(19)日本国特許庁(JP)　(12)特　許　公　報(B2)　(11)特許番号
特許第6519473号
(P6519473)
(45)発行日　令和1年6月5日(2019.6.5)　(24)登録日　令和1年5月10日(2019.5.10)

(51)Int.Cl.			F I		
H04L	9/06	(2006.01)	H04L	9/00	611Z
G09C	1/00	(2006.01)	G09C	1/00	610A
H04L	9/32	(2006.01)	H04L	9/00	675A

請求項の数 12　(全 59 頁)

(21)出願番号　特願2015-529336(P2015-529336)
(86)(22)出願日　平成26年6月24日(2014.6.24)
(86)国際出願番号　PCT/JP2014/003382
(87)国際公開番号　WO2015/015702
(87)国際公開日　平成27年2月5日(2015.2.5)
審査請求日　平成29年5月11日(2017.5.11)
(31)優先権主張番号　特願2013-161446(P2013-161446)
(32)優先日　平成25年8月2日(2013.8.2)
(33)優先権主張国　日本国(JP)

(73)特許権者　000004237
日本電気株式会社
東京都港区芝五丁目７番１号
(74)代理人　100103090
弁理士　岩壁　冬樹
(74)代理人　100124501
弁理士　塩川　誠人
(72)発明者　峯松　一彦
東京都港区芝五丁目７番１号　日本電気株式会社内

審査官　中里　裕正

最終頁に続く

(54)【発明の名称】認証暗号装置、認証暗号方法および認証暗号用プログラム

(57)【特許請求の範囲】
【請求項１】
入力された平文または暗号文に対して、2ブロックごとに、補助変数を入れた暗号化関数をラウンド関数に用いた2ラウンドFeistel構造を適用して暗号文または復号された平文を生成し、入力された前記平文または復号された前記平文のうちの一部のビットを用いて算出されるチェックサムに対して前記暗号化関数を適用して認証タグを生成する認証暗号手段を備え、
前記認証暗号手段は、暗号化手段を含み、
前記暗号化手段は、
暗号化対象の平文と初期ベクトルとを入力する平文入力手段と、
前記初期ベクトルと入力された平文のサイズとに基づき、前記暗号化関数の各々に与える補助変数を生成する補助変数生成手段と、
前記平文を2ブロックごとのチャンクに分けたときの各平文チャンクに対して2ラウンドFeistel構造を適用することにより、当該平文チャンクに対応する暗号文チャンクを生成する2ラウンドFeistel暗号化手段と、
前記平文のチェックサムを計算し、得られたチェックサムに対して、補助変数を入れた暗号化関数を適用させて認証タグを生成するタグ計算手段とを有し、
前記2ラウンドFeistel暗号化手段は、初期ベクトルをN、チャンクのインデックスをi、i番目の平文チャンクをMC[i] = (M[i_1], M[i_2])、当該平文チャンクMC[i]に含まれる2つの平文ブロックに対応する補助変数を(N,Tw_i_1)と(N,Tw_i_2)の組、暗号化関数をF_K(

,)とすると、i番目の暗号文チャンクCC[i] = (C[i_1], C[i_2])を、
C[i_1] = F_K((N,Tw_i_1), M[i_1]) xor M[i_2] ,
C[i_2] = F_K((N,Tw_i_2), C[i_1]) xor M[i_1]
と求め、
前記タグ計算手段は、平文のチェックサムを、各平文チャンクに含まれる平文ブロックM[i_2]を用いて計算し、得られたチェックサムをSUM、認証タグ生成時に用いる暗号化関数に与える補助変数を(N,Tw_T_1)、暗号化関数をF_K(*,*)とすると、認証タグTを、
T = F_K((N,Tw_T_1), SUM)
と求め、
前記補助変数生成手段は、入力された平文のサイズが偶数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズに満たない場合に、認証タグ生成時に用いる暗号化関数に与える補助変数として、第2の認証タグ用補助変数を生成し、
前記暗号化手段は、
入力された平文のサイズが偶数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズに満たない場合に、最終の平文ブロックを含む最終の平文チャンクに対して、所定の2ラウンドFeistel構造を適用して、最終の暗号文ブロックを含む最終の暗号文チャンクを生成する第2の2ラウンドFeistel暗号化手段と、
入力された平文のサイズが偶数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズに満たない場合に、平文のチェックサムを、入力された平文と、前記第2の2ラウンドFeistel暗号化手段からの出力とを用いて計算し、得られたチェックサムに対して補助変数を与えた暗号化関数を適用させて、認証タグを生成する第2のタグ計算手段とを有し、
前記第2の2ラウンドFeistel暗号化手段は、最終の平文チャンクのインデックスをm、最終の平文ブロックをM[m_2]、最終の平文チャンクをMC[m] = (M[m_1], M[m_2])、最終の平文チャンクMC[m]に含まれる2つの平文ブロックに対応する補助変数を(N,Tw_m_1)と(N,Tw_m_2)の組、暗号化関数をF_K(*,*)、最終の平文ブロックのサイズをs、ブロックサイズをn、sビットからnビットへのパディング処理をpad_n()、nビットからsビットへのカッティング処理をcut_s()とすると、sビットの最終の暗号文ブロックC[m_2]を含む最終の暗号文チャンクCC[m] = (C[m_1], C[m_2])を、
C[m_2] = cut_s(Z) xor M[m_2],
C[m_1] = F_K((N,Tw_m_2), pad_n(C[m_2])) xor M[m_1]
ただし、Z = F_K((N,Tw_m_1), M[m_1])
と求め、
前記第2のタグ計算手段は、平文のチェックサムを、前記最終の平文チャンクを除く各平文チャンクに含まれる平文ブロックM[i_2]と、前記Zと、前記C[m_2]をnビットにパディングした結果であるC_n[m_2]と、を用いて計算し、得られたチェックサムをSUM、前記第2の認証タグ用補助変数を(N,Tw_T_2)、暗号化関数をF_K(*,*)とすると、認証タグTを、
T = F_K((N,Tw_T_2), SUM)
と求める
ことを特徴とする認証暗号装置。

【請求項２】
入力された平文または暗号文に対して、2ブロックごとに、補助変数を入れた暗号化関数をラウンド関数に用いた2ラウンドFeistel構造を適用して暗号文または復号された平文を生成し、入力された前記平文または復号された前記平文のうちの一部のビットを用いて算出されるチェックサムに対して前記暗号化関数を適用して認証タグを生成する認証暗号手段を備え、
前記認証暗号手段は、復号手段を含み、
前記復号手段は、
復号対象の暗号文と初期ベクトルと認証タグとを入力する暗号文入力手段と、
前記初期ベクトルと入力された暗号文のサイズとに基づき、前記暗号化関数の各々に与

える補助変数であって暗号化時と同じ補助変数を生成する復号用補助変数生成手段と、
前記暗号文を2ブロックごとのチャンクに分けたときの各暗号文チャンクに対して2ラウンドFeistel構造を適用することにより、当該暗号文チャンクに対応する、復号された平文チャンクを生成する2ラウンドFeistel復号手段と、
前記復号された平文のチェックサムを計算し、得られたチェックサムに対して、補助変数を入れた暗号化関数を適用させて復号検証用の認証タグを生成する復号検証用タグ計算手段と、
前記復号検証用タグ計算手段が生成した復号検証用の認証タグと入力された認証タグとに基づいて、復号の成功または失敗を判定する判定手段とを有し、
前記2ラウンドFeistel復号手段は、初期ベクトルをN、チャンクのインデックスをi、i番目の暗号文チャンクをCC[i] = (C[i_1], C[i_2])、当該暗号文チャンクCC[i]に含まれる2つの暗号文ブロックに対応する補助変数を(N,Tw_i_1)と(N,Tw_i_2)の組、暗号化関数をF_K(*,*)とすると、i番目の復号された平文チャンクMC'[i] = (M'[i_1], M'[i_2])を、
M'[i_1] = F_K((N,Tw_i_2), C[i_1]) xor C[i_2],
M'[i_2] = F_K((N,Tw_i_1), M'[i_1]) xor C[i_1]
と求め、
前記復号検証用タグ計算手段は、復号された平文のチェックサムを、復号された各平文チャンクに含まれる復号された各平文ブロックM'[i_2]を用いて計算し、得られたチェックサムをSUM'、復号検証用の認証タグ生成時に用いる暗号化関数に与える補助変数を(N,Tw_T_1)、暗号化関数をF_K(*,*)とすると、復号検証用の認証タグT'を、
T' = F_K((N,Tw_T_1), SUM')
と求め、
前記復号用補助変数生成手段は、入力された暗号文のサイズが偶数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズに満たない場合に、復号検証用の認証タグ生成時に用いる暗号化関数に与える補助変数として、暗号時と同じ第2の認証タグ用補助変数を生成し、
前記復号手段は、
入力された暗号文のサイズが偶数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズに満たない場合に、最終の暗号文ブロックを含む最終の暗号文チャンクに対して、所定の2ラウンドFeistel構造を適用して、最終の暗号ブロックの復号された平文ブロックを含む最終の復号された平文チャンクを生成する第2の2ラウンドFeistel復号手段と、
入力された暗号文のサイズが偶数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズに満たない場合に、復号された平文のチェックサムを、前記2ラウンドFeistel復号手段からの出力と、前記第2の2ラウンドFeistel復号手段からの出力と、前記最終の暗号文ブロックとを用いて計算し、得られたチェックサムに対して補助変数を与えた暗号化関数を適用させて、復号検証用の認証タグを生成する第2の復号検証用タグ計算手段とを有し、
前記第2の2ラウンドFeistel復号手段は、最終の暗号文チャンクのインデックスをm、最終の暗号文ブロックをC[m_2]、最終の暗号文チャンクをCC[m] = (C[m_1], C[m_2])、最終の暗号文チャンクCC[m]に含まれる2つの暗号文ブロックに対応する補助変数を(N,Tw_m_1)と(N,Tw_m_2)の組、暗号化関数をF_K(*,*)、最終の暗号文ブロックのサイズをs、ブロックサイズをn、sビットからnビットへのパディング処理をpad_n()、nビットからsビットへのカッティング処理をcut_s()とすると、sビットの最終の復号された平文ブロックM'[m_2]を含む最終の復号された平文チャンクMC'[m] = (M'[m_1], M'[m_2])を、
M'[m_1] = F_K((N,Tw_m_2), pad_n(C[m_2])) xor C[m_1],
M'[m_2] = cut_s(Z') xor C[m_2],
ただし、Z' = F_K((N,Tw_m_1), M'[m_1])
と求め、
前記第2の復号検証用タグ計算手段は、復号された平文のチェックサムを、前記最終の

復号された平文チャンクを除く復号された各平文チャンクに含まれる復号された平文ブロックM'[i_2]と、前記Z'と、前記C[m_2]をnビットにパディングした結果であるC_n[m_2]と、を用いて計算し、得られたチェックサムをSUM'、前記第2の認証タグ用補助変数を(N,Tw_T_2)、暗号化関数をF_K(*,*)とすると、復号検証用の認証タグT'を、

T' = F_K((N,Tw_T_2), SUM')

と求める

ことを特徴とする認証暗号装置。

【請求項３】

前記補助変数生成手段は、入力された平文のサイズが奇数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズと同じである場合に、認証タグ生成時に用いる暗号化関数に与える補助変数として、第3の認証タグ用補助変数を生成し、入力された平文のサイズが奇数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズに満たない場合に、認証タグ生成時に用いる暗号化関数に与える補助変数として、第4の認証タグ用補助変数を生成し、

前記暗号化手段は、

入力された平文のサイズが奇数のブロックに分割されるサイズである場合に、最終の平文ブロックを含む最終の平文チャンクに対して、所定の1ラウンドFeistel構造を適用して、最終の暗号文ブロックを含む最終の暗号文チャンクを生成する1ラウンドFeistel暗号化手段と、

入力された平文のサイズが奇数のブロックに分割されるサイズである場合に、平文のチェックサムを、入力された平文と、前記1ラウンドFeistel暗号化手段からの出力とを用いて計算し、得られたチェックサムに対して補助変数を与えた暗号化関数を適用させて、認証タグを生成する第3のタグ計算手段とを有し、

前記1ラウンドFeistel暗号化手段は、最終の平文チャンクのインデックスをm、最終の平文ブロックをM[m_1]、最終の平文チャンクをMC[m] = (M[m_1])、最終の平文ブロックに対応する補助変数を(N,Tw_m_1)、暗号化関数をF_K(*,*)、最終の平文ブロックのサイズをs、ブロックサイズをn、nビットからsビットへのカッティング処理をcut_s()とすると、sビットの最終の暗号文ブロックC[m_1]を含む最終の暗号文チャンクCC[m] = (C[m_1])を、

C[m_1] = cut_s(F_K((N,Tw_m_1),0^n)) xor M[m_1]

ただし、s=nのときはcut_s()は省略可能

と求め、

前記第3のタグ計算手段は、もしs=nであれば、平文のチェックサムを、前記最終の平文チャンクを除く各平文チャンクに含まれる平文ブロックM[i_2]と、前記M[m_1]をnビットにパディングした結果であるM_n[m_1]（ただし、s=nのときはパディング処理は省略可能）、とを用いて計算し、得られたチェックサムをSUM、前記第3の認証タグ用補助変数を(N,Tw_T_3)、暗号化関数をF_K(*,*)とすると、認証タグTを、

T = F_K((N,Tw_T_3), SUM)

と求め、もしs<nであれば、平文のチェックサムを、前記M[i_2]と前記M_n[m_1]とを用いて計算し、得られたチェックサムをSUM、前記第4の認証タグ用補助変数を(N,Tw_T_4)、暗号化関数をF_K(*,*)とすると、認証タグTを、

T = F_K((N,Tw_T_4), SUM)

と求める

請求項１に記載の認証暗号装置。

【請求項４】

前記復号用補助変数生成手段は、入力された暗号文のサイズが奇数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズと同じである場合に、復号検証用の認証タグ生成時に用いる暗号化関数に与える補助変数として、暗号時と同じ第3の認証タグ用補助変数を生成し、入力された暗号文のサイズが奇数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズに満たない場合に、復号検証用の認証タグ生成時に用いる暗号化関数に与える補助変数として、暗号時と同じ第4

の認証タグ用補助変数を生成し、
前記復号手段は、
入力された暗号文のサイズが奇数のブロックに分割されるサイズである場合に、最終の暗号文ブロックを含む最終の暗号文チャンクに対して、所定の1ラウンドFeistel構造を適用して、最終の復号された平文ブロックを含む最終の平文チャンクを生成する1ラウンドFeistel復号手段と、
入力された暗号文のサイズが奇数のブロックに分割されるサイズである場合に、復号された平文のチェックサムを、前記2ラウンドFeistel復号手段からの出力と、前記1ラウンドFeistel復号手段からの出力とを用いて計算し、得られたチェックサムに対して補助変数を与えた暗号化関数を適用させて、復号検証用の認証タグを生成する第3の復号検証用タグ計算手段とを有し、
前記1ラウンドFeistel復号手段は、最終の暗号文チャンクのインデックスをm、最終の暗号文ブロックをC[m_1]、最終の暗号文チャンクをCC[m] = (C[m_1])、最終の暗号文ブロックに対応する補助変数を(N,Tw_m_1)、暗号化関数をF_K(*,*)、最終の暗号文ブロックのサイズをs、ブロックサイズをn、nビットからsビットへのカッティング処理をcut_s()とすると、sビットの最終の復号された平文ブロックM'[m_1]を含む最終の復号された平文チャンクMC'[m] = (M'[m_1])を、
M'[m_1] = cut_s(F_K((N,Tw_m_1),0^n)) xor C[m_1]
ただし、s=nのときはcut_s()は省略可能
と求め、
前記第3の復号検証用タグ計算手段は、もしs=nであれば、復号された平文のチェックサムを、前記最終の復号された平文チャンクを除く復号された各平文チャンクに含まれる平文ブロックM'[i_2]と、前記M'[m_1]をnビットにパディングした結果であるM_n'[m_1]（ただし、s=nのときはパディング処理は省略可能）と、を用いて計算し、得られたチェックサムをSUM'、前記第3の認証タグ用補助変数を(N,Tw_T_3)、暗号化関数をF_K(*,*)とすると、復号検証用の認証タグT'を、
T' = F_K((N,Tw_T_3), SUM')
と求め、もしs<nであれば、復号された平文のチェックサムを、前記M'[i_2]と前記M_n'[m_1] とを用いて計算し、得られたチェックサムをSUM'、前記第4の認証タグ用補助変数を(N,Tw_T_4)、暗号化関数をF_K(*,*)とすると、復号検証用の認証タグT'を、
T' = F_K((N,Tw_T_4), SUM')
と求める
請求項２に記載の認証暗号装置。
【請求項５】
暗号化関数が、Tweakと呼ばれる補助変数を含む2変数入力のTweakable ブロック暗号である
請求項１から請求項４のうちのいずれか１項に記載の認証暗号装置。
【請求項６】
暗号化関数が、入力される第1の変数と第2の変数とを連結したものを入力とする、鍵付きハッシュ関数である
請求項１から請求項４のうちのいずれか１項に記載の認証暗号装置。
【請求項７】
入力された平文に対して、2ブロックごとに、補助変数を入れた暗号化関数をラウンド関数に用いた2ラウンドFeistel構造を適用して暗号文を生成し、入力された前記平文のうちの一部のビットを用いて算出されるチェックサムに対して前記暗号化関数を適用して認証タグを生成する暗号化手段を備え、
前記暗号化手段は、
暗号化対象の平文と初期ベクトルとを入力する平文入力手段と、
前記初期ベクトルと入力された平文のサイズとに基づき、前記暗号化関数の各々に与える補助変数を生成する補助変数生成手段と、

前記平文を2ブロックごとのチャンクに分けたときの各平文チャンクに対して2ラウンドFeistel構造を適用することにより、当該平文チャンクに対応する暗号文チャンクを生成する2ラウンドFeistel暗号化手段と、
前記平文のチェックサムを計算し、得られたチェックサムに対して、補助変数を入れた暗号化関数を適用させて認証タグを生成するタグ計算手段とを有し、
前記2ラウンドFeistel暗号化手段は、初期ベクトルをN、チャンクのインデックスをi、i番目の平文チャンクをMC[i] = (M[i_1], M[i_2])、当該平文チャンクMC[i]に含まれる2つの平文ブロックに対応する補助変数を(N,Tw_i_1)と(N,Tw_i_2)の組、暗号化関数をF_K(*,*)とすると、i番目の暗号文チャンクCC[i] = (C[i_1], C[i_2])を、
C[i_1] = F_K((N,Tw_i_1), M[i_1]) xor M[i_2] ,
C[i_2] = F_K((N,Tw_i_2), C[i_1]) xor M[i_1]
と求め、
前記タグ計算手段は、平文のチェックサムを、各平文チャンクに含まれる平文ブロックM[i_2]を用いて計算し、得られたチェックサムをSUM、認証タグ生成時に用いる暗号化関数に与える補助変数を(N,Tw_T_1)、暗号化関数をF_K(*,*)とすると、認証タグTを、
T = F_K((N,Tw_T_1), SUM)
と求め、
前記補助変数生成手段は、入力された平文のサイズが偶数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズに満たない場合に、認証タグ生成時に用いる暗号化関数に与える補助変数として、第2の認証タグ用補助変数を生成し、
前記暗号化手段は、
入力された平文のサイズが偶数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズに満たない場合に、最終の平文ブロックを含む最終の平文チャンクに対して、所定の2ラウンドFeistel構造を適用して、最終の暗号文ブロックを含む最終の暗号文チャンクを生成する第2の2ラウンドFeistel暗号化手段と、
入力された平文のサイズが偶数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズに満たない場合に、平文のチェックサムを、入力された平文と、前記第2の2ラウンドFeistel暗号化手段からの出力とを用いて計算し、得られたチェックサムに対して補助変数を与えた暗号化関数を適用させて、認証タグを生成する第2のタグ計算手段とを有し、
前記第2の2ラウンドFeistel暗号化手段は、最終の平文チャンクのインデックスをm、最終の平文ブロックをM[m_2]、最終の平文チャンクをMC[m] = (M[m_1], M[m_2])、最終の平文チャンクMC[m]に含まれる2つの平文ブロックに対応する補助変数を(N,Tw_m_1)と(N,Tw_m_2)の組、暗号化関数をF_K(*,*)、最終の平文ブロックのサイズをs、ブロックサイズをn、sビットからnビットへのパディング処理をpad_n()、nビットからsビットへのカッティング処理をcut_s()とすると、sビットの最終の暗号文ブロックC[m_2]を含む最終の暗号文チャンクCC[m] = (C[m_1], C[m_2])を、
C[m_2] = cut_s(Z) xor M[m_2],
C[m_1] = F_K((N,Tw_m_2), pad_n(C[m_2])) xor M[m_1]
ただし、Z = F_K((N,Tw_m_1), M[m_1])
と求め、
前記第2のタグ計算手段は、平文のチェックサムを、前記最終の平文チャンクを除く各平文チャンクに含まれる平文ブロックM[i_2]と、前記Zと、前記C[m_2]をnビットにパディングした結果であるC_n[m_2]と、を用いて計算し、得られたチェックサムをSUM、前記第2の認証タグ用補助変数を(N,Tw_T_2)、暗号化関数をF_K(*,*)とすると、認証タグTを、
T = F_K((N,Tw_T_2), SUM)
と求める
ことを特徴とする暗号化装置。
【請求項８】
入力された暗号文に対して、2ブロックごとに、補助変数を入れた暗号化関数をラウン

ド関数に用いた2ラウンドFeistel構造を適用して復号された平文を生成し、復号された前記平文のうちの一部のビットを用いて算出されるチェックサムに対して前記暗号化関数を適用して認証タグを生成する復号手段を備え、

前記復号手段は、

復号対象の暗号文と初期ベクトルと認証タグとを入力する暗号文入力手段と、

前記初期ベクトルと入力された暗号文のサイズとに基づき、前記暗号化関数の各々に与える補助変数であって暗号化時と同じ補助変数を生成する復号用補助変数生成手段と、

前記暗号文を2ブロックごとのチャンクに分けたときの各暗号文チャンクに対して2ラウンドFeistel構造を適用することにより、当該暗号文チャンクに対応する、復号された平文チャンクを生成する2ラウンドFeistel復号手段と、

前記復号された平文のチェックサムを計算し、得られたチェックサムに対して、補助変数を入れた暗号化関数を適用させて復号検証用の認証タグを生成する復号検証用タグ計算手段と、

前記復号検証用タグ計算手段が生成した復号検証用の認証タグと入力された認証タグとに基づいて、復号の成功または失敗を判定する判定手段とを有し、

前記2ラウンドFeistel復号手段は、初期ベクトルをN、チャンクのインデックスをi、i番目の暗号文チャンクをCC[i] = (C[i_1], C[i_2])、当該暗号文チャンクCC[i]に含まれる2つの暗号文ブロックに対応する補助変数を(N,Tw_i_1)と(N,Tw_i_2)の組、暗号化関数をF_K(*,*)とすると、i番目の復号された平文チャンクMC'[i] = (M'[i_1], M'[i_2])を、

M'[i_1] = F_K((N,Tw_i_2), C[i_1]) xor C[i_2],

M'[i_2] = F_K((N,Tw_i_1), M'[i_1]) xor C[i_1]

と求め、

前記復号検証用タグ計算手段は、復号された平文のチェックサムを、復号された各平文チャンクに含まれる復号された各平文ブロックM'[i_2]を用いて計算し、得られたチェックサムをSUM'、復号検証用の認証タグ生成時に用いる暗号化関数に与える補助変数を(N,Tw_T_1)、暗号化関数をF_K(*,*)とすると、復号検証用の認証タグT'を、

T' = F_K((N,Tw_T_1), SUM')

と求め、

前記復号用補助変数生成手段は、入力された暗号文のサイズが偶数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズに満たない場合に、復号検証用の認証タグ生成時に用いる暗号化関数に与える補助変数として、暗号時と同じ第2の認証タグ用補助変数を生成し、

前記復号手段は、

入力された暗号文のサイズが偶数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズに満たない場合に、最終の暗号文ブロックを含む最終の暗号文チャンクに対して、所定の2ラウンドFeistel構造を適用して、最終の暗号ブロックの復号された平文ブロックを含む最終の復号された平文チャンクを生成する第2の2ラウンドFeistel復号手段と、

入力された暗号文のサイズが偶数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズに満たない場合に、復号された平文のチェックサムを、前記2ラウンドFeistel復号手段からの出力と、前記第2の2ラウンドFeistel復号手段からの出力と、前記最終の暗号文ブロックとを用いて計算し、得られたチェックサムに対して補助変数を与えた暗号化関数を適用させて、復号検証用の認証タグを生成する第2の復号検証用タグ計算手段とを有し、

前記第2の2ラウンドFeistel復号手段は、最終の暗号文チャンクのインデックスをm、最終の暗号文ブロックをC[m_2]、最終の暗号文チャンクをCC[m] = (C[m_1], C[m_2])、最終の暗号文チャンクCC[m]に含まれる2つの暗号文ブロックに対応する補助変数を(N,Tw_m_1)と(N,Tw_m_2)の組、暗号化関数をF_K(*,*)、最終の暗号文ブロックのサイズをs、ブロックサイズをn、sビットからnビットへのパディング処理をpad_n()、nビットからsビットへのカッティング処理をcut_s()とすると、sビットの最終の復号された平文ブロックM'[m_2

]を含む最終の復号された平文チャンクMC'[m] = (M'[m_1], M'[m_2])を、
M'[m_1] = F_K((N,Tw_m_2), pad_n(C[m_2])) xor C[m_1],
M'[m_2] = cut_s(Z') xor C[m_2],
ただし、Z' = F_K((N,Tw_m_1), M'[m_1])
と求め、
前記第2の復号検証用タグ計算手段は、復号された平文のチェックサムを、前記最終の復号された平文チャンクを除く復号された各平文チャンクに含まれる復号された平文ブロックM'[i_2]と、前記Z'と、前記C[m_2]をnビットにパディングした結果であるC_n[m_2]と、を用いて計算し、得られたチェックサムをSUM'、前記第2の認証タグ用補助変数を(N,Tw_T_2)、暗号化関数をF_K(*,*)とすると、復号検証用の認証タグT'を、
T' = F_K((N,Tw_T_2), SUM')
と求める
ことを特徴とする復号装置。

【請求項９】

情報処理装置が、入力された平文または暗号文に対して、2ブロックごとに、補助変数を入れた暗号化関数をラウンド関数に用いた2ラウンドFeistel構造を適用して暗号文または復号された平文を生成し、入力された前記平文または復号された前記平文のうちの一部のビットを用いて算出されるチェックサムに対して前記暗号化関数を適用して認証タグを生成し、
前記情報処理装置が、前記認証タグを生成する処理で、暗号化処理を実行し、
前記暗号化処理で、
暗号化対象の平文と初期ベクトルとを入力する平文入力処理と、
前記初期ベクトルと入力された平文のサイズとに基づき、前記暗号化関数の各々に与える補助変数を生成する補助変数生成処理と、
前記平文を2ブロックごとのチャンクに分けたときの各平文チャンクに対して2ラウンドFeistel構造を適用することにより、当該平文チャンクに対応する暗号文チャンクを生成する2ラウンドFeistel暗号化処理と、
前記平文のチェックサムを計算し、得られたチェックサムに対して、補助変数を入れた暗号化関数を適用させて認証タグを生成するタグ計算処理とを実行し、
前記2ラウンドFeistel暗号化処理で、初期ベクトルをN、チャンクのインデックスをi、i番目の平文チャンクをMC[i] = (M[i_1], M[i_2])、当該平文チャンクMC[i]に含まれる2つの平文ブロックに対応する補助変数を(N,Tw_i_1)と(N,Tw_i_2)の組、暗号化関数をF_K(*,*)とすると、i番目の暗号文チャンクCC[i] = (C[i_1], C[i_2])を、
C[i_1] = F_K((N,Tw_i_1), M[i_1]) xor M[i_2] ,
C[i_2] = F_K((N,Tw_i_2), C[i_1]) xor M[i_1]
と求め、
前記タグ計算処理で、平文のチェックサムを、各平文チャンクに含まれる平文ブロックM[i_2]を用いて計算し、得られたチェックサムをSUM、認証タグ生成時に用いる暗号化関数に与える補助変数を(N,Tw_T_1)、暗号化関数をF_K(*,*)とすると、認証タグTを、
T = F_K((N,Tw_T_1), SUM)
と求め、
前記補助変数生成処理で、入力された平文のサイズが偶数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズに満たない場合に、認証タグ生成時に用いる暗号化関数に与える補助変数として、第2の認証タグ用補助変数を生成し、
前記暗号化処理で、
入力された平文のサイズが偶数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズに満たない場合に、最終の平文ブロックを含む最終の平文チャンクに対して、所定の2ラウンドFeistel構造を適用して、最終の暗号文ブロックを含む最終の暗号文チャンクを生成する第2の2ラウンドFeistel暗号化処理と、
入力された平文のサイズが偶数のブロックに分割されるサイズであり、かつ最終のブロ

