I240 暗号理論 2019
モード型共通鍵暗号と MAC（2）

2019/10/30 講師 藤崎

1 はじめに

モード型共通鍵暗号、MAC に共通する長い可変長の平文をどういうルールで分割するかを説明する。CBC モードは共通鍵暗号、MAC への利用ともに脆弱性があることを解説する。

2 メッセージの分割とパディング

モード型共通鍵暗号、MAC では、任意長の平文 $m \in \{0,1\}^*$ を内部で固定長のブロックに分解し、ブロック暗号に入力して行く。ブロック暗号のブロックのサイズ n は方式によるが、AES ならば $n = 128$ ビット（16 バイト）である。この時、平文長が（ブロック暗号への平文（暗号文）の入力が n ビットとして）n の倍数になっていない場合、padding によりデータを埋めて n の倍数の長さにする必要がある。例えば、IETF (Internet Engineering Task Force) の RFC2040 では余ったバイト数の数だけ、そのバイト数を表現する数を埋める。例えば、22, 333, 55555 のように。実際は 16 進表現で、

0x02 0x02,　　0x03 0x03 0x03,　　0x05 0x05 0x05 0x05 0x05

のようなパディングが後方につけられる。平文がブロックの長さのぴったり倍数である場合は（1 ブロック＝16 バイトの場合)、余分に 1 ブロック追加し、$\overbrace{\mathsf{0x10}\cdots\mathsf{0x10}}^{16}$ で埋める。

他に、余ったバイトを全て 0x00 で埋めて（決まっていれば 0x00 でなくても良い)、1 ブロック余計に加えてそのブロックに m のビット数 $|m|$ を書き出すなど。これ以外にも様々なパディング方法があるがここでは紹介しない。

可変長の平文を扱うニーズは確実にあるが、しばしば脆弱性の原因となる（CBC モード暗号や CBC-MAC のケース)。

3 CBC モード暗号への攻撃

■CBC モード暗号（再掲載）　$\mathbf{E}, \mathbf{D}$ をそれぞれブロック暗号の暗号化アルゴリズム、復号アルゴリズムとする。平文 $m \in \{0,1\}^*$ をブロック暗号の入力サイズ n により t 個に分割 $m = m_1|\cdots|m_t$ $(m_i \in \{0,1\}^n)$ した後、図 2 のように暗号化、復号を行う。復号の際、平文のパディングが正しいか確認し、パディングが正しくない場合は $\perp$（エラー）を返す。

■CBC モード暗号への攻撃　RFC2040 の平文のパディングを少し一般化する。ブロック暗号の 1 ブロックを b 個のワードからできているとし、W を 1 ワード取りうる個数とする（$b \le W$)。RFC2040 では、$W = 256$, $b = 16$ $n = 128$ となっている。平文が $(b - w)$ ワード（$1 \le w < b$）から出来ていれば、最後の w ワードを $\overbrace{\mathsf{w}\cdots\mathsf{w}}^{w}$ で埋める（平文がちょうど b ワードなら、1 ブロック追加して、$\overbrace{\mathsf{b}\cdots\mathsf{b}}^{b}$ で埋める)。

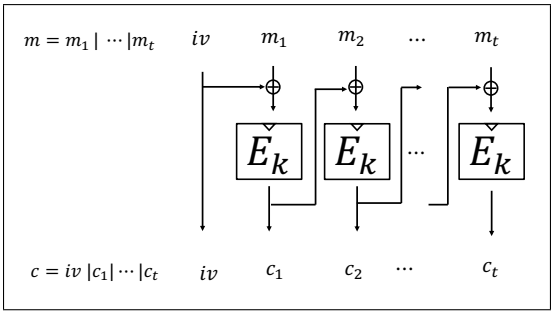

図 1　CBC モード

CBC-Enc $\mathbf{E}^{\mathrm{CBC}}(k, m)$
$iv \leftarrow \{0,1\}^n$; $c_0 := iv$;
for $i = 1, \ldots, t$ do
$c_i := \mathbf{E}(k, c_{i-1} \oplus m_i)$;
return $c := iv|c_1|\cdots|c_t$.

