率可能有效），解密预言返回$\perp$, 所以解密预言并没有给攻击者额外的帮助，所以若 Π_E 是CPA安全的话，该方案是CCA安全的。

假设A是概率多项式时间的攻击者在CCA-Game中攻击构造4.3.

称密文$< c,t >$是新的如果A没从加密预言中获得过$< c,t >$.

定义事件$ValidQuery$为A提交给解密预言的密文是新的且是有效的，$Vrfy(c,t)=1$.

Claim 4.5: $Pr[ValidQuery]$是可以忽略的。

Proof of Claim 4.5: 令$q(n)$为攻击者A进行解密查询的次数，我们构造新的攻击者A_M, 把A作为子程序调用，在MAC伪造Game中攻击MAC方案Π_M.

攻击者A_M: 输入1^n, 允许其能进行MAC预言查询$Mac_{k_M}(\cdot)$.

1. $k_E \leftarrow \{0,1\}^n, i \in \{1,2,\ldots,q(n)\}$.
2. 当A用m进行加密查询时：
   (1). $c \leftarrow Enc_{k_E}(m)$,
   (2). 用c进行MAC预言查询，获得$c \leftarrow Mac_{k_M}(\cdot)$. 将$< c,t >$返回给$A$.
   (3). 用同样的方式生成挑战密文。
3. 当A用$< c,t >$进行解密查询时，若这恰好是第i次查询，A_M直接输出$< c,t >$, 否则：
   (1). 若$< c,t >$是之前A用m加密查询生成的，直接返回m.
   (2). 否则输出$\perp$.

显然A_E是概率多项式时间算法。

直观上看，A_M预测A第i次查询 $< c,t >$ 是新的且有效的查询。则 $< c,t >$ 从未被A_M查询过，A_M 成功伪造了c 的MAC值t.

令事件S_M为A_M成功伪造了MAC值。A进行新且有效查询的概率为$Pr[ValidQuery]$, 该查询恰好是第i次查询的概率为$Pr[ValidQuery]/q(n)$. 所以$Pr[S_M]=Pr[ValidQuery]/q(n)$. 而我们假设 Π_M 是安全的MAC方案，所以$Pr[S_M]$是可忽略的，而又$q(n)$是多项式，所以$Pr[ValidQuery]$ 也是可忽略的。

下面证构造3.5 Π' 是密文不可伪造的。

假设攻击者 A' 能对方案 Π' 进行密文伪造，A将A'作为子程序调用，当A'成功伪造$< c,t >$时，A 用 $< c,t >$进行密文查询。但是我们已经证明A进行有效密文查询的概率可忽略，所以A'伪造成功的概率也可忽略，所以 Π' 是不可伪造的加密方案。

下面证Π' 是CCA安全的。

令事件 S 为攻击者A在CCA游戏中成功击破了方案Π'.

$Pr[S]=Pr[S \wedge ValidQuery]+Pr[S \wedge \overline{ValidQuery}] \leq Pr[ValidQuery]+Pr[S \wedge \overline{ValidQuery}]$.

已知$Pr[ValidQuery]$可忽略，所以只要证明如下Claim.

Claim 4.6: $Pr[S \wedge \overline{ValidQuery}] \leq \frac{1}{2}+negl(n)$.

Proof of Claim 4.6: 直观上看由于$ValidQuery$事件未发生，所有解密查询都是无效的，解密预言未提供任何额外能力，所以如果Π_E是CPA安全的话，Π'也是CCA安全的。

我们构造攻击者A_E在CPA-Game中攻击方案Π_E.

攻击者A_E: 输入为1^n, 能进行加密查询$Enc_{k_E}(\cdot)$.

1. $k_M \leftarrow \{0,1\}^n$.
2. 当A用消息m进行加密查询时，
   (1). 用m进行加密查询$Enc_{k_E}(\cdot)$, 获得$c \leftarrow Enc_{k_E}(\cdot)$.
   (2). 计算$t \leftarrow Mac_{k_M}(c)$, 将$< c, t >$返回给$A$.
3. 当A用$< c, t >$进行解密查询时，
   (1). 若$< c, t >$是之前消息m加密查询的返回值，直接输出m.
   (2). 反之，输出$\perp$.
4. A提交两段等长明文m_0, m_1，
   (1). A_E 将这两段明文提交给自己的挑战者得到挑战密文c.
   (2). 计算$t \leftarrow Mac_{k_M}(c)$, 将$< c, t >$作为$A$的挑战密文.
5. 当A输出1个bit后，A_E 输出相同的bit.

显然，A_E是概率多项式时间的算法。

当事件$ValidQuery$未发生时，A作为A_E的子程序与在原始的CCA-Game中完全一致。

令事件S_E为A_E在CPA-Game中成功。从上述算法中我们可得，A_E成功当且仅当A成功，所以

$Pr[S_E \wedge \overline{ValidQuery}] = Pr[S \wedge \overline{ValidQuery}]$.

我们得出

$Pr[S_E] \geq Pr[S_E \wedge \overline{ValidQuery}] = Pr[S \wedge \overline{ValidQuery}]$

又 Π_E 是CPA安全的，$Pr[S_E] \leq 1/2 + negl(n)$. 所以$Pr[S \wedge \overline{ValidQuery}] \leq \frac{1}{2} + negl(n)$.

所以，$Pr[S] \leq Pr[ValidQuery] + Pr[S \wedge \overline{ValidQuery}] \leq 1/2 + negl'(n)$.

Π' 是CCA安全的方案。

综上Π'是安全的认证加密方案。

4.3. CCA安全加密

可伪造与CCA安全不等价，存在可伪造但还是CCA安全的方案。

认证加密与CCA安全其实也不等价，CCA安全方案与认证加密目的不同，认证加密我们要求的是消息的完整性与安全性，CCA安全不考虑消息的完整性，只考虑能进行解密查询的敌手。在公钥加密体制中，CCA安全方案与认证加密的区别较大。