ックが所定のブロックサイズに満たない場合に、平文のチェックサムを、入力された平文と、前記第2の2ラウンドFeistel暗号化処理の出力とを用いて計算し、得られたチェックサムに対して補助変数を与えた暗号化関数を適用させて、認証タグを生成する第2のタグ計算処理とを実行し、

前記第2の2ラウンドFeistel暗号化処理で、最終の平文チャンクのインデックスをm、最終の平文ブロックをM[m_2]、最終の平文チャンクをMC[m] = (M[m_1], M[m_2])、最終の平文チャンクMC[m]に含まれる2つの平文ブロックに対応する補助変数を(N,Tw_m_1)と(N,Tw_m_2)の組、暗号化関数をF_K(*,*)、最終の平文ブロックのサイズをs、ブロックサイズをn、sビットからnビットへのパディング処理をpad_n()、nビットからsビットへのカッティング処理をcut_s()とすると、sビットの最終の暗号文ブロックC[m_2]を含む最終の暗号文チャンクCC[m] = (C[m_1], C[m_2])を、

C[m_2] = cut_s(Z) xor M[m_2],

C[m_1] = F_K((N,Tw_m_2), pad_n(C[m_2])) xor M[m_1]

ただし、Z = F_K((N,Tw_m_1), M[m_1])

と求め、

前記第2のタグ計算処理で、平文のチェックサムを、前記最終の平文チャンクを除く各平文チャンクに含まれる平文ブロックM[i_2]と、前記Zと、前記C[m_2]をnビットにパディングした結果であるC_n[m_2]と、を用いて計算し、得られたチェックサムをSUM、前記第2の認証タグ用補助変数を(N,Tw_T_2)、暗号化関数をF_K(*,*)とすると、認証タグTを、

T = F_K((N,Tw_T_2), SUM)

と求める

ことを特徴とする認証暗号方法。

【請求項１０】

情報処理装置が、入力された平文または暗号文に対して、2ブロックごとに、補助変数を入れた暗号化関数をラウンド関数に用いた2ラウンドFeistel構造を適用して暗号文または復号された平文を生成し、入力された前記平文または復号された前記平文のうちの一部のビットを用いて算出されるチェックサムに対して前記暗号化関数を適用して認証タグを生成し、

前記情報処理装置が、前記認証タグを生成する処理で、復号処理を実行し、

前記復号処理で、

復号対象の暗号文と初期ベクトルと認証タグとを入力する暗号文入力処理と、

前記初期ベクトルと入力された暗号文のサイズとに基づき、前記暗号化関数の各々に与える補助変数であって暗号化時と同じ補助変数を生成する復号用補助変数生成処理と、

前記暗号文を2ブロックごとのチャンクに分けたときの各暗号文チャンクに対して2ラウンドFeistel構造を適用することにより、当該暗号文チャンクに対応する、復号された平文チャンクを生成する2ラウンドFeistel復号処理と、

前記復号された平文のチェックサムを計算し、得られたチェックサムに対して、補助変数を入れた暗号化関数を適用させて復号検証用の認証タグを生成する復号検証用タグ計算処理と、

前記復号検証用タグ計算処理で生成した復号検証用の認証タグと入力された認証タグとに基づいて、復号の成功または失敗を判定する判定処理とを実行し、

前記2ラウンドFeistel復号処理で、初期ベクトルをN、チャンクのインデックスをi、i番目の暗号文チャンクをCC[i] = (C[i_1], C[i_2])、当該暗号文チャンクCC[i]に含まれる2つの暗号文ブロックに対応する補助変数を(N,Tw_i_1)と(N,Tw_i_2)の組、暗号化関数をF_K(*,*)とすると、i番目の復号された平文チャンクMC'[i] = (M'[i_1], M'[i_2])を、

M'[i_1] = F_K((N,Tw_i_2), C[i_1]) xor C[i_2],

M'[i_2] = F_K((N,Tw_i_1), M'[i_1]) xor C[i_1]

と求め、

前記復号検証用タグ計算処理で、復号された平文のチェックサムを、復号された各平文チャンクに含まれる復号された各平文ブロックM'[i_2]を用いて計算し、得られたチェッ

クサムをSUM'、復号検証用の認証タグ生成時に用いる暗号化関数に与える補助変数を(N,Tw_T_1)、暗号化関数をF_K(*,*)とすると、復号検証用の認証タグT'を、
T' = F_K((N,Tw_T_1), SUM')
と求め、
前記復号用補助変数生成処理で、入力された暗号文のサイズが偶数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズに満たない場合に、復号検証用の認証タグ生成時に用いる暗号化関数に与える補助変数として、暗号時と同じ第2の認証タグ用補助変数を生成し、
前記復号処理で、
入力された暗号文のサイズが偶数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズに満たない場合に、最終の暗号文ブロックを含む最終の暗号文チャンクに対して、所定の2ラウンドFeistel構造を適用して、最終の暗号ブロックの復号された平文ブロックを含む最終の復号された平文チャンクを生成する第2の2ラウンドFeistel復号処理と、
入力された暗号文のサイズが偶数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズに満たない場合に、復号された平文のチェックサムを、前記2ラウンドFeistel復号処理の出力と、前記第2の2ラウンドFeistel復号処理の出力と、前記最終の暗号文ブロックとを用いて計算し、得られたチェックサムに対して補助変数を与えた暗号化関数を適用させて、復号検証用の認証タグを生成する第2の復号検証用タグ計算処理とを実行し、
前記第2の2ラウンドFeistel復号処理で、最終の暗号文チャンクのインデックスをm、最終の暗号文ブロックをC[m_2]、最終の暗号文チャンクをCC[m] = (C[m_1], C[m_2])、最終の暗号文チャンクCC[m]に含まれる2つの暗号文ブロックに対応する補助変数を(N,Tw_m_1)と(N,Tw_m_2)の組、暗号化関数をF_K(*,*)、最終の暗号文ブロックのサイズをs、ブロックサイズをn、sビットからnビットへのパディング処理をpad_n()、nビットからsビットへのカッティング処理をcut_s()とすると、sビットの最終の復号された平文ブロックM'[m_2]を含む最終の復号された平文チャンクMC'[m] = (M'[m_1], M'[m_2])を、
M'[m_1] = F_K((N,Tw_m_2), pad_n(C[m_2])) xor C[m_1],
M'[m_2] = cut_s(Z') xor C[m_2],
ただし、Z' = F_K((N,Tw_m_1), M'[m_1])
と求め、
前記第2の復号検証用タグ計算処理で、復号された平文のチェックサムを、前記最終の復号された平文チャンクを除く復号された各平文チャンクに含まれる復号された平文ブロックM'[i_2]と、前記Z'と、前記C[m_2]をnビットにパディングした結果であるC_n[m_2]と、を用いて計算し、得られたチェックサムをSUM'、前記第2の認証タグ用補助変数を(N,Tw_T_2)、暗号化関数をF_K(*,*)とすると、復号検証用の認証タグT'を、
T' = F_K((N,Tw_T_2), SUM')
と求める
ことを特徴とする認証暗号方法。

【請求項１１】
コンピュータに、
入力された平文または暗号文に対して、2ブロックごとに、補助変数を入れた暗号化関数をラウンド関数に用いた2ラウンドFeistel構造を適用して暗号文または復号された平文を生成し、入力された前記平文または復号された前記平文のうちの一部のビットを用いて算出されるチェックサムに対して前記暗号化関数を適用して認証タグを生成する処理
を実行させ、
前記処理で、暗号化処理を実行させ、
前記暗号化処理で、
暗号化対象の平文と初期ベクトルとを入力する平文入力処理と、
前記初期ベクトルと入力された平文のサイズとに基づき、前記暗号化関数の各々に与え

る補助変数を生成する補助変数生成処理と、
前記平文を2ブロックごとのチャンクに分けたときの各平文チャンクに対して2ラウンドFeistel構造を適用することにより、当該平文チャンクに対応する暗号文チャンクを生成する2ラウンドFeistel暗号化処理と、
前記平文のチェックサムを計算し、得られたチェックサムに対して、補助変数を入れた暗号化関数を適用させて認証タグを生成するタグ計算処理とを実行させ、
前記2ラウンドFeistel暗号化処理で、初期ベクトルをN、チャンクのインデックスをi、i番目の平文チャンクをMC[i] = (M[i_1], M[i_2])、当該平文チャンクMC[i]に含まれる2つの平文ブロックに対応する補助変数を(N,Tw_i_1)と(N,Tw_i_2)の組、暗号化関数をF_K(*,*)とすると、i番目の暗号文チャンクCC[i] = (C[i_1], C[i_2])を、
C[i_1] = F_K((N,Tw_i_1), M[i_1]) xor M[i_2] ,
C[i_2] = F_K((N,Tw_i_2), C[i_1]) xor M[i_1]
と求めさせ、
前記タグ計算処理で、平文のチェックサムを、各平文チャンクに含まれる平文ブロックM[i_2]を用いて計算させ、得られたチェックサムをSUM、認証タグ生成時に用いる暗号化関数に与える補助変数を(N,Tw_T_1)、暗号化関数をF_K(*,*)とすると、認証タグTを、
T = F_K((N,Tw_T_1), SUM)
と求めさせ、
前記補助変数生成処理で、入力された平文のサイズが偶数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズに満たない場合に、認証タグ生成時に用いる暗号化関数に与える補助変数として、第2の認証タグ用補助変数を生成させ、
前記暗号化処理で、
入力された平文のサイズが偶数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズに満たない場合に、最終の平文ブロックを含む最終の平文チャンクに対して、所定の2ラウンドFeistel構造を適用して、最終の暗号文ブロックを含む最終の暗号文チャンクを生成する第2の2ラウンドFeistel暗号化処理と、
入力された平文のサイズが偶数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズに満たない場合に、平文のチェックサムを、入力された平文と、前記第2の2ラウンドFeistel暗号化処理の出力とを用いて計算し、得られたチェックサムに対して補助変数を与えた暗号化関数を適用させて、認証タグを生成する第2のタグ計算処理とを実行させ、
前記第2の2ラウンドFeistel暗号化処理で、最終の平文チャンクのインデックスをm、最終の平文ブロックをM[m_2]、最終の平文チャンクをMC[m] = (M[m_1], M[m_2])、最終の平文チャンクMC[m]に含まれる2つの平文ブロックに対応する補助変数を(N,Tw_m_1)と(N,Tw_m_2)の組、暗号化関数をF_K(*,*)、最終の平文ブロックのサイズをs、ブロックサイズをn、sビットからnビットへのパディング処理をpad_n()、nビットからsビットへのカッティング処理をcut_s()とすると、sビットの最終の暗号文ブロックC[m_2]を含む最終の暗号文チャンクCC[m] = (C[m_1], C[m_2])を、
C[m_2] = cut_s(Z) xor M[m_2],
C[m_1] = F_K((N,Tw_m_2), pad_n(C[m_2])) xor M[m_1]
ただし、Z = F_K((N,Tw_m_1), M[m_1])
と求めさせ、
前記第2のタグ計算処理で、平文のチェックサムを、前記最終の平文チャンクを除く各平文チャンクに含まれる平文ブロックM[i_2]と、前記Zと、前記C[m_2]をnビットにパディングした結果であるC_n[m_2]と、を用いて計算させ、得られたチェックサムをSUM、前記第2の認証タグ用補助変数を(N,Tw_T_2)、暗号化関数をF_K(*,*)とすると、認証タグTを、
T = F_K((N,Tw_T_2), SUM)
と求めさせる
ための認証暗号用プログラム。
【請求項１２】

コンピュータに、
入力された平文または暗号文に対して、2ブロックごとに、補助変数を入れた暗号化関数をラウンド関数に用いた2ラウンドFeistel構造を適用して暗号文または復号された平文を生成し、入力された前記平文または復号された前記平文のうちの一部のビットを用いて算出されるチェックサムに対して前記暗号化関数を適用して認証タグを生成する処理
を実行させ、
前記処理で、復号処理を実行させ、
前記復号処理で、
復号対象の暗号文と初期ベクトルと認証タグとを入力する暗号文入力処理と、
前記初期ベクトルと入力された暗号文のサイズとに基づき、前記暗号化関数の各々に与える補助変数であって暗号化時と同じ補助変数を生成する復号用補助変数生成処理と、
前記暗号文を2ブロックごとのチャンクに分けたときの各暗号文チャンクに対して2ラウンドFeistel構造を適用することにより、当該暗号文チャンクに対応する、復号された平文チャンクを生成する2ラウンドFeistel復号処理と、
前記復号された平文のチェックサムを計算し、得られたチェックサムに対して、補助変数を入れた暗号化関数を適用させて復号検証用の認証タグを生成する復号検証用タグ計算処理と、
前記復号検証用タグ計算処理で生成した復号検証用の認証タグと入力された認証タグとに基づいて、復号の成功または失敗を判定する判定処理とを実行させ、
前記2ラウンドFeistel復号処理で、初期ベクトルをN、チャンクのインデックスをi、i番目の暗号文チャンクをCC[i] = (C[i_1], C[i_2])、当該暗号文チャンクCC[i]に含まれる2つの暗号文ブロックに対応する補助変数を(N,Tw_i_1)と(N,Tw_i_2)の組、暗号化関数をF_K(*,*)とすると、i番目の復号された平文チャンクMC'[i] = (M'[i_1], M'[i_2])を、
M'[i_1] = F_K((N,Tw_i_2), C[i_1]) xor C[i_2],
M'[i_2] = F_K((N,Tw_i_1), M'[i_1]) xor C[i_1]
と求めさせ、
前記復号検証用タグ計算処理で、復号された平文のチェックサムを、復号された各平文チャンクに含まれる復号された各平文ブロックM'[i_2]を用いて計算させ、得られたチェックサムをSUM'、復号検証用の認証タグ生成時に用いる暗号化関数に与える補助変数を(N,Tw_T_1)、暗号化関数をF_K(*,*)とすると、復号検証用の認証タグT'を、
T' = F_K((N,Tw_T_1), SUM')
と求めさせ、
前記復号用補助変数生成処理で、入力された暗号文のサイズが偶数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズに満たない場合に、復号検証用の認証タグ生成時に用いる暗号化関数に与える補助変数として、暗号時と同じ第2の認証タグ用補助変数を生成させ、
前記復号処理で、
入力された暗号文のサイズが偶数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズに満たない場合に、最終の暗号文ブロックを含む最終の暗号文チャンクに対して、所定の2ラウンドFeistel構造を適用して、最終の暗号ブロックの復号された平文ブロックを含む最終の復号された平文チャンクを生成する第2の2ラウンドFeistel復号処理と、
入力された暗号文のサイズが偶数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズに満たない場合に、復号された平文のチェックサムを、前記2ラウンドFeistel復号処理の出力と、前記第2の2ラウンドFeistel復号処理の出力と、前記最終の暗号文ブロックとを用いて計算し、得られたチェックサムに対して補助変数を与えた暗号化関数を適用させて、復号検証用の認証タグを生成する第2の復号検証用タグ計算処理とを実行させ、
前記第2の2ラウンドFeistel復号処理で、最終の暗号文チャンクのインデックスをm、最終の暗号文ブロックをC[m_2]、最終の暗号文チャンクをCC[m] = (C[m_1], C[m_2])、最終

の暗号文チャンクCC[m]に含まれる2つの暗号文ブロックに対応する補助変数を(N,Tw_m_1)と(N,Tw_m_2)の組、暗号化関数をF_K(*,*)、最終の暗号文ブロックのサイズをs、ブロックサイズをn、sビットからnビットへのパディング処理をpad_n()、nビットからsビットへのカッティング処理をcut_s()とすると、sビットの最終の復号された平文ブロックM'[m_2]を含む最終の復号された平文チャンクMC'[m] = (M'[m_1], M'[m_2])を、
M'[m_1] = F_K((N,Tw_m_2), pad_n(C[m_2])) xor C[m_1],
M'[m_2] = cut_s(Z') xor C[m_2],
ただし、Z' = F_K((N,Tw_m_1), M'[m_1])
と求めさせ、
前記第2の復号検証用タグ計算処理で、復号された平文のチェックサムを、前記最終の復号された平文チャンクを除く復号された各平文チャンクに含まれる復号された平文ブロックM'[i_2]と、前記Z'と、前記C[m_2]をnビットにパディングした結果であるC_n[m_2]と、を用いて計算させ、得られたチェックサムをSUM'、前記第2の認証タグ用補助変数を(N,Tw_T_2)、暗号化関数をF_K(*,*)とすると、復号検証用の認証タグT'を、
T' = F_K((N,Tw_T_2), SUM')
と求めさせる
ための認証暗号用プログラム。

【発明の詳細な説明】

【技術分野】

【０００１】

本発明は、共通鍵を用いて認証暗号を行う認証暗号装置、暗号化装置、復号装置、認証暗号方法および認証暗号用プログラムに関する。

【背景技術】

【０００２】

認証暗号（Authenticated Encryption, AE）とは、事前に共有された秘密鍵を用いて、平文メッセージに対して暗号化とメッセージ認証コード（Message authentication code, MAC）の付与とを同時に適用する技術である。認証暗号を適用することにより、盗聴に対する内容の秘匿と、不正な改ざんに対する検知が可能となる。通信路に認証暗号を適用すれば、通信内容に対する強力な保護が実現される。

【０００３】

以下に、認証暗号の基本的な入出力を示す。なお、以下では秘密鍵Ｋを共有する２者としてAliceとBobを考え、AliceからBobへ認証暗号による暗号化を行ったメッセージを通信するものとする。

【０００４】

認証暗号の暗号化関数をAEnc_K、復号関数をADec_Kとする。暗号化したい平文をMとし、さらに初期ベクトルと呼ばれる変数Nを導入する。初期ベクトルNはAliceにより生成され、通常は短い固定長の乱数やカウンターなどである。

【０００５】

まずAlice側の暗号化処理について説明する。Aliceは初期ベクトルNを生成後、(C,T)=AEnc_K(N,M)を実行する。ここで、AEnc_Kは鍵Kをパラメータとした暗号化関数、C は暗号文、Tは認証タグと呼ばれる、固定長の改ざん検出用の変数である。Aliceは初期ベクトルNと得られた暗号文C と得られた認証タグTの組(N,C,T)をBobに送信する。

【０００６】

次に、Bob側の復号処理について説明する。ここでは、BobがAliceから受信した情報を仮に(N',C',T')とする。BobはAliceから情報を受信すると、復号処理としてADec_K(N',C',T')を実行する。ADec_Kは鍵Kをパラメータとした復号関数である。もし通信の途中に改ざんがあり、(N',C',T')が(N,C,T)と異なっていた場合、ADec_K(N',C',T')は改ざんがあったことを示すシンボル、以下仮に\botとすると、\botを出力する。もし改ざんがなく、(N',C',T')=(N,C,T)であれば、ADec_K(N',C',T')はAliceが暗号化した平文Mと同じ内容の復号された平文M'を出力する。これにより、Mが正しく復号される。

【０００７】
なお、実用的には上記の入出力に、ヘッダHと呼ばれる変数を含ませることが多い。ヘッダHは、暗号化の対象にはしないものの、メッセージ認証の対象となる情報であり、例えばプロトコルのバージョンなどを表すのに用いられる。
【０００８】
ヘッダHを含めた場合、暗号化関数は(C,T)=AEnc_K(N,M,H)という入出力となり、平文Mに対して暗号化するとともに平文Mとヘッダの組に対してメッセージ認証コードの付与を行う。また、Aliceは初期ベクトルN とヘッダHと得られた暗号文Cと得られた認証タグTの組(N,H,C,T)をBobに送信する。
【０００９】
復号関数はADec_K(N',C',T',H')という入出力となり、(N',C',T',H')が(N,C,T,H)と異なっていた場合、改ざんを示すシンボル\botを出力する。受信側のBobは受信した(N',H',C',T')に改ざんがなければ、すなわち(N',H',C',T')=(N,H,C,T)であれば、正しくMを復号でき、さらにヘッダHに改ざんがないことを確認できる。
【００１０】
このような入出力にヘッダHを加えた認証暗号を、ヘッダ付き認証暗号（Authenticated Encryption with Associated Data ,AEAD）と呼ぶこともあるが、以後は特に言及しないかぎり、区別をせず単に「認証暗号」と表記する。
【００１１】
認証暗号の実現方法の一つに、汎用的結合（generic composition）に基づくものがある。これは、安全な暗号化方式と安全なMAC方式とを組み合わせて用いる方法である。例えば、一般的に知られるEnc-then-Authというタイプの組み合わせの場合、二つの鍵K1とK2を用いて、(C,T)=MAC_K2(N,Enc_K1(M))として認証暗号を実現する。ここで、Enc_XXが暗号化方式で用いる暗号化関数、MAC_XXがMAC方式で用いるMACの付与関数を示す。
【００１２】
AES(Advanced Encryption Standard)暗号などのブロック暗号を用いる場合、例えば暗号化方式はAESのカウンターモード暗号化を用い、MAC方式はCMAC-AES(Cipher-based MAC-AES)を用いることが可能である。さらに、二つの鍵を用いず、ブロック暗号の鍵一つで認証暗号を行う方法として、CCMモード(Counter with CBC-MAC)と呼ばれている認証暗号方式が知られている（例えば、非特許文献1）。
【００１３】
しかし、上記方法のいずれも、暗号化とMACの付与とで2パスの処理を必要とする。すなわち、データ全体を少なくとも2回走査する必要がある。さらに、CCM方式で用いる暗号化とMACの関数は、入力された平文がmブロックの場合、ブロック暗号を約m回コールする必要がある。このため、mブロック平文に対する認証暗号としての処理には約2m回のブロック暗号コールを必要とする。すなわち、平文ブロックにつき、暗号化関数やMACの付与関数といった処理関数を2回コールしなければならない。このような、各ブロックにつき処理関数を2回コールする方式は2レート方式とも呼ばれる。2パスまたは2レート方式の場合、処理に時間がかかる、負荷が大きいといった問題があった。
【００１４】
このような問題を解決するアプローチとして、ブロック暗号を用いた1パスの認証暗号方式がある。
【００１５】
まず、特許文献1に記載されている、OCBモードと呼ばれる認証暗号方式である（以下、OCB方式という。）。OCB方式は、非特許文献2に記載されている、Tweakableブロック暗号と呼ばれるブロック暗号を拡張したものである。
【００１６】
Tweakableブロック暗号は、暗号化と復号の際に、Tweakと呼ばれる補助変数を導入した認証暗号方式である。ブロックサイズがn-bitのとき、Tweakableブロック暗号による暗号化は「TE_K(Tw,M) = C」と表すことができ、復号は「TD_K(Tw,C) = M」と表すことができ

る。任意の(K,Tw)の組について、TE_K(Tw,*)はn-bit空間上の置換を構成する。なお、その逆置換がTD_K(Tw,*)である。なお、Twは補助変数Tweakを表し、*は任意の変数を表す。補助変数Twは復号に必要な変数であるが、公開したとしてもTweakableブロック暗号の安全性には影響しない。

【００１７】

OCB方式ではまず、通常のブロック暗号の暗号化関数を非特許文献3に記載されているXEXモードを用いてTweakableブロック暗号の暗号化関数すなわちTweakを入れた暗号化関数に変換する。次いで、初期ベクトルN、平文M = (M[1], M[2], ... , M[m])について、次に示すようなTE_K関数を呼ぶことで暗号化を行う。なお、各M[i]はnビットのブロックとする。OCB方式で用いるTE_K関数では、Tweakに相当する変数として(N,i)を用いている。なお、Nは初期ベクトル、iはブロックの識別番号である。

【００１８】

```
C[1] = TE_K((N,1), M[1]),
C[2] = TE_K((N,2), M[2]),
 ... ,
C[m] = TE_K((N,m), M[m])
```

【００１９】

認証タグTは、全平文ブロックのXOR（排他的論理和）であるSUM = M[1] xor M[2] xor .... xor M[m]について、メッセージと同じTE_K関数を例えば次のように呼ぶことで求められる。

【００２０】

```
T = TE_K((N,m+1), SUM)
```

【００２１】

OCB方式で用いるTE_K関数は、Tw=(N,i)と、秘密鍵Kから計算される系列mask_K(N,i)とをブロック暗号の暗号化関数E_Kの入出力に加算することで行われる（XEXモードによる変換）。XEXモードによる変換式は、次のように表される。以下、この変換式をXEX変換式と呼ぶ場合がある。

【００２２】

```
TE_K((N,i), M[i]) = E_K(M[i] xor mask_K(N,i)) xor mask_K(N,i)
```

【００２３】

maskの計算にはE_Kを用いるが、OCB方式は効率的な逐次処理が可能である。すなわちmask_K(N,i)からmask_K(N,i+1)の計算を効率的に行うことが可能である。

【００２４】

図19は、OCB方式における暗号化処理を模式的に示す説明図である。図19において、破線で示すブロックがTE_K((N,i), *)に相当する。なお、図19では、mask系列を計算する処理の過程は省略している。

【００２５】

図19に示すように、OCB方式ではmask系列を求めるためのLと認証タグTの計算のために各々1回E_Kが呼ばれているが、全体として1パス処理が可能である。また、mask系列の計算を除けば各ブロックの処理は並列に行うことが可能である。より具体的には、mブロックの平文に対するブロック暗号のコール数はほぼmであり、上述のCCM方式やGCM(Galois/Counter Mode)方式といった他の2パスの認証暗号方式に比べ、約半分の処理量となっている。

【先行技術文献】

【特許文献】

【００２６】

【特許文献１】米国特許第8321675号明細書

【非特許文献】

【００２７】

【非特許文献１】Morris Dworkin, "Recommendation for Block Cipher Modes of Operat

ion: The CCM Mode for Authentication and Confidentiality.", [online] May 2005, NIST Special Publication 800-38C, インターネット<URL: http://csrc.nist.gov/publications/nistpubs/800-38C/SP800-38C_updated-July20_2007.pdf>
【非特許文献２】Moses Liskov, Ronald L. Rivest, David Wagner, "Tweakable Block Ciphers. ", 2002, Advances in Cryptology - CRYPTO 2002, Lecture Notes in Computer Science 2442 Springer 2002, p. 31-46.
【非特許文献３】P. Rogaway, "Efficient Instantiations of Tweakable Blockciphers and Refinements to Modes OCB and PMAC.", 2004, Advances in Cryptology- ASIACRYPT '04, LNCS 3329, p. 16-31.