CBC-DEC $\mathbf{D}^{\mathrm{CBC}}(k, c)$:
Parse as $c = c_0|c_1|\cdots|c_t$;
for $i = 1, \ldots, t$ do
$x_i := D(k, c_i) \oplus c_{i-1}$
If x_t is a correct padding,
return $m := x_1|\cdots|x_t$,
otherwise $\perp$.

図 2　CBC-Encryption

■パディングオラクル攻撃　実環境で、自分の望む暗号文を自由に復号してくれる選択暗号文攻撃可能な環境はなかなか無い。だが、暗号文が正しい暗号文かそれとも壊れた正しく無い暗号文かを判定してくれる環境は実環境でも十分ありうる。実際 TLS/SSL などでもそのようなことが起きる。パディングオラクルとは、暗号文を復号したとき、平文のパディングが正しければ 1 を、正しくなければ 0 を返し、作られた暗号文が正しいものか否かを判定してくれるオラクルである。Vaudenay はそのようなパディングオラクルがあれば、CBC モード暗号を破れることを示した。

定理 1（Vaudenay [Vau02]）　平文へのパディングが RFC2040 に準拠するものとする。このとき、敵は高々 NbW 回のパディングオラクルへのアクセスで、ターゲット暗号文を全て解読できる。ここで、N はターゲット暗号文 c のブロック数（$|c| = nN$）である。実際は、平均 $\frac{NbW}{2}$ 回のアクセスで暗号文を解読できる。

(証明) 簡単のためターゲット暗号文 $c = (iv, c_1)$（$iv, c_1 \in \{0,1\}^n$）とする（すなわち、$N = 1$）。敵は、$r_b \in [1, W-1]$ を一つ選び、最初の $b-1$ ワードを 0 にしたブロック暗号の 1 ブロック（n ビット長）を作る。

$$R_b = (\overbrace{0, \ldots, 0}^{b-1}, r_b) \in \{0,1\}^n.$$

次に、暗号文 $c' = ((R_b \oplus iv), c_1)$ をパディングオラクルへ渡す。パディングオラクルは $x' = (R_b \oplus iv) \oplus D_k(c_1)$ を計算し、x' のパディングが正しければ 1 を返し、そうでなければ 0 を返す。いま、ターゲット暗号文の平文を $m = (x_1, \ldots, x_b)$ とする（$x_i \in [0, W-1]$）。x' の最後のワードは $r_b \oplus x_b$ であることに注意する。もし、$r_b \oplus x_b = 1$ であれば、x' は $(b-1)$ ワードの平文に最後のワードに 1 をパディングしたものとなるので、パディングオラクルは 1 を返し、そうでなければ、0 を返してくる。よって、$r_b \in [1, W]$ をすべてのワードで試すことで、どこかで、$r_b \oplus x_b = 1$ を見つけることができ、それにより、$x_b = r_b \oplus 1$ を求めることができる。

この手順を繰り返すことで平文全てを次のように求めることができる。x_b はすでに知っているので、$r_b := x_b \oplus 2$ と r_b を改めて定義しなおす。適当なワード r_{b-1} を選び、

$$R_{b-1} = (\overbrace{0, \ldots, 0}^{b-2}, r_{b-1}, r_b) \in \{0,1\}^n$$

とする。次に暗号文 $c' = ((R_{b-1} \oplus iv), c_1)$ をパディングオラクルへ渡す。前と同様、パディングオラクルは $x' = (R_b \oplus iv) \oplus D_k(c_1)$ を計算する。x' の最後のワードは、$2(= x_b \oplus r_b)$ となっているので、もし、$x_{b-1} \oplus r_{b-1} = 2$ であれば正しいパディングとなる。r_{b-1} を全てのワードで試すことで、$x_{b-1} \oplus r_{b-1} = 2$ を見つけられるので、x_{b-1} を得られる。いま、x_{b-1}, x_b がわかるようになったので、$r_{b-1} := x_{b-1} \oplus 3$, $r_b := x_b \oplus 3$ と改めて定義し同じことを繰り返すことで、最終的に平文 $m = (x_1, \ldots, x_b)$ を全て求めることができる。その時、パディングオラクルに問い合わせる回数は高々 bW 回になる。