【発明の概要】

【発明が解決しようとする課題】

【００２８】

図20は、OCB方式における復号処理を模式的に示す説明図である。CCMモードやGCMが、ブロック暗号の暗号化関数E_Kのみで認証暗号としての復号処理を実現するのに対し、OCB方式では、図20に示すように、Tweakableブロック暗号の復号処理にブロック暗号の暗号化関数E_Kだけでなくブロック暗号の復号関数D_Kも必要とする。

【００２９】

認証暗号の暗号化と復号において、部品となるブロック暗号の暗号化関数と復号関数の両方を必要とする場合、メモリや回路規模などの点で実装上の負担が増える。また、代表的なブロック暗号であるＡＥＳは、一般的に暗号化と比べて復号の処理が遅いことが知られており、このことはＡＥＳを用いたＯＣＢ方式などにおいて、暗号化処理と復号処理に性能の差が生じることを意味する。

【００３０】

ＯＣＢとそれに類する１パス認証暗号方式では、上記のような課題を解決することはできない。

【００３１】

そこで、本発明は、１パスおよび１レート方式の認証暗号であって、並列処理が可能で、かつ１つの暗号化関数のみで全体の暗号化と復号の処理を実行可能な認証暗号を実現する認証暗号装置、暗号化装置、復号装置、認証暗号方法および認証暗号用プログラムを提供することを目的とする。

【課題を解決するための手段】

【００３２】

本発明による認証暗号装置は、入力された平文または暗号文に対して、2ブロックごとに、補助変数を入れた暗号化関数をラウンド関数に用いた2ラウンドFeistel構造を適用して暗号文または復号された平文を生成し、入力された平文または復号された平文のうちの一部のビットを用いて算出されるチェックサムに対して暗号化関数を適用して認証タグを生成する認証暗号手段を備え、前記認証暗号手段が、暗号化手段を含み、前記暗号化手段が、暗号化対象の平文と初期ベクトルとを入力する平文入力手段と、前記初期ベクトルと入力された平文のサイズとに基づき、前記暗号化関数の各々に与える補助変数を生成する補助変数生成手段と、前記平文を2ブロックごとのチャンクに分けたときの各平文チャンクに対して2ラウンドFeistel構造を適用することにより、当該平文チャンクに対応する暗号文チャンクを生成する2ラウンドFeistel暗号化手段と、前記平文のチェックサムを計算し、得られたチェックサムに対して、補助変数を入れた暗号化関数を適用させて認証タグを生成するタグ計算手段とを有し、前記2ラウンドFeistel暗号化手段は、初期ベクトルをN、チャンクのインデックスをi、i番目の平文チャンクをMC[i] = (M[i_1], M[i_2])、当該平文チャンクMC[i]に含まれる2つの平文ブロックに対応する補助変数を(N,Tw_i_1)と(N,Tw_i_2)の組、暗号化関数をF_K(*,*)とすると、i番目の暗号文チャンクCC[i] = (C[i_1], C[i_2])を、

C[i_1] = F_K((N,Tw_i_1), M[i_1]) xor M[i_2] ,

C[i_2] = F_K((N,Tw_i_2), C[i_1]) xor M[i_1]

と求め、前記タグ計算手段が、平文のチェックサムを、各平文チャンクに含まれる平文ブロックM[i_2]を用いて計算し、得られたチェックサムをSUM、認証タグ生成時に用いる暗号化関数に与える補助変数を(N,Tw_T_1)、暗号化関数をF_K(*,*)とすると、認証タグTを、

T = F_K((N,Tw_T_1), SUM)

と求め、前記補助変数生成手段が、入力された平文のサイズが偶数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズに満たない場合に、認証タグ生成時に用いる暗号化関数に与える補助変数として、第2の認証タグ用補助変数を生成し、前記暗号化手段が、入力された平文のサイズが偶数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズに満たない場合に、最終の平文ブロックを含む最終の平文チャンクに対して、所定の2ラウンドFeistel構造を適用して、最終の暗号文ブロックを含む最終の暗号文チャンクを生成する第2の2ラウンドFeistel暗号化手段と、入力された平文のサイズが偶数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズに満たない場合に、平文のチェックサムを、入力された平文と、前記第2の2ラウンドFeistel暗号化手段からの出力とを用いて計算し、得られたチェックサムに対して補助変数を与えた暗号化関数を適用させて、認証タグを生成する第2のタグ計算手段とを有し、前記第2の2ラウンドFeistel暗号化手段が、最終の平文チャンクのインデックスをm、最終の平文ブロックをM[m_2]、最終の平文チャンクをMC[m] = (M[m_1], M[m_2])、最終の平文チャンクMC[m]に含まれる2つの平文ブロックに対応する補助変数を(N,Tw_m_1)と(N,Tw_m_2)の組、暗号化関数をF_K(*,*)、最終の平文ブロックのサイズをs、ブロックサイズをn、sビットからnビットへのパディング処理をpad_n()、nビットからsビットへのカッティング処理をcut_s()とすると、sビットの最終の暗号文ブロックC[m_2]を含む最終の暗号文チャンクCC[m] = (C[m_1], C[m_2])を、

C[m_2] = cut_s(Z) xor M[m_2],

C[m_1] = F_K((N,Tw_m_2), pad_n(C[m_2])) xor M[m_1]

ただし、Z = F_K((N,Tw_m_1), M[m_1])

と求め、前記第2のタグ計算手段が、平文のチェックサムを、前記最終の平文チャンクを除く各平文チャンクに含まれる平文ブロックM[i_2]と、前記Zと、前記C[m_2]をnビットにパディングした結果であるC_n[m_2]と、を用いて計算し、得られたチェックサムをSUM、前記第2の認証タグ用補助変数を(N,Tw_T_2)、暗号化関数をF_K(*,*)とすると、認証タグTを、

T = F_K((N,Tw_T_2), SUM)

と求めることを特徴とする。

また、本発明による認証暗号装置は、入力された平文または暗号文に対して、2ブロックごとに、補助変数を入れた暗号化関数をラウンド関数に用いた2ラウンドFeistel構造を適用して暗号文または復号された平文を生成し、入力された前記平文または復号された前記平文のうちの一部のビットを用いて算出されるチェックサムに対して前記暗号化関数を適用して認証タグを生成する認証暗号手段を備え、前記認証暗号手段が、復号手段を含み、前記復号手段が、復号対象の暗号文と初期ベクトルと認証タグとを入力する暗号文入力手段と、前記初期ベクトルと入力された暗号文のサイズとに基づき、前記暗号化関数の各々に与える補助変数であって暗号化時と同じ補助変数を生成する復号用補助変数生成手段と、前記暗号文を2ブロックごとのチャンクに分けたときの各暗号文チャンクに対して2ラウンドFeistel構造を適用することにより、当該暗号文チャンクに対応する、復号された平文チャンクを生成する2ラウンドFeistel復号手段と、前記復号された平文のチェックサムを計算し、得られたチェックサムに対して、補助変数を入れた暗号化関数を適用させて復号検証用の認証タグを生成する復号検証用タグ計算手段と、前記復号検証用タグ計算手段が生成した復号検証用の認証タグと入力された認証タグとに基づいて、復号の成功または失敗を判定する判定手段とを有し、前記2ラウンドFeistel復号手段が、初期ベクトルをN、チャンクのインデックスをi、i番目の暗号文チャンクをCC[i] = (C[i_1], C[i_2])、当該暗号文チャンクCC[i]に含まれる2つの暗号文ブロックに対応する補助変数を(N,Tw_i_

1)と(N,Tw_i_2)の組、暗号化関数をF_K(*,*)とすると、i番目の復号された平文チャンクMC'[i] = (M'[i_1], M'[i_2])を、
M'[i_1] = F_K((N,Tw_i_2), C[i_1]) xor C[i_2],
M'[i_2] = F_K((N,Tw_i_1), M'[i_1]) xor C[i_1]
と求め、前記復号検証用タグ計算手段が、復号された平文のチェックサムを、復号された各平文チャンクに含まれる復号された各平文ブロックM'[i_2]を用いて計算し、得られたチェックサムをSUM'、復号検証用の認証タグ生成時に用いる暗号化関数に与える補助変数を(N,Tw_T_1)、暗号化関数をF_K(*,*)とすると、復号検証用の認証タグT'を、
T' = F_K((N,Tw_T_1), SUM')
と求め、前記復号用補助変数生成手段が、入力された暗号文のサイズが偶数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズに満たない場合に、復号検証用の認証タグ生成時に用いる暗号化関数に与える補助変数として、暗号時と同じ第2の認証タグ用補助変数を生成し、前記復号手段が、入力された暗号文のサイズが偶数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズに満たない場合に、最終の暗号文ブロックを含む最終の暗号文チャンクに対して、所定の2ラウンドFeistel構造を適用して、最終の暗号ブロックの復号された平文ブロックを含む最終の復号された平文チャンクを生成する第2の2ラウンドFeistel復号手段と、入力された暗号文のサイズが偶数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズに満たない場合に、復号された平文のチェックサムを、前記2ラウンドFeistel復号手段からの出力と、前記第2の2ラウンドFeistel復号手段からの出力と、前記最終の暗号文ブロックとを用いて計算し、得られたチェックサムに対して補助変数を与えた暗号化関数を適用させて、復号検証用の認証タグを生成する第2の復号検証用タグ計算手段とを有し、前記第2の2ラウンドFeistel復号手段が、最終の暗号文チャンクのインデックスをm、最終の暗号文ブロックをC[m_2]、最終の暗号文チャンクをCC[m] = (C[m_1], C[m_2])、最終の暗号文チャンクCC[m]に含まれる2つの暗号文ブロックに対応する補助変数を(N,Tw_m_1)と(N,Tw_m_2)の組、暗号化関数をF_K(*,*)、最終の暗号文ブロックのサイズをs、ブロックサイズをn、sビットからnビットへのパディング処理をpad_n()、nビットからsビットへのカッティング処理をcut_s()とすると、sビットの最終の復号された平文ブロックM'[m_2]を含む最終の復号された平文チャンクMC'[m] = (M'[m_1], M'[m_2])を、
M'[m_1] = F_K((N,Tw_m_2), pad_n(C[m_2])) xor C[m_1],
M'[m_2] = cut_s(Z') xor C[m_2],
ただし、Z' = F_K((N,Tw_m_1), M'[m_1])
と求め、前記第2の復号検証用タグ計算手段が、復号された平文のチェックサムを、前記最終の復号された平文チャンクを除く復号された各平文チャンクに含まれる復号された平文ブロックM'[i_2]と、前記Z'と、前記C[m_2]をnビットにパディングした結果であるC_n[m_2]と、を用いて計算し、得られたチェックサムをSUM'、前記第2の認証タグ用補助変数を(N,Tw_T_2)、暗号化関数をF_K(*,*)とすると、復号検証用の認証タグT'を、
T' = F_K((N,Tw_T_2), SUM')
と求めることを特徴とする。

【００３３】

本発明による暗号化装置は、入力された平文に対して、2ブロックごとに、補助変数を入れた暗号化関数をラウンド関数に用いた2ラウンドFeistel構造を適用して暗号文を生成し、入力された平文のうちの一部のビットを用いて算出されるチェックサムに対して暗号化関数を適用して認証タグを生成する暗号化手段を備え、前記暗号化手段が、暗号化対象の平文と初期ベクトルとを入力する平文入力手段と、前記初期ベクトルと入力された平文のサイズとに基づき、前記暗号化関数の各々に与える補助変数を生成する補助変数生成手段と、前記平文を2ブロックごとのチャンクに分けたときの各平文チャンクに対して2ラウンドFeistel構造を適用することにより、当該平文チャンクに対応する暗号文チャンクを生成する2ラウンドFeistel暗号化手段と、前記平文のチェックサムを計算し、得られたチ

ェックサムに対して、補助変数を入れた暗号化関数を適用させて認証タグを生成するタグ計算手段とを有し、前記2ラウンドFeistel暗号化手段が、初期ベクトルをN、チャンクのインデックスをi、i番目の平文チャンクをMC[i] = (M[i_1], M[i_2])、当該平文チャンクMC[i]に含まれる2つの平文ブロックに対応する補助変数を(N,Tw_i_1)と(N,Tw_i_2)の組、暗号化関数をF_K(*,*)とすると、i番目の暗号文チャンクCC[i] = (C[i_1], C[i_2])を、

C[i_1] = F_K((N,Tw_i_1), M[i_1]) xor M[i_2] ,

C[i_2] = F_K((N,Tw_i_2), C[i_1]) xor M[i_1]

と求め、前記タグ計算手段が、平文のチェックサムを、各平文チャンクに含まれる平文ブロックM[i_2]を用いて計算し、得られたチェックサムをSUM、認証タグ生成時に用いる暗号化関数に与える補助変数を(N,Tw_T_1)、暗号化関数をF_K(*,*)とすると、認証タグTを、

T = F_K((N,Tw_T_1), SUM)

と求め、前記補助変数生成手段が、入力された平文のサイズが偶数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズに満たない場合に、認証タグ生成時に用いる暗号化関数に与える補助変数として、第2の認証タグ用補助変数を生成し、前記暗号化手段が、入力された平文のサイズが偶数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズに満たない場合に、最終の平文ブロックを含む最終の平文チャンクに対して、所定の2ラウンドFeistel構造を適用して、最終の暗号文ブロックを含む最終の暗号文チャンクを生成する第2の2ラウンドFeistel暗号化手段と、入力された平文のサイズが偶数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズに満たない場合に、平文のチェックサムを、入力された平文と、前記第2の2ラウンドFeistel暗号化手段からの出力とを用いて計算し、得られたチェックサムに対して補助変数を与えた暗号化関数を適用させて、認証タグを生成する第2のタグ計算手段とを有し、前記第2の2ラウンドFeistel暗号化手段が、最終の平文チャンクのインデックスをm、最終の平文ブロックをM[m_2]、最終の平文チャンクをMC[m] = (M[m_1], M[m_2])、最終の平文チャンクMC[m]に含まれる2つの平文ブロックに対応する補助変数を(N,Tw_m_1)と(N,Tw_m_2)の組、暗号化関数をF_K(*,*)、最終の平文ブロックのサイズをs、ブロックサイズをn、sビットからnビットへのパディング処理をpad_n()、nビットからsビットへのカッティング処理をcut_s()とすると、sビットの最終の暗号文ブロックC[m_2]を含む最終の暗号文チャンクCC[m] = (C[m_1], C[m_2])を、

C[m_2] = cut_s(Z) xor M[m_2],

C[m_1] = F_K((N,Tw_m_2), pad_n(C[m_2])) xor M[m_1]

ただし、Z = F_K((N,Tw_m_1), M[m_1])

と求め、前記第2のタグ計算手段が、平文のチェックサムを、前記最終の平文チャンクを除く各平文チャンクに含まれる平文ブロックM[i_2]と、前記Zと、前記C[m_2]をnビットにパディングした結果であるC_n[m_2]と、を用いて計算し、得られたチェックサムをSUM、前記第2の認証タグ用補助変数を(N,Tw_T_2)、暗号化関数をF_K(*,*)とすると、認証タグTを、

T = F_K((N,Tw_T_2), SUM)

と求めることを特徴とする。

【００３４】

本発明による復号装置は、入力された暗号文に対して、2ブロックごとに、補助変数を入れた暗号化関数をラウンド関数に用いた2ラウンドFeistel構造を適用して復号された平文を生成し、復号された平文のうちの一部のビットを用いて算出されるチェックサムに対して暗号化関数を適用して認証タグを生成する復号手段を備え、前記復号手段が、復号対象の暗号文と初期ベクトルと認証タグとを入力する暗号文入力手段と、前記初期ベクトルと入力された暗号文のサイズとに基づき、前記暗号化関数の各々に与える補助変数であって暗号化時と同じ補助変数を生成する復号用補助変数生成手段と、前記暗号文を2ブロックごとのチャンクに分けたときの各暗号文チャンクに対して2ラウンドFeistel構造を適用することにより、当該暗号文チャンクに対応する、復号された平文チャンクを生成する2

ラウンドFeistel復号手段と、前記復号された平文のチェックサムを計算し、得られたチェックサムに対して、補助変数を入れた暗号化関数を適用させて復号検証用の認証タグを生成する復号検証用タグ計算手段と、前記復号検証用タグ計算手段が生成した復号検証用の認証タグと入力された認証タグとに基づいて、復号の成功または失敗を判定する判定手段とを有し、前記2ラウンドFeistel復号手段は、初期ベクトルをN、チャンクのインデックスをi、i番目の暗号文チャンクをCC[i] = (C[i_1], C[i_2])、当該暗号文チャンクCC[i]に含まれる2つの暗号文ブロックに対応する補助変数を(N,Tw_i_1)と(N,Tw_i_2)の組、暗号化関数をF_K(*,*)とすると、i番目の復号された平文チャンクMC'[i] = (M'[i_1], M'[i_2])を、

M'[i_1] = F_K((N,Tw_i_2), C[i_1]) xor C[i_2],

M'[i_2] = F_K((N,Tw_i_1), M'[i_1]) xor C[i_1]

と求め、前記復号検証用タグ計算手段が、復号された平文のチェックサムを、復号された各平文チャンクに含まれる復号された各平文ブロックM'[i_2]を用いて計算し、得られたチェックサムをSUM'、復号検証用の認証タグ生成時に用いる暗号化関数に与える補助変数を(N,Tw_T_1)、暗号化関数をF_K(*,*)とすると、復号検証用の認証タグT'を、

T' = F_K((N,Tw_T_1), SUM')

と求め、前記復号用補助変数生成手段が、入力された暗号文のサイズが偶数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズに満たない場合に、復号検証用の認証タグ生成時に用いる暗号化関数に与える補助変数として、暗号時と同じ第2の認証タグ用補助変数を生成し、前記復号手段が、入力された暗号文のサイズが偶数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズに満たない場合に、最終の暗号文ブロックを含む最終の暗号文チャンクに対して、所定の2ラウンドFeistel構造を適用して、最終の暗号ブロックの復号された平文ブロックを含む最終の復号された平文チャンクを生成する第2の2ラウンドFeistel復号手段と、入力された暗号文のサイズが偶数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズに満たない場合に、復号された平文のチェックサムを、前記2ラウンドFeistel復号手段からの出力と、前記第2の2ラウンドFeistel復号手段からの出力と、前記最終の暗号文ブロックとを用いて計算し、得られたチェックサムに対して補助変数を与えた暗号化関数を適用させて、復号検証用の認証タグを生成する第2の復号検証用タグ計算手段とを有し、前記第2の2ラウンドFeistel復号手段が、最終の暗号文チャンクのインデックスをm、最終の暗号文ブロックをC[m_2]、最終の暗号文チャンクをCC[m] = (C[m_1], C[m_2])、最終の暗号文チャンクCC[m]に含まれる2つの暗号文ブロックに対応する補助変数を(N,Tw_m_1)と(N,Tw_m_2)の組、暗号化関数をF_K(*,*)、最終の暗号文ブロックのサイズをs、ブロックサイズをn、sビットからnビットへのパディング処理をpad_n()、nビットからsビットへのカッティング処理をcut_s()とすると、sビットの最終の復号された平文ブロックM'[m_2]を含む最終の復号された平文チャンクMC'[m] = (M'[m_1], M'[m_2])を、

M'[m_1] = F_K((N,Tw_m_2), pad_n(C[m_2])) xor C[m_1],

M'[m_2] = cut_s(Z') xor C[m_2],

ただし、Z' = F_K((N,Tw_m_1), M'[m_1])

と求め、前記第2の復号検証用タグ計算手段が、復号された平文のチェックサムを、前記最終の復号された平文チャンクを除く復号された各平文チャンクに含まれる復号された平文ブロックM'[i_2]と、前記Z'と、前記C[m_2]をnビットにパディングした結果であるC_n[m_2]と、を用いて計算し、得られたチェックサムをSUM'、前記第2の認証タグ用補助変数を(N,Tw_T_2)、暗号化関数をF_K(*,*)とすると、復号検証用の認証タグT'を、

T' = F_K((N,Tw_T_2), SUM')

と求めることを特徴とする。

【００３５】

本発明による認証暗号方法は、情報処理装置が、入力された平文または暗号文に対して、2ブロックごとに、補助変数を入れた暗号化関数をラウンド関数に用いた2ラウンドFeis

tel構造を適用して暗号文または復号された平文を生成し、入力された平文または復号された平文のうちの一部のビットを用いて算出されるチェックサムに対して暗号化関数を適用して認証タグを生成し、前記情報処理装置が、前記認証タグを生成する処理で、暗号化処理を実行し、前記暗号化処理で、暗号化対象の平文と初期ベクトルとを入力する平文入力処理と、前記初期ベクトルと入力された平文のサイズとに基づき、前記暗号化関数の各々に与える補助変数を生成する補助変数生成処理と、前記平文を2ブロックごとのチャンクに分けたときの各平文チャンクに対して2ラウンドFeistel構造を適用することにより、当該平文チャンクに対応する暗号文チャンクを生成する2ラウンドFeistel暗号化処理と、前記平文のチェックサムを計算し、得られたチェックサムに対して、補助変数を入れた暗号化関数を適用させて認証タグを生成するタグ計算処理とを実行し、前記2ラウンドFeistel暗号化処理で、初期ベクトルをN、チャンクのインデックスをi、i番目の平文チャンクをMC[i] = (M[i_1], M[i_2])、当該平文チャンクMC[i]に含まれる2つの平文ブロックに対応する補助変数を(N,Tw_i_1)と(N,Tw_i_2)の組、暗号化関数をF_K(*,*)とすると、i番目の暗号文チャンクCC[i] = (C[i_1], C[i_2])を、

C[i_1] = F_K((N,Tw_i_1), M[i_1]) xor M[i_2] ,

C[i_2] = F_K((N,Tw_i_2), C[i_1]) xor M[i_1]

と求め、前記タグ計算処理で、平文のチェックサムを、各平文チャンクに含まれる平文ブロックM[i_2]を用いて計算し、得られたチェックサムをSUM、認証タグ生成時に用いる暗号化関数に与える補助変数を(N,Tw_T_1)、暗号化関数をF_K(*,*)とすると、認証タグTを、

T = F_K((N,Tw_T_1), SUM)

と求め、前記補助変数生成処理で、入力された平文のサイズが偶数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズに満たない場合に、認証タグ生成時に用いる暗号化関数に与える補助変数として、第2の認証タグ用補助変数を生成し、前記暗号化処理で、入力された平文のサイズが偶数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズに満たない場合に、最終の平文ブロックを含む最終の平文チャンクに対して、所定の2ラウンドFeistel構造を適用して、最終の暗号文ブロックを含む最終の暗号文チャンクを生成する第2の2ラウンドFeistel暗号化処理と、入力された平文のサイズが偶数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズに満たない場合に、平文のチェックサムを、入力された平文と、前記第2の2ラウンドFeistel暗号化処理の出力とを用いて計算し、得られたチェックサムに対して補助変数を与えた暗号化関数を適用させて、認証タグを生成する第2のタグ計算処理とを実行し、前記第2の2ラウンドFeistel暗号化処理で、最終の平文チャンクのインデックスをm、最終の平文ブロックをM[m_2]、最終の平文チャンクをMC[m] = (M[m_1], M[m_2])、最終の平文チャンクMC[m]に含まれる2つの平文ブロックに対応する補助変数を(N,Tw_m_1)と(N,Tw_m_2)の組、暗号化関数をF_K(*,*)、最終の平文ブロックのサイズをs、ブロックサイズをn、sビットからnビットへのパディング処理をpad_n()、nビットからsビットへのカッティング処理をcut_s()とすると、sビットの最終の暗号文ブロックC[m_2]を含む最終の暗号文チャンクCC[m] = (C[m_1], C[m_2])を、

C[m_2] = cut_s(Z) xor M[m_2],

C[m_1] = F_K((N,Tw_m_2), pad_n(C[m_2])) xor M[m_1]

ただし、Z = F_K((N,Tw_m_1), M[m_1])

と求め、前記第2のタグ計算処理で、平文のチェックサムを、前記最終の平文チャンクを除く各平文チャンクに含まれる平文ブロックM[i_2]と、前記Zと、前記C[m_2]をnビットにパディングした結果であるC_n[m_2]と、を用いて計算し、得られたチェックサムをSUM、前記第2の認証タグ用補助変数を(N,Tw_T_2)、暗号化関数をF_K(*,*)とすると、認証タグTを、

T = F_K((N,Tw_T_2), SUM)

と求めることを特徴とする。

また、本発明による認証暗号方法は、情報処理装置が、入力された平文または暗号文に

対して、2ブロックごとに、補助変数を入れた暗号化関数をラウンド関数に用いた2ラウンドFeistel構造を適用して暗号文または復号された平文を生成し、入力された前記平文または復号された前記平文のうちの一部のビットを用いて算出されるチェックサムに対して前記暗号化関数を適用して認証タグを生成し、前記情報処理装置が、前記認証タグを生成する処理で、復号処理を実行し、前記復号処理で、復号対象の暗号文と初期ベクトルと認証タグとを入力する暗号文入力処理と、前記初期ベクトルと入力された暗号文のサイズとに基づき、前記暗号化関数の各々に与える補助変数であって暗号化時と同じ補助変数を生成する復号用補助変数生成処理と、前記暗号文を2ブロックごとのチャンクに分けたときの各暗号文チャンクに対して2ラウンドFeistel構造を適用することにより、当該暗号文チャンクに対応する、復号された平文チャンクを生成する2ラウンドFeistel復号処理と、前記復号された平文のチェックサムを計算し、得られたチェックサムに対して、補助変数を入れた暗号化関数を適用させて復号検証用の認証タグを生成する復号検証用タグ計算処理と、前記復号検証用タグ計算処理で生成した復号検証用の認証タグと入力された認証タグとに基づいて、復号の成功または失敗を判定する判定処理とを実行し、前記2ラウンドFeistel復号処理で、初期ベクトルをN、チャンクのインデックスをi、i番目の暗号文チャンクをCC[i] = (C[i_1], C[i_2])、当該暗号文チャンクCC[i]に含まれる2つの暗号文ブロックに対応する補助変数を(N,Tw_i_1)と(N,Tw_i_2)の組、暗号化関数をF_K(*,*)とすると、i番目の復号された平文チャンクMC'[i] = (M'[i_1], M'[i_2])を、

M'[i_1] = F_K((N,Tw_i_2), C[i_1]) xor C[i_2],

M'[i_2] = F_K((N,Tw_i_1), M'[i_1]) xor C[i_1]

と求め、前記復号検証用タグ計算処理で、復号された平文のチェックサムを、復号された各平文チャンクに含まれる復号された各平文ブロックM'[i_2]を用いて計算し、得られたチェックサムをSUM'、復号検証用の認証タグ生成時に用いる暗号化関数に与える補助変数を(N,Tw_T_1)、暗号化関数をF_K(*,*)とすると、復号検証用の認証タグT'を、

T' = F_K((N,Tw_T_1), SUM')

と求め、前記復号用補助変数生成処理で、入力された暗号文のサイズが偶数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズに満たない場合に、復号検証用の認証タグ生成時に用いる暗号化関数に与える補助変数として、暗号時と同じ第2の認証タグ用補助変数を生成し、前記復号処理で、入力された暗号文のサイズが偶数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズに満たない場合に、最終の暗号文ブロックを含む最終の暗号文チャンクに対して、所定の2ラウンドFeistel構造を適用して、最終の暗号ブロックの復号された平文ブロックを含む最終の復号された平文チャンクを生成する第2の2ラウンドFeistel復号処理と、入力された暗号文のサイズが偶数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズに満たない場合に、復号された平文のチェックサムを、前記2ラウンドFeistel復号処理の出力と、前記第2の2ラウンドFeistel復号処理の出力と、前記最終の暗号文ブロックとを用いて計算し、得られたチェックサムに対して補助変数を与えた暗号化関数を適用させて、復号検証用の認証タグを生成する第2の復号検証用タグ計算処理とを実行し、前記第2の2ラウンドFeistel復号処理で、最終の暗号文チャンクのインデックスをm、最終の暗号文ブロックをC[m_2]、最終の暗号文チャンクをCC[m] = (C[m_1], C[m_2])、最終の暗号文チャンクCC[m]に含まれる2つの暗号文ブロックに対応する補助変数を(N,Tw_m_1)と(N,Tw_m_2)の組、暗号化関数をF_K(*,*)、最終の暗号文ブロックのサイズをs、ブロックサイズをn、sビットからnビットへのパディング処理をpad_n()、nビットからsビットへのカッティング処理をcut_s()とすると、sビットの最終の復号された平文ブロックM'[m_2]を含む最終の復号された平文チャンクMC'[m] = (M'[m_1], M'[m_2])を、

M'[m_1] = F_K((N,Tw_m_2), pad_n(C[m_2])) xor C[m_1],

M'[m_2] = cut_s(Z') xor C[m_2],

ただし、Z' = F_K((N,Tw_m_1), M'[m_1])

と求め、前記第2の復号検証用タグ計算処理で、復号された平文のチェックサムを、前記最終の復号された平文チャンクを除く復号された各平文チャンクに含まれる復号された平

文ブロックM'[i_2]と、前記Z'と、前記C[m_2]をnビットにパディングした結果であるC_n[m_2]と、を用いて計算し、得られたチェックサムをSUM'、前記第2の認証タグ用補助変数を(N,Tw_T_2)、暗号化関数をF_K(*,*)とすると、復号検証用の認証タグT'を、

T' = F_K((N,Tw_T_2), SUM')

と求めることを特徴とする。

【００３６】

本発明による認証暗号用プログラムは、コンピュータに、入力された平文または暗号文に対して、2ブロックごとに、補助変数を入れた暗号化関数をラウンド関数に用いた2ラウンドFeistel構造を適用して暗号文または復号された平文を生成し、入力された平文または復号された平文のうちの一部のビットを用いて算出されるチェックサムに対して暗号化関数を適用して認証タグを生成する処理を実行させ、前記処理で、暗号化処理を実行させ、前記暗号化処理で、暗号化対象の平文と初期ベクトルとを入力する平文入力処理と、前記初期ベクトルと入力された平文のサイズとに基づき、前記暗号化関数の各々に与える補助変数を生成する補助変数生成処理と、前記平文を2ブロックごとのチャンクに分けたときの各平文チャンクに対して2ラウンドFeistel構造を適用することにより、当該平文チャンクに対応する暗号文チャンクを生成する2ラウンドFeistel暗号化処理と、前記平文のチェックサムを計算し、得られたチェックサムに対して、補助変数を入れた暗号化関数を適用させて認証タグを生成するタグ計算処理とを実行させ、前記2ラウンドFeistel暗号化処理で、初期ベクトルをN、チャンクのインデックスをi、i番目の平文チャンクをMC[i] = (M[i_1], M[i_2])、当該平文チャンクMC[i]に含まれる2つの平文ブロックに対応する補助変数を(N,Tw_i_1)と(N,Tw_i_2)の組、暗号化関数をF_K(*,*)とすると、i番目の暗号文チャンクCC[i] = (C[i_1], C[i_2])を、

C[i_1] = F_K((N,Tw_i_1), M[i_1]) xor M[i_2] ,

C[i_2] = F_K((N,Tw_i_2), C[i_1]) xor M[i_1]

と求めさせ、前記タグ計算処理で、平文のチェックサムを、各平文チャンクに含まれる平文ブロックM[i_2]を用いて計算させ、得られたチェックサムをSUM、認証タグ生成時に用いる暗号化関数に与える補助変数を(N,Tw_T_1)、暗号化関数をF_K(*,*)とすると、認証タグTを、

T = F_K((N,Tw_T_1), SUM)

と求めさせ、前記補助変数生成処理で、入力された平文のサイズが偶数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズに満たない場合に、認証タグ生成時に用いる暗号化関数に与える補助変数として、第2の認証タグ用補助変数を生成させ、前記暗号化処理で、入力された平文のサイズが偶数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズに満たない場合に、最終の平文ブロックを含む最終の平文チャンクに対して、所定の2ラウンドFeistel構造を適用して、最終の暗号文ブロックを含む最終の暗号文チャンクを生成する第2の2ラウンドFeistel暗号化処理と、入力された平文のサイズが偶数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズに満たない場合に、平文のチェックサムを、入力された平文と、前記第2の2ラウンドFeistel暗号化処理の出力とを用いて計算し、得られたチェックサムに対して補助変数を与えた暗号化関数を適用させて、認証タグを生成する第2のタグ計算処理とを実行させ、前記第2の2ラウンドFeistel暗号化処理で、最終の平文チャンクのインデックスをm、最終の平文ブロックをM[m_2]、最終の平文チャンクをMC[m] = (M[m_1], M[m_2])、最終の平文チャンクMC[m]に含まれる2つの平文ブロックに対応する補助変数を(N,Tw_m_1)と(N,Tw_m_2)の組、暗号化関数をF_K(*,*)、最終の平文ブロックのサイズをs、ブロックサイズをn、sビットからnビットへのパディング処理をpad_n()、nビットからsビットへのカッティング処理をcut_s()とすると、sビットの最終の暗号文ブロックC[m_2]を含む最終の暗号文チャンクCC[m] = (C[m_1], C[m_2])を、

C[m_2] = cut_s(Z) xor M[m_2],

C[m_1] = F_K((N,Tw_m_2), pad_n(C[m_2])) xor M[m_1]

ただし、Z = F_K((N,Tw_m_1), M[m_1])

と求めさせ、前記第2のタグ計算処理で、平文のチェックサムを、前記最終の平文チャンクを除く各平文チャンクに含まれる平文ブロックM[i_2]と、前記Zと、前記C[m_2]をnビットにパディングした結果であるC_n[m_2]と、を用いて計算させ、得られたチェックサムをSUM、前記第2の認証タグ用補助変数を(N,Tw_T_2)、暗号化関数をF_K(*,*)とすると、認証タグTを、

T = F_K((N,Tw_T_2), SUM)

と求めさせることを特徴とする。

また、本発明による認証暗号用プログラムは、コンピュータに、入力された平文または暗号文に対して、2ブロックごとに、補助変数を入れた暗号化関数をラウンド関数に用いた2ラウンドFeistel構造を適用して暗号文または復号された平文を生成し、入力された前記平文または復号された前記平文のうちの一部のビットを用いて算出されるチェックサムに対して前記暗号化関数を適用して認証タグを生成する処理を実行させ、前記処理で、復号処理を実行させ、前記復号処理で、復号対象の暗号文と初期ベクトルと認証タグとを入力する暗号文入力処理と、前記初期ベクトルと入力された暗号文のサイズとに基づき、前記暗号化関数の各々に与える補助変数であって暗号化時と同じ補助変数を生成する復号用補助変数生成処理と、前記暗号文を2ブロックごとのチャンクに分けたときの各暗号文チャンクに対して2ラウンドFeistel構造を適用することにより、当該暗号文チャンクに対応する、復号された平文チャンクを生成する2ラウンドFeistel復号処理と、前記復号された平文のチェックサムを計算し、得られたチェックサムに対して、補助変数を入れた暗号化関数を適用させて復号検証用の認証タグを生成する復号検証用タグ計算処理と、前記復号検証用タグ計算処理で生成した復号検証用の認証タグと入力された認証タグとに基づいて、復号の成功または失敗を判定する判定処理とを実行させ、前記2ラウンドFeistel復号処理で、初期ベクトルをN、チャンクのインデックスをi、i番目の暗号文チャンクをCC[i] = (C[i_1], C[i_2])、当該暗号文チャンクCC[i]に含まれる2つの暗号文ブロックに対応する補助変数を(N,Tw_i_1)と(N,Tw_i_2)の組、暗号化関数をF_K(*,*)とすると、i番目の復号された平文チャンクMC'[i] = (M'[i_1], M'[i_2])を、

M'[i_1] = F_K((N,Tw_i_2), C[i_1]) xor C[i_2],

M'[i_2] = F_K((N,Tw_i_1), M'[i_1]) xor C[i_1]

と求めさせ、前記復号検証用タグ計算処理で、復号された平文のチェックサムを、復号された各平文チャンクに含まれる復号された各平文ブロックM'[i_2]を用いて計算させ、得られたチェックサムをSUM'、復号検証用の認証タグ生成時に用いる暗号化関数に与える補助変数を(N,Tw_T_1)、暗号化関数をF_K(*,*)とすると、復号検証用の認証タグT'を、

T' = F_K((N,Tw_T_1), SUM')

と求めさせ、前記復号用補助変数生成処理で、入力された暗号文のサイズが偶数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズに満たない場合に、復号検証用の認証タグ生成時に用いる暗号化関数に与える補助変数として、暗号時と同じ第2の認証タグ用補助変数を生成させ、前記復号処理で、入力された暗号文のサイズが偶数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズに満たない場合に、最終の暗号文ブロックを含む最終の暗号文チャンクに対して、所定の2ラウンドFeistel構造を適用して、最終の暗号ブロックの復号された平文ブロックを含む最終の復号された平文チャンクを生成する第2の2ラウンドFeistel復号処理と、

入力された暗号文のサイズが偶数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズに満たない場合に、復号された平文のチェックサムを、前記2ラウンドFeistel復号処理の出力と、前記第2の2ラウンドFeistel復号処理の出力と、前記最終の暗号文ブロックとを用いて計算し、得られたチェックサムに対して補助変数を与えた暗号化関数を適用させて、復号検証用の認証タグを生成する第2の復号検証用タグ計算処理とを実行させ、前記第2の2ラウンドFeistel復号処理で、最終の暗号文チャンクのインデックスをm、最終の暗号文ブロックをC[m_2]、最終の暗号文チャンクをCC[m] = (C[m_1], C[m_2])、最終の暗号文チャンクCC[m]に含まれる2つの暗号文ブロックに対応する補助変数を(N,Tw_m_1)と(N,Tw_m_2)の組、暗号化関数をF_K(*,*)、最終の暗号文ブロッ

クのサイズをs、ブロックサイズをn、sビットからnビットへのパディング処理をpad_n()、nビットからsビットへのカッティング処理をcut_s()とすると、sビットの最終の復号された平文ブロックM'[m_2]を含む最終の復号された平文チャンクMC'[m] = (M'[m_1], M'[m_2])を、
M'[m_1] = F_K((N,Tw_m_2), pad_n(C[m_2])) xor C[m_1],
M'[m_2] = cut_s(Z') xor C[m_2],
ただし、Z' = F_K((N,Tw_m_1), M'[m_1])
と求めさせ、前記第2の復号検証用タグ計算処理で、復号された平文のチェックサムを、前記最終の復号された平文チャンクを除く復号された各平文チャンクに含まれる復号された平文ブロックM'[i_2]と、前記Z'と、前記C[m_2]をnビットにパディングした結果であるC_n[m_2]と、を用いて計算させ、得られたチェックサムをSUM'、前記第2の認証タグ用補助変数を(N,Tw_T_2)、暗号化関数をF_K(*,*)とすると、復号検証用の認証タグT'を、
T' = F_K((N,Tw_T_2), SUM')
と求めさせることを特徴とする。

【発明の効果】
【００３７】
本発明によれば、１パスおよび１レート方式の認証暗号であって、並列処理が可能で、かつ１つの暗号化関数のみで全体の暗号化と復号の処理を実行可能な認証暗号を実現できる。

【図面の簡単な説明】
【００３８】
【図１】本発明の暗号化処理全体の処理フローの一例を模式的に示す説明図である。
【図２】本発明の復号処理全体の処理フローの一例を模式的に示す説明図である。
【図３】認証暗号システムが備える装置の例を示すブロック図である。
【図４】認証暗号システムの機能構成例を示すブロック図である。
【図５】２変数入力の疑似ランダム関数の実現例を模式的に示す説明図である。
【図６】（ａ）はブロック暗号のＸＥＸモードを用いた疑似ランダム関数による暗号化処理全体の処理フローの一例を模式的に示す説明図である。（ｂ）はブロック暗号のＸＥＸモードを用いた疑似ランダム関数による復号処理全体の処理フローの一例を模式的に示す説明図である。
【図７】２変数入力の疑似ランダム関数の他の実現例を模式的に示す説明図である。
【図８】第１の実施形態の認証暗号システムの暗号化動作の一例を示すフローチャートである。
【図９】第１の実施形態の認証暗号システムの復号動作の一例を示すフローチャートである。
【図１０】認証暗号システムの他の構成例を示すブロック図である。
【図１１】第２の実施形態における暗号化処理全体の処理フローの一例を模式的に示す説明図である。
【図１２】第２の実施形態における復号処理全体の処理フローの一例を模式的に示す説明図である。
【図１３】第3の実施形態における暗号化処理全体の処理フローの一例を模式的に示す説明図である。
【図１４】第3の実施形態における復号処理全体の処理フローの一例を模式的に示す説明図である。
【図１５】並列処理に対応した暗号化装置100の構成例を示すブロック図である。
【図１６】並列処理に対応した復号装置200の構成例を示すブロック図である。
【図１７】本発明の認証暗号装置の最小構成例を示すブロック図である。
【図１８】本発明の認証暗号装置の他の構成例を示すブロック図である。
【図１９】OCB方式による暗号化処理全体の処理フローの一例を模式的に示す説明図であ

る。
【図２０】OCB方式による復号処理全体の処理フローの一例を模式的に示す説明図である。
【発明を実施するための形態】
【００３９】
まず、本発明の概要を説明する。本発明は、共通秘密鍵を用いた暗号化方式であって、所定のサイズごとに暗号化を行うブロック暗号方式をベースにしている。ただし、本発明では、2ブロックごとに2ラウンドFeistel構造を適用する。そして、その2ラウンドFeistel構造のラウンド関数に、Tweak（調整値）と呼ばれる補助変数Twを入れた暗号化関数を用いる。このようにして、Feistel構造を2ブロック単位で認証暗号に用いることにより、上述した課題を解決する。
【００４０】
以下、より具体的に本発明の認証暗号方式を説明する。本発明では、暗号化関数に導入する補助変数Twとして次のような変数の系列を用いる。以下、補助変数Twとして用いる変数の系列を指して「補助系列」と呼ぶ場合がある。補助系列は、1つの鍵で暗号化を行っている間、呼び出す暗号化関数に対して全て異なる値が入力されるように構成する。以下では、1つの平文を暗号化するごとに値の異なる初期ベクトルNと、2ブロック単位のチャンクを識別する識別子iと、そのチャンク内のブロックおよびその他の処理を識別するjとを組み合わせた補助系列、すなわち(N,i,j)の組を補助変数Twとして用いる場合を例に示すが、これに限定されない。また、補助系列は、１つの平文に対して、暗号化時と復号時で同じ値のものが生成できるようにする。例えば、補助系列をどのような値で生成するかを予め定義づけておく。なお、上記の例は、初期ベクトルと、平文ないし暗号文の長さとから補助系列の値が一意に定まるため、上記条件を満たす。
【００４１】
例えば、各ブロックをnビットとしたとき、処理対象の平文または暗号文のブロック数が2mであったならば、各ブロックのインデックスに応じた補助系列として、(N,1,1), (N,1,2), (N,2,1), (N,2,2), ... , (N,m,1), (N,m,2)を用い、認証タグ生成用の補助系列として(N,m,3)を用いる、としてもよい。
【００４２】
このように補助系列を定めた上で、暗号化処理では、暗号化対象の平文Mを2ブロックごとのチャンクに分けて処理する。i番目の平文チャンク(M[2i-1],M[2i]、ただし(i=1, ... ,m))に対して、補助系列 (N,i,1)、(N,i,2)と、2変数入力の疑似ランダム関数であって鍵付きの疑似ランダム関数F_K(*,*)とを用いて、i番目の暗号文チャンク(C[2i], C[2i-1])を生成する。なお、「疑似ランダム関数」は、暗号化関数をその性質によって表すときの呼称である。疑似ランダム関数は、例えばブロック暗号の暗号化関数であってもよいし、鍵付きのハッシュ関数であってもよい。
【００４３】
図1は、本発明の暗号化処理全体の処理フローの一例を模式的に示す説明図である。図1において破線で囲んだブロックが2ラウンドのFeistel構造であって、2ブロック単位の暗号化処理の処理ブロックに相当する。各処理ブロックでは、例えば、第1の入力変数を補助系列とすると、補助系列 (N,i,1)を第1の入力変数としたF_K((N,i,1), *)と、補助系列 (N,i,2)を第1の入力変数としたF_K((N,i,2), *)とを使って、次のようにしてi番目の暗号文チャンク(C[2i], C[2i-1])を得る。なお、xorはビットごとの排他的論理和を表す。
【００４４】
C[2i-1] = F_K((N,i,1), M[2i-1]) xor M[2i],
C[2i] = F_K((N,i,2), C[2i-1]) xor M[2i-1]
・・・式（1）
【００４５】
上述の式（1）は、i番目の平文チャンクの奇数ブロックであるM[2i-1]に対して、当該i番目の平文チャンクの奇数ブロックに対応する補助系列 (N,i,1)を用いて疑似ランダム関

数F_K を実行して得た結果と当該i番目の平文チャンクの偶数ブロックであるM[2i]との排他的論理和を、i番目の暗号文チャンクの奇数ブロックである暗号文ブロックC[2i-1]とするとともに、そのようにして得た暗号文ブロックC[2i-1]に対して、当該i番目の平文チャンクの偶数ブロックに対応する補助系列（N,i,2)を用いて疑似ランダム関数F_Kを実行して得た結果と当該i番目の平文チャンクの奇数ブロックであるM[2i-1]との排他的論理和を、i番目の暗号文チャンクの偶数ブロックである暗号文ブロックC[2i]とすることを表している。これらをすべてのチャンクに対して行う。

【００４６】

さらに、上述の処理に加えて、平文の偶数ブロックM[2], M[4], ... ,M[2m]についてすべて排他的論理和をとった平文チェックサムSUM = M[2] xor M[4] xor ... xor M[2m]に対して、当該平文の認証タグ生成用の補助系列（N,m,3)と、鍵Kを持つ2変数入力の疑似ランダム関数F_Kとを用いて、認証タグTを生成する。なお、図1において1点鎖線で囲んだブロックが認証タグ生成処理の処理ブロックに相当する。認証タグTは、例えば、補助系列（N,m,3）を第1の入力変数としたF_K((N,m,3),*)を使って、次のようにして得られる。なお、SUMの計算における排他的論理和は任意の群における加算、例えば算術加算でもよい。

【００４７】

T = F_K((N,m,3), SUM) ・・・式（2）

【００４８】

なお、偶数ブロックに対する処理と奇数ブロックに対する処理を入れ替えることも可能である。そのような場合には、各奇数ブロックから平文チェックサムを生成すればよい。また、2ブロック単位のチャンクへの分け方もこの限りではない。オンライン計算をしないなど逐次処理の必要性がそれほど高くない場合には、例えば1ブロック目と3ブロック目とで1つのチャンクを作るといったことも可能である。また、補助系列も、図中の疑似ランダム関数に対して、1つの鍵で暗号化を行っている間、全て異なる値が入力されるよう構成されたものであればよい。なお、図中の、○の中に加算記号＋を書いた記号は排他的論理和をとることを表している。

【００４９】

復号側へは、暗号文Cと初期ベクトルNと認証タグTとが送られる。

【００５０】

復号側は、復号対象の暗号文Cと初期ベクトルNと認証タグTとが入力されると、まず上記と同じルールに従い、補助系列を決定する。例えば、各ブロックをnビットとしたとき暗号文C = (C[1], C[2], ... , C[2m])であれば、初期ベクトルNと暗号文の長さ情報から、暗号文Cの各ブロックのインデックスに応じた補助系列として、(N,1,1), (N,1,2), (N,2,1), (N,2,2), ... , (N,m,1), (N,m,2)を用い、認証タグ生成用の補助系列として(N,m,3)を用いると決定すればよい。

【００５１】

復号処理でも、復号対象の暗号文Cを2ブロックごとのチャンクに分けて、i番目の暗号文チャンク(C[2i-1], C[2i] 、ただし(i=1, ... , m))に対して、暗号化時と同じ補助系列を導入した疑似ランダム関数F_K(*,*)により構成される2ラウンドFeistel構造を利用した復号処理を行う。これにより、復号された平文チャンク(M'[2i], M'[2i-1])を得ることができる。

【００５２】

さらに、得られた平文の偶数ブロックM'[2], M'[4], ... ,M'[2m]についてすべて排他的論理和をとった平文チェックサムから、暗号化時に行った処理と同様の処理を行えば、復号検証用の認証タグTを得られる。

【００５３】

図2は、本発明の復号処理全体の処理フローの一例を模式的に示した説明図である。図2において破線で囲んだブロックが2ラウンドのFeistel構造であって、2ブロック単位の復号処理の処理ブロックに相当する。各処理ブロックでは、補助系列(N,i,1)を第1の入力変

数としたF_K((N,i,1), *)と、補助系列(N,i,2)を第1の入力変数としたF_K((N,i,2), *)とを使って、次のようにして復号された平文ブロックM'[2i], M'[2i-1]を得る。

【００５４】

M'[2i-1] = F_K((N,i,2), C[2i-1]) xor C[2i],
M'[2i] = F_K((N,i,1), M'[2i-1]) xor C[2i-1]
・・・式（3）

【００５５】

これらをすべての暗号文チャンクに対して行う。そして、復号された平文の偶数ブロックM'[2], M'[4], ... ,M'[2m]についてすべて排他的論理和をとった復号検証用の平文チェックサムSUM' = M'[2] xor M'[4] xor ... xor M'[2m]に対して、当該復号された平文の認証タグ生成用の補助系列（N,m,3)と、鍵Kを持つ2変数入力の疑似ランダム関数F_Kとを用いて、復号検証用の認証タグT'を生成する。なお、図2において1点鎖線で囲んだブロックが認証タグ生成処理の処理ブロックに相当する。復号検証用の認証タグT'は、例えば、補助系列(N,m,3)を第１の入力変数としたF_K((N,m,3), *)を使って、次のようにして得られる。

【００５６】

T' = F_K((N,m,3), SUM') ・・・式（4）

【００５７】

復号検証用の認証タグT'を得ると、入力された認証タグTと復号検証用の認証タグT'が一致するか否かを検査し、一致した場合は、復号された平文M' = (M'[1], M'[2], ..., M'[2m])を出力する。もし、一致しない場合、復号誤りを示すエラーメッセージを出力する。

【００５８】

本復号方式が暗号文を正しく復号できるのは、2ラウンドFeistel構造は任意のラウンド関数について置換を構成するため、暗号化関数の鍵と補助変数が決まれば、平文チャンクと暗号文チャンクが一対一対応すること、および、補助系列は初期ベクトルと平文ないし暗号文の長さから一意に定まるよう構成されており暗号化と復号の際に同じものが用いられることによる。

【００５９】

さらに付言すると、2ラウンドFeistel構造は個々のラウンドでの処理においてラウンド関数自体の逆処理（関数の出力から入力を求める処理）が不要であるだけでなく、全体としてもラウンド関数の逆処理が不要である。本方式では、ラウンド関数に暗号化関数を用いているので平文から暗号文への変換において、暗号化関数の逆処理を必要としない。また、認証タグの生成処理は、暗号化時と復号時で同じ処理を行うだけなので、この処理においても暗号化関数の逆処理を必要としない。

【００６０】

また、本方式の安全性は、暗号化関数の安全性に帰着できる。これは、本方式が2ラウンドFeistel構造において暗号化関数が2回適用される偶数ブロックのSUMから認証タグを生成するよう構成されていることによる。改ざんされた暗号文を復号した場合、復号結果の偶数ブロック目のどこかで改ざんを行った攻撃者（当然鍵は知らないものとする）にとって予測不可能な乱数が高い確率で発生するため、その排他的論理和である平文チェックサムも予測不可能となる。すると、これを暗号化関数に入力して得られる復号検証用の認証タグT'も予測不可能となるからである。

【００６１】

また、本方式によれば、認証暗号を暗号化関数のみで実現できるので、例えばHMACなどの鍵付きのハッシュ関数を用いることも可能となる。また、ブロック暗号やハッシュ関数によるもの以外にも様々な暗号技術をベースとすることが可能となる。

【００６２】

また、本方式によれば、2ブロック単位で処理が独立しており、また認証タグ用の平文チェックサムも得られた平文ブロックを順次演算処理すれば得られるので、オンライン計

算が可能である。

【００６３】

以下の第1の実施形態では、偶数ブロックに分割される平文を対象にした認証暗号方式を適用した認証暗号システムを説明する。第2の実施形態以降では、さらに拡張して、最終ブロックのサイズがブロックサイズに満たない場合や奇数ブロックとなる場合にも対応可能な認証暗号方式を適用した認証暗号システムを説明する。

【００６４】

実施形態1.

本発明による第1の実施形態に係る認証暗号システムの構成例を図3、図4、図5を参照して説明する。図3は、本実施形態の認証暗号システムが備える装置の例を示すブロック図である。図3に示すように、本実施形態のシステムは、情報処理装置50を備えている。情報処理装置50は、演算部51、記憶部52および入出力部53を含む。情報処理装置50は、例えばプログラムに従って動作するパーソナルコンピュータ等である。また、この場合、演算部51、記憶部52および入出力部53は、それぞれＣＰＵ、メモリおよび各種入出力装置（例えば、キーボード、マウス、ネットワークインタフェース部等）によって実現される。なお、図3では、1つの装置が演算部51、記憶部52および入出力部53の全てを含む例を示したが、これら演算部51、記憶部52および入出力部53は複数の装置に分散されていてもよい。

【００６５】

また、図4は、本実施形態の認証暗号システムの機能構成例を示すブロック図である。図4に示すように、認証暗号システムは、暗号化手段10を含む暗号化装置100と、復号手段20を含む復号装置200とを備えていてもよい。暗号化装置100および復号装置200は、例えば図3に示すような情報処理装置50によって実現される。

【００６６】

まず、暗号化装置100が備える暗号化手段10について説明する。暗号化手段10は、入力手段101と、補助変数生成手段102と、2ラウンドFeistel暗号化手段103と、タグ計算手段104と、出力手段105とを有する。

【００６７】

以下、とくに断りのない限り、１ブロックの長さをnビットとする。

【００６８】

入力手段101は、暗号化の対象となる平文Mと、初期ベクトルNを入力する。入力手段101は、例えばキーボードなどの文字入力装置により実現される。以下では偶数個のブロックを持つ平文M = (M[1], ... , M[2m])が入力されたとする。また、簡単のため、以降は初期ベクトルNはnビットであるものとするが、仮に短い場合は適当なパディングを行うか、別途nビット出力の可変長入力疑似ランダム関数（例えばCMAC、HMACにより実現可能）を適用してnビットに短縮するものとする。ここで、パディングとは、バイナリ系列に対して後ろに固定の系列を連結して、特定の長さにすることをいう。例えば、0詰めや10*詰め（最初が1で後ろが00...0となる）がある。なお、後者は、長さの違う系列でパディング後が同じ値になるのを防ぐ効果がある。本ケースの場合、0詰めでよい。

【００６９】

補助変数生成手段102は、初期ベクトルNと平文Mの長さ情報を元に、暗号化処理において疑似ランダム関数に与える、一般にTweakまたは調整値と呼ばれる補助変数を生成する。本実施形態では、補助変数として次のような変数の系列（補助系列）を生成する。一つの補助系列は、正整数iとjについて、(N,i,j)という形式の、3つの要素を持つベクトルで表される。平文が2mブロックある場合、補助系列は、(N,1,1), (N,1,2), (N,2,1), (N,2,2), ..., (N,m-1,1), (N,m-1,2), (N,m,1), (N,m,2), (N,m,3)となる。最後の一つを除いて暗号化に用いられ、最後の一つのみが認証タグの生成に用いられる。

【００７０】

2ラウンドFeistel暗号化手段103は、図1において破線で囲んだ各ブロック処理を実行する手段であり、平文Mを、2ブロック単位で分割し、補助入力と2変数入力の疑似ランダム関数F_K(*,*)を用いて暗号化を行う。2変数入力の疑似ランダム関数F_K(*,*)は、鍵付き

のn-bit出力関数であり、任意のx, yについて、F_K(x,y)が、鍵Kを知らない者には乱数と見分けがつかない出力となるものである。

【００７１】

以下では、F_K(*,*)の一番目の入力変数には補助系列のいずれかが入り、二番目には暗号化の対象とする平文ブロックであるn-bit変数が入る場合を例に説明する。本実施形態においても、平文M = (M[1], ... , M[2m])についてMC[i] = (M[2i-1], M[2i])とし、その１つであるMC[i]をi番目の平文チャンクと呼ぶ。また、CC[i] = (C[2i-1], C[2i])とし、その１つであるCC[i]をｉ番目の暗号文チャンクと呼ぶ。

【００７２】

2ラウンドFeistel暗号化手段103は、各i=1, ... , mについて、平文チャンクMC[i]を、例えば上述の式（1）のように処理して、暗号文チャンクCC[i]を得る。

【００７３】

既に説明したように、F_K(*,*)は様々な暗号学的関数により実現が可能である。例えば、ブロック暗号の暗号化関数を用いることも可能である。n-bitブロック暗号の暗号化関数E(*)を用いる場合、Y = F_K((N,i,j), X)の計算は、上述のTweakableブロック暗号と呼ばれるブロック暗号の拡張を利用することで効率よく行うことが可能である。