もし、暗号文が $c = (iv, c_1, \ldots, c_N)$ であれば、平文全てを求めるのに最大 NbW 回のパディングオラクルへのアクセスが必要となる。

問題 2　もし、IND-CCA ゲームであれば、敵は高々 W 回の復号オラクルへのアクセスでゲームに勝つだけでなく、ターゲットの平文を全て知ることができることを示せ。

注釈 3　平文を RFC2040 パディングのようにしたのち、余剰の 1 ブロックに平文のビット数を書き込む $|m|$ 方法では、Vaudenay の攻撃は本質的に防げないことが知られている。

パディングをどうにかして CBC モード暗号を安全にするという試みは証明可能安全性という観点では恐らく成功していない。

4　CBC-MAC と EMAC

4.1　MAC の安全性モデル（再掲載）

メッセージ認証 $\mathsf{MAC} = (\mathcal{K}, \mathcal{M}, \mathsf{S}, \mathsf{V})$ の (q, ϵ)-安全性は、次のようなゲームを通じて定義される。鍵の長さ（セキュリティパラメータ）を $\kappa := \log |\mathcal{K}|$ と書くことにする。

■EUF-CMA ゲーム（MAC 版）　敵 (adversary) A とチャレンジャー C の間で行われるメッセージ認証方式 MAC の安全性を試すゲームである [*1]。

1. C は鍵空間 $\mathcal{K}$ から一様ランダムに鍵 k を選ぶ。$k \leftarrow \mathcal{K}$.
2. A は平文 m を $\mathsf{S}(k, \cdot)$ オラクルに送り、対応する認証子 $\tau \leftarrow \mathsf{S}(k, m)$ を答えとしてもらう。平文 m の選び方は A の戦略による。A は最大 q 回まで質問 (query) をすることができる。A が、平文を質問しその答えとして認証子をもらう行為を、A の $\mathsf{S}(k, \cdot)$ オラクルへのアクセスと呼ぶ。$\mathsf{S}(k, \cdot)$ オラクルへのアクセス履歴を $L = \{(m_1, \tau_1), \ldots, (m_q, \tau_q)\}$ とし、アクセス平文の集合を特に $L(m) = \{m_1, \ldots, m_q\}$ と書くことにする。
3. A は、平文と認証子の組 (m^*, τ^*) を出力する。
4. もし、m^* が、$\mathsf{S}(k, \cdot)$ オラクルへ質問したことのない新しい平文であり $(m^* \notin L(m))$, $\mathsf{V}(k, m^*, \tau^*) = 1$ を満たすのであれば、A の勝ちと定義する。

A が、メッセージ認証方式 MAC に対する EUF-CMA ゲームに勝つ確率を、A の MAC に対するアドバン

[*1] チェレンジャーは明示的には表現されないときもある。

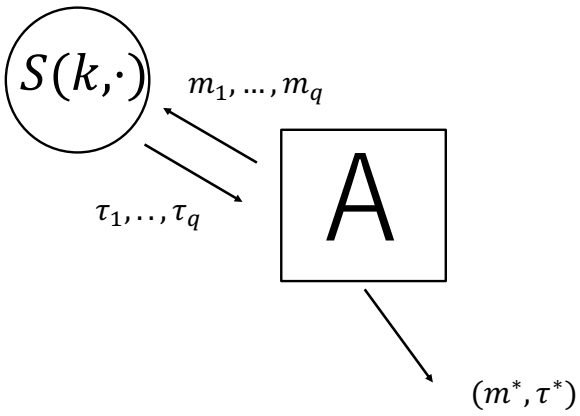

図 3　MAC の EUF-CMA ゲーム

テージと呼び、

$$\mathsf{Adv}^{\mathsf{euf\text{-}cma}}_{A,\mathsf{MAC}}(q,\kappa) := \Pr[k \leftarrow \mathcal{K};\ (m^*,\tau^*) \leftarrow A^{\mathsf{S}(k,\cdot)} : \mathsf{V}(k,m^*,\tau^*) = 1 \text{ and } m^* \notin L(m)].$$