【００７４】

具体的には、上述の非特許文献2に記載のXEXモードを用いて、XEX変換式と同様に、Tweakである(N,i,j)と、秘密鍵Kから計算される系列mask_K(N,i,j)とをブロック暗号の入力に加算することで実現できる。このモードにおける変換式は、次のように表される。以下、この変換式をXE変換式と呼ぶ場合がある。

【００７５】

TE_K((N,i,j),X) = E_K(X xor mask_K(N,i,j))　・・・式（5）

【００７６】

式（5）に示すXE変換式を適用することにより、Y = F((N,i,j), X) に相当する計算が可能である。具体的なmask_K(N,i,j) の計算方法としては、例えば次に示す方法が挙げられる。

【００７７】

mask_K(N,i,j) = 2^i 3^(j-1) L, where L = E_K(N)　・・・式（6）

【００７８】

ここで、2^i や3^j は2や3を有限体GF(2^n)上の定数とみなしたうえでのべき乗演算であり、2^i 3^j LはLも有限体GF(2^n)上の要素とみなしたうえでの(2^i 3^j)との乗算を意味する。図5は、本実施形態で用いる2変数入力の疑似ランダム関数の、ブロック暗号のXEXモードを用いた実現例を模式的に示す説明図である。図5には、上述の方法により実現される2変数入力の疑似ランダム関数の例が示されている。また、図6（a）は、2変数入力の疑似ランダム関数を、ブロック暗号のXEXモードを用いて実現した場合の本実施形態の暗号化処理全体の処理フローの一例を模式的に示す説明図である。

【００７９】

ここで、Lは式（6）においてwhere節で示すように、L=E_K(N)である。したがって、Nが決まる度に1回E_Kを動作するだけでよい。Lが変化せずに、iやjが逐次的に変化するとき、B = 2^i 3^j Lは、過去の計算結果を利用したきわめて効率的な計算が可能であるため、最初にL=E(N)を求めた上では、上記の計算量を、実質的にほぼY = E(B xor X)の計算量と見なすことができる。有限体GF(2^n)上の定数の設定に関しては他にも非特許文献2に記載のXEXモードを用いた様々な方式が可能である。

【００８０】

なお、上述のXE変換式は、OCB方式が用いるXEX変換式とは異なり、外側にmask_K出力を加算していないが、これはOCB方式ではTE_Kの復号関数TD_Kを必要とするのに対し、本発明ではTE_Kのみで処理が可能であることに起因する。

【００８１】

なお、XEXモードの他にも、文献「Kazuhiko Minematsu, "Improved Security Analysis

of XEX and LRW Modes.", Selected Areas in Cryptography 2006, p.96-113.」（非特許文献4）に記載のTweakableブロック暗号や、文献「Niels ferguson, et al.,"The Skein Hash Function Family.", [online] 2008, インターネット<URL: http://www.skein-hash.info/sites/default/files/skein1.1.pdf"> 」（非特許文献5）に記載のTweakableブロック暗号の暗号化関数Threefishを用いることも可能である。

【００８２】

また、他の実現例としてF_K(*,*)に、例えばHMACなどの鍵付きハッシュ関数を用いることも可能である。この場合、Y = F_K((N,i,j), X)の計算は、(N,i,j)へ適当な可逆符号化を施したのち、Xと連結してHMACの入力とすればよい。

【００８３】

例えば、補助系列に用いた変数N,i,jを適当な固定長（例えば、128-bit）のバイナリ表現にして、それらを連結してもよい。図7は、本実施形態で用いる2変数入力の疑似ランダム関数の、鍵付きハッシュ関数を用いた実現例を模式的に示す説明図である。図7には、上述の方法により実現される2変数入力の疑似ランダム関数の例が示されている。図7における"||"記号はビット連結を表している。なお、入力Xと補助入力(N,i,j)については、HMAC_K(N || i || j || X)という出力になる。

【００８４】

また、例えば図5に示すE_Kの代わりにHMAC_Kを用いることによってもF_K(*,*)を実現できる。

【００８５】

i=1, ... , mについて上記の処理を行い、得られたC = (C[1], ... , C[2m])が暗号文となる。

【００８６】

タグ計算手段104は、図1において1点鎖線で囲んだブロック処理を実行する手段であり、補助変数生成手段102の出力する補助系列と、入力された平文とを用いて、メッセージ認証のための認証タグを計算する。

【００８７】

本実施形態におけるタグ計算手段104は、まず認証タグの生成に用いる平文チェックサムSUMを、平文の偶数ブロックM[2], M[4], ... , M[2m]を用いて次のように求める。なお、SUMの計算における排他的論理和は任意の群における加算、例えば算術加算でもよい。

【００８８】

SUM = M[2] xor M[4] xor ... xor M[2m]　・・・式（7）

【００８９】

すなわち、平文の偶数ブロックM[2], M[4], ... , M[2m]の全部の和を求めて、SUMとする。次いで、求めた平文チェックサムSUMに対して、2変数入力を持つ疑似ランダム関数F_K(*,*)と、認証タグ生成用の補助系列(N,m,3)とを用いて、上述の式（2）に示す処理を行い、認証タグTを求める。

【００９０】

出力手段105は、2ラウンドFeistel暗号化手段103の出力する暗号文C = (C[1], ... , C[2m])と、タグ計算手段104の出力する認証タグTを出力する。出力手段105は、例えば、暗号化を要求した上位アプリケーションに出力してもよいし、通信デバイス等を介して通信経路に出力してもよい。また、コンピュータディスプレイやプリンタなどに出力してもよい。

【００９１】

次に、復号装置200が備える復号手段20について説明する。復号手段20は、入力手段201と、補助変数生成手段202と、2ラウンドFeistel復号手段203と、復号検証用タグ計算手段204と、判定手段205と、出力手段206とを有する。

【００９２】

入力手段201は、復号の対象となる暗号文Cと、初期ベクトルNと、当該暗号文に対応づけられた認証タグTとを入力する。暗号化手段10と同様、以下では複数個のブロックを持

つ暗号文C = (C[1], ... , C[2m])が入力されたとする。また、簡単のため、以降は初期ベクトルNはnビットであるものとするが、仮に短い場合は適当なパディングを行うか、別途nビット出力の疑似ランダム関数を適用してn ビットに短縮するものとする。

【００９３】

補助変数生成手段202は、初期ベクトルNと暗号文Cの長さ情報を元に、復号処理において疑似ランダム関数に与える補助変数を生成する。本実施形態では、補助変数として、暗号化手段10の補助変数生成手段102と同じ出力を行う。

【００９４】

2ラウンドFeistel復号手段203は、図2において破線で囲んだ各ブロックに相当する手段であり、暗号文Cを、2ブロック単位で分割し、補助変数生成手段202の出力する補助系列と2変数入力の疑似ランダム関数F_K(*,*)を用いて復号を行う。ここで、2ラウンドFeistel復号手段203が用いる2変数入力の疑似ランダム関数F_K(*,*)は、暗号化手段10の2ラウンドFeistel暗号化手段103が用いるF_K(*,*)と同じものである。

【００９５】

2ラウンドFeistel復号手段203は、各i=1, ..., mについて、暗号文チャンクCC[i]を、例えば上述の式（3）のように処理して、復号された平文チャンクMC'[i] = (M'[2i-1], M'[2i])を得る。

【００９６】

各iについて上記の処理を行い、得られたM' = (M'[1], ... , M'[2m])が復号された平文となる。

【００９７】

復号検証用タグ計算手段204は、図2において1点鎖線で囲んだブロックに相当する手段であり、補助変数生成手段202の出力する補助系列と、2ラウンドFeistel復号手段203によって復号された平文M'を用いて、復号結果の検証を行うための復号検証用の認証タグT'を計算する。ここで、復号検証用タグ計算手段204が用いる2変数入力の疑似ランダム関数F_K(*,*)は、暗号化手段10のタグ計算手段104が用いるF_K(*,*)と同じものである。

【００９８】

復号検証用タグ計算手段204は、まず復号された平文M'を用いて復号検証用の平文チェックサムSUM'を、復号された平文の偶数ブロックM'[2], M'[4], ... , M'[2m]を用いて次のように求める。

【００９９】

SUM' = M'[2] xor M'[4] xor ... xor M'[2m]　・・・式（8）

【０１００】

すなわち、復号された平文の偶数ブロックM'[2], M'[4], ... , M'[2m]の全部の和を求めて、SUM'とする。次いで、求めた復号検証用の平文チェックサムSUM'に対して、2変数入力を持つ疑似ランダム関数F_K(*,*)と、認証タグ生成用の補助系列(N,m,3)とを用いて、上述の式（4）に示す処理を行い、認証タグT'を求める。なお、SUMの計算における排他的論理和は任意の群における加算、例えば算術加算でもよい。

【０１０１】

なお、図6（b）は、2変数入力の疑似ランダム関数を、ブロック暗号のXEXモードを用いて実現した場合の本実施形態の復号処理全体の処理フローの一例を模式的に示す説明図である。図6（b）に示す例においても、2変数入力の疑似ランダム関数へ外部より与えるパラメータを変えるだけで、図6（a）に示した暗号化処理と同じ処理によって復号された平文M'および復号検証用の認証タグT'が得られることがわかる。

【０１０２】

判定手段205は、入力された認証タグTと、復号検証用タグ計算手段204によって生成された復号検証用の認証タグT'とを比較し、一致する場合には、2ラウンドFeistel復号手段203の出力する、復号された平文M' = (M'[1], ... , M'[2m])を正しいものとして、復号成功と判定する。一方、T'とTが異なっている場合には、入力手段201が入力した(N,C,T)に改ざんがあったとして、復号失敗と判定する。

【０１０３】
出力手段206は、判定手段205による判定の結果、復号に成功したと判定された場合には、復号された平文M' = (M'[1], ... , M'[2m])を出力する。一方、復号に失敗したと判定された場合には、復号された平文M' = (M'[1], ... , M'[2m])を出力せずに、復号誤りを示すエラーを出力する。結果は、上位アプリケーションや、通信経路や、ディスプレイ装置やプリンタなどに出力される。
【０１０４】
本実施形態において、入力手段101および入力手段201は、例えば、装置が備えるキーボード、マウス、ネットワークインタフェース部等の各種入力装置とその制御部とによって実現される。また、出力手段105および出力手段206は、例えば、装置が備えるディスプレイ装置や、プリンタとのデバイスインタフェース部や、ネットワークインタフェース部等の各種出力装置とその制御部とによって実現される。また、補助変数生成手段102、補助変数生成手段202、2ラウンドFeistel暗号化手段103、2ラウンドFeistel復号手段203、タグ計算手段104、復号検証用タグ計算手段204、判定手段205は、例えば、装置が備える、プログラムに従って動作するCPU 等によって実現される。
【０１０５】
なお、図示省略しているが、暗号化手段10および復号手段20には、各々、上記各手段を適宜呼び出すなど暗号化処理または復号処理を取りまとめる制御手段が含まれている。
【０１０６】
次に、本実施形態の動作について説明する。図8および図9は、本実施形態の認証暗号システムの動作の一例を示すフローチャートである。なお、図8は、暗号化動作の一例を示すフローチャートであり、図9は、復号動作の一例を示すフローチャートである。
【０１０７】
まず、図8を参照して本実施形態による暗号化動作を説明する。図8に示す例では、まず入力手段101が、暗号化の対象となる偶数個のブロックを持つ平文M = (M[1], ... , M[2m])と、初期ベクトルNを入力する（ステップS101）。
【０１０８】
次に、補助変数生成手段102が、初期ベクトルNと平文Mの長さ情報を元に、補助系列を生成する（ステップS102）。本例では、2m個のブロックに分割できるとして、(N,1,1), (N,1,2), (N,2,1), (N,2,2), ... , (N,m-1,1), (N,m-1,2), (N,m,1), (N,m,2), (N,m,3)を出力する。なお、補助系列は一度に全てを生成する必要はなく、どのような補助系列を用いるかを決定しておけば、暗号化関数を呼び出す度にその暗号化関数に応じた補助系列を生成し、出力するようにしてもよい。
【０１０９】
次に、2ラウンドFeistel暗号化手段103が、平文Mを、2ブロック単位で分割し、補助変数生成手段102によって生成された補助系列と、所定の疑似ランダム関数F_K(*,*)とを用いて暗号化を行い、暗号文C = (C[1], ... , C[2m])を求める（ステップS103～S106）。2ラウンドFeistel暗号化手段103は、例えば、最初にiをi=1と初期化した上で（ステップS103）、iが示す平文チャンクMC[i] = (M[2i-1], M[2i])に対して、2ブロック単位でのFeistel暗号化処理すなわち上述の式（1）を実行する処理（ステップS104）を、++i > mを満たすまで繰り返し行えばよい。
【０１１０】
次に、タグ計算手段104が、平文Mの偶数ブロックを用いて平文チェックサムSUMを計算し（ステップS107）、得られたSUMと、認証タグ生成用の補助系列(N,m,3)とを用いて認証タグTを計算する（ステップS108）。
【０１１１】
最後に、出力手段105が得られた暗号文CとタグTを出力する（ステップS109）。
【０１１２】
次に、図9を参照して本実施形態による復号動作を説明する。図9に示す例では、まず入力手段201が、復号の対象となる偶数個のブロックを持つ暗号文C = (C[1], ... , C[2m])

と、初期ベクトルNと、認証タグTとを入力する（ステップS201）。

【０１１３】

次に、補助変数生成手段202が、初期ベクトルNと暗号文Cの長さ情報を元に、補助系列を生成する（ステップS202）。本例では、2m個のブロックに分割できるとして、(N,1,1), (N,1,2), (N,2,1), (N,2,2), ... , (N,m-1,1), (N,m-1,2), (N,m,1), (N,m,2), (N,m,3)を出力する。なお、補助系列は一度に全てを生成する必要はなく、どのような補助系列を用いるかを決定しておけば、暗号化関数を呼び出す度にその暗号化関数に応じた補助系列を生成し、出力するようにしてもよい。

【０１１４】

次に、2ラウンドFeistel復号手段203が、暗号文Cを、2ブロック単位で分割し、補助変数生成手段202によって生成された補助系列と、所定の疑似ランダム関数F_K(*,*)とを用いて復号を行い、復号された平文M' = (M'[1], ... , M'[2m])を求める（ステップS203～S206）。2ラウンドFeistel復号手段203は、例えば、最初にiをi=1と初期化した上で（ステップS203）、iが示す暗号文チャンクCC[i] = (C[2i-1], C[2i])に対して、2ブロック単位でのFeistel復号処理すなわち上述の式(3)を実行する処理（ステップS204）を、++i > mを満たすまで繰り返し行えばよい。

【０１１５】

次に、復号検証用タグ計算手段204が、復号された平文M'を用いて復号検証用の平文チェックサムSUM'を計算し（ステップS207）、得られたSUM'と、認証タグ生成用の補助系列(N,m,3)とを用いて復号検証用の認証タグT'を計算する（ステップS208）。

【０１１６】

次に、判定手段205が、入力された認証タグTと復号検証用の認証タグT'を比較し（ステップS209）、両者が等しいときには復号成功と判定し、等しくないときには復号失敗と判定する。

【０１１７】

最後に、出力手段206が、判定手段205の判定結果に基づき、復号された平文M'またはエラーメッセージを出力する。出力手段206は、判定手段205による判定の結果、復号成功のときは復号された平文M'を出力（ステップS210）し、復号失敗のときはエラーメッセージを出力する（ステップS211）。

【０１１８】

なお、上記は、暗号化装置と復号装置とが異なる装置により実現されている場合を例に説明したが、図10に示すように、一つの装置が暗号化手段10と復号手段20の両方を含んでいてもよい。そのような場合、暗号化手段10と復号手段20とを含む認証暗号手段30を備え、その認証暗号手段30において、復号手段20が、暗号化手段10の補助変数生成手段102、2ラウンドFeistel暗号化手段103およびタグ計算手段104を、補助変数生成手段202、2ラウンドFeistel復号手段203および復号検証用タグ計算手段204の代わりに用いることも可能である。なお、一つの装置が暗号化手段10と復号手段20の両方を備える場合においても、各手段を複数の装置に分けて実装することも可能である。

【０１１９】

暗号化手段10と復号手段20とで補助変数生成手段102を共用する場合、例えば呼び出し元が初期ベクトルや長さを指定できるようにすればよい。同様に、2ラウンドFeistel暗号化手段103を共用する場合、例えば呼び出し元が上段の2変数入力の疑似ランダム関数F_K(*,*)への入力および下段の2変数入力の疑似ランダム関数F_K(*,*)への入力を指定できるようにすればよい。同様に、タグ計算手段104を共用する場合、例えば呼び出し元が平文チェックサムを指定できるようにすればよい。

【０１２０】

以上のように、本実施形態によれば、高速でコンパクトな認証暗号を実現できる。その理由は、本発明の認証暗号方式は、S個のブロックの平文に対して、2変数入力の疑似ランダム関数 F_K(*,*)をS+1回コールすることで暗号化と復号を行うからである。

【０１２１】

例えば、2変数入力の疑似ランダム関数F_K(*,*)は、ブロック暗号の暗号化関数E(*)を用いる場合、暗号化関数EをTweakableブロック暗号対応の暗号化関数TE(*,*)に変換することで実現できるが、上述したE -> TE変換方式を利用すると、ブロック暗号の暗号化関数E(*)をたかだかS+ｈ回コールすることで、暗号化、復号ともが実現可能となる。

【０１２２】

一方、既存の2パス認証暗号方式では2×2m回以上の暗号化関数の呼び出しが必要となる。OCB方式の場合は、1パス認証暗号方式であるので本発明とほぼ同等の、S+2～S+3回程度の呼び出し回数となるが、復号処理においてブロック暗号の復号関数を必要とするため、ソフトウェア実装におけるROM/RAM使用量や、ハードウェア実装における回路規模の増加を引き起こす。

【０１２３】

また、本発明の認証暗号方式によれば、2ブロック単位で完全に並列処理が可能であるため、マルチコアCPUでのソフトウェア実装や、ハードウェア実装においてはさらなる高速化を実現することができる。

【０１２４】

実施形態2.

第2の実施形態における認証暗号システムは、暗号化ないし復号の対象が、偶数個のブロックを持つが最後のブロックがnビット未満の長さであるケースに対応したものである。なお、基本的な構成は第1の実施形態と同様であるので、以下異なる点についてのみ説明する。

【０１２５】

以下に示す例では、1ブロックをnビットとしたとき、偶数ブロックを持つ平文の最終ブロックM[2m]のビット長sがs < nであったとする。

【０１２６】

本実施形態では、新たに認証タグ生成用の第2の補助系列を定義する。以下に示す例では、(N,m,4)を新たに定義する。

【０１２７】

そして、2ブロック単位でのFeistel暗号化処理およびFeistel復号処理において、m番目のチャンクに対して、次のように処理する。なお、i=1, .. .,m-1番目のチャンクに対しては第1の実施形態と同様でよい。

【０１２８】

暗号化処理では、まずm番目の平文チャンクの奇数ブロックである平文ブロックM[2m-1]に対して補助系列(N,m,1)を与えた2変数入力の疑似ランダム関数F_K(*,*)を適用させる。そして、得られた出力を中間出力Zとするとともに、その中間出力Zの任意の固定箇所からsビットを取り出したsビットのバイナリ系列Z_sと、m番目の平文チャンクの偶数ブロックである平文ブロックM[2m]との排他的論理和をとることにより、m番目の暗号文チャンクの偶数ブロックである暗号文ブロックC_s[2m]を求める。なお、C_s[*]は該当する暗号文ブロックのサイズがsビットであることを意味している。

【０１２９】

さらに、そのようにして得たsビットの暗号文ブロックC_s[2m]に対してs→nビットのパディングを行い、その結果であるC_n[2m]に対して補助系列(N,m,2)を与えた2変数入力の疑似ランダム関数F_K(*,*)を適用させた後、m番目の平文チャンクの奇数ブロックである平文ブロックM[2m-1]との排他的論理和をとることにより、m番目の暗号文チャンクの奇数ブロックである暗号文ブロックC[2m-1]を求める。本実施形態では、そのようにしてm番目の暗号文チャンクCC[m] = (C[2m-1], C_s[2m])を得る。本実施形態におけるm番目の暗号文チャンクを得るための処理の一例を式に表すと、次のように表される。

【０１３０】

C_s[2m] = cut_s(Z) xor M[2m],
C[2m-1] = F_K((N,m,2), pad_n(C_s[2m]))) xor M[2m-1]
ただし、Z=F_K((N,m,1), M[2m-1])

・・・式（9）

【０１３１】

なお、cut_s(A)は、バイナリ系列Aのうち任意の固定箇所からsビットを取り出す処理を表す。cut_s(A)は、例えばmsb_s(A)であってもよい。msb_s(A)はバイナリ系列Aの最上位からsビットを取り出す処理である。また、pad_n(A)は、バイナリ系列Aがnビットとなるように任意の固定ビット列を用いてパディングを行う処理を表す。なお、本例では、パディングは10*詰めとする。10*詰めとすることで、長さの違う系列でパディング後が同じ値になるのを防ぐことができる。なお、長さの違う系列でパディング後が同じ値とならない形式であれば上記の例に限定されない。

【０１３２】

また、認証タグTは次のようにして求められる。まず、平文チェックサムSUMを、m-1番目までの平文チャンクの偶数ブロックM[2], ... , M[2(m-1)]と、上述の処理により得られた中間出力Zと、上述の処理により得られたm番目の暗号文チャンクのうちの偶数ブロックである暗号文ブロックC_s[2m]をnビットへパディングしたC_n[2m]とを用いて求める。例えば以下の式（10）に示すように、これらの排他的論理和をとることによって求める。

【０１３３】

SUM = M[2] xor M[4] xor ... xor M[2m-2] xor C_n[2m] xor Z

・・・式（10）

【０１３４】

なお、第1の実施形態で生成する平文チェックサムと比べて、M[2m]の代わりに「C_n[2m] xor Z」を用いる点が異なる。

【０１３５】

そして、そのようにして得た平文チェックサムSUMに対して、(N,m,4)を補助系列とする2変数入力の疑似ランダム関数F_K((N,m,4),*)を適用させることにより、認証タグTを求める。

【０１３６】

図11は本実施形態における暗号化処理全体の処理フローの一例を模式的に示す説明図である。図11に示すように、本実施形態における暗号化処理は、認証タグ生成に用いる補助系列と、ビットサイズがn未満の偶数ブロックを含むチャンクに対する暗号化処理と、平文チェックサムの計算方法が異なる以外は、第１の実施形態と同様である。なお、図11において「C_s[2m] || 10*」はC_s[2m]を10*詰めパディングした結果を表している。

【０１３７】

本実施形態の暗号化手段10は、例えば、偶数ブロックを持つ平文の最終ブロック M[2m]がs（s < n）ビットの場合に、補助変数生成手段102が補助系列(N,1,1), (N,1,2), (N,2,1), (N,2,2), … , (N,m,1), (N,m,2), (N,m,4)を生成するとともに、m番目の平文チャンクに対して上述した処理を行い暗号文チャンクCC[m]と中間出力Zとを出力する第2の2ラウンドFeistel暗号化手段と、上述した処理によって認証タグTを計算する第2のタグ計算手段とを備えていてもよい。または、第2の2ラウンドFeistel暗号化手段と第2のタグ計算手段とを備える代わりに、2ラウンドFeistel暗号化手段103およびタグ計算手段104が、平文のサイズに応じて第1の実施形態の動作と、上述した動作とを切り替えて実行してもよい。

【０１３８】

復号処理では、まずm番目の暗号文チャンクの偶数ブロックであるsビットの暗号文ブロックC_s[2m]に対してs→nビットのパディングを行う。そして、パディングして得られたnビットの暗号文ブロックC_n[2m]ビットに対して補助系列(N,m,2)を与えた2変数入力の疑似ランダム関数F_K(*,*)を適用させる。その結果と、m番目の暗号文チャンクの奇数ブロックに相当する暗号文ブロックC[2m-1]との排他的論理和をとることにより、m番目の復号された平文チャンクの奇数ブロックに相当する、復号された平文ブロックM'[2m-1]を求める。

【０１３９】

次に、そのようにして得た、復号された平文ブロックM'[2m-1]に対して補助系列(N,m,1)を与えた2変数入力の疑似ランダム関数F_K(*,*)を適用させる。そして、得られた出力を復号用中間出力Z'とするとともに、その復号用中間出力Z'の任意の固定箇所からsビットを取り出したsビットのバイナリ系列Z_s'と、m番目の暗号文チャンクの偶数ブロックである暗号文ブロックC_s[2m]との排他的論理和をとることにより、m番目の復号された平文チャンクの偶数ブロックに相当する、復号された平文ブロックM_s'[2m]を求める。なお、M_s'[*]は該当する復号された平文ブロックのサイズがsビットであることを意味している。本実施形態では、そのようにしてm番目の復号された平文チャンクMC'[m] = (M'[2m-1], M_s'[2m])を得る。本実施形態におけるm番目の復号された平文チャンクを得るための処理の一例を式に表すと、次のようになる。

【０１４０】

M'[2m-1] = F_K((N,m,2), pad_n(C_s[2m]))) xor C[2m-1],
M_s'[2m] = cut_s(Z') xor C_s[2m]
ただし、Z'=F_K((N,m,1), M'[2m-1])

・・・式（11）

【０１４１】

また、復号検証用の認証タグは次のようにして求められる。まず、復号された平文を用いて復号検証用の平文チェックサムSUM'を、m-1番目までの復号された平文チャンクの偶数ブロックM'[2], ... ,M'[2(m-1)]と、上述の処理により得られた復号用中間出力Z'と、入力されたm番目の暗号文チャンクの偶数ブロックである暗号文ブロックC_s[2m]をnビットへパディングしたC_n[2m]とを用いて求める。例えば以下の式（12）に示すように、これらの排他的論理和をとることによって求める。

【０１４２】

SUM' = M'[2] xor M'[4] xor ... xor M'[2m-2] xor C_n[2m] xor Z'

・・・式（12）

【０１４３】

なお、第1の実施形態で生成する復号検証用の平文チェックサムと比べて、M'[2m]の代わりに「C_n[2m] xor Z'」を用いる点が異なる。

【０１４４】

そして、そのようにして得た復号検証用の平文チェックサムSUM'に対して、(N,m,4)を補助系列とする2変数入力の疑似ランダム関数F_K((N,m,4),*)を適用させることにより、復号検証用の認証タグT'を計算する。

【０１４５】

図12は本実施形態における復号処理全体の処理フローの一例を模式的に示す説明図である。図12に示すように、本実施形態における復号処理は、認証タグ生成用の補助系列と、ビットサイズがn未満の偶数ブロックを含むチャンクに対する復号処理と、復号検証用の平文チェックサムの計算方法が異なる以外は、第１の実施形態と同様でよい。なお、図12において、「C_s[2m] || 10*」はC_s[2m]を10*詰めパディングした結果を表している。

【０１４６】

本実施形態の復号手段20は、偶数ブロックを持つ暗号文の最終ブロック C[2m]がs（s < n）ビットの場合に、m番目の暗号文チャンクに対して上述した処理を行い復号された平文チャンクMC'と復号用中間出力Z'とを出力する第2の2ラウンドFeistel復号手段と、上述した処理によって復号検証用の認証タグT'を計算する第2の復号検証用タグ計算手段とを備え、補助変数生成手段202が、補助系列(N,1,1), (N,1,2), (N,2,1), (N,2,2), … , (N,m,1), (N,m,2), (N,m,4)を生成するように構成されていてもよい。または、既に説明した2ラウンドFeistel復号手段203および復号検証用タグ計算手段204が各々、平文のサイズに応じて、第1の実施形態の動作と上述した動作とを切り替えて実行するように構成されていてもよい。

【０１４７】

なお、本実施形態においても、m番目のブロックに対する暗号化処理と復号処理とでは

、上段と下段が入れ替わっているだけであるので、例えば、上段と下段とを分けて部品化すれば暗号化処理と復号処理とで共用できる。また、認証タグの生成に関しても、第１の実施形態と同様、与えるパラメータを呼び出し元で指定するようにすれば共用できる。

【０１４８】

本実施形態においても、最後のチャンクが、補助入力と鍵が決まれば平文チャンクと暗号文チャンクが1対1で対応するように構成されているため、正しく復号できる。また、Feistel構造では個々のラウンドでの処理において暗号化関数F_K自体の逆処理が不要であり、本実施形態においても復号検証用の認証タグを暗号化処理と同様の方法により求めているため、全体としても暗号化関数F_K自体の逆処理が不要である点は、第1の実施形態と同様である。

【０１４９】

実施形態3.

第3の実施形態における認証暗号システムは、暗号化ないし復号の対象が、奇数個のブロックを持つケースに対応したものである。なお、基本的な構成は第1の実施形態と同様であるので、以下異なる点についてのみ説明する。

【０１５０】

本実施形態では、1ブロックをnビットとしたとき、平文のブロック数が奇数（2m-1）であり、その最終ブロックのビット長sがs ≦ nの場合に、次のように認証暗号を行う。

【０１５１】

まず、新たに認証タグ生成用の第3および第4の補助系列を定義する。以下に示す例では、(N,m,5)と(N,m,6)を新たに定義する。なお、奇数ブロックを持つ平文を対象とする場合、最終チャンクの偶数ブロックに対応する補助系列(N,m,2)は不要である。

【０１５２】

そして、2ブロック単位でのFeistel暗号化処理およびFeistel復号処理において、m番目のチャンクに対して、次のように処理する。なお、i=1, .. .,m-1番目のチャンクに対しては第1の実施形態と同様でよい。

【０１５３】

暗号化処理では、まず全て0からなるnビットのバイナリ系列を用意し、そのバイナリ系列に対して補助系列(N,m,1)を与えた2変数入力の疑似ランダム関数F_K(*,*)を適用させる。そして、得られた出力の任意の固定箇所からsビット取り出したsビットのバイナリ系列Z_sと、m番目の平文チャンクの奇数ブロックであり平文の最終ブロックであるsビットの平文ブロックM_s[2m-1]との排他的論理和をとることにより、m番目の暗号文チャンクにおける奇数ブロックすなわち暗号文の最終ブロックに相当する暗号文ブロックC_s[2m-1]を求める。

【０１５４】

本実施形態では、そのようにしてm番目の暗号文チャンクCC[m] = (C_s[2m-1])を得る。本実施形態におけるm番目の暗号文チャンクを得るための処理の一例を式に表すと、次のように表される。

【０１５５】

C_s[2m-1] = cut_s(F_K((N,m,1), 0^n)) xor M_s[2m-1]

・・・式（13）

【０１５６】

なお、最終ブロックのサイズがs=nである場合はcut_s()を省略して、以下の式（14）に示す処理を行うことにより、CC[m] = (C[2m-1])を得ればよい。

【０１５７】

C[2m-1] = F_K((N,m,1), 0^n) xor M[2m-1]

・・・式（14）

【０１５８】

また、認証タグTは次のようにして求められる。本実施形態では、もし最終ブロックのサイズs = nならば、平文チェックサムSUMを、m-1番目までの平文チャンクの偶数ブロッ

クM[2], ... , M[2(m-1)]と、最終ブロックである平文ブロックM[2m-1]とを用いて求める。例えば以下の式（15）に示すように、これらの排他的論理和をとることによって求める。

【０１５９】

SUM = M[2] xor M[4] xor ... xor M[2m-2] xor M[2m-1]

・・・式（15）

【０１６０】

一方、最終ブロックのサイズs < nならば、平文チェックサムSUMを、m-1番目までの平文チャンクの偶数ブロックM[2], ... , M[2(m-1)]と、最終ブロックであるsビットの平文ブロックM_s[2m-1]をnビットへパディングしたM_n[2m-1]とを用いて求める。例えば以下の式（16）に示すように、これらの排他的論理和をとることによって求める。

【０１６１】

SUM = M[2] xor M[4] xor ... xor M[2m-2] xor pad_n(M_s[2m-1])

・・・式（16）

【０１６２】

なお、第1の実施形態で生成する平文チェックサムと比べて、M[2m]の代わりに、M_n[2m-1]（M[2m-1]またはM_s[2m-1]をnビットに拡張したもの）を用いている点が異なる。

【０１６３】

そして、そのようにして得た平文チェックサムSUMに対して、s = nであれば(N,m,5)を補助系列とする2変数入力の疑似ランダム関数F_K((N,m,5),*)を適用させ、s < nであれば(N,m,6)を補助系列とする2変数入力の疑似ランダム関数F_K((N,m,6),*)を適用させることにより、認証タグTを求める。

【０１６４】

図13は本実施形態における暗号化処理全体の処理フローの一例を模式的に示す説明図である。図13に示すように、本実施形態における暗号化処理は、m番目のチャンクに対する暗号化処理と、認証タグの生成処理が異なる以外は、第１の実施形態と同様である。なお、図13において、「M_s[2m-1] || 10*」はsビットの平文ブロックM_s[2m]を10*詰めパディングした結果を表している。図13では最終ブロックのサイズがs < nの場合を示しているが、s=nの場合はm番目のブロックに対する処理におけるカッティング（図中の「cut_s」）と認証タグ生成処理におけるパディング（図中の「|| 10*」）を省略すればよい。なお、s=nであれば、M_s[m-1]=M[m-1]、C_s[2m-1]=C[2m-1]であり、仮にcut_s()やpad_n()を実行しても実質何ら処理が行われないため、cut_s()やpad_n()を省略しなくても特に問題はない。

【０１６５】

本実施形態の暗号化手段10は、例えば、平文が2m-1個のブロックを持つ場合に、補助変数生成手段102が補助系列(N,1,1), (N,1,2), (N,2,1), (N,2,2), … , (N,m,1), (N,m,5), (N,m,6)を生成するとともに、m番目の平文チャンクに対して上述した処理を行い暗号文チャンクCC[m]=(C_s[2m-1])を出力する1ラウンドFeistel暗号化手段と、上述した処理によって認証タグTを計算する第3のタグ計算手段とを備えていてもよい。または、1ラウンドFeistel暗号化手段と第3のタグ計算手段とを備える代わりに、2ラウンドFeistel暗号化手段103およびタグ計算手段104が、平文のサイズに応じて第1の実施形態の動作と、上述した動作とを切り替えて実行するように構成されていてもよい。なお、補助変数生成手段102、2ラウンドFeistel暗号化手段103およびタグ計算手段104が、平文のサイズに応じて、第1の実施形態の動作と第2の実施形態の動作と上述した動作とを切り替えて実行するように構成されていれば、ブロック数が偶数の場合、奇数の場合、最終ブロックがnサイズ未満の場合のいずれの場合にも対応できる。

【０１６６】

復号処理でも、まず全て0からなるnビットのバイナリ系列を用意し、そのバイナリ系列に対して補助系列(N,m,1)を与えた2変数入力の疑似ランダム関数F_K(*,*)を適用させる。そして、得られた出力の任意の固定箇所からsビット取り出したsビットのバイナリ系列Z_

sと、m番目の暗号文チャンクの奇数ブロックであり暗号文の最終ブロックであるsビットの暗号文ブロックC_s[2m-1]との排他的論理和をとることにより、m番目の復号された平文チャンクにおける奇数ブロックすなわち復号された平文の最終ブロックに相当する、復号された平文ブロックM_s'[2m-1]を求める。

【０１６７】

本実施形態では、そのようにしてm番目の復号された平文チャンクMC'[m] = (M_s'[2m-1])を得る。本実施形態におけるm番目の復号された平文チャンクを得るための処理の一例を式に表すと、次のように表される。

【０１６８】

M_s'[2m-1] = cut_s(F_K((N,m,1), 0^n)) xor C_s[2m-1]

・・・式（17）

【０１６９】

なお、最終ブロックのサイズがs=nである場合はcut_s()を省略して、以下の式（18）に示す処理を行うことにより、MC'[m] = (M'[2m-1])を得ればよい。

【０１７０】

M'[2m-1] = F_K((N,m,1), 0^n) xor C[2m-1]

・・・式（18）

【０１７１】

また、復号検証用の認証タグT'は、暗号化処理において平文を用いていたところを、復号された平文に置き換えることにより、求められる。すなわち、もし最終ブロックのサイズs = nならば、復号検証用の平文チェックサムSUM'を、m-1番目までの復号された平文チャンクの偶数ブロックM'[2], ... , M'[2(m-1)]と、最終ブロックである復号された平文ブロックM'[2m-1]とを用いて求める。例えば以下の式（19）に示すように、これらの排他的論理和をとることによって求める。

【０１７２】

SUM' = M'[2] xor M'[4] xor ... xor M'[2m-2] xor M'[2m-1]

・・・式（19）

【０１７３】

一方、最終ブロックのサイズs < nならば、復号検証用の平文チェックサムSUM'を、m-1番目までの復号された平文チャンクの偶数ブロックM'[2], ... , M'[2(m-1)]と、最終ブロックであるsビットの平文ブロックM_s'[2m-1]をnビットへパディングしたM_n'[2m-1]とを用いて求める。例えば以下の式（20）に示すように、これらの排他的論理和をとることによって求める。

【０１７４】

SUM' = M'[2] xor M'[4] xor ... xor M'[2m-2] xor pad_n(M_s'[2m-1])

・・・式（20）

【０１７５】

なお、第1の実施形態で生成する復号検証用の平文チェックサムと比べて、M'[2m]の代わりに、M_n'[2m-1]（M'[2m-1]またはM_s'[2m-1]をnビットに拡張したもの）を用いている点が異なる。

【０１７６】

そして、そのようにして得た復号検証用の平文チェックサムSUM'に対して、s = nであれば(N,m,5)を補助系列とする2変数入力の疑似ランダム関数F_K((N,m,5),*)を適用させ、s < nであれば(N,m,6)を補助系列とする2変数入力の疑似ランダム関数F_K((N,m,6),*)を適用させることにより、復号検証用の認証タグT'を求める。

【０１７７】

図14は本実施形態における復号処理全体の処理フローの一例を模式的に示す説明図である。図14に示すように、本実施形態における復号処理は、m番目の暗号文チャンクに対する復号処理と、復号検証用の認証タグの生成処理が異なる以外は、第１の実施形態と同様でよい。図14において、「M_s'[2m-1] || 10*」はsビットの復号された平文ブロックM_s'

[2m-1]を10*詰めパディングした結果を表している。図14では最終ブロックのサイズがs < nの場合を示しているが、s=nの場合はm番目のブロックに対する処理におけるカッティング（図中の「cut_s」）と認証タグ生成処理におけるパディング（図中の「|| 10*」）を省略すればよい。なお、s=nであれば、C_s[2m-1]=C[2m-1]、M_s'[m-1]=M'[m-1]であり、仮にcut_s()やpad_n()を実行しても実質何ら処理が行われないため、cut_s()やpad_n()を省略しなくても特に問題はない。

【０１７８】

本実施形態の復号手段20は、例えば、暗号文が2m-1個のブロックを持つ場合に、補助変数生成手段202が補助系列(N,1,1), (N,1,2), (N,2,1), (N,2,2), … , (N,m,1), (N,m,5), (N,m,6)を生成するとともに、m番目の暗号文チャンクに対して上述した処理を行い復号された平文チャンクMC'[m]=(M_s'[2m-1])を出力する1ラウンドFeistel復号手段と、上述した処理によって復号検証用の認証タグT'を計算する第3の復号検証用タグ計算手段とを備えていてもよい。または、1ラウンドFeistel復号手段と第3の復号検証用タグ計算手段とを備える代わりに、2ラウンドFeistel復号手段203および復号検証用タグ計算手段204が、暗号文のサイズに応じて第1の実施形態の動作と、上述した動作とを切り替えて実行するように構成されていてもよい。なお、補助変数生成手段202、2ラウンドFeistel復号手段203および復号検証用タグ計算手段204が、暗号文のサイズに応じて、第1の実施形態の動作と第2の実施形態の動作と上述した動作とを切り替えて実行するように構成されていれば、ブロック数が偶数の場合、奇数の場合、最終ブロックがnサイズ未満の場合のいずれの場合にも対応できる。

【０１７９】

なお、本実施形態においても、m番目のブロックに対する暗号化処理と復号処理とでは、与えるパラメータが異なるだけで処理内容は同じであるため、第１の実施形態と同様、与えるパラメータを呼び出し元で指定するようにすれば共用できる。また、認証タグの生成に関しても、第１の実施形態と同様、与えるパラメータを呼び出し元で指定するようにすれば共用できる。

【０１８０】

本実施形態においても、最後のチャンクが、補助入力と鍵が決まれば平文チャンクと暗号文チャンクが1対1で対応するように構成されているため、正しく復号できる。また、Feistel構造では個々のラウンドでの処理において暗号化関数F_K自体の逆処理が不要であり、本実施形態においても復号検証用の認証タグを暗号化処理と同様の方法により求めているため、全体としても暗号化関数F_K自体の逆処理が不要である点は、第1の実施形態および第2の実施形態と同様である。

【０１８１】

また、図15は並列処理に対応した暗号化装置100の構成例を示すブロック図である。また、図16は並列処理に対応した復号装置200の構成例を示すブロック図である。図15に示すように、複数の2ラウンドFeistel暗号化手段103を備えることにより、各平文チャンクに対する暗号化処理を2ブロック単位で並列に処理することができる。なお、図15において、一部の2ラウンドFeistel暗号化手段103は、平文のサイズに応じて、第2の2ラウンドFeistel暗号化手段や1ラウンドFeistel暗号化手段として動作する。また、タグ計算手段104は、平文のサイズに応じて、第2のタグ計算手段や第3のタグ計算手段として動作する。また図16に示すように、複数の2ラウンドFeistel復号手段203を備えることにより、各暗号文チャンクに対する復号処理を2ブロック単位で並列に処理することができる。なお、図16において、一部の2ラウンドFeistel復号手段203は、暗号文のサイズに応じて、第2の2ラウンドFeistel復号手段や1ラウンドFeistel復号手段として動作する。また、復号検証用タグ計算手段204は、暗号文のサイズに応じて、第2の復号検証用タグ計算手段や第3の復号検証用タグ計算手段として動作する。

【０１８２】

また、各実施形態においてヘッダ付認証暗号AEADに対応するには、入力にヘッダHを追加した上で、任意の可変長入力疑似ランダム関数g_K'(*)を用意し、生成した認証タグ（T

またはT'）と、上記可変長入力疑似ランダム関数g_K'(H)からの出力との排他的論理和をとったものを最終的な認証タグ（T2またはT2'）とすればよい。すなわち、暗号化処理では、上述したように認証タグTを求めた上で、g_K'(*)を用いて最終的な認証タグT2をT2 = T xor g_K'(H)と求めればよい。そして、暗号化の結果を、(N, C, H, T2)とすればよい。また、復号処理では、上述したように復号検証用の認証タグT'を求めた上で、g_K'(*)を用いて最終的な復号検証用の認証タグT2'= T' xor g_K'(H)を求めて、受信したT2との一致をみればよい。

【０１８３】

ここで、K'は例えば2変数入力の疑似ランダム関数F_K(*,*)の鍵とは独立に選んだ鍵である。この場合、処理全体の鍵は(K, K')のペアになる。

【０１８４】

可変長入力疑似ランダム関数g_K'は、例えばCMAC, HMACなどで実現可能である。CMACやHMACは鍵を用いて任意の入力を固定長の出力に短縮する暗号化関数である。例えば、CMACの場合は、K'を鍵としたブロック関数E_K'を用い、ヘッダHの各ブロック（例えば、H[1], ... , H[h]）に対してCBC_MACと呼ばれる連鎖処理と、終端処理とから構成されている。

【０１８５】

具体例を挙げると、用いるブロック暗号がn-bitブロックの場合、

Y[0] = 00..0,

Y[i] = E_K'(H[1] xor Y[i-1]), for i=1,2, ... , (h-1)

Y[h] = E_K'(H[h] xor Y[i-1] xor 2*E_K'(00..0)), if H[h]がn-bit

Y[h] = E_K'((H[h] || 10*) xor Y[h-1] xor 4*E_K'(00..0)), if H[h]がn-bit未満

といった処理を行う。なお、2*, 4*は有限体上の定数との乗算処理を表している。

【０１８６】

次に、本発明の最小の構成について説明する。図17は本発明による認証暗号装置の最小の構成例を示すブロック図である。図17に示すように、本発明による認証暗号装置は、最小の構成要素として、認証暗号手段60を備える。

【０１８７】

図17に示す認証暗号装置では、認証暗号手段60は、入力された平文または暗号文に対して、2ブロックごとに、補助変数を入れた暗号化関数をラウンド関数に用いた2ラウンドFeistel構造を適用して暗号文または復号された平文を生成する。

【０１８８】

最小構成の認証暗号装置によれば、暗号化関数を一方向にのみ用いて暗号化処理と復号処理になるので、1パスおよび1レート方式の認証暗号であって、並列処理が可能で、かつ１つの暗号化関数のみで全体の暗号化と復号の処理を実行可能な認証暗号を実現できる。

【０１８９】

図18は、認証暗号手段60のより具体的な構成例を示すブロック図であって、図18（a）は認証暗号装置を暗号化装置とする場合の認証暗号手段60の構成例を示し、図18（b）は認証暗号装置を復号装置とする場合の認証暗号手段60の構成例を示している。

【０１９０】

図18（a）に示すように、認証暗号手段60は、暗号化手段61（例えば、暗号化手段10）を含み、暗号化手段61は、平文入力手段611と、補助変数生成手段612と、2ラウンドFeistel暗号化手段613と、タグ計算手段614とを有していてもよい。

【０１９１】

平文入力手段611（例えば、入力手段101）は、暗号化対象の平文と初期ベクトルとを入力する。

【０１９２】

補助変数生成手段612（例えば、補助変数生成手段102）は、初期ベクトルと入力された平文のサイズとに基づき、暗号化関数の各々に与える補助変数を生成する。

【０１９３】

2ラウンドFeistel暗号化手段613（例えば、2ラウンドFeistel暗号化手段103）は、平文

を2ブロックごとのチャンクに分けたときの各平文チャンクに対して2ラウンドFeistel構造を適用することにより、当該平文チャンクに対応する暗号文チャンクを生成する。

【０１９４】

2ラウンドFeistel暗号化手段613は、初期ベクトルをN、チャンクのインデックスをi、i番目の平文チャンクをMC[i] = (M[i_1], M[i_2])、当該平文チャンクMC[i]に含まれる2つの平文ブロックに対応する補助変数を(N,Tw_i_1)と(N,Tw_i_2)の組、暗号化関数をF_K(*,*)とすると、i番目の暗号文チャンクCC[i] = (C[i_1], C[i_2])を、

C[i_1] = F_K((N,Tw_i_1), M[i_1]) xor M[i_2] ,

C[i_2] = F_K((N,Tw_i_2), C[i_1]) xor M[i_1]

と求めてもよい。

【０１９５】

タグ計算手段614（例えば、タグ計算手段104）は、平文のチェックサムを計算し、得られたチェックサムに対して、補助変数を入れた暗号化関数を適用させて認証タグを生成する。

【０１９６】

タグ計算手段614は、平文のチェックサムを、各平文チャンクに含まれる平文ブロックM[i_2]を用いて計算し、得られたチェックサムをSUM、認証タグ生成時に用いる暗号化関数に与える補助変数を(N,Tw_T_1)、暗号化関数をF_K(*,*)とすると、認証タグTを、

T = F_K((N,Tw_T_1), SUM)

と求めてもよい。

【０１９７】

また、図18（b）に示すように、認証暗号手段60は、復号手段62（例えば、復号手段20）を含み、復号手段62は、暗号文入力手段621と、復号用補助変数生成手段622と、2ラウンドFeistel復号手段623と、復号検証用タグ計算手段624と、判定手段625とを有していてもよい。

【０１９８】

暗号文入力手段621（例えば、入力手段201）は、復号対象の暗号文と初期ベクトルと認証タグとを入力する。

【０１９９】

復号用補助変数生成手段622（例えば、補助変数生成手段202）は、初期ベクトルと入力された暗号文のサイズとに基づき、暗号化関数の各々に与える補助変数であって暗号化時と同じ補助変数を生成する。

【０２００】

2ラウンドFeistel復号手段623（例えば、2ラウンドFeistel復号手段203）は、暗号文を2ブロックごとのチャンクに分けたときの各暗号文チャンクに対して2ラウンドFeistel構造を適用することにより、当該暗号文チャンクに対応する、復号された平文チャンクを生成する。

【０２０１】

2ラウンドFeistel復号手段623は、初期ベクトルをN、チャンクのインデックスをi、i番目の暗号文チャンクをCC[i] = (C[i_1], C[i_2])、当該暗号文チャンクCC[i]に含まれる2つの暗号文ブロックに対応する補助変数を(N,Tw_i_1)と(N,Tw_i_2)の組、暗号化関数をF_K(*,*)とすると、i番目の復号された平文チャンクMC'[i] = (M'[i_1], M'[i_2])を、

M'[i_1] = F_K((N,Tw_i_2), C[i_1]) xor C[i_2],

M'[i_2] = F_K((N,Tw_i_1), M'[i_1]) xor C[i_1]

と求めてもよい。

【０２０２】

復号検証用タグ計算手段624（例えば、復号検証用タグ計算手段204）は、復号された平文のチェックサムを計算し、得られたチェックサムに対して、補助変数を入れた暗号化関数を適用させて復号検証用の認証タグを生成する。

【０２０３】

復号検証用タグ計算手段624は、復号された平文のチェックサムを、復号された各平文チャンクに含まれる復号された各平文ブロックM'[i_2]を用いて計算し、得られたチェックサムをSUM'、復号検証用の認証タグ生成時に用いる暗号化関数に与える補助変数を(N,Tw_T_1)、暗号化関数をF_K(*,*)とすると、復号検証用の認証タグT'を、
T' = F_K((N,Tw_T_1), SUM')
と求めてもよい。

【０２０４】

判定手段625（例えば、判定手段205）は、復号検証用タグ計算手段624が生成した復号検証用の認証タグと入力された認証タグとに基づいて、復号の成功または失敗を判定する。

【０２０５】

以上、実施形態及び実施例を参照して本願発明を説明したが、本願発明は上記実施形態に限定されるものではない。本願発明の構成や詳細には、本願発明のスコープ内で当業者が理解し得る様々な変更をすることができる。

【０２０６】

また、上記の実施形態の一部または全部は、以下の付記のようにも記載されうるが、以下には限られない。

【０２０７】

（付記１）入力された平文に対して、2ブロックごとに、補助変数を入れた暗号化関数をラウンド関数に用いた2ラウンドFeistel構造を適用して暗号文を生成する暗号化手段を備えたことを特徴とする暗号化装置。

【０２０８】

（付記２）暗号化手段は、暗号化対象の平文と初期ベクトルとを入力する平文入力手段と、初期ベクトルと入力された平文のサイズとに基づき、暗号化関数の各々に与える補助変数を生成する補助変数生成手段と、平文を2ブロックごとのチャンクに分けたときの各平文チャンクに対して2ラウンドFeistel構造を適用することにより、当該平文チャンクに対応する暗号文チャンクを生成する2ラウンドFeistel暗号化手段と、平文のチェックサムを計算し、得られたチェックサムに対して、補助変数を入れた暗号化関数を適用させて認証タグを生成するタグ計算手段とを有し、2ラウンドFeistel暗号化手段は、初期ベクトルをN、チャンクのインデックスをi、i番目の平文チャンクをMC[i] = (M[i_1], M[i_2])、当該平文チャンクMC[i]に含まれる2つの平文ブロックに対応する補助変数を(N,Tw_i_1)と(N,Tw_i_2)の組、暗号化関数をF_K(*,*)とすると、i番目の暗号文チャンクCC[i] = (C[i_1], C[i_2])を、
C[i_1] = F_K((N,Tw_i_1), M[i_1]) xor M[i_2] ,
C[i_2] = F_K((N,Tw_i_2), C[i_1]) xor M[i_1]
と求め、タグ計算手段は、平文のチェックサムを、各平文チャンクに含まれる平文ブロックM[i_2]を用いて計算し、得られたチェックサムをSUM、認証タグ生成時に用いる暗号化関数に与える補助変数を(N,Tw_T_1)、暗号化関数をF_K(*,*)とすると、認証タグTを、
T = F_K((N,Tw_T_1), SUM)
と求める付記１に記載の暗号化装置。

【０２０９】

（付記３）補助変数生成手段は、入力された平文のサイズが偶数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズに満たない場合に、認証タグ生成時に用いる暗号化関数に与える補助変数として、第2の認証タグ用補助変数を生成し、暗号化手段は、最終の平文ブロックを含む最終の平文チャンクに対して、所定の2ラウンドFeistel構造を適用して、最終の暗号文ブロックを含む最終の暗号文チャンクを生成する第2の2ラウンドFeistel暗号化手段と、平文のチェックサムを、入力された平文と、第2の2ラウンドFeistel暗号化手段からの出力とを用いて計算し、得られたチェックサムに対して補助変数を与えた暗号化関数を適用させて、認証タグを生成する第2のタグ計算手段とを有し、第2の2ラウンドFeistel暗号化手段は、最終の平文チャンクのインデッ

クスをm、最終の平文ブロックをM[m_2]、最終の平文チャンクをMC[m] = (M[m_1], M[m_2])、最終の平文チャンクMC[m]に含まれる2つの平文ブロックに対応する補助変数を(N,Tw_m_1)と(N,Tw_m_2)の組、暗号化関数をF_K(*,*)、最終の平文ブロックのサイズをs、ブロックサイズをn、sサイズからnサイズへのパディング処理をpad_n()、nサイズからsサイズへのカッティング処理をcut_s()とすると、sサイズの最終の暗号文ブロックC[m_2]を含む最終の暗号文チャンクCC[m] = (C[m_1], C[m_2])を、
C[m_2] = cut_s(Z) xor M[m_2],
C[m_1] = F_K((N,Tw_m_2), pad_n(C[m_2])) xor M[m_1]
ただし、Z = F_K((N,Tw_m_1), M[m_1])
と求め、第2のタグ計算手段は、平文のチェックサムを、最終の平文チャンクを除く各平文チャンクに含まれる平文ブロックM[i_2]とZとC[m_2]をnサイズにパディングした結果であるC_n[m_2]とを用いて計算し、得られたチェックサムをSUM、第2の認証タグ用補助変数を(N,Tw_T_2)、暗号化関数をF_K(*,*)とすると、認証タグTを、
T = F_K((N,Tw_T_2), SUM)
と求める付記２に記載の暗号化装置。

【０２１０】

（付記４）補助変数生成手段は、入力された平文のサイズが奇数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズと同じである場合に、認証タグ生成時に用いる暗号化関数に与える補助変数として、第3の認証タグ用補助変数を生成し、入力された平文のサイズが奇数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズに満たない場合に、認証タグ生成時に用いる暗号化関数に与える補助変数として、第4の認証タグ用補助変数を生成し、暗号化手段は、最終の平文ブロックを含む最終の平文チャンクに対して、所定の1ラウンドFeistel構造を適用して、最終の暗号文ブロックを含む最終の暗号文チャンクを生成する1ラウンドFeistel暗号化手段と、平文のチェックサムを、入力された平文と、1ラウンドFeistel暗号化手段からの出力とを用いて計算し、得られたチェックサムに対して補助変数を与えた暗号化関数を適用させて、認証タグを生成する第3のタグ計算手段とを有し、1ラウンドFeistel暗号化手段は、最終の平文チャンクのインデックスをm、最終の平文ブロックをM[m_1]、最終の平文チャンクをMC[m] = (M[m_1])、最終の平文ブロックに対応する補助変数を(N,Tw_m_1)、暗号化関数をF_K(*,*)、最終の平文ブロックのサイズをs、ブロックサイズをn、nサイズからsサイズへのカッティング処理をcut_s()とすると、sサイズの最終の暗号文ブロックC[m_1]を含む最終の暗号文チャンクCC[m] = (C[m_1])を、
C[m_1] = cut_s(F_K((N,Tw_m_1),0^n)) xor M[m_1]
ただし、s=nのときはcut_s()は省略可能
と求め、第3のタグ計算手段は、もしs=nであれば、平文のチェックサムを、最終の平文チャンクを除く各平文チャンクに含まれる平文ブロックM[i_2]とC[m_1]とを用いて計算し、得られたチェックサムをSUM、第3の認証タグ用補助変数を(N,Tw_T_3)、暗号化関数をF_K(*,*)とすると、認証タグTを、
T = F_K((N,Tw_T_3), SUM)
と求め、もしs<nであれば、平文のチェックサムを、最終の平文チャンクを除く各平文チャンクに含まれる平文ブロックM[i_2]とC[m_1]をnサイズにパディングした結果であるC_n[m_1]とを用いて計算し、得られたチェックサムをSUM、第4の認証タグ用補助変数を(N,Tw_T_4)、暗号化関数をF_K(*,*)とすると、認証タグTを、
T = F_K((N,Tw_T_4), SUM)
と求める付記２または付記３に記載の暗号化装置。