定義 4（メッセージ認証の安全性） 最大 q 回の質問をする任意の敵 A のアドバンテージが、$\mathsf{Adv}^{\mathsf{euf\text{-}cma}}_{A,\mathsf{MAC}}(q,\kappa) \leq \epsilon$ であるとき、メッセージ認証方式 MAC は (q,ϵ)-安全であるという。特に、$q = O(\kappa)$ のとき、$\mathsf{Adv}^{\mathsf{euf\text{-}cma}}_{A,\mathsf{MAC}}(\kappa) = \mathsf{negl}(\kappa)$ であるならば、メッセージ認証方式 MAC は安全であるという。

注釈 5　$q = 1$ のとき、MAC は OT-安全ということにする。$(1,\epsilon)$-安全 $= \epsilon$-OT-安全.

4.2　モード型 MAC

■CBC-MAC　CBC-MAC の仕様。認証子作成は、平文 $m \in \{0,1\}^*$ をブロック暗号の入力サイズ n により t 個に分割 $m = m_1|\cdots|m_t$ $(m_i \in \{0,1\}^n)$ した後、図 4 のように秘密鍵 k を使って認証子 τ を生成する。認証子の検証は、(m,τ) を受け取った後、m から図 4 のように秘密鍵 k を使って認証子 τ' を作成し、$\tau = \tau'$ であれば受理し、$\tau \neq \tau'$ であれば否認する。

Input m.
Output τ.
　Parse $m = m_1|\cdots\cdots|m_t$;
　$c_0 := 0^n$;
　For $i = 1,\ldots,t$ do
　　$c_i := \mathbf{E}(k, c_{i-1} \oplus m_i)$;
　return $\tau := c_t$.

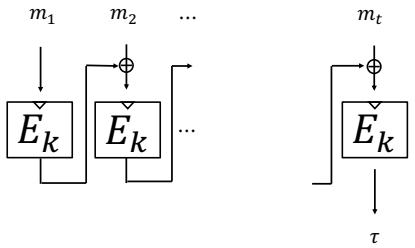

図 4　CBC-MAC

CBC-MAC は平文の長さが固定長のときは安全性証明がつくことが知られているが [BKR00]、逆に可変長のときは安全性に問題があることが知られている。

4.2.1 CBC-MAC への攻撃

$\mathbf{E}_k^{(1)}(m_1) := \mathbf{E}_k(m_1)$ とし、

$$\mathbf{E}_k^{(t)}(m_1, \ldots, m_t) := \mathbf{E}_k(\mathbf{E}_k^{(t-1)}(m_1, \ldots, m_{t-1}) \oplus m_t) \quad \text{where } t \geq 2.$$

と定義する。そのとき、$m = m_1|\cdots|m_t$ の CBC-MAC の認証子は $\tau = \mathbf{E}_k^{(t)}(m_1, \ldots, m_t)$ となる。

敵は次のように CBC-MAC を攻撃できる。任意の $m = m_1|\cdots|m_t$（$t \geq 1$）を MAC オラクルに質問し、認証子 τ をもらう。偽造の平文と認証子のペアーとして、

$$m' := m_1 \,|\, m_2 \,|\, \cdots \,|\, m_t \,|\, (\tau \oplus m_1) \,|\, m_2 \,|\, \cdots \,|\, m_t$$

と τ を出力する。

$$\mathbf{E}_k^{(t+1)}(m_1, \ldots, m_t, \tau \oplus m_1) = \mathbf{E}_k(\tau \oplus (\tau \oplus m_1)) = \mathbf{E}_k(m_1)$$

に注意すると、τ が m' の認証子になっていることがわかる。これは、RFC2040 のパディングをしても防げないことはすぐわかるが、実は後ろに m のビット長 $|m|$ をブロックとして追加しても偽造が防げない。

$$\mathbf{E}_k^*(m_1, \ldots, m_t) := \mathbf{E}_k(\mathbf{E}_k^{(t)}(m_1, \ldots, m_t) \oplus |m|)$$