【０２１１】

（付記５）暗号化関数が、Tweakと呼ばれる補助変数を含む2変数入力のTweakable ブロック暗号である付記１から付記４のうちのいずれかに記載の暗号化装置。

【０２１２】

（付記６）暗号化関数が、入力される第1の変数と第2の変数とを連結したものを入力と

する、鍵付きハッシュ関数である付記１から付記４のうちのいずれかに記載の暗号化装置。

【０２１３】

（付記７）入力された暗号文に対して、2ブロックごとに、補助変数を入れた暗号化関数をラウンド関数に用いた2ラウンドFeistel構造を適用して復号された平文を生成する復号手段を備えたことを特徴とする復号装置。

【０２１４】

（付記８）復号手段は、復号対象の暗号文と初期ベクトルと認証タグとを入力する暗号文入力手段と、初期ベクトルと入力された暗号文のサイズとに基づき、暗号化関数の各々に与える補助変数であって暗号化時と同じ補助変数を生成する復号用補助変数生成手段と、暗号文を2ブロックごとのチャンクに分けたときの各暗号文チャンクに対して2ラウンドFeistel構造を適用することにより、当該暗号文チャンクに対応する、復号された平文チャンクを生成する2ラウンドFeistel復号手段と、復号された平文のチェックサムを計算し、得られたチェックサムに対して、補助変数を入れた暗号化関数を適用させて復号検証用の認証タグを生成する復号検証用タグ計算手段と、復号検証用タグ計算手段が生成した復号検証用の認証タグと入力された認証タグとに基づいて、復号の成功または失敗を判定する判定手段とを有し、2ラウンドFeistel復号手段は、初期ベクトルをN、チャンクのインデックスをi、i番目の暗号文チャンクをCC[i] = (C[i_1], C[i_2])、当該暗号文チャンクCC[i]に含まれる2つの暗号文ブロックに対応する補助変数を(N,Tw_i_1)と(N,Tw_i_2)の組、暗号化関数をF_K(*,*)とすると、i番目の復号された平文チャンクMC'[i] = (M'[i_1], M'[i_2])を、

M'[i_1] = F_K((N,Tw_i_2), C[i_1]) xor C[i_2],

M'[i_2] = F_K((N,Tw_i_1), M'[i_1]) xor C[i_1]

と求め、復号検証用タグ計算手段は、復号された平文のチェックサムを、復号された各平文チャンクに含まれる復号された各平文ブロックM'[i_2]を用いて計算し、得られたチェックサムをSUM'、復号検証用の認証タグ生成時に用いる暗号化関数に与える補助変数を(N,Tw_T_1)、暗号化関数をF_K(*,*)とすると、復号検証用の認証タグT'を、

T' = F_K((N,Tw_T_1), SUM')

と求める付記７に記載の復号装置。

【０２１５】

（付記９）復号用補助変数生成手段は、入力された暗号文のサイズが偶数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズに満たない場合に、復号検証用の認証タグ生成時に用いる暗号化関数に与える補助変数として、暗号時と同じ第2の認証タグ用補助変数を生成し、復号手段は、最終の暗号文ブロックを含む最終の暗号文チャンクに対して、所定の2ラウンドFeistel構造を適用して、最終の暗号ブロックの復号された平文ブロックを含む最終の復号された平文チャンクを生成する第2の2ラウンドFeistel復号手段と、復号された平文のチェックサムを、2ラウンドFeistel復号手段からの出力と、第2の2ラウンドFeistel復号手段からの出力と、最終の暗号文ブロックとを用いて計算し、得られたチェックサムに対して補助変数を与えた暗号化関数を適用させて、復号検証用の認証タグを生成する第2の復号検証用タグ計算手段とを有し、第2の2ラウンドFeistel復号手段は、最終の暗号文チャンクのインデックスをm、最終の暗号文ブロックをC[m_2]、最終の暗号文チャンクをCC[m] = (C[m_1], C[m_2])、最終の暗号文チャンクCC[m]に含まれる2つの暗号文ブロックに対応する補助変数を(N,Tw_m_1)と(N,Tw_m_2)の組、暗号化関数をF_K(*,*)、最終の暗号文ブロックのサイズをs、ブロックサイズをn、sサイズからnサイズへのパディング処理をpad_n()、nサイズからsサイズへのカッティング処理をcut_s()とすると、sサイズの最終の復号された平文ブロックM'[m_2]を含む最終の復号された平文チャンクMC'[m] = (M'[m_1], M'[m_2])を、

M'[m_1] = F_K((N,Tw_m_2), pad_n(C[m_2])) xor C[m_1],

M'[m_2] = cut_s(Z') xor C[m_2],

ただし、Z' = F_K((N,Tw_m_1), M'[m_1])

と求め、第2の復号検証用タグ計算手段は、復号された平文のチェックサムを、最終の復号された平文チャンクを除く復号された各平文チャンクに含まれる復号された平文ブロックM[i_2]とZ'とM'[m_2]をnサイズにパディングした結果であるM_n'[m_2]とを用いて計算し、得られたチェックサムをSUM'、第2の認証タグ用補助変数を(N,Tw_T_2)、暗号化関数をF_K(*,*)とすると、復号検証用の認証タグT'を、

T' = F_K((N,Tw_T_2), SUM')

と求める付記８に記載の復号装置。

【０２１６】

（付記１０）補助変数生成手段は、入力された暗号文のサイズが奇数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズと同じである場合に、復号検証用の認証タグ生成時に用いる暗号化関数に与える補助変数として、暗号時と同じ第3の認証タグ用補助変数を生成し、入力された暗号文のサイズが奇数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズに満たない場合に、認証タグ生成時に用いる暗号化関数に与える補助変数として、暗号時と同じ第4の認証タグ用補助変数を生成し、復号手段は、最終の暗号文ブロックを含む最終の暗号文チャンクに対して、所定の1ラウンドFeistel構造を適用して、最終の復号された平文ブロックを含む最終の平文チャンクを生成する1ラウンドFeistel復号手段と、復号された平文のチェックサムを、2ラウンドFeistel復号手段からの出力と、1ラウンドFeistel復号手段からの出力とを用いて計算し、得られたチェックサムに対して補助変数を与えた暗号化関数を適用させて、復号検証用の認証タグを生成する第3の復号検証用タグ計算手段とを有し、1ラウンドFeistel復号手段は、最終の暗号文チャンクのインデックスをm、最終の暗号文ブロックをC[m_1]、最終の暗号文チャンクをCC[m] = (C[m_1])、最終の暗号文ブロックに対応する補助変数を(N,Tw_m_1)、暗号化関数をF_K(*,*)、最終の暗号文ブロックのサイズをs、ブロックサイズをn、nサイズからsサイズへのカッティング処理をcut_s()とすると、sサイズの最終の復号された平文ブロックM'[m_1]を含む最終の復号された平文チャンクMC'[m] = (M'[m_1])を、

M'[m_1] = cut_s(F_K((N,Tw_m_1),0^n)) xor C[m_1]

ただし、s=nのときはcut_s()は省略可能

と求め、第3の復号検証用タグ計算手段は、もしs=nであれば、復号された平文のチェックサムを、最終の復号された平文チャンクを除く復号された各平文チャンクに含まれる平文ブロックM'[i_2]とM'[m_1]とを用いて計算し、得られたチェックサムをSUM'、第3の認証タグ用補助変数を(N,Tw_T_3)、暗号化関数をF_K(*,*)とすると、復号検証用の認証タグT'を、

T' = F_K((N,Tw_T_3), SUM')

と求め、もしs<nであれば、復号された平文のチェックサムを、最終の復号された平文チャンクを除く復号された各平文チャンクに含まれる平文ブロックM'[i_2]とM'[m_1]をnサイズにパディングした結果であるM_n'[m_1]とを用いて計算し、得られたチェックサムをSUM'、第4の認証タグ用補助変数を(N,Tw_T_4)、暗号化関数をF_K(*,*)とすると、復号検証用の認証タグT'を、

T' = F_K((N,Tw_T_4), SUM')

と求める付記８または付記９に記載の復号装置。

【０２１７】

（付記１１）暗号化関数が、Tweakと呼ばれる補助変数を含む2変数入力のTweakable ブロック暗号である付記７から付記１０のうちのいずれかに記載の復号装置。

【０２１８】

（付記１２）暗号化関数が、入力される第1の変数と第2の変数とを連結したものを入力とする、鍵付きハッシュ関数である付記７から付記１０のうちのいずれかに記載の復号装置。

【０２１９】

（付記１３）情報処理装置が、暗号化対象の平文と初期ベクトルとを入力し、初期ベク

トルと入力された平文のサイズとに基づき、暗号化関数の各々に与える補助変数を生成し、平文を2ブロックごとのチャンクに分けたときの各平文チャンクに対して2ラウンドFeistel構造を適用することにより、当該平文チャンクに対応する暗号文チャンクを生成する2ラウンドFeistel暗号化処理において、初期ベクトルをN、チャンクのインデックスをi、i番目の平文チャンクをMC[i] = (M[i_1], M[i_2])、当該平文チャンクMC[i]に含まれる2つの平文ブロックに対応する補助変数を(N,Tw_i_1)と(N,Tw_i_2)の組、暗号化関数をF_K(*,*)とすると、i番目の暗号文チャンクCC[i] = (C[i_1], C[i_2])を、
C[i_1] = F_K((N,Tw_i_1), M[i_1]) xor M[i_2] ,
C[i_2] = F_K((N,Tw_i_2), C[i_1]) xor M[i_1]
と求め、平文のチェックサムを、各平文チャンクに含まれる平文ブロックM[i_2]を用いて計算し、得られたチェックサムをSUM、認証タグ生成時に用いる暗号化関数に与える補助変数を(N,Tw_T_1)、暗号化関数をF_K(*,*)とすると、認証タグTを、
T = F_K((N,Tw_T_1), SUM)
と求めることを特徴とする認証暗号方法。

【０２２０】

（付記１４）情報処理装置が、入力された平文のサイズが偶数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズに満たない場合に、認証タグ生成時に用いる暗号化関数に与える補助変数として、第2の認証タグ用補助変数を生成し、最終の平文ブロックを含む最終の平文チャンクに対して、所定の2ラウンドFeistel構造を適用して、最終の暗号文ブロックを含む最終の暗号文チャンクを生成する第2の2ラウンドFeistel暗号化処理において、最終の平文チャンクのインデックスをm、最終の平文ブロックをM[m_2]、最終の平文チャンクをMC[m] = (M[m_1], M[m_2])、最終の平文チャンクMC[m]に含まれる2つの平文ブロックに対応する補助変数を(N,Tw_m_1)と(N,Tw_m_2)の組、暗号化関数をF_K(*,*)、最終の平文ブロックのサイズをs、ブロックサイズをn、sサイズからnサイズへのパディング処理をpad_n()、nサイズからsサイズへのカッティング処理をcut_s()とすると、sサイズの最終の暗号文ブロックC[m_2]を含む最終の暗号文チャンクCC[m] = (C[m_1], C[m_2])を、
C[m_2] = cut_s(Z) xor M[m_2],
C[m_1] = F_K((N,Tw_m_2), pad_n(C[m_2])) xor M[m_1]
ただし、Z = F_K((N,Tw_m_1), M[m_1])
と求め、平文のチェックサムを、最終の平文チャンクを除く各平文チャンクに含まれる平文ブロックM[i_2]とZとC[m_2]をnサイズにパディングした結果であるC_n[m_2]とを用いて計算し、得られたチェックサムをSUM、第2の認証タグ用補助変数を(N,Tw_T_2)、暗号化関数をF_K(*,*)とすると、認証タグTを、
T = F_K((N,Tw_T_2), SUM)
と求める付記１３に記載の認証暗号方法。

【０２２１】

（付記１５）入力された平文のサイズが奇数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズと同じである場合に、認証タグ生成時に用いる暗号化関数に与える補助変数として、第3の認証タグ用補助変数を生成し、入力された平文のサイズが奇数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズに満たない場合に、認証タグ生成時に用いる暗号化関数に与える補助変数として、第4の認証タグ用補助変数を生成し、最終の平文ブロックを含む最終の平文チャンクに対して、所定の1ラウンドFeistel構造を適用して、最終の暗号文ブロックを含む最終の暗号文チャンクを生成する1ラウンドFeistel暗号化処理において、最終の平文チャンクのインデックスをm、最終の平文ブロックをM[m_1]、最終の平文チャンクをMC[m] = (M[m_1])、最終の平文ブロックに対応する補助変数を(N,Tw_m_1)、暗号化関数をF_K(*,*)、最終の平文ブロックのサイズをs、ブロックサイズをn、nサイズからsサイズへのカッティング処理をcut_s()とすると、sサイズの最終の暗号文ブロックC[m_1]を含む最終の暗号文チャンクCC[m] = (C[m_1])を、

C[m_1] = cut_s(F_K((N,Tw_m_1),0^n)) xor M[m_1]
ただし、s=nのときはcut_s()は省略可能
と求め、もしs=nであれば、平文のチェックサムを、最終の平文チャンクを除く各平文チャンクに含まれる平文ブロックM[i_2]とC[m_1]とを用いて計算し、得られたチェックサムをSUM、第3の認証タグ用補助変数を(N,Tw_T_3)、暗号化関数をF_K(*,*)とすると、認証タグTを、
T = F_K((N,Tw_T_3), SUM)
と求め、もしs<nであれば、平文のチェックサムを、最終の平文チャンクを除く各平文チャンクに含まれる平文ブロックM[i_2]とC[m_1]をnサイズにパディングした結果であるC_n[m_1]とを用いて計算し、得られたチェックサムをSUM、第4の認証タグ用補助変数を(N,Tw_T_4)、暗号化関数をF_K(*,*)とすると、認証タグTを、
T = F_K((N,Tw_T_4), SUM)
と求める付記１３または付記１４に記載の認証暗号方法。

【０２２２】

（付記１６）情報処理装置が、復号対象の暗号文と初期ベクトルと認証タグとを入力し、初期ベクトルと入力された暗号文のサイズとに基づき、暗号化関数の各々に与える補助変数であって暗号化時と同じ補助変数を生成し、暗号文を2ブロックごとのチャンクに分けたときの各暗号文チャンクに対して2ラウンドFeistel構造を適用することにより、当該暗号文チャンクに対応する、復号された平文チャンクを生成する2ラウンドFeistel復号処理において、初期ベクトルをN、チャンクのインデックスをi、i番目の暗号文チャンクをCC[i] = (C[i_1], C[i_2])、当該暗号文チャンクCC[i]に含まれる2つの暗号文ブロックに対応する補助変数を(N,Tw_i_1)と(N,Tw_i_2)の組、暗号化関数をF_K(*,*)とすると、i番目の復号された平文チャンクMC'[i] = (M'[i_1], M'[i_2])を、
M'[i_1] = F_K((N,Tw_i_2), C[i_1]) xor C[i_2],
M'[i_2] = F_K((N,Tw_i_1), M'[i_1]) xor C[i_1]
と求め、復号された平文のチェックサムを、復号された各平文チャンクに含まれる復号された各平文ブロックM'[i_2]を用いて計算し、得られたチェックサムをSUM'、復号検証用の認証タグ生成時に用いる暗号化関数に与える補助変数を(N,Tw_T_1)、暗号化関数をF_K(*,*)とすると、復号検証用の認証タグT'を、
T' = F_K((N,Tw_T_1), SUM')
と求めることを特徴とする認証暗号方法。

【０２２３】

（付記１７）入力された暗号文のサイズが偶数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズに満たない場合に、復号検証用の認証タグ生成時に用いる暗号化関数に与える補助変数として、暗号時と同じ第2の認証タグ用補助変数を生成し、最終の暗号文ブロックを含む最終の暗号文チャンクに対して、所定の2ラウンドFeistel構造を適用して、最終の暗号ブロックの復号された平文ブロックを含む最終の復号された平文チャンクを生成する第2の2ラウンドFeistel復号処理において、最終の暗号文チャンクのインデックスをm、最終の暗号文ブロックをC[m_2]、最終の暗号文チャンクをCC[m] = (C[m_1], C[m_2])、最終の暗号文チャンクCC[m]に含まれる2つの暗号文ブロックに対応する補助変数を(N,Tw_m_1)と(N,Tw_m_2)の組、暗号化関数をF_K(*,*)、最終の暗号文ブロックのサイズをs、ブロックサイズをn、sサイズからnサイズへのパディング処理をpad_n()、nサイズからsサイズへのカッティング処理をcut_s()とすると、sサイズの最終の復号された平文ブロックM'[m_2]を含む最終の復号された平文チャンクMC'[m] = (M'[m_1], M'[m_2])を、
M'[m_1] = F_K((N,Tw_m_2), pad_n(C[m_2])) xor C[m_1],
M'[m_2] = cut_s(Z') xor C[m_2],
ただし、Z' = F_K((N,Tw_m_1), M'[m_1])
と求め、復号された平文のチェックサムを、最終の復号された平文チャンクを除く復号された各平文チャンクに含まれる復号された平文ブロックM[i_2]とZ'とM'[m_2]をnサイズに

パディングした結果であるM_n'[m_2]とを用いて計算し、得られたチェックサムをSUM'、第2の認証タグ用補助変数を(N,Tw_T_2)、暗号化関数をF_K(*,*)とすると、復号検証用の認証タグT'を、
T' = F_K((N,Tw_T_2), SUM')
と求める付記１６に記載の認証暗号方法。

【０２２４】

（付記１８）入力された暗号文のサイズが奇数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズと同じである場合に、復号検証用の認証タグ生成時に用いる暗号化関数に与える補助変数として、暗号時と同じ第3の認証タグ用補助変数を生成し、入力された暗号文のサイズが奇数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズに満たない場合に、認証タグ生成時に用いる暗号化関数に与える補助変数として、暗号時と同じ第4の認証タグ用補助変数を生成し、最終の暗号文ブロックを含む最終の暗号文チャンクに対して、所定の1ラウンドFeistel構造を適用して、最終の復号された平文ブロックを含む最終の平文チャンクを生成する1ラウンドFeistel復号処理において、最終の暗号文チャンクのインデックスをm、最終の暗号文ブロックをC[m_1]、最終の暗号文チャンクをCC[m] = (C[m_1])、最終の暗号文ブロックに対応する補助変数を(N,Tw_m_1)、暗号化関数をF_K(*,*)、最終の暗号文ブロックのサイズをs、ブロックサイズをn、nサイズからsサイズへのカッティング処理をcut_s()とすると、sサイズの最終の復号された平文ブロックM'[m_1]を含む最終の復号された平文チャンクMC'[m] = (M'[m_1])を、
M'[m_1] = cut_s(F_K((N,Tw_m_1),0^n)) xor C[m_1]
ただし、s=nのときはcut_s()は省略可能
と求め、もしs=nであれば、復号された平文のチェックサムを、最終の復号された平文チャンクを除く復号された各平文チャンクに含まれる平文ブロックM'[i_2]とM'[m_1]とを用いて計算し、得られたチェックサムをSUM'、第3の認証タグ用補助変数を(N,Tw_T_3)、暗号化関数をF_K(*,*)とすると、復号検証用の認証タグT'を、
T' = F_K((N,Tw_T_3), SUM')
と求め、もしs<nであれば、復号された平文のチェックサムを、最終の復号された平文チャンクを除く復号された各平文チャンクに含まれる平文ブロックM'[i_2]とM'[m_1]をnサイズにパディングした結果であるM_n'[m_1]とを用いて計算し、得られたチェックサムをSUM'、第4の認証タグ用補助変数を(N,Tw_T_4)、暗号化関数をF_K(*,*)とすると、復号検証用の認証タグT'を、
T' = F_K((N,Tw_T_4), SUM')
と求める付記１６または付記１７に記載の認証暗号方法。

【０２２５】

（付記１９）暗号化関数が、Tweakと呼ばれる補助変数を含む2変数入力のTweakable ブロック暗号である付記１３から付記１８のうちのいずれかに記載の認証暗号方法。

【０２２６】

（付記２０）暗号化関数が、入力される第1の変数と第2の変数とを連結したものを入力とする、鍵付きハッシュ関数である付記１３から付記１８のうちのいずれかに記載の認証暗号方法。

【０２２７】

（付記２１）コンピュータに、暗号化対象の平文と初期ベクトルとを入力する平文入力処理、初期ベクトルと入力された平文のサイズとに基づき、暗号化関数の各々に与える補助変数を生成する補助変数生成処理、平文を2ブロックごとのチャンクに分けたときの各平文チャンクに対して2ラウンドFeistel構造を適用することにより、当該平文チャンクに対応する暗号文チャンクを生成する2ラウンドFeistel暗号化処理、および、平文のチェックサムを計算し、得られたチェックサムに対して、補助変数を入れた暗号化関数を適用させて認証タグを生成するタグ計算処理を実行させ、2ラウンドFeistel暗号化処理で、初期ベクトルをN、チャンクのインデックスをi、i番目の平文チャンクをMC[i] = (M[i_1], M[

i_2])、当該平文チャンクMC[i]に含まれる2つの平文ブロックに対応する補助変数を(N,Tw_i_1)と(N,Tw_i_2)の組、暗号化関数をF_K(*,*)とすると、i番目の暗号文チャンクCC[i] = (C[i_1], C[i_2])を、
C[i_1] = F_K((N,Tw_i_1), M[i_1]) xor M[i_2] ,
C[i_2] = F_K((N,Tw_i_2), C[i_1]) xor M[i_1]
と求めさせ、タグ計算処理で、平文のチェックサムを、各平文チャンクに含まれる平文ブロックM[i_2]を用いて計算し、得られたチェックサムをSUM、認証タグ生成時に用いる暗号化関数に与える補助変数を(N,Tw_T_1)、暗号化関数をF_K(*,*)とすると、認証タグTを、
T = F_K((N,Tw_T_1), SUM)
と求めさせることを特徴とする認証暗号用プログラム。

【０２２８】

（付記２２）コンピュータに、補助変数生成処理で、入力された平文のサイズが偶数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズに満たない場合に、認証タグ生成時に用いる暗号化関数に与える補助変数として、第2の認証タグ用補助変数を生成させ、2ラウンドFeistel暗号化処理で、最終の平文チャンクのインデックスをm、最終の平文ブロックをM[m_2]、最終の平文チャンクをMC[m] = (M[m_1], M[m_2])、最終の平文チャンクMC[m]に含まれる2つの平文ブロックに対応する補助変数を(N,Tw_m_1)と(N,Tw_m_2)の組、暗号化関数をF_K(*,*)、最終の平文ブロックのサイズをs、ブロックサイズをn、sサイズからnサイズへのパディング処理をpad_n()、nサイズからsサイズへのカッティング処理をcut_s()とすると、sサイズの最終の暗号文ブロックC[m_2]を含む最終の暗号文チャンクCC[m] = (C[m_1], C[m_2])を、
C[m_2] = cut_s(Z) xor M[m_2],
C[m_1] = F_K((N,Tw_m_2), pad_n(C[m_2])) xor M[m_1]
ただし、Z = F_K((N,Tw_m_1), M[m_1])
と求めさせ、第2のタグ計算処理で、平文のチェックサムを、最終の平文チャンクを除く各平文チャンクに含まれる平文ブロックM[i_2]とZとC[m_2]をnサイズにパディングした結果であるC_n[m_2]とを用いて計算し、得られたチェックサムをSUM、第2の認証タグ用補助変数を(N,Tw_T_2)、暗号化関数をF_K(*,*)とすると、認証タグTを、
T = F_K((N,Tw_T_2), SUM)
と求めさせる付記２１に記載の認証暗号用プログラム。

【０２２９】

（付記２３）コンピュータに、補助変数生成処理で、入力された平文のサイズが奇数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズと同じである場合に、認証タグ生成時に用いる暗号化関数に与える補助変数として、第3の認証タグ用補助変数を生成させ、入力された平文のサイズが奇数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズに満たない場合に、認証タグ生成時に用いる暗号化関数に与える補助変数として、第4の認証タグ用補助変数を生成させ、2ラウンドFeistel暗号化処理で、最終の平文チャンクのインデックスをm、最終の平文ブロックをM[m_1]、最終の平文チャンクをMC[m] = (M[m_1])、最終の平文ブロックに対応する補助変数を(N,Tw_m_1)、暗号化関数をF_K(*,*)、最終の平文ブロックのサイズをs、ブロックサイズをn、nサイズからsサイズへのカッティング処理をcut_s()とすると、sサイズの最終の暗号文ブロックC[m_1]を含む最終の暗号文チャンクCC[m] = (C[m_1])を、
C[m_1] = cut_s(F_K((N,Tw_m_1),0^n)) xor M[m_1]
ただし、s=nのときはcut_s()は省略可能
と求めさせ、タグ計算処理で、もしs=nであれば、平文のチェックサムを、最終の平文チャンクを除く各平文チャンクに含まれる平文ブロックM[i_2]とC[m_1]とを用いて計算し、得られたチェックサムをSUM、第3の認証タグ用補助変数を(N,Tw_T_3)、暗号化関数をF_K(*,*)とすると、認証タグTを、
T = F_K((N,Tw_T_3), SUM)

と求めさせ、もしs<nであれば、平文のチェックサムを、最終の平文チャンクを除く各平文チャンクに含まれる平文ブロックM[i_2]とC[m_1]をnサイズにパディングした結果であるC_n[m_1]とを用いて計算し、得られたチェックサムをSUM、第4の認証タグ用補助変数を(N,Tw_T_4)、暗号化関数をF_K(*,*)とすると、認証タグTを、
T = F_K((N,Tw_T_4), SUM)
と求めさせる付記２１または付記２２に記載の認証暗号用プログラム。

【０２３０】

（付記２４）コンピュータに、復号対象の暗号文と初期ベクトルと認証タグとを入力する暗号文入力処理、初期ベクトルと入力された暗号文のサイズとに基づき、暗号化関数の各々に与える補助変数であって暗号化時と同じ補助変数を生成する復号用補助変数生成処理、暗号文を2ブロックごとのチャンクに分けたときの各暗号文チャンクに対して2ラウンドFeistel構造を適用することにより、当該暗号文チャンクに対応する、復号された平文チャンクを生成する2ラウンドFeistel復号処理、復号された平文のチェックサムを計算し、得られたチェックサムに対して、補助変数を入れた暗号化関数を適用させて復号検証用の認証タグを生成する復号検証用タグ計算処理、および、復号検証用タグ計算処理で生成された復号検証用の認証タグと入力された認証タグとに基づいて、復号の成功または失敗を判定する判定処理を実行させ、2ラウンドFeistel復号処理で、初期ベクトルをN、チャンクのインデックスをi、i番目の暗号文チャンクをCC[i] = (C[i_1], C[i_2])、当該暗号文チャンクCC[i]に含まれる2つの暗号文ブロックに対応する補助変数を(N,Tw_i_1)と(N,Tw_i_2)の組、暗号化関数をF_K(*,*)とすると、i番目の復号された平文チャンクMC'[i] = (M'[i_1], M'[i_2])を、
M'[i_1] = F_K((N,Tw_i_2), C[i_1]) xor C[i_2],
M'[i_2] = F_K((N,Tw_i_1), M'[i_1]) xor C[i_1]
と求めさせ、復号検証用タグ計算処理で、復号された平文のチェックサムを、復号された各平文チャンクに含まれる復号された各平文ブロックM'[i_2]を用いて計算し、得られたチェックサムをSUM'、復号検証用の認証タグ生成時に用いる暗号化関数に与える補助変数を(N,Tw_T_1)、暗号化関数をF_K(*,*)とすると、復号検証用の認証タグT'を、
T' = F_K((N,Tw_T_1), SUM')
と求めさせることを特徴とする認証暗号用プログラム。

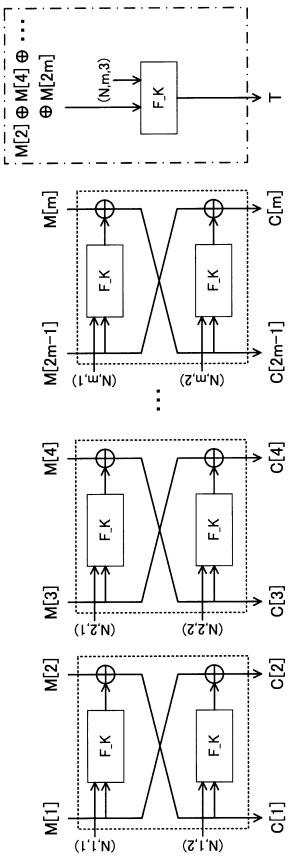
【０２３１】

（付記２５）コンピュータに、復号用補助変数生成処理で、入力された暗号文のサイズが偶数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズに満たない場合に、復号検証用の認証タグ生成時に用いる暗号化関数に与える補助変数として、暗号時と同じ第2の認証タグ用補助変数を生成させ、2ラウンドFeistel復号処理で、最終の暗号文チャンクのインデックスをm、最終の暗号文ブロックをC[m_2]、最終の暗号文チャンクをCC[m] = (C[m_1], C[m_2])、最終の暗号文チャンクCC[m]に含まれる2つの暗号文ブロックに対応する補助変数を(N,Tw_m_1)と(N,Tw_m_2)の組、暗号化関数をF_K(*,*)、最終の暗号文ブロックのサイズをs、ブロックサイズをn、sサイズからnサイズへのパディング処理をpad_n()、nサイズからsサイズへのカッティング処理をcut_s()とすると、sサイズの最終の復号された平文ブロックM'[m_2]を含む最終の復号された平文チャンクMC'[m] = (M'[m_1], M'[m_2])を、
M'[m_1] = F_K((N,Tw_m_2), pad_n(C[m_2])) xor C[m_1],
M'[m_2] = cut_s(Z') xor C[m_2],
ただし、Z' = F_K((N,Tw_m_1), M'[m_1])
と求めさせ、復号検証用タグ計算処理で、復号された平文のチェックサムを、最終の復号された平文チャンクを除く復号された各平文チャンクに含まれる復号された平文ブロックM[i_2]とZ'とM'[m_2]をnサイズにパディングした結果であるM_n'[m_2]とを用いて計算し、得られたチェックサムをSUM'、第2の認証タグ用補助変数を(N,Tw_T_2)、暗号化関数をF_K(*,*)とすると、復号検証用の認証タグT'を、
T' = F_K((N,Tw_T_2), SUM')

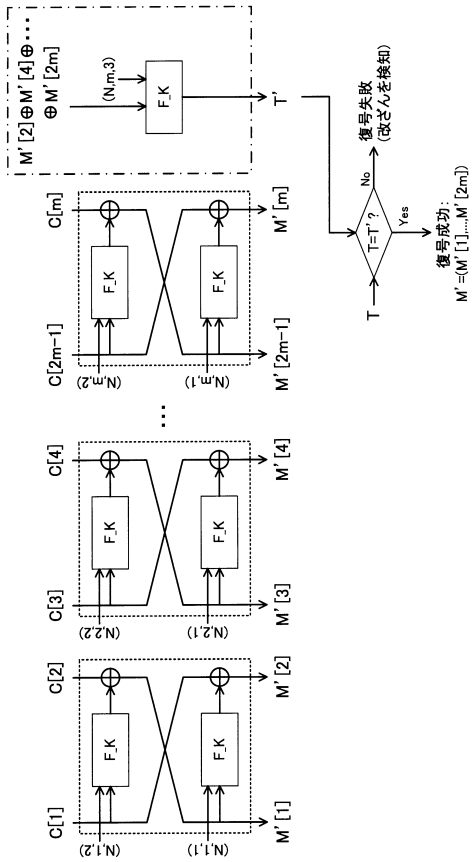
と求めさせる付記２４に記載の認証暗号用プログラム。
【０２３２】
（付記２６）コンピュータに、補助変数生成処理で、入力された暗号文のサイズが奇数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズと同じである場合に、復号検証用の認証タグ生成時に用いる暗号化関数に与える補助変数として、暗号時と同じ第3の認証タグ用補助変数を生成させ、入力された暗号文のサイズが奇数のブロックに分割されるサイズであり、かつ最終のブロックが所定のブロックサイズに満たない場合に、認証タグ生成時に用いる暗号化関数に与える補助変数として、暗号時と同じ第4の認証タグ用補助変数を生成させ、2ラウンドFeistel復号処理で、最終の暗号文チャンクのインデックスをm、最終の暗号文ブロックをC[m_1]、最終の暗号文チャンクをC C[m] = (C[m_1])、最終の暗号文ブロックに対応する補助変数を(N,Tw_m_1)、暗号化関数をF_K(*,*)、最終の暗号文ブロックのサイズをs、ブロックサイズをn、nサイズからsサイズへのカッティング処理をcut_s()とすると、sサイズの最終の復号された平文ブロックM'[m_1]を含む最終の復号された平文チャンクMC'[m] = (M'[m_1])を、
M'[m_1] = cut_s(F_K((N,Tw_m_1),0^n)) xor C[m_1]
ただし、s=nのときはcut_s()は省略可能
と求めさせ、復号検証用タグ計算処理で、もしs=nであれば、復号された平文のチェックサムを、最終の復号された平文チャンクを除く復号された各平文チャンクに含まれる平文ブロックM'[i_2]とM'[m_1]とを用いて計算し、得られたチェックサムをSUM'、第3の認証タグ用補助変数を(N,Tw_T_3)、暗号化関数をF_K(*,*)とすると、復号検証用の認証タグT'を、
T' = F_K((N,Tw_T_3), SUM')
と求めさせ、もしs<nであれば、復号された平文のチェックサムを、最終の復号された平文チャンクを除く復号された各平文チャンクに含まれる平文ブロックM'[i_2]とM'[m_1]をnサイズにパディングした結果であるM_n'[m_1]とを用いて計算し、得られたチェックサムをSUM'、第4の認証タグ用補助変数を(N,Tw_T_4)、暗号化関数をF_K(*,*)とすると、復号検証用の認証タグT'を、
T' = F_K((N,Tw_T_4), SUM')
と求めさせる付記２４または付記２５に記載の認証暗号用プログラム。
【０２３３】
（付記２７）暗号化関数が、Tweakと呼ばれる補助変数を含む2変数入力のTweakable ブロック暗号である付記２１から付記２６のうちのいずれかに記載の認証暗号用プログラム。
【０２３４】
（付記２８）暗号化関数が、入力される第1の変数と第2の変数とを連結したものを入力とする、鍵付きハッシュ関数である付記２１から付記２６のうちのいずれかに記載の認証暗号用プログラム。
【０２３５】
この出願は、２０１３年８月２日に出願された日本特許出願２０１３－１６１４４６を基礎とする優先権を主張し、その開示の全てをここに取り込む。
【産業上の利用の可能性】
【０２３６】
本発明は、無線もしくは有線のデータ通信における暗号化とメッセージ認証、データベースなどストレージの保護といった用途に好適に適用できる。
【符号の説明】
【０２３７】
30,60　認証暗号手段
10,61　暗号化手段
101　入力手段
611　平文入力手段

102,612　補助変数生成手段
103,613　2ラウンドFeistel暗号化手段
104,614　タグ計算手段
105　出力手段
20,62　復号手段
201　入力手段
621　暗号文入力手段
202　補助変数生成手段
622　復号用補助変数生成手段
203,623　2ラウンドFeistel復号手段
204,624　復号検証用タグ計算手段
205,625　判定手段
206　出力手段

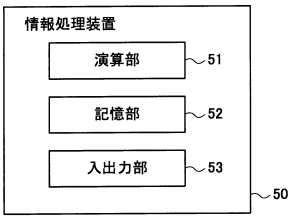
【図１】



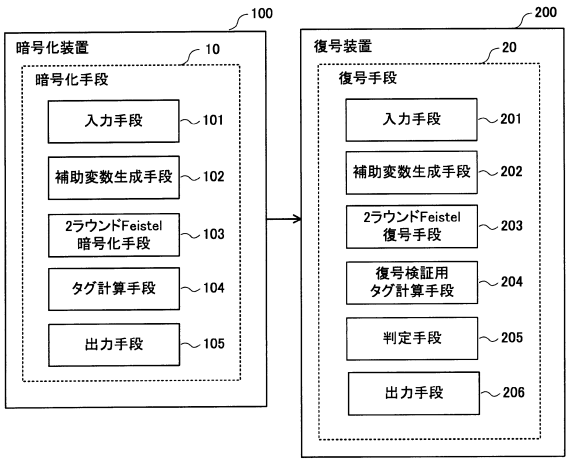
【図２】



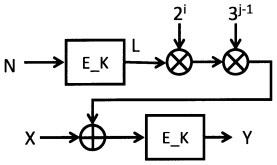
【図３】


情報処理装置
演算部
51
記憶部
52
入出力部
53
50

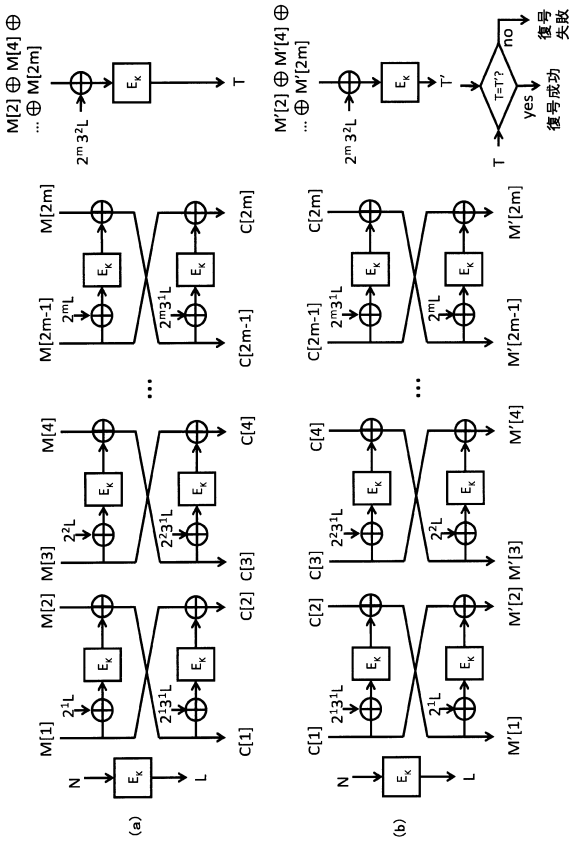
【図４】


100
暗号化装置
10
暗号化手段
入力手段
101
補助変数生成手段
102
2ラウンドFeistel
暗号化手段
103
タグ計算手段
104
出力手段
105
200
復号装置
20
復号手段
入力手段
201
補助変数生成手段
202
2ラウンドFeistel
復号手段
203
復号検証用
タグ計算手段
204
判定手段
205
出力手段
206

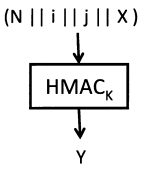
【図５】


N
E_K
L
2^i
3^j-1
X
E_K
Y

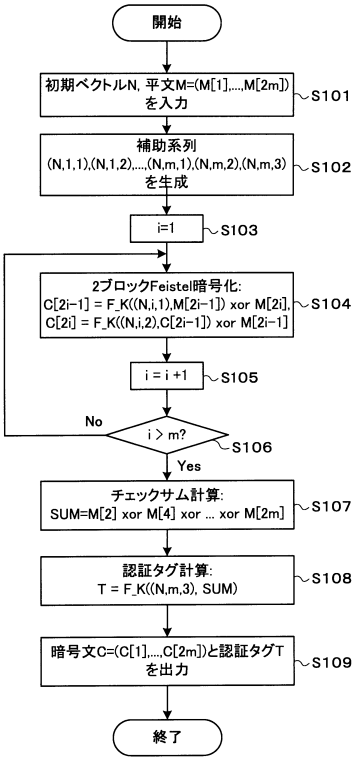
【図６】


M[2] ⊕ M[4] ⊕ ... ⊕ M[2m]
T
M'[2] ⊕ M'[4] ⊕ ... ⊕ M'[2m]
T'
T=T'?
no
復号失敗
yes
復号成功
M[2m]
C[2m]
M[2m-1]
C[2m-1]
M[4]
C[4]
M[3]
C[3]
M[2]
C[2]
M[1]
C[1]
M'[2m]
M'[2m-1]
M'[4]
M'[3]
M'[2]
M'[1]
N
L
(a)
(b)

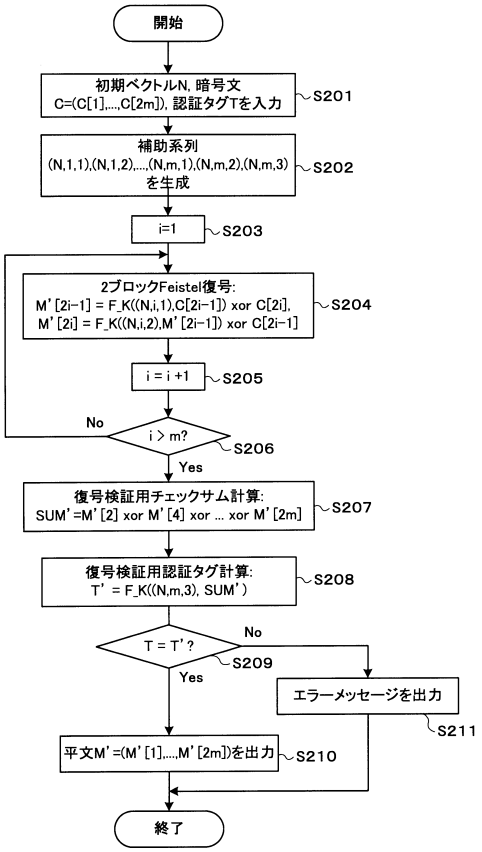
【図７】


(N || i || j || X)
HMAC_K
Y

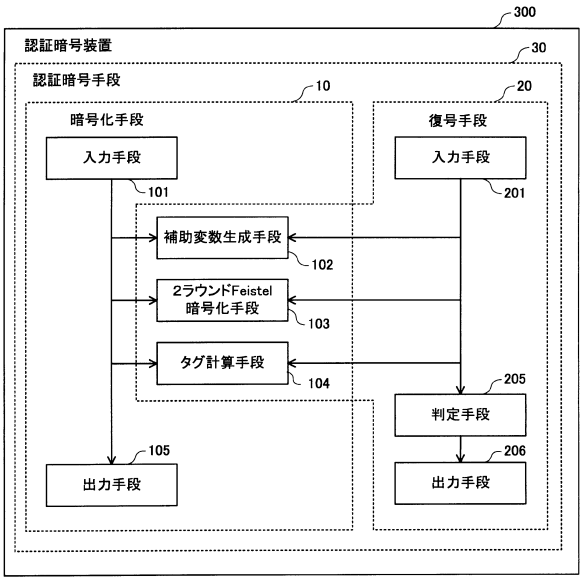
【図８】


開始
初期ベクトルN, 平文M=(M[1],...,M[2m])を入力
S101
補助系列(N,1,1),(N,1,2),...,(N,m,1),(N,m,2),(N,m,3)を生成
S102
i=1
S103
2ブロックFeistel暗号化:
C[2i−1] = F_K((N,i,1),M[2i−1]) xor M[2i],
C[2i] = F_K((N,i,2),C[2i−1]) xor M[2i−1]
S104
i = i +1
S105
No
i > m?
S106
Yes
チェックサム計算:
SUM=M[2] xor M[4] xor ... xor M[2m]
S107
認証タグ計算:
T = F_K((N,m,3), SUM)
S108
暗号文C=(C[1],...,C[2m])と認証タグTを出力
S109
終了

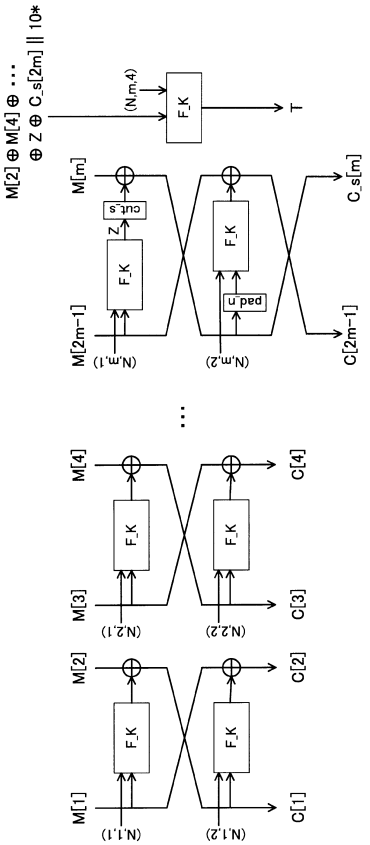
【図９】


開始
初期ベクトルN, 暗号文C=(C[1],...,C[2m]), 認証タグTを入力
S201
補助系列(N,1,1),(N,1,2),...,(N,m,1),(N,m,2),(N,m,3)を生成
S202
i=1
S203
2ブロックFeistel復号:
M'[2i−1] = F_K((N,i,1),C[2i−1]) xor C[2i],
M'[2i] = F_K((N,i,2),M'[2i−1]) xor C[2i−1]
S204
i = i +1
S205
No
i > m?
S206
Yes
復号検証用チェックサム計算:
SUM'=M'[2] xor M'[4] xor ... xor M'[2m]
S207
復号検証用認証タグ計算:
T' = F_K((N,m,3), SUM')
S208
T = T'?
No
S209
Yes
エラーメッセージを出力
S211
平文M'=(M'[1],...,M'[2m])を出力
S210
終了

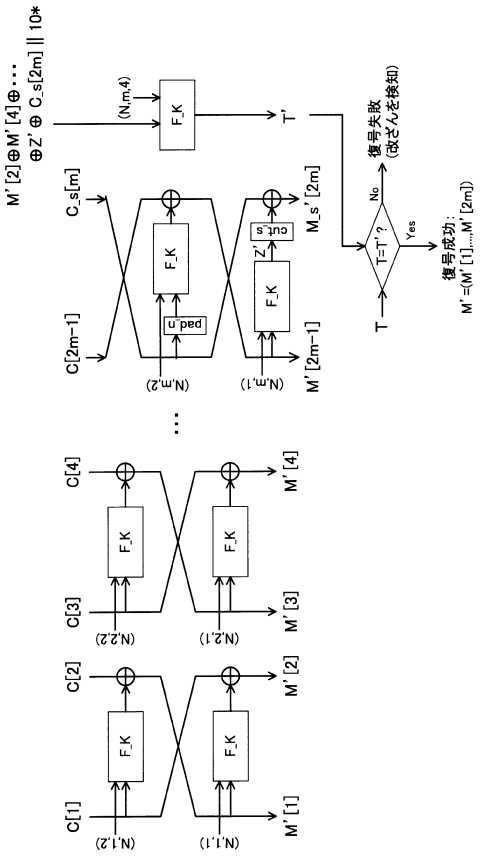
【図１０】


300
認証暗号装置
30
認証暗号手段
10
暗号化手段
入力手段
101
補助変数生成手段
102
2ラウンドFeistel暗号化手段
103
タグ計算手段
104
出力手段
105
20
復号手段
入力手段
201
205
判定手段
206
出力手段

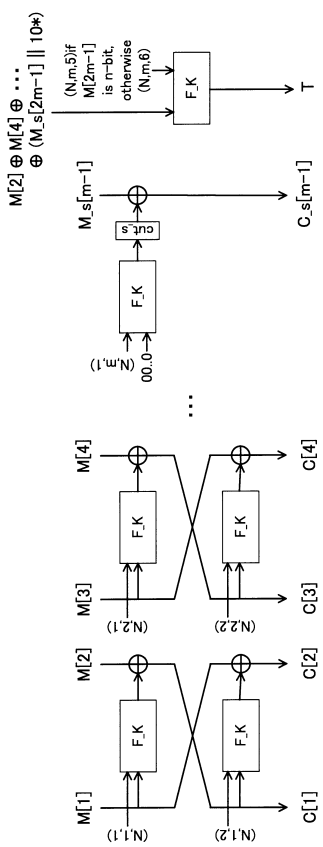
【図１１】


M[2] ⊕ M[4] ⊕ ... ⊕ Z ⊕ C_s[2m] || 10*
(N,m,4)
F_K
T
M[m]
C_s[m]
cut_s
Z
F_K
F_K
pad_n
M[2m−1]
C[2m−1]
(N,m,1)
(N,m,2)
M[4]
C[4]
M[3]
C[3]
(N,2,1)
(N,2,2)
M[2]
C[2]
M[1]
C[1]
(N,1,1)
(N,1,2)

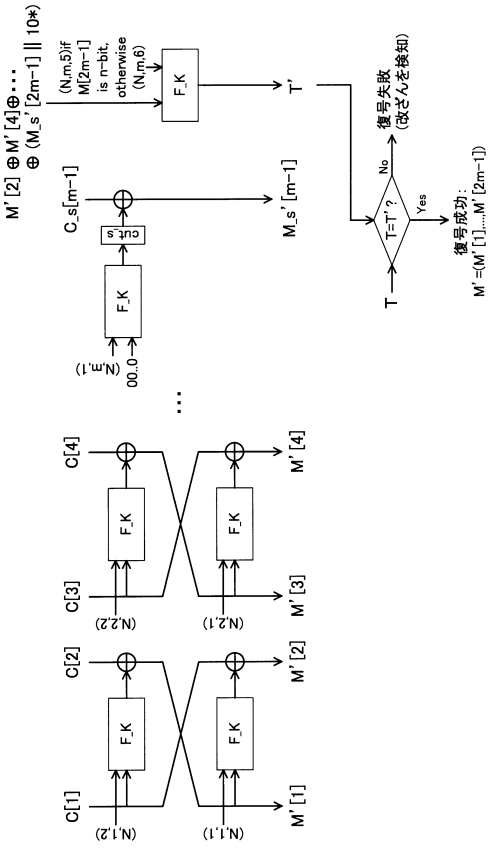
【図１２】



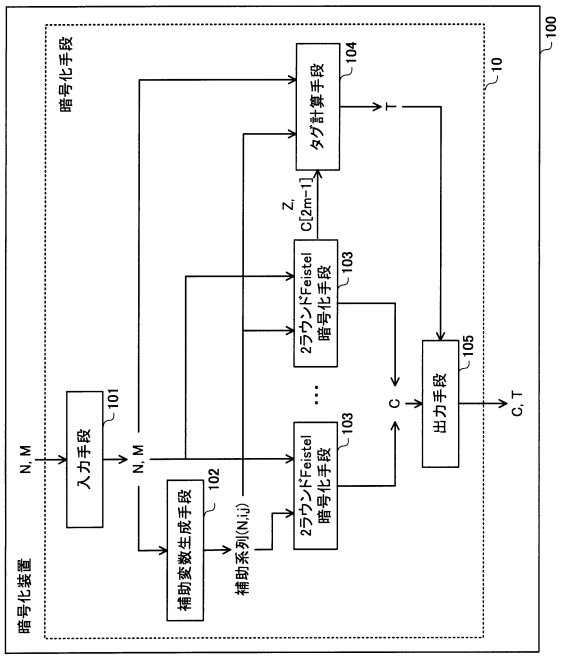
【図１３】



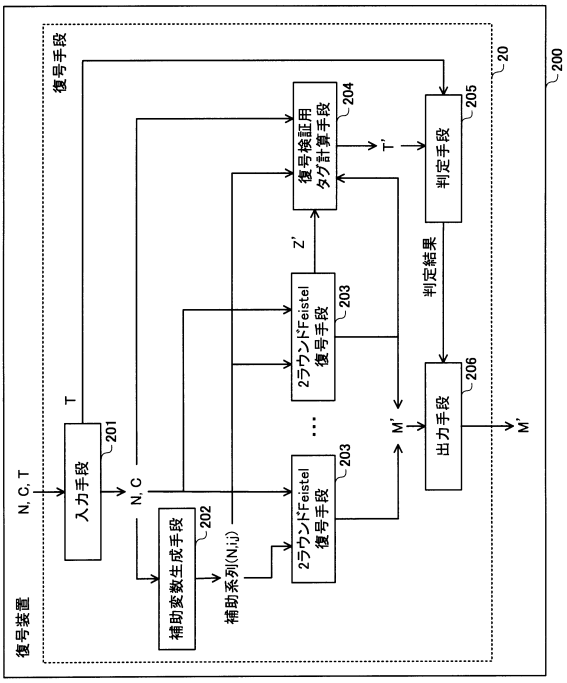
【図１４】



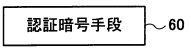
【図１５】



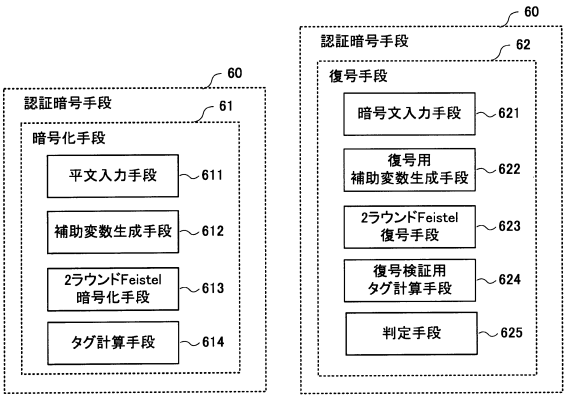
【図１６】



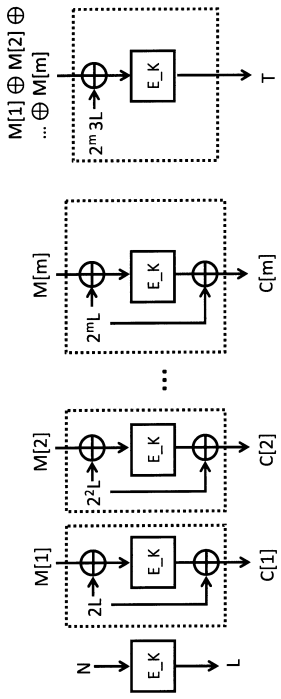
【図１７】



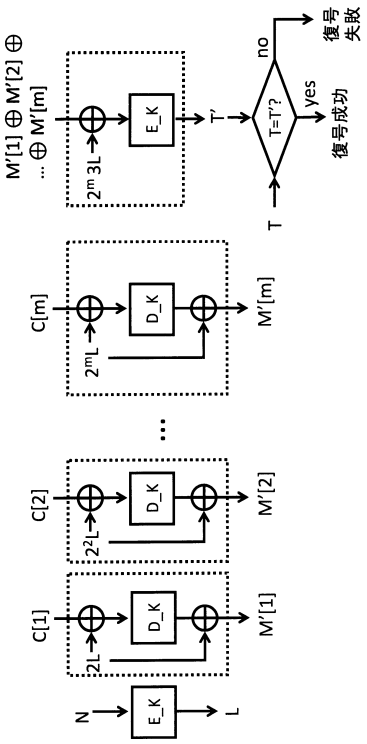
【図１８】



【図１９】



【図２０】



フロントページの続き

(56)参考文献 米国特許第０５６２３５４９（ＵＳ，Ａ）
米国特許出願公開第２００６／０２８５６８４（ＵＳ，Ａ１）
ANDERSON, E. et al., Manticore and CS mode : parallelizable encryption with joint cipher-state authentication, SANDIA REPORT, Sandia National Laboratories, ２００４年１０月１日, SAND2004-5113, 2018年7月26日検索, ＵＲＬ, http://www.osti.gov/scitech/biblio/919631

(58)調査した分野(Int.Cl., ＤＢ名)
Ｈ０４Ｌ　　９／０６
Ｇ０９Ｃ　　１／００
Ｈ０４Ｌ　　９／３２
ＪＳＴＰｌｕｓ／ＪＭＥＤＰｌｕｓ／ＪＳＴ７５８０（ＪＤｒｅａｍＩＩＩ）
ＩＥＥＥ Ｘｐｌｏｒｅ