と定義する。後ろに $|m|$ をパディングする方式を採用した場合、$m = m_1|\cdots|m_t$ の認証子は $\tau = \mathbf{E}_k^*(m_1, \ldots, m_t)$ になる。今、敵は次のような 3 つの質問を MAC オラクルにし、認証子を受け取る。m_1, m_2, m' は 1 ブロック（$= n$ ビット）とするが同じ長さなら何でも良い。

- $\tau_1 = \mathbf{E}_k^*(m_1) = \mathbf{E}_k^{(2)}(m_1, n) = \mathbf{E}_k(\mathbf{E}_k(m_1) \oplus n)$
- $\tau_2 = \mathbf{E}_k^*(m_2) = \mathbf{E}_k^{(2)}(m_2, n) = \mathbf{E}_k(\mathbf{E}_k(m_2) \oplus n)$
- $\tau' = \mathbf{E}_k^*(m_1, n, m') = \mathbf{E}_k^*(\mathbf{E}_k(m_1) \oplus n, m') = \mathbf{E}_k^*(\tau_1 \oplus m')$
  $= \mathbf{E}_k^{(2)}(\tau_1 \oplus m', 3n) = \mathbf{E}_k(\mathbf{E}_k(\tau_1 \oplus m') \oplus 3n)$

上の 3 式から、$3n$ ビットの平文 $m_3 := m_2|n|(\tau_1 \oplus \tau_2 \oplus m')$ の認証子は

$$\begin{aligned}\mathbf{E}_k^*(m_2, n, (\tau_1 \oplus \tau_2 \oplus m_3)) &= \mathbf{E}_k^*((\mathbf{E}_k(m_2) \oplus n), (\tau_1 \oplus \tau_2 \oplus m')) \\ &= \mathbf{E}_k^*(\tau_2 \oplus (\tau_1 \oplus \tau_2 \oplus m')) \\ &= \mathbf{E}_k^*(\tau_1 \oplus m')\end{aligned}$$

よって、$\tau' = \mathbf{E}_k^*(m_2, n, (\tau_1 \oplus \tau_2 \oplus m_3))$ なので、平文 m_3 への認証子が偽造できた。

問題 6 CBC-MAC は RFC2040 のパディングをしても安全にならないことを示せ。

■EMAC CBC-MAC の最後の c_t を認証子にする代わりに、独立の秘密鍵 k' で $\tau = \mathbf{E}(k', c_t)$ と c_t を暗号化し、τ を認証子とする。EMAC は平文の長さがブロックの整数倍（すなわち、$|m| = n\ell$）で有る限り、可変長であっても「安全」になる。EMAC をさらに少し変形して、平文の長さが任意の可変長であっても安全にすることができる [BR05].

参考文献

[BKR00] Mihir Bellare, Joe Kilian, and Phillip Rogaway. The security of the cipher block chaining message authentication code. *Journal of Computer and System Sciences*, 61(3):362–399, 2000.

Input m.
Output τ.
  Parse $m = m_1|\cdots\cdots|m_t$;
  $c_0 := 0^n$;
  For $i = 1, \ldots, t$ do
    $c_i := \mathbf{E}(k, c_{i-1} \oplus m_i)$;
  $\tau := \mathbf{E}(k', c_t)$;
  return τ.

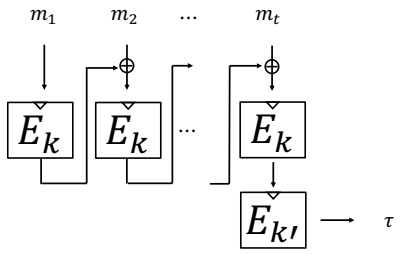

図 5　EMAC

[BR05] John Black and Phillip Rogaway. CBC MACs for arbitrary-length messages: The three-key constructions. *Journal of Cryptology*, 18(2):111–131, April 2005.

[Vau02] Serge Vaudenay. Security flaws induced by CBC padding - applications to SSL, IPSEC, WTLS... In Lars R. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 534–546. Springer, Heidelberg, April / May 2